Usable Security Heuristics for Instant Messaging Application Development

C.M. van Niekerk

2022

Usable Security Heuristics for Instant Messaging Application Development

By

Craig Michael van Niekerk

Submitted in fulfilment of the requirements for the degree of Master of Information Technology to be awarded at the Nelson Mandela University

April 2022

Supervisor: Professor Lynn Futcher

Declaration

I, Craig Michael van Niekerk 216035929, hereby declare that the dissertation for Master of Information Technology to be awarded is my own work and that it has not previously been submitted for assessment or completion of any postgraduate qualification to another University or for another qualification.

Craig Michael van Niekerk

NELSON MANDELA
UNIVERSITY

PERMISSION TO SUBMIT FINAL COPIES OF TREATISE/DISSERTATION/THESIS TO THE EXAMINATION OFFICE

Please type or complete in black ink

FACULTY: Engineering, the Built Environment and Information Technology

SCHOOL/DEPARTMENT: School of Information Technology - Centre for Research in Information and Cyber Security (CRICS)

I, (surname and initials of supervisor) Futcher L

and (surname and initials of co-supervisor) ____________________

the supervisor and co-supervisor respectively for (surname and initials of candidate) van Niekerk CM

(student number) 216035929 a candidate for the (full description of qualification)

Master of Information Technology

with a treatise/dissertation/thesis entitled (full title of treatise/dissertation/thesis):

Usable Security Heuristics for Instant Messaging Application Development

It is hereby certified that the proposed amendments to the treatise/dissertation/thesis have been effected and that **permission is granted to the candidate to submit** the final copies of his/her treatise/dissertation/thesis to the examination office.

	07/03/2022
SUPERVISOR	**DATE**

And

CO-SUPERVISOR	**DATE**

Abstract

As instant messaging (IM) applications have become more popular, the privacy and security concerns associated with their usage has become ever more relevant. As with many software programs, IM applications have a history of security vulnerabilities. Although IM application usage is globally increasing, it has been found that currently no generally recognised standards exist to aid IM application developers when developing the usability of the security features they implement. The problem is further exacerbated as research suggests that typical users have neither the requisite understanding of the available IM security features, nor the capacity to make full use of those protection features. The primary objective of this study is to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications. This primary objective is further divided into several secondary objectives, which collectively aim to address the proposed problem. Therefore, the secondary objectives are to determine IM security risks and their related implications on users; to identify and investigate existing security and usability heuristics, guidelines, standards and best practices for mobile application development; to map the identified security and usability heuristics, guidelines, standards and best practices to IM applications; and to develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical IM application. First, a comprehensive literature study is used to determine and understand the information security threats relevant to IM applications, how IM applications operate, the security features implemented by IM applications and the potential impact the relevant information security threats could have on IM application users. Thereafter, a further literature review and content analysis are used to identify and investigate existing heuristics, guidelines, standards, and best practices for mobile application development. The findings from the content analysis, in combination with the previously identified threats to IM applications, are then mapped to IM applications, and a preliminary set of usable security heuristics for IM application development is established. This preliminary set of usable security heuristics undergoes multiple iterations of refinement to establish the proposed set of usable security heuristics for IM application development. Furthermore, an expert review is conducted to validate the proposed set of usable security heuristics from the perspectives of security, usability, and mobile application development. In addition, the expert review was also used to determine the efficacy, utility, and quality of the proposed usable security heuristics. To further validate the proposed heuristics, a proof-of-concept prototype is used, in addition to the expert review, to demonstrate the applicability of the proposed set of usable security heuristics to a typical IM application. Such a set of usable security heuristics would be useful for IM application developers and would result in the

improved implementation of usable security, leading to an improvement in the security of IM applications. The proposed set of usable security heuristics therefore adds a further contribution to this research area, providing a solid foundation for future research.

Acknowledgements

I would like to express my sincere gratitude to:

- My supervisor, Professor Lynn Futcher, for her excellent guidance, support, incisive input, time and patience in helping me complete this dissertation.
- The Postgraduate Research Scholarship for the financial assistance they provided.
- My parents, Anton and Linda Barnes, for their support and motivation throughout this dissertation.
- My girlfriend, April Stroud, and her family for their support and motivation throughout this dissertation.
- My sister Meredith van Rooyen and her family for their support and guidance throughout this dissertation.
- The expert review participants for their time and valuable input.
- My proof-reader and editor, Ms Ricky Woods, for the hours she put into reviewing and editing this dissertation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CIA	Confidentiality, Integrity and Availability
HTTP	Hypertext transfer protocol
IEC	International Electrotechnical Commission
IM	Instant Messaging
IMEI	International Mobile Equipment Identity
iOS	iPhone Operating System
IP	Internet protocol
ISO	International Organization for Standardization
ISPs	Internet Service Providers
MiTM	Man-in-The-Middle Attack
NIST	National Institute of Standards and Technology
NMU	Nelson Mandela University
OECD	Organisation for Economic Co-operation and Development
OS	Operating System
OTP	One Time Pin
OWASP	Open Web Application Security Project
POCP	Proof-of-concept prototype
PRO	Primary Research Objective
REC-H	Research Ethics Committee: Human
SABS	South African Bureau of Standards
SDLC	Software Development Life Cycle
SMS	Short Message Service
SPIM	Instant Messaging Spam
SQL	Structured Query Language
SRO1	Secondary Research Objective One
SRO2	Secondary Research Objective Two
SRO3	Secondary Research Objective Three
SRO4	Secondary Research Objective Four
SSDF	Secure Software Development Framework
SSL	Secure socket layer
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VOIP	Voice Over IP

Chapter 1 - Introduction

1.1 Introduction

The purpose of this chapter is to offer the necessary background information as well as the research objectives of the study. This is accomplished by documenting the following:

- background to this study's research area,
- a description of the problem area,
- the problem statement for this study,
- the research objectives of the study,
- the research process of this study,
- ethical considerations for this study,
- delineation of this study,
- layout of chapters.

This chapter structure is as follows. Section 1.2 introduces the background for the research area of this study, while Section 1.3 provides a description of the problem area. Section 1.4 documents both the primary and secondary objectives identified in order to address the problem. Section 1.5 focuses on the research methods relating to each of the objectives specified for this study. Section 1.6 highlights the research process, while Section 1.7 presents the ethical considerations of the study. Section 1.8 presents the delineation of the study, while Section 1.9 documents the layout of the chapters within this study. Lastly, Section 1.10 concludes this chapter.

1.2 Background

Cybersecurity is an area that focuses on defending against unauthorised access, alteration or degradation of computers, databases, programs, and networks by providing the ability to control cyberspace behaviour and rules. This requires sufficient information about Information and Communications Technology (ICT) stability, weaknesses and vulnerabilities and essential operational factors in cyberspace (Awan & Memon, 2016). It is achieved through a combination of innovative technology and the understanding of the human user (Pfleeger & Caputo, 2012).

Cybersecurity-related incidents can be traced back to the 1980s. Espionage, theft and hacking were recorded as cyber-related incidents from these times (Warner, 2012). In today's technologically enhanced age, government agencies, corporations, hospitals, financial institutions and other groups collect and store personal information online and transmit it over the internet (Khari et al., 2017). The constant increase of user information being transmitted, stored and processed online results in a need for protection against unauthorised access and use of this information (Voskoboinicov & Melnyk, 2018). As technology advances, three main vulnerability categories have emerged, namely human vulnerabilities, hardware vulnerabilities, and software vulnerabilities. The human category refers to the accidental or deliberate actions taken by a user that could lead to a compromise of security systems; the hardware category includes vulnerabilities related to the physical systems and the inherent vulnerabilities fall within the software category, which includes any vulnerabilities from software installed or software run on a device (Gcaza et al., 2017).

In comparison to computers, mobile devices have been found to have weaker security capabilities (He, 2013). In 2018, high-risk vulnerabilities were found in mobile applications, with 38 per cent relating to the iPhone Operating System (IOS) and 43 per cent relating to Android applications. However, these high-risk vulnerabilities were not inherently the product of some single vulnerability. In certain cases, they were the result of many seemingly small deficiencies in various parts of mobile applications. Taken together, these deficiencies create a larger vulnerability. Most cases were due to vulnerabilities in features (74 per cent and 57 per cent respectively for iOS and Android devices, and 42 per cent for components on the server side). These vulnerabilities typically creep in during the design stage and fixing them often requires major code changes. These vulnerabilities could have significant implications like financial damages for customers and reputational damage for developers (Ptsecurity.com, 2019). Additionally, these vulnerabilities could affect end users significantly.

A problem identified with applications is a gap between what the system expects of users with regard to security, and what users could actually accomplish (Mujinga et al., 2013). This is particularly true for mobile applications which also require a balance between user experience (UX) and security. Part of the issue with this balance is that users have vastly different knowledge of security and computers and varying levels of computer literacy (Rajivan et al., 2017).

A study was conducted between 2011 and 2015 by OECD (the Organisation for Economic Co-operation and Development), as cited in Nielsen (2016), to assess the computer skills of the adult population (people aged between 16 and 65 years). The study included 215942 participants from 33 countries. The results of this study showed a general lack of computer skills in the adult population (Nielsen, 2016), creating a gap between developers' expectations and what end users are able to achieve. This gap needs to be reduced, or users will continue to demonstrate a range of different responses to security notifications and warnings (Rajivan et al., 2017) leading to an increase in cybersecurity attacks.

In the current technologically advanced age, usability and user experience (UX) are critical attributes that need to be prioritised, as they have an influence over a users' selection of a particular product, system or service (Li, et al., 2020). Bevan et al. (2016) define usability as

> *'the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use' (pp. 269)*.

Additionally, they defined UX as

> *'a person's perceptions and responses resulting from the use and/or anticipated use of an interactive system, and from the user's interaction with the organisation that supplies or delivers the interactive system; from discovering the system, adopting and using it, through to final use' (pp. 271)*.

The usability and user experience of a software application can be determined through a heuristic evaluation. Heuristics are used to guide human judgement, problem solving and the decision-making process of users (Bhatia, 2015). One of the most widely used methods of evaluation is the heuristic evaluation. This can be implemented on actual operating systems or performed during the development of interactive systems. Heuristics are typically developed to assist developers in the creation of a User Interface (UI) and to assist heuristic evaluators when using a UI (Bonastre & Granollers, 2014). Heuristics are also commonly known as principles for user interface design, guidelines, user interface design patterns and standards (Masip et al., 2012).

In 1990, Nielsen, in collaboration with Molich, developed a set of usability heuristics (Nielsen & Molich, 1990). In 1994, Nielsen further refined these heuristics to obtain a set of heuristics with the highest explanatory strength. This revised set of heuristics resulted in Nielsen's ten usability heuristics for user interface design (Nielsen, 1995). Nielsen's innovative usability heuristics formed the foundation for most later established heuristics in the areas of usability and usable security.

Pribeanu (2017) states that users no longer want products that just satisfy their needs; they want a fully secure and usable system. With this in mind, software developers are now tasked with determining the type of users, in which conditions the system would work, and the main characteristics of the system. If all of these criteria are met, users would view the system as efficient, safe and convenient (Bitkina et al., 2020). However, securing a user's data and personal information involves a high level of complexity. This level of complexity often causes users to avoid interacting with the security and privacy features available to them, which results in the system and related information becoming vulnerable (Nimgaonkar & Kumbhar, 2020). There is therefore a need for usable security, especially with regard to widely used and popular mobile applications, like instant messaging. Caputo et al. (2016) define usable security as *'delivering the required levels of security and also user effectiveness, efficiency, and satisfaction' (pp. 3)*.

One way to compromise a computing system is to take advantage of technical flaws and vulnerabilities. However, the most effective and reliable way to break into a computing system is to get a user to make bad decisions, thereby compromising the system (Joyce, 2016). The end user is therefore often viewed and labelled as the weakest link in the security of any system (Still et al., 2017)

In the same way that a heuristic evaluation is used to identify any usability design issues associated with a user interface, so too can such an evaluation be used to identify any usable security design issues. However, currently a generally accepted set of usable security heuristics for Instant Messaging application development does not exist.

1.3 Description of Problem Area

Companies and users are seeing many benefits from mobile devices and their applications, as they provide users with portability, location awareness and accessibility (Nayebi et al., 2012). Instant Messaging networks offer a service called instant messaging (IM), which allows the transfer of text messages to other users or user groups in real time. Most instant messaging applications also allow users to share images or arbitrary files (Paul & Hof, 2016).

Statista.com released statistics, for 2021, stating that there were 4.79 million mobile apps available for download from the Apple App Store and 2.79 million available for download from the Google Play Store (Statista, 2021c, 2021b). Among all these applications, Instant Messaging (IM) apps were the most popular. Similarly, Statista.com provides statistics relating to the global monthly active users of the top IM applications. In July 2021 these were as follows: WhatsApp, 2000 million users; Facebook Messenger, 1300 million users; WeChat, 1242 million users; QQ Mobile, 606 million users; Telegram 550, million users; and Snapchat, 514 million users (Statista, 2021a). Thus, IM applications were likely to be used by the majority of mobile users.

As IM applications became increasingly popular, the privacy and security concerns associated with their usage became ever more relevant (De Luca et al., 2016). Just like every other software program, popular IM apps have a history of common security vulnerabilities. Potential threats to IM applications include confidential information leakage, surveillance and retention problems, and distribution of malicious code. Installing an IM application often introduces these threats to a computer system or device (Nyakomitta et al., 2016).

An example of confidential information leakage is presented by Jagwani (2016), who conducted a case study on WhatsApp and noted that all messages and information about connections were stored on servers, which are typically managed by the IM network provider. In general, IM communications and correspondence occur in plain text, rendering them vulnerable to eavesdropping. Nowadays, most IM features have a certain degree of encryption. However, this is not sufficient for protecting confidential information sent across a network.

Surveillance and retention problems are typically linked to IM communications occurring in plain text (Jagwani, 2016), making them vulnerable to eavesdropping (Nyakomitta et al., 2016). The attackers (malicious users) generally benefit from stolen passwords that provide access to other user accounts. In addition, attackers benefit from gathering confidential information and then selling it or exploiting it to fulfil their malicious intentions (Fahrnberger, 2015).

Distribution of malicious code, like viruses, in IM applications has gained rapid attention as attackers shift their emphasis from better-protected email systems to these IM networks (Nyakomitta et al., 2016). The rise in short text messaging, along with unlimited text messaging, makes malicious messages popular and barely cost the attackers anything to distribute. This, combined with the confidence that users instinctively have in their mobile devices, has made users vulnerable to such attacks (Almeida et al., 2016). Some of these threats could be defined as Instant Messaging Spam (SPIM), which is a form of spam that specifically targets users of instant messaging services (Odukoya et al., 2018).

In addition to the above-mentioned IM threats, another threat is an IM application's permission system. Android implemented a specific authorisation system to prevent apps from accessing computer resources and users' sensitive data in an unauthorised way (Liu et al., 2019). The permissions that an app requests are normally related to providing the required functionality of the app. However, some apps are intentionally hungry for permission, facilitating greater access to a user's personal data and information. Besides software developers and marketers potentially benefiting from this behaviour, mobile protection and privacy may be compromised if highly privileged devices have vulnerabilities that can be exploited (Taylor & Martinovic, 2016).

Most IM users have neither the requisite understanding of the available security features, nor the capacity to make full use of such security features. For example, when downloading free Android applications, users often grant dangerous permissions (Li & Clark, 2013). This is generally true for mobile apps and is therefore applicable to IM applications too. By making the interaction with the security features of a system more transparent and less strenuous, users would implement the security controls available (Still et al., 2017). However, the pressure placed on users to handle complicated security options gives attackers the ability to exploit the difference between the perception and the reality of danger (Li & Clark, 2013). Therefore, it is necessary for app developers to recognise and mitigate the concerns of mobile users about their confidentiality (Degirmenci, 2020). A system which has more usable security features is more controllable and reliable, making it more usable to the users. Thus, greater usability of an IM application's security features eliminates uncertainty, making it more efficient and secure (Nimgaonkar & Kumbhar, 2020).

In order to mitigate the various threats to IM applications, developers integrate various security features into these applications. However, IM users are often not able to apply the built-in IM security features effectively. In addition, they are often not aware of the risks in these applications, owing to the lack of usability of the security features provided.

The problem statement for this study is therefore stated as follows:

Many developers of instant messaging mobile applications do not consider the usability of the security features they implement, thereby exposing the users of these applications to unnecessary risk.

1.4 Research Objectives

To address the problem stated above, the following primary research objective was defined for this study:

To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications.

The secondary research objectives identified to achieve the primary research objective of this study are as follows:

1. To determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users (SRO1).
2. To identify and analyse existing security and usability heuristics, guidelines, standards and best practices for mobile application development (SRO2).
3. To map the identified security and usability heuristics, guidelines, standards and best practices to instant messaging application development (SRO3).
4. To develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application (SRO4).

1.5 Research Methods

The research methods utilised in this study are briefly discussed in this section. The research methods identified to attain the primary and secondary research objectives, as established by this study, are presented in Table 1.1.

Table 1.1 Research Methods Associated with the Research Objectives

	Research Objective	**Research Method**
PRO	To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications.	Critical reasoning/argumentation and Expert review
SRO1	To determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users.	Literature review

SRO2	To identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development.	Literature review and Content analysis
SRO3	To map the identified security and usability heuristics, guidelines, standards, and best practices to instant messaging application development.	Critical reasoning/argumentation
SRO4	To develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application.	Critical reasoning/argumentation and Prototype

The research objectives listed in Table 1.1 are discussed in the subsections below.

1.5.1 Critical Reasoning/Argumentation

Critical reasoning recognises the existence of a natural order in social events. It claims that this social order cannot only be observed through the pattern of events that unfold. The underlying order needs to be discovered through the interpretation process (Walliman, 2010). Critical reasoning is also referred to as argumentation. Van Eemeren and Grootendorst (2003) define argumentation as *'a verbal, social, and rational activity aimed at convincing a reasonable critic of the acceptability of a standpoint by putting forward a constellation of propositions justifying or refuting the proposition expressed in the standpoint' (pp. 1).*

In terms of this study, the aim of critical reasoning and argumentation is to aid in the argument towards the creation of a set of usable security heuristics to assist developers of IM applications to consider the usability of the security features implemented in these applications, and to map the identified security and usability heuristics, guidelines, standards, and best practices to IM application development. In addition, critical reasoning and argumentation is used in the development of the prototype to demonstrate the applicability of the proposed set of usable security heuristics to a typical IM application.

1.5.2 Literature Review

A literature review is utilised as an introduction to the topic at hand and provides important information on why a topic is worth pursuing. It consists of an appraisal of the relevant existing literature (Walliman, 2010). Webster and Watson (2002) state:

> *'A review of prior, relevant literature is an essential feature of any academic project. An effective review creates a firm foundation for advancing knowledge. It facilitates theory development, closes areas where a plethora of research exists, and uncovers areas where research is needed' (pp. 13).*

This highlights the importance of a literature review to any research study.

This research utilised a variety of relevant sources, including the National Institute of Standards and Technology (NIST), to perform the literature reviews, as shown in Table 1.1. NIST is a government technologies organisation that works with industry to create and implement technology, measurements, and standards. Various International Organization for Standardization (ISO) standards are also included in this research. ISO is a global standard-setting organisation made up of delegates from multiple national standards organisations. In addition, relevant academic articles and books were sourced from Elsevier, Emerald Insight, Google Scholar, Institute of Electrical and Electronics Engineers (IEEE), ResearchGate, Science Direct and Springer.

The aim of the literature review conducted in this study was to determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users, and to identify existing security and usability heuristics, guidelines, standards, and best practices for mobile application development.

1.5.3 Content Analysis

According Krippendorff (2004) a content analysis is *'a research technique for making replicable and valid inferences from texts (or other meaningful matter) to the contexts of their use' (pp. 18).* As a research methodology, it is designed to draw replicable and true inferences from texts (or other relevant matter) to their implementation contexts. As a technique, the study of the material requires advanced procedures which are learnable and can be divorced from the researcher's personal authority. A content analysis, as a research tool, offers new perspectives, improves a researcher's knowledge of specific phenomena, or informs realistic behaviour. It can be either qualitative or quantitative in nature, but for this study a qualitative content analysis was conducted, which provided an understanding of the concepts, thoughts and experiences shared in the literature studied.

The aim of the content analysis conducted during this study was to analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development, as presented in Chapter 5.

1.5.4 Expert Review

Kovesdi and Joe (2017) define an expert review as *'the evaluation of a system, by a subject matter expert, against a standardised set of evaluation criteria' (pp. 1262).* Expert reviews generally build on heuristic tests by reviewing the design, not only for compliance with heuristics, but also against other established usability criteria, usability-related concepts (such as cognitive psychology and human–computer interaction), and the expertise of the reviewers with previous experience in the field. The emphasis on the previous experience of the reviewer and knowledge of the concepts of usability is why this form of design review is often referred to as an expert review.

The purpose of the expert review in terms of this research was to validate the preliminary set of usable security heuristics for instant messaging application development. Five experts were used to conduct the expert review. The selection criteria for the experts included a combination of relevant experience in security, usability, and mobile application development. In this study, the expert review was utilised to determine the quality, efficacy, and utility of the proposed set of usable security heuristics for instant messaging application development, as presented in Chapter 7. To accomplish this, the experts assessed the set of usable security heuristics, highlighted possible issues, and offered suggestions for improvement and other comments. In this way, the expert review was utilised to improve the credibility and validity of the proposed set of usable security heuristics for instant messaging application development.

1.5.5 Prototyping

A prototype is a type of model that appears to function because it could look and/or act in a similar way to the targeted design. Prototypes are typically produced for analysis, demonstration, or research purposes. Prototyping can also be used to test that design elements are correct (Norgren, 2004; Hess, 2012; Jobbins, 2012). Prototypes can take several forms (Sapin & Duy, 2011; Hess, 2012; Jobbins, 2012), from sketches on paper (Norgren, 2004) or cardboard mock-ups (Robinson, 2005) that reflect limited elements of a design, to more complex prototypes that embody multiple elements (Dunne et al., 2004), for example, online working models or graphical user interfaces (Jobbins, 2012). Proper use of prototyping often increases the efficiency of the production process (Norgren, 2004). The classical prototype of software simulates a graphical user interface as an aid to achieving and validating specifications with the customer or users (Jobbins, 2012).

Software prototyping is referred to as the process of creating an early, incomplete version that includes the essential elements of the final product on which later versions can be based (Kristoffer & Vasbotten, 2016). Prototype development is seen as a process that starts by defining the requirements for the product. This means understanding the very basic software specifications, particularly regarding the user interface. Priority is normally given to the prototype look and feel (Kristoffer & Vasbotten, 2016).

The aim of prototyping in this research was to develop a prototype to demonstrate the applicability of the proposed set of usable security heuristics to a typical instant messaging application. For this study, a proof-of-concept prototype was developed and presented in Chapter 8.

The following section discusses the research methods highlighted as they relate to the research process followed during this study.

1.6 Research Process

This section describes the research process for this study as depicted in Figure 1.1. The discussion follows the logical flow through this research process diagram.

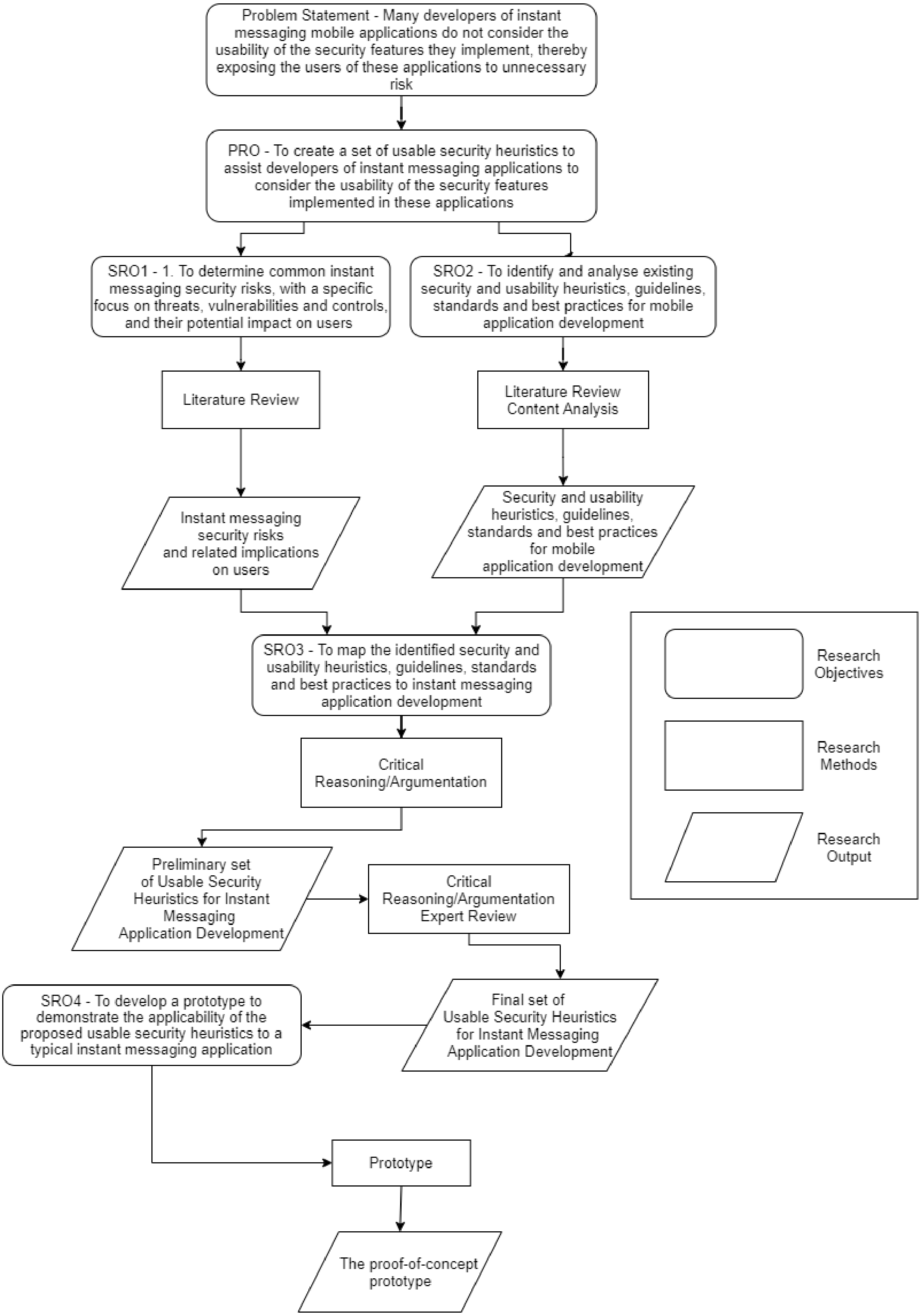

Problem Statement - Many developers of instant messaging mobile applications do not consider the usability of the security features they implement, thereby exposing the users of these applications to unnecessary risk
PRO - To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications
SRO1 - 1. To determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users
SRO2 - To identify and analyse existing security and usability heuristics, guidelines, standards and best practices for mobile application development
Literature Review
Literature Review Content Analysis
Instant messaging security risks and related implications on users
Security and usability heuristics, guidelines, standards and best practices for mobile application development
SRO3 - To map the identified security and usability heuristics, guidelines, standards and best practices to instant messaging application development
Critical Reasoning/Argumentation
Preliminary set of Usable Security Heuristics for Instant Messaging Application Development
Critical Reasoning/Argumentation Expert Review
Final set of Usable Security Heuristics for Instant Messaging Application Development
SRO4 - To develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application
Prototype
The proof-of-concept prototype
Research Objectives
Research Methods
Research Output

Figure 1.1 Research Process

The research process for this study started with a preliminary literature review to define the problem to be addressed. This led to the problem statement that *many developers of instant messaging mobile applications do not consider the usability of the security features they implement, thereby posing unnecessary risk to users of these applications*. To address this problem, the following PRO was identified, namely: *to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*. To support the PRO, a further four secondary research objectives were determined.

The research continued with a thorough literature review to ascertain common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users (SRO1). To accomplish this, common information security threats, which are relevant to application security, were examined. The application security threats were then assessed against IM application security, thereby identifying the most common IM application security threats. Thereafter, these IM applications were further assessed, with a specific focus on IM application security and the identified IM application security threats. This helped in determining the relevant common IM security threats which lead to IM security risk.

A further literature review and a rigorous content analysis was conducted to identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development (SRO2). The literature review helped in identifying and defining existing security and usability heuristics, guidelines, standards, and best practices. The content analysis process consisted of four main steps (Planning, Data Collection, Data Analysis and Reporting of results). The results from the content analysis, in combination with the identified IM security risks and their related implications on users, were utilised in achieving the research output for SRO3.

The research process continued with critical reasoning and argumentation to map the identified security and usability heuristics, guidelines, standards, and best practices to instant messaging application development (SRO3). The development of the proposed set of usable security heuristics followed a four-step process, located in Chapter 6, Section 6.2.

Having developed the preliminary set of usable security heuristics, an expert review was conducted to determine the validity of the preliminary set of usable security heuristics for instant messaging application development. The set of usable security heuristics for instant messaging application development was evaluated by experts in the fields of usability, security, and mobile application development. The feedback from the experts was examined and critical reasoning and argumentation were utilised to assess whether changes were required to the preliminary set of usable security heuristics. Based on the results of the expert review, any changes implemented to the preliminary set of usable security heuristics were highlighted and the finalised set of proposed usable security heuristics for instant messaging application development were presented. This resulted in the research output of a final set of usable security heuristics for instant messaging application development and the accomplishment of the PRO.

To further validate the final set of usable security heuristics, a proof-of-concept prototype was developed to demonstrate the applicability of the proposed set of usable security heuristics to a typical instant messaging application (SRO4). Each usable security heuristic was applied to the typical IM application. The combination of the proof-of-concept prototype and the validated usable security heuristics fully met the requirements of the PRO by presenting a full set of validated and applicable security heuristics usable security for instant messaging application development.

1.7 Ethical Considerations

According to the Nelson Mandela University Research Ethics Committee NMU REC-H committee, faculty level ethical clearance was required for this study, since it involved human participants for the expert review. Ethical approval to conduct the expert review was granted by the Nelson Mandela University Research Ethics Committee: Human, with the ethical approval reference number H21-ENG-ITe-006. The research conducted for this study adhered to the ethical principles and guidelines presented in the Belmont Report (National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research, 1979).

1.8 Delineation

This study focused on developing a set of usable security heuristics for IM application development. It specifically targeted non-malicious developers, who aim at improving the usability of the security features and controls they implement in IM applications.

1.9 Layout of Chapters

The layout of chapters for this study, along with a brief overview of each chapter, is presented in Table 1.2.

Table 1.2 Chapter Layout

Chapter Title	**Chapter Overview**
Chapter 1 - Introduction	This chapter provides a background to the research area of this study, a description of the problem area, the problem statement, the research objectives, the research process, ethical considerations, and delineation of this study.
Chapter 2 – Information Security	This chapter provides broad context to this study by discussing the importance of information security and identifying common security threats.
Chapter 3 – Instant Messaging	This chapter discusses instant messaging, what it is, how it works, the differences between the current popular instant messaging applications available, and the threats facing instant messaging.
Chapter 4 – Instant Messaging Security	This chapter discusses instant messaging security and privacy controls and features, and the role that these controls and features play in protecting instant messaging users, both individuals and corporations.
Chapter 5 – Security and Usability Heuristics, Guidelines, Standards and Best Practices	This chapter analyses existing security and usability heuristics, guidelines, standards, and best practices for mobile application development.
Chapter 6 – Proposed set of Usable Security Heuristics for Instant Messaging Application Development	This chapter maps the identified security and usability heuristics, guidelines, standards, and best practices for instant messaging application development, as highlighted in Chapter 5. Based on this mapping, it proposes a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications.

Chapter 7 – Validation of the set of Proposed Usable Security Heuristics	This chapter validates the proposed set of usable security heuristics presented in Chapter 6.
Chapter 8 – Proof-of-Concept Prototype	This chapter provides a proof-of-concept prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application.
Chapter 9 – Conclusion	This chapter concludes the study by summarising each chapter and motivating how each of the research objectives specified in Chapter 1 were attained.

1.10 Conclusion

Usable security is vital to securing IM applications and the confidential information of their users. Therefore, there is a demand for usable security solutions in IM application development. Combining this with the fact that the current skill level of typical IM users is not up to the expectation of developers, and that IM users want a fully secure and usable system, it is clear that a set of usable security heuristics for IM application development is needed to assist developers in implementing usable security features. This research presents such a set of usable security heuristics for IM application development.

In order to provide broad context to this study, the following chapter discusses information security and its related implications for users.

Chapter 2 – Information Security

2.1 Introduction

Chapter 1 introduced the background and problem area. It highlighted the gap in literature and the need for this gap to be addressed. The research process for this study was therefore designed to address this gap.

The preservation of information from unauthorised and malicious parties is called information security. Information security highlights the expectations and requirements that organisations are expected to meet, when working with confidential information (Aldawood & Skinner, 2019b; Mujinga et al., 2019). Information security makes it possible for individuals to understand these expectations and requirements, as well as assisting them in identifying organisations that are not up to standard. By maintaining strong and up-to-date information security, organisations and individuals can address threats and risks associated with information assets (Alkhudhayr et al., 2019; Rai et al., 2020).

The aim of this chapter is to provide broad context to this study by discussing the importance of information security and identifying common security threats. This chapter links to the partial completion of SRO1, *to determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users.*

The chapter structure is as follows: Section 2.2 introduces information security with a specific focus on its underlying goals, while Section 2.3 highlights some of the more common threats to information security. Section 2.4 provides a discussion regarding the identified threats and their related vulnerabilities and Section 2.5 concludes this chapter.

2.2 Information Security

Nowadays, the majority of organisations and individuals are interested in technology services to accomplish processes faster than by conventional methods. In order to make their systems more effective, the information that is stored, processed and transmitted must be secured from threats and information security must be preserved (Alkhudhayr et al., 2019). In addition, organisations are put under increased pressure by new legislative requirements to ensure that they practise sufficient levels of information security within their organisation (Antoniou, 2018).

All information that an organisation maintains, uses and processes is subject to numerous risks, including attacks, unintended errors, and natural disasters. The term information security focuses on information being deemed to be an object with a value that requires sufficient protection. Providing accurate and complete information to those with an approved need in a timely manner is a catalyst for business efficiency (International Organization for Standardization, 2018). Protecting an organisation, their information and their customer information highlights the importance of information security (Alkhudhayr et al., 2019).

In order to ensure the confidentiality, integrity and availability of information and operating procedures, information security is characterised as the protection of the hardware, the system and the information used, stored and transmitted (National Institute of Standards and Technology, 2017; Alkhudhayr et al., 2019; Paliszkiewicz, 2019; Rai et al., 2020). The main goals of information security are to guarantee that the confidentiality, integrity and accessibility of that information are preserved (Alkhudhayr et al., 2019). These goals are referred to as the CIA triad or the CIA model. They are seen as a means to categorise attributes and controls of information security to achieve security results (Brooks et al., 2017; Nweke, 2017; Alkhudhayr et al., 2019; Covert et al., 2020).

The C in CIA stands for **Confidentiality**. Information security demands data and information privacy. While confidentiality is similar to privacy, it is not the same thing. Privacy is required in order to maintain information security and to ensure that confidentiality is maintained. Confidentiality is a privacy component that protects information from unauthorised access. Any individual, device or process that is not authorised to access information, files, and objects (such as usernames, password combinations, medical records, etc.) should be restricted from accessing them. Confidentiality is about accessing data or information, and if the wrong people access data or information that they are not authorised to access, this can lead to complications, including but not limited to, theft of account credentials, acquisition of customer information or distribution of extremely confidential organisational information (International Organization for Standardization, 2013a; 2016; 2018; Nweke, 2017; Sinha et al., 2019).

The I in CIA stands for **Integrity**. Information security requires organisations and individuals to be secure in the information distributed, interpreted, and retained. In addition, organisations and individuals need to be assured that the information will not be changed from its original state inadvertently or maliciously. If there is a change in one bit of a message, for example, the entire message can change. The corruption or unreadability of the whole message can also result from this. The purpose of integrity is to ensure the reliability, accuracy and completeness of the transmitted, stored, relayed or received information (International Organization for Standardization, 2016; 2018; Nweke, 2017; Sinha et al., 2019).

The A in CIA stands for **Availability**. Availability means that with all the security features in place to deal with infrastructure, software, individuals, and systems, it should be possible for authorised users to access these resources as needed. This assures that authorised users should be able to access the tools and resources required to accomplish their work easily, while ensuring that the tools and resources are accessible in the event of an information security breach or disaster (International Organization for Standardization, 2016; 2018; Nweke, 2017; Sinha et al., 2019).

The following terms and concepts are taken from relevant ISO/IEC standards, namely ISO/IEC 27005 (International Organization for Standardization, 2011) and ISO/IEC 21827 (International Organization for Standardization, 2008). These terms and concepts, together with their respective relationships, are illustrated in Figure 2.1, and they include:

- **Asset** - anything that an organisation or individual deems as valuable.
- **Threat** - a possible cause of an undesirable incident that can damage an individual, system or organisation.
- **Threat agent** - the original creator and/or facilitator of intentional or inadvertent man-made threats.
- **Vulnerability** - a weakness found in an asset or collection of assets that can be exploited by one or more threats.
- **Control** - means of risk management that can be institutional, technical, administrative, or legal in nature, including policies, protocols, instructions, practices or organisational strategies (also known as protections or countermeasures).

- **Risk** - potential for a given threat to manipulate the vulnerabilities of an asset or collection of assets to cause loss or harm to the assets.

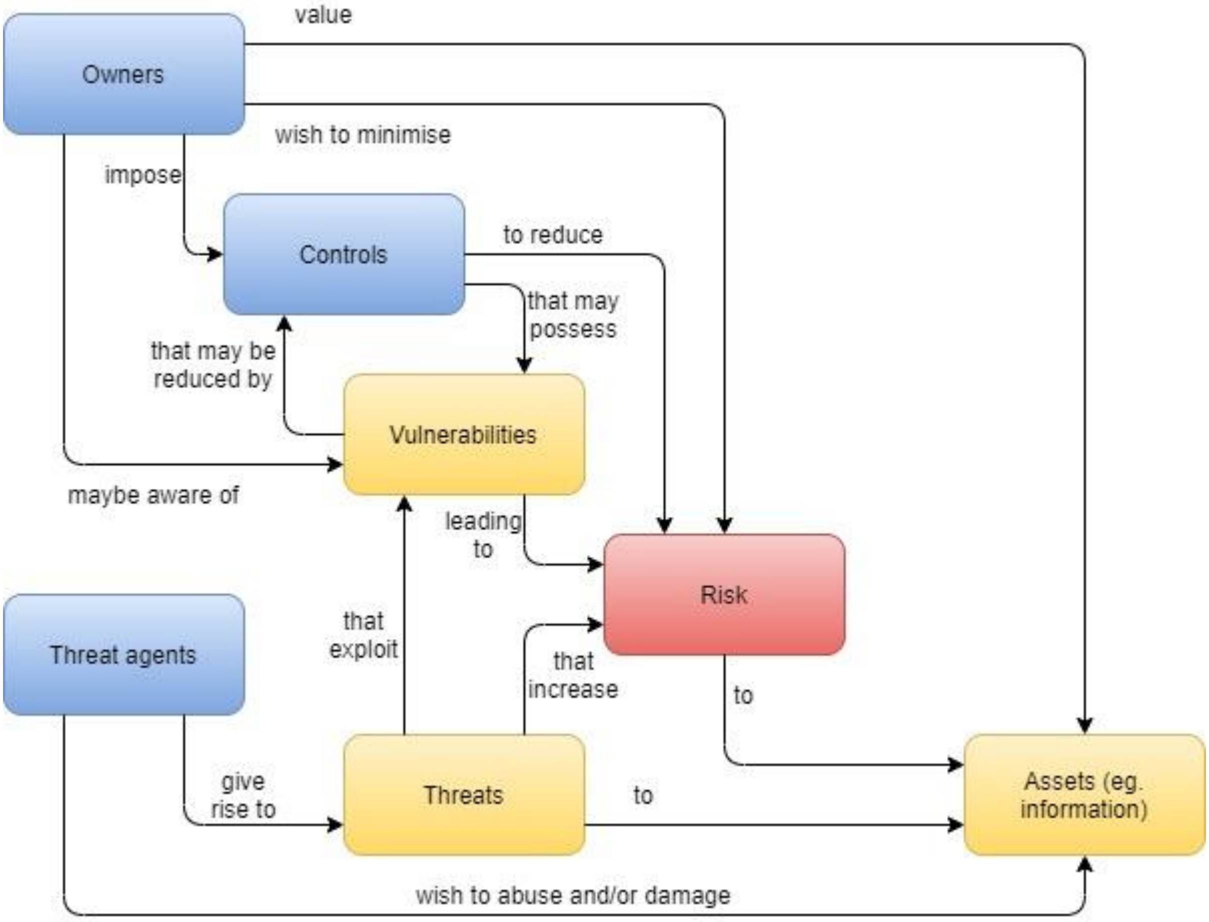


Figure 2.1 Key elements relating to Information Security Risk (adapted from (Common Criteria) cited in Task Force Report (National Cyber Security Partnership, 2004))

Protecting assets is the aim of security. It is evident in Figure 2.1 that the assets that require the most protection are those on which someone places value (owner, user, or threat agent). Such assets are information assets that need protection, from a software development perspective, when stored, analysed, and distributed by software applications. It is therefore necessary for software applications to comply with the security requirements defined by information asset owners. Since information assets may theoretically be valued by threat agents (malicious users), the intent of these threat agents could be to compromise or destroy them. This may amount to confidentiality, integrity, or availability being lost (Futcher, 2011).

Identifying threats and vulnerabilities in their applications is crucial for software developers. Developers are accountable for minimising risks from threat agents to information assets through implementing software controls to prevent and resolve the potential threats and vulnerabilities.

2.3 Application Security Threats

There are many other forms of security including network security, infrastructure security, cloud security, and end point security (Cisco, 2019). The focus of this study is application security.

This section discusses seven main application security threats identified through the literature review, namely: confidential information leakage, distribution of malicious code, man-in-the-middle attacks, permission system vulnerability, shrink wrap code attacks, social engineering attacks, and SQL injection attack.

2.3.1 Confidential Information Leakage

Accidental or deliberate dissemination of confidential information to an unauthorised party is known as information leakage. Examples of confidential information from individuals and organisations are intellectual property, financial information, medical information, and organisational information. This though, depending on the sector and industry, can vary. While the number of incidents and the cost to those who suffer from them continue to increase, information leakage is a widespread issue for individuals and organisations (Kaur et al., 2017).

The reality that information exchanged (both inbound and outbound), including emails, instant messaging, website settings, and file transfers, is mostly uncontrolled and unregulated on its path to its destinations, magnifies information leakage. An information leakage incident is synonymous with direct and indirect losses (Bhavani et al., 2017). Direct losses are known as identifiable damage and are easy to measure or approximate quantitatively. These damages include, but are not limited to, violations of regulations, such as those targeted at consumer protection, which can result in fines, damages or victim restitution payment cases involving claims, loss of potential prosecution transaction costs and penalties for remedies or restorations. Indirect losses are defined as losses that are more difficult to quantify and have a more widespread cost, position, and time effect. Indirect losses include, but are not confined to, a decrease in the share price arising from adverse marketing damage to the credibility and prestige of a company owing to neglect of customers and violation of intellectual property, such as business strategies, codes, financial results and meeting agendas for competitors (Bose & Vishwanath, 2016).

External attackers are not the only threat to an organisation's network security. Insiders seeking financial, political or any other type of benefit pose a large threat to an organisation (Zimmermann & Renaud, 2019). The most potentially harmful vulnerability to online protection for organisations is not malware but warmware. The Australian Government (2016) defines warmware as *'the ability of a trusted insider to cause significant disruption to a network or to use legitimate access to acquire sensitive information and then unlawfully reveal it' (pp. 3).*

In the first half of 2018, InfoWatch Analytical Center registered 1039 information leaks published in the worldwide media and other outlets, which is 12 per cent more than in the first half of 2017 (925 leaks). The information leaks compromised 2.39 billion personal and payment information records, including social security numbers, bank card details, and other critical information, compared to 7.78 billion records over the same period of 2017. External attacks were behind 35.5 per cent of information leaks, while insiders triggered 64.5 per cent of the leaks. The overall amount of information leaked during the period and the volume of information released by potential perpetrators are most likely limited owing to regulatory penalties imposed by governments (primarily in the U.S. and Europe) on organisations that failed to avoid leakages. Clearly, many such significant fines released in early 2018 forced corporate executives who were handling vast amounts of information to consider implementing improved confidential information management methods (InfoWatch Analytics Center, 2018a).

In 2020, IBM released an updated report on the cost of an information breach. The United States is reported still to have the highest average cost for information leakage, increasing from $8.19 million in 2019 to $8.64 million in 2020 (US dollars). The average total cost of an information breach decreased from $3.92 million to $3.86 million. The highest industry average cost remained in the healthcare industry, increasing from $6.45 to $7.13 million. As the root cause of an information breach, the cost of malicious attacks decreased from $4.45 to $4.27 million and human error decreased from $3.54 to $3.33 million. The average cost per lost record decreased from $150 to $146 (IBM, 2020a). The average size of an information breach for 2020 was not recorded but over 8.5 billion records were compromised in 2019 (IBM, 2020b).

As the majority of all security strategies rely heavily on human behaviour, the human aspect is important for information security (Wong et al., 2019). Legitimate flow of information causes information leakage owing to human naïveté or malicious intent and vulnerabilities in the application (Rajamenakshi & Padmavathi, 2016). These mistakes lead to a breach of two of the three CIA triad information security goals. As unauthorised individuals gain access to private information, confidentiality is infringed. Once the information is accessed by these unauthorised individuals, it is unclear whether they have manipulated or modified it in any way. This could therefore result in a violation of integrity.

From the above discussion, it is evident that the key vulnerability causing confidential information leakage is insecure user behaviour. Whether deliberate or accidental behaviour, the effect is the same, directly leading to information leakage. When developing applications, software developers should be aware of and consider this insecure user behaviour to prevent the leakage of confidential information.

2.3.2 Distribution of Malicious Code

Malware refers to self-replicating malicious software that is intended to perform undesirable acts on the network, which distributes over a network, without interaction or initiation, (Baror & Venter, 2019). In general, there are three types of methods to propagate malicious code: software vulnerability, user behaviour or a combination of these two methods. Some malicious code will initiate automatically without user interaction (Liu et al., 2016). Malware emerges in various forms including viruses, worms, Trojan horses, spyware, greyware, and spam.

Viruses and worms spread via instant messages, emails and networking using a compromised device. They disclose to the hacker private or confidential information or display it throughout the internet. Worms can change the settings, wallpapers, and so on. They often delete a user's hard drive files and folders without the administrator being aware of it. Both viruses and worms can cause software instability resulting in software showing errors whenever opened, software hanging, or software shutdown without reason. As a result of the infection, a user's computer will start becoming really slow, which makes processing very difficult (Jagadish et al., 2019). Most worms do not damage their host during replication, in order to spread more effectively. The host can not necessarily know that it has been contaminated by a virus or worm. When the user browses a webpage, opens an email or IM attachment, or connects an infected device to their system that contains these malicious codes, the host will become infected by this code (Liu et al., 2016).

Trojan horses, also referred to as trojans, are programs that appear useful but are not. Trojans potentially damage the device of the user, compromise its information, disrupt the stability of the device, or typically inflict some other malicious behaviour on user information or computer network (Ljuban, 2021). For example, an extension to a web browser may appear useful to the user, but it may steal passwords and other confidential information entered by the user (Reddy, 2019). Trojans can be supplied and installed easily (Agham, 2016). A trojan does not replicate itself by keeping the victim unaware of the attack. These kinds of malicious codes are concealed within some regular mail or IM attachment or some free programs like games (Johansen, 2020; Prasad & Rohokale, 2020).

Spyware is malware that, without the permission of the user, is remotely installed on a device. Spyware can be found in freeware and is freely accessible to anyone on the internet. The key purpose of spyware, without the user's knowledge, is to obtain information about the target. When enabled, spyware assists an attacker to obtain a range of intimate personal details about the monitored target. Spyware monitors user activity and can provide access to private information including instant messages, emails, browser history, images, videos, incoming and outgoing phone calls, GPS coordinates, banking or other account passwords and social media profiles, both in real time and remotely. Spyware can hinder a user’s control over their device and allow an attacker to download additional malware, divert browsers to malicious content, redirect advertising revenue to a third party, or alter device settings, or alter assorted landing pages, which often results in poor or unreliable functionality of the device, slow communication speeds. Spyware is often used stealthily while the software is blatantly used in other situations to threaten, harass, or blackmail the monitored target (Ramakrishnan & Tandon, 2018; Khoo et al., 2019; Prasad & Rohokale, 2020).

Greyware is another threat to mobile apps, in addition to mobile malware. Advertising fraud apps, for example, may be categorised as greyware as these apps contain irritating, unwanted or hidden habits that cannot be categorised as malware (Zhao et al., 2020). The key purpose of greyware is to gather user information for the purpose of profiling, which will be utilised to send marketing information back to the user. The goals of grayware distributor companies, however, are not to hurt users; instead, they provide the host user with some kind of functionality and significance. If the information collection process of a grayware is problematic, users may complain and block the grayware services. Unlike malware and spyware, the illegal use of grayware in many developed countries is punished by fines rather than prevented by any personal statements. That is why the boundaries of legality and illegality are often called grayware (Faisal et al., 2019).

Spam remains a significant vector for transmission because, unlike low-volume high-value cybercrime targeting banks and financial services and requiring sophisticated hacking skills, spam enables malware to reach high-volume low-value targets that are less likely to have successful antivirus or other countermeasures in place (Alazab & Broadhurst, 2017). Baror and Venter (2019) define spam as *‘Unsolicited junk messages, images and advertisements that are sent by every possible electronic means available, including email, blogs, search engines, instant messaging and smartphones’ (pp. 507).* Spammers spread spam using botnets and virus-infected networks. Spam sometimes includes a malicious attachment or a link to legitimate websites which have been compromised for web attack. A recent criminal innovation involves attacking devices indirectly by disguising intrusions through an intermediate website (sites that are likely to be visited by the target), which also hosts malicious code on the homepage (Alazab & Broadhurst, 2017).

Malware is among the most damaging pieces of software that can attack a device or network. When malicious code reaches some of the network's systems, it can harm the entire network, which will inevitably lead to system failure (Jagadish et al., 2019). Malware is able to delete documents or collect confidential information found on a device or network, without permission (Prasad & Rohokale, 2020). It is one of the key examples of intelligent design systems that can trigger security threats and can be described as programs that propagate and compromise different types of vulnerabilities in host networks (Jagadish et al., 2019).

Distribution of malicious code violates all three of the information security goals defined by the CIA triad. The malicious code allows unauthorised individuals to access and manipulate private information, resulting in a breach of confidentiality. When the private information is manipulated or altered, owing to the breach in confidentiality, integrity is potentially breached. When executed, malicious code may lead the device, system or network where it is found to slow down or become unusable. As authorised users are unable to access the tools and resources that are required to do their job, or the process of accessing their tools and resources is becoming slow, this is a violation of availability.

The key cause of system or device infection with malware, is insecure user behaviour and vulnerabilities in software. Both the user's behaviour, not observing the defined protocols and most likely exposing the network to malware, and developers not recognising and mitigating their software's vulnerabilities, leave it in a vulnerable state. It is the duty of network administrators to ensure that network and device security is periodically updated, while software developers can direct the behaviour of users to prevent users from triggering or enabling malicious code to spread. Such human behaviour may be deliberate or accidental. In both human behaviour and software vulnerabilities, keeping system and device security updated will help to minimise infection.

2.3.3 Man-in-The-Middle Attacks

Man-in-the-middle attacks refer to the monitoring of a network, device, or system activities, to acquire confidential information. Man-in-the-middle is an active network attack in which an attacker is remotely positioned to capture, transmit, and receive interactions between two or more parties. The attacker is able to imitate one or all parties involved to access information (Rotem & Segev, 2018; Taleqani et al., 2018; Alwazzeh et al., 2020; Prasad & Rohokale, 2020; Symeonidis & Lenzini, 2020).

MiTM attacks are often successful owing to the nature of the American Standard Code for Information Interchange (ASCII)-based hypertext transfer protocol (HTTP) and data transfer. The MiTM attack intercepts two or more systems of communication. The software uses various strategies to create a secure socket layer (SSL) connection with the attacker and the attacker creates another SSL connection to the web server. When users visit the system through a web browser, the browser will alert the user that the digital certificates used are not legitimate, but the warning could be dismissed owing to the lack of information. In certain cases, it is likely that the alert does not appear. Mobile devices continuously connect to remote services. Many of these are vulnerable with plain text information that is accessible to third parties during transit. This displays confidential details and leaves the device vulnerable to attacks through Man-in-the-middle (Chong et al., 2018).

MiTM attacks compromise two of the three CIA triad-defined information security goals. The attacks allow an unauthorised individual to access, duplicate, store or alter confidential information on a network, thus compromising confidentiality. In addition, they can manipulate or modify network data, resulting in a potential violation of integrity. These attacks are made possible by the inability of users to authenticate their incoming messages despite the existence of messaging platforms (Rotem & Segev, 2018).

The primary cause of MiTM attacks is weak network security, obsolete systems and insecure network administrator, developer, and user behaviour. There is a broad range of insecure network administrator behaviour, including doing everything manually, making changes without logging them in change control, allowing ports and protocols outbound to the internet, and failing to update policies frequently (Rayome, 2017). Network administrators need to ensure that the insecure behaviour committed by administrators, developers, and users does not impact their ability to provide adequate security. Similarly, there is a wide variety of unsafe behaviour among developers, namely no integrity checking, no certificate validation, hard-coded IP address bindings, constructing SQL statements upon user input, and establishing unsecure connections (Nguyen et al., 2017; Rahman et al., 2019; Rahman, Rahman, & Williams, 2019). Furthermore, there are different types of insecure user behaviour, such as continuing to choose weak passwords, seeking to reuse passwords, and largely applying one-factor authentication controls (Dempsey & Kelliher, 2018; Rishika & Damodaran, 2020). Developers need to guarantee that insecure user behaviour is catered for by the application they develop.

2.3.4 Permission System Vulnerability

The most recent privacy-related cases of mobile providers have highlighted the dilemma of mobile device information security and privacy considerations that is faced by app stores and distributors. Privacy violations can discourage users from downloading an app and can potentially contribute to the deletion of an app (Degirmenci, 2020). Smartphones have become a repository of highly confidential user information, which mobile applications regularly collect and manipulate (Diamantaris et al., 2019). The unprecedented access provided to apps opens a new route to mobile privacy infringements (Gu et al., 2017). Protecting confidential user information from unauthorised access is essential (Reardon et al., 2019).

Smartphone operating systems (OS) enforce permission-based controls to protect access to device resources and information. Smartphones are used as general all-purpose devices. This requires them to have access to various critical resources (location, microphone and camera), confidential end-user information (user credentials, email and contacts) and various permanent identifiers (International Mobile Equipment Identity (IMEI)) (Reardon et al., 2019; Momen & Fritsch, 2020). The permission-based control is extensively used to restrict each application's operations and the user information and device resources that the application can access (Bagheri et al., 2015; 2018; Dawoud & Bugiel, 2019).

An application's permission request is a request to access user information or device resources. If granted, an application can manipulate user information and device resources to obtain its desired result, for example: vibrate the device, access GPS location, or read contact information. The reported requests for permission notify users that an app would have access to the personal information (Gu et al., 2017; Liu et al., 2019). Currently, Apple's iPhone Operating System (iOS) and Google's Android are the two most popular smartphone platforms with Android holding the largest market share (StatCounter, 2020), and each introducing its own permission system.

iOS uses a permission system designed to prevent unauthorised operations from occurring by applications. To protect the integrity of the device, all third-party apps are run as non-privileged users who are partitioned as read-only by the OS, prohibiting apps from changing device files or making unauthorised system calls. The iOS application programming interface (API) often prevents applications from extending their permissions or accessing files belonging to other applications. For example, when an application is wanting to access a user's contacts, the user would be asked to grant or refuse the device permission first (Lutaaya, 2018; Apple Inc, 2020; Raymond et al., 2020). User information can only be accessed by the use of declared permissions that are digitally signed, which are used for unique privileged operations by some of the device apps. iOS also prohibits users from accessing information from other files by allocating them at random to a unique home directory when installed (Apple Inc, 2020; Raymond et al., 2020). Permission managers allow users to change the access privileges provided to apps installed on their mobile device. However, despite these controls, fundamental flaws restrict the degree to which users can protect their personal information (Lutaaya, 2018). Developers often embed additional permissions used to gather information for monitoring and behavioural marketing purposes, such as geolocation information and address book contacts. In these instances, such permissions are overprivileged for the intended purpose of the applications and violate the security and privacy requirements of the user (Raymond et al., 2020). Most users are overwhelmed with requests for permission and do not fully understand the implications after indiscriminately granting all requests or preventing notifications, completely entrusting their private and confidential information to all apps (Bhatt et al., 2019).

To maintain security and privacy, Android uses a permissions control that allows developers to state specifically the permissions that their applications require (Bagheri et al., 2018; Liu et al., 2019). Android applications run within a sandbox, which restricts the operations at the system level that the app can use. An app will request permission to use resources outside this sandbox, but without this permission apps cannot access or use the resources located outside this sandbox (Jain & Prachi, 2016; Bagheri et al., 2018; Dawoud & Bugiel, 2019; Liu et al., 2019). The permissions required by an app are specifically defined and declared in a manifest file shipped with the application (Liu et al., 2019). The authorisation model is based on the concept of least privilege and assumes that applications should operate at a basic level, even if users do not permit access to information that could impact their privacy. Android has introduced security controls for users to revoke application permissions to improve user control over their information (Andriotis et al., 2017; Wijesekera et al., 2018; Diamantaris et al., 2019; Reardon et al., 2019).

Users are granted the option to approve or deny an application's request for permission. This security control empowers the user and provides a sense of control over his or her information. Even with that sense of empowerment and control, users are still unaware of who is asking for their information, why they need it, and how much it will be used (Lutaaya, 2018). Enforcing permissions is not sufficient to prevent violations of security, as permissions may be mismanaged, either intentionally or unintentionally (Bagheri et al., 2015). Application requests for permission have been found to have a major impact on information privacy issues (Degirmenci, 2020). It is the responsibility of app developers to protect user information and to ease their fears about information security and privacy (Bhatt et al., 2019; Degirmenci, 2020). Users have become desensitised to unreasonable demands for permissions resulting in users mistakenly granting permissions, leaving them at the mercy of application developers and adversaries of the app (Fu, 2017; Raber & Krueger, 2017; Taylor & Degirmenci, 2020;). It is the responsibility of users to accept or deny access to confidential resources (Andriotis et al., 2017); however, users do not have the requisite information or explanations to make such important decisions. App permission requests will require a change owing to increasing privacy concerns of mobile users (Degirmenci, 2020).

Both forms of OS implement a permission system in their own unique way. However, the existence of a permission system, with the ability to grant and deny permissions, is not enough. Software developers need to ensure that the permissions they request are necessary for the application's functionality and are not to harvest user information. In addition, developers should confirm that applications can function in a limited manner if certain permissions are denied and not force users to accept potentially dangerous permission requests.

Permission system vulnerabilities violate two of the three information security goals established by the CIA triad. Confidential information is made available by an application's excessive permission requests, resulting in a confidentiality violation. The lack of confidentiality causes users to lose confidence in the information they send across the network, since they do not want this information to be manipulated in a malicious manner, which could lead to a violation of integrity.

Based on the above discussion, a combination of insecure user and developer behaviour and permission-hungry applications is the main cause of permission system vulnerabilities. Software developers develop poorly designed and permission-hungry applications to collect user information, manipulating and exploiting users for their own benefit. Users simply accept requests for permission without a complete understanding of what they are authorising. Such poorly designed applications request more permissions than required for the application functionality to be performed.

2.3.5 Shrink Wrap Code Attacks

In such an attack the gaps in poorly designed applications and unpatched operating systems are exploited by attackers. When developers write the code, applications are typically not carefully reviewed for vulnerabilities, which can leave several programming weaknesses that a hacker can manipulate. Most application development is feature driven, meaning that developers are under a deadline to churn out the most functional application, as quickly as possible. Many shared code libraries are utilised to add functionality fast. These libraries are not familiar to the average developer and can contain vulnerabilities, which can potentially lead to the exploitation of the application (Madan, 2012; Sabillon et al., 2016; Pardeshi & Pardeshi, 2020). After the initial installation, this vulnerability will be discovered early. Once discovered, hackers will potentially attempt to utilise the vulnerability to access the confidential information located in the application. It is important for developers to identify and eliminate these vulnerabilities (Sinha et al., 2019).

In terms of information security, shrink wrap code attacks violate two of the three goals specified by the CIA triad. Shrink wrap code attacks allow an unauthorised party to read, duplicate or store confidential information on a network or system, resulting in confidentiality violation. As the accessed network or system information can be manipulated and altered for malicious purposes, the violation of confidentiality results in a potential violation of integrity.

Poorly designed applications and unpatched operating systems are the key cause of shrink wrap code attacks. To mitigate the risk of vulnerabilities found in poorly designed applications, developers need to ensure that they use strong design standards for their applications. Developers need to issue updates or patches for their systems or software on a regular basis, which will help to mitigate vulnerabilities discovered after deployment.

2.3.6 Social Engineering Attacks

The art of manipulating human weaknesses to achieve a malicious objective is referred to as social engineering. Social engineering is a technique that requires no advanced specialised technology, can be used by anyone, and is affordable. In the scope of information security, attackers violate defences to access confidential information. Attackers especially target the human willingness to trust and provoke their victims to violate security protocols, which relinquish confidential information for an efficient, more tailored attack. In certain instances, victims are manipulated unwittingly to infect the system itself and compromise it. (Albladi & Weir, 2016; Beckers & Pape, 2016; Breda et al., 2017; Aldawood & Skinner, 2019b). Social engineering may be deployed in several ways, with the use of contact details, short message services (SMS), instant messaging or direct access. When performed in a proper way, social engineering can be very beneficial to the attacker (Lohani, 2019).

According to Koyun and Al Janabi (2017), social engineering can be categorised into two forms, namely human based and software based.

- **Human-based:** The attack is performed in person by an individual, hence the name human based. In other words, to get information, the attacker communicates directly with the target. The number of targets for human-based social engineering is reduced owing to lower capability relative to an automated attack (Koyun & Al Janabi, 2017).
- **Software-based:** The attacks are automated and carried out to get the desired information, with the assistance of systems and software, hence the name software based (Koyun & Al Janabi, 2017).

Although social engineering attacks differ from one another, they have a general trend of equivalent stages. In the typical pattern, four steps are included, as depicted in Figure 2.2:

Step 1: Collect information on the target.

Step 2: Establish a connection with the target.

Step 3: Manipulate the information available and execute the attack.

Step 4: Escape without any trace.

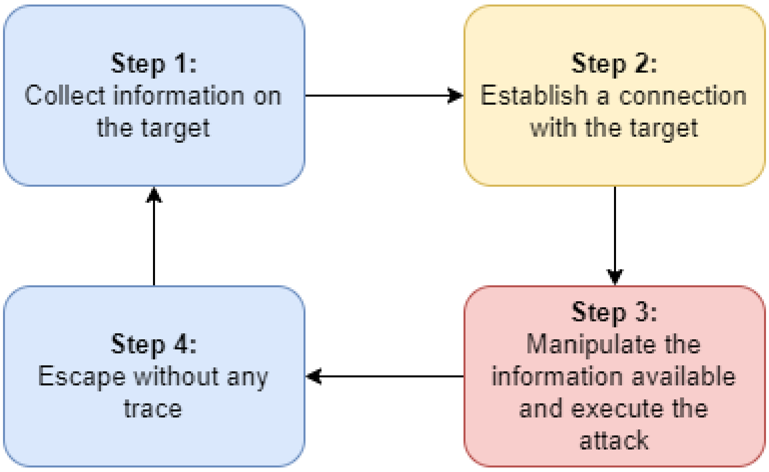

Step 1:
Collect information on the target
Step 2:
Establish a connection with the target
Step 3:
Manipulate the information available and execute the attack
Step 4:
Escape without any trace

Figure 2.2 Overview of the Social Engineering Attack Steps (adapted from (Koyun & Al Janabi, 2017; Aldawood & Skinner, 2019))

Phishing is a popular form of social engineering and will be expanded upon owing to its popularity. Phishing can be broken down into various social engineering attacks, including phishing cloning, spear phishing, and whaling. In addition to those attacks, pretexting attacks will also be examined (Lohani, 2019).

To reach a broad audience with the intention of obtaining several victims, who are particularly vulnerable to being deceived, phishing cloning utilises email, instant messaging, and other forms of mass communication. A message is delivered to the potential victims, which contains a seemingly legitimate link to a website and instructions for the victim to use the link. The victim is taken to a seemingly legitimate website and is required to enter their credentials and login, allowing the attacker to retain those credentials within their own server in a database. As an authenticated user, the attacker then redirects the victim to the trusted legitimate website. Phishing is known as one of the most powerful attacks and over the years the technique has become more advanced (Chaudhry et al., 2016; Gomes et al., 2020; Hu et al., 2020; Luse & Burkman, 2021).

Spear phishing is a targeted type of phishing that focuses, instead of targeting a large unknown audience, on targeting a single individual, community, or organisation. Spear phishing, like standard phishing, also attempts to gain personal information from the targets by forwarding them to an evidently valid website and requiring their login credentials. Spear phishing allows attackers to gain confidential and desired information on the target for attack. Attackers will obtain and use personal information about the target and construct a message that matches the target's situation and circumstances. These messages are constructed with the intention of appearing trustworthy (Yasin et al., 2019; Gomes et al., 2020; Luse & Burkman, 2021).

Another focused method of phishing is whaling. Whaling also attempts to gain personal information from the targets by forwarding them to an evidently valid website and requiring their login credentials. Whaling targets a single high-profile individual with a high level of influence or money. These targets include executives of companies, politicians, and celebrities. The attacker takes more time to attack by focusing on this small demographic. When crafting the message, the attacker is meticulous and precise to be more successful in the attack (Heartfield & Loukas, 2018; Pakhomov et al., 2019; Gomes et al., 2020).

The best example of pretexting is reverse social engineering, which uses a scripted situation to trick the victim to disclose confidential information unknowingly or to conduct other malicious behaviours. The attacker generates a scene or scenario and introduces himself as a trustworthy person who can provide support. The attacker waits for the victim to request assistance (Airehrour et al., 2018; Yasin et al., 2019; Luse & Burkman, 2021).

Integrity is undermined by social engineering, regardless of the reliability of their firewalls, encryption methods, controls for combating intrusions, and antivirus software. In comparison to computers or technology, people are more likely to trust other people. Individuals are thus deemed the weakest point in security (Salahdine & Kaabouch, 2019).

Two of the three goals identified by the CIA triad are breached by social engineering. When users are tricked into supplying the intruder with access to confidential information located on the system or network, confidentiality is breached. An intruder can utilise the confidential information to access company repositories or personal collections of information, potentially breaching integrity.

The key cause of a social engineering attack is the exploitation of the human factor and insecure user behaviour. To avoid this exploitation and insecure behaviour, software developers need to make sure that they guide human behaviour by putting adequate security controls in place.

2.3.7 SQL Injection Attack

In a SQL injection attack, the Structured Query Language (SQL) command is applied to the front- or back end of the web form field of the application. The aim is to interfere with the requests submitted to the database from the application. The attacker would be able to manipulate unauthorised information if properly executed. This information may be about other users or the application itself. This information may potentially be edited or deleted by the attacker, affecting the users and the application. From within front-end applications or database processes, these attacks are the product of a lack of adherence to good coding standards. Any website, web application or applications using a SQL database, like MySQL, Oracle or SQL Server, can be influenced by SQL injection attacks. This involves both dynamic SQL inline calls and stored procedure calls. It presents a serious threat to the users and the database as it actually impacts the information of the user by entering malicious code into the system (Sinha et al., 2019).

SQL injection attacks compromise two of the three goals identified by the CIA triad in terms of information security. SQL injection attacks allow an unauthorised party to access and manipulate confidential information stored on a system, resulting in a confidentiality violation. As the accessed system information can be manipulated and changed for malicious purposes, the violation of confidentiality could result in a violation of integrity.

Lack of adherence to good coding standards and poorly designed applications are the key causes of SQL injection attacks. To mitigate the risk of vulnerabilities found in poorly designed applications, developers need to ensure that they use secure design standards and best practices for their applications. During development, developers need to ensure adherence to secure coding practices, which will help mitigate vulnerabilities found after deployment.

2.4 Application Security Vulnerabilities

The application security threats discussed in Section 2.3.1, create application vulnerabilities. As demonstrated in Figure 2.1, in Section 2.2, threats exploit vulnerabilities to increase risk and ultimately to obtain assets. Table 2.1 contains the identified threat, the related CIA triad violation, and the main vulnerability associated with this threat.

Insecure user and developer behaviour and poor design and implementation are mutually inclusive. Software developers often design and implement applications which they deem to be acceptable and free from error. Users, however, often do not behave in a way expected by the developer. Many users are finding shortcuts and work-arounds in order to accomplish their tasks quicker, despite the costs to security. These insecure behaviours and poor designs are leading to errors, vulnerabilities and breaches in the system (Bandi, 2016; Mekruksavanich, 2017; Jongprasit & Senivongse, 2020).

Table 2.1 Application Security Vulnerabilities

Identified threat	**CIA Violation**	**Main vulnerability**
Confidential information leakage	Confidentiality Integrity	Insecure user behaviour.
Distribution of malicious code	Confidentiality Integrity Availability	Insecure user behaviour and vulnerabilities in software.
Man-in-the-middle attack	Confidentiality Integrity	Weak network security, obsolete systems and insecure network administrator, developer and user behaviour.
Permission system vulnerability	Confidentiality Integrity	Insecure user and developer behaviour and permission-hungry applications.
Shrink wrap code attack	Confidentiality	Poorly designed applications and unpatched operating systems.

Social engineering	Confidentiality Integrity	Exploitation of the human factor and insecure user behaviour.
SQL injection attack	Confidentiality Integrity	Lack of adherence to good coding standards and poorly designed applications.

Competition in the software industry, which puts pressure on organisations to produce new products and features more quickly, also causes development teams to make poor design decisions (Morales et al., 2017). Developers also take shortcuts that seem to get the work completed but weaken the quality of the design. When the design does not address certain aspects directly, it is the developer who, while coding, ends up making crucial design decisions. In such a scenario, if design standards and best practices are applied incorrectly or not applied at all, there will be flaws or vulnerabilities in the code (Suryanarayana et al., 2015).

Application security has been one of the key concerns for organisations to protect their systems against vulnerabilities. Application security defines security controls incorporated at the application level that help to avoid stolen or compromised information or code inside the application. During the development phase, much of this occurs. Hardware, software, and procedures that detect or mitigate security vulnerabilities may be part of application security. Various forms of application security are available, including authentication, authorisation, encryption, logging, application security testing, and application permission systems (Kandukuri & Srikanth, 2019; Strom, 2020; Thaduri, 2020). Developers must ensure that permission requests are made for functionality-related resources rather than to access confidential user information (Bagheri et al., 2015; Lutaaya, 2018).

The quicker and sooner a developer can identify and address security vulnerabilities during the application development stage, the stronger and more secure a user’s application will be. Since everybody makes mistakes, discovering such mistakes in a timely manner is the challenge. The demand and motivation to ensure protection, not just at the level of the network but also within the applications themselves, is rising. One explanation for this is that, with their attacks, hackers are going after applications more now than in the past. A typical coding error, for instance, might allow unverified inputs. If a hacker detects them, this error could turn into SQL injection attacks and then information leaks (Kandukuri & Srikanth, 2019; Strom, 2020). Developers must continue to monitor, identify, secure, and avoid vulnerabilities in terms of application development and release (Thaduri, 2020).

2.5 Conclusion

In order to uphold the information security goals set by the CIA triad, developers need to ensure that they meet the necessary requirements. Developers need to maintain a satisfactory degree of usability for the typical user of the security controls and features they introduce.

Two distinct aspects currently exist: the developer's expectation of what the users are able to achieve, and what the users are actually capable of. This has created a gap between the user and the developer. Users are expected to understand security on their own and to enforce it. Users are expected to do this whether they are utilising security controls in an application or maintaining system security.

The effective implementation of application security is crucial to assist in securing the apps utilised by users. When implemented correctly, application security ensures the upholding of the CIA triad.

However, insecure user and developer behaviour can contribute directly to the compromise of the CIA triad. Developers need to ensure that they develop an application that caters for potentially insecure user behaviour, by guiding users as they perform their actions, and developing an app that utilises secure coding practices.

This chapter discussed various information security threats and their potential implications, which assists in addressing SRO1. These threats will be utilised to identify vulnerabilities in instant messaging applications. Chapter 3 will discuss instant messaging, what it is, how it works, the differences between common instant messaging applications, and the threats facing instant messaging.

Chapter 3 – Instant Messaging

3.1 Introduction

In Chapter 2, information security was introduced. Information security was defined and various threats and vulnerabilities relating to information security were highlighted.

The aim of this chapter is to discuss instant messaging, what it is, how it works, the differences between the current popular instant messaging applications available, and the threats facing instant messaging. In addition to Chapter 2, this chapter links to the partial completion of SRO1: determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users.

This chapter is structured as follows. Section 3.2 describes what instant messaging is, while Section 3.3 discusses how instant messaging works. Section 3.4 introduces common instant messaging applications and compares their features. Section 3.5 highlights the threats that instant messaging applications face based on the study conducted in Chapter 2, Section 2.3. Section 3.6 reintroduces the key elements relating to instant messaging security risk. Lastly, Section 3.7 concludes this chapter.

3.2 What is Instant Messaging?

Companies and users are seeing many benefits from mobile devices and their applications, as they provide portability, location information and availability for the user (Nayebi et al., 2012). Instant Messaging (IM) is a type of computer-mediated communication that allows two or more users to transmit text-based communications or to interact privately across the internet or other networks with other users in real time. The conditions for sending an instant message are that the same IM application must be downloaded and enabled by at least two users who require access to the internet or another similar network (Rajendran et al., 2019). Users are then able to communicate back and forth synchronously and engage with each other. This differs from static communications, including email, where a message is left for the recipient to obtain at a future stage (Bruin, 2018).

With the combination of improved technologies, wide internet access availability, and the human aspect of social need, IM innovations and social skills are becoming part of the mainstream. IM has been an integral aspect of daily communication and an indispensable social interaction for many (Bruin, 2018; Huang & Zhang, 2019). Consequently, businesses are attempting to use business-centric tools to tackle the enterprise IM area (Ammirato et al., 2019). The implementation of IM applications in the workplace has resulted in a variety of findings from employees, including improvement in job efficiency, gratification with work, and social fulfilment (Sheer & Rice, 2017).

In our everyday lives and businesses, both IM technology and intelligent mobile devices have become widely used. IM applications for mobile devices enhance communication efficiency between individuals by allowing the freedom to share information with anybody at any time regardless of their location (Cai & Wu, 2018). IM applications are used not only for the purpose of communication but they are also utilised by businesses for marketing and advertisement. Owing to the expanding reach of IM today, it helps in providing businesses with an additional avenue of communication, favoured by most. Through IM, people interact directly with vendors or consumers and converse without any third-party interference (Sahoo & Gupta, 2018).

The development of IM currently accounts for billions of people from around the globe. IM removes the boundaries of standard communication, enabling people to communicate freely with one another.

IM has built a social economy and dissolved stereotypes and challenges that would previously have seemed impenetrable. Users can socialise and connect, exchange thoughts and transmit content, provide feedback and announcements, engage in online programmes and festivals, share files (photos, audio and video) and hold detailed real-time conversations around the globe (Idowu & Dominic, 2019).

3.3 How Does Instant Messaging Work?

Typically, on their devices, users install an IM application to function as a client with IM servers. The IM application allows IM clients to access IM functionality on an IM service. To receive different services, the IM client enables users to register their specific usernames and passwords and to use these credentials to connect to IM servers (refer to Step 1 in Figure 3.1). Using the correct credentials, once the client is signed in, it sends its connection information, such as its Internet Protocol (IP) address and port number, to the IM server (Barry & Tom, 2009; Larson, 2016). Figure 3.1 shows a standard instant messaging procedural model that involves an IM server, and two IM clients, Client A and Client B. Refer to Figure 3.1 for how Client A and B connect to the IM server.

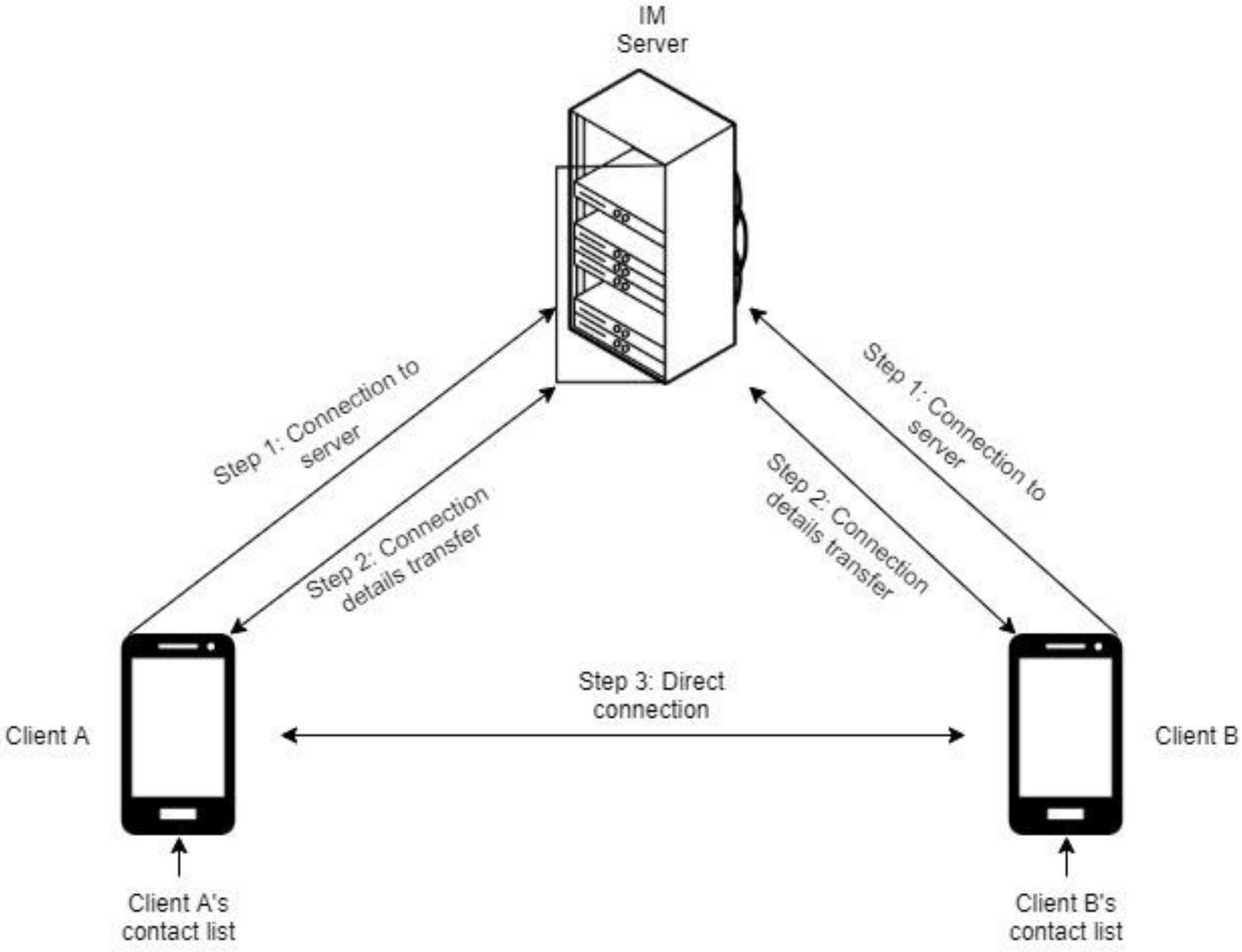

IM Server
Step 1: Connection to server
Step 2: Connection details transfer
Step 1: Connection to server
Step 2: Connection details transfer
Client A
Step 3: Direct connection
Client B
Client A's contact list
Client B's contact list

Figure 3.1 Standard Instant Messaging Procedural Model (adapted from (Barry & Tom, 2010))

The IM server generates a temporary file containing client connection details and a list of client contacts, and tests whether any of the users on the client contact list are already signed in. If the server detects that some of the client contacts are signed in, it sends the user's connection details back to the client. The server also transfers client connection details on the client contact list to signed-in users. Such connection details enable IM clients either directly, referred to as Step 3 in Figure 3.1, or via servers to send messages to the intended contact, referred to as Step 2 in Figure 3.1. Internet service providers (ISPs) play an important part in instant messaging over the internet. ISPs deliver information about client connections to the required IM servers. Alongside other information, they also transmit user messages between clients in a session (Barry & Tom, 2009; Larson, 2016). This is shown in Figure 3.1 with the two-way connection between Client A and B and the IM server, which is utilised to relay information.

Instant messaging applications authorise correspondence based on a contact list for users (Client A's and Client B's contact list), often known as a list of friends. Before correspondence will begin, one user will require the contact information of another loaded on their contact list. For instance, in Figure 3.1, Client A will have Client B's contact details loaded into their contact list or vice versa. Users are also able to create multi-participant group chats and can permit other users to invite participants from their own contact lists (Abed & Salah, 2019). For later reference, it is normally possible to save a text conversation. Instant messages are often documented in the message history of a chat, stored on the user's device, which is comparable to the continuous existence of emails (Bruin, 2018; Abed & Salah, 2019). The majority of IM applications enable users to exchange files (including images, videos, documents and audio), share hyperlinks, perform voice over IP (VOIP) phone calls and video chat (Bruin, 2018; Abed & Salah, 2019; Rajendran et al., 2019). The quality of communication, work tasks and relationships between individuals can be assisted or enhanced through IM applications (Rajendran et al., 2019).

3.4 Common Instant Messaging Applications

From the various IM applications available to users, the following are amongst the most popular: WhatsApp, Facebook Messenger, WeChat, QQ Mobile, Telegram and Snapchat (Statista, 2021a). These IM applications are available from both the Apple Store and Google Play Store. Statista.com (Statista, 2021c; 2021b) released statistics, for 2021, where 4.79 million mobile applications were available for download from the Apple App Store and 2.79 million from the Google Play Store. Among all these applications, IM applications are the most popular. Statista.com (Statista, 2021a) provides statistics of global monthly active users of the top instant messaging applications (refer to Figure 3.2). From Figure 3.2 it is clear that WhatsApp, Facebook, and WeChat are the top three instant messaging applications with over 1000 million monthly active users each. Thus, IM applications are most likely to be used by the majority of mobile users.

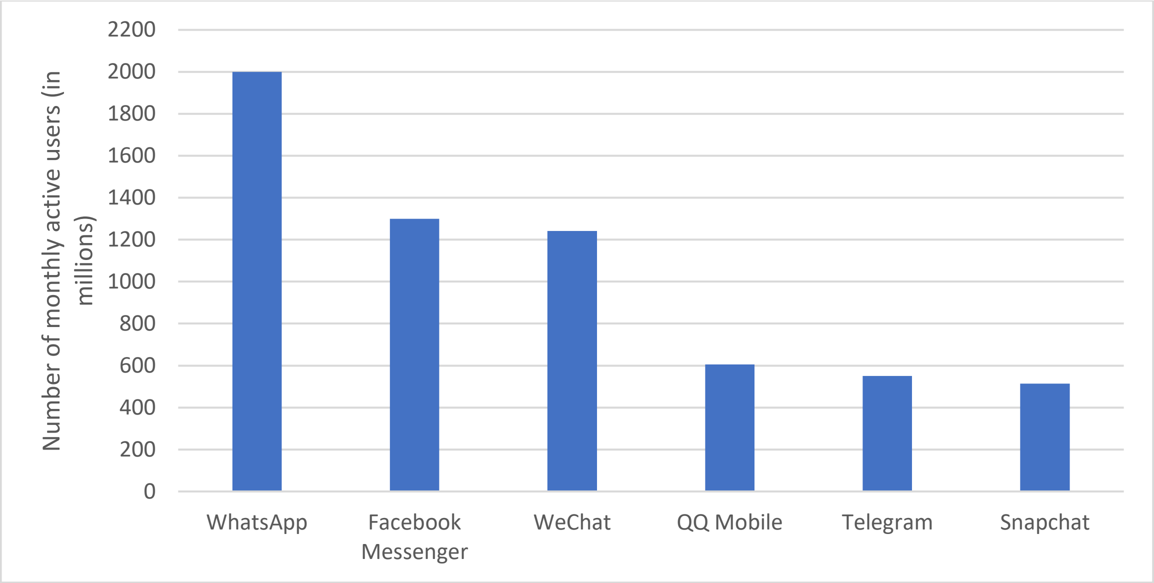


Figure 3.2 Most Popular Global Instant Messaging Applications as of July 2019 (adapted from (Statista, 2021a))

However, as IM applications have become more popular, the privacy and security concerns associated with their usage are becoming ever more relevant (De Luca et al., 2016). They have an easy, all-in-one way to access confidential information through clearly specified sources. Today, mobile phones provide location information, contacts, email, text and instant message access, as well as encrypted communications and corporate data (BlackBerry, 2019).

Both QQ Mobile and WeChat were developed by the social media corporation Tencent. QQ Mobile developed in 1999 and WeChat in 2011 (Thomala, 2020). Although QQ Mobile was developed 12 years prior to WeChat, WeChat has more than double the users. This can be attributed to the newer and more favourable features located in the WeChat application (Thomala, 2020; Statista, 2021a). QQ Mobile and WeChat are the two largest instant messaging applications utilised in Asia. QQ Mobile will be excluded as WeChat was developed as an upgrade to the application (Thomala, 2020).

Table 3.1 Comparison of Common Instant Messaging Application Features

Application features	**Facebook Messenger** (Lin, 2018;Botha et al., 2019; Facebook, 2020b, 2021a, 2021e)	**Snapchat** (Snapchat, 2017b, 2017c, 2017g, 2019, 2020a, 2020b)	**Telegram** (Telegram, 2015b; 2018; 2020c; 2020b; 2020a; Botha et al., 2019; Bhardwaj, 2020; Ricle, 2020)	**Viber** (Miroshnichenko, 2016; Bauernfreund, 2019; Cohen-Sheffer, 2019; Viber, 2019c; 2019a; 2021a; 2021b; Nash, 2020)	**WeChat** (WeChat, 2018; 2020h; 2020g; 2020i; 2020j; 2020f; 2021)	**WhatsApp** (Botha et al., 2019; WhatsApp Inc., 2020g; 2020d; 2020e; 2020f; 2020j; 2020k; 2021)
Document transfer			✓	✓	✓	✓
Group chat	✓	✓	✓	✓	✓	✓
Image, video and audio transfer	✓	✓	✓	✓	✓	✓
Voice call	✓	✓	✓	✓	✓	✓
Video call	✓	✓	✓	✓	✓	✓
Group voice call	✓	✓	✓	✓	✓	✓
Group video call	✓	✓		✓	✓	✓
Messages stored on servers	✓	✓	✓			
Information stored on device	✓		✓	✓	✓	✓
Download collected information	✓	✓				
Delete collected information		✓				
Chat history backup			✓	✓		✓

Table 3.1 contains a comparison of common features found in IM applications. It shows that Facebook Messenger and Snapchat do not support the transfer of documents. The six IM applications examined all have functionality for group chats, image, video and audio transfer, voice call and video calls. Only Telegram does not support group video calls. Facebook Messenger, Snapchat and Telegram store messages on their servers. Snapchat is the only IM application that does not store information on the local device. Facebook Messenger and Snapchat enable the user to download the information that the application has collected on them. Only Snapchat enables users to delete the information that has

been collected about them. Lastly, Facebook Messenger, Snapchat and WeChat do not allow users to create external backups of their chat history.

3.5 Application Security Threats to Instant Messaging

As stated in Chapter 2, Section 2.3.1, application security is threatened at an application security level. The effects of application security threats on IM applications are discussed in this section. The relevance of all seven application security threats identified in Chapter 2, Section 2.3.1, are considered in relation to IM.

Table 3.2 Application Security Threats Relevant to IM Applications

Threat	Relevant to IM Applications	Not Relevant to IM Applications
Confidential information leakage	✓	
Distribution of malicious code	✓	
Man-in-the-middle attack	✓	
Permission system vulnerability	✓	
Shrink wrap code attack		✓
Social engineering	✓	
SQL injection attack		✓

Table 3.2 identifies the application security threats that are relevant to IM applications. Based on the study conducted, SQL injection and shrink wrap code attacks are not relevant to IM applications.

As stated in Chapter 2, Section 2.3.1.5, for a successful shrink wrap code attack, attackers exploit the gaps in poorly designed applications and unpatched operating systems (Sinha et al., 2019). All applications are vulnerable after launch or initial installation (Sabillon et al., 2016). This highlights how shrink wrap code attacks are general for all applications and do not specifically target any individual application, which leads to their exclusion as it is not relevant to IM applications.

As stated in Chapter 2, Section 2.3.1.7, SQL injection attacks target websites, web applications and applications that utilise a SQL database, like MySQL, Oracle or SQL Server (Sinha et al., 2019). There is no literature to support IM applications using SQL databases, which results in SQL injection attacks not being relevant and therefore being excluded.

3.5.1 Confidential Information Leakage

Confidential information leakage is known as the accidental or deliberate dissemination of confidential information to an unauthorised party (Kaur et al., 2017). As stated in Chapter 2, Section 2.3.1.1, a data leakage incident is synonymous with direct and indirect losses (Bhavani et al., 2017).

The increasing acceptance in the workplace of instant messaging is no surprise. IM holds a great attraction for employees who want to connect and interact with other employees or people in real time. When individuals use instant messaging for business or leisure their communications may be monitored and traced, resulting in leakage or revealing of their personal data or confidential business information. File transfer capability, supported by practically all common instant messaging services, is utilised to send company documents and other files between employees. This usage creates an opening for programmes such as packet sniffers, specially designed to target IM applications, to intercept and display the confidential information transmitted, both files and messages, over the IM application to malicious individuals. Malicious individuals are not limited to targeting IM transmissions and communications. They can manipulate human vulnerabilities to provide the confidential information willingly, which can then be utilised for malicious purposes. Therefore, IM can contribute to the risk of confidential data leakage for both individuals and organisations (Rana et al., 2015; Fischer, 2017; Okereafor & Adelaiye, 2020;).

The most recent statistical reports from InfoWatch Analytics Center were released in 2018 (InfoWatch Analytics Center, 2018b; 2018c). These reports cover data leakage over both 2017 and 2018. Figure 3.3 shows the types of compromised data and their respective percentages as documented in the reports. From an IM perspective, personal data and payment details are often compromised. These two forms of data are regularly found on IM applications. IM users regularly transfer personal data without being aware of how this information can be misused. Users typically send personal data, including their live location, email address, residential address, and identification number, across an IM application. IM users also willingly send payment and other financial details across IM applications. Proof of payment, financial account numbers, credit card data, and bank pin numbers are some of the payment details transmitted. Users are unaware of the importance that these types of data have to a person with nefarious intent (InfoWatch Analytics Center, 2018b; 2018c).

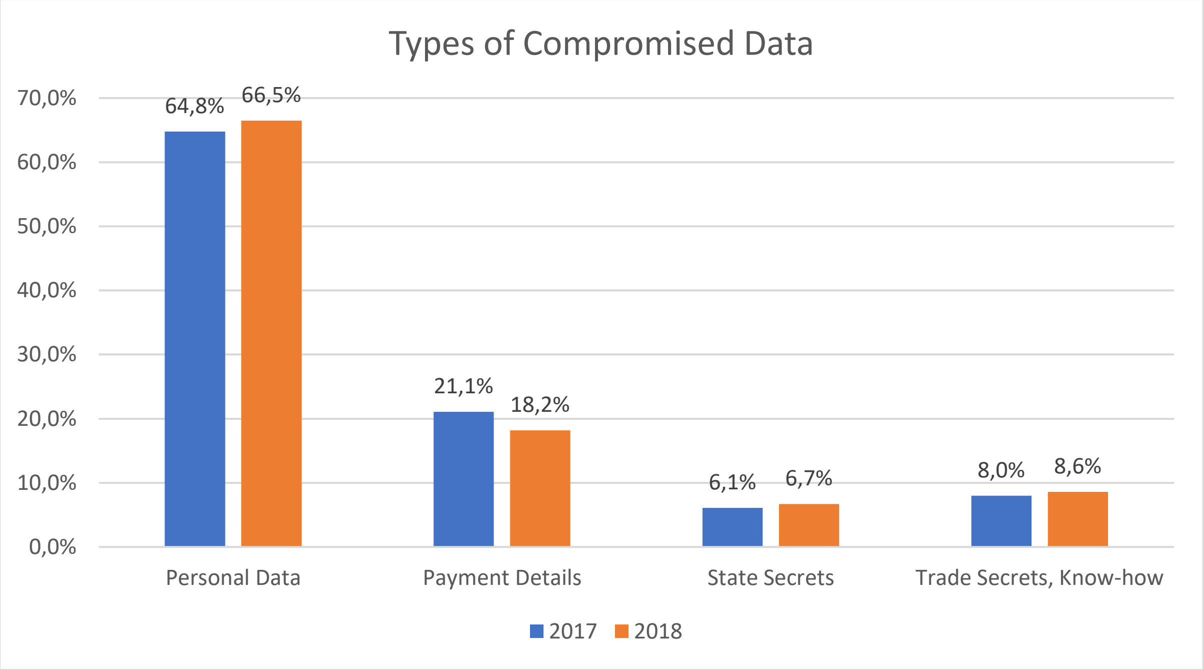


Figure 3.3 Types of Compromised Data for 2017 and 2018 (adapted from (InfoWatch Analytics Center, 2018b; 2018c))

By the end of 2018, instant messaging, as a medium for information leakage, was responsible for 4.5 per cent of data in comparison to the 2.4 per cent registered previously in 2017 over the same period (InfoWatch Analytics Center, 2018b; 2018c). These numbers of incidents for IM are likely to increase as IM is an emerging communication channel in the corporate environment.

With IM having been responsible for 4.5 per cent of data leaked in 2018 (InfoWatch Analytics Center, 2018a) and considering the growth in IM, the number of incidents is unlikely to reduce in the coming years. We can use the 4.5 per cent from 2018 as a baseline for 2019. Previously mentioned in Chapter 2, Section 2.3, there were 8.5 billion reported exposed records. Using the 4.5 per cent, we can estimate that 382.5 million of the 8.5 billion exposed records were exposed through IM applications. Using the 382.5 million in combination with the IBM reported a cost of $146 per record lost (the cost per record for 2020), previously stated in Chapter 2, Section 2.3.1.1, resulting in IM costing the global industries $55.84 billion in leaked records for the year of 2019.

The issue of information leaks has elevated beyond the domain of industry becoming a national or even international concern. As a result, the protection of user information and the responsibility of organisations that handle vast quantities of user information has become a global concern (InfoWatch Analytics Center, 2018a).

3.5.2 Distribution of Malicious Code (Malware)

Owing to its popularity, IM applications have become one of the most widely used malware attack channels. The broad user base and swiftness of communication is especially optimal for the dissemination of malware. IM malware can spread rapidly and stealthily owing to IM functionality and social engineering tricks, posing a significant security threat not only to personal IM users, but also to companies that allow instant messaging to be used in the workplace (Xie et al., 2012; del Rey et al., 2015). All malware developers have the primary objective of inserting and spreading their malware into as many devices as possible. The majority of known IM malware is transmitted via public IM networks. IM malware-induced security breaches not only result in personal device disruption and financial damages, but often also greatly degrade IM service usability (Xie et al., 2012; Samantray et al., 2018).

Two main IM malware spreading mechanisms are file transfer and embedded message Uniform Resource Locator (URL). The malware propagates itself after breaching an IM client by either producing a malicious file transfer or delivering an instant message containing a malicious URL to the users located in the contact list of the victim. If the file or URL is clicked on by these unvigilant users, malicious code is triggered to execute or download from the URL and be implemented, and then the spread of malware begins at an increasingly large pace. To lure victims or to evade network filters, malicious files are usually altered. If a victim clicks on the file, the malware is invoked and attempts to infect more victims via the contact list (Xie et al., 2012; Ramakrishnan & Tandon, 2018; Jagadish et al., 2019). Malicious URL messages are now much more common for IM malware dissemination than malicious file transfers. These messages are labelled as IM SPAM, which is also referred to as SPIM (Odukoya et al., 2018). IM malware sends a message featuring a malicious URL to contacts instead of transmitting a file. Either a malware binary is downloaded and executed once a victim clicks the link, or other malicious web scripts are run to exploit the web browser or vulnerabilities of other associated applications. Owing to the unusual method of propagation, network administrators and common IM users lack adequate methods to protect their networks and devices from falling prey to IM malware (Xie et al., 2012).

As previously stated in Section 3.4, IM applications are utilised to transmit confidential information and will therefore contain confidential information. This leads to IM applications being targeted for attacks, as malicious individuals want to attain this confidential data and utilise it for their nefarious purposes. Malware that can be utilised for this purpose include trojans, spyware, and greyware.

Trojans are utilised to target devices and extract confidential information. An IM trojan, commonly referred to as Trojan-IM, has been designed and utilised to obtain confidential information found in IM applications on the infected device. This information includes usernames, passwords, chat history, and call log. This information is highly useful and valuable to malicious individuals (Johansen, 2020;

Ljuban, 2021). As IM applications contain large amounts of valuable data, malicious individuals have turned their attention to these applications. Once spyware is installed on a device, the attacker who installed it receives reports of user activities. These reports include all activities by a user utilising their IM application, which provides the attacker with a copy of this confidential information (Ramakrishnan & Tandon, 2018; Prasad & Rohokale, 2020). Greyware is another threat to IM applications. IM applications contain information which is highly useful for profiling and advertising. If successful, greyware will utilise the information transmitted across IM applications to profile the user and supply targeted advertisements (Faisal et al., 2019; Zhao et al., 2020). In the possession of a malicious individual, this amount of user-specific data can be incredibly harmful.

3.5.3 Man-in-The-Middle Attack

As stated in Chapter 2, Section 2.3.1.3, Man-in-the-middle (MiTM) attacks refer to the monitoring of a network, device or system activities, to acquire confidential information. MiTM is an active network attack in which an attacker is remotely positioned to capture, transmit, and receive interactions between two or more parties. The attacker is able to imitate one or all parties involved to access information (Rotem & Segev, 2018; Taleqani et al., 2018; Alwazzeh et al., 2020; Prasad & Rohokale, 2020; Symeonidis & Lenzini, 2020).

A key problem in securing messaging platforms is that when setting up secure end-to-end networks, defending against MiTM attacks is difficult. These attacks are made possible by the inability of users to authenticate their incoming messages despite the existence of messaging platforms (Rotem & Segev, 2018). Users do not ensure that the intended receiver is receiving the messages they send. As illustrated in Figure 3.4, MiTM is an active attack in which a malicious actor is secretly positioned to intercept, send, and receive communications between two or more parties, or to imitate a party involved to gain the information desired (Rotem & Segev, 2018; Taleqani et al., 2018; Alwazzeh et al., 2020; Prasad & Rohokale, 2020).

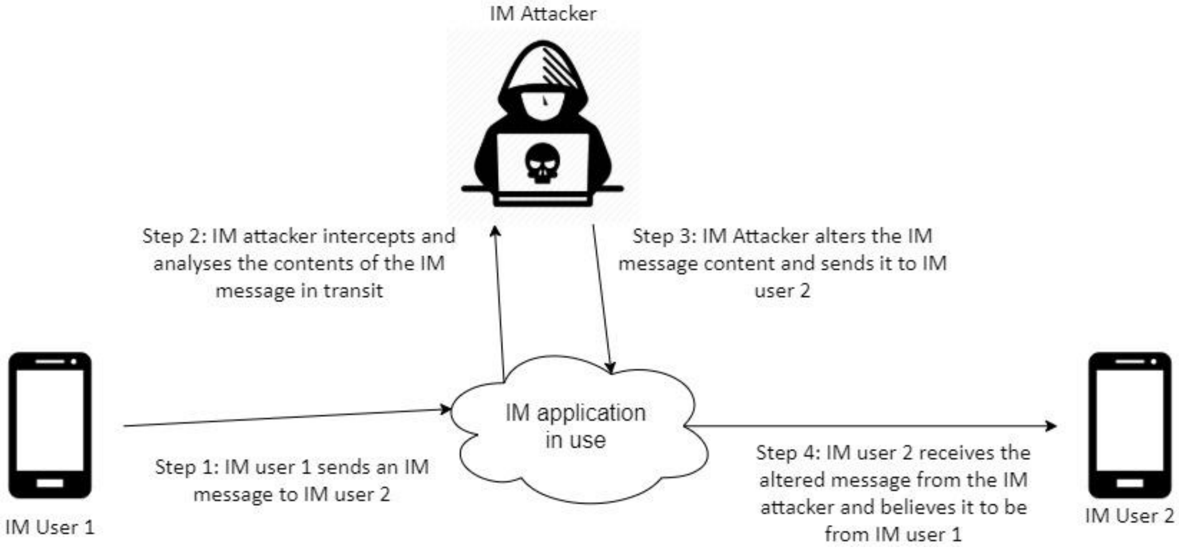

IM Attacker
Step 2: IM attacker intercepts and analyses the contents of the IM message in transit
Step 3: IM Attacker alters the IM message content and sends it to IM user 2
IM application in use
Step 1: IM user 1 sends an IM message to IM user 2
Step 4: IM user 2 receives the altered message from the IM attacker and believes it to be from IM user 1
IM User 1
IM User 2

Figure 3.4 Man-in-the-Middle Attack (adapted from (Alwazzeh et al., 2020))

Not all IM applications implement effective end-to-end encryption, and without encrypted communication being implemented, messages between devices and applications are susceptible to MiTM attacks. MiTM attacks are more likely to occur where no form of encryption is utilised (Prasad & Rohokale, 2020). When there is communication between two users, the attackers can find it harder to execute their attack successfully. When moving to the group setting, however, an MiTM attacker has many more ways to interrupt the interactions between the parties, making security a much more complex task (Rotem & Segev, 2018). This complexity allows attackers to find gaps in the implemented security, in order to execute their attacks. In 2017, security researchers explored the BlueBorne attack. This attack enabled a hacker to acquire mobile device control and to apply a MiTM attack to steal information. This vulnerability has been found in smartphone, desktop, and IoT operating systems such as Android, iOS, Windows, and Linux. This attack does not need internet for spreading. Without user knowledge, the hacker can connect to the target device quietly and take control of it to initiate the next attacks (Prasad & Rohokale, 2020). One problem with secure instant messaging is that there is no way to tell if a MiTM attack has occurred (Johansen et al., 2018).

3.5.4 Permission System Vulnerability

As stated in Chapter 2, Section 2.3.1.4, an application's permission request is a request to access user data or device resources. If granted, an application can manipulate user information and device resources to obtain its desired result (Gu et al., 2017; Liu et al., 2019).

With the growth in the usage of instant messaging applications, the permissions that messaging applications require during download and registering have also increased. IM application request access to resources, including contacts, images, video and audio, system resources, camera, microphone, and location. Just because a user's permission is requested by an IM application, it does not mean that they are going to be clear on what they want to do with a user's information. With IM, several permissions are either introduced in rapid succession at once, or one after another, and the vocabulary can be ambiguous or strangely worded. The precise context of the permission or why it is proposed is often not explained, even though the wording is plain. In addition, awareness of the ramifications of voluntarily granting permission is seldom provided (Baldikov, 2020)

By providing IM applications with certain permissions the following could potentially happen, including but not limited to: users could be spied on, eavesdropped, or monitored unknowingly; user images and multimedia could be snooped and/or stolen; and user confidential files and records could be manipulated. Certain instant messengers, or their parent corporations, have become some of the worst consumer data and information violators in recent times (Baldikov, 2020).

3.5.5 Social Engineering

As stated in Chapter 2, Section 2.3.1.6, social engineering is referred to as the art of manipulating human weaknesses to achieve a malicious objective (Breda et al., 2017). As the users themselves are the most insecure aspect of the system, social engineering is preferable to most other ways of hacking in that it can breach even the most secure systems. Research has demonstrated that, in many ways, social engineering is simple to automate and can thus be carried out on a broad scale (Krombholz et al., 2015; Aldawood & Skinner, 2019a).

Social engineers are beginning to turn their attention to IM applications as a tool for phishing and reverse engineering attacks. IM applications make it easy to perform identity theft and to manipulate trustworthy relationships (Krombholz et al., 2015). Social engineers have access to millions of potential victims through IM applications. The wide range of social engineering attacks enables social engineers to accomplish their objectives. With phishing cloning, a large volume of IM users can be attacked by social engineers. Social engineers can perform spear phishing, whaling or reverse social engineering for a more targeted attack, using IM to implement these attacks. The targeted individual is at risk of financial loss and loss of confidential data, including passwords, financial details, and contact information (Krombholz et al., 2015; Koyun & Al Janabi, 2017; Aldawood & Skinner, 2019a) .

3.5.6 Impact of Application Security Threats

In order to assess the potential damage each of the relevant application security threats can achieve, these threats need to be assessed in terms of their potential impact. To assess the potential of the threat successfully, assessment criteria are required. The assessment criteria, presented in Table 3.3, were developed, based on development processes and guidance established in South African National Standard 27003 and 27004, Risk Assessment in Practice, and IT Incident Criteria (Curtis & Carey, 2012; Roberts, 2016; South African National Standard, 2020a; 2020b). Table 3.3 contains these assessment criteria categorised according to five main ratings ranging from incidental (1) to extreme (5).

Table 3.3 Assessment Criteria (adapted from (Curtis & Carey, 2012))

Rating	Descriptor	Definition
5	Extreme	• Impacts all users and/or IM organisation. • Extreme financial loss, with the potential to bankrupt or cripple the organisation. • International long-term negative media coverage; game-changing loss of market share. Potentially unrecoverable reputational damage. • Significant prosecution and fines, litigation and potential incarceration of leadership. • Extreme damage to users and/ or IM organisation. • Potential for total system failure. Requires immediate response from IM organisation to resolve the complexity of the impact.
4	Major	• Impacts the majority of users and/ or IM organisation. • Major financial loss. • National long-term negative media coverage; significant loss of market share. Major reputational damage. • Major damage to users and/ or IM organisation. • Requires prompt intervention from IM organisation to mitigate the critical damages.
3	Moderate	• Impacts a large number of users and/ or IM organisation. • Moderate financial loss. • National short-term negative media coverage. Nationwide reputational damage. • Moderate damage to users and/ or IM organisation. • Users can intervene or requires intervention from IM organisation to mitigate damages.
2	Minor	• Impacts multiple users. • Minor financial loss. • Local reputational damage. • No or minor damage to users.
1	Incidental	• Impacts a single user. • No damage to the user. • Mainly an inconvenience.

Table 3.3 defines the assessment criteria with Extreme being the most dangerous and Incidental being the least. These five criteria will be utilised in Table 3.4, to assess the potential impact of the application security threats discussed in Chapter 3, Section 3.5.

Table 3.4 Impact of Application Security Threats

Threat	Rating	Description
Confidential information leakage	4	Confidential information leakage is synonymous with direct and indirect losses. Both forms of losses could lead to large financial and reputational damage. Confidential information leakage is also known to target individuals, large groups and organisations and has the potential to cause major damage to the intended target. The IM organisation is required to respond promptly to this threat, to mitigate the damages. Thus, confidential information leakage is regarded as a major threat.
Distribution of malicious code	4	Distribution of malicious code targets individuals, large groups and organisations and has the potential to cause major damage to the intended target, which is assisted by the rapid rate of propagation by the malware utilised. Distribution of malicious code has been linked to large financial and reputational damage. The IM organisation is required to respond promptly to this threat, to mitigate the damages. Thus, distribution of malicious code is regarded as a major threat.
Man-in-the-middle attack	3	MiTM attacks can impact a large number of users and/ or an organisation. MiTM has been known to cause moderate to major damage to users; the level of damage is dependent on the target. The IM organisation can be required to mitigate damages; however, users can intervene. The potential for financial and reputation damage is large. Thus, MiTM is regarded as a moderate threat.
Permission system vulnerability	4	Permission system vulnerability requires prompt intervention from IM organisation to mitigate the critical damages. Permission system vulnerability has the potential for major damage to users, reputation and financial loss. Although permission system vulnerability affects all users, it is regarded as a major threat, and not extreme.
Social engineering	4	The majority of users and members of the IM organisation have the potential to fall victim to social engineering. Social engineering has been linked with major financial, reputational and user damage, depending on the intended target. Thus, social engineering is regarded as a major threat.

As seen in Table 3.4, four application security threats have been rated as major threats, namely confidential information leakage, distribution of malicious code, permission system vulnerability, and social engineering. Only one application security threat is rated as moderate threat, namely MiTM attack. No application security threats were rated as extreme, minor, or incidental. This demonstrates that MiTM attack is the least threatening, while confidential information leakage, distribution of malicious code, permission system vulnerability, and social engineering are the most threatening of those discussed.

As presented in Table 3.2, confidential information leakage, distribution of malicious code, Man-in-the-middle attack, permission system vulnerability, and social engineering are application threats which are relevant to IM applications. In addition, these five threats have a moderate to major impact on IM application security, as displayed in Table 3.4. Therefore, these five application threats can be deemed instant messaging security threats.

3.6 Key Elements Relating to Instant Messaging Security Risk

Figure 3.5, through the lens of the instant messaging perspective, depicts the challenges and complications in the world of instant messaging. Four of the seven key elements, found in Figure 3.5, are discussed, namely IM hackers and/or attackers (previously named threat agents), IM threats (previously named threats), IM vulnerabilities (previously named vulnerabilities) and IM risk (previously named risk). The remaining three elements, namely: users and/or instant messaging organisation (previously named owners), IM controls (previously named controls) and IM assets (previously named assets), will be discussed in Chapter 4.

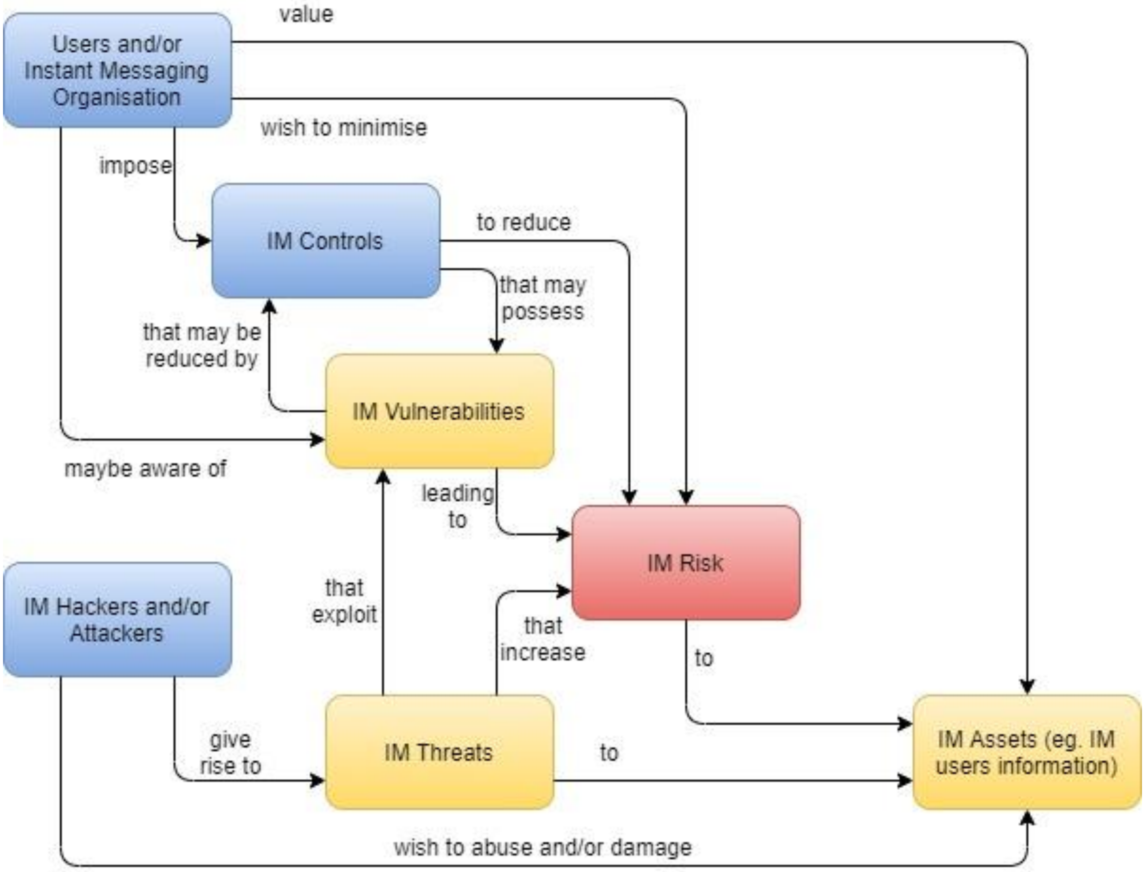


Figure 3.5 Key Elements Relating to Instant Messaging Security Risk (adapted from (Common Criteria) cited in Task Force Report (National Cyber Security Partnership, 2004))

From Figure 3.5 it is clear that Threat Agents may be perceived as hackers or attackers from an IM perspective. IM hackers give rise to threats to the assets of the IM user and organisation. IM hackers will exploit the IM vulnerabilities located in the IM platform, which will increase the IM risks to these IM assets. For their nefarious intentions, IM hackers intend to exploit and manipulate the IM asset.

The instant messaging security threats discussed in Chapter 3, Section 3.5, could potentially be utilised by an IM hacker. IM hackers will increase the IM threats which, as a result, will increase the IM risk to an IM asset. The threats discussed in Section 3.5, are the IM threats labelled in Figure 3.5, which will exploit IM vulnerabilities to achieve the IM hacker's objectives. IM vulnerabilities were mentioned in both Chapter 2, Section 2.3.1, and Chapter 3, Section 3.5. They include the human element, permission-hungry applications, software vulnerabilities, and poor application design. Once IM threats exploit these IM vulnerabilities, this contributes to an increase in IM risk to the IM information assets.

Reducing IM risk, by addressing IM vulnerabilities and the associated potential IM threats is the only way to mitigate the IM risk to IM information assets.

3.7 Conclusion

A popular form of daily life is instant messaging applications. Everywhere, from the business world to personal lives, these applications are used. A key contributor to IM application being targeted, is the high use and general success of these applications. Developers of IM applications have introduced numerous features to the user experience of these applications.

To increase the user experience of their applications, IM developers have developed their own code libraries or implemented external libraries. Owing to time constraints, developers do not test these libraries thoroughly. The lack of security testing creates vulnerabilities in the IM application which, once detected, can be exploited by hackers.

Users themselves have been viewed and labelled as the weakest link in security. The human nature of trusting others leads to vulnerabilities for IM applications. Users can be tricked and manipulated unknowingly into performing malicious activities.

IM developers need to protect their users and need to alleviate the security and privacy burden on users. IM developers need to develop, with security and privacy as a priority. General IM users do not have the knowledge or skill to identify a dangerous situation and to handle it appropriately.

This chapter discussed instant messaging applications, what it is, how it works, common IM applications and compared their various features. Also included in this chapter were the application security threats identified in Chapter 2, Section 2.3.1. These application security threats were adapted to IM and were examined from the IM perspective. The application security threats deemed relevant to IM application are now referred to as instant messaging security threats, which assists in the partial completion of SRO1. Chapter 4 will concentrate on examining the currently available and implemented IM security and privacy controls, and the role controls play in protecting instant messaging users, both individuals and corporations. The security and privacy controls, currently available, will be examined, to determine whether they protect IM users effectively.

Chapter 4 – Instant Messaging Security

4.1 Introduction

In Chapter 3, instant messaging was introduced. Chapter 3 defined instant messaging, elaborated on how instant messaging works and provided a comparison of the popular instant messaging applications and their functionality. The application threats, from Chapter 2, Section 2.3, were adapted to instant messaging applications and became instant messaging security threats. The threats they pose to Instant Messaging applications were discussed.

The aim of this chapter is to discuss instant messaging security and privacy controls and the role that these controls play in protecting instant messaging users, both individuals and corporations.

The findings of this chapter, in combination with the findings of Chapters 2 and 3, address SRO1 of this study. The requirements of SRO1 are *to determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users*.

The structure of this chapter is as follows. Section 4.2 discusses how these controls are used to alleviate threats and secure user information in the instant messaging environment, while Section 4.3 highlights the security and privacy controls implemented in instant messaging applications. Section 4.4 concludes the chapter.

4.2 Key Elements Relating to Instant Messaging Security Risk

Looking at Figure 4.1, which is a repeat of Figure 3.5, through the lens of the instant messaging (IM) perspective, the challenges, and complications in the world of instant messaging are evident. Three of the seven elements, found in Figure 4.1, are discussed, namely owners (now named users and/or instant messaging organisation), controls (now named IM controls), and assets (now named IM assets). The remaining four elements were discussed in Chapter 3, Section 3.6.

In Figure 4.1 it is shown that owners own assets that have value and want to reduce the vulnerabilities and risk to their assets by implementing controls. The owners can be viewed as both the IM organisations (Facebook Messenger, Snapchat, Telegram, Viber, WeChat, and WhatsApp) or the actual user. The user is the individual who provides the IM asset to the IM platform. Without the user there is no IM asset. IM assets can be viewed as a collection of information about each specific user on the IM platform. Once the IM organisation acquires these IM assets, they also become owners of the IM asset. However, the true original owner is the user.

Users do not have the ability that an IM organisation has to implement IM controls. Through the implementation of IM controls, IM organisations can reduce IM vulnerabilities and IM risks to IM assets. When implementing an IM control, there is the potential for the new IM control to contain unidentified vulnerabilities that can lead to IM risks. IM organisations wish to secure their IM assets, to protect the value of the IM assets to the organisation. Chapter 4, Section 4.3, highlights current existing IM controls that IM organisations have implemented to secure their IM assets. These controls include end-to-end encryption, encryption in transit, two-factor authentication, and password locks.

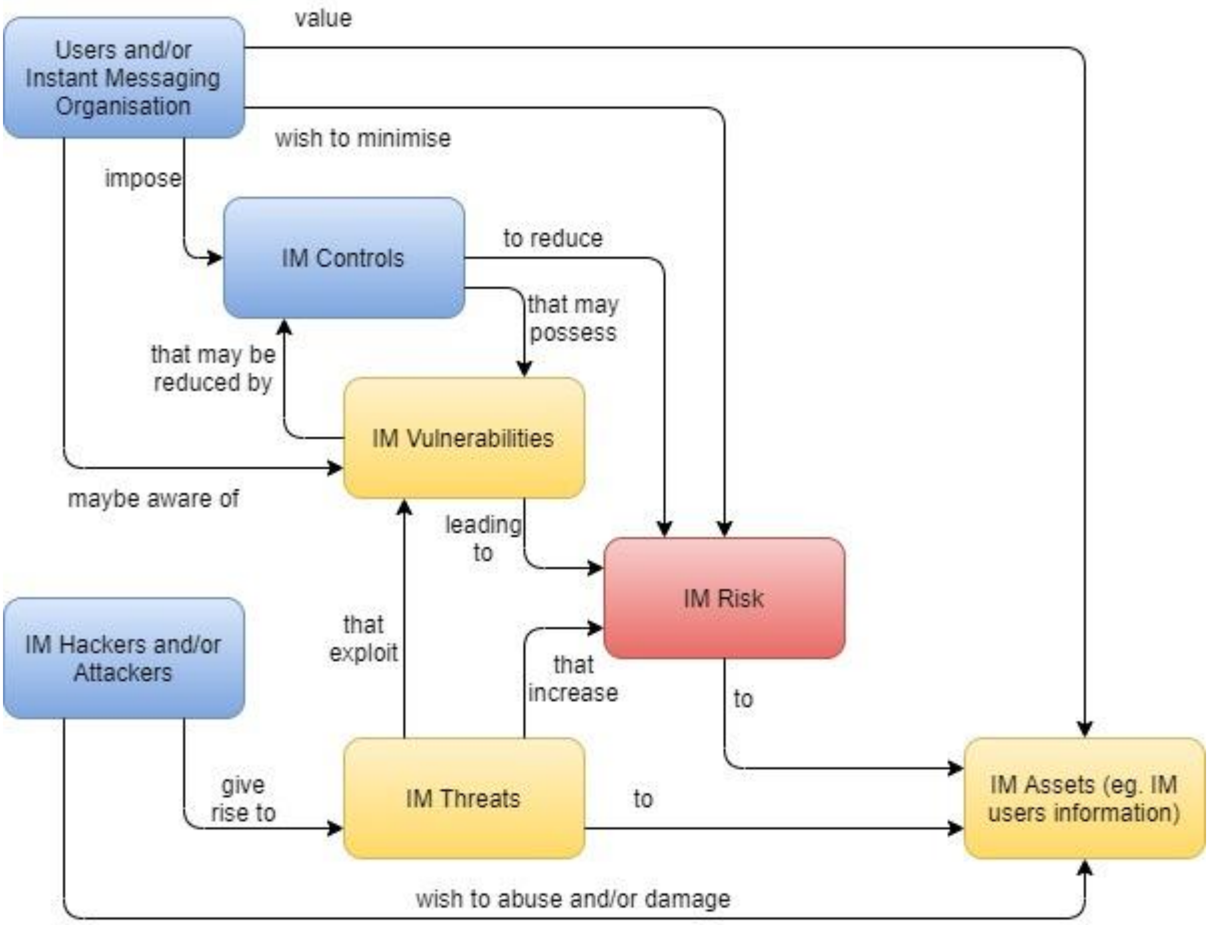


Figure 4.1 Key Elements Relating to Instant Messaging Security Risk (adapted from (Common Criteria) cited in Task Force Report (National Cyber Security Partnership, 2004))

As stated, IM assets are the collections of information on a particular user. These sets of information are highly valuable to users, IM organisations, and IM hackers. IM organisations can gain profit by selling these IM assets to other organisations for advertising purposes. IM hackers can utilise this information for nefarious purposes and gain profit from these activities. Users value this information as they do not want their data exploited for malicious reasons.

The implementation of IM controls is critical to reduce both IM vulnerabilities and IM risk. The reduction of IM vulnerabilities has a direct correlation with the reduction of IM threats. The added protection from IM controls will lead to user satisfaction and IM organisations maintaining the value of their respective IM assets.

4.3 Instant Messaging Security and Privacy Controls

In Chapter 2, Section 2.3, threats to information security were discussed. These threats and their relation to IM applications were discussed in Chapter 3, Section 3.5. The following subsections discuss several IM security and privacy controls that intend to alleviate various IM security threats. These IM controls, namely: End-to-end Encryption, Encryption in Transit, Deleting Messages, Self-destruct Messages, Two-factor Authentication, Verification SMS/Email, Password Lock, Screenshot Detection, Remote Log Out and Account Self-destruct, are discussed as they appear in various instant messaging applications including Facebook Messenger, Snapchat, Telegram, Viber, WeChat, and WhatsApp.

4.3.1 End-to-end Encryption and Encryption in Transit

Software development organisations have tried to address the problem of IM vulnerabilities and IM threats, as users demand improved protection and privacy in instant messaging applications. One of the controls has been the introduction of end-to-end encryption within these applications. End-to-end encryption relates to where messages are encrypted during communication, and where no version is left unencrypted on the repositories of the service providers. These communications cannot be interpreted by someone other than the people communicating; no third party, not even the government nor the developers of those apps. Instead of plain text, information is conveyed using a special code. End-to-end encryption includes encryption in transit (Shirvanian et al., 2017; Zaharia & Cihodariu, 2019).

Encryption in transit is another form of encryption that is utilised by instant messaging apps. This indicates that the message is encrypted, between the user and the service provider, but stored in the repositories in a plain text format. This represents a risk because the repositories may be read, duplicated or abused by the service provider or by other third parties (Botha et al., 2019).

The following is a discussion of how End-to-end Encryption and Encryption in Transit are implemented in the selected IM applications:

- **Facebook Messenger –** As a default privacy and security control, Facebook Messenger does not implement end-to-end encryption. To use end-to-end encryption, a secret conversation must be enabled (Corrigan, 2020). Facebook Messenger does not notify users that regular communications use a weaker encryption method in the form of encryption in transit (Amnesty International, 2016; Botha et al., 2019). Currently, secret conversations are only accessible through the mobile application; thus, they will not appear on Facebook chat, the chat located on the Facebook website, or messenger.com (Woollaston, 2016). With regard to privacy, Messenger received a rating of 73 out of 100 (Amnesty International, 2016).

- **Snapchat –** Snapchat introduced end-to-end encryption in 2018 (Titcomb, 2019). However, the encryption only extends to images and videos (Caffo, 2018) leaving text messages in a vulnerable state. With regard to privacy, Snapchat received a rating of 26 out of 100 (Amnesty International, 2016).

- **Telegram –** Telegram has a security control that allows users to use end-to-end encryption to protect their communications, called secret chat. However, it is not a default security control, so users will need to enable it (Corrigan, 2020). By default, regular chats use a weaker encryption form, namely encryption in transit (Botha et al., 2019). With regard to privacy, Telegram received a rating of 67 out of 100 (Amnesty International, 2016).

- **Viber –** By default, Viber provides all correspondence with end-to-end encryption. However, it does not post a transparency report or reveal full details of how encryption is applied (Amnesty International, 2016; Caffo, 2018).

- **WeChat –** WeChat has major issues, according to the 2016/17 Amnesty International study on privacy in instant messaging applications. With a privacy score of 0 out of 100, WeChat ranked last (Amnesty International, 2016). WeChat does not provide end-to-end encryption and has not released transparency reports on Chinese government demands for information.

Based on that, WeChat was exposed to both censorship and surveillance. Users should believe completely that everything they communicate on WeChat is not private (Grigg, 2018). It is better for users to uninstall the app from their device owing to WeChat's lack of privacy and security (Botha et al., 2019).

- **WhatsApp –** As a default privacy and security control, WhatsApp incorporates end-to-end encryption (Caffo, 2018; Zaharia & Cihodariu, 2019). With regard to privacy, WhatsApp received a rating of 73 out of 100 (Amnesty International, 2016).

4.3.2 Deleting Messages

The functionality of deleting individual messages from chat history, both individual and group chats, was implemented by IM application developers, to increase the privacy and controls that IM users have over their messages. The reasons that IM users delete messages include spelling correction, withdrawing mistakenly sent messages, withdrawing inappropriate messages, and erasing confidential information (Schnitzler et al., 2020).

The following is a discussion of how Deleting Messages is implemented in the selected IM applications:

- **Facebook Messenger –** Facebook Messenger allows users to erase a message sent to others in the conversation permanently, or only cover from the sender’s view. If users choose *Remove for You*, other persons in the conversation can still see the messages on their chat screen. If users select *Remove*, then *Unsend*, the deleted message will not be available to people participating in the conversation. Bear in mind that people who received the message may have seen the message already and are still able to report the conversation. Users can delete a message for anyone in a chat only up to 10 minutes after a message has been received (Facebook, 2020c).

- **Snapchat –** Snapchat enables users to delete a message sent in a chat. Once the message has been deleted, a notification will remain in the chat to notify users that a message was deleted (Constine, 2018; Snapchat, 2020b).

- **Telegram –** While there was a 48-hour time limit, Telegram allowed users to retract a message received in a chat back in 2017. Since then, Telegram has eliminated the time limit and allows users at any time to retract any message (Telegram, 2017, 2019a).

- **Viber –** Viber enables users to delete a message from their own phone and it will be automatically erased from all devices to which they have sent it. This works in both 1-on-1 and group chats. Users are given the option *Delete for Myself* to remove it only from their device; or *Delete for Everyone* to remove it from their own device and from the devices of all recipients (Fosker, 2015).

- **WeChat –** Users of WeChat can recall a message sent within a two-minute period. The person or individuals in the chat can only see that a message has been recalled by the sender, but not the message itself. Users can recall all forms of messages, including speech, text, pictures, video, contact cards, and location messages (WeChat, 2020a; 2020d).

- **WhatsApp –** WhatsApp allows users to retract their sent messages, within one hour of being sent. Deleting messages for everyone makes it easier for a user to delete specific messages sent to an individual chat or group. This is especially helpful if a user has sent a message to the wrong chat, or if there is an error in the message sent. To inform chat participants that a message has been removed, messages that have been effectively deleted will be replaced by a message saying, *This message has been deleted* (WhatsApp Inc., 2020b).

4.3.3 Self-destruct Messages

Self-destructing messages were introduced to improve the personal security of IM users. Some IM users do not want messages saved to chat history, while chatting; this is why self-destructing messages were implemented (Aggarwal et al., 2018).

The following is a discussion of how Self-destruct Messages is implemented in the selected IM applications:

- **Facebook Messenger –** Facebook Messenger enables users to create secret chats. Users can add a timer or expiration to the message in a secret chat, so that it can self-destruct within the period the user specified after their recipient sees it. (Nield, 2018; Otachi, 2019; Reshmi & Raja, 2019).

- **Snapchat –** Snapchat enables users to share images, videos or text messages that are time limited. The amount of time a recipient is permitted to view the content can be chosen by the sender. The content disappears after this time and is no longer available to the recipient (Roesner et al., 2014; Bayer et al., 2016; Piwek & Joinson, 2016).

- **Telegram –** Telegram enables users to create secret chats. Users get the option to set a self-destruction timer when creating a secret chat. This will trigger messages sent in the secret chat to vanish from the devices of both the sender and the receiver after the set time period has elapsed (Sutikno et al., 2016; Abu-Salma et al., 2017; Faramarzi et al., 2019).

- **Viber –** For messages in a secret chat, Viber allows a user to set a self-destruct timer so that it is automatically removed from the Viber chat after their message is read on both sides of the conversation. Screenshot notifications will be active in the chat when this function is on (Viber, 2020).

- **WeChat –** WeChat does not provide users with a control that self-destructs messages (Botha et al., 2019).

- **WhatsApp –** A control named *Disappearing messages* was introduced by WhatsApp. Once they are seven days old, the messages will disappear, but no media messages sent (pictures, video, or audio) will be erased from the device. This function can be enabled in a one-to-one chat by any chat participant, while the admin would have the choice in a group chat (Facebook, 2020d; WhatsApp Inc., 2020i).

4.3.4 Two-factor Authentication

Two-factor authentication is a security control that safeguards an IM user's account. Using a password, which is set by the IM user, and a registered token, which is a piece of software located on the IM users' device, the goal is to secure the IM user's device even if the password is compromised.

The following is a discussion of how Two-factor Authentication is implemented in the selected IM applications:

- **Facebook Messenger –** An optional control that adds more protection to a user's Facebook account is two-step verification. Users are required to set a pin when they allow two-step verification. An extra login stage is introduced by two-step verification. Users are required to enter their email address. This enables Facebook to give users a reset link in case they ever forget their pin and helps secure their account as well. When logging into their Facebook account from an unrecognised device, a user can see the two-step verification screen (Facebook, 2020e).

- **Snapchat –** Two-factor authentication is an optional security control to validate that when a user logs into their Snapchat account, it is really them. In addition to their Snapchat username and password, two-factor authentication introduces a second login stage. This makes it more secure for their account (Snapchat, 2017d).
- **Telegram –** Telegram enables two-step verification to be activated. In addition to the code that users receive in the SMS, two-step verification enables users to create a password that would be used any time they log in from a new device into their account. Users need to be careful, though, since they will not be able to access their messages from other devices if they forget this password. If they plan to turn on two-step verification, it is suggested that they set up a recovery email or at least a password hint (Telegram, 2015a).
- **Viber –** Viber does not provide users with a two-factor authentication control (Botha et al., 2019).
- **WeChat –** WeChat enables users to connect their phone or email address to their account, which can be used for two-factor authentication or account recovery. As users log into their account, two-factor authentication provides an extra layer of security (BtCIRT, 2017; TOKOK, 2018).
- **WhatsApp –** An optional control that adds more protection to a user's WhatsApp account is two-step verification. Users are required to set a pin when they allow two-step verification. An extra login stage is introduced by two-step verification. Users have the option of entering their address by email. This enables WhatsApp to give the user a reset link in case they ever forget their pin and helps to secure their account as well. After successfully registering their phone number when logging into WhatsApp, they can see the two-step verification screen (WhatsApp Inc., 2020a).

4.3.5 Verification SMS/Email

Verification SMS/email, is utilised to ensure that the phone number or email address provided belongs to the individual attempting to register on an IM application (Ali & Alsaad, 2020).

The following is a discussion of how Verification SMS/Email is implemented in the selected IM applications:

- **Facebook Messenger –** When registering a user for their application or verifying a user when signing in, Facebook Messenger uses SMS verification in the form of a One Time Pin (OTP). Users are expected to register their mobile number, and an OTP will be sent by the application server to the mobile number. The application requires this OTP to verify the user (Chaudhari, 2015). When resetting passwords, Messenger also utilises OTPs. (Gelernter et al., 2017).
- **Snapchat –** Snapchat uses SMS verification when registering a user for their application or verifying a user upon signing in. Users are expected to enter their mobile number, and Snapchat will send the code to the mobile number. To verify their number, users must input

this code into the Snapchat app (Snapchat, 2017f). For extra protection, Snapchat conducts a similar process to verify a user's email address (Snapchat, 2017e).

- **Telegram –** SMS verification is used by Telegram to validate users. An SMS message containing a code is sent to a user-registered number. The code then needs to be inserted into the Telegram application by the user. This process links a Telegram user account to the represented phone number (Telegram, 2019b; T9gram.com, 2020).

- **Viber –** SMS verification is used for the validation of users by Viber. An SMS message will be sent to a user-registered number containing a code. This code must then be loaded into the Viber app by the user. This procedure connects the Viber account to the phone number represented. This is required in order to complete and enable the account registration process (Viber, 2019b).

- **WeChat –** When registering a user for their application or verifying a user upon signing in, WeChat implements SMS verification. For both business and standard account registration, WeChat requires this. It is expected that users will enter their mobile number, and WeChat will provide the mobile number with a code. Users must enter this code into the WeChat app to confirm their number. In addition, WeChat performs a similar procedure to check the email address of a user, which is needed for business accounts (Tanner, 2018; Silas, 2020).

- **WhatsApp –** WhatsApp utilises SMS verification to discourage malicious users from impersonating someone else by using the number of the victim. An SMS containing a four-digit code is sent to the number registered by the user. The code must then be copied by the user into the WhatsApp application. This process connects a WhatsApp user account to the phone number represented (Jhala & Patel, 2015; WhatsApp Inc., 2020h).

4.3.6 Password Lock

Password locks have been implemented to add an additional layer of security and privacy control to an IM application. To access the IM application, an individual must satisfy the requirements of the password lock. If the requirements are not met, access is not granted and the information located on the IM application remains secure (Weichbroth & Łysik, 2020).

The following is a discussion of how Password Lock is implemented in the selected IM applications:

- **Facebook Messenger –** A function called App Lock was added to Messenger. To unlock the Facebook Messenger app, App Lock uses the privacy settings of a user’s device, such as fingerprint or face authentication. Facebook ensures that their touch or face ID is not transmitted nor stored (Sullivan, 2020).

- **Snapchat –** For their application, Snapchat does not have a built-in password lock. This is further endorsed by third-party developers who build their own apps to provide Snapchat users with this functionality (Sharma, 2020).

- **Telegram –** Telegram provides users with a security control to prevent unauthorised access to the application. In order to open an application, a passcode is required. To block access to your communications or Telegram call logs by intruders, users can set a four-digit password, which could be beneficial if their device has been misplaced or stolen (Corrigan, 2020; Pabreja et al., 2020).

- **Viber –** Viber has developed a password lock variation of its own. Viber allows users to hide and access chats from their chat list with a PIN whenever needed by the user (Viber, 2020).
- **WeChat –** WeChat has a password control that, when implemented, uses the privacy settings of a user's device to access the WeChat app, such as fingerprint authentication. Voiceprint, a voice recognition system that can be used to gain access to the app, was also introduced by WeChat (WeChat, 2015).
- **WhatsApp –** WhatsApp provides users with a security control to shut down unauthorised access to the application. In order to unlock an application, a passcode is required, which prevents unauthorised access to conversations, messages or WhatsApp call logs (Pabreja et al., 2020).

4.3.7 Screenshot Detection

Screenshot detection has been implemented to assist IM users with controlling the flow of their confidential information. IM users will be notified if a screenshot occurs, thus informing the IM user that their information has been permanently captured by the recipient (Ashktorab, 2018).

The following is a discussion of how Screenshot Detection is implemented in the selected IM applications:

- **Facebook Messenger –** Unfortunately, when screenshots are taken, Facebook Messenger does not alert users. Without a user knowing, the recipient can take a screenshot of an image or messages (Techjunkie, 2020; Tripathi, 2020).
- **Snapchat –** When a user takes a screenshot of the content of others on Snapchat, Snapchat notifies the users. If users take a screenshot of a text message, picture or video, the app sends an alert (Ashktorab, 2018; John, 2019; Tripathi, 2020).
- **Telegram –** Telegram, in secret chats, prevents a user from screenshotting. A screenshot can be taken in standard chats, but in secret chats, there is no way to do it. If a user is able to take a screenshot in a secret chat, however, a notification will be sent to the chat members, alerting them that a screenshot has taken place (Meyers, 2019; Telegram, 2019b).
- **Viber –** There is no prevention or alerts about screenshots while in a regular chat. Viber does not allow the recipient to take a screenshot of the conversation in a secret chat, on Android. If anyone has attempted to forward or screenshot the chat, you will receive a warning on iOS (Viber, 2018).
- **WeChat –** When users screenshot conversations, WeChat has no notifications or warnings (Botha et al., 2019).
- **WhatsApp –** After screenshots have been taken, WhatsApp does not alert users (Botha et al., 2019; Gogoi, 2019; Griffin, 2019).

4.3.8 Remote Log Out

Several IM applications allow IM users to log in from multiple devices. Through this option, an IM user can logout of all devices from the device currently being used (Botha et al., 2019; Corrigan, 2020).

The following is a discussion of how Remote Log Out is implemented in the selected IM applications:

- **Facebook Messenger –** Facebook Messenger does not provide the option for users to log out of the app (Facebook, 2020a).

- **Snapchat –** Snapchat refers to logging out as unlinking from an account. Snapchat requires users, in order to avoid unwanted access to their account, to unlink an account. To guarantee that the account is disconnected by a user, Snapchat uses two-factor authentication (Snapchat, 2017a).

- **Telegram –** Since Telegram enables users to log in simultaneously from several devices (web, PC, tablet, or smartphone), Telegram allows users remotely to log out of other devices from the same device in use. This helps to protect the device if it is compromised or lost. Telegram also offers a function to self-destruct an account after a specified period of time, to guarantee user privacy (Corrigan, 2020).

- **Viber –** Viber does not provide users with the option to log out of the application (Botha et al., 2019).

- **WeChat –** WeChat allows users to link their account to multiple devices, this can be achieved through Manage Devices in the account security settings. Users are also able to log their account out of devices using this control. This enables them to log out of an account remotely from an additional device (WeChat, 2020b; 2020c; 2020e).

- **WhatsApp –** WhatsApp allows users to use a desktop app or web browser to access their account. To ensure security control, WhatsApp enables users to log out of the mobile application from these access points. However, users are not able to log out of the mobile application (WhatsApp Inc., 2020c).

4.3.9 Account Self-destruct

Account self-destruct was created to ensure that an IM user’s confidential information will be erased automatically, if the IM user is inactive for a set period of time (Botha et al., 2019).

The following is a discussion of how Account Self-destruct is implemented in the selected IM applications:

- **Facebook Messenger –** Facebook Messenger does not provide a function for self-destructing a user’s account (Botha et al., 2019).

- **Snapchat –** Snapchat does not provide a function for self-destructing a user’s account (Botha et al., 2019).

- **Telegram –** A user’s account will be branded inactive and removed along with all messages, media, contacts and any other piece of information they store in the Telegram cloud if they cease using Telegram and do not come online for an allocated period of time, default of six months, which can be changed in their account settings (Telegram, 2019b; Corrigan, 2020).

- **Viber –** Viber does not provide a function for self-destructing a user’s account (Botha et al., 2019).

- **WeChat –** WeChat does not provide a function for self-destructing a user’s account (Botha et al., 2019).

- **WhatsApp –** WhatsApp does not provide a function for self-destructing a user's account (Botha et al., 2019).

4.4 Discussion of Instant Messaging Security

Table 4.1 summarises and provides an overall comparison of the IM security and privacy controls discussed in Chapter 4, Section 4.2.

As presented in Table 4.1, the tick symbol (✓) means the IM application does have the security and privacy control while the tick symbol with the word *optional* in brackets means that the security and privacy control is available in the IM application, but is not enforced by default, which makes the usage of the security and privacy control optional. For example, the tick symbol followed by *pictures and video only* in brackets, located in Table 4.1 under Snapchat, means that only picture and video files receive end-to-end encryption. The tick symbol followed by *only in secret chats* in brackets, located in Table 4.1 under Telegram and Viber, means that screenshot detection is only available to users when the secret chats feature is utilised. Similarly, the tick symbol followed by *only on desktop app or in browser* in brackets, located in Table 4.1 under WhatsApp, means that remote log out is only available to users when the desktop application is being utilised or when WhatsApp is being operated through a web browser.

Table 4.1 Summary of Instant Messaging Security and Privacy Controls

Security and Privacy Controls	**Facebook Messenger**	**Snapchat**	**Telegram**	**Viber**	**WeChat**	**WhatsApp**
End-to-end encryption	✓ (optional)	✓ (pictures and video only)	✓ (optional)	✓		✓
Encryption in transit	✓	✓	✓	✓		✓
Deleting messages	✓	✓	✓	✓	✓	✓
Self-destructing messages	✓	✓	✓	✓		✓
Two-factor authentication	✓	✓	✓		✓	✓
Verification SMS/Email	✓	✓	✓	✓	✓	✓
Password lock	✓		✓	✓	✓	✓
Screenshot detection		✓	✓ (only in secret chats)	✓ (only in secret chats)		
Remote log out		✓	✓		✓	✓ (only on desktop app or in browser)
Account self-destruct			✓			

When implemented and utilised, these IM security and privacy controls assist in alleviating IM vulnerabilities in the IM application which results in an alleviation of IM threats. As depicted in Figure 4.1, IM threats exploit IM vulnerabilities, which lead to IM risk. The various IM security and privacy controls cannot address IM vulnerabilities if they are not implemented by the IM developer or not understood and utilised by the IM user. As mentioned in Chapter 1, Section 1.3, the average IM user does not have the adequate knowledge or skill to ensure their own security and privacy when utilising IM applications.

Table 4.2 presents the security and privacy controls, identified in Chapter 4, Section 4.3, mapped against the IM security threats, identified in Chapter 3, Section 3.5.

Table 4.2 Instant Messaging Security and Privacy Controls Mapped Against Instant Messaging Security Threats

Security and Privacy Controls	Confidential information leakage	Distribution of malicious code	Man-in-the-middle attack	Permission system vulnerability	Social engineering	Total
End-to-end encryption	✓		✓			2
Encryption in transit	✓		✓			2
Deleting messages	✓	✓	✓			3
Self-destructing messages	✓	✓	✓			3
Two-factor authentication	✓		✓			2
Verification SMS/Email	✓					1
Password lock	✓					1
Screenshot detection	✓					1
Remote log out	✓	✓	✓			3
Account self-destruct	✓	✓	✓			3
Total	**10**	**4**	**7**	**0**	**0**	**21**

As shown in Table 4.2, all the identified security and privacy controls could assist in alleviating confidential information leakage. However, none of the security and privacy controls assists in alleviating the threats of permission system vulnerability nor social engineering. Permission system vulnerability is not mapped to any of the identified security and privacy controls, as these controls do not have an impact on the securing or validating of the requested permissions. Similarly, the identified security and privacy controls also do not have a direct or large enough impact on the user's operation or the general operation of the IM application, which results in social engineering not being mapped to any of the identified security and privacy controls. Potentially, the best security and privacy controls to alleviate the threats of permission system vulnerability and social engineering could be user education and awareness. Man-in-the-middle attack is mapped to seven of the ten security and privacy controls, whilst distribution of malicious code is mapped to four of the ten.

From examining the information presented in Tables 4.1 and 4.2, it can be noted that from the IM applications examined, Telegram is deemed to be the most secure since it implements all the identified security and privacy controls. WeChat, on the other hand, is the least secure IM application from those examined, as WeChat only implements five of the identified security and privacy controls.

4.5 Conclusion

The security and privacy controls introduced by IM application developers ensure that the IM application is protected to a certain degree. However, IM applications are not 100 per cent secure, and when comparing the application security of a select group of IM applications (Facebook Messenger, Snapchat, Telegram, Viber, WeChat, and WhatsApp) some IM applications are more secure than others. IM developers are causing IM users to be exposed and vulnerable to attack by leaving security and privacy decisions to IM users. Many IM users do not have the experience or resources to make these security decisions, and IM application developers should therefore keep this in mind when designing the security of their IM applications.

The findings of this chapter, in combination with the findings of Chapters 2 and 3, address SRO1 of this study, *to determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users*.

To assist IM developers in the design, development and implementation of IM controls and security features, heuristics, guidelines, standards and best practices were developed (Gonçalves et al., 2016). Heuristics, guidelines, standards and best practices are built by researchers and practitioners, as a result of research and experience, to guide software development (Neumann et al., 2018).

This chapter discussed security and privacy controls of IM applications and the role that these controls play in protecting IM users, both individuals and corporations. In Chapter 5, the topic of existing security and usability heuristics, guidelines, standards, and best practices will be researched and discussed. The currently existing security and usability heuristics, guidelines, standards, and best practices will be identified and their relation to instant messaging application development will be examined.

Chapter 5 – Security and Usability Heuristics, Guidelines, Standards and Best Practices

5.1 Introduction

In Chapter 4, security and privacy controls of instant messaging applications were discussed together with the role these controls play in protecting instant messaging users.

The aim of this chapter is to analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development. In so doing, it addresses SRO2 of this study. The requirements of SRO2 are *to identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development*. This was achieved through conducting a detailed content analysis of relevant and widely accepted online documents relating to each.

Heuristics, guidelines, standards, and best practices are utilised to assist developers during the development process by providing insight during this process.

The chapter structure is as follows: Section 5.2 briefly defines heuristics, guidelines, standards, and best practices while Section 5.3 discusses the four-step content analysis process conducted during this study, including planning (Section 5.3.1), data collection (Section 5.3.2), data analysis (Section 5.3.3), and reporting results (5.3.4). Section 5.4 concludes the chapter.

5.2 Defining Heuristics, Guidelines, Standards, and Best Practices

In order to understand fully each of the concepts of heuristics, guidelines, standards, and best practices, each is examined individually before considering them as a whole.

Chami (2017) defines **heuristics** as *'the rules of thumb, which make decision-making easier, especially in complex and uncertain environments by reducing the complexity of assessing probabilities and predicting values to simpler judgments' (pp. 7)*, whereas, Miller et al. (2018) define heuristics as *'systematically designed procedures that do not guarantee an optimal solution, but provide near-optimal solutions' (pp. 20).* From these definitions, heuristics intend to make decisions easier and quicker, while ensuring a near-optimal outcome. Heuristics are more associated with user interface design and can be specific to various domains (Nielsen & Molich, 1990). The goal of heuristics is to assist the heuristics user, which is achieved by easing the pressure on them during the decision-making process.

Cambridge Dictionary (2021) defines **guidelines** as *'information intended to advise people on how something should be done or what something should be' (pp. 1).* Mojapelo (2015), however, defines guidelines as *'a recommended series of suggestions or procedures for accomplishing a given task or achieving a set of goals and objectives' (pp. 40).* Guidelines are not utilised in one specific domain and are used in most organisations and industries. They are adaptable and should be altered depending on their purpose of use (Äijälä, 2018). Guidelines assist their user by providing them with recommended actions, suggestions, and procedures for dealing with situations, ultimately assisting the user's decision-making process.

Äijälä (2018) defines **standards** as *'mandatory activities and rules that must be enforced to be effective' (pp. 15)*. A more specific information technology definition is provided by Gordon and Gordon (2002) as *'allowable characteristics for information processing and communication hardware and software acquired or developed by the company' (pp. 66)*. National, regional, and international standards organisations, like the South African Bureau of Standards (SABS), the International Organization for Standardization (ISO), and the National Institute of Standards and Technology (NIST), typically develop sets of standards. These sets of standards can vary in use and application, to meet specific needs (International Organization for Standardization, 2019). From an information technology perspective, standards are required to ensure that hardware and software are utilised correctly and that they maintain the appropriate standard, as determined by the user, developer, or company.

Charles and Dawson (2011) define **best practices** as *'commonly perceived actions, processes, methodologies or patterns of behaviour that, applied in a specific context, produce superior outcomes and could be used as organisational rules of thumb' (pp. 346).* Best practices are crucial to ensure that systems or workers operate correctly and smoothly. To put it simply, they are the best possible way to do things, based on previous trial and error (Kolar & Grembergen, 2017). This definition indicates that best practices are often used to produce superior processes, behaviour, and outcomes.

The common themes entrenched in heuristics, guidelines, standards, and best practices, are assistance and quality. Heuristics, guidelines, standards, and best practices can all be used to assist their users in maintaining a high quality of work and in producing high quality results.

For most organisations, having heuristics, guidelines, standards, and best practices is beneficial as they are documented and updated according to changing situations and circumstances. In addition, they are intended to explain and provide the appropriate recommendations to accomplish a task (Hamid et al., 2019; Taole, 2020) and should be used for, but not limited to, simplifying instructions and procedures. Heuristics, guidelines, standards and best practices must be adapted to various scenarios and circumstances under which the user intends to use it (Taole, 2020). For instance, to advise interface design, heuristics, meant to guide interface evaluation, may mistakenly be deployed (Renaud & Van Biljon, 2017). This implies that a heuristic, guideline, standard or best practice serves its purpose only if it meets the needs of decision-making assistance, processing measures or offering recommendations (Taole, 2020). One of the most common problems with heuristics, guidelines, standards and best practices is a lack of guidance about how to utilise them effectively to achieve the set objectives (Hamid et al., 2019).

5.3 The Content Analysis Process

Content analysis, as a research technique, requires the use of specialised methods. Krippendorff defines content analysis as *'a research technique for making replicable and valid inferences from texts (or other meaningful matter) to the contexts of their use' (2004, p. 18).* A content analysis is learnable and independent of the researcher's own authority and potential biases. It provides innovative ideas, improves a researcher's comprehension of a certain phenomenon, or directs practical actions. The technique is anticipated to be dependable, and the results should be repeatable. With content analysis the most crucial type of dependability is repeatability (Krippendorff, 2004; Bengtsson, 2016).

The content analysis for this study was conducted based on the steps developed by Bengtsson (2016). Bengtsson (2016) developed the four main steps for a rigorous content analysis based on the writings of content analysis experts, including Downe-Wamboldt, (1992); Morse & Richards, (2002); Patton, (2002); Krippendorff, (2004); and Silverman, (2015).

The content analysis process developed by Bengtsson (2016), consists of the following four steps:

- **Step 1:** Planning
- **Step 2:** Data Collection

- **Step 3:** Data Analysis
  - **Stage 1:** Decontextualisation
  - **Stage 2:** Recontextualisation
  - **Stage 3:** Categorisation
  - **Stage 4:** Compilation
- **Step 4:** Reporting of Results

Figure 5.1 depicts the content analysis process, highlighting each step and how they flow into the following step.

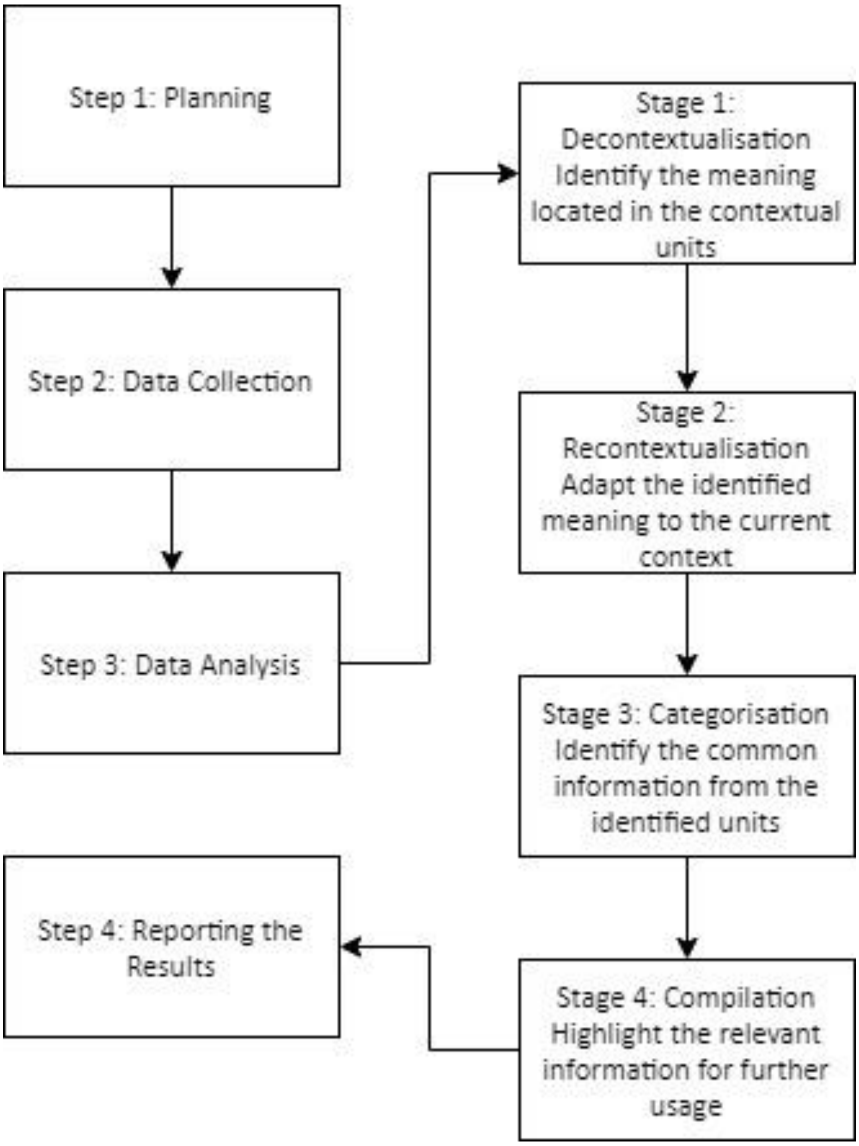

Step 1: Planning
Step 2: Data Collection
Step 3: Data Analysis
Step 4: Reporting the Results
Stage 1: Decontextualisation Identify the meaning located in the contextual units
Stage 2: Recontextualisation Adapt the identified meaning to the current context
Stage 3: Categorisation Identify the common information from the identified units
Stage 4: Compilation Highlight the relevant information for further usage

Figure 5.1 Content Analysis Process (adapted from (Bengtsson, 2016))

These four steps are discussed in Sections 5.3.1 to 5.3.4. According to Figure 5.1, Step 3: Data Analysis consists of four stages. These four stages are discussed in detail in Section 5.3.3.

5.3.1 Step 1: Planning

Before starting a content analysis, one needs to plan and design how to go about conducting the content analysis (Bengtsson, 2016). To ensure that a rigorous process is followed, it is necessary to design the content analysis according to previously established standards. In Section 4.1, Krippendorff (2004, p. 82) states : *'Content analysis has to address prior questions'.* To achieve this, one needs to establish an aim and define the criteria to be used when searching for content.

The aim of this content analysis is to meet SRO2 of this study, namely: *to identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development*.

After establishing the aim, one needs to identify the content to be analysed. The content is also referred to as units. Krippendorff (2004, p. 98–103) identified the following three types of units:

1. **Sampling units** – 'Units that have been distinguished for selective inclusion in an analysis'.
2. **Recording/coding units** – 'Units that are distinguished for separate description, transcription, recording or coding'.
3. **Context units** – 'Units of textual matter that set limits on the information to be considered in the description of recording units'.

For this content analysis, context units were analysed since they were deemed to fit the requirements of this study best. Context units require the individual conducting the content analysis to understand the context of the source of content to ensure an accurate analysis of the information (Krippendorff, 2004, p. 101). The process followed to identify the contextual units is discussed in Section 5.3.2.

Prior to searching for the relevant units, it is important to establish a coding or categorisation scheme. Relating to this, Krippendorff (2004) identified five types of distinctions, namely:

- **Physical** – based on units utilising a physical medium. For example, the digitisation of photographic images.
- **Syntactical** – based on the grammar of the unit of data being examined.
- **Categorical** – based on the units having information in common with one another.
- **Propositional** – based on the construction of the unit. Units that display a particular propositional form or have a certain semantic relationship.
- **Thematic** – units that are based on the analysis of verbal, story-like material.

For the purposes of this study, categorical distinction was utilised for the contextual units identified. From the information provided by Krippendorff (2004), categorical distinctions work best as a categorisation scheme with contextual units. Whereas contextual units are based on the information in the unit, categorical distinctions are focused on the similarities of the contents of a unit.

When categorising and analysing the units, it is important to establish a set of rules to follow (Krippendorff, 2004). To identify the relevant information in the contextual units, categories were established prior to searching for the units. The general and specific categories identified as most relevant to this study are presented in Table 5.1.

Table 5.1 General and Specific Categories Identified

General Categories	Specific Categories
Best practices	Instant messaging
Guidelines	Mobile application development
Heuristics	Security
Standards	Usability
	Usable security

The categories in Table 5.1 assisted in the identification of potential keywords to be used when searching for contextual units. During Step 3: data analysis, the identified contextual units were labelled with one or more of the established categories, based on the relevant information found in the unit.

5.3.2 Step 2: Data Collection

When gathering the content for the analysis, relevant contextual units from reputable sources were gathered. The following databases were utilised to gather the majority of the contextual units:

- Elsevier - https://www.elsevier.com/en-xm
- Emerald Insight - https://www.emerald.com/insight/
- Google Scholar - https://scholar.google.com/
- Institute of Electrical and Electronics Engineers (IEEE) - http://www.ieee.org.za/
- ResearchGate - https://www.researchgate.net/
- Science Direct - https://www.sciencedirect.com/
- Springer - https://www.springer.com/

Keywords and phrases were identified to filter the search results and locate relevant contextual units. Table 5.1 contains all the keywords and phrases utilised for the creation of the search phrases. The general categories were combined with the specific categories and the relevant wildcard, to create the phrases utilised to filter the search results. The wildcard chosen was the asterisk (*), which broadens search results by identifying words that start with the letters before the asterisk. Various combinations of the keywords and phrases presented in Table 5.1 were used in combination with the wildcard to refine the search results and to ensure that relevant contextual units could be located. The final search phrases used included, but were not limited to, the following:

- “Secur*” “Heuristic*” “Instant Messaging”
- “Usability guideline*” “Mobile application develop*”
- "Best practice*" "Mobile application develop*"
- “Secur*” “Standard”
- "Instant messaging" "Usable security” “Heuristic*"
- “Secur*” “Standard” “Mobile application develop*”

The wildcards “AND”, “+”, “OR”, “–“ and “NOT” were not utilised during this search. However, the search results would not change if the “AND” wildcard was used. For example, “Secur*” “Heuristic*” “Instant Messaging” would provide the same search results as “Secur*” AND “Heuristic*” AND “Instant Messaging”.

To further refine the search results, the search phrases were combined with a parameter filter, in the form of a date range from 2017 to 2020. In addition to the units found through the search, 15 units not located during the search were included. These additional units were identified as being seminal sources relevant to this study, as indicated in Table 5.2. The seminal units adhere to the search phrases utilised during the search. The reason for each document not fitting into the search criteria and a motivation for their inclusion are also indicated in Table 5.2.

Table 5.2 Seminal Units Included in the Study

Title and reference	Inclusion reason
10 Usability Heuristics for User Interface Design (Nielsen, 1995)	Jakob Nielsen 10 usability heuristics did not appear during the search, as it did not fall in the 2017 to 2020 date range. These 10 heuristics are commonly referred to when developing domain-specific or general sets of heuristics. Nielsen’s heuristics hold wide recognition and are often referred to as the industry standard. This makes them an important consideration.
IT Security Guidelines for Mobile Apps (National Cyber Security Center (NCSC-NL), 2018)	This unit was not found during the search as it is most likely not located in any of the seven databases utilised. This unit was referred to by other units and upon inspection contained highly valuable and useful information.
OWASP API Security Top 10 2019: The Ten Most Critical API Security Risks (OWASP, 2019)	This unit was not found during the search as it is most likely not located in any of the seven databases utilised. OWASP is the Open Web Application Security Project. OWASP works to improve the software security. In 2019, OWASP released a report on the top 10 application program interface (API) security risks. These risks need to be considered when developing an application to ensure a high standard of security.
OWASP Mobile Top 10 Risks – 2016 (OWASP, 2016)	This unit was not found during the search as it is most likely not located in any of the seven databases utilised. OWASP is the Open Web Application Security Project. OWASP works to improve the software security. This report was released in 2016 and does not fall in the 2017 to 2020 date range. In 2016, OWASP released a report on the top 10 risks to mobile software development. These 10 risks need to be considered as they are directly related to mobile development.
OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks (OWASP, 2017)	This unit was not found during the search as it is most likely not located in any of the seven databases utilised. OWASP is the Open Web Application Security Project. OWASP works to improve the software security. In 2017, OWASP released a report on the top 10 web application security risks. It is important to consider these risks when developing an application. Utilising the OWASP report will assist in alleviating the potential of risks documented in the report.

Information technology — Security techniques — Information security management systems — Requirements ISO/IEC 27001:2013 (International Organization for Standardization, 2013b)	This unit was not found during the search as it did not fall in the date range of 2017 to 2020. The standard units, from the International Organization for Standardization, were not attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards units is critical as they ensure that the current international level of standards is upheld.
Information technology — Security techniques — Code of practice for information security controls ISO/IEC 27002:2013 (International Organization for Standardization, 2013a)	This unit was not found during the search as it did not fall in the date range of 2017 to 2020. The standard units, from the International Organization for Standardization, were not attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards units is critical as they ensure that the current international level of standards is upheld.
Information technology — Security techniques — Information security management systems — Guidance SANS 27003:2020 (South African National Standard, 2020b)	The standard units, from the International Organization for Standardization, were not attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards unit is critical as they ensure that the current international level of standards is upheld.
Information technology — Security techniques — Information security management — Monitoring, measurement, analysis and evaluation SANS 27004:2020 (South African National Standard, 2020a)	The standard units, from the International Organization for Standardization, were not attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards units is critical as they ensure that the current international level of standard is upheld.
Information technology — Security techniques — Information security risk management ISO/IEC 27005:2011 (International Organization for Standardization, 2011)	This unit was not found during the search as it did not fall in the date range of 2017 to 2020. The standard units were, from the International Organization for Standardization, not attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards units is critical as they ensure that the current international level of standards is upheld.
Information technology — Security techniques — Information security management for inter-sector and inter-organisational communications SANS 27010:2018 (South African National Standard, 2018)	This unit was not found during the search as it did not fall in the date range of 2017-2020. The standard units, from the International Organization for Standardization, were not attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards units is critical as they ensure that the current international level of standards is upheld.
Information technology — Security techniques — Information security	This unit was not found during the search as it did not fall in the date range of 2017 to 2020. The standard units, from the International Organization for Standardization, were not

management guidelines for telecommunications organisations based on ISO/IEC 27002 SANS 27011:2009 (South African National Standard, 2009)	attained from any of the seven databases listed. This standard was obtained through the SABS with the access provided by the Nelson Mandela University. The inclusion of standards units is critical as they ensure that the current international level of standards is upheld.
Security Considerations for Code Signing (Cooper et al., 2018)	This standard unit was not found during the search as it was located through the National Institute of Standards and Technology database and not from any of the seven databases listed. It is important to examine the security of the code of an application. Unsecure code can create vulnerabilities in the application.
Security and Privacy Controls for Information Systems and Organizations (NIST SP800-53, 2020)	This standard unit was not found during the search as it was located through the National Institute of Standards and Technology database and not from any of the seven databases listed. Examining standards for security and privacy controls can lead to a more secure application.
Guidelines for Managing the Security of Mobile Devices in the Enterprise (Souppaya & Scarfone, 2013)	This unit was not found during the search as it was located through the National Institute of Standards and Technology database and not from any of the seven databases listed and this unit does not fall in the date range of 2017 to 2020. Reviewing guidelines for mobile device security is important to ensure that the overall security of mobile devices is up to standards and does not create any vulnerabilities for the applications on the devices.

The results of the data collection are summarised in Table 5.3. The term search units utilised in Table 5.3, refers to the contextual units identified when searching through the seven databases.

Table 5.3 Data Collection Results

Type of Unit	Included	Excluded	Total
Seminal units	15	0	**15**
Search units	69	25	**94**
Total	**84**	**25**	**109**

Table 5.3 shows that according to the search phrases and filters used for data collection, as described in this section, 94 units were found. With the data collection results available, Stage 1, decontextualisation, of the four-stage analysis process could begin. According to the aim of the content analysis, 69 units of the 94 were deemed as relevant. The meaning located in the remaining 25 contextual units did not align with the aim of the content analysis; this led to the exclusion of these 25 units. The 69 relevant units and the 15 seminal units were taken forward as the 84 included units.

5.3.3 Step 3: Data Analysis

The next step in the content analysis process was to analyse the gathered units. To ensure a thorough analysis of the units, a four-stage analysis process was utilised, adapted from Bengtsson, (2016). Figure 5.2 illustrates the four-stage analysis process.

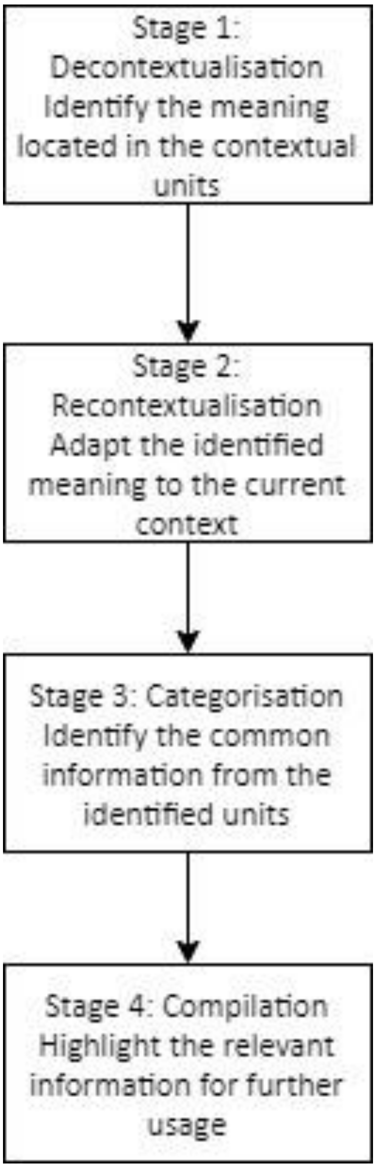

Stage 1:
Decontextualisation
Identify the meaning located in the contextual units
Stage 2:
Recontextualisation
Adapt the identified meaning to the current context
Stage 3: Categorisation
Identify the common information from the identified units
Stage 4: Compilation
Highlight the relevant information for further usage

Figure 5.2 Four-stage Analysis Process (adapted from (Bengtsson, 2016))

Each of the four-stage analysis process will be discussed in their own subsections.

5.3.3.1 Stage 1: Decontextualisation

In Stage 1, decontextualisation, the researcher analysed the contents in the units to become familiar with it. The goal was to understand the aim and purpose of the unit to ensure that the researcher understood what was being discussed and meant by the unit. Once this understanding was achieved, the researcher was able to identify the relevant and not relevant contextual units, Table 5.3 was part of this stage. The relevant data located in the included contextual units was highlighted. Units of meaning were created. They consisted of the unit's aim, purpose, and the highlighted relevant data. For example:

- **Aim –** to identify 10 mobile application heuristics.
- **Purpose –** to improve mobile applications.

- **Highlighted relevant data –** the heuristics that were identified in their study.

These smaller units of meaning decreased the difficulty of working with large quantities of data.

The inclusion and exclusion of contextual units, located in Table 5.3, was conducted in the stage of the analysis process.

5.3.3.2 Stage 2: Recontextualisation

In Stage 2, recontextualisation, the units were re-analysed by the researcher along with the units of meaning. This was done to ensure that the data in the unit and the related unit of meaning align with the aim of the content analysis, established during planning. For example:

- **Aim –** to implement mobile healthcare application guidelines.
- **Purpose –** to improve the operation of healthcare applications.
- **Highlighted relevant data –** the mobile healthcare application guidelines.

In this example, the aim, purpose and highlighted relevant data all seem relevant at surface level. However, when re-analysed, it would be noted that the guidelines do not assist the development of the healthcare application but are guidelines to assist the users in utilising the already developed application. Therefore, this contextual unit would not have aligned with the established aim of this content analysis.

5.3.3.3 Stage 3: Categorisation

For Stage 3, categorisation, the researcher placed the relevant units into homogenous categories, previously stated in Section 5.3.2.

The results of Stage 3, categorisation, are shown in Table 5.4.

Table 5.4 Categorisation Results

Category	Included	Excluded	Total
Security	26	11	**37**
Mobile application development	17	15	**32**
Guidelines	20	11	**31**
Usability	19	11	**30**
Heuristics	11	5	**16**
Usable security	12	0	**12**
Best practices	6	4	**10**
Standards	2	7	**9**
Instant messaging	5	1	**6**
Total	**118**	**65**	**183**

Table 5.4 shows that the total categorised units included (118) are greater than the initial included units (84), as a unit can be categorised into more than one category. The same holds for the total number of categorised units excluded. The 65 units excluded were a result of Stage 2, recontextualisation. The 65 units did not align with the aim of this content analysis. The largest number of included units was found in the Security category (26), while the lowest number was found

in the Standards category (2). The largest number of excluded units was found in the Mobile Application Development category (15), while the lowest number is in the Usable Security category (0).

After the content was categorised, the included contextual units (118) were further analysed to highlight the extent to which they mapped against the identified general and specific categories, as shown in Table 5.5.

Table 5.5 Mapping of Relevant Units Against General and Specific Categories

Category	**Security**	**Usability**	**Mobile application development**	**Usable security**	**Instant messaging**	**Total**
Guidelines	10	10	12	3	1	**36**
Heuristics		8	2	1	3	**14**
Best practices	3		2			**5**
Standards	2					**2**
Total	**15**	**18**	**16**	**4**	**4**	**57**

Referring to Table 5.5, the total number of relevant heuristics, guidelines, standards, and best practices located in the analysed units is 57. The total number of 57 is high as an individual set of existing heuristics, guidelines, standards and best practices can fall into more than one of the following specific categories, as highlighted in Table 5.1, namely security, usability, mobile application development, usable security and instant messaging. The largest category of existing heuristics, guidelines, standards, and best practices is the guidelines category (36), while the lowest is the standards category (2). The largest specific category of heuristics, guidelines, standards and best practices is usability (18), while the lowest is usable security (4) and instant messaging (4).

The discovery of such a low number of usable security heuristics, guidelines, standards, and best practices highlights the need for further research and development in the usable security field. Both general and domain- specific usable security heuristics, guidelines, standards, and best practices are needed globally. In addition, only four existing heuristics, guidelines, standards, and best practices were found relating directly to instant messaging application development. This further supports the need for a set of usable security heuristics for instant messaging application development.

5.3.3.4 Stage 4: Compilation

In Stage 4, compilation, the content deemed most relevant and useful was extracted into Appendices A, B, C, and D. Appendix A consists of the sets of heuristics deemed as most relevant. The sets of guidelines deemed most relevant were compiled into Appendix B. Appendix C consists of the most relevant standards. Lastly, Appendix D is a compilation of the most relevant best practices. The compiled content is reported on in Section 5.3.4.

To address the requirements of Stage 4, compilation, from the 57 existing heuristics, guidelines, standards, and best practices, 37 were identified as most relevant and useful. These 37 have been categorised in Table 5.6.

Table 5.6 Mapping of Most Relevant Units Against General and Specific Categories

Category	Security	Usability	Mobile application development	Usable Security	Instant messaging	Total	Total unique units
Heuristics		7	4	1	3	15	**8**
Guidelines	3	5	6	1		15	**9**
Best practices	3		2			5	**5**
Standards	2					2	**2**
Total	**8**	**12**	**12**	**2**	**3**	**37**	**24**

The 37 identified sources are made up of 24 unique individual units. These units were further analysed to extract the heuristics, guidelines, standards, and best practices, for the creation of Appendices A, B, C, and D. Appendix A consists of eight unique sets of heuristics. Appendix B contains nine unique sets of guidelines. Appendix C comprises two unique groupings of standards. Lastly, Appendix D consists of five unique sets of best practices. All the heuristics, guidelines, standards, and best practices, compiled into appendices A, B, C, and D, have been taken directly from their respective sources.

Appendices A, B, C, and D have been coded in the following way. Heuristics have been labelled with the letter H, guidelines with the letter G, standards with the letter S and best practices with the letter B. In addition, each set of heuristics, guidelines, standards, and best practices, located in these appendices, has been numbered sequentially from 01. Each individual heuristic, guideline, standard and best practice within each source, has also been numbered. This creates a code that would look like the following: H.01.01. This code would refer to the first heuristic located in the first set of heuristics. The codes for guidelines, standards and best practices all look and function in the same manner.

Referring to the heuristics located in Appendix A, the majority of the sets of heuristics have some relation to or have been based on Nielsen's 10 Usability Heuristics for User Interface Design (Nielsen, 1995). The sets of guidelines, standards and best practices are unique in comparison to the heuristics, although there are similar guidelines mentioned throughout the sets of guidelines.

5.3.4 Step 4: Reporting of Results

Appendices A, B, C, and D contain the relevant existing heuristics, guidelines, standards, and best practices. The development of these appendices satisfies the aim, which was defined in Section 5.3.1. These appendices were used in the development of the set of usable security heuristics for instant messaging application development proposed in Chapter 6. In addition, they highlight the relevance of the identified heuristics, guidelines, standards, and best practices that could assist in alleviating the potential instant messaging threats identified in Chapter 3, Section 3.5.

In Chapter 3, Section 3.5, confidential information leakage, distribution of malicious code, man-in-the-middle attack, permission system vulnerability and social engineering were identified as potential threats to IM applications. In addition to those identified, the OWASP Top 10 Mobile Risks (OWASP, 2016) were also considered, based on their relevance to IM.

5.3.4.1 Confidential Information Leakage

As stated in Chapter 2, Section 2.3.1.1, confidential information leakage is known as the accidental or deliberate dissemination of confidential information to an unauthorised party. The security and usability heuristics, guidelines, standards, and best practices indicated in Table 5.7 were identified, from Appendices A, B, C, and D as being able to assist in the mitigation of confidential information leakage.

Table 5.7 Existing Heuristics, Guidelines, and Best Practices Relevant to Confidential Information Leakage

Code	Name of heuristic, guideline, standard or best practice	Relevance to confidential information leakage
H.02.05, H.04.05, H.05.06	**Error prevention**	As stated in the definition of error prevention, users should receive a warning when conducting critical actions. For example, when sending confidential banking information across an IM platform, users should receive a warning about the dangers and risks involved when sharing this confidential information.
H.03.02	**Reliable**	Security and privacy features must be available and explained in plain text. Users need to understand how to implement the feature and how the feature will assist in ensuring their security and privacy is maintained. Assisting users to understand their security and privacy features could lead to a reduction in confidential information leakage.
H.05.14, H.07.13, G.07.12, S.02.10, B.01.12	**Privacy**	Organisations collect confidential information on their users, through the applications they develop. Organisations need to ensure that this confidential information is only used for the reasons stated in the user agreement and that the confidential information is protected from unauthorised personnel. Protecting this confidential information will assist in alleviating confidential information leakage.
H.07.14, H.08.11	**Security and recovery of user account**	As stated in the definition, the account of a user should be highly secure. The personal information about the user is in their account details. If there is not adequate security to protect this confidential information, it could be leaked to malicious individuals and cause harm to the user.
G.02.12, G.07.11, S.02.10, B.01.13, B.02.10, B.04.08	**Security**	All security mechanisms and features should ensure that the fundamental information security principles of confidentiality, integrity and availability (CIA) are upheld. Ensuring that CIA is upheld assists in avoiding unauthorised access and disclosure of confidential information from the application.
G.05.11	**Provide security mechanisms and information to the user**	Implementing the necessary security mechanisms will deter attackers from targeting applications. By increasing the security, you increase the difficulty for an attacker. The more secure the user's confidential information, the less likely it is that this confidential information will be leaked. Providing the security is not enough, the user also needs to be informed about the security, how it works and how to ensure it is activated. Having the security mechanism is meaningless if it is not activated.
G.08.02	**Secure server-side application**	Applications communicate with a server to ensure that their functionality is successful. If the server side is not secured, this would lead to confidential information leakage from the server.

G.08.03, B.04.10	**Third-party apps**	Third-party applications are utilised when implementing the functionality of an application. For example, document viewers and keyboards. These applications could have hidden malicious functionality that can lead to confidential information leakage. Developers should ensure that third-party applications cannot access the confidential information processed on their applications.
G.08.04	**Secure code on delivery**	Utilising external code libraries can assist in accelerating the development of applications. These external libraries can introduce additional vulnerabilities to the application. These vulnerabilities could be manipulated to leak confidential information. Securing the external libraries and ensuring that they do not pose a threat to the application is crucial.
G.08.05	**Secure operation of the app**	Applications run on the device on which they are loaded. This creates a vulnerability if the device on which the application is becomes compromised. The malicious individual who compromises the device can access and manipulate all information and applications on the device. This manipulation can lead to confidential information leakage. The application needs to ensure that it is secure at a running level to prevent manipulation from a malicious individual.
G.08.19	**Up-to-date apps**	Maintaining up-to-date applications ensures that previously identified vulnerabilities are addressed. Malicious individuals will target outdated applications as their vulnerabilities are more well-known and easier to manipulate. Up-to-date applications will assist in preventing threats such as confidential information leakage.
S.02.01	**Access enforcement**	Enforcing access control will limit unauthorised access to confidential information. Ensuring authorised access to information and system resources will reduce the possibility of confidential information leakage.
S.02.08	**Device lock**	Securing devices after inactivity or leaving a workstation improves the security of that device. This also deters malicious individuals from attempting to break into the device and steal confidential information.

From Table 5.7, it is evident that Security appeared the most (6), while Security and recovery of user account and Third-party apps appeared the least (2). Including the security and usability heuristics, guidelines, and best practices that appeared in multiple appendices, Table 5.7 consists of a final set of: eight heuristics (from Appendix A), nine guidelines (from Appendix B), four standards (from Appendix C) and five best practices (from Appendix D).

5.3.4.2 Distribution of Malicious Code

As stated in Chapter 2, Section 2.3.1.2, malware refers to self-replicating malicious software, intended to perform undesirable acts on the network, that distributes over a network without interaction or initiation, (Baror & Venter, 2019). The security and usability heuristics, guidelines, standards, and best practices indicated in Table 5.8 were identified, from Appendices A, B, C, and D, as being able to assist in the mitigation of the distribution of malicious code.

Table 5.8 Existing Heuristics, Guidelines, Standards, and Best Practices Relevant to Distribution Of Malicious Code

Code	Name of heuristic, guideline, standard or best practice	Relevance to distribution of malicious code
H.03.02	**Reliable**	Security and privacy features must be available and explained in plain text. Users need to understand how to implement the feature and how the feature will assist in ensuring that their security and privacy is maintained. Assisting users to understand their security and privacy features could lead to a reduction in distribution of malicious code.
H.05.14, H.07.13, G.07.12, S.02.10, B.01.12	**Privacy**	Organisations collect confidential information on their users, through the applications they develop. Organisations need to ensure that this confidential information is only used for the reasons stated in the user agreement and that the confidential information is protected from unauthorised personnel. Protecting this confidential information will assist in alleviating threats to the users and the application, such as the distribution of malicious code.
G.02.12, G.07.11, S.02.10, B.01.13, B.02.10, B.04.08	**Security**	All security mechanisms and features should ensure that the fundamental information security principles of confidentiality, integrity and availability (CIA) are upheld. Ensuring that CIA is upheld assists in avoiding unauthorised access and disclosure of information from the application. By maintaining the CIA principles, threats like the distribution of malicious code, can be reduced.
G.05.11	**Provide security mechanisms and information to the user**	Implementing the necessary security mechanisms will deter attackers from targeting applications. By increasing the security, the difficulty is increased for an attacker. The more secure the application, the less likely it is that the application will be targeted, which can potentially reduce the occurrence of distribution of malicious code. Providing the security is not enough, the user also needs to be informed about the security, how it works, and how to ensure that it is activated. Having the security mechanism is meaningless if it is not activated.
G.08.03, B.04.10	**Third-party apps**	Third-party applications are utilised when implementing the functionality of an application. For example, document viewers and keyboards. These applications could have hidden malicious functionality that can lead to the distribution of malicious code. Developers should ensure that third-party applications cannot manipulate the functionality of their applications.
G.08.04	**Secure code on delivery**	Utilising external code libraries can assist in accelerating the development of applications. These external libraries can introduce additional vulnerabilities to the application. These vulnerabilities could be manipulated to distribute malicious code across the platform. Securing the external libraries and

		ensuring that they do not pose a threat to the application is crucial.
G.08.05	**Secure operation of the app**	Applications run on the device on which they are loaded. This creates a vulnerability if the device on which the application is becomes compromised. The malicious individual who compromises the device can access and manipulate all information and applications on the device. This manipulation can lead to the distribution of malicious code. The application needs to ensure that it is secure at a running level to prevent manipulation from a malicious individual.
G.08.14, S.02.03	**Principle of least privilege for other apps**	Least privilege ensures that authorised access is only given to accomplished necessary processes. This limits access to system resources and information, which could hinder the distribution of malicious code.
G.08.19	**Up-to-date apps**	Maintaining up-to-date applications ensures that previously identified vulnerabilities are addressed. Malicious individuals will target outdated applications as their vulnerabilities are more well-known and easier to manipulate. Up-to-date applications will assist in preventing threats such as the distribution of malicious code.
G.09.08, B.04.05	**Code tampering**	Attackers will modify the code of applications for third-party app stores and attempt to trick users into installing their malicious modified version of the application. This modified version of the application introduces threats to the user, such as the distribution of malicious code.
S.01.06	**Controls against malicious code**	Applications should feature controls to detect, prevent and recover from malware, such as the distribution of malicious code. These controls should be implemented and combined with the appropriate user awareness to ensure that malware attacks are not successful.
S.01.07	**Controls against mobile code Control**	Applications should ensure that only the authorised mobile code is executed and operates according to the defined security policy. No unauthorised mobile code should be executed. This assists in reducing the distribution of malicious code.

From Table 5.8, it is evident that Security appeared the most (6), Third-party apps, Principle of least privilege for other apps and Code tampering appeared the least (2). Including the security and usability heuristics, guidelines, and best practices that appeared in multiple appendices, Table 5.8 consists of a final set of: three heuristics (from Appendix A), ten guidelines (from Appendix B), five standards (from Appendix C) and six best practices (from Appendix D).

5.3.4.3 Man-in-The-Middle Attack

As stated in Chapter 2, Section 2.3.1.3, Man-in-the-middle is an active network attack in which an attacker is positioned remotely to capture, transmit, and receive interactions between two or more parties. The security and usability heuristics, guidelines, standards, and best practices indicated in Table 5.9 were identified from Appendices A, B, C ,and D as being able to assist in the mitigation of Man-in-the-middle attacks.

Table 5.9 Existing Heuristics, Guidelines, Standards, and Best Practices Relevant to Man-in-the-Middle Attack

Code	Name of heuristic, guideline, standard or best practice	Relevance to Man-in-The-Middle Attack
H.05.14, H.07.13, G.07.12, S.02.10, B.01.12	**Privacy**	Organisations collect confidential information on their users, through the applications they develop. Organisations need to ensure that this confidential information is only used for the reasons stated in the user agreement and that the confidential information is protected from unauthorised personnel. Securing the application and protecting this confidential information will assist in alleviating threats, such as Man-in-the-middle attacks.
H.07.14, H.08.11	**Security and recovery of user account**	As stated in the definition, the account of a user should have recovery options. During a Man-in-the-middle attack, a malicious individual could hijack a user's account and impersonate them. The account recovery option could assist in mitigating the man-in -the-middle attack and return the account to the user.
G.02.12, G.07.11, S.02.10, B.01.13, B.02.10, B.04.08	**Security**	All security mechanisms and features should ensure that the fundamental information security principles of confidentiality, integrity and availability (CIA) are upheld. Ensuring that CIA is upheld, assists in avoiding unauthorised access and disclosure of information from the application. By maintaining the CIA principles threats, like Man-in-the-middle attacks, can be reduced.
G.08.02	**Secure server-side application**	Applications communicate with a server to ensure that their functionality is successful. If the server side is not secured, this would create a vulnerability which can be manipulated by man-in-the middle attacks.
G.08.11	**Transport encryption**	Encrypting the information transmitted from the application to the server has become a crucial component in securing an application. The communications could be transmitted across a publicly accessible server, which creates more vulnerabilities for attack. The lack of encryption creates a vulnerability that could be manipulated and exploited by threats, such as Man-in-the-middle attacks
G.08.19	**Up-to-date apps**	Maintaining up-to-date applications ensures that previously identified vulnerabilities are addressed. Malicious individuals will target outdated applications as their vulnerabilities are more well-known and easier to manipulate. Up-to-date applications will assist in preventing threats such as man-in-the-middle attacks.
G.09.03	**Insecure communication**	Encrypting the information transmitted from the application to the server has become a crucial component in securing an application. The communications could be transmitted across a publicly accessible server, which creates more vulnerabilities for attack. The lack of encryption creates a vulnerability that

		could be manipulated and exploited by threats, such as Man-in-the-middle attacks.
G.09.08, B.04.05	**Code tampering**	Attackers will modify the code of applications for third-party app stores and attempt to trick users into installing their malicious modified version of the application. This modified version of the application introduces threats to the user, such as Man-in-the-middle attacks.
S.01.07	**Controls against mobile code control**	Applications should ensure that only the authorised mobile code is executed and operates according to the defined security policy. No unauthorised mobile code should be executed. This assists in reducing the threat of Man-in-the-middle attacks.
S.02.11	**Wireless access**	Ensuring authorisation for wireless access will improve the security of the application and device. This will also deter and hinder attackers, such as Man-in-the-middle attacks.
B.04.16	**Configure the software to have secure settings by default**	Ensuring that application is not deployed and utilised with weaker security settings decreases its chances of being compromised. This also leads to an improvement of overall security.

From Table 5.9, it is evident that Security appeared the most (6), Security and recovery of user account and Third-party apps appeared the least (2). Including the security and usability heuristics, guidelines, and best practices that appeared in multiple appendices, Table 5.9 consists of a final set of: four heuristics (from Appendix A), eight guidelines (from Appendix B), four standards (from Appendix C) and six best practices (from Appendix D).

5.3.4.4 Permission System Vulnerability

As stated in Chapter 2, Section 2.3.1.4, permission-based mechanisms are used extensively to restrict each applications operation and the user information and device resources that the application can access. The security and usability heuristics, guidelines, standards, and best practices indicated in Table 5.10 were identified, from Appendices A, B, C, and D as being able to assist in the mitigation of potential breaches of the permission-based mechanisms.

Table 5.10 Existing Heuristics, Guidelines, Standards, and Best Practices Relevant to Permission System Vulnerability

Code	Name of heuristic, guideline, standard or best practice	Relevance to permission system vulnerability
H.05.14, H.07.13, G.07.12, S.02.10, B.01.12	**Privacy**	Organisations collect confidential information on their users, through the applications they develop. Organisations need to ensure that this confidential information is used only for the reasons stated in the user agreement and that the confidential information is protected from unauthorised personnel. Collecting and utilising the confidential information, as stated in the user agreement, will assist in mitigating the permission system vulnerability.
G.02.12, G.07.11, S.02.10,	**Security**	All security mechanisms and features should ensure that the fundamental information security principles of confidentiality, integrity and availability (CIA) are upheld. Ensuring that CIA is

B.01.13, B.02.10, B.04.08		upheld, assists in avoiding unauthorised access and disclosure of information from the application. By maintaining the CIA principles, threats like the permission system vulnerability, can be reduced.
G.08.03, B.04.10	**Third-party apps**	Third-party applications are utilised when implementing the functionality of an application. For example, document viewers and keyboards. These applications could have flaws or vulnerabilities, such as the permission system vulnerability. Developers should ensure that third-party applications cannot access the confidential information processed on their applications.
G.08.04	**Secure code on delivery**	Utilising external code libraries can assist in accelerating the development of applications. These external libraries can introduce additional vulnerabilities to the application. These vulnerabilities could be related to the permission system of the application, which would result in the application having a permission system vulnerability. Securing the external libraries and ensuring that they do not pose a threat to the application is crucial.
G.08.14, S.02.03	**Principle of least privilege for other apps**	Ensuring that an application is limited to the privileges/permissions it requires for its functionality is crucial. Overprivileged applications create vulnerabilities which can be manipulated, such as the permission system vulnerability.
G.08.19	**Up-to-date apps**	Maintaining up-to-date applications ensures that previously identified vulnerabilities are addressed. Malicious individuals will target outdated applications as their vulnerabilities are more well-known and easier to manipulate. Up-to-date applications will assist in preventing threats such as the permission system vulnerability.
G.09.08, B.04.05	**Code tampering**	Attackers will modify the code of applications for third-party app stores and attempt to trick users into installing their malicious modified version of the application. This modified version of the application introduces threats to the user, such as the permission system vulnerability.
S.01.07	**Controls against mobile code Control**	Applications should ensure that only the authorised mobile code is executed and operates according to the defined security policy. No unauthorised mobile code should be executed. This assists in reducing threats, such as the permission system vulnerability.
B.04.16	**Configure the software to have secure settings by default**	Ensuring that the application is not deployed and utilised with weaker security settings, decreases its chances of being compromised. This also leads to an improvement of overall security.

From Table 5.10, it is evident that Security appeared the most (6), Third-party apps, Principle of least privilege for other apps and Code tampering appeared the least (2). Including the security and usability heuristics, guidelines, and best practices that appeared in multiple appendices, Table 5.10 consists of a final set of: two heuristics (from Appendix A), eight guidelines (from Appendix B), four standards (from Appendix C) and seven best practices (from Appendix D).

5.3.4.5 Social Engineering

As stated in Chapter 2, Section 2.3.1.5, the art of manipulating human weaknesses to achieve a malicious objective is referred to as social engineering. Social engineering is a technique that requires no advanced specialised technology, can be used by anyone, and is affordable. The security and usability heuristics, guidelines, and best practices indicated in Table 5.11 were identified from Appendices A, B, C and D as being able to assist in the mitigation of social engineering.

Table 5.11 Existing Heuristics, Guidelines, Standards and Best Practices Relevant to Social Engineering

Code	Name of heuristic, guideline, standard or best practice	Relevance to social engineering
H.02.10, H.04.09, H.05.11, H.07.09	**Help users recognise, diagnose, and recover from errors**	Informing users of errors in plain language to assist the users to understand and providing precise instructions on how to recover from these errors, can lead to a reduction in the success of social engineering.
H.03.04	**Assistive**	Guiding users through the usage of an application and keeping the user informed during the decision-making process, can lead to a reduction in the success of social engineering.
H.05.14, H.07.13, G.07.12, S.02.10, B.01.12	**Privacy**	The application should ensure that the user is in control of their information. Users need to authorise the usage of their information by the application, and the application should only use their information in the way agreed upon. The application should keep users informed about this, to assist in mitigating social engineering.
G.02.12, G.07.11, S.02.10, B.01.13, B.02.10, B.04.08	**Security**	All security mechanisms and features should ensure that the fundamental information security principles of confidentiality, integrity and availability (CIA) are upheld. Ensuring that CIA is upheld, assists in avoiding unauthorised access and disclosure of information from the application. By maintaining the CIA principles and keeping users informed, threats like social engineering can be reduced.
G.07.09	**User suitability**	Keeping users informed and providing options that are suitable for them, should assist in their overall usage of the application. Providing the options they are familiar with will also assist in their recovery from errors and prevent the success of threats, such as social engineering.

From Table 5.11, it is evident that Security appeared the most (6), help users recognise, diagnose, and recover from errors appeared the least (4). Including the security and usability heuristics, guidelines, and best practices that appeared in multiple appendices, Table 5.10 consists of a final set of: seven heuristics (from Appendix A), four guidelines (from Appendix B), two standards (from Appendix C) and four best practices (from Appendix D).

From the identified existing security and usability heuristics, guidelines, standards, and best practices in Appendices A, B, C, and D, those deemed relevant to the identified IM threats will be utilised in Chapter 6 to create the preliminary set of usable security heuristics for instant messaging application development.

5.4 Conclusion

The content analysis conducted identified many existing security and usability heuristics, guidelines, standards, and best practices. Though there were many existing security and usability heuristics, guidelines, standards, and best practices, not all of the existing security and usability heuristics, guidelines, standards, and best practices were relevant to instant messaging application development. From the existing security and usability heuristics, guidelines, standards, and best practices which were identified as relevant to instant messaging application development, those that were relevant to the identified information security threats to instant messaging, located in Chapter 3, Section 3.5, were highlighted. The relevance of the security and usability heuristics, guidelines, standards, or best practices was addressed with regard to each of the identified information security threats to instant messaging.

These relevant existing security and usability heuristics, guidelines, standards, and best practices will be utilised going forward, to assist in the creation of the preliminary set of usable security heuristics for instant messaging application development, in Chapter 6.

Chapter 6 – Proposed Set of Usable Security Heuristics for Instant Messaging Application Development

6.1 Introduction

In Chapter 5, existing security and usability heuristics, guidelines, standards, and best practices and their relevance to instant messaging application development were discussed.

The aim of this chapter is to meet the requirements of SRO3 and the PRO. The requirements of SRO3 are *to map the identified security and usability heuristics, guidelines, standards, and best practices to instant messaging application development*. The requirements of the PRO are *to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*.

The chapter structure is as follows: Section 6.2 describes the process followed in developing the proposed set of usable security heuristics. Section 6.3 focuses on the adaption of the security and usability heuristics, guidelines, standards, and best practices and how they will work for instant messaging application development. In so doing, it addresses SRO3 of this study. Section 6.4 presents the preliminary set of usable security heuristics for instant messaging application development, while Section 6.5 maps the preliminary heuristics against the previous identified instant messaging threats and security and privacy controls. Section 6.6 concludes the chapter.

6.2 The Heuristic Development Process

The development of heuristics is a widely researched topic with many different experts producing different results. For this study, the four steps of Quiñones and Rusu (2017) to create usability heuristics were considered. They propose four simple steps to create usability heuristics, based on the studies of experts in the field of the development and creation of heuristics. The studies examined by Quiñones and Rusu (2017) include:

- Evaluating a methodology to establish usability heuristics (Jimenez et al.,2012)
- A three-phase process to develop heuristics (van Greunen et al., 2011)
- A user-centric methodology to establish usability heuristics for specific domains (Hermawati & Lawson, 2018)
- User involvement in developing usability heuristics (Lechner et al., 2013)
- Heuristic evaluation of usability of public administration portal (Hub & Čapková, 2010)
- Usability heuristics for collaborative augmented reality remote systems (Franklin et al., 2014)
- Design science in information systems research (Hevner et al., 2004)

The resulting four steps proposed by Quiñones and Rusu (2017) are as follows:

- **Step 1:** ‘Determine the specific features of the application in order to evaluate these features based on the new set of heuristics’.

- **Step 2:** 'Identify existing sets of usability heuristics in order to determine how these existing sets can help to define the new heuristics (for instance, which heuristics can be reused and which elements to use to define heuristics)'.
- **Step 3:** 'Specify the new set of heuristics following a standard template in order to obtain a set of heuristics that is well defined and easy to understand'.
- **Step 4:** 'Validate the new set of heuristics in order to determine if the heuristics make it possible (1) to find usability problems; and (2) to detect specific usability problems related to the application' (pp. 99).

Since this four-step process was initially proposed for the development of usability heuristics, it required minor modifications in order to suit this study and the development of usable security heuristics. It was important to consider the definition of usable security, as presented in Chapter 1, Section 1.1, when adapting the four-step process. The usability aspect of the four-step process meets the usable requirement of usable security, which is to ensure that the required level of effectiveness, efficiency and satisfaction are obtained, as stated in Chapter 1, Section 1.1. However, security needs to be incorporated into the four-step process to meet the adequate security requirements, as stated in Chapter 1, Section 1.1. The following four-step process, adapted from Quiñones and Rusu (2017), was utilised in the development of the proposed set of usable security heuristics:

- **Step 1:** Determine the most prominent instant messaging (IM) application threats and the related IM security controls and features. Utilise the new set of usable security heuristics to evaluate these security controls and features.
- **Step 2:** Identify existing security and usability heuristics, guidelines, standards, and best practices in order to determine how they can assist in defining a set of usable security heuristics for IM applications.
- **Step 3:** Define a set of usable security heuristics for IM applications following a rigorous approach, which is clearly stated and easy to understand.
- **Step 4:** Validate the proposed set of usable security heuristics to its efficacy, utility and quality, and its applicability to IM applications.

Table 6.1 displays each of the four steps, their respective outputs, and the location of their output within this study.

Table 6.1 Usable Security Steps and Related Outputs

Step	**Output**	**Chapter/Section**
Step 1	Most prominent IM application threats IM security and privacy controls	Chapter 3, Section 3.5 Chapter 4, Section 4.5
Step 2	Existing usability and security heuristics, guidelines, standards, and best practices	Chapter 5 Appendices A,B,C, and D
Step 3	New set of usable security heuristics	Chapter 6, Section 6.4
Step 4	Validated set of usable security heuristics	Chapter 7

To achieve the requirements set out by Step 1, the most prominent IM application threats were examined in Chapter 3, Section 3.4; and in Chapter 4, Section 4.5, the current available IM security, and privacy controls were discussed. These IM threats and security controls are an important factor to be aware of when considering the heuristics to be included for the preliminary set of usable security heuristics for instant messaging application development.

To meet the goal set out by Step 2, in Chapter 5, existing security and usability heuristics, guidelines, standards, and best practices were identified. Appendices A, B, C, and D were developed based on the existing security and usability heuristics, guidelines, standards, and best practices, which were deemed most relevant to the IM context.

The appendices identified in Step 2 were utilised in Chapter 5, Section 5.3.5, to identify existing security and usability heuristics, guidelines, standards, and best practices, which could assist in alleviating the most prominent IM threats, identified in Chapter 3, Section 3.5. The combination of the IM security and privacy controls (from Chapter 4, Section 4.5), the appendices, and the existing security and usability heuristics, guidelines, standards, and best practices which could assist in alleviating the most prominent IM threats (from Chapter 5, Section 5.3.5) will be utilised in the development of the preliminary set of usable security heuristics for instant messaging application development. This preliminary set of usable security heuristics for instant messaging application development will meet the requirements of Step 3.

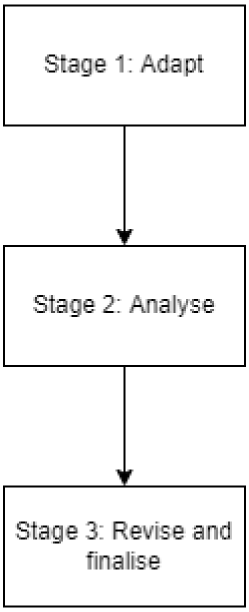


Figure 6.1 Three-stage Rigorous Process

Figure 6.1 displays the three stages to the rigorous process which will be followed within Step 3. Each stage will be labelled and documented.

Stage 1: The current existing security and usability heuristics, guidelines, standards, and best practices, which have been deemed relevant, were adapted to align with the requirements of usable security. These security and usability heuristics, guidelines, standards, and best practices were also aligned with the IM application environment.

Stage 2: The now usable security heuristics were examined to ensure that each usable security heuristic aligns with both the usability and security aspects, defined by usable security. If a usable security heuristic did not align with these requirements, it was either further adapted, combined with another heuristic to meet the requirements, or removed from the set of usable security heuristics.

Stage 3: The set of usable security heuristics will be reviewed to ensure that the heuristic names and definitions align with the requirements of both usable security and IM applications. The proposed set of usable security heuristics will be finalised at this stage.

The validation of the proposed preliminary set of usable security heuristics for instant messaging application development will be achieved through an expert review and a proof-of-concept prototype. The validation will be accomplished in Chapter 7 and will meet the requirements for Step 4.

As Step 1 and Step 2 have been accomplished, the following sections will discuss Stages 1-3 and the completion of Step 3.

6.3 Stage 1: Adapt

The adaption of the current relevant existing security and usability heuristics, guidelines, standards, and best practices is an important step in the development of the set of usable security heuristics. To ensure the accuracy and uniformity of the set of usable security heuristics, the term security feature is used in this study as it encompasses security controls, mechanisms, and other general security features deemed relevant to this study.

The following section documents Step 3, Stage 1, of the three-stage rigorous process.

In Chapter 3, Section 3.5, confidential information leakage, distribution of malicious code, Man-in-the-middle attack, permission system vulnerability and social engineering were identified as the five most prominent threats to IM applications. In Chapter 5, Section 5.3.4, these five threats were analysed, and security and usability heuristics, guidelines, standards ,and best practices were mapped to the five threats. These security and usability heuristics, guidelines, standards, and best practices were compiled in Table 6.2.

Table 6.2 Heuristics, Guidelines, Standards, and Best Practices Mapped to the Five IM Threats

Code	Name of heuristic, guideline, standard or best practice	Threats allocated to
H.02.05, H.04.05, H.05.06	**Error prevention**	Confidential information leakage
H.02.10, H.04.09, H.05.11, H.07.09	**Help users recognise, diagnose, and recover from errors**	Social engineering
H.03.02	**Reliable**	Confidential information leakage and distribution of malicious code
H.03.04	**Assistive**	Social engineering
H.05.14, H.07.13, G.07.12, S.02.10, B.01.12	**Privacy**	Confidential information leakage, distribution of malicious code, man-in-the-middle attack, permission system vulnerability, and social engineering
H.07.14, H.08.11	**Security and recovery of user account**	Confidential information leakage and man-in-the-middle attack

G.02.12, G.07.11, S.02.10, B.01.13, B.02.10, B.04.08	**Security**	Confidential information leakage, distribution of malicious code, man-in-the-middle attack, permission system vulnerability and social engineering
G.05.11	**Provide security mechanisms and information to the user**	Confidential information leakage and distribution of malicious code
G.07.09	**User suitability**	Social engineering
G.08.02	**Secure server-side application**	Confidential information leakage and man-in-the-middle attack
G.08.03, B.04.10	**Third-party apps**	Confidential information leakage, distribution of malicious code and permission system vulnerability
G.08.04	**Secure code on delivery**	Confidential information leakage, distribution of malicious code and permission system vulnerability
G.08.05	**Secure operation of the app**	Confidential information leakage and distribution of malicious code
G.08.11	**Transport encryption**	Man-in-the-middle attack
G.08.14, S.02.03	**Principle of least privilege for other apps**	Distribution of malicious code and permission system vulnerability
G.08.19	**Up-to-date apps**	Confidential information leakage, distribution of malicious code, man-in-the-middle attack, and permission system vulnerability
G.09.03	**Insecure communication**	Man-in-the-middle attack
G.09.08, B.04.05	**Code tampering**	Distribution of malicious code, man-in-the-middle attack, and permission system vulnerability
S.01.06	**Controls against malicious code**	Distribution of malicious code
S.01.07	**Controls against mobile code control**	Distribution of malicious code, man-in-the-middle attack, and permission system vulnerability
S.02.01	**Access enforcement**	Confidential information leakage
S.02.08	**Device lock**	Confidential information leakage
S.02.11	**Wireless access**	Man-in-the-middle attack
B.04.16	**Configure the software to have secure settings by default**	Man-in-the-middle attack and permission system vulnerability

Table 6.2 consists of the 24 identified security and usability heuristics, guidelines, standards, and best practices, the code related to the security and usability heuristics, guidelines, standards, and best practice, and which of the five most prominent IM threats it is related to. These 24 security and usability heuristics, guidelines, standards, and best practices were examined and adapted to the IM context.

From the 24 security and usability heuristics, guidelines, standards, and best practices identified in Table 6.2, a number of them address similar concerns and could result in the same outcome. Presenting a set of heuristics that contain heuristics which result in the same outcome is not a concise set of heuristics. Before adding any of the security and usability heuristics, guidelines, standards, and best practices listed in Table 6.2, the similarities between individual security and usability heuristics, guidelines, standards, and best practices need to be addressed. This was achieved by grouping similar security and usability heuristics, guidelines, standards, and best practices, based on their definitions. The security and usability heuristics, guidelines, standards, and best practices deemed as similar were combined to create a new usable security heuristic and the definition of this heuristic was adapted to the IM context.

Table 6.3 Adaption and Combination of Similar Heuristics, Guidelines, Standards, and Best Practices to IM Context

Name and Code	Similarities	New heuristic name	New definition
Error prevention (H.02.05, H.04.05, H.05.06) **Help users recognise, diagnose, and recover from errors** (H.02.10, H.04.09, H.05.11, H.07.09) **Assistive** (H.03.04)	The definitions, listed in Appendix A for the identified heuristics focus on assisting the users while utilising the system. They also highlight warning and communication with the user, during usage in plaintext format.	**Threat prevention and user guidance**	IM applications should present security messages in a plaintext format to the user. IM applications should guide the user during usage, by hiding unavailable functions, warn users about their actions. and assist users to recognise, diagnose, and avoid potential threats.
Reliable (H.03.02) **Privacy** (H.05.14, H.07.13, G.07.12, S.02.10, B.01.12) **Security and recovery of user account** (H.07.14, H.08.11) **Security** (G.02.12, G.07.11, S.02.10, B.01.13, B.02.10, B.04.08) **Provide security mechanisms and information to the user** (G.05.11)	All definitions, listen in appendices A, B, C, and D, for the identified heuristics, guidelines, standards, and best practices focus on securing the user and their personal information according to the CIA information security principles. The definitions highlight how the user account not only needs to be secure but also recoverable. The definitions mention that this can be achieved through the implementation of security and privacy measures.	**Compliance of security and privacy controls**	IM applications must provide the current industry standard of security and privacy controls with basic plaintext instructions for users, on how to implement and utilise these features effectively. The privacy features need to align with international standards, such as the South African Protection of Personal Information Act (POPIA) and the European General Data Protection Regulation (GDPR).
Third-party apps (G.08.03, B.04.10)	The definitions, listed in Appendices B and D, for the identified guidelines and	**Securing from third-party**	IM applications, its code, the information stored in it, and the information

Secure code on delivery (G.08.04) **Code tampering** (G.09.08, B.04.05)	best practices, highlight securing the application, its code, the information stored in it, and the information located on the device from third-party sources. The third-party sources include third-party applications, external code libraries, and third-party app stores.		located on the device, must be secured from third-party sources. These third-party sources include but are not limited to third-party applications, external code libraries, and third-party app stores.
Transport encryption (G.08.11) **Insecure communication** (G.09.03)	The definitions located in Appendix B, for the identified guidelines, highlight the importance of encryption. Encrypting the application session and the storage and transmission of information will increase the security of the application.	**Encryption of application session and information**	IM applications need to be encrypted to the current industry level of encryption. The encryption level must be made clear to the user. If there is more than one level of encryption available, it must be clear which is active, and the user must be guided on how to select the relevant encryption feature. It is crucial for an IM application to encrypt the application session and the storage and transmission of information.
Principle of least privilege for other apps (G.08.14, S.02.03) **Controls against mobile code control** (S.01.07)	The definitions, listed in Appendices B and C, for the identified guidelines and standards, focus on limiting applications access to the system resources. Limiting applications to the minimum system resources required is mentioned in each definition.	**Least privilege by default**	IM applications need to be developed with the principle of least privilege, which is to ensure that the permissions requested by the application are limited to the minimum permissions required for functionality. Each permission requested must be clearly and concisely explained to the user, to ensure that an informed decision is made by the user. This will also reduce the cognitive load on the user.
Access enforcement (S.02.01) **Device lock** (S.02.08)	The definitions listed in Appendix D, for the identified standards, focus on the limitation of access to system information and	**Secure access control**	No unauthorised access must be given to an IM application. The application must secure itself from all forms of

Wireless access (S.02.11)	resources. The definitions mention that only authorised individuals, connections, and functions should receive access to system information and resources.		attempted access from unauthorised entities.

Table 6.3 consists of 18 security and usability heuristics, guidelines, standards, and best practices which were deemed similar. These similarities were mentioned and utilised when combining the security and usability heuristics, guidelines, standards, and best practices into one heuristic. The security and usability heuristics, guidelines, standards, and best practices which were combined are listed, alongside the new heuristic and its definition based on the IM context. The 18 security and usability heuristics, guidelines, standards, and best practices were reduced to six heuristics after the combining process. From Table 6.2, six security and usability guidelines, standards, and best practices still require adaption to the IM context, namely:

- User suitability (G.07.09)
- Secure server-side application (G.08.02)
- Secure operation of the app (G.08.05)
- Up-to-date apps (G.08.19)
- Controls against malicious code (S.01.06)
- Configured software to have secure settings by default (B.04.16)

These six security and usability guidelines, standards, and best practices were converted to usable security heuristics, which were adapted to the IM context, in Table 6.4.

Table 6.4 Adaption of Remaining Guidelines, Standards, and Best Practices to IM Context

Name and Code	New Heuristic Name	New Definition
User suitability (G.07.09)	**Flexibility of user security expertise**	The security features of IM applications need to provide plaintext options suitable for users with diverse levels of skills and experience in security.
Secure server-side application (G.08.02)	**Secure server-side application**	The server side of IM applications need to be secured to the current industry standard level. Without a secure server, IM applications will be vulnerable and not suitable for storing and transmitting confidential information.
Secure operation of the app (G.08.05)	**Secure application operation**	IM applications need to be secured during their operation. No malicious entity should be able to interfere or manipulate an IM application or its operations.
Up-to-date apps (G.08.19)	**Notification of security updates**	To ensure optimal security, IM applications need to alert the user about application updates. To mitigate vulnerabilities of older applications, IM applications need to remain updated.

Controls against malicious code (S.01.06)	**Secure malware controls**	IM applications need to implement controls to detect, prevent and recover from malware. Such applications should also inform and keep users aware of the situation.
Configured software to have secure settings by default (B.04.16)	**Secure by default**	IM applications need to ensure that the optimal security settings are active by default. This will reduce the chances of IM applications being utilised with weaker security.

Table 6.4 consists of the six security and usability guidelines, standards, and best practices which required adaption from Table 6.2. These six security and usability guidelines, standards, and best practices were not combined and adapted with the other security and usability heuristics, guidelines, standards, and best practices in Table 6.3. These six security and usability guidelines, standards, and best practices were adapted into usable security heuristics relevant to the IM context.

The usable security heuristics found in Table 6.3 and 6.5 are focused on the securing of an IM application. These usable security heuristics are more security oriented than usability oriented. To ensure that a balanced set of usable security heuristics is attained, more usability-oriented usable security heuristics were added.

To attain a balanced set of usable security heuristics, Nielsen's 10 Usability Heuristics for User Interface Design (Nielsen, 1995), located under H.04 in Appendix A, was adapted to the IM context by removing and adapting the heuristics in the set, based on the requirements of the usable security and the IM context.

The 10 usability heuristics for interface design developed by Nielsen in 1995 are general usability heuristics and not domain specific and were adapted to the IM context to ensure relevance to this study. Two of Nielsen's 10 heuristics were already included in Table 6.3, specifically error prevention (H.04.05) and help users recognise, diagnose, and recover from errors (H.04.09). These two heuristics were excluded from the adaption process for Nielsen's heuristics, as they were previously adapted.

Table 6.5 Adaption of Nielsen's Usability Heuristics

Old heuristic name and code	New heuristic name	New definition
Visibility of system status (H.04.01)	**Visibility of security status**	IM applications should always keep users informed about the security status of the application, through appropriate feedback within reasonable time.
Match between system and the real world (H.04.02)	**Match between security features and the real world**	IM applications security features should speak the users' language, using terms, phrases, and security ideas that they are acquainted with. Follow real-world standards to present data in a logical and natural arrangement.
User control and freedom (H.04.03)	**User security control and freedom**	Users frequently choose IM application security functions by accident, necessitating the presence of a clearly indicated 'emergency escape' that allows them to quit the undesirable state without having to go through a lengthy dialogue. Undo and redo are recommended.

Consistency and standards (H.04.04)	**Security consistency and standards**	When using an IM application's security features, users should not have to question whether various phrases, circumstances, or actions imply the same thing. Observe the security protocols established by IM and other applications.
Recognition rather than recall (H.04.06)	**Security recognition rather than recall**	Make security objects, actions, and choices accessible to reduce an IM application user's memory burden. The user should not be required to recall information from one section of the security interaction to the next. When applicable, instructions for using the security features should be visible or easily accessible.
Flexibility and efficiency of use (H.04.07)	**Flexibility and efficiency of use for security features**	Unseen by the inexperienced user, accelerators may commonly speed up the interaction for the expert user, allowing the security features to accommodate both inexperienced and experienced users. Allow users to customise security-related features that they perform on a regular basis.
Aesthetic and minimalist design (H.04.08)	**Aesthetic and minimalist security design**	Information that is useless or is used seldom should not be included in security dialogues. In a security dialogue, every additional unit of information conflicts with the essential pieces of information, lowering their relative visibility.
Help and documentation (H.04.10)	**Security help and documentation**	Even though it is preferable for the security features to be operated without documentation, assistance and documentation may be required. Any such security information should be simple to find, concentrate on the user's security duty, have a list of clear procedures to follow, and be manageable in size.

Table 6.5 presents the adapted version of Nielsen's usability heuristics for interface design. These heuristics could potentially succeed in assisting developers in developing usable and secure IM applications. However, this set of usable heuristics is more usability oriented than security oriented and must be combined with the other, previously adapted, usable security heuristics.

The six usable security heuristics from Table 6.3 and the 6 from Table 6.4, will be combined with the adapted version of Nielsen's 10 Usability Heuristics for User Interface Design, from Table 6.5, to present a set of 20 heuristics for the first draft of the preliminary set of usable security heuristics for instant messaging application development.

6.4 Stage 2: Analyse

The following section documents Step 3, Stage 2, of the 3-Stage rigorous process.

The first draft of the preliminary set of usable security heuristics for instant messaging application development is displayed in Table 6.6. The heuristics have been coded, utilising a coding scheme similar to the scheme utilised in Appendices A to D. The D1 stands for first draft, US stands for usable security, H stands for heuristic, and the number represents the heuristics number in the set. For example: D1.US.H.01 would be the code for the first heuristic from the first draft in the set of usable security heuristics.

Table 6.6 Preliminary Set of Usable Security Heuristics for Instant Messaging Application Development – Draft 1 (D1)

Heuristic code	Heuristic name	Definition
D1.US.H.01	**Visibility of security status**	IM applications should always keep users informed about the security status of the application through appropriate feedback within reasonable time.
D1.US.H.02	**Match between security features and the real world**	An IM application's security features should speak the users' language, using terms, phrases, and security ideas that they are acquainted with. They should follow real-world standards to present data in a logical and natural arrangement.
D1.US.H.03	**User security control and freedom**	Users frequently choose IM application security functions by accident, necessitating the presence of a clearly indicated 'emergency escape' that allows them to quit the undesirable state without having to go through a lengthy dialogue. Undo and redo are recommended.
D1.US.H.04	**Security consistency and standards**	When using an IM application's security features, users should not have to question whether various phrases, circumstances, or actions imply the same thing. They should observe the security protocols established by IM and other applications.
D1.US.H.05	**Security recognition rather than recall**	Make security objects, actions, and choices accessible to reduce IM application user's memory burden. The user should not be required to recall information from one section of the security interaction to the next. When applicable, instructions for using the security features should be visible or easily accessible.
D1.US.H.06	**Flexibility and efficiency of use for security features**	Unseen by the inexperienced user, accelerators may commonly speed up the interaction for the expert user, allowing the security features to accommodate both inexperienced and experienced users. Users should be allowed to customise security-related features that they perform on a regular basis.
D1.US.H.07	**Aesthetic and minimalist security design**	Information that is useless or is used seldom should not be included in security dialogues. In a security dialogue, every additional unit of information conflicts with the essential pieces of information, lowering their relative visibility.
D1.US.H.08	**Threat prevention and user guidance**	IM applications should present security messages in a plaintext format to the user. IM applications should guide the user during usage by hiding unavailable functions, warning users about their actions, and assisting users to recognise, diagnose, and avoid potential threats.
D1.US.H.09	**Security help and documentation**	Even though it is preferable for the security features to be operated without documentation, assistance and documentation may be required. Any such security information should be simple to find, should concentrate on the user's security duty, should have a list of clear procedures to follow, and should be manageable in size.

D1.US.H.10	**Compliance of security and privacy controls**	IM applications must provide the current industry standard of security and privacy controls with basic plaintext instructions for users on how to implement and utilise these features effectively. The privacy features need to align with international standards, such as the South African Protection of Personal Information Act (POPIA) and the European General Data Protection Regulation (GDPR).
D1.US.H.11	**Securing from third-party**	IM applications, its code, the information stored in it, and the information located on the device must be secured from third-party sources. These third-party sources include but are not limited to third-party applications, external code libraries, and third-party app stores.
D1.US.H.12	**Encryption of application session and information**	IM applications need to be encrypted to the current industry level of encryption. The encryption level must be made clear to the user. If there is more than one level of encryption available, it must be clear which is active, and the user must be guided in how to select the relevant encryption feature. It is crucial for an IM application to encrypt the application session and the storage and transmission of information.
D1.US.H.13	**Least privilege by default**	IM applications need to be developed with the principle of least privilege, which is to ensure that the permissions requested by the application is limited to the minimum permissions required for functionality. IM applications must not request more permissions than those required. Each permission requested must be clearly and concisely explained to the user, to ensure that an informed decision is made by the user. This will also reduce the cognitive load on the user.
D1.US.H.14	**Secure access control**	No unauthorised access must be given to an IM application. The application must secure itself from all forms of attempted access from unauthorised entities.
D1.US.H.15	**Flexibility of user security expertise**	The security features of IM applications need to provide plaintext options suitable for users with diverse levels of skills and experience in security.
D1.US.H.16	**Secure server-side application**	The server-side of an IM application needs to be secured to the current industry standard level. Without a secure server, IM applications will be vulnerable and not suitable for storing and transmitting confidential information.
D1.US.H.17	**Secure application operation**	IM applications need to be secured during their operation. No malicious entity should be able to interfere or manipulate an IM application or its operations.
D1.US.H.18	**Notification of security updates**	To ensure optimal security, IM applications need to alert the user about application updates. To mitigate vulnerabilities of older applications, IM applications need to remain updated.
D1.US.H.19	**Secure malware controls**	IM applications need to implement controls to detect, prevent and recover from malware. Such applications should also inform and keep users aware of the situation.

D1.US.H.20	**Secure by default**	IM applications need to ensure that the optimal security settings are active, by default. This will reduce the chances of IM applications being utilised with weaker security.

To ensure that the set of usable security heuristics meets the requirements of usable security, each heuristic was examined to identify whether the heuristic aligns with both usability and security. The examination conducted is presented in Table 6.7 as a matrix.

Table 6.7 Usability and Security Matrix

Heuristic Code	Heuristic name	Usability	Security
D1.US.H.01	**Visibility of security status**	✓	✓
D1.US.H.02	**Match between security features and the real world**	✓	
D1.US.H.03	**User security control and freedom**	✓	✓
D1.US.H.04	**Security consistency and standards**	✓	
D1.US.H.05	**Security recognition rather than recall**	✓	✓
D1.US.H.06	**Flexibility and efficiency of use for security features**	✓	✓
D1.US.H.07	**Aesthetic and minimalist security design**	✓	
D1.US.H.08	**Threat prevention and user guidance**	✓	✓
D1.US.H.09	**Security help and documentation**	✓	✓
D1.US.H.10	**Compliance of security and privacy controls**	✓	✓
D1.US.H.11	**Securing from third-party**		✓
D1.US.H.12	**Encryption of application session and information**	✓	✓
D1.US.H.13	**Least privilege by default**	✓	✓
D1.US.H.14	**Secure access control**	✓	✓
D1.US.H.15	**Flexibility of user security expertise**	✓	✓
D1.US.H.16	**Secure server-side application**		✓
D1.US.H.17	**Secure application operation**		✓
D1.US.H.18	**Notification of security updates**	✓	✓
D1.US.H.19	**Secure malware controls**	✓	✓
D1.US.H.20	**Secure by default**	✓	✓

Not all the heuristics in the first draft meet the requirements for usable security. From the 20 usable security heuristics presented, 14 align with both the usability and security aspects required to be defined as usable security.

The following heuristics are not usable from the user's perspective and could be seen as security heuristics and not as usable security. These three heuristics have been identified, namely:

- D1.US.H.11 – Securing from third-party
- D1.US.H.16 – Secure server-side application
- D1.US.H.17 – Secure application operation

These three preliminary heuristics do not have a usability aspect to them, as the results of their implementation will not be seen or felt by the users of the IM application. They do provide a security contribution but without the usability aspect of these heuristics, they are not usable security heuristics.

The following heuristics do not directly relate to the improvement of the IM applications security. This leads the heuristics to be seen as usability heuristics as they do not meet the security requirements of usable security. These three heuristics have been identified, namely:

- D1.US.H.02 – Match between security feature and the real world
- D1.US.H.04 – Security consistency and standards
- D1.US.H.07 – Aesthetic and minimalist security design

These three preliminary heuristics do not have a security aspect to them, as the results of their implementation will not improve the IM applications security. They do provide a usability contribution, as the results of their implementation will be seen or felt by the users of the IM application. However, without the security aspect of these heuristics, they are not usable security heuristics.

The heuristics, which were identified as not aligning with the established requirements of usable security, were further analysed. Three of the definitions of the six heuristics were further adapted as a result of the analysis. Table 6.8 contains the three heuristics which were further adapted.

Table 6.8 Further Adaption of Usable Security Heuristics

Heuristic Code	**Heuristic Name**	**Definition**
D1.US.H.02	**Match between security features and the real world**	An IM application's security features should speak the users' language, using real-world standards for terms, phrases, and security ideas with which they are acquainted. This guarantees that the user is well-informed and aware of the influence of the security features on the IM application.
D1.US.H.04	**Security consistency and standards**	When using IM applications security features, users should not have to question whether various security phrases, circumstances, or actions imply the same thing. An IM application’s security features should be aligned with other IM applications to ensure that users maintain an understanding of the security features. This ensures that users maintain an understanding of the security features within the IM application environment.
D1.US.H.07	**Aesthetic and minimalist security design**	Information that is useless or is seldom used should not be included in security dialogues. In a security dialogue, every additional unit of information, which conflicts with the essential pieces of information, lowers their relative visibility. Ensuring that the security dialogue utilised remains concise and specific to the topic at hand, will improve the user’s decision-making with regard to the impact of the related IM application security features.

To ensure that the preliminary proposed heuristics align with the requirements of usable security, the three heuristics listed in Table 6.8 were further adapted to meet the established requirements. These adaptions allow the three heuristics to maintain their position in the proposed preliminary set of usable security heuristics for instant messaging application development.

The remaining three identified preliminary heuristics, which do not align with the requirements of usable security, will be removed from the preliminary set of usable security heuristics. The removal of the heuristics is to ensure an accurate and concise second draft of the preliminary set of usable security heuristics. The following heuristics have been removed, namely:

- **D1.US.H.11 – Securing from a third party:** Users will not interact with the securing of the application from a third party. Without user interaction, there will not be a usability aspect to this. As a result, the heuristic does not comply with usable security criteria.
- **D1.US.H.16 – Secure server-side application:** When utilising an IM application, users do not interact with or operate the servers of the IM application. There is no user interaction with the servers. As a result, the heuristic does not comply with usable security criteria.
- **D1.US.H.17 – Secure application operation:** The responsibility of securing an IM application during operation does not lie with the users of that IM application. The developers are responsible for securing the IM application. Users do not have any role or interaction with the securing of the IM application. As a result, the heuristic does not comply with usable security criteria.

The removal of these three heuristics reduces the number of usable security heuristics, in the preliminary set, from 20 to 17.

6.5 Stage 3: Revise and Finalise

The second draft of the preliminary set of usable security heuristics for instant messaging application development is shown in Table 6.9. The coding scheme was utilised and reimplemented for the second draft of the preliminary heuristics. In the coding scheme D1 was replaced with D2, as this is the second draft.

Table 6.9 Preliminary Set of Usable Security Heuristics for Instant Messaging Application Development – Draft (D2)

Heuristic code	Heuristic name	Definition
D2.US.H.01	**Visibility of security status**	IM applications should always keep users informed about the security status of the application through appropriate feedback within reasonable time.
D2.US.H.02	**Match between security features and the real world**	An IM application's security features should speak the users' language, using real-world standards for terms, phrases, and security ideas with which they are acquainted. This guarantees that the user is well-informed and aware of the influence of the security features on the IM application.
D2.US.H.03	**User security control and freedom**	Users frequently choose IM application security functions by accident, necessitating the presence of a clearly indicated ‘emergency escape’ that allows them to quit the undesirable state without having to go through a lengthy dialogue. Undo and redo are recommended.
D2.US.H.04	**Security consistency and standards**	When using IM applications security features, users should not have to question whether various security phrases, circumstances, or actions imply the same thing. An IM application’s security features should be aligned with other

		IM applications to ensure that users maintain an understanding of the security features. This ensures that users maintain an understanding of the security features within the IM application environment.
D2.US.H.05	**Security recognition rather than recall**	Make security objects, actions, and choices accessible to reduce IM application user's memory burden. The user should not be required to recall information from one section of the security interaction to the next. When applicable, instructions for using the security features should be visible or easily accessible.
D2.US.H.06	**Flexibility and efficiency of use for security features**	Unseen by the inexperienced user, accelerators may commonly speed up the interaction for the expert user, allowing the security features to accommodate both inexperienced and experienced users. Users should be allowed to customise security-related features that they perform on a regular basis.
D2.US.H.07	**Aesthetic and minimalist security design**	Information that is useless or is seldom used should not be included in security dialogues. In a security dialogue, every additional unit of information, which conflicts with the essential pieces of information, lowers their relative visibility. Ensuring that the security dialogue utilised remains concise and specific to the topic at hand, will improve the user's decision-making with regard to the impact of the related IM application security features.
D2.US.H.08	**Threat prevention and user guidance**	IM applications should present security messages in a plaintext format to the user. IM applications should guide the user during usage by hiding unavailable functions, warning users about their actions, and assisting users to recognise, diagnose, and avoid potential threats.
D2.US.H.09	**Security help and documentation**	Even though it is preferable for the security features to be operated without documentation, assistance and documentation may be required. Any such security information should be simple to find, should concentrate on the user's security duty, should have a list of clear procedures to follow, and should be manageable in size.
D2.US.H.10	**Compliance of security and privacy controls**	IM applications must provide the current industry standard of security and privacy controls with basic plaintext instructions for users on how to implement and utilise these features effectively. The privacy features need to align with international standards, such as the South African Protection of Personal Information Act (POPIA) and the European General Data Protection Regulation (GDPR).
D2.US.H.11	**Encryption of application session and information**	IM applications need to be encrypted to the current industry level of encryption. The encryption level must be made clear to the user. If there is more than one level of encryption available, it must be clear which is active, and the user must be guided in how to select the relevant encryption feature. It is crucial for an IM application to encrypt the application session and the storage and transmission of information.

D2.US.H.12	**Least privilege by default**	IM applications need to be developed with the principle of least privilege, which is to ensure that the permissions requested by the application is limited to the minimum permissions required for functionality. IM applications must not request more permissions than those required. Each permission requested must be clearly and concisely explained to the user, to ensure that an informed decision is made by the user. This will also reduce the cognitive load on the user.
D2.US.H.13	**Secure access control**	No unauthorised access must be given to an IM application. The application must secure itself from all forms of attempted access from unauthorised entities.
D2.US.H.14	**Flexibility of user security expertise**	The security features of IM applications need to provide plaintext options suitable for users with diverse levels of skills and experience in security.
D2.US.H.15	**Notification of security updates**	To ensure optimal security, IM applications need to alert the user about application updates. To mitigate vulnerabilities of older applications, IM applications need to remain updated.
D2.US.H.16	**Secure malware controls**	IM applications need to implement controls to detect, prevent and recover from malware. Such applications should also inform and keep users aware of the situation.
D2.US.H.17	**Secure by default**	IM applications need to ensure that the optimal security settings are active, by default. This will reduce the chances of IM applications being utilised with weaker security.

The 17 preliminary heuristics displayed in Table 6.9 meet the requirements for usable security, established by the definition of usable security stated in Chapter 1, Section 1.1. When implemented correctly, these preliminary heuristics could potentially assist IM application developers in developing a usable and secure IM application for users of all computer literacy and security levels.

6.6 Mapping of Preliminary Set of Usable Security Heuristics

The following section is split into two subsections. The first subsection focuses on the preliminary set of usable security heuristics mapped against identified instant messaging threats, while the second focuses on the set of usable security heuristics mapped against the identified instant messaging security and privacy features.

6.6.1 Mapping Against Identified Instant Messaging Threats

In the creation of the preliminary set of usable security heuristics, the five threats to IM were utilised. To ensure that the current second draft of the preliminary set of usable security heuristics address the risk and exposure associated with the five most prominent IM threats sufficiently, the preliminary set of usable security heuristics were mapped against these threats. The mapping of the set of usable security heuristics is compiled in Table 6.10.

Table 6.10 Preliminary Set of Usable Security Heuristics Mapped Against the Instant Messaging Threat Matrix

Heuristic code	Confidential Information Leakage	Distribution of Malicious Code	Man-in-The-Middle Attack	Permission System Vulnerability	Social Engineering	Total
D2.US.H.01						**0**
D2.US.H.02						**0**
D2.US.H.03	✓	✓			✓	**3**
D2.US.H.04						**0**
D2.US.H.05						**0**
D2.US.H.06						**0**
D2.US.H.07						**0**
D2.US.H.08	✓	✓	✓	✓	✓	**5**
D2.US.H.09						**0**
D2.US.H.10	✓	✓	✓	✓	✓	**5**
D2.US.H.11	✓		✓			**2**
D2.US.H.12	✓	✓	✓	✓		**4**
D2.US.H.13	✓		✓			**2**
D2.US.H.14	✓				✓	**2**
D2.US.H.15	✓	✓	✓	✓		**4**
D2.US.H.16	✓	✓	✓			**3**
D2.US.H.17	✓	✓	✓	✓	✓	**5**
Total	**10**	**7**	**8**	**5**	**5**	**35**

The preliminary set of usable security heuristics were mapped against the five most prominent IM threats identified in Chapter 3, Section 3.5. From the preliminary set of usable security heuristics, 10 of the 17 heuristics are deemed relevant in potentially mitigating the impact and exposure of the five most prominent IM threats.

US.H.01 to US.H.09 focus more on the usability and user experience of the security functions within the IM application, while US.H.10 to US.H.17 focus on the security of the IM application. This balance of usability and security assists in ensuring the overall validity of the preliminary set of usable security heuristics.

However, even though it can be noted that seven of the 17 usable security heuristics were not directly linked to the five most prominent IM security threats, this does not mean that these six usable security heuristics are not relevant to the overall security of the IM application. To further indicate the relevance of the 17 usable security heuristics, each heuristic and its relevance, based on its definition and potential implementation, is discussed. The potential impact on the five most prominent IM threats and the overall role of the heuristic with regard to security is considered in the discussion.

D2.US.H.01 – Focuses on keeping the user informed of the security status. If something goes wrong or security is breached the users will be alerted to this situation. This heuristic does not address an individual threat specifically but rather the overall security status, which makes it potentially relevant to all threats, if detected.

D2.US.H.02 – This heuristic ensures that the security feature is usable by utilising language familiar to the user and following real-world conventions. This heuristic does not address security issues, but it does ensure that the security functions implemented are usable by the user. Ensuring that users can utilise the security functions available to them is just as important as the security feature itself.

D2.US.H.03 – Ensuring that the users have an undo and redo dialogue improves the users' perception of that functionality. This heuristic ensures that users are able to correct their actions when utilising the security features available to them. A user might mistakenly deactivate a security feature and instead of going through a long dialogue to correct this, the user could use the undo function. This heuristic has been linked to three of the five most prominent IM security threats, namely: confidential information leakage, distribution of malicious code, and social engineering.

D2.US.H.04 – This heuristic focuses on maintaining consistency and a linear standard for all security features on the IM application. This heuristic does not address security issues but ensures the consistency of all the security features across the IM application environment. This would improve the usability of the security features in the eyes of the user.

D2.US.H.05 – The focus of this heuristic is to ease the cognitive load on the user when utilising the security features. Users should not need to remember information from one step to the next. The security feature should provide all of the necessary information when needed. This heuristic does not address security issues but ensures that users will be provided with all the necessary information when interacting with the security features. This ensures that users can make well-informed decisions when interaction with security features.

D2.US.H.06 – Increasing the flexibility and efficacy of the IM application security features is the focus of this heuristic. Allowing users to customise their usage of the IM security features potentially leads to a positive user experience. This customisation would be done with accelerators, to speed up the process when utilising security features. This does not address security directly; however, it does ensure that experienced users will not be frustrated by extended security dialogues and drawn-out processes. These frustrations could potentially lead to users avoiding security features and operating the IM application with suboptimal security.

D2.US.H.07 – This heuristic focuses on keeping the security dialogue concise. Extra, unnecessary information could potentially confuse a user while they interact with the security feature. Ensuring that the information provided is concise and relevant should lead to an improved user experience. The clarity provided to the user, by ensuring that the security dialogue is concise, will ensure that users understand the security communications.

D2.US.H.08 – The focus of this heuristic is the communication and guidance provided to the user when utilising the IM application. Preventing threats by guiding users and utilising coding practices could result in a positive user experience. Owing to the possibility of threat mitigation through the correct implementation of this heuristic, it has been linked to all five of the most prominent IM security threats.

D2.US.H.09 – It is critical to provide users with security-related help and documentation, which is the focus of this heuristic. Though this heuristic does not address security issues, it is still crucial for the overall function of the security features. Without providing the proper information or access to this information, users would not be able to make the appropriate informed decisions when interacting with the security features.

D2.US.H.10 – This heuristic is focused on implementing the current industry standard for both security and privacy controls. The successful implementation of this heuristic will address various security and privacy concerns. This heuristic has been linked with all five of the most prominent IM security threats.

D2.US.H.11 – Ensuring that the IM application is encrypted to the current industry standard level is the focus of this heuristic. Encrypting the IM application increases both security and privacy concerns. This heuristic has been linked to two of the five most prominent IM security threats, namely confidential information leakage and man-in-the-middle attack.

D2.US.H.12 – Limiting the access of an application to the user's device is the focus of this heuristic. Least privilege ensures that the application does not request or gain more access than is necessary for the functionality of the application. This heuristic has been linked to four of the five most prominent IM security threats, namely confidential information leakage, distribution of malicious code, man-in-the-middle attack and permission system vulnerability.

D2.US.H.13 – Securing the access of the IM application and preventing all unauthorised access is the highlight of this heuristic. Ensuring that no unauthorised individuals can access the IM application directly improves the security and privacy of the application. This heuristic has been linked to two of the five most prominent IM security threats, namely confidential information leakage and man-in-the-middle attack.

D2.US.H.14 – The focus of this heuristic is to ensure that the application caters for all users, regardless of the skills and experience in security. By catering for all users, the usability and security of the application will increase, as users will potentially find the security features to be usable and users will be less susceptible to mistakes. This heuristic has been linked to two of the five most prominent IM security threats, namely confidential information leakage and social engineering.

D2.US.H.15 – This heuristic ensures that users will keep their IM application up to date. By ensuring that the application is updated regularly, vulnerabilities located in the IM application will be addressed. This will ensure that the IM application security will remain at the optimal level. This heuristic has been linked to four of the five most prominent IM security threats, namely confidential information leakage, distribution of malicious code, man-in-the-middle attack, and permission system vulnerability.

D2.US.H.16 – Ensuring that the IM application is not vulnerable to malware is the focus of this heuristic. Malicious data is spread across IM applications regularly. To ensure that users are protected from this, sufficient security controls and controls need to be implemented. This heuristic has been linked to three of the five most prominent IM security threats, namely confidential information leakage, distribution of malicious code and man-in-the-middle attack.

D2.US.H.17 – This heuristic focuses on overall security by ensuring that the security features located in the IM application are set to their optimal levels by default. By ensuring the default optimum settings, the user has less reason to interact with the security settings. Users might not fully understand the security settings and configure their security at a suboptimal level, which could lead to vulnerabilities being introduced to the IM application. This heuristic has been linked with all five of the most prominent IM security threats.

Ensuring that users can utilise the security functions available to them is just as important as the function of the security itself. Without the usability of the security features, users who are not familiar with security will have a difficult experience attempting to utilise the security features available to them. This difficult experience will result in a negative user experience and frustration, which could potentially lead a user to uninstall the application or to operate the application with suboptimal security. The discussion of each heuristic reinstates its relevance and importance for inclusion in the second draft of the preliminary set of usable security heuristics for instant messaging application development.

6.6.2 Mapping Against Identified Instant Messaging Security and Privacy Controls

In Chapter 4, Section 4.5, the current IM security and privacy controls were identified. To ensure that the propose preliminary set of usable security heuristics are aligned with the requirements of the identified IM security and privacy controls, Table 6.11 contains a mapping of the IM security and privacy controls against the proposed set of usable security heuristics.

Table 6.11 Preliminary Set of Usable Security Heuristics Mapped Against the Instant Messaging Security and Privacy Controls Matrix

Heuristic code	**Encryption**	**Deleting messages**	**Self-destructing messages**	**Two-factor authentication**	**Verification SMS/email**	**Password lock**	**Screenshot detection**	**Remote log out**	**Account self-destruct**	**Total**
D2.US.H.01		✓					✓			**2**
D2.US.H.02	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.03		✓	✓	✓	✓	✓	✓	✓	✓	**8**
D2.US.H.04	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.05	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.06	✓	✓	✓					✓	✓	**5**
D2.US.H.07	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.08		✓	✓	✓	✓	✓	✓	✓	✓	**8**
D2.US.H.09	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.10	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.11	✓									**1**
D2.US.H.12										**0**
D2.US.H.13	✓			✓	✓	✓	✓	✓	✓	**7**
D2.US.H.14	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.15	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
D2.US.H.16	✓									**1**
D2.US.H.17	✓	✓	✓	✓	✓	✓	✓	✓	✓	**9**
Total	**13**	**13**	**12**	**12**	**12**	**12**	**13**	**13**	**13**	**113**

The preliminary set of usable security heuristics were mapped against the identified security and privacy IM controls identified in Chapter 4, Section 4.5. From the preliminary set of usable security heuristics, only one heuristic did not directly relate to any of the IM security and privacy controls, D2.US.H.12. Of the heuristics that did relate to the IM security and privacy controls two only mapped once, namely:

- **D2.US.H.11** – Encryption of information

- **D2.US.H.16** – Secure malware controls

Of the heuristics that did relate to the IM security and privacy controls, nine mapped to all of those identified, namely:

- **D2.US.H.02** – Match between security features and the real world
- **D2.US.H.04** – Security consistency and standards
- **D2.US.H.05** – Security recognition rather than recall
- **D2.US.H.07** – Aesthetic and minimalist security design
- **D2.US.H.09** – Security help and documentation
- **D2.US.H.10** – Compliance between security and privacy controls
- **D2.US.H.14** – Flexibility of user security expertise
- **D2.US.H.15** – Notification of security updates
- **D2.US.H.17** – Secure by default

Each of the heuristics' relation to the IM security and privacy controls is discussed below.

D2.US.H.01 – Alerting and keeping users informed about the security status of the IM application only related directly to screenshot detection. Screenshot detection alerts users when an individual takes a screenshot of their conversation. Although this heuristic only directly related to one IM security and privacy control, it still has a large contribution to the application overall security.

D2.US.H.02 – Matching the language of the security feature to the language used in the real world related to all the IM security and privacy controls. This heuristic assists in ensuring that the IM security and privacy controls will be understood by users.

D2.US.H.03 – Utilising undo and redo functions within the IM security and privacy features related to eight of the nine IM security and privacy controls. This heuristic did not relate directly to the implementation of encryption. This heuristic will assist in providing clear navigation to users and will prevent unnecessary dialogue.

D2.US.H.04 – Ensuring consistency across the IM application environment related to all the IM security and privacy controls. This heuristic assists in ensuring that the IM security and privacy features observe the protocols established by IM applications and other applications.

D2.US.H.05 – Supplying users with easy to access instructions and information related to all the IM security and privacy controls. This heuristic assists in ensuring that the IM security and privacy features will be utilised appropriately.

D2.US.H.06 – Improving the speed and efficiency with which IM security and privacy features may be used might potentially increase the number of users who utilise them. Ensuring the usability of features assists in potentially increasing their usage and securing the users of the IM application.

D2.US.H.07 – Maintaining clear and concise security dialogue related to all the IM security and privacy controls. This heuristic assists in ensuring that the dialogue and communications of IM security and privacy features are simply and easily understood by users.

D2.US.H.08 – Threat prevention by guiding users' actions withing the IM application related to eight of the nine IM security and privacy controls. This heuristic ensures that users will receive guidance and assistance to recognise, diagnose, and avoid potential threats.

D2.US.H.09 – Providing help and documentation when utilising security features related to all the IM security and privacy controls. This heuristic assists in ensuring that the IM security and privacy features will have the appropriate help and documentation available to the users, which will result in the proper utilisation of the IM security and privacy features.

D2.US.H.10 – The implementation of the current industry standard of security and privacy controls related to all the IM security and privacy controls. This heuristic assists in ensuring that the appropriate IM security and privacy controls are available within the IM application.

D2.US.H.11 – The encryption of information links to one IM security and privacy control, encryption. Ensuring that the appropriate industry level of encryption is implemented in the IM application is essential to the securing of the IM application.

D2.US.H.12 – The implementation of least privilege related to no IM security and privacy controls. Even though it had no relation to the IM security and privacy controls, it is an important heuristic for the overall security of the IM application.

D2.US.H.13 – Ensuring that only authorised access is provided to the IM application related to seven of the nine IM security and privacy controls. Preventing unauthorised access is the focus of this heuristic, which is a crucial component of the overall IM application security.

D2.US.H.14 – Providing plaintext options suitable for all users related to all the IM security and privacy controls. This heuristic assists in ensuring that the IM security and privacy features are usable for all users, regardless of their levels of skills and experience in security.

D2.US.H.15 – IM application updates related to all the IM security and privacy controls. This heuristic assists in ensuring that the IM security and privacy features remain up to date. Maintaining an up-to-date IM application ensures that the IM application can operate with optimal security.

D2.US.H.16 – Securing the IM application from malware related to one of the nine IM security and privacy controls, encryption This informs and ensures user awareness of malware situations assisting in improving the overall IM application security.

D2.US.H.17 – Operating with optimal security setting, by default, related to all the IM security and privacy controls. This heuristic assists in ensuring that all IM security and privacy features are implemented to provide the user with a fully secure IM application by default.

The majority of the proposed preliminary set of usable security heuristics related to the identified IM security and privacy controls. This documents how the proposed preliminary set of usable security heuristics could be utilised potentially during the development of IM applications, to assist IM application developers in implementing IM security and privacy features. The discussion on each heuristic highlights their importance to the overall set of proposed preliminary set of usable security heuristics for instant messaging application development.

6.7 Conclusion

By following the four-step process, adapted from Quiñones & Rusu (2017) to develop usable security heuristics, this study was able to present a preliminary set of usable security heuristics for instant messaging application development. Steps 1, 2, and 3 of the four-step process, have been completed. Step 1 was completed in Chapters 3 and 4; Step 2 in Chapter 5; and Step 3 in Chapter 6. The presentation of the second draft of preliminary usable security heuristics for instant messaging application development, satisfies the requirements of SRO3. Step 4, which focuses on the validation of the proposed set of preliminary usable security heuristics, will be completed in Chapter 7.

The second draft of the preliminary set of usable security heuristics, located in Table 6.9, will be taken forward into Chapter 7 for validation. The second draft of the preliminary set of usable security heuristics will undergo validation through an expert review. Upon completion of the expert review, the recommended changes presented by the experts will be made, which will result in the final set of usable security heuristics for instant messaging application development. This final set of usable security heuristics will be further validated with the creation of a proof-of-concept prototype.

Chapter 7 – Validation of the Proposed Set of Usable Security Heuristics

7.1 Introduction

In Chapter 6, the proposed preliminary set of usable security heuristics for instant messaging application development was presented.

The aim of this chapter is to further meet the requirements of the Primary Research Objective, *to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*, of this study by validating the proposed set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications. This was accomplished through an expert review, as discussed in this chapter.

The chapter structure is as follows: Section 7.2 discusses the overview of the expert review, while Section 7.3 briefly discusses the expert review instrument design. Section 7.4 discusses the feedback from the expert review, while Section 7.5 discusses any potential changes to the set of usable security heuristics based on the expert review feedback. Section 7.6 concludes the chapter.

7.2 Overview of Expert Review

To satisfy the requirements of the PRO for this study fully, an expert review was conducted. The definition of an expert review and the aim of this study's expert review were documented in Chapter 1, Section 1.6.4. An expert review, as defined by Kovesdi and Joe (2017) is *'the evaluation of a system, by a subject matter expert, against a standardised set of evaluation criteria' (pp. 1262).* In Chapter 1, Section 1.6.4, the aim of the expert review conducted during this study was stated as, *to validate the utility, quality, and efficacy of the proposed set of usable security* heuristics. In terms of this study, these three important characteristics are simply defined, as follows:

- **Efficacy** – The ability to produce a desired or intended result.
- **Utility** – The state of being useful, or beneficial.
- **Quality** – The standard as measured against other heuristics.

Six experts were identified as potential participants for this study's expert review based on specific selection criteria. These identified participants were deemed experts owing to their experience in either security, usability or mobile application development. Ethical approval to conduct the expert review was granted by Nelson Mandela University Research Ethics Committee: Human, with the ethical approval reference number of H21-ENG-ITe-006.

Having obtained ethical approval, the identified experts were contacted via email to enquire about their availability and willingness to participate in the expert review. The experts, who were willing to participate, completed an informed consent form, as shown in Appendix F. Each of the participating experts was provided with a copy of the guideline document, as presented in Appendix E. The guideline document provided experts with the following information:

- The **purpose of the study** – which highlights the relevant background information and context for the study.
- The **participant selection criteria** – provided insight into the criteria used to identify relevant experts for this study.

- The **role of the expert** – explained their role in the study and how the feedback they provided would be utilised.

- The **proposed set of preliminary usable security heuristics** – provided background information regarding the development of the proposed set of usable security heuristics together with the proposed set of usable security heuristics.

Within the guideline document, experts were provided with a questionnaire, presented in Appendix G. The experts were guided in terms of how to answer the questionnaire, thus providing valuable feedback on the proposed set of usable security heuristics. The following section briefly describes the design of the expert review questionnaire.

7.3 Expert Review Instrument Design

In order to collect and analyse relevant feedback from the participants of the expert review, an instrument was designed in the form of a questionnaire. Since the specific focus of the proposed heuristics involved three main fields of expertise, namely usability, security, and mobile application development, these were addressed in separate sections. The two other sections included biographical information and general feedback regarding the proposed set of usable heuristics. The five sections of the expert review questionnaire, as presented in Appendix G, are summarised as follows:

- **Biographical information** – including the experience and confidence level of the experts regarding heuristics in general.

- **Security section** – addressed the extent to which each of the set of proposed usable security heuristics satisfies the security threats and concerns relating to IM applications.

- **Usability section** – addressed the extent to which each of the set of proposed usable security heuristics satisfies the usability concerns relating to the security of IM applications.

- **Mobile application development section** – addressed the extent to which each of the set of proposed usable security heuristics satisfies the IM application development concerns relating to the security of IM applications.

- **General section** – focused on the overall impression of the proposed set of usable security heuristics, with regard to their efficacy, utility, and quality, and provided an opportunity for any final comments.

Many of the questions within the questionnaire used a Likert scale of 1 to 5, where 1= very low and 5 = very high. This provided for ease of analysis of the feedback regarding each of the proposed usable security heuristics.
The results and findings from the expert review are discussed in the following section.

7.4 Expert Review Results

From the eight potential experts, six responded and agreed to participate in the study. However, only five participated by actually providing feedback.

The reporting of the expert review results is presented in subsections 7.4.1 to 7.4.5. The reporting follows the same layout of the subsections indicated in the questionnaire presented in Appendix G.

For each of the tables in subsections 7.4.2 to 7.4.5, the average of the scores presented by the experts, based on the 5-point Likert scale utilised, are presented.

The usable security heuristics which received an average score of 3.0 or more were deemed acceptable and do not need to be refined further since they had accomplished their intended purpose. The usable security heuristics which received an average score of less than 3.0 were examined for possible refinement.

The usable security heuristics remained labelled with D2, to signify the utilisation of the second draft of usable security heuristics, from Chapter 6, Section 6.5, Table 6.9.

7.4.1 Biographical Information

The experts were asked to select which of the three stipulated relevant fields, security, usability, and mobile application development, describes their field of expertise best. The results are depicted in Table 7.1.

Table 7.1 Fields of Expertise of Expert Participants

Field of Expertise	Number of Experts
Security	3
Usability	1
Mobile application development	1
Total	**5**

The expertise of the five participants covered all three identified fields, confirming their adherence to the selection criteria stipulated. From Table 7.1 it is evident that most (three) of the experts indicated security as their field of expertise, with one expert in each of the other fields of interest. This spread of expertise ensures some diversity in the feedback received.

7.4.2 Security Section

For this section, the experts were requested to indicate the extent to which each of the proposed set of usable security heuristics satisfied the security threats and concerns related to IM applications.

Table 7.2 Expert Feedback Based on Satisfaction of Security Threats and Concerns

Heuristic code	Heuristic name	Security scale	Average usable security scale
D2.US.H.01	**Visibility of security status**	**Expert 1:** 2 = Low **Expert 2:** 3 = Moderate **Expert 3:** 3 = Moderate **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	3.2
D2.US.H.02	**Match between security features and the real world**	**Expert 1:** 3 = Moderate **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 4 = High **Expert 5:** 4 = High	4.75
D2.US.H.03	**User security control and freedom**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate	3.6

		Expert 3: 3 = Moderate **Expert 4:** 4 = High **Expert 5:** 4 = High	
D2.US.H.04	**Security consistency and standards**	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 4 = High **Expert 5:** 5 = Very High	4.6
D2.US.H.05	**Security recognition rather than recall**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 3 = Moderate **Expert 5:** 5 = Very High	4.2
D2.US.H.06	**Flexibility and efficiency of use for security features**	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 4 = High **Expert 5:** 3 = Moderate	3.4
D2.US.H.07	**Aesthetic and minimalist security design**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 4 = High	3.4
D2.US.H.08	**Threat prevention and user guidance**	**Expert 1:** 3 = Moderate **Expert 2:** 3 = Moderate **Expert 3:** 4 = High **Expert 4:** 4 = High **Expert 5:** 5 = Very High	3.8
D2.US.H.09	**Security help and documentation**	**Expert 1:** 2 = Low **Expert 2:** 3 = Moderate **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 4 = High	3.6
D2.US.H.10	**Compliance of security and privacy controls**	**Expert 1:** 5 = Very High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	4.2
D2.US.H.11	**Encryption of application session and information**	**Expert 1:** 5 = Very High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	4.6
D2.US.H.12	**Least privilege by default**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 4 = High **Expert 5:** 4 = High	4.2
D2.US.H.13	**Secure access control**	**Expert 1:** 4 = High **Expert 2:** 4 = High	4.0

		Expert 3: 3 = Moderate **Expert 4:** 5 = Very High **Expert 5:** 4 = High	
D2.US.H.14	**Flexibility of user security expertise**	**Expert 1:** 3 = Moderate **Expert 2:** 3 = Moderate **Expert 3:** 2 = Low **Expert 4:** 4 = High **Expert 5:** 3 = Moderate	3.0
D2.US.H.15	**Notification of security updates**	**Expert 1:** 2 = Low **Expert 2:** 3 = Moderate **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	3.8
D2.US.H.16	**Secure malware controls**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	4.2
D2.US.H.17	**Secure by default**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	4.6

Table 7.2 displays the feedback received from the experts with regard to the extent to which each of the proposed set of usable security heuristics satisfied the security threats and concerns related to IM applications. As shown in Table 7.2, none of the usable security heuristics received an average score of less than 3.0. The lowest score received was US.H.14 with 3.0, while the highest score received was US.H.02 with 4.75.

When asked whether sufficient IM application security and threat exposure had been addressed by the proposed set of usable security heuristics, all the experts agreed that this had been achieved.

When asked if the experts foresee any challenges when implementing the proposed set of usable security heuristics from an IM security perspective, all the experts agreed that there would be challenges. The challenges mentioned include:

- **Expert 1:** ‘From an actual development perspective, some of the concepts may be logically challenging, but that’s about it’.
- **Expert 3:** ‘As with most security, the biggest problems will stem from the fact that users see security as an obstacle. In IM apps the pressure to prioritise ease of use over security will be extra high’.
- **Expert 4:** The adoption and implementation of the proposed set of usable security heuristics was a concern for an expert. To alleviate this concern, the expert said, *‘I propose that more specific examples for the use of each heuristic is provided to developers in the specific context. This is a general comment that would apply to any heuristics. It would make it easier to understand where each heuristic is applied, making their adoption higher by developers’.*

These challenges are valid and can be addressed by the correct usage of the set of usable security heuristics. To ensure the correct usage, as proposed by one of the experts, providing developers with specific examples of each heuristic could potentially lead to a stronger understanding and more accurate implementation of the set of usable security heuristics.

7.4.3 Usability Section

For this section, the experts were requested to indicate the extent to which each of the proposed usable security heuristic satisfied the usability concerns related to the security of IM applications.

Table 7.3 Expert Feedback Based on Satisfaction of Usability Concerns

Heuristic code	Heuristic name	Usability scale	Average usable security scale
D2.US.H.01	**Visibility of security status**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate **Expert 3:** 3 = Moderate **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	3.6
D2.US.H.02	**Match between security features and the real world**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	4.0
D2.US.H.03	**User security control and freedom**	**Expert 1:** 2 = Low **Expert 2:** 3 = Moderate **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	3.4
D2.US.H.04	**Security consistency and standards**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	4.0
D2.US.H.05	**Security recognition rather than recall**	**Expert 1:** 3 = Moderate **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 4 = High	4.0
D2.US.H.06	**Flexibility and efficiency of use for security features**	**Expert 1:** 2 = Low **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 2 = Low	3.4
D2.US.H.07	**Aesthetic and minimalist security design**	**Expert 1:** 5 = Very High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	4.2
D2.US.H.08	**Threat prevention and user guidance**	**Expert 1:** 4 = High	3.8

		Expert 2: 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	
D2.US.H.09	**Security help and documentation**	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	4.2
D2.US.H.10	**Compliance of security and privacy controls**	**Expert 1:** 3 = Moderate **Expert 2:** 3 = Moderate **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 4 = High	3.2
D2.US.H.11	**Encryption of application session and information**	**Expert 1:** 2 = Low **Expert 2:** 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 4 = High	3.2
D2.US.H.12	**Least privilege by default**	**Expert 1:** 1 = Very Low **Expert 2:** 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 3 = Moderate	2.8
D2.US.H.13	**Secure access control**	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 4 = High	3.4
D2.US.H.14	**Flexibility of user security expertise**	**Expert:** 3 = Moderate **Expert:** 4 = High **Expert:** 3 = Moderate **Expert:** 5 = Very High **Expert:** 3 = Moderate	3.6
D2.US.H.15	**Notification of security updates**	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 3 = Moderate **Expert 4:** 4 = High **Expert 5:** 3 = Moderate	3.4
D2.US.H.16	**Secure malware controls**	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 2 = Low **Expert 4:** 3 = Moderate **Expert 5:** 3 = Moderate	3.0
D2.US.H.17	**Secure by default**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 3 = Moderate **Expert 5:** 5 = Very High	4.2

Table 7.3 displays the feedback received from the experts with regard to the extent to which each of the proposed usable security heuristic satisfied the usability concerns related to the security of IM applications. As shown in Table 7.3, the highest average score received was 4.2, which was achieved for three heuristics, namely US.H.07 aesthetic and minimalist security design, US.H.09 security help and documentation, and US.H.17 secure by default. Only one usable security heuristic, US.H.12 least privilege by default (2.8), received an average score of less than 3.0.

The average score of less than 3.0 for US.H.12, least privilege by default is due to experts rating this usable security heuristic with a 1, with regard to its impact on usability concerns. This low rating is understandable as it focuses on the IM application implementing least privilege by default, which would then not require any interaction with the user, thus not directly affecting the usability of the IM application.

When asked whether the proposed set of usable security heuristics aids in improving the usability of the security within IM applications, all the experts agreed that this had been achieved.

When asked whether the experts foresee any challenges when implementing the proposed set of usable security heuristics from an IM usability perspective, three experts did not foresee any challenges and the other two experts did foresee challenges, although no further elaboration was provided.

7.4.4 Mobile Application Development Section

For this section, the experts were requested to indicate the extent to which each of the proposed usable security heuristic satisfied the IM application development concerns related to the security of IM applications.

Table 7.4 Expert Feedback Based on Satisfaction of IM Application Development Concerns

Heuristic code	Heuristic name	Development scale	Average usable security scale
D2.US.H.01	**Visibility of security status**	**Expert 1:** 3 = Moderate **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	3.8
D2.US.H.02	**Match between security features and the real world**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 3 = Moderate **Expert 5:** 3 = Moderate	3.6
D2.US.H.03	**User security control and freedom**	**Expert 1:** 4 = High **Expert 2:** 3 = Moderate **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 2 = Low	3.0
D2.US.H.04	**Security consistency and standards**	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 4 = High **Expert 5:** 5 = Very High	4.6

D2.US.H.05	**Security recognition rather than recall**	**Expert 1:** 2 = Low **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 3 = Moderate **Expert 5:** 4 = High	3.6
D2.US.H.06	**Flexibility and efficiency of use for security features**	**Expert 1:** 2 = Low **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 4 = High **Expert 5:** 4 = High	3.6
D2.US.H.07	**Aesthetic and minimalist security design**	**Expert 1:** 5 = Very High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 3 = Moderate **Expert 5:** 3 = Moderate	3.8
D2.US.H.08	**Threat prevention and user guidance**	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 4 = High **Expert 5:** 3 = Moderate	3.6
D2.US.H.09	**Security help and documentation**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 3 = Moderate **Expert 5:** 2 = Low	3.4
D2.US.H.10	**Compliance of security and privacy controls**	**Expert 1:** 5 = Very High **Expert 2:** 3 = Moderate **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	4.6
D2.US.H.11	**Encryption of application session and information**	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	4.6
D2.US.H.12	**Least privilege by default**	**Expert 1:** 2 = Low **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 5 = Very High **Expert 5:** 4 = High	4.0
D2.US.H.13	**Secure access control**	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 4 = High	4.2
D2.US.H.14	**Flexibility of user security expertise**	**Expert 1:** 2 = Low **Expert 2:** 3 = Moderate **Expert 3:** 3 = Moderate **Expert 4:** 3 = Moderate **Expert 5:** 3 = Moderate	2.8

D2.US.H.15	**Notification of security updates**	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	3.8
D2.US.H.16	**Secure malware controls**	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 2 = Low **Expert 4:** 5 = Very High **Expert 5:** 3 = Moderate	3.8
D2.US.H.17	**Secure by default**	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 5 = Very High **Expert 4:** 4 = High **Expert 5:** 5 = Very High	4.6

Table 7.4 displays the feedback received from the experts with regard to the extent to which each of the proposed usable security heuristic satisfied the IM mobile application development concerns related to the security of IM applications. As presented in Table 7.4, the highest average score received was US.H.04 security consistency and standards, US.H.10 compliance security and privacy controls, US.H.11 encryption of application session and information, and US.H.17 secure by default with a score of 4.6. One usable security heuristic received an average score of less than 3.0, which is also the lowest score attained. this usable security heuristic was US.H.14, flexibility of user security expertise.

The concern for this usable security heuristic, US.H.14 flexibility of user security expertise, stems from an expert believing that this heuristic and US.H.06 flexibility and efficiency of use for security features, accomplish the same objective. He suggested that US.H.06 and US.H.14 could be combined into a single usable security heuristic. However, the researcher is not in agreement with suggestion, as US.H.06 focuses on the implementation, by IM developers, to allow the security features to be utilised efficiently by both inexperienced and experienced users, while US.H.14 focuses on ensuring that the security features of the IM application are understandable and suitable to all users regardless of their skill level or security expertise. Both heuristics aim to accomplish different objectives and should therefore not be combined.

When asked whether the proposed set of usable security heuristics could be implemented during the IM mobile application development process, all the experts agreed that this was possible.

When asked whether the experts foresee any challenges when implementing the proposed set of usable security heuristics from an IM mobile application development perspective, two experts did not foresee any challenges while the other three experts foresaw the following challenges:

- **Expert 2:** ‘While these heuristics can be implemented during the application development phase, actually doing so would require buy-in from the developer. Too many developers do not conform to acceptable security practices when writing code (especially, less “professional” development companies out to make a buck)’.

- **Expert 4:** ‘Whether the developers’ knowledge is sufficient in the domain of security in order to consider the heuristics in the development process’.

- **Expert 5:** 'Clear instruction and user operation of these settings can be particularly challenging'.

The first challenge is difficult to address as some companies could potentially not prioritise security. Making this set of usable security heuristics available could potentially assist in ensuring that such companies prioritise security. The second and third challenges could potentially be alleviated by accompanying the set of usable security heuristics with a detailed guideline for usage document, which could include relevant examples to support the successful implementation of the proposed heuristics.

7.4.5 General Section

For this section, the experts were requested to indicate their impression of the proposed set of usable security heuristics holistically, regarding their efficacy, utility and quality.

Table 7.5 Expert Feedback Based on Their Holistic View of the Set of Usable Security Heuristics

Characteristic	Usable security scale	Average usable security scale
Efficacy	**Expert 1:** 5 = Very High **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 4 = High **Expert 5:** 4 = High	4.2
Utility	**Expert 1:** 3 = Moderate **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 5 = Very High **Expert 5:** 5 = Very High	4.2
Quality	**Expert 1:** 4 = High **Expert 2:** 4 = High **Expert 3:** 4 = High **Expert 4:** 4 = High **Expert 5:** 4 = High	4.0

Table 7.5 displays the holistic feedback received from the experts with regard to efficacy, utility, and quality of the set of usable security heuristics. The highest scores received were for efficacy and utility, each with a score of 4.2, while the lowest score was for quality with 4.0. None of the average scores presented in Table 7.5, are below 3.0, which confirms that the set of usable security heuristics for instant messaging application development aligns with the requirements of efficacy, utility, and quality.

7.5 Changes Implemented Based on Expert Review

Based on the feedback received from the experts, as reported on in Section 7.4, no specific changes were deemed necessary to the preliminary set of usable security heuristics. The finalised set of proposed usable security heuristics for IM application development remains unchanged, as presented in Table 7.6.

The usable security heuristics are no longer labelled with D2, to signify the finalisation of the proposed usable security heuristics, presented in Table 7.6.

Table 7.6 Finalised Set of Proposed Usable Security Heuristics for Instant Messaging Application Development

Heuristic code	Heuristic name	Definition
US.H.01	**Visibility of security status**	IM applications should always keep users informed about the security status of the application through appropriate feedback within reasonable time.
US.H.02	**Match between security features and the real world**	An IM application's security features should speak the users' language, using real-world standards for terms, phrases, and security ideas with which they are acquainted. This guarantees that the user is well-informed and aware of the influence of the security features on the IM application.
US.H.03	**User security control and freedom**	Users frequently choose IM application security functions by accident, necessitating the presence of a clearly indicated ‘emergency escape’ that allows them to quit the undesirable state without having to go through a lengthy dialogue. Undo and redo are recommended.
US.H.04	**Security consistency and standards**	When using IM applications security features, users should not have to question whether various security phrases, circumstances, or actions imply the same thing. An IM application’s security features should be aligned with other IM applications to ensure that users maintain an understanding of the security features. This ensures that users maintain an understanding of the security features within the IM application environment.
US.H.05	**Security recognition rather than recall**	Make security objects, actions, and choices accessible to reduce IM application user's memory burden. The user should not be required to recall information from one section of the security interaction to the next. When applicable, instructions for using the security features should be visible or easily accessible.
US.H.06	**Flexibility and efficiency of use for security features**	Unseen by the inexperienced user, accelerators may commonly speed up the interaction for the expert user, allowing the security features to accommodate both inexperienced and experienced users. Users should be allowed to customise security-related features that they perform on a regular basis.
US.H.07	**Aesthetic and minimalist security design**	Information that is useless or is seldom used should not be included in security dialogues. In a security dialogue, every additional unit of information, which conflicts with the essential pieces of information, lowers their relative visibility. Ensuring that the security dialogue utilised remains concise and specific to the topic at hand, will improve the user’s decision-making with regard to the impact of the related IM application security features.

US.H.08	**Threat prevention and user guidance**	IM applications should present security messages in a plaintext format to the user. IM applications should guide the user during usage by hiding unavailable functions, warning users about their actions, and assisting users to recognise, diagnose, and avoid potential threats.
US.H.09	**Security help and documentation**	Even though it is preferable for the security features to be operated without documentation, assistance and documentation may be required. Any such security information should be simple to find, should concentrate on the user's security duty, should have a list of clear procedures to follow, and should be manageable in size.
US.H.10	**Compliance of security and privacy controls**	IM applications must provide the current industry standard of security and privacy controls with basic plaintext instructions for users on how to implement and utilise these features effectively. The privacy features need to align with international standards, such as the South African Protection of Personal Information Act (POPIA) and the European General Data Protection Regulation (GDPR).
US.H.11	**Encryption of application session and information**	IM applications need to be encrypted to the current industry level of encryption. The encryption level must be made clear to the user. If there is more than one level of encryption available, it must be clear which is active, and the user must be guided in how to select the relevant encryption feature. It is crucial for an IM application to encrypt the application session and the storage and transmission of information.
US.H.12	**Least privilege by default**	IM applications need to be developed with the principle of least privilege, which is to ensure that the permissions requested by the application is limited to the minimum permissions required for functionality. IM applications must not request more permissions than those required. Each permission requested must be clearly and concisely explained to the user, to ensure that an informed decision is made by the user. This will also reduce the cognitive load on the user.
US.H.13	**Secure access control**	No unauthorised access must be given to an IM application. The application must secure itself from all forms of attempted access from unauthorised entities.
US.H.14	**Flexibility of user security expertise**	The security features of IM applications need to provide plaintext options suitable for users with diverse levels of skills and experience in security.
US.H.15	**Notification of security updates**	To ensure optimal security, IM applications need to alert the user about application updates. To mitigate vulnerabilities of older applications, IM applications need to remain updated.
US.H.16	**Secure malware controls**	IM applications need to implement controls to detect, prevent and recover from malware. Such applications should also inform and keep users aware of the situation.

US.H.17	**Secure by default**	IM applications need to ensure that the optimal security settings are active, by default. This will reduce the chances of IM applications being utilised with weaker security.

Table 7.6 presents the finalised set of usable security heuristics for instant messaging application development. This set of usable security heuristics was therefore used in the development of a proof-of-concept prototype to demonstrate the applicability of the proposed set of usable security heuristics to a typical instant messaging application, as discussed in Chapter 8.

7.6 Conclusion

The preliminary set of usable security heuristics, presented in Chapter 6, Section 6.4.2, Table 6.9, underwent validation in the form of an expert review. The expert review highlighted the validity of the set of usable security heuristics from three varying perspectives, namely security, usability, and mobile application development.

In addition, the set of usable security heuristics met the requirements of efficacy, utility, and quality, further highlighting the validity of the set of usable security heuristics for instant messaging application development. The conducting of the expert review fulfilled the validation requirements of the PRO, *to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*.

The following chapter, Chapter 8, further validates the proposed set of usable heuristics by presenting a proof-of-concept prototype to demonstrate the applicability of each heuristic to a specific IM application, namely Facebook Messenger.

Chapter 8 – Proof-of-Concept Prototype

8.1 Introduction

In Chapter 7, the finalised set of usable security heuristics for instant messaging application development was presented.

The aim of this chapter is to meet the requirements of SRO4 of this study, namely *to develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application*. This was accomplished by developing a proof-of-concept prototype. The finalised set of usable security heuristics for instant messaging application development from Chapter 7, Section 7.2.4, Table 7.6, was utilised in the development of this prototype.

The chapter structure is as follows: Section 8.2 focuses on the overview of the prototype, while Section 8.3 documents the selection of the IM application utilised in the development of the proof-of-concept prototype. Section 8.4 focuses on the application of the set of usable security heuristics to the selected IM application, and Section 8.5 concludes the chapter.

8.2 Overview of Prototype

To further validate the proposed set of usable security heuristics, a prototype was developed. Prototypes are produced for analysis, demonstration, or research purposes. Prototyping could be used to test that the designed elements were correct (Norgren, 2004; Hess, 2012; Jobbins, 2012;). Proper use of prototyping often increased the efficiency of the production process (Norgren, 2004). The development of a prototype in this study is important as it could look and/or act similarly to the targeted design, which provides an idea of how the set of usable security heuristics for instant messaging application development could impact a typical IM application. In addition, the prototype can provide an IM application developer with an example of how to implement the set of usable security heuristics for instant messaging application development onto an IM application.

To accomplish the requirements set out by SRO4 of this study, a prototype, in the form of proof-of-concept prototype (POCP), was developed. The definition of a prototype and the aim of the POCP for this study were documented in Chapter 1, Section 1.6.5. To reiterate, the aim of the POCP is *to visualise the proposed usable security heuristics applied to a typical IM application.* The applicability of the finalised set of usable security heuristics was therefore demonstrated through the POCP.

8.3 Selection of the Instant Messaging Application for the Prototype

Before the development of the proof-of-concept prototype could begin, the IM application to be evaluated against the proposed set of usable security heuristics had to be identified. From the study of common IM applications conducted in Chapter 3, Section 3.4, the three top candidates included WhatsApp, Facebook Messenger, and WeChat. These were considered to be the top candidates as they are the only three IM applications to have over 1000 million monthly active users (Statista, 2021a). These three IM applications also have similar features, as depicted in Table 3.3 (Chapter 3, Section 3.4). Furthermore, from the study of IM security and privacy features, conducted in Chapter 4, Section 4.3, WeChat had the weakest IM security and privacy features, followed by Facebook Messenger, while WhatsApp was found to have the best security and privacy features.

Since WhatsApp is the most popular IM application, but also the most secure of the three IM applications considered, it was decided that WhatsApp would not be utilised for the POCP. On the other hand, WeChat is the least popular of the three applications and the weakest in terms of IM security and privacy features. However, information on WeChat is difficult to acquire and the majority of the user base is located in Asia (Bucher, 2020; Statista, 2021a). For these reasons WeChat was not selected for the POCP. Facebook Messenger is the second most popular IM application of the three IM applications examined, and it is also the second most secure IM application. In addition, Facebook Messenger has a globally spread user base and is not limited to an individual location (Bucher, 2020). For these reasons Facebook Messenger was selected as the most appropriate IM application for the POCP.

8.4 Application of the Set of Usable Security Heuristics

In the development of the prototype, Facebook Messenger was analysed from the perspective of each individual usable security heuristic as presented in Chapter 7, Section 7.2.4, Table 7.6. The analysis identified relevant sections of Facebook Messenger where the set of usable security heuristics could be applied.

The POCP was developed using screenshots taken from a device which utilised the Android 10 operating system. The installed Facebook Messenger application was up to date and last updated with the 02 August 2021 updates.

8.4.1 US.H.01 – Visibility of Security Status

This usable security heuristic requires that *an IM application should constantly keep users informed about the program's security state by providing relevant feedback in a timely manner*.

From the evaluation conducted, Facebook Messenger currently appears to lack updates and alerts with regard to its security status and the status of the available security features. Figure 8.1 depicts the current general chat screen. From Figure 8.1 it is evident that Facebook Messenger does not inform or update the user on which form of encryption is being utilised by the IM application, thus not adhering to this heuristic (US.H.01). Figure 8.2 provides an example of how this could be addressed within Facebook Messenger.

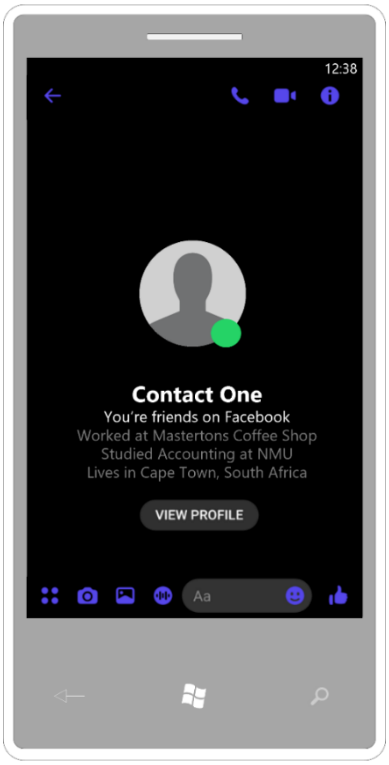

12:38
Contact One
You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa
VIEW PROFILE
Aa

Figure 8.1 Current Facebook Messenger General Chat Screen

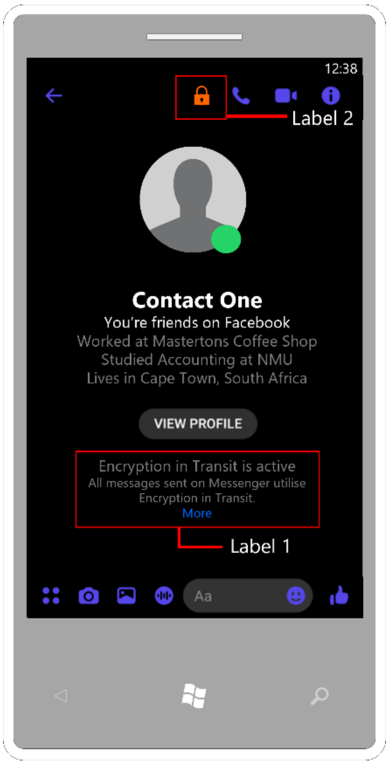

12:38
Label 2
Contact One
You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa
VIEW PROFILE
Encryption in Transit is active
All messages sent on Messenger utilise Encryption in Transit.
More
Label 1
Aa

Figure 8.2 US.H.01 Applied to Facebook Messenger General Chat Screen

As depicted in Figure 8.2, Label 1, the application of US.H.01 includes a status update or alert to ensure that the user is informed of the encryption status of the IM application. In this example, Encryption in Transit is active. A *'More'* link is included in the status update, which will be explored at a later stage. It therefore reminds the user that encryption is being utilised in addition to the type of encryption that is being implemented. Similarly, this could be done for various other status updates.

In addition to the status message presented in Figure 8.2, Label 2, a lock symbol was added. The status message would disappear as the conversation progresses. The lock symbol would be locked when encryption is active and unlocked when no encryption is active. The lock could also display different colours based on the status of the IM applications encryption. For example, when the lock is purple, optimal encryption is active; however, when the lock is orange, as in Figure 8.2, optimal encryption is not active. Therefore, the addition of the ever-present lock symbol will assist in reminding users that encryption is being utilised.

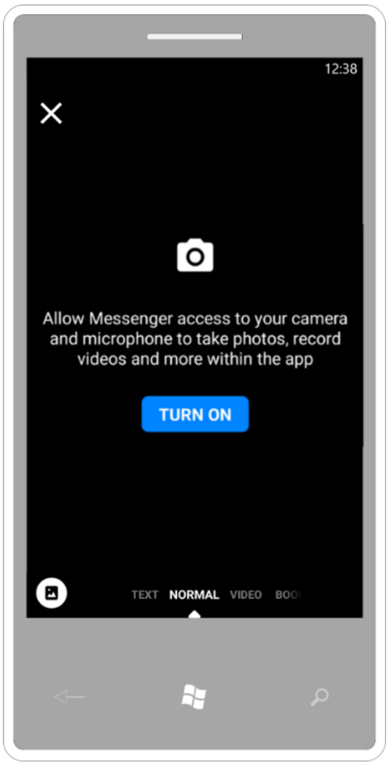
As depicted in Figure 8.2, the inclusion of a security status update on encryption keeps users informed of the encryption status of the IM application. As stated in Chapter 4, Section 4.3.1, Facebook Messenger does not implement end-to-end encryption by default. The default method of encryption utilised by Facebook Messenger is currently encryption in transit. The security status update in Figure 8.2 alerts users to the current active form of encryption.

The inclusion of a security status update, as depicted in Figure 8.2, would ensure that Facebook Messenger users are informed of the application's security state in a timely manner, thus adhering to heuristic **US.H.01 – Visibility of Security Status**.

8.4.2 US.H.02 – Match Between Security Features and the Real World

This usable security heuristic requires that *an IM application's security features should speak the users' language, using terms, phrases, and security ideas that they are acquainted with. Real-world standards should be followed to present data in a logical and natural arrangement.*

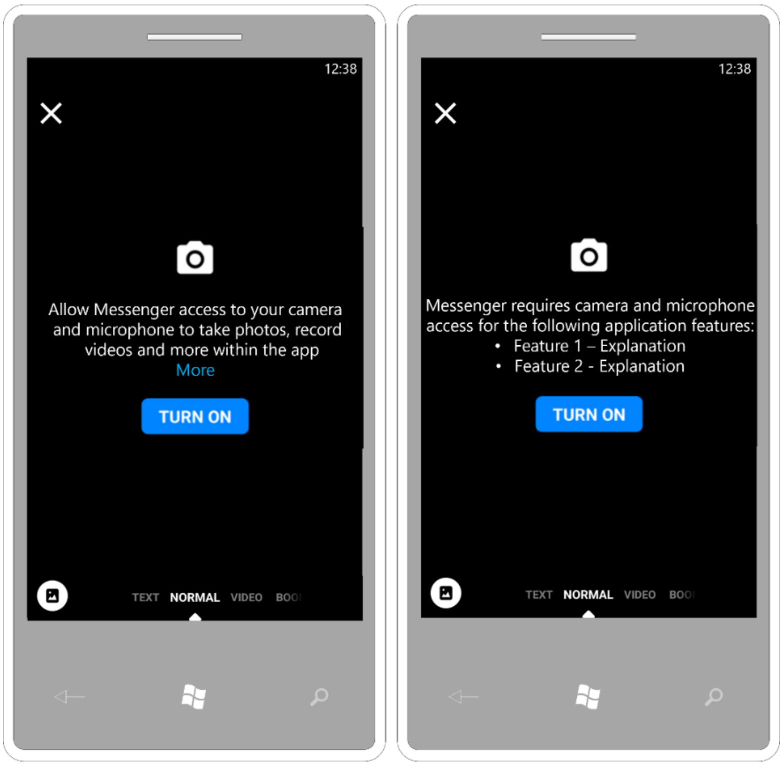
From the evaluation conducted, there is no evidence that Facebook Messenger is currently transparent nor informative with the permissions requested. They currently neglect to inform the user of the security implications of the permissions requested, and how they may affect the security of the user's information. Figure 8.3 depicts an example of Facebook Messenger’s current camera permission request screen.



Allow Messenger access to your camera and microphone to take photos, record videos and more within the app

Figure 8.3 Current Facebook Messenger Camera Permission Request Screen

As displayed in Figure 8.3, the permission request presented to the user is not informative. The request is deemed to be ambiguous, as Facebook Messenger does not elaborate on the meaning of *‘and more within the app’*. The unknown uses related to this permission request could potentially jeopardise the security and privacy of the user’s information.



Allow Messenger access to your camera and microphone to take photos, record videos and more within the app

Messenger requires camera and microphone access for the following application features:
- Feature 1 – Explanation
- Feature 2 - Explanation

Figure 8.4 US.H.02 Applied to Facebook Messenger Camera Permission Request Screen

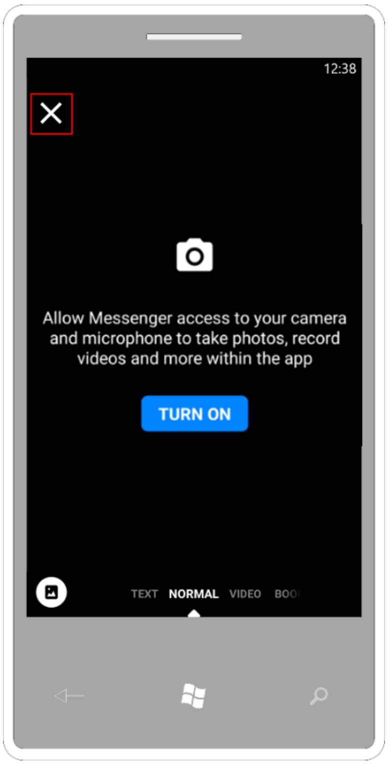
Figure 8.4 presents a revised version of a Facebook Messenger permission request after implementing US.H.02. The *'More'* link located on the left-hand screen leads to the information displayed on the right-hand screen, thereby alerting the user to the potential impact and influence that the requested permission could have on the IM application, if granted.

Including the relevant explanation and information, within the IM security feature (permission request in this example), in a plaintext format aligns with the requirements of **US.H.02 – Match Between Security Features and the Real World**.

8.4.3 US.H.03 – User Security Control and Freedom

This usable security heuristic refers to the concern that *users frequently choose IM application security functions by accident, necessitating the presence of a clearly indicated 'emergency escape' that allows them to quit the undesirable state without having to go through a lengthy dialogue.*

From the evaluation conducted, Facebook Messenger currently ensures that appropriate and effective security navigation is made available to users. The navigation implemented enables users to navigate effectively and efficiently through the Facebook Messenger security features while avoiding drawn-out dialogues.



12:38

Allow Messenger access to your camera and microphone to take photos, record videos and more within the app

Figure 8.5 Indication of Clear Exit on Current Facebook Messenger Camera Permission Request

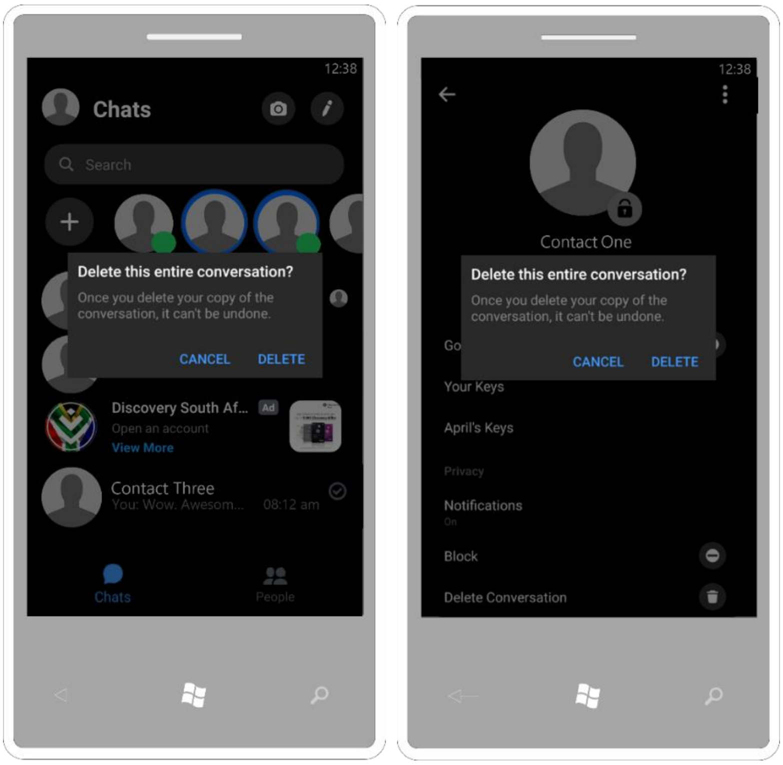
Figure 8.5 shows how an indicated exit from the permission request security feature is presented to the user, located within the red square. When utilised, this exit takes the user back to their previous screen, without unnecessary security dialogue. This clearly indicates that exit options are present within Facebook Messenger.

The existence of these navigation options already aligns with the requirements of **US.H.03 – User Security Control and Freedom**.

8.4.4 US.H.04 – Security Consistency and Standards

This usable security heuristic refers to the concern that *when using an IM application's security features, users should not have to question whether various phrases, circumstances, or actions imply the same thing. An IM application should observe the security protocols established by IM applications and other applications.*

From the evaluation conducted, Facebook Messenger currently maintains consistency throughout the application's operation and implementation of security features. By maintaining this consistency and standard, users become familiar with the IM application security features and find it easier to utilise.

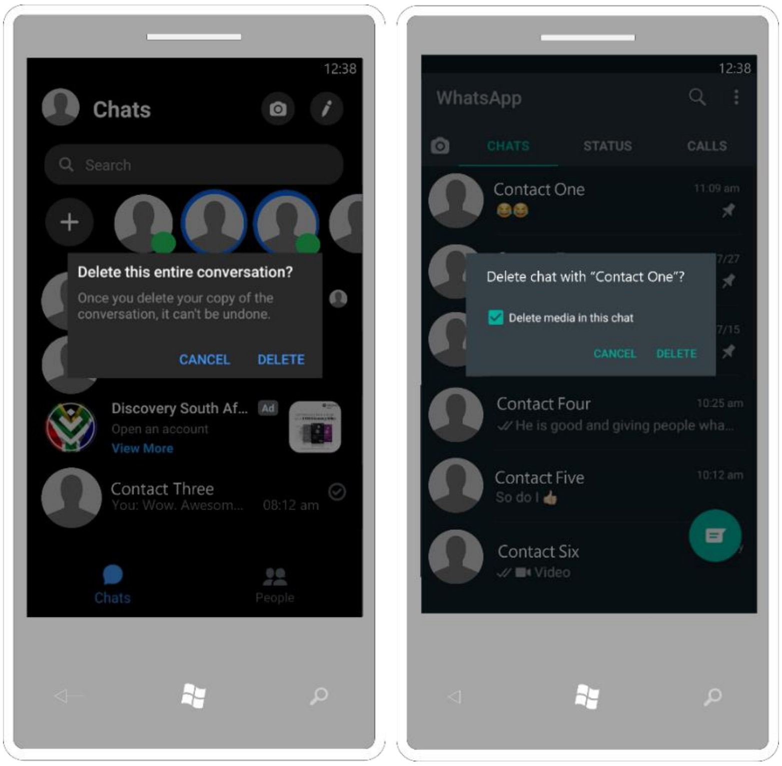


Delete this entire conversation?

Once you delete your copy of the conversation, it can't be undone.

Figure 8.6 Indication of Consistency Across the Current Facebook Messenger Deletion of Chats Security Feature

Figure 8.6 presents two examples of the deletion of conversation (chats) security feature within the Facebook Messenger application. The left-hand screen shows the deletion of a regular conversation, while the right-hand screen shows the deletion of a secret conversation. From the comparison of these two screens, it was noted that the conversation deletion security feature followed the same standard and was consistent across the Facebook Messenger conversation. The phrases, circumstances and actions consistently implied the same thing.



Delete this entire conversation?

Once you delete your copy of the conversation, it can't be undone.

CANCEL DELETE

Delete chat with "Contact One"?

☑ Delete media in this chat

CANCEL DELETE

Figure 8.7 Indication of Consistency of the Current Facebook Messenger Deletion of Chats Security Feature in Comparison to WhatsApp

Facebook Messenger did not maintain consistency with other IM applications, like WhatsApp. Figure 8.7 shows a comparison of Facebook Messenger's conversation deletion security feature and the chat deletion security feature of WhatsApp.

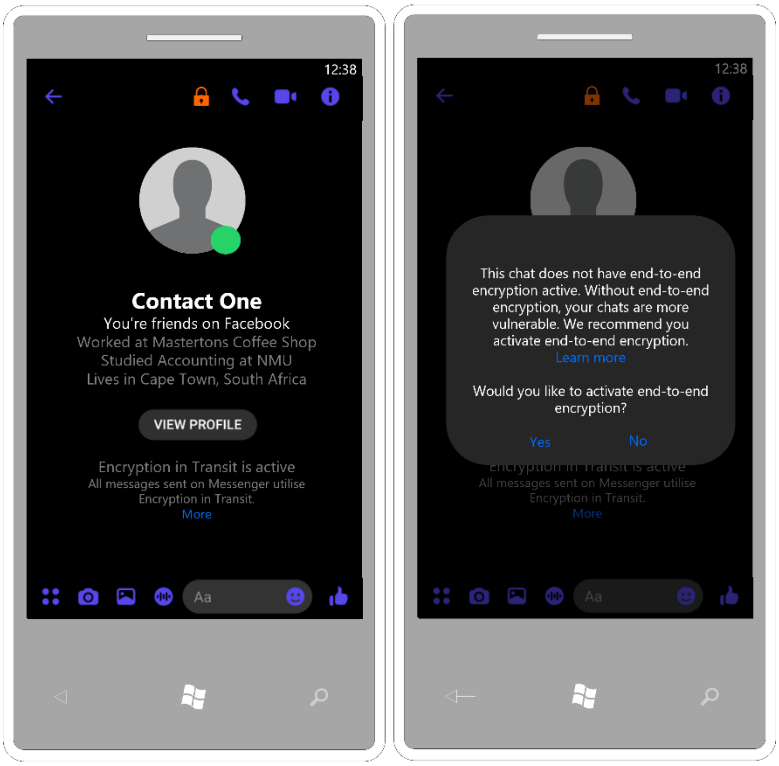
Both Facebook Messenger and WhatsApp adhere to similar display and utilise plain language that users would be familiar with. However, WhatsApp provides their users with an option to remove all media linked to the chat being deleted, while Facebook Messenger does not provide this for their conversations.

Facebook Messenger displays the evidence of internal security consistency and standards but does not provide evidence for external security consistency and standards. Therefore, Facebook Messenger partially aligns with the requirements related to **US.H.04 – Security Consistency and Standards**.

8.4.5 US.H.05 – Security Recognition Rather than Recall

This usable security heuristic refers to the concern that *the user should not be required to recall information from one section of the security interaction to the next. When applicable, instructions for using the security features should be visible or easily accessible.*

From the evaluation conducted, it appears that Facebook Messenger currently does not consistently provide users with enough information to utilise security features with ease. This lack of information increases the burden on the memory of the IM application user. To ease this memory burden, users should be provided with easy access to the relevant information required to operate the various Facebook Messenger security features.



Contact One
You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa

VIEW PROFILE

Encryption in Transit is active
All messages sent on Messenger utilise Encryption in Transit.
More

This chat does not have end-to-end encryption active. Without end-to-end encryption, your chats are more vulnerable. We recommend you activate end-to-end encryption.
Learn more

Would you like to activate end-to-end encryption?

Yes No

Figure 8.8 US.H.05 Applied to Facebook Messenger General Chat Screen

Figure 8.8 displays an edited version of the Facebook Messenger chat screen; the current screen can be seen in Figure 8.1. The chat screen was previously edited in Figure 8.2, to display the status of encryption in the chat. Within the edit located in Figure 8.2, a *'More'* link was added to the screen, which can also be seen in Figure 8.8 on the left-hand screen. The *More*' link would navigate the user to the right-hand screen of Figure 8.8. The right-hand screen displays all the necessary information to the user with regard to the security feature being utilised. In this example, the user is being prompted to activate end-to-end encryption. Ensuring that this information is easily accessible and presenting it in a plaintext, easy to read manner, could potentially lower the memory burden on the user.

Facebook Messenger has designed their security features to be easy to use. This is displayed in Figure 8.6, as the deletion of chats feature was easy to use. However, the lack of clear visibility and access to instructions or information could potentially increase a user's memory burden. Including clear visibility and access to instructions or information in Facebook Messengers easy-to-use security features, aligns with the requirements of **US.H.05 – Security Recognition Rather than Recall**.

8.4.6 US.H.06 – Flexibility and Efficiency of Use for Security Features

This usable security heuristic refers to the concern that *IM applications lack accelerators. Unseen by the inexperienced user, accelerators may commonly speed up the interaction for the expert user, allowing the security features to accommodate both inexperienced and experienced users. Users should be allowed to customise security-related activities that they perform on a regular basis.*

From the evaluation conducted, Facebook Messenger appears to lack an accelerator to increase the flexibility of utilising the end-to-end encryption security feature. To activate a conversation which utilises end-to-end encryption, a user is currently required to start a new chat. The user then activates the secret conversation, as seen in the red squares. After activating the secret conversation, the user selects the contact they want to chat with and then they can chat with end-to-end encryption activated. The activation of the secret conversation from creating a new message screen is shown in Figure 8.9.

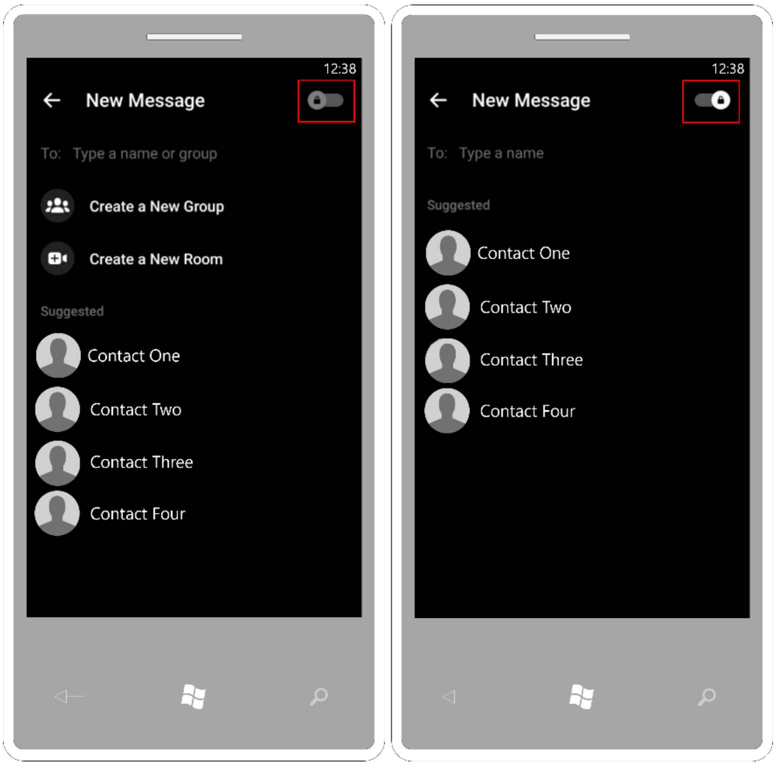
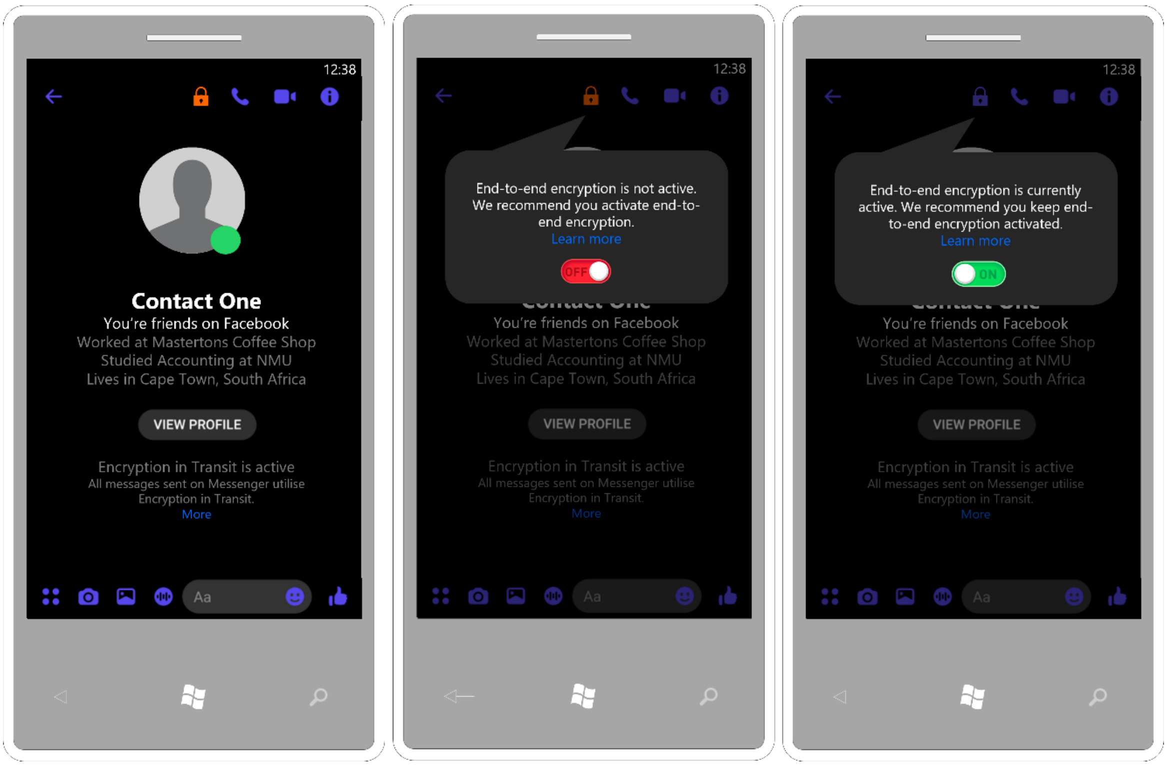


Figure 8.9 Current Facebook Messenger Secret Conversation Activation Screen

To provide flexibility and efficiency to the users, multiple options should be provided to activate end-to-end encryption. For the additional end-to-end encryption accelerators, two locations were identified. First, as Figure 8.10 depicts, within the chat itself and second, as Figure 8.11 depicts, on the chat display screen.



You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa

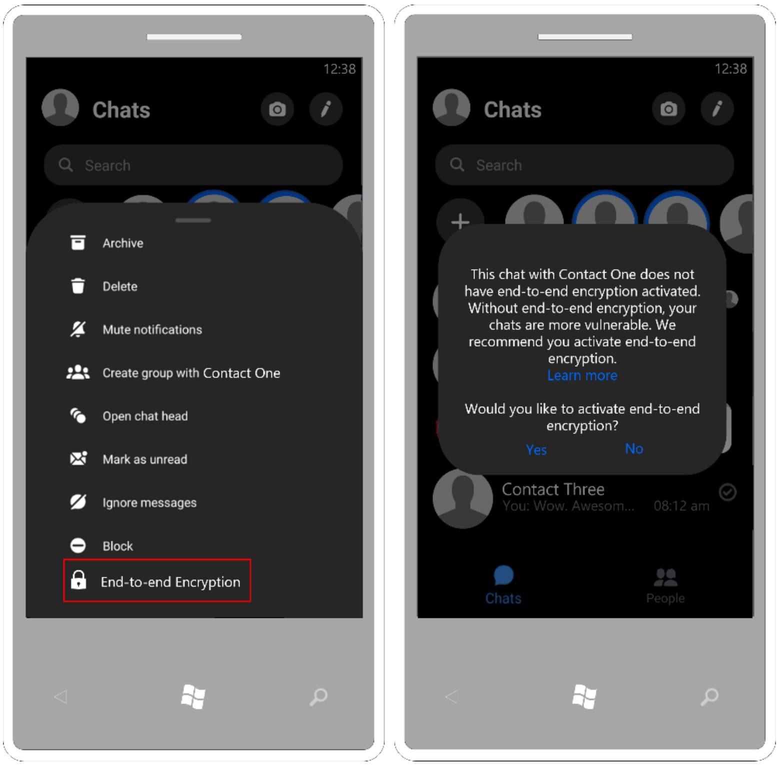
Encryption in Transit is active
All messages sent on Messenger utilise Encryption in Transit.

End-to-end encryption is not active. We recommend you activate end-to-end encryption.
Learn more

End-to-end encryption is currently active. We recommend you keep end-to-end encryption activated.
Learn more

Figure 8.10 US.H.06 Applied to Facebook Messenger General Chat Screen

Figure 8.10 displays the accelerator to activate end-to-end encryption from within the chat itself. The left-hand screen displays the chat with the status update, which notifies users that encryption in transit is currently active. The previously added lock icon introduces the accelerator, which leads to the middle- and right-hand screens that provide users with a brief description of the situation and a link to a more detailed description. The user is also presented with a toggle option to activate or deactivate end-to-end encryption for this specific chat, as seen in the middle- and right-hand screens of Figure 8.10.



This chat with Contact One does not have end-to-end encryption activated. Without end-to-end encryption, your chats are more vulnerable. We recommend you activate end-to-end encryption.

Would you like to activate end-to-end encryption?

Figure 8.11 US.H.06 Applied to Facebook Messenger General Chat Display Screen

Figure 8.11 displays the accelerator to activate end-to-end encryption from the chats display screen. The left-hand screen displays the options provided to the user when a chat was highlighted. The additional option added, to introduce the accelerator, was *'end-to-end encryption*, which can be seen at the bottom of the list of features in the red square. When selecting this option, the right-hand screen is then displayed. The right-hand screen displays the same details as the right-hand screen of Figure 8.10 with the same brief description of the situation and a link to a more detailed description. The user is then also provided with the option to activate end-to-end encryption for this specific chat via the Yes and No options provided on the right-hand screen.

The inclusion of these accelerators, as depicted in Figures 8.10 and 8.11, would allow experienced users to speed up their interaction with Facebook Messenger's security features, which could lead to an improvement of their user experience, thus adhering to heuristic **US.H.06 – Flexibility and Efficiency of Use for Security Features.**

8.4.7 US.H.07 – Aesthetic and Minimalist Security Design

This usable security heuristic refers to the concern that *information that is useless or is used seldom should not be included in security dialogues. In a security dialogue, every additional unit of information which conflicts with the essential pieces of information, lowers its relative visibility. Ensuring that the security dialogue utilised remains concise and specific to the topic at hand, will improve the user's decision-making with regard to the impact of the related IM application security feature.*

From the evaluation conducted, Facebook Messenger currently provides users with concise security dialogues with regard to their security features; however, the security dialogues tend to leave out important details and information. Since the user is often not adequately informed to make a security decision, this could be potentially harmful to them. The security feature utilised in Figure 8.3 is the permission system, which is currently showing a permission request. As previously stated, the description accompanying the permission request does not contain all the important information to make a decision. The description is also ambiguous and could potentially leave users with more questions.

Figure 8.4 presents the edited version of the permission request on the left-hand screen. The addition of the *'More'* link provides the users with an option to acquire additional information on the topic, while not distracting them with pointless information. The right-hand screen presents this important information in a concise manner, without additional content to mislead or distract the user. This important information assists the user in making an informed security decision.

The inclusion of access to additional information and ensuring that other meaningless content does not conflict with the important information presented within the security dialogue, as depicted in Figure 8.4, aligns with the requirements of **US.H.07 – Aesthetic and Minimalist Security Design**.

8.4.8 US.H.08 – Threat Prevention and User Guidance

This usable security heuristic refers to the concern that *an IM application should present security messages in a plaintext format to the user. The application should guide the user during usage, by hiding unavailable functions, warning users about their actions, and assisting users to recognise, diagnose, and avoid potential threats.*

From the evaluation conducted, Facebook Messenger currently appears to lack threat prevention and user guidance. This was noted in the situation surrounding their lack of end-to-end encryption. They did not implement it by default, did not inform the user of this, and they did not guide the user to implement it themselves. This could potentially lead to more threats being introduced, as users currently use Facebook Messenger with a false sense of security.

The lack of threat prevention and user guidance was seen again with the permission requests since Facebook Messenger currently presents the user with ambiguous statements that do not inform them of the full impact relating to the permissions being requested. This lack of user guidance could lead to the introduction of potential threats. Currently, the user is also not guided to further information on the permissions requested.

Facebook Messenger needs to present their security messages in plaintext to ensure that both experienced and inexperienced users can easily understand them. Facebook Messenger should provide guidance to users, ensuring that assistance and information is available to help users to recognise, diagnose, and prevent potential IM security threats.

Figures 8.4 and 8.8 display different forms of user guidance when utilising Facebook Messenger's security features. This user guidance links to threat prevention; when the IM security feature is correctly utilised, these features assist in mitigating threats to the IM application.

As depicted in Figure 8.4, presenting users with a *'More'* link, located on the left-hand screen, leads to the information displayed on the right-hand screen, thereby alerting the user to the potential impact and influence the requested permission could have on the IM application, if granted. This form of user guidance assists users in preventing unwanted, unnecessary, or even dangerous permissions from being granted, therefore, potentially assisting in mitigating threats to the IM application.

Figure 8.8 displays an edited version of the Facebook Messenger chat screen. The *'More'* link, located on the left-hand screen, would navigate the user to the right-hand screen of Figure 8.8. The right-hand screen displays all the necessary information to the user with regard to the security feature being utilised. In this example, the user is being prompted to activate end-to-end encryption. This form of user guidance assists users in ensuring that the appropriate level of encryption is implemented within their chats. Ensuring that the appropriate level of encryption is utilised assists in securing the contents of the users' chats from unauthorised individuals, therefore, potentially assisting in mitigating threats to the IM application.

The inclusion of user guidance, as depicted in Figures 8.4 and 8.8, and its link to threat prevention assists in ensuring that Facebook Messenger aligns with the requirements of **US.H.08 – Threat Prevention and User Guidance**.

8.4.9 US.H.09 – Security Help and Documentation

This usable security heuristic refers to the concern that *even though it is preferable if the security feature can be operated without documentation, assistance and documentation may be required. Any such security information should be simple to find, should concentrate on the user's security duty, should have a list of clear procedures to follow, and should be manageable in size.*

Facebook Messenger currently contains help and documentation information for the majority of the application's operation, located within the profile settings. However, the current security help information is inadequate. The help and documentation are currently located under Help, as displayed in Figure 8.12, and highlighted by the red square on the right-hand screen.

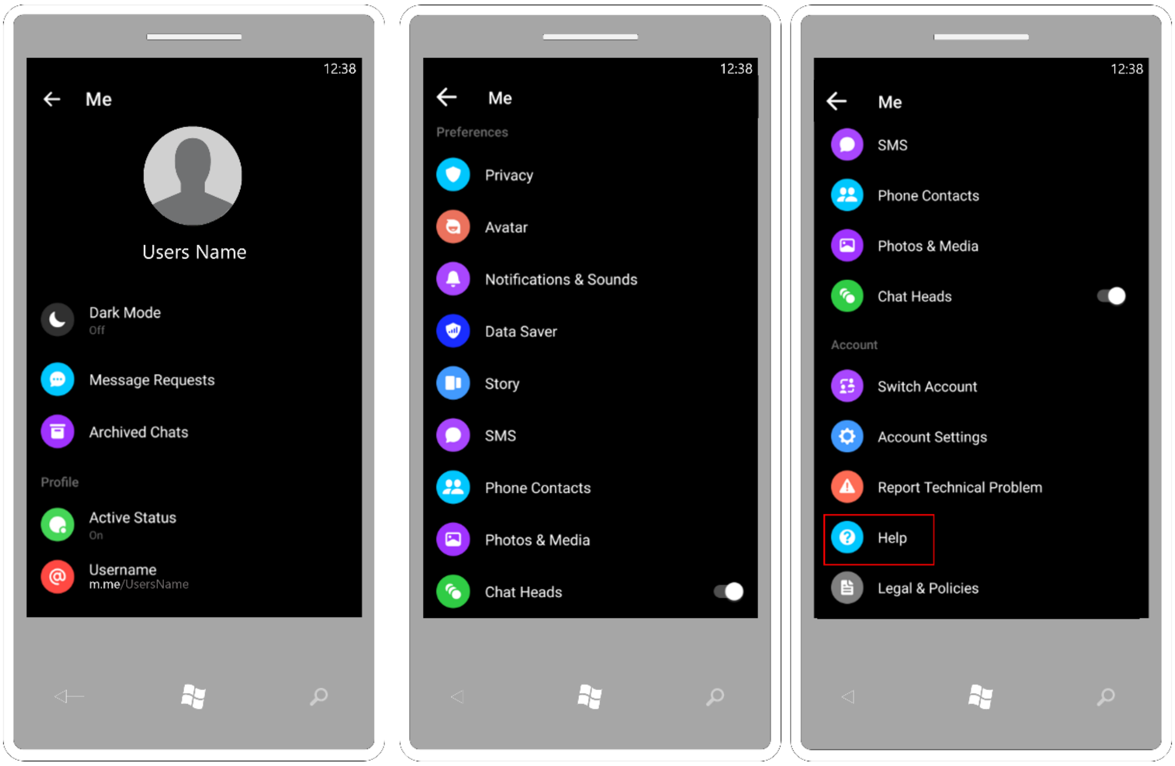
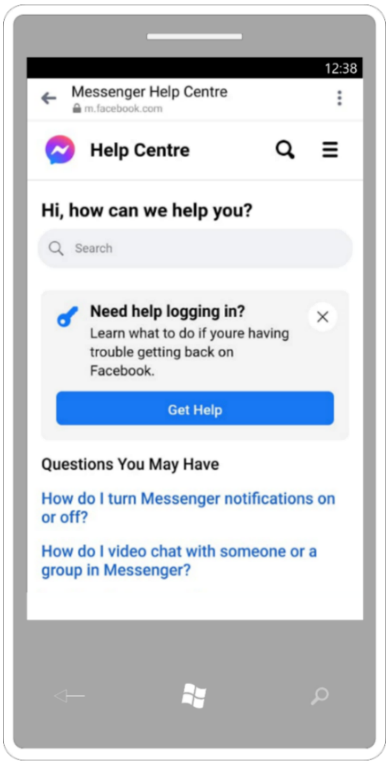


Figure 8.12 Current Facebook Messenger Help and Documentation

As depicted in Figure 8.12, the Help feature is currently poorly located. The Help feature is the second last option located within the profile setting. To access the Help feature, a user must scroll through the other options located in the left-hand- and middle screens. The Help feature is essentially buried beneath the other options. It is not easily seen, which could potentially be an issue for an inexperienced user or a user who is not familiar with the layout of Facebook Messenger.



12:38

Messenger Help Centre
m.facebook.com

Help Centre

Hi, how can we help you?

Search

Need help logging in?
Learn what to do if youre having trouble getting back on Facebook.

Get Help

Questions You May Have

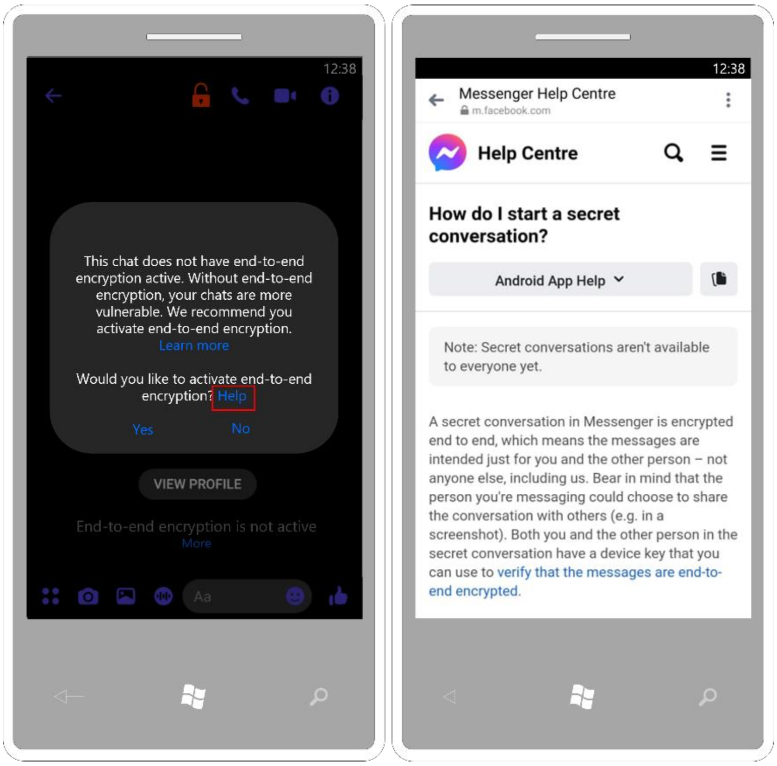
How do I turn Messenger notifications on or off?

How do I video chat with someone or a group in Messenger?

Figure 8.13 Current Facebook Messenger Help Center Home Screen

Figure 8.13 displays the screen currently presented to users after following the Help option displayed in Figure 8.12. Users can search and navigate through the Facebook Messenger Help Center to locate the required information. Although all the help information is available through the Facebook Messenger Help Center, it is difficult to navigate and utilise. In addition to this, the current security help information located in the Facebook Messenger Help Center is inadequate. Facebook Messenger needs to ensure that their security help and documentation is kept up to date with the development of the platform and that help is available for all features located within the application.

Although Facebook Messenger currently provides access to their help and documentation, it could be improved upon. For instance, linking security features directly to their relevant help and documentation could prevent users from needing to search through the entire Help section. Directly navigating the user to the relevant help regarding its security features could also ensure that the user is receiving the appropriate information for these features in an efficient manner.



This chat does not have end-to-end encryption active. Without end-to-end encryption, your chats are more vulnerable. We recommend you activate end-to-end encryption. Learn more

Would you like to activate end-to-end encryption? Help

End-to-end encryption is not active
More

How do I start a secret conversation?

Note: Secret conversations aren't available to everyone yet.

A secret conversation in Messenger is encrypted end to end, which means the messages are intended just for you and the other person – not anyone else, including us. Bear in mind that the person you're messaging could choose to share the conversation with others (e.g. in a screenshot). Both you and the other person in the secret conversation have a device key that you can use to verify that the messages are end-to-end encrypted.

Figure 8.14 US.H.09 Applied to Facebook Messenger General Chat Screen

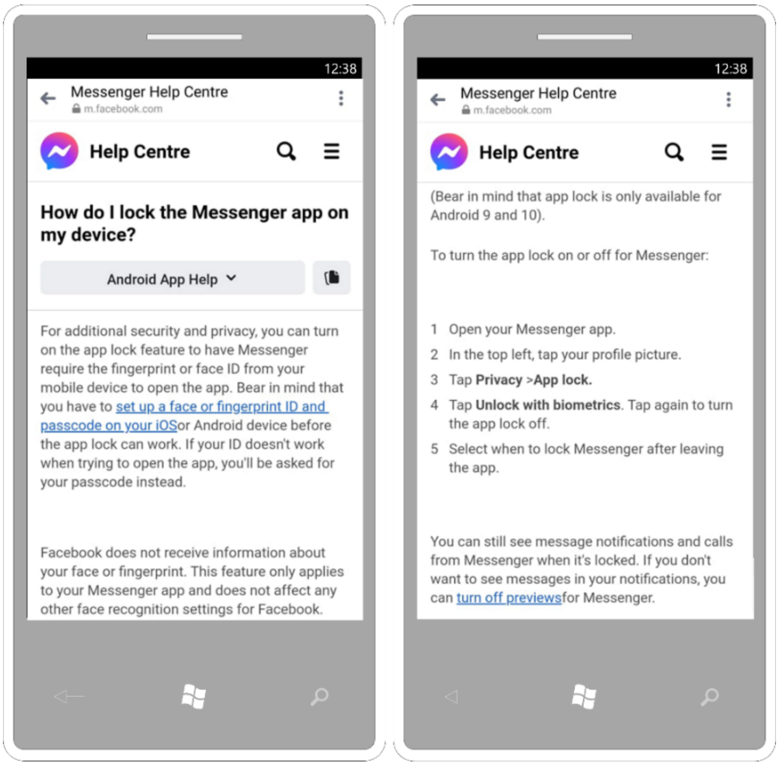
Figure 8.14 displays an edited version of the Facebook Messenger chat screen after following the *'More'* link located in the status of the encryption notification. To navigate users efficiently and effectively to the relevant help for its security feature, the *'Help'* link has been included in the design, located on the left-hand screen in the red square. The inclusion of this *'Help*' assists in mitigating the need for users to search through the entire Messenger Help Center to locate the information they require.

Ensuring that adequate help and documentation is provided for all the security features located within Facebook Messenger and the inclusion of directly linking an IM security feature to its relevant help and documentation, as depicted in Figure 8.14, assists in ensuring that Facebook Messenger aligns with the requirements of **US.H.09 – Security Help and Documentation**.

8.4.10 US.H.10 – Compliance of Security and Privacy Controls

This usable security heuristic refers to the concern that *an IM application must provide the current industry standard of security and privacy controls with basic plaintext instructions for users on how to implement and utilise these controls effectively. The privacy controls need to align with international standards, such as the South African Protection of Personal Information Act (POPIA) and the European General Data Protection Regulation (GDPR).*

As previously stated in Chapter 4, Section 4.3, Facebook Messenger implements various security controls including: end-to-end encryption, encryption in transit, deleting of messages, self-destructing messages, two-factor authentication, verification SMS/email, and a password lock. However, end-to-end encryption and self-destructing messages are not available to users within a regular chat. Users are required to activate a secret conversation to gain access to these security and privacy controls. However, this is not made clear to users upon their initial use of Facebook Messenger. In addition, as previously stated in Chapter 4, Section 4.3.6, Facebook Messenger provides users with a password lock called App Lock. However, App Lock has not been made available to Android users and is only accessible to users on iOS devices.



How do I lock the Messenger app on my device?

For additional security and privacy, you can turn on the app lock feature to have Messenger require the fingerprint or face ID from your mobile device to open the app. Bear in mind that you have to set up a face or fingerprint ID and passcode on your iOSor Android device before the app lock can work. If your ID doesn't work when trying to open the app, you'll be asked for your passcode instead.

Facebook does not receive information about your face or fingerprint. This feature only applies to your Messenger app and does not affect any other face recognition settings for Facebook.

(Bear in mind that app lock is only available for Android 9 and 10).

To turn the app lock on or off for Messenger:

1 Open your Messenger app.
2 In the top left, tap your profile picture.
3 Tap **Privacy** >**App lock.**
4 Tap **Unlock with biometrics**. Tap again to turn the app lock off.
5 Select when to lock Messenger after leaving the app.

You can still see message notifications and calls from Messenger when it's locked. If you don't want to see messages in your notifications, you can turn off previewsfor Messenger.

Figure 8.15 Current Facebook Messenger App Lock Help and Documentation

Figure 8.15 displays the help and documentation of the App Lock security control. As stated previously, Android 10 was utilised in the development of this POCP. However, even though Android 10 was utilised, App Lock was unavailable.

As previously stated in Chapter 4, Section 4.3.6, App Lock is Facebook Messenger's version of a password lock. The purpose of a password lock is to add an additional layer of security to protect the confidential information located within an application. The addition of App Lock to the Facebook Messenger application provides an supplementary security layer to protect the highly confidential information located within Facebook Messenger.

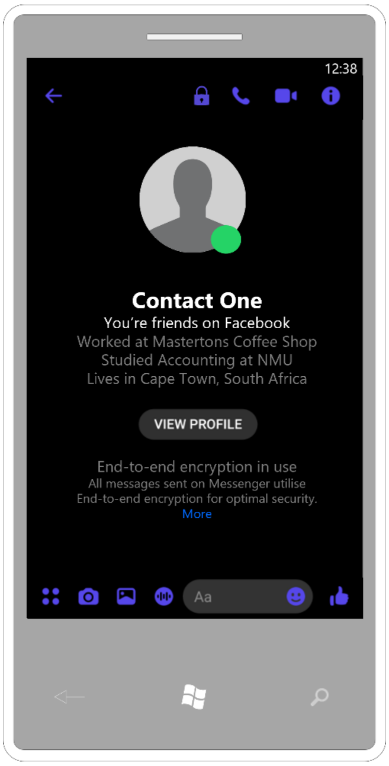
In addition to providing another layer of security protection to Facebook Messenger, App Lock would also assist Facebook Messenger in aligning with the requirements of international legislation, such as POPIA and GDPR. POPIA and GDPR require organisations to protect confidential information and to ensure the privacy rights of anyone located in their respective territories (General Data Protection Regulation, 2018; Protection of Personal Information Act (POPIA), 2019). The implementation of App Lock assists Facebook Messenger in securing the confidential information of its users from unauthorised individuals.

The inclusion of the standard IM security and privacy controls and ensuring that the controls meet international standards, assists Facebook Messenger in partially aligning with the requirements of US.H.10. However, to ensure that Facebook Messenger fully aligns with the requirements of **US.H.10 – Compliance of Security and Privacy Controls**, all standard IM security and privacy controls must be made available to all users, regardless of the operating system being used.

8.4.11 US.H.11 – Encryption of Application Session and Information

This usable security heuristic refers to the concern that *an IM application needs to be encrypted to the current industry level of encryption. The encryption level must be made clear to the user. If there is more than one level of encryption available, it must be clear which is active and the user must be guided in terms of how to select the relevant encryption feature. It is crucial for an IM application to encrypt the application session and the storage and transmission of information*.

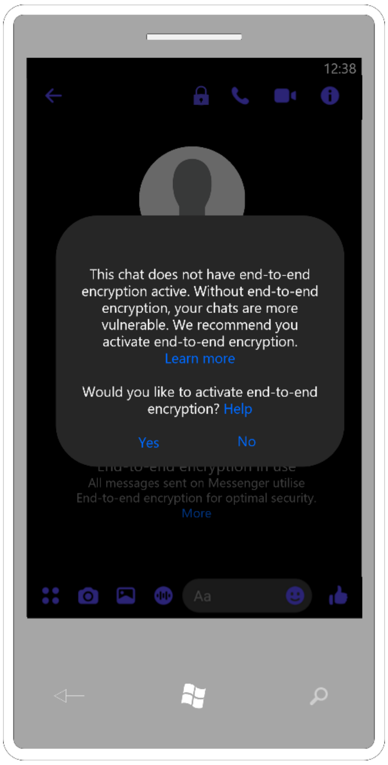
As stated in Chapter 4, Section 4.3.1, Facebook Messenger currently does not utilise the industry standard end-to-end encryption by default. Instead, it utilises the weaker form of encryption, encryption in transit, as their default encryption method. Users are also not actively informed of this, as seen in Figure 8.1 where no notification of Facebook Messenger's encryption method is presented to the user.



Contact One
You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa

End-to-end encryption in use
All messages sent on Messenger utilise End-to-end encryption for optimal security.
More

Figure 8.16 US.H.11 Applied to Facebook Messenger General Chat Screen



This chat does not have end-to-end encryption active. Without end-to-end encryption, your chats are more vulnerable. We recommend you activate end-to-end encryption.
Learn more

Would you like to activate end-to-end encryption? Help

Yes No

All messages sent on Messenger utilise End-to-end encryption for optimal security.
More

Figure 8.17 US.H.11 Applied to Facebook Messenger General Chat Screen End-to-end Encryption Notification

To meet the requirements of US.H.11, the industry standard of encryption needs to be utilised by default and not encryption in transit. Figure 8.16 displays the chat screen which has end-to-end encryption implemented by default. The users must be kept informed of the status of the applications encryption, which can be seen in Figures 8.2 and 8.16. Where this is not the case, guidance must also be presented to users to ensure that they are able to utilise the appropriate form of encryption, as seen in Figure 8.17.

By utilising the industry standard of encryption, as depicted in Figure 8.16, ensuring that users are aware and informed of the encryption being utilised, as depicted in Figures 8.16 and 8.17, and guidance being available to assist users to utilise the encryption, as depicted in Figure 8.17, Facebook

Messenger could adhere to the requirements of **US.H.11 – Encryption of Application Session and Information**.

8.4.12 US.H.12 – Least Privilege by Default

This usable security heuristic refers to the concern that *IM applications need to be developed with the principle of least privilege, which is to ensure that the permissions requested by the application are limited to the minimum permissions required for functionality. IM applications must not request more permissions than those required. Each permission requested must be clearly and concisely explained to the user, to ensure that an informed decision is made by the user. This will also reduce the cognitive load on the user.*

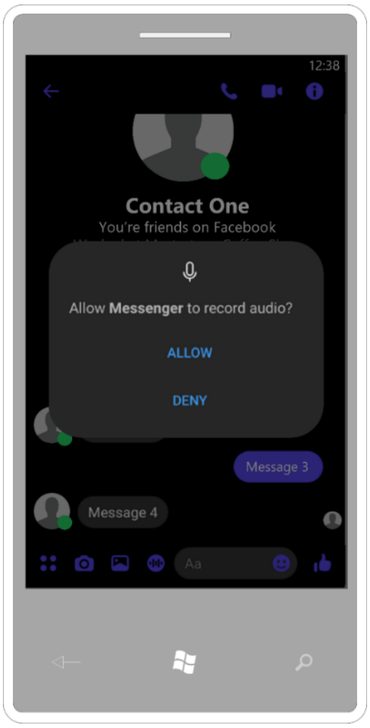
Facebook Messenger currently requests various permissions to ensure the operation of the application. However, Facebook Messenger currently does not limit the permissions requested to only those required for operation and is not transparent about the usage of the granted permissions. The permissions requested by Facebook Messenger, as listed in the Google Play Store (Facebook, 2021d), are displayed in Table 8.1.

Table 8.1 Security Risk Level of Facebook Messenger Permission Requests

Low security risk	**Medium security risk**	**High security risk**
Find accounts on the device	Record audio	Precise location (GPS and network-based)
Read your own contact card	Read your text messages (SMS or MMS)	Read calendar events plus confidential information
	Take pictures and videos	Read phone status and identity
	Connect and disconnect from Wi-Fi	Download files without notification
		Full network access

From the permissions listed in Table 8.1, some of them are deemed to be a high security risk, namely precise location (GPS and network-based), read calendar events plus confidential information, read phone status and identity, download files without notification and full network access. The user's precise location, their calendar schedule and confidential information, and full network access in combination could result in Facebook Messenger knowing the location of the user and what the user is doing. These permissions could expose the user to various IM threats; thus, currently, Facebook Messenger does not align with US.H.12 - least privilege by default.

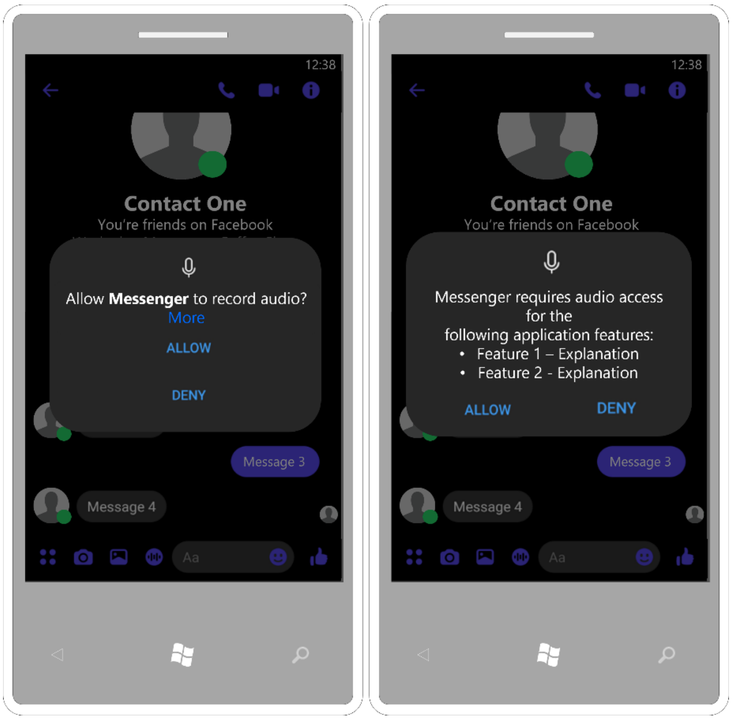
When requesting permissions, Facebook Messenger currently lists them individually, but they do not provide any detail relating to the potential threats associated with each permission requested. The permission requests are not clear and do not inform the user of the potential ramifications when granting the permission. Referring to Figure 8.3, the permission request is ambiguous and does not fully inform the user. However, the edited version of the permission request in Figure 8.4 does provide the user with the option to acquire more information, thereby becoming informed. The trend of ambiguous and uninformative permission requests was noticed in all the permission requests on Facebook Messenger.



Allow **Messenger** to record audio?

Figure 8.18 Current Facebook Messenger Permission Request for Audio

Figure 8.18 presented Facebook Messenger's current permission request to use the device's microphone to record audio. The message is concise but uninformative. It also does not currently provide the user with an opportunity to access further information.



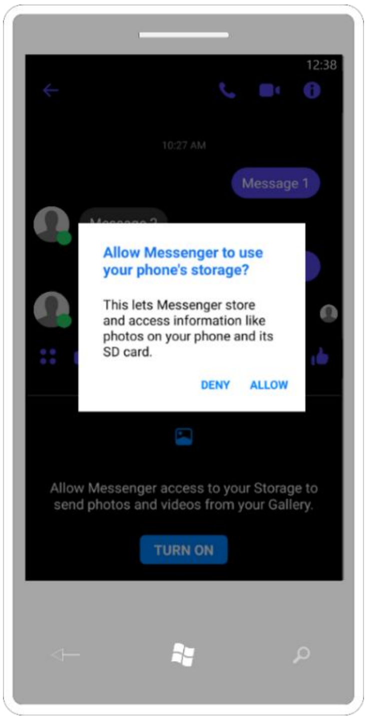
Allow **Messenger** to record audio?

Messenger requires audio access for the
following application features:

- Feature 1 – Explanation
- Feature 2 - Explanation

Figure 8.19 US.H.12 Applied to Facebook Messenger Audio Permission Request

Figure 8.19 displays an edited version of Figure 8.18, following the same style of editing seen previously in Figure 8.4. The inclusion of a link to access more information on the specific permission request on hand enables users to acquire further information and to make informed decisions.



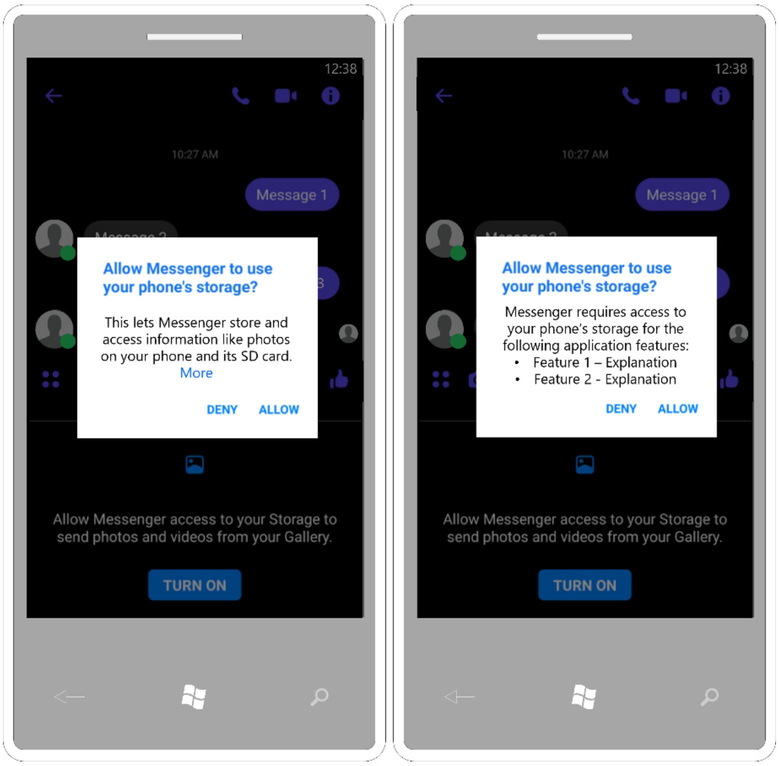
Allow Messenger to use your phone's storage?

This lets Messenger store and access information like photos on your phone and its SD card.

Allow Messenger access to your Storage to send photos and videos from your Gallery.

Figure 8.20 Current Facebook Messenger Permission Request for Storage

Figure 8.20 presents Facebook Messenger’s permission request for the usage of the device’s internal and external storage. This permission request was the same as the requests found in Figures 8.3 and 8.18, concise but uninformative. No opportunity was presented for the user to access further information.



12:38

10:27 AM

Message 1

Allow Messenger to use your phone's storage?

This lets Messenger store and access information like photos on your phone and its SD card. More

Allow Messenger access to your Storage to send photos and videos from your Gallery.

12:38

10:27 AM

Message 1

Allow Messenger to use your phone's storage?

Messenger requires access to your phone's storage for the following application features:

- Feature 1 – Explanation
- Feature 2 - Explanation

Allow Messenger access to your Storage to send photos and videos from your Gallery.

Figure 8.21 US.H.12 Applied to Facebook Messenger Storage Permission Request

Figure 8.21 displays an edited version of Figure 8.20, following the same style of editing seen previously in Figures 8.4 and 8.19. The inclusion of a link to access more information on the specific permission request on hand enables users to acquire further information and to make informed decisions.

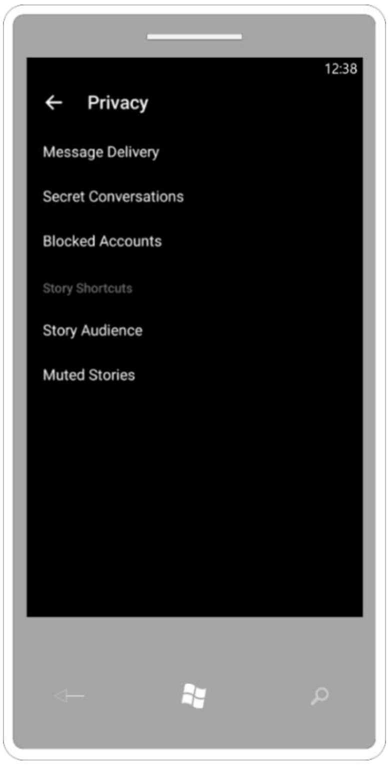
Without providing the users with the opportunity to make informed decisions, IM applications could take advantage of inexperienced users for their own malicious purposes.

Ensuring that Facebook Messenger permissions are only requested for the operation of the IM application, its features, and security features, and that users are made aware of the ramifications of the permission requests, as depicted in Figures 8.19 and 8.21, aligns with the requirements of **US.H.12 – Least Privilege by Default**.

8.4.13 US.H.13 – Secure Access Control

This usable security heuristic refers to the concern that *no unauthorised access must be given to an IM application. The application must secure itself from all forms of attempted access from unauthorised entities*.

From the evaluation conducted, it is evident that Facebook Messenger currently assists users in securing their IM application from unauthorised access by providing multiple IM security features such as end-to-end encryption, encryption in transit, two-factor authentication and App Lock. As previously stated in Chapter 4, Section 4.3.6 App Lock uses the privacy settings of a users' device, such as fingerprint or face authentication to ensure that no unauthorised individuals can access the application. Although Facebook Messenger advertised that App Lock is available for both iOS and Android, it was unavailable on the Android 10 device utilised in the development of the POCP.



12:38

← Privacy

Message Delivery

Secret Conversations

Blocked Accounts

Story Shortcuts

Story Audience

Muted Stories

Figure 8.22 Current Facebook Messenger Privacy Settings for Android 10

The Facebook Messenger IM application version used in the development of this prototype was version 323.1.0.12.119. App Lock is advertised for Android 9 and upward. According to Facebook (2021a), the App Lock security feature should have been available through the privacy section, but it was not. Figure 8.22 displays the options provided to Android 10 users within the privacy settings. App Lock is not currently presented to these users.

To ensure that Facebook Messenger adheres to the requirements of **US.H.13 – Secure Access Control**, Facebook Messenger needs to ensure that App Lock and all other advertised IM security features are available to all users on all forms of devices.

8.4.14 US.H.14 – Flexibility of User Security Expertise

This usable security heuristic refers to the concern that *the security features of an application need to provide plaintext options suitable for users with diverse levels of skills and experience in security.*

From the evaluation conducted, it appears that Facebook Messenger currently does not provide adequate plaintext options for users of diverse skill levels when utilising the various IM security features. As previously identified in Figures 8.3, 8.18, and 8.20, the permission requests are not informative enough for an inexperienced user to understand the potential ramifications associated with granting the permission. As mentioned in Chapter 8, Section 8.2.3.1, users are not made aware of the status of the IM application's encryption. The encryption security feature does not keep users aware of its status nor does it provide users with easily accessible plaintext options to activate end-to-end encryption.

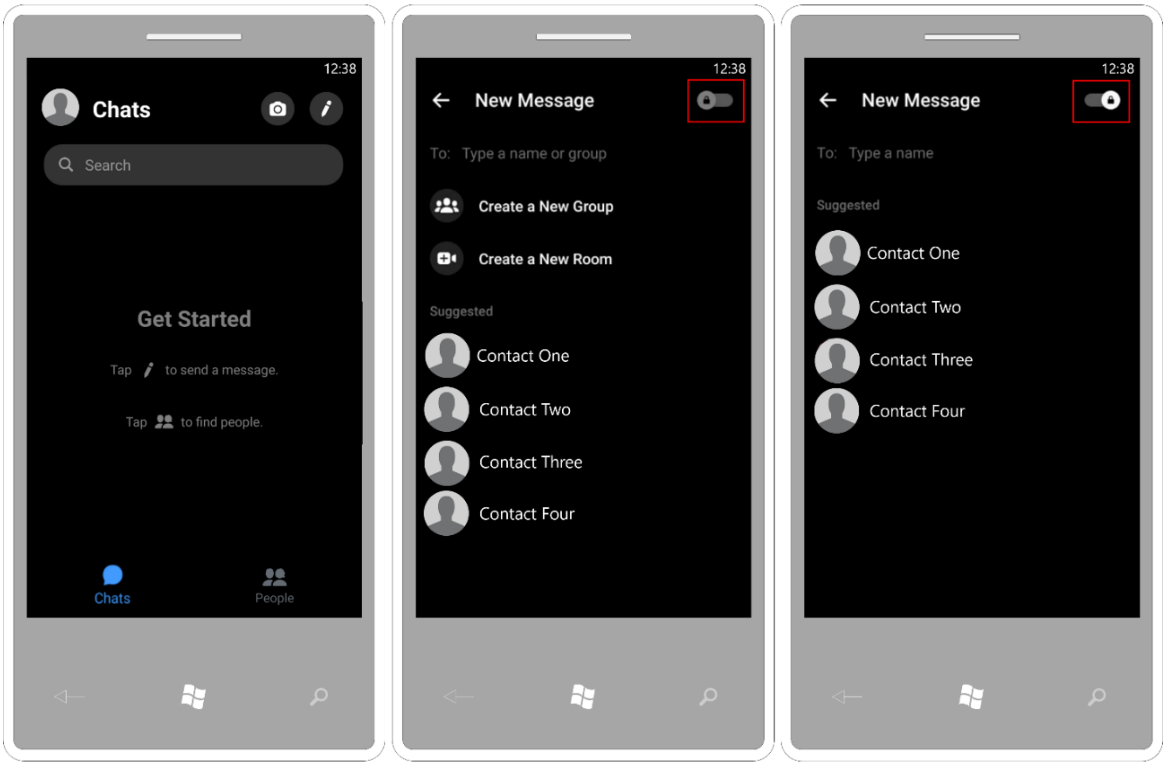


Figure 8.23 Current Facebook Messenger Activation of Secret Conversation

To activate end-to-end encryption, users first had to start a new chat and to activate the small secret conversation switch in the top right corner of their screens, in the red square. This secret conversation switch can be seen in Figure 8.23 in the central- and right-hand screens. The current available option to activate end-to-end encryption is not suitable for inexperienced users, as these users are not explicitly made aware of this IM security feature.

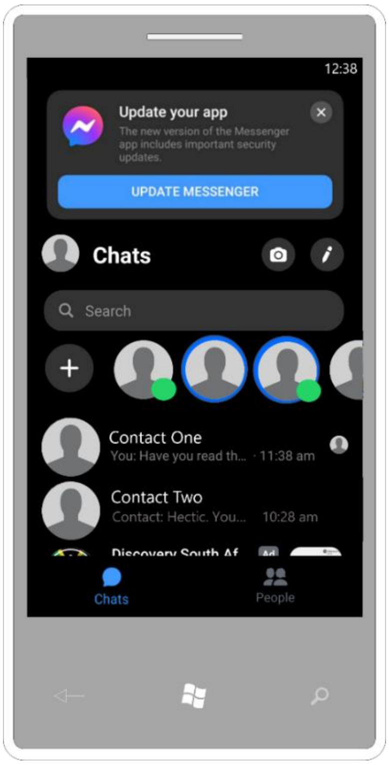
Although experienced users could immediately recognise the on-off toggle located in the red square of Figure 8.23, new and inexperienced users typically need to walk through a step-by-step process to accomplish this. Figure 8.11 depicted the end-to-end encryption process with the relevant required plaintext explanations. This plaintext explanation can also be seen in Figure 8.17. The inclusion of plaintext explanations assists in educating and keeping inexperienced users informed of the potential ramifications of their security decisions.

The inclusion of IM security features with step-by-step guidance, accompanied by easy to access plaintext explanations, could assist Facebook Messenger in adhering to the requirements of **US.H.14 – Flexibility of User Security Expertise**.

8.4.15 US.H.15 – Notification of Security Updates

This usable security heuristic refers to the concern that *to ensure optimal security, IM applications need to alert the user of application updates. To mitigate vulnerabilities of older applications, IM applications need to remain updated.*

From the evaluation conducted, it was evident that Facebook Messenger currently ensures that notifications are distributed when new versions of its IM application become available. The Messenger Help Center states that this feature is only available on Android devices which had the IM application preloaded on the device (Facebook, 2021c).



Update your app

The new version of the Messenger app includes important security updates.

Figure 8.24 Current Facebook Messenger Update Notification

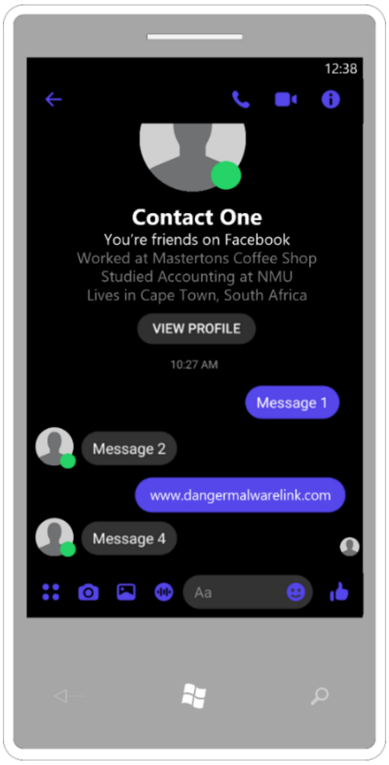
When utilising Facebook Messenger, in order to receive update notifications, including security updates, the Facebook Messenger application needs to be preloaded on the device. During the evaluation of version 323.1.0.12.119, it was already preloaded on the device utilised for the development of the POCP, thus adhering to the preloading requirement to receive update notifications, including security updates.

The existence of this notification of security updates assists in the partial alignment with the requirements of **US.H.15 – Notification of Security Updates**. To ensure full alignment with the requirements of **US.H.15 – Notification of Security Updates**, all Facebook Messenger users, regardless of their platform and whether the application is preloaded or not, need to receive notifications of IM application updates, including security updates.

8.4.16 US.H.16 – Secure Malware Controls

This usable security heuristic refers to the concern that *IM applications need to implement controls to detect, prevent and recover from malware. Such applications should also inform and keep users aware of the situation.*

Facebook Messenger, like any other IM application, could potentially be the target of malware. Malicious messages could be sent across the IM application, in an attempt to spread them to as many users as possible. The success of malware tends to rely on the ability to trick or fool users into unsafe behaviours. However, during the evaluation conducted, it appeared that Facebook Messenger currently lacks malware controls to keep users informed of the potential risk. Facebook Messenger currently cannot detect or prevent the spread of potentially malicious messages.



12:38

Contact One

You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa

10:27 AM

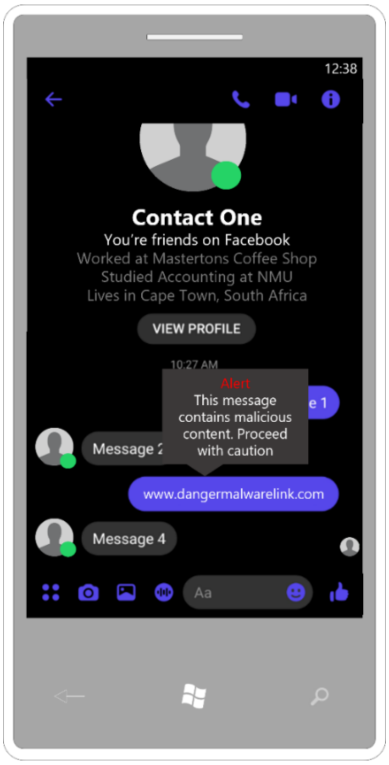
Message 1

Message 2

www.dangermalwarelink.com

Message 4

Figure 8.25 Current Facebook Messenger General Chat Screen with a Malicious Message



12:38

Contact One

You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa

10:27 AM

Alert
This message contains malicious content. Proceed with caution

e 1

Message 2

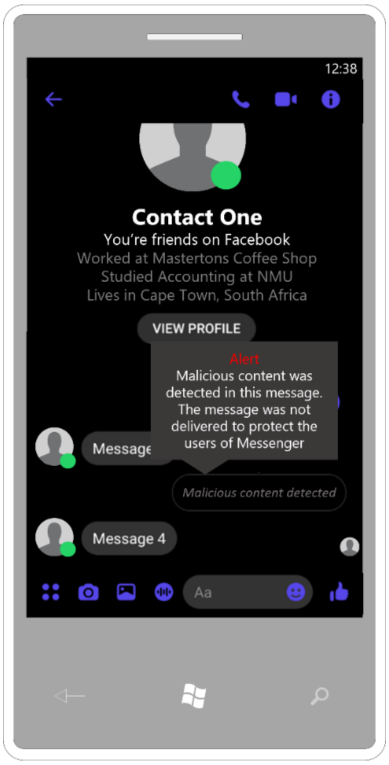
www.dangermalwarelink.com

Message 4

Figure 8.26 US.H.16 Applied to Facebook Messenger General Chat Screen for Detection of a Malicious Message

Figure 8.25 displays a chat in which a malicious link was sent. This malicious link was not detected, and users did not receive any alert to inform them of the potential dangers associated with this link. Inexperienced users would blindly trust their friends and colleagues which could result in their clicking the link and potentially compromising their Facebook Messenger application.

To secure their users from the threat of malware, Facebook Messenger needs IM security controls to detect, prevent, and recover from malicious messages. Figure 8.26 displays an edited version of Figure 8.25. In this edit, the malicious link is detected by Facebook Messenger and an alert message is presented to the user. The alert message warns the user of the danger within the detected link, thereby assisting users to recover from potentially sending or clicking on a malicious message.



You're friends on Facebook
Worked at Mastertons Coffee Shop
Studied Accounting at NMU
Lives in Cape Town, South Africa

Alert
Malicious content was detected in this message. The message was not delivered to protect the users of Messenger

Malicious content detected

Figure 8.27 US.H.16 Applied to Facebook Messenger General Chat Screen for Prevention of a Malicious Message

In the case of the message being prevented and not transmitted, the message would not be delivered, as seen in Figure 8.27. Instead of the malicious message being delivered to the user, Facebook Messenger presents the user with a status update message, which is accompanied by an alert message. The alert message informs the user of the reason for the status message appearing and the reasons for not delivering the message. This alert, similar to the alert found in Figure 8.26, forms part of assisting users to recover from sending or receiving a malicious message.

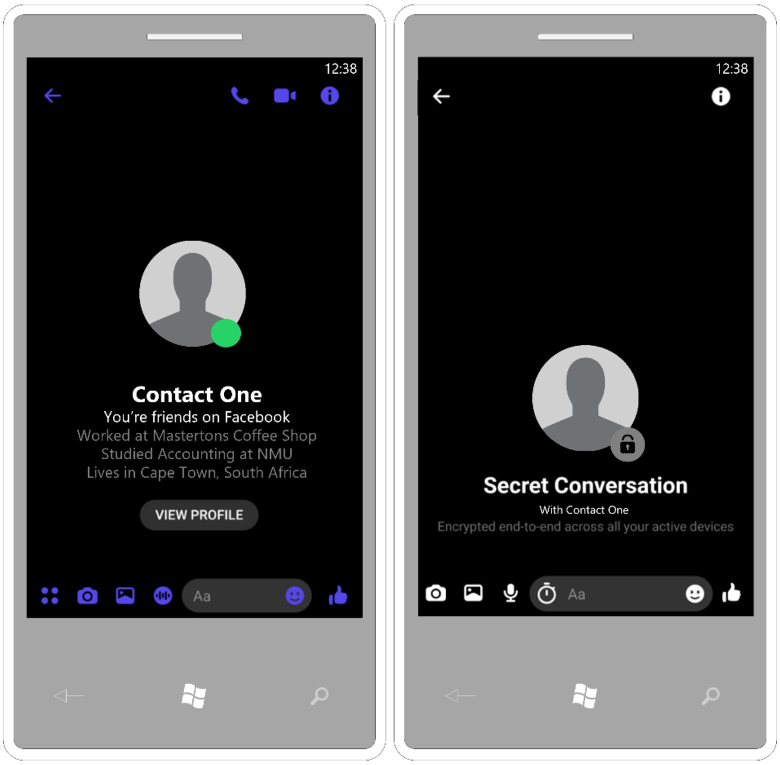
In the scenario presented in Figure 8.26, users could still potentially click and spread the malicious link as the link is still delivered. However, in the Figure 8.27 scenario, users do not receive the link, which prevents users from clicking and spreading the link.

The inclusion of detecting, preventing, and recovering from malicious messages, as depicted in Figures 8.26 and 8.27, would ensure that Facebook Messenger users are informed and aware of malicious messages, in a timely manner, thus adhering to heuristic **US.H.16 – Secure Malware Controls.**

8.4.17 US.H.17 – Secure by default

This usable security heuristic refers to the concern that *IM applications need to ensure that the optimal security settings are active by default. This will reduce the chances of IM applications being utilised with weaker security.*

On conducting the evaluation, it was evident that Facebook Messenger was not secure by default. As previously stated in Chapter 4, Section 4.3.1, Facebook Messenger currently uses two forms of encryption, but the weaker encryption in transit was active by default. This is an indication that Facebook Messenger does not operate with optimal security by default.



12:38

Contact One

You're friends on Facebook

Worked at Mastertons Coffee Shop

Studied Accounting at NMU

Lives in Cape Town, South Africa

12:38

Secret Conversation

With Contact One

Encrypted end-to-end across all your active devices

Figure 8.28 Current Facebook Messenger Regular Conversation in Comparison to Secret Conversation

Figure 8.28 presents a comparison of the regular and secret conversations. The secret conversation (right-hand screen) notifies users that the stronger form of encryption, end-to-end encryption, is utilised, whereas the regular conversation (left-hand screen) did not inform users about any forms of encryption being utilised.

As seen in Figure 8.16, end-to-end encryption needs to be activated and users need to be made aware of this.

The utilisation of optimal security settings would ensure that Facebook Messenger users are secured by default, thus adhering to heuristic **US.H.17 – Secure by default.**

8.5 Conclusion

The finalised set of usable security heuristics, presented in Chapter 7, Section 7.5, Table 7.6, were utilised in the development of a POCP. As stated in Chapter 8, Section 8.2, the aim of the POCP was to demonstrate the applicability of the proposed set of usable security heuristics to a typical instant messaging application. The typical IM application selected for this POCP was Facebook Messenger. This POCP demonstrated the applicability of the proposed set of usable security heuristics to Facebook Messenger. The discussion and explanations of the POCP displayed the effectiveness of the set of usable security heuristics for instant messaging application development. Furthermore, this met the requirements of SRO4. In addition, the POCP further validated the proposed set of usable security heuristics.

The following chapter brings the research to a close by summarising the research findings from each chapter. In addition, the chapter illustrates how each of the research objectives was fulfilled, as well as arguing for this study's contribution, and making recommendations for future research.

Chapter 9 – Conclusion

9.1 Introduction

In Chapter 8, the proof-of-concept prototype was presented and discussed based on the final proposed set of usable security heuristics for instant messaging application development.

The aim of this chapter is to conclude the study by summarising each chapter and motivating how each of the research objectives from Chapter 1 were attained.

The chapter structure is as follows: Section 9.2 provides a summary of each chapter, while Section 9.3 argues how each of the established research objectives were accomplished. Section 9.4 documents the contributions of this research, while Section 9.5 presents the limitations of the research. Section 9.6 highlights the potential for future research based on this study and finally, Section 9.7 concludes this chapter.

9.2 Chapter Summaries

This section presents a brief summary of each chapter as a reflection of how each contributes to this study.

Chapter 1 – Introduction

The purpose of this chapter was to document the problem area, problem statement, research objectives, and research process. In so doing, the concept of usable security heuristics is introduced and the importance of usability and user experience is highlighted. IM applications are also introduced with a focus on the potential threats to the security of IM applications and their users. In addition, it was noted that most IM applications expect users to do more than they are capable of with regard to their security. Furthermore, it states that currently a generally accepted set of usable security heuristics for IM application development does not exist. The research objectives for this study are also discussed in this chapter, as well as the research process that was followed during this study.

Chapter 2 – Information Security

The purpose of Chapter 2 was to assist in addressing SRO1 - *To determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users.* In so doing, it provides a high-level background to information security and the importance of maintaining information security in general. Furthermore, several threats to application security were discussed and assessed. In addition, the application vulnerabilities were highlighted which typically lead to the identified threats. An increase in threats and vulnerabilities results in an increase in risk to an information asset. The application security threats identified in this chapter are used in Chapter 3 to argue their applicability to IM applications.

Chapter 3 – Instant Messaging

As with Chapter 2, the purpose of Chapter 3 was to further assist in addressing SRO1 - *To determine common instant messaging security risks, with a specific focus on threats vulnerabilities and controls, and their potential impact on users.* However, Chapter 3 is more focused, providing an overview of instant messaging, what it is, and how it works. In addition, common IM applications are discussed, and their general features are compared. The common IM applications discussed include Facebook Messenger, Snapchat, Telegram, Viber, WeChat, and WhatsApp. The application security threats from Chapter 2 are used in this chapter to argue their applicability to IM applications. In addition, the impact of the identified application security threats is also documented. The severity of the potential impact of the identified application security threats are also assessed based on the risk they pose to the information assets of IM users and organisations. The most relevant application security threats identified are then argued to be the most common instant messaging security threats. Furthermore, the IM vulnerabilities which could be exploited by the IM security threats and their potential impact on IM information assets are documented.

Chapter 4 – Instant Messaging Security

As with Chapters 2 and 3, the purpose of Chapter 4 was to further assist in addressing SRO1 - *To determine common instant messaging security risks, with a specific focus on threats vulnerabilities and controls, and their potential impact on users.* In so doing, Chapter 4 discusses several IM security and privacy controls implemented in the main IM applications identified, namely Facebook Messenger, Snapchat, Telegram, Viber, WeChat, and WhatsApp. The IM security and privacy controls highlighted are intended to mitigate the vulnerabilities typically found in IM applications and potentially the IM security threats identified in Chapter 3. This would assist in securing IM information assets by reducing potential IM risk. The combination of the findings across Chapters 2, 3, and 4 therefore achieved the requirements of SRO1.

Chapter 5 –Security and Usability Heuristics, Guidelines, Standards and Best Practices

The purpose of Chapter 5 was to address SRO2 - *To identify and analyse existing security and usability heuristics, guidelines, standards and best practices for mobile application development*. In so doing, it investigates current existing security and usability heuristics, guidelines, standards, and best practices as they relate to one or more of the following categories: instant messaging, mobile application development, security, usability and usable security. In this chapter, the content analysis process was documented according to four main steps, namely planning, data collection, data analysis and reporting of results. The detailed results of the content analysis and the identified security and usability heuristics, guidelines, standards, and best practices, which could potentially alleviate the identified instant messaging security threats, from Chapter 4, are discussed. The identified security and usability heuristics, guidelines, standards, and best practices deemed most relevant to this study are provided in four appendices as follows: Appendix A contains the identified heuristics; Appendix B contains the identified guidelines; Appendix C contains the identified standards; and Appendix D contains the identified best practices.

Chapter 6 – Proposed set of Usable Security Heuristics for Instant Messaging Application Development

The primary purpose of Chapter 6 was to address SRO3 - *To map the identified security and usability heuristics, guidelines, standards, and best practices to instant messaging application development*. In so doing, it also partially addresses the PRO - *To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*. The proposed set of usable security heuristics was based on the literature studied (Chapters 2, 3, and 4) and the content analysed (Chapter 5). Furthermore, the heuristic development process followed four main steps, namely:

- **Step 1:** Determine the most prominent instant messaging (IM) application threats and the related IM security controls and features. Utilise the new set of usable security heuristics to evaluate these security controls and features.

- **Step 2:** Identify existing security and usability heuristics, guidelines, standards, and best practices in order to determine how they can assist in defining a set of usable security heuristics for IM applications.
- **Step 3:** Define a set of usable security heuristics for IM applications following a rigorous approach which is clearly stated and easy to understand.
- **Step 4:** Validate the proposed set of usable security heuristics to its efficacy, utility, and quality, and its applicability to IM applications.

Furthermore, the adaption of the heuristics, guidelines, standards, and best practices identified in Chapter 5, Section 5.3.4, consist of three stages, namely:

Stage 1: Adapt;

Stage 2: Analyse; and

Stage 3: Revise and Finalise.

The preliminary set of usable security heuristics for instant messaging application development, containing 17 usable security heuristics, are mapped against the identified instant messaging threats, from Chapter 3, and against the identified IM security and privacy controls, from Chapter 4.

Chapter 7 – Validation of the Proposed set of Usable Security Heuristics

The purpose of Chapter 7 is to further address the PRO - *To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*. In so doing, it presents the validation, in the form of an expert review, of the set of usable security heuristics for instant messaging application development proposed in Chapter 6. The expert review consists of five experts and was conducted in the form of a questionnaire, which contained the following five sections:

Biographical information section,

Security section,

Usability section,

Mobile application development section, and

General section.

In this chapter, the feedback from the experts is analysed and reported on, together with the recommend changes. The feedback from the expert review confirms that the set of usable security heuristics meets the requirements of efficacy, utility, and quality, thereby validating the proposed set of usable security heuristics for instant messaging application development.

Chapter 8 – Proof-of-Concept Prototype

As with Chapter 7, the purpose of Chapter 8 is to further address the PRO - *To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications*. In addition, Chapter 7 addresses SRO4 - *To develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a* *typical instant messaging application*. This is achieved by further validating the proposed set of usable security heuristics in the form of a proof-of-concept prototype (POCP) based on Facebook Messenger. An overview of the prototype is documented, together with the reasoning for the selection of Facebook Messenger as the IM application utilised in the POCP. The POCP demonstrates the applicability of the proposed set of usable security heuristics to a typical instant messaging application. Each usable security heuristic is discussed individually in the context of Facebook Messenger.

9.3 Accomplishment of Research Objectives

This section discusses how each of the research objectives were achieved.

9.3.1 Accomplishment of the Primary Research Objective

The primary objective of this study was *to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security mechanisms implemented in these applications.*

This objective was attained through the accomplishment of various sub-objectives. In order to meet the primary research objective, the researcher was required to determine common instant messaging security risks, with a specific focus on threats, vulnerabilities, and controls, and their potential impact on users (SRO1); to identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development (SRO2); to map the identified security and usability heuristics, guidelines, standards, and best practices to instant messaging application development (SRO3); and to develop a prototype to demonstrate the applicability of the finalised set of usable security heuristics to a typical instant messaging application (SRO4).

Table 9.1 contains the research objectives, the research methods utilised to attain each objective, and the specific sections where each objective was addressed.

Table 9.1 Research Methods with Associated Research Objectives and their Relevant Sections

	Research objective	Research method	Relevant section
PRO	To create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications.	Critical reasoning/ argumentation and Expert review	Chapter 6, Sections 6.3 and 6.4. Chapter 7, Sections 7.2 and 7.3.
SRO1	To determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users.	Literature review	Chapter 2, Sections 2.3 and 2.4. Chapter 3, Section 3.5. Chapter 4, Section 4.3.
SRO2	To identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development.	Literature review and Content analysis	Chapter 5, Section 5.3. Appendices A, B, C, and D.
SRO3	To map the identified security and usability heuristics, guidelines,	Critical reasoning/ argumentation	Chapter 6, Sections 6.3, 6.4, and 6.5.

	standards, and best practices to instant messaging application development.		
SRO4	To develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application.	Critical reasoning/ argumentation and Prototype	Chapter 8, Section 8.4.

Collectively, the secondary research objectives address the primary research objective. The primary research objective, in turn, comprehensively addresses the problem statement by proposing a set of usable security heuristics for instant messaging application development. Therefore, by meeting the secondary research objectives of this study, the primary research objective*, 'to create a set of usable security heuristics to assist developers of instant messaging applications to consider the usability of the security features implemented in these applications'*, was achieved.

The proposed set of usable security heuristics for instant messaging application development is presented in Chapter 6 and validated through an expert review, as presented in Chapter 7. The set of usable security heuristics for instant messaging application development was evaluated by experts in the fields of usability, security, and mobile application development. The overview of the expert review is presented in Chapter 7, Section 7.2, while the expert review instrument design is located in Chapter 7, Section 7.3. The results of the expert review are documented and processed in Chapter 7, Section 7.4. Based on the results of the expert review, any changes implemented to the preliminary set of usable security heuristics are highlighted and the finalised set of proposed usable security heuristics for instant messaging application development are presented in Chapter 7, Section 7.5. Furthermore, the applicability of the proposed set of set of usable security heuristics for instant messaging application development was assessed through a proof-of-concept prototype, as presented in Chapter 8.

9.3.2 Accomplishment of the Secondary Research Objectives

In order to address the PRO, the following SROs were established and attained within this study.

SRO1 – *To determine common instant messaging security risks, with a specific focus on threats, vulnerabilities and controls, and their potential impact on users*. This objective was accomplished through a literature review, as documented in Chapter 2, focused on information security, and identifying the information security threats and vulnerabilities which are relevant to application security. These relevant information security threats became known as application security threats. Chapter 3 highlighted IM applications, what they are, and how they function. It is important to note that, in this chapter the application security threats were assessed against IM application security, thereby identifying the most common IM application security threats. Chapter 4 introduced the typical IM security and privacy features which assist in mitigating the identified IM application security threats. Chapters 2, 3, and 4 assisted in addressing SRO1 and provided the research output of IM security threats and their potential impact on users.

SRO2 – *To identify and analyse existing security and usability heuristics, guidelines, standards, and best practices for mobile application development*. This objective was accomplished via a literature review and content analysis, as presented in Chapter 5. The literature review helped in identifying and defining existing security and usability heuristics, guidelines, standards, and best practices in Chapter

5, Section 5.2. The content analysis process, which consisted of four main steps (planning, data collection, data analysis and reporting of results), was discussed in Chapter 5, Section 5.3. The detailed results of the content analysis were assessed to identify the security and usability heuristics, guidelines, standards, and best practices deemed most relevant to instant messaging application development. These heuristics, guidelines, standards, and best practices are provided in Appendices A, B, C, and D, respectively. The results of the content analysis addressed SRO2 and provided the research output of existing security and usability heuristics, guidelines, standards, and best practices.

SRO3 – *To map the identified security and usability heuristics, guidelines, standards, and best practices to instant messaging application development.* This objective was accomplished using critical reasoning/argumentation, as presented in Chapter 6, which used the results of the content analysis (Chapter 5) to propose a set of usable security heuristics for instant messaging application development. The heuristic development process is documented in Chapter 6, Section 6.2, which includes the four-step process utilised to develop the set of usable security heuristics. To provide further rigour, the proposed set of usable security heuristics are mapped against the previously identified instant messaging security threats in Chapter 6, Section 6.5.1, and instant messaging security and privacy features in Chapter 6, Section 6.5.2. Chapter 6 addressed SRO3 and provided the research output of a preliminary set of usable security heuristics for instant messaging application development.

SRO4 – *To develop a prototype to demonstrate the applicability of the proposed usable security heuristics to a typical instant messaging application*. This objective was accomplished via a proof-of-concept prototype, as presented in Chapter 8. This chapter highlights the applicability of the proposed set of usable security heuristics to Facebook Messenger, a typical instant messaging application. The overview of the proof-of-concept prototype is presented in Chapter 8, Section 8.2, while the selection of the typical IM application to be used in the development of the proof-of-concept prototype occurs in Chapter 8, Section 8.3. The application of the set of usable security heuristics to the typical IM application is presented in Chapter 8, Section 8.4. The POCP addressed each individual usable security heuristic and its related impact on the typical IM application, in this instance Facebook Messenger. Various screenshots were taken of several features in Facebook Messenger (updated last on 02 August 2021), highlighting the usable security gaps within the current application. These screenshots were then modified to demonstrate the applicability of each of the proposed usable security heuristic. The demonstration of the proof-of-concept prototype addressed SRO4.

9.4 Summary of Contributions

Typically, in software development, security is an afterthought. It is implemented late in the development process without considering the impact on the user. The primary contribution of this study is therefore a set of usable security heuristics for instant messaging application development to assist IM application developers in considering the usability of the security features they implement. Implementing the proposed set of usable security heuristics during the development process would assist developers in prioritising security and providing users with both a secure and a usable IM application. This set of usable security heuristics adds a valuable contribution to the research in this area.

Furthermore, the detailed content analysis process followed, documented in Chapter 5, for the identification of existing security and usability heuristics, guidelines, standards, and best practices relevant to mobile application development, could potentially assist future researchers conducting similar content analyses. Future researchers could replicate the content analysis or utilise Appendices

A, B, C, and D, which contain the most relevant security and usability heuristics, guidelines, standards, and best practices, respectively. This detailed content analysis therefore adds a further contribution to this research area.

The proof-of-concept prototype, presented in Chapter 8, provides a visualisation of the applicability of the proposed set of usable security heuristics applied to a typical IM application, in this case Facebook Messenger. Presenting a visualisation of the set of usable security heuristics could assist IM application developers in improving their understanding of how to implement the set of usable security heuristics, and in visualising the improvements to the usability of the IM security features. The POCP therefore adds a further contribution to this research area.

9.5 Limitations

This study did not address all possible security threats and vulnerabilities relating to IM applications, but only those deemed to be most relevant. Furthermore, owing to time constraints, this research study only demonstrated the applicability of the proposed set of usable security heuristics to Facebook Messenger. Future research could consider creating comparable prototypes for a variety of IM applications, in order to demonstrate broader applicability.

9.6 Future Research

As the demand for usable and secure applications is increasing, this proposed set of usable security heuristics for instant messaging application development could be further generalised to meet the demand for secure yet usable applications. This could involve modifying the proposed set of usable security heuristics for instant messaging application development to fit the needs of other IM applications, social media applications, or other web and mobile applications. Another potential area for future research could be developing a guideline document to accompany the proposed set of usable security heuristics for instant messaging application development. The guideline document could contain instructions, code snippets and examples of how and where to implement each heuristic when developing an IM application.

9.7 Conclusion

Owing to the increasing demand of both usable and secure applications and the wide range of skills among users, there are many barriers to the implementation of security controls and features within IM applications. The proposed set of usable security heuristics for instant messaging application development provides a standard approach to aid developers in considering the usability of the security controls and features they implement. This would provide users with the opportunity to remain informed and aware of their security status when utilising an IM application. In addition, ensuring that users are aware of their security and privacy status assists in mitigating information security-related risks.

As a result, it can be concluded that this research study proposes a valuable set of usable security heuristics for instant messaging application development to aid IM application developers to consider the usability of the security controls and features they implement.

References

Abed, A. M., & Salah, M. (2019). A Review of Instant Messaging. *SIMCA ICI2TM-2019*, 116–118. http://ici2tm.sinhgad.edu/pcproc/ICI2TM2019_P/data/IC19069.pdf

Abu-Salma, R., Krol, K., Parkin, S., Koh, V., Kwan, K., Mahboob, J., Traboulsi, Z., & Sasse, M. A. (2017). *The Security Blanket of the Chat World: An Analytic Evaluation and a User Study of Telegram*. https://doi.org/10.14722/eurousec.2017.23006

Aggarwal, P. K., Grover, P. S., & Ahuja, L. (2018). Security Aspect in Instant Mobile Messaging Applications. *2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS)*, 1–5. https://doi.org/10.1109/RAETCS.2018.8443844

Agham, V. (2016). Unified Threat Management. *International Research Journal of Engineering and Technology (IRJET)*, *03*(04), 32–36. http://search.proquest.com.library.capella.edu/docview/1282108711?accountid=27965

Ahmad, N., Rextin, A., & Kulsoom, U. E. (2018). Perspectives on usability guidelines for smartphone applications: An empirical investigation and systematic literature review. *Information and Software Technology*, *94*(October 2017), 130–149. https://doi.org/10.1016/j.infsof.2017.10.005

Äijälä, T. (2018). *CISSP certification-accreditation value for employees and recruiters*. https://www.theseus.fi/bitstream/handle/10024/148953/TA_CISSP_master_thesis_final_2018-05-28.pdf?sequence=1&isAllowed=y

Airehrour, D., Nair, N. V., & Madanian, S. (2018). Social Engineering Attacks and Countermeasures in the New Zealand Banking System: Advancing a User-Reflective Mitigation Model. *Information*, *9*(5), 110. https://doi.org/10.3390/info9050110

Ajit Kumar, N., Krishna, K. T. H., & Manjula, R. (2016). Challenges and Best Practices in Mobile Application Development. *Imperial Journal of Interdisciplinary Research*, *2*(12), 2454–1362. https://www.onlinejournal.in/IJIRV2I12/253.pdf

Alazab, M., & Broadhurst, R. (2017). An Analysis of the Nature of Spam as Cybercrime. *Cyber-Physical Security*, 251–266. https://doi.org/10.1007/978-3-319-32824-9

Albladi, S., & Weir, G. R. S. (2016). Vulnerability to social engineering in social networks: A proposed user-centric framework. *2016 IEEE International Conference on Cybercrime and Computer Forensic, ICCCF 2016*, 2–7. https://doi.org/10.1109/ICCCF.2016.7740435

Aldawood, H., & Skinner, G. (2019a). A Taxonomy for Social Engineering Attacks via Personal Devices. *International Journal of Computer Applications*, *178*(50), 19–26. https://doi.org/10.5120/ijca2019919411

Aldawood, H., & Skinner, G. (2019b). Educating and Raising Awareness on Cyber Security Social Engineering: A Literature Review. *Proceedings of 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering, TALE 2018*, *December*, 62–68. https://doi.org/10.1109/TALE.2018.8615162

Aldayel, A., & Alnafjan, K. (2017). Challenges and Best Practices for Mobile Application Development. *Proceedings of the International Conference on Compute and Data Analysis - ICCDA '17*, *Part F1302*, 41–48. https://doi.org/10.1145/3093241.3093245

Alenezi, M., & Almuairfi, S. (2019). Security Risks in the Software Development Lifecycle. *International Journal of Recent Technology and Engineering*, *8*(3), 7048–7055. https://doi.org/10.35940/ijrte.C5374.098319

Ali, R. M., & Alsaad, S. N. (2020). Instant messaging security and privacy secure instant messenger design. *IOP Conference Series: Materials Science and Engineering*, *881*(1), 012117. https://doi.org/10.1088/1757-899X/881/1/012117

Alkhudhayr, F., Alfarraj, S., Aljameeli, B., & Elkhdiri, S. (2019). Information Security: A Review of Information Security Issues and Techniques. *2nd International Conference on Computer Applications and Information Security, ICCAIS 2019*. https://doi.org/10.1109/CAIS.2019.8769504

Almeida, T. A., Silva, T. P., Santos, I., & Gómez Hidalgo, J. M. (2016). Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering. *Knowledge-Based Systems*, *108*, 25–32. https://doi.org/10.1016/j.knosys.2016.05.001

Alwazzeh, M., Karaman, S., & Shamma, M. N. (2020). Man in The Middle Attacks Against SSL/TLS: Mitigation and Defeat. *Journal of Cyber Security and Mobility*, *9*, 449–468. https://doi.org/10.13052/jcsm2245-1439.933

Ammirato, S., Felicetti, A. M., Della Gala, M., Aramo-Immonen, H., Jussila, J. J., & Kärkkäinen, H. (2019). The use of social media for knowledge acquisition and dissemination in B2B companies: an empirical study of Finnish technology industries. *Knowledge Management Research and Practice*, *17*(1), 52–69. https://doi.org/10.1080/14778238.2018.1541779

Amnesty International. (2016). *How private are your favourite messaging apps?* Amnesty.Org. https://www.amnesty.org/en/latest/campaigns/2016/10/which-messaging-apps-best-protect-your-privacy/

Andriotis, P., Sasse, M. A., & Stringhini, G. (2017). Permissions snapshots: Assessing users' adaptation to the Android runtime permission model. *8th IEEE International Workshop on Information Forensics and Security, WIFS*. https://doi.org/10.1109/WIFS.2016.7823922

Antoniou, G. S. (2018). A Framework for the Governance of Information Security: Can it be Used in an Organisation. *Conference Proceedings - IEEE SOUTHEASTCON*, *2018-April*, 1–30. https://doi.org/10.1109/SECON.2018.8479032

Apple Inc. (2020). *Apple Platform Security*. 157. https://manuals.info.apple.com/MANUALS/1000/MA1902/en_US/apple-platform-security-guide.pdf

Ashktorab, Z. (2018). *"The Continuum of Harm" Taxonomy of Cyberbullying Mitigation and Prevention*. 211–227. https://doi.org/10.1007/978-3-319-78583-7_9

Australian Government. (2016). *Australia's Cyber Security Strategy: Enabling innovation , growth & prosperity*. https://cybersecuritystrategy.homeaffairs.gov.au/AssetLibrary/dist/assets/images/PMC-Cyber-Strategy.pdf

Awan, J., & Memon, S. (2016). Threats of cyber security and challenges for Pakistan. *Proceedings of the 11th International Conference on Cyber Warfare and Security, ICCWS 2016*, 425–430. https://d1wqtxts1xzle7.cloudfront.net/55848375/2016-Threats_of_Cyber_Security_and_Challenges_for_Pakistan.pdf?1519108472=&response-content-disposition=inline%3B+filename%3DThreats_of_Cyber_Security_and_Challenges.pdf&Expires=1594811295&Signature=SWN9x333k

Bagheri, H., Kang, E., Malek, S., & Jackson, D. (2018). A formal approach for detection of security flaws in the android permission system. *Formal Aspects of Computing*, *30*(5), 525–544. https://doi.org/10.1007/s00165-017-0445-z

Bagheri, H., Sadeghi, A., Garcia, J., & Malek, S. (2015). COVERT: Compositional Analysis of Android Inter-App Permission Leakage. *IEEE Transactions on Software Engineering*, *41*(9), 866–886. https://doi.org/10.1109/TSE.2015.2419611

Bajenaru, L., Marinescu, I. A., & Dobre, C. (2018). Different approaches to modeling user experience in the context of mobile application challenges. *Proceedings of the IE 2018 International Conference*, *May*. https://www.researchgate.net/profile/Lidia_Bajenaru/publication/325294880_Different_approaches_to_modeling_user_experience_in_the_context_of_mobile_application_challenges/links/5bb208db45851574f7f3ae72/Different-approaches-to-modeling-user-experience-in-t

Baldikov, N. (2020). *Is Your Free Instant Messenger Really Free?* Brosix.Com. https://www.brosix.com/blog/free-instant-messenger/

Bandi, A. (2016). Developers' perspectives on architecture violations: A survey. *25th International Conference on Software Engineering and Data Engineering, SEDE 2016*, 91–96. https://www.nwmissouri.edu/csis/msacs/PDF/Publications/Developers Perspectives on Architecture.pdf

Baror, S. O., & Venter, H. (2019). A taxonomy for cybercrime attack in the public cloud. *14th International Conference on Cyber Warfare and Security, ICCWS 2019*, *September*, 505–515. https://www.researchgate.net/profile/Stacey_Baror/publication/335927227_A_Taxonomy_for_Cybercrime_Attack_in_the_Public_Cloud/links/5d8453d1458515cbd19f4c9e/A-Taxonomy-for-Cybercrime-Attack-in-the-Public-Cloud.pdf

Barry, B., & Tom, F. M. (2009). Instant Messaging: Standards, Protocols, Application and Research Directions. *Internet Policies and Issues*, *7*(July 2010), 17–25. https://www.researchgate.net/profile/Bazara_Barry/publication/280307922_Instant_Messaging_Standards_Protocols_Applications_and_Research_Directions/links/55b1093008ae9289a084a94e/Instant-Messaging-Standards-Protocols-Applications-and-Research-Directions.pd

Bauer, L., Bravo-Lillo, C., Cranor, L. F., & Fragkaki, E. (2013). *Warning Design Guidelines*. *CMU-CyLab-13-002*. http://www.cylab.cmu.edu/research/techreports/2013/tr_cylab13002.html

Bauernfreund, M. (2019). *Communities vs group chats – what's best for you*. Viber.Com. https://www.viber.com/en/blog/2019-11-04/communities-vs-group-chats-whats-best-for-you/

Bayer, J. B., Ellison, N. B., Schoenebeck, S. Y., & Falk, E. B. (2016). Sharing the small moments: ephemeral social interaction on Snapchat. *Information Communication and Society*, *19*(7), 956–977. https://doi.org/10.1080/1369118X.2015.1084349

Beckers, K., & Pape, S. (2016). A Serious Game for Eliciting Social Engineering Security Requirements. *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference, RE 2016*, 16–25. https://doi.org/10.1109/RE.2016.39

Bengtsson, M. (2016). How to plan and perform a qualitative study using content analysis. *NursingPlus Open*, *2*, 8–14. https://doi.org/10.1016/j.npls.2016.01.001

Bevan, N., Carter, J., Earthy, J., Geis, T., & Harker, S. (2016). New ISO Standards for Usability, Usability Reports and Usability Measures. *International Conference on Human-Computer Interaction (Pp. 268-278). Springer, Cham.*, *9731*(July), 268–278. https://doi.org/10.1007/978-3-319-39510-4

Bhardwaj, A. (2020). *How to Recover Deleted Telegram Messages [updated 2020]*. Teknologya.Com. https://teknologya.com/recover-deleted-telegram-messages/

Bhatia, S. (2015). The Power of the Representativeness Heuristic. *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, 232–238. https://pdfs.semanticscholar.org/049d/9ef520c23fcc02f852f36b566916fe1bdb3f.pdf

Bhatt, A. J., Gupta, C., & Mittal, S. (2019). iShield: A Framework for Preserving Privacy of iOS App User. *Journal of Cyber Security and Mobility*, *8*(4), 493–536. https://doi.org/10.13052/jcsm2245-1439.845

Bhavani, R., Jayashree, R., Sushmitha, S., & Kalaichelvi, D. T. (2017). Data Leak Prevention on Sensitive Data Using Levenshtein Distance Algorithm. *International Research Journal of Engineering and Technology(IRJET)*, *4*(2), 2026–2031. https://irjet.net/archives/V4/i2/IRJET-V4I2406.pdf

Bitkina, O. V., Kim, H. K., & Park, J. (2020). Usability and user experience of medical devices: An overview of the current state, analysis methodologies, and future challenges. *International Journal of Industrial Ergonomics*, *76*(February), 102932. https://doi.org/10.1016/j.ergon.2020.102932

BlackBerry. (2019). *Mobile Malware and APT Espionage Prolific, Pervasive, and Cross-Platform*. https://www.blackberry.com/content/dam/blackberry-com/asset/enterprise/pdf/direct/mobile-malware-report.pdf

Bonastre, L., & Granollers, T. (2014). A set of heuristics for user experience evaluation in E-commerce websites. *ACHI 2014 - 7th International Conference on Advances in Computer-Human Interactions*, *c*, 27–34. https://d1wqtxts1xzle7.cloudfront.net/44494283/achi_2014_2_10_20126_1.pdf?1459996504=&response-content-disposition=inline%3B+filename%3DA_Set_Of_Heuristics_for_User_Experience.pdf&Expires=1594811587&Signature=N0zp~-huz5HEhUmqUUZDx6Aw56Z-WGhHroXgfvQJRb4u1R

Bose, N., & Vishwanath, N. (2016). An Improved Method for Preventing Data Leakage in an Organisation. *Int. Journal of Engineering Research and Applications*, *6*(4), 01–07. https://d1wqtxts1xzle7.cloudfront.net/47580452/A0604060107.pdf?1469680192=&response-content-disposition=inline%3B+filename%3DAn_Improved_Method_for_Preventing_Data_L.pdf&Expires=1595062353&Signature=RsbrESpg5UbBjKp8YTOw5s-FSkF6m1qPbr0aasKK0c4sqN5N4wc-ZnPk

Botha, J., Van ‘t Wout, C., & Leenen, L. (2019). A Comparison of Chat Applications in Terms of Security and Privacy. *European Conference on Information Warfare and Security, ECCWS*, *2019-July*, 55–62. https://researchspace.csir.co.za/dspace/bitstream/handle/10204/11140/Botha_2019.pdf?sequence=1

Bravo-Lillo, C., Cranor, L. F., Downs, J., Komanduri, S., & Sleeper, M. (2011). Improving Computer Security Dialogs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 6949 LNCS* (Issue PART 4, pp. 18–35). https://doi.org/10.1007/978-3-642-23768-3_2

Breda, F., Barbosa, H., & Morais, T. (2017). Social Engineering and Cyber Security. *INTED2017 Proceedings*, *1*, 4204–4211. https://doi.org/10.21125/inted.2017.1008

Brooks, S., Garcia, M., Lefkovitz, N., Lightman, S., & Nadeau, E. (2017). *An introduction to privacy engineering and risk management in federal systems*. https://doi.org/10.6028/NIST.IR.8062

Bruin, N. (2018). *The Evolution of Instant Messaging*. https://doi.org/10.13140/RG.2.2.22305.10085

BtCIRT. (2017). *WeChat Alert*. Btcirt.Bt. https://www.btcirt.bt/wechat-alert/#

Bucher, B. (2020). *WhatsApp, WeChat and Facebook Messenger Apps - Global User Penetration and Statistics*. MessengerPeople.Com. https://www.messengerpeople.com/global-messenger-usage-statistics/

Caffo, A. (2018). *The best (and most secure) chat apps*. Avira.Com. https://www.avira.com/en/blog/best-chat-apps-smartphone

Cai, Y., & Wu, F. (2018). Data Security Framework for Electric Company Mobile Apps to Prevent Information Leakage. *Procedia Computer Science*, *139*, 280–286. https://doi.org/10.1016/j.procs.2018.10.269

Cambridge Dictionary. (2021). *Meaning of guideline in English*. Dictionary.Cambridge.Org.

https://dictionary.cambridge.org/dictionary/english/guideline

Caputo, D. D., Pfleeger, S. L., Sasse, M. A., Ammann, P., Offutt, J., & Deng, L. (2016). Barriers to Usable Security? Three Organisational Case Studies. *IEEE Security and Privacy*, *14*(5), 22–32. https://doi.org/10.1109/MSP.2016.95

Caro-Alvaro, S., Garcia-Lopez, E., Garcia-Cabot, A., De-Marcos, L., & Martinez-Herraiz, J.-J. (2018). Identifying Usability Issues in Instant Messaging Apps on iOS and Android Platforms. *Mobile Information Systems*, *2018*, 1–19. https://doi.org/10.1155/2018/2056290

Chami, F. C. (2017). *Behavioural Finance Factors Affecting Investment Performance By Retail Investors In The Nairobi Securities Exchange*. http://41.204.183.105/bitstream/handle/11732/3471/FILBERT CALIST CHAMI MBA 2017.pdf?sequence=1&isAllowed=y

Charles, K., & Dawson, P. (2011). Dispersed Change Agency and the Improvisation of Strategies During Processes of Change. *Journal of Change Management*, *11*(3), 329–351. https://doi.org/10.1080/14697017.2011.576653

Chaudhari, A. S. (2015). *Security analysis of SMS and related technologies Security*. https://pure.tue.nl/ws/files/46916565/840165-1.pdf

Chaudhry, J. A., Chaudhry, S. A., & Rittenhouse, R. G. (2016). Phishing Attacks and Defenses. *International Journal of Security and Its Applications*, *10*(1), 247–256. https://doi.org/10.14257/ijsia.2016.10.1.23

Chong, K., Malik, M. I., & Hannay, P. (2018). Mitigating man-in-the-middle attacks on mobile devices by blocking insecure http traffic without using vpn. *Proceedings of the 16th Australian Information Security Management Conference*, 1–13. https://doi.org/10.25958/5c526c2966688

Cisco. (2019). *What is Information Security?* Cisco.Com. https://www.cisco.com/c/en/us/products/security/what-is-information-security-infosec.html

Cohen-Sheffer, N. (2019). *Get Together With Group Calls on Viber*. Viber.Com. https://www.viber.com/en/blog/2019-03-07/new-on-viber-group-calls/#:~:text=You can now make group,with one of your friends.

Constine, J. (2018). *Now Snapchat lets you unsend messages like Facebook promised*. Techcrunch.Com. https://techcrunch.com/2018/06/11/snapchat-unsend/

Cooper, D., Regenscheid, A., Souppaya, M., Bean, C., Boyle, M., Cooley, D., & Jenkins, M. (2018). Security Considerations for Code Signing. *Nist*. https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.01262018.pdf

Corrigan, C. (2020). *The Very Best Encrypted Messaging Apps*. Avg.Com. https://www.avg.com/en/signal/secure-message-apps

Covert, Q., Steinhagen, D., Francis, M., & Streff, K. (2020). Towards a Triad for Data Privacy. *Proceedings of the 53rd Hawaii International Conference on System Sciences*, *3*, 4379–4387. https://doi.org/10.24251/hicss.2020.535

Curtis, P., & Carey, M. (2012). *Risk Assessment in Practice*. https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Governance-Risk-Compliance/dttl-grc-riskassessmentinpractice.pdf

Data and Application Security Group TH Köln. (2019a). *Guidelines used to redesign warnings*. Das.h-Brs.De. https://das.h-brs.de/usecured/guidelines/guidelines-used-to-redesign-warnings

Data and Application Security Group TH Köln. (2019b). *Warning Design Guidelines*. Das.h-Brs.De. https://das.h-brs.de/usecured/guidelines/warning-design-guidelines

Dawoud, A., & Bugiel, S. (2019). DroidCap: OS Support for Capability-based Permissions in Android. *Network and Distributed Systems Security (NDSS) Symposium 2019*, *February*. https://doi.org/10.14722/ndss.2019.23398

De Luca, A., Das, S., Ortlieb, M., Ion, I., & Laurie, B. (2016). Expert and Non-Expert Attitudes towards (Secure) Instant Messaging. *Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, *Soups*. https://www.usenix.org/system/files/conference/soups2016/soups2016-paper-de-luca.pdf

Degirmenci, K. (2020). Mobile users' information privacy concerns and the role of app permission requests. *International Journal of Information Management*, *50*(May), 261–272. https://doi.org/10.1016/j.ijinfomgt.2019.05.010

del Rey, A. M., Encinas, A. H., Vaquero, J. M., Dios, A. Q., & Sánchez, G. R. (2015). A Cellular Automata Model for Mobile Worm Propagation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9108*, 107–116. https://doi.org/10.1007/978-3-319-18833-1_12

Dempsey, D., & Kelliher, F. (2018). Business-to-Business Client Relationships in the Cloud Computing Software as a Service Realm. *Industry Trends in Cloud Computing*, 83–109. https://doi.org/10.1007/978-3-319-63994-9_5

Diamantaris, M., Papadopoulos, E. P., Markatos, E. P., Ioannidis, S., & Polakis, J. (2019). Reaper: Real-time app analysis for augmenting the android permission system. *CODASPY 2019 - Proceedings of the 9th ACM Conference on Data and Application Security and Privacy*, 37–48. https://doi.org/10.1145/3292006.3300027

Dodson, D., Souppaya, M., & Scarfone, K. (2020). Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF). *Nist*. https://doi.org/10.6028/NIST.CSWP.04232020%0Ahttps://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04232020.pdf

Downe-Wamboldt, B. (1992). Content analysis: Method, applications, and issues. *Health Care for Women International*, *13*(3), 313–321. https://doi.org/10.1080/07399339209516006

Dunne, P., Soe, S. P., Byrne, G., Venus, A., & Wheatley, A. R. (2004). Some demands on rapid prototypes used as master patterns in rapid tooling for injection moulding. *Journal of Materials Processing Technology*, *150*(3), 201–207. https://doi.org/10.1016/S0924-0136(03)00571-5

Facebook. (2020a). *Can I log out of Messenger?* Facebook.Com. https://www.facebook.com/help/messenger-app/android/719351428125983?helpref=platform_switcher&rdrhc

Facebook. (2020b). *How do I download a copy of my information on Facebook?* Facebook.Com. https://www.facebook.com/help/android-app/212802592074644?helpref=platform_switcher&rdrhc

Facebook. (2020c). *How do I remove or unsend a message that I've sent in Messenger?* Facebook.Com. https://www.facebook.com/help/messenger-app/android/194400311449172?helpref=platform_switcher&rdrhc

Facebook. (2020d). *Introducing Disappearing Messages on WhatsApp*. About.Fb.Com. https://about.fb.com/news/2020/11/introducing-disappearing-messages-on-whatsapp/

Facebook. (2020e). *What is the face recognition setting on Facebook and how does it work?* Facebook.Com. https://www.facebook.com/help/android-app/148233965247823?helpref=platform_switcher

Facebook. (2021a). *How do I add a file to my message on Facebook?* Facebook.Com. https://www.facebook.com/help/121288674619000

Facebook. (2021b). *How do I lock the Messenger app on my device ?* Facebook.Com. https://www.facebook.com/help/messenger-app/2585155295072006/?cms_platform=android-app&helpref=platform_switcher

Facebook. (2021c). *How do I turn update notifications on or off in Messenger ?* Facebook.Com. https://www.facebook.com/help/messenger-app/1692918830946285

Facebook. (2021d). *Messenger (Version 323.1.0.12.119) [Mobile Application]*. Play.Google.Com. https://play.google.com/store/apps/details?id=com.facebook.orca

Facebook. (2021e). *Voice and Video Calling Rooms*. Facebook.Com. https://www.facebook.com/help/messenger-app/1673374996287506

Fahrnberger, G. (2015). SIMS: A comprehensive approach for a secure instant messaging sifter. *Proceedings - 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2014*, *September 2014*, 164–173. https://doi.org/10.1109/TrustCom.2014.25

Faisal, M., Abbas, S., Rahman, H. U., Khan, M. Z., & Rahman, A. U. (2019). An Analysis of DDoS Attacks on the Instant Messengers. *Security and Communication Networks*, *2019*. https://doi.org/10.1155/2019/1751285

Faramarzi, S., Tabrizi, H. H., & Chalak, A. (2019). Telegram: An instant messaging application to assist distance language learning. *Teaching English with Technology*, *19*(1), 132–147. https://www.researchgate.net/profile/Sajad-Faramarzi/publication/331302501_Telegram_An_instant_messaging_application_to_assist_distance_language_learning/links/5cf7af44a6fdcc84750690ac/Telegram-An-instant-messaging-application-to-assist-distance-language-

Fischer, T. (2017). *IT disaster recovery, cloud computing and information security news The security and compliance issues related to instant messaging use*. Continuitycentral.Com. https://www.continuitycentral.com/index.php/news/technology/2270-the-security-and-compliance-issues-related-to-instant-messaging-use

Fosker, M. (2015). *Delete Those Messages You Never Meant to Send*. https://www.viber.com/en/blog/2015-11-29/delete-those-messages-you-never-meant-send/

Franklin, F., Breyer, F., & Kelner, J. (2014). Usability Heuristics for Collaborative Augmented Reality Remote Systems. *2014 XVI Symposium on Virtual and Augmented Reality*, *March*, 53–62. https://doi.org/10.1109/SVR.2014.31

Fu, H. (2017). *Improving Smartphone Permission Access Disclosures*. https://rucore.libraries.rutgers.edu/rutgers-lib/55469/PDF/1/

Futcher, L. (2011). *An Integrated Risk-Based Approach to Support IT Undergraduate Students in Secure Software Development*. http://dspace.nmmu.ac.za:8080/jspui/handle/10948/1673

Gcaza, N., Von Solms, R., Grobler, M. M., & Van Vuuren, J. J. (2017). A general morphological analysis: Delineating a cyber-security culture. *Information and Computer Security*, *25*(3), 259–278. https://doi.org/10.1108/ICS-12-2015-0046

Gelernter, N., Kalma, S., Magnezi, B., & Porcilan, H. (2017). The Password Reset MitM Attack. *Proceedings - IEEE Symposium on Security and Privacy*, 251–267. https://doi.org/10.1109/SP.2017.9

General Data Protection Regulation. (2018). *General Data Protection Regulation (GDPR) Compliance*. Gdpr.Eu. https://gdpr.eu/what-is-gdpr/

Gogoi, N. (2019). *Does WhatsApp Notify When You Take Screenshots of Status*. Guidingtech.Com. https://www.guidingtech.com/does-whatsapp-notify-screenshots-status/

Gomes, V., Reis, J., & Alturas, B. (2020). Social Engineering and the Dangers of Phishing. *Iberian Conference on Information Systems and Technologies, CISTI*, *2020-June*(June), 24–27. https://doi.org/10.23919/CISTI49556.2020.9140445

Gordon, J. R., & Gordon, S. R. (2002). Information Technology Service Delivery: an International Comparison. *Information Systems Management*, *19*(1), 62–70. https://doi.org/10.1201/1078/43199.19.1.20020101/31478.9

Gorski, P. L., von Zezschwitz, E., Lo Iacono, L., & Smith, M. (2019). On providing systematized access to consolidated principles, guidelines and patterns for usable security research and development†. *Journal of Cybersecurity*, *5*(1), 1–19. https://doi.org/10.1093/cybsec/tyz014

Griffin, A. (2019). *WhatsApp update to stop users taking screenshots of private chats*. Independent.Co.Uk. https://www.independent.co.uk/life-style/gadgets-and-tech/news/whatsapp-screenshot-chat-private-update-ios-android-feature-a8882116.html

Grigg, A. (2018). *WeChat's privacy issues mean you should delete China's No. 1 messaging app*. Financial Times. https://www.afr.com/world/asia/wechats-privacy-issues-mean-you-should-delete-chinas-no1-messaging-app-20180221-h0wgct

Gu, J., Xu, Y. (Calvin), Xu, H., Zhang, C., & Ling, H. (2017). Privacy concerns for mobile app download: An elaboration likelihood model perspective. *Decision Support Systems*, *94*, 19–28. https://doi.org/10.1016/j.dss.2016.10.002

Guidini Gonçalves, T., Marçal De Oliveira, K., & Kolski, C. (2016). HCI engineering integrated with capability maturity models: A study focused on requirements development. *Proceedings - International Conference on Research Challenges in Information Science*, *2016-Augus*. https://doi.org/10.1109/RCIS.2016.7549319

Hamid, N. A. A., Liew, C. W., Abdullah, N. H., & Omar, S. S. (2019). The Role of Information Technology Human Capability in the Implementation of Information Technology Governance (ITG): A Systematic Literature Review on Malaysian Organizations. *Advances in Science, Technology and Engineering Systems Journal*, *4*(4), 314–322. https://doi.org/10.25046/aj040440

He, W. (2013). A survey of security risks of mobile social media through blog mining and an extensive literature search. *Information Management and Computer Security*, *21*(5), 381–400. https://doi.org/10.1108/IMCS-12-2012-0068

Heartfield, R., & Loukas, G. (2018). Protection Against Semantic Social Engineering Attacks. In *Advances in Information Security* (Vol. 72, Issue October, pp. 99–140). https://doi.org/10.1007/978-3-319-97643-3_4

Hermawati, S., & Lawson, G. (2018). A user-centric methodology to establish usability heuristics for specific domains. In *Contemporary Ergonomics and Human Factors 2015* (Issue April, pp. 96–101). Taylor & Francis. https://doi.org/10.1201/b18293-12

Hess, T. A. (2012). *Investigation of Prototype Roles in Conceptual Design Using Case Study and Protocol Study Methods*. *August*. https://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=2418&context=all_theses

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75. https://doi.org/10.2307/25148625

Hu, Z., Buriachok, V., & Sokolov, V. (2020). Implementation of social engineering attack at institution of higher education. *CEUR Workshop Proceedings*, *2654*, 155–164. https://doi.org/10.2139/ssrn.3679106

Huang, L. V., & Zhang, K. (2019). Engagement, formality, and visibility: Managing paradoxes of using mobile instant messaging for work. *International Journal of Communication*, *13*, 1919–1938. https://www.ijoc.org/index.php/ijoc/article/download/7782/2633

Hub, M., & Čapková, V. (2010). Heuristic evaluation of usability of public administration portal. *International Conference on Applied Computer Science - Proceedings*, 234–239. http://www.wseas.us/e-library/conferences/2010/Malta/ACS/ACS-32.pdf

IBM. (2020a). Cost of a Data Breach Report - 2020. *IBM Security*, 76. https://www.ibm.com/downloads/cas/ZBZLY7KL

IBM. (2020b). X-Force Threat Intelligence Index 2020. In *IBM Security*. https://www.ibm.com/downloads/cas/DEDOLR3W

Idowu, S., & Dominic, E. D. (2019). Security Vulnerabilities of Skype Application Artifacts: A Digital Forensic Approach. *International Journal of Applied Information Systems (IJAIS)*, *12*(18), 5–10. https://www.ijais.org/archives/volume12/number18/idowu-2019-ijais-451784.pdf

InfoWatch Analytics Center. (2018a). *Data Breach Report: A Study on Global Data Leaks in H1 2018*. https://infowatch.com/sites/default/files/report/analytics/Data_Breach_Report_Global_Data_Leaks_H1_2018.pdf

InfoWatch Analytics Center. (2018b). *Data Breach Report 2018 - A Study of Data Leaks in the Middle East*. https://infowatch.com/sites/default/files/report/analytics/a_study_of_data_leaks_in_the_middle_east_in_2017-2018_.pdf

InfoWatch Analytics Center. (2018c). *Global Data Leakage Report, 2017*. https://infowatch.com/sites/default/files/report/Global_Data_Leak_Report_2017_ENG.pdf

Inostroza, R., Rusu, C., Roncagliolo, S., Rusu, V., & Collazos, C. A. (2016). Developing SMASH: A set of SMArtphone's uSability Heuristics. *Computer Standards and Interfaces*, *43*, 40–52. https://doi.org/10.1016/j.csi.2015.08.007

International Organisation for Standardisation. (2008). *Information technology — Security techniques — Systems Security Engineering — Capability Maturity Model ISO/IEC 21827:2008*. *2008*, 144. https://standards.iso.org/ittf/PubliclyAvailableStandards/c044716_ISO_IEC_21827_2008.zip

International Organisation for Standardisation. (2011). *Information technology — Security techniques — Information security risk management ISO/IEC 27005:2011*. https://www.iso.org/standard/75281.html

International Organisation for Standardisation. (2013a). *Information technology — Security techniques — Code of practice for information security controls ISO/IEC 27002:2013* (2nd ed.). www.sabs.co.za

International Organisation for Standardisation. (2013b). *Information technology — Security techniques — Information security management systems — Requirements ISO/IEC 27001:2013*. https://www.iso.org/standard/54534.html

International Organisation for Standardisation. (2016). *Information technology - Security techniques - Code of practice for Information security controls based on ISO/IEC 27002 for telecommunications organizations (ISO/IEC Standard No. 27011:2016)*. https://www.iso.org/obp/ui/#iso:std:iso-iec:27011:ed-2:v1:en

International Organisation for Standardisation. (2018). Information technology — Security techniques — Information security management systems — Overview and vocabulary (Standard No. ISO/IEC 27000). *ACM Workshop on Formal Methods in Security Engineering.Washington, DC, USA*, *34*(19), 45–55. https://www.iso.org/standard/73906.html

International Organisation for Standardisation. (2019). *Consumers and Standards: Partnership for a Better World*. Iso.Org. https://www.iso.org/sites/ConsumersStandards/1_standards.html

Jagadish, G., Jaswanth, L., Sowjanya, K., Sri Harsha, P., & Nikhil Kumar, M. (2019). A novel prototype to secure network using malware detection framework against malware attack in wireless network. *International Journal of Advance Research, Ideas and Innovations in Technology*, *5*(2), 286–292. https://d1wqtxts1xzle7.cloudfront.net/61217602/V5I2-127820191114-81814-r09pgs.pdf?1573754101=&response-content-disposition=inline%3B+filename%3DA_novel_prototype_to_secure_network_usin.pdf&Expires=1595063041&Signature=Fik-SXMcjJifvccWGo7IXqIVXPInAIgVAWnQD

Jagwani, P. (2016). *Analyzing Instant Messaging Applications for Threats: WhatsApp Case Study*. http://www.aspirare.org/VolNo2/PritiJagwani.pdf

Jain, A., & Prachi. (2016). Android Security: Permission Based Attacks. *2016 International Conference on Computing for Sustainable Global Development (INDIACom)*, 2754–2759. https://ieeexplore.ieee.org/document/7724765

Jhala, K. Y., & Patel, D. (2015). Dearth the Security of Smartphone Messaging Application: WhatsApp. *International Journal for Scientific Research & Development*, *3*(01), 110–113. https://d1wqtxts1xzle7.cloudfront.net/40978080/IJSRDV3I1055.pdf?1452065507=&response-content-disposition=inline%3B+filename%3DDearth_the_Security_of_Smartphone_Messag.pdf&Expires=1604908248&Signature=LyH1o4TPebLG-2mlIlTOmK8qHni7sJ8YiyxrR4hbvrCc9EqBWSQvUZb

Jimenez, C., Rusu, C., Roncagliolo, S., Inostroza, R., & Rusu, V. (2012). Evaluating a Methodology to Establish Usability Heuristics. *2012 31st International Conference of the Chilean Computer Science Society*, 51–59. https://doi.org/10.1109/SCCC.2012.14

Jobbins, S. (2012). Mind the gap! From simulation to reality. *Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications, HPCC-2012 - 9th IEEE International Conference on Embedded Software and Systems, ICESS-2012*, 1502–1507. https://doi.org/10.1109/HPCC.2012.219

Johansen, A. G. (2020). *What is a Trojan? Is It Virus or Malware?* Norton - Security Centre. https://us.norton.com/internetsecurity-malware-what-is-a-trojan.html

Johansen, C., Mujaj, A., Arshad, H., & Noll, J. (2018). *The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications*. http://arxiv.org/abs/1807.07952

John, S. (2019). *"Does Snapchat notify users when you take screenshots ?": Here's what you need to know*. Businessinsider.Com. https://www.businessinsider.com/does-snapchat-notify-screenshots?IR=T

Jongprasit, N., & Senivongse, T. (2020). Software Developer Performance Measurement Based on Code Smells in Distributed Version Control System. In R. Lee (Ed.), *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (Vol. 850, pp. 17–32). Springer. https://doi.org/10.1007/978-3-030-26428-4

Joyce, R. (2016). Disrupting Nation State Hackers. *Usenix Enigma, 2016*, 1–16. https://www.usenix.org/conference/enigma2016/conference-program/presentation/joyce

Kandukuri, S., & Srikanth, G. (2019). A Research Paper on Social Engineering and Growing Challenges in Cyber Security. *Think India*, *22*(41), 11–17. https://d1wqtxts1xzle7.cloudfront.net/63286461/19252-Article_Text-27814-1-10-2020022620200512-80643-1ug9t3v.pdf?1589299247=&response-content-disposition=inline%3B+filename%3DA_Research_Paper_on_Social_Engineering_a.pdf&Expires=1605858857&Signature=M2evwub

Kaur, K., Gupta, I., & Singh, A. K. (2017). *A Comparative Evaluation of Data Leakage/Loss Prevention Systems (DLPS)*. 87–95. https://doi.org/10.5121/csit.2017.71008

Khari, M., Gupta, S., Shrivastava, G., & Gupta, R. (2017). Role of cyber security in today's scenario. *Detecting and Mitigating Robotic Cyber Security Risks*, 177–191. https://doi.org/10.4018/978-1-5225-2154-9.ch013

Khoo, C., Robertson, K., & Deibert, R. (2019). *Installing Fear: A Canadian Legal and Policy Analysis of Using, Developing, and Selling Smartphone Spyware and Stalkerware Applications*. *JUNE*. https://www.citizenlab.ca/docs/stalkerware-legal.pdf

Kolar, A., & Grembergen, V. (2017). Standards, Best Practices and Codes of Ethics Impact on IT Service Quality – The Case of Slovenian IT Departments. *Economic and Business Review*, *19*(1), 51–72. https://doi.org/10.15458/85451.39

Kovesdi, C., & Joe, J. (2017). A novel tool for improving the data collection process during control room modernization human-system interface testing and evaluation activities. *10th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC and HMIT 2017*, *2*, 1261–1271. https://www.osti.gov/servlets/purl/1375335

Koyun, A., & Al Janabi, E. (2017). Social Engineering Attacks. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, *4*(6). https://www.jmest.org/wp-content/uploads/JMESTN42352270.pdf

Krippendorff, K. (2004). Content Analysis: An Introduction to Its Methodology (2nd ed.). In *Sage Publications* (Second). Sage Publications. https://doi.org/10.1177/1094428108324513

Kristoffer, A. T., & Vasbotten, M. T. (2016). *Lessons from implementing a league table application in the health sector - A case from Malawi*. https://www.duo.uio.no/bitstream/handle/10852/51613/thesis.pdf?sequence=8

Krombholz, K., Hobel, H., Huber, M., & Weippl, E. (2015). Advanced social engineering attacks. *Journal of Information Security and Applications*, *22*, 113–122. https://doi.org/10.1016/j.jisa.2014.09.005

Larson, G. W. (2016). *Instant messaging*. Encyclopedia Britannica; Encyclopaedia Britannica, Inc. https://www.britannica.com/topic/instant-messaging

Lechner, B., Petter, S., Fruhling, A., & Siy, H. (2013). The chicken and the pig: User involvement in developing usability heuristics. In *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime* (Vol. 5, pp. 3263–3270). https://core.ac.uk/reader/301360120

Li, Q., & Clark, G. (2013). Mobile Security: A Look Ahead. *IEEE Computer and Reliability Societies*, *February*, 5–55. https://doi.org/10.1016/b978-0-12-804629-6.00002-x

Li, Y., You, F., Ji, M., & You, X. (2020). The Influence of Smartphone Text Input Method, Posture, and Environment on User Experience. *International Journal of Human-Computer Interaction*, *00*(00), 1–12. https://doi.org/10.1080/10447318.2020.1719465

Lin, X. (2018). Android Forensics. In *Introductory Computer Forensics* (pp. 335–371). Springer International Publishing. https://doi.org/10.1007/978-3-030-00581-8_15

Liu, X., Wang, Y., Zhao, D., Zhang, W., & Shi, L. (2016). Patching by automatically tending to hub nodes based on social trust. *Computer Standards and Interfaces*, *44*, 94–101. https://doi.org/10.1016/j.csi.2015.08.001

Liu, Z., Xia, X., Lo, D., & Grundy, J. (2019). Automatic, highly accurate app permission recommendation. *Automated Software Engineering*, 1–34. https://doi.org/10.1007/s10515-019-00254-6

Ljuban, R. (2021). *Types of cyber attacks on businesses and theirs defense measures*. https://zir.nsk.hr/islandora/object/ffzg:3568/datastream/PDF/download

Lohani, S. (2019). Social Engineering: Hacking into Humans. *International Journal of Advanced Studies of Scientific Research*, *4*(1), 10. https://poseidon01.ssrn.com/delivery.php?ID=1811250840691270810760940700640041020580720420460630571120981190781131040761150051201220001070310460151051240681161101191090060420460870390830961261191030650180051100310080850260311200171060720131100030931261011

Luse, A., & Burkman, J. (2021). Gophish : Implementing a Real-World Phishing Exercise to Teach Social Engineering. *Journal of Cybersecurity Education, Research and Practice*, *2020*(2). https://digitalcommons.kennesaw.edu/jcerp/vol2020/iss2/5

Lutaaya, M. (2018). Rethinking App Permissions on iOS. *Conference on Human Factors in Computing Systems - Proceedings*, *2018-April*, 1–6. https://doi.org/10.1145/3170427.3180284

Madan, K. (2012). *Design and Implement the High Interaction Honeypot for a Campus Network*. *June*. http://117.203.246.91:8080/jspui/bitstream/10266/1717/3/1717l.pdf

Martin, R. (2018). *Developing a Complex User Interface for Mobile Data Collection Applications*. http://dbis.eprints.uni-ulm.de/1595/1/BA_MAR_2018.pdf

Masip, L., Martinie, C., Winckler, M., Palanque, P., Granollers, T., & Oliva, M. (2012). A design process for exhibiting design choices and trade-offs in (potentially) conflicting user interface guidelines. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7623 LNCS*, 53–71. https://doi.org/10.1007/978-3-642-34347-6_4

Mekruksavanich, S. (2017). Identifying Behavioral Design Flaws in Evolving Object-Oriented Software Using an Ontology-Based Approach. *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, *2018-Janua*, 424–429. https://doi.org/10.1109/SITIS.2017.76

Meyers, J. (2019). *How to Take Screenshots of Telegram Secret Chats on Android*. Android.Gadgethacks.Com. https://android.gadgethacks.com/how-to/take-screenshots-telegram-secret-chats-android-0179502/

Miller, K., Capan, M., Weldon, D., Noaiseh, Y., Kowalski, R., Kraft, R., Schwartz, S., Weintraub, W. S., & Arnold, R. (2018). The design of decisions: Matching clinical decision support recommendations to Nielsen's design heuristics. *International Journal of Medical Informatics*, *117*(May), 19–25. https://doi.org/10.1016/j.ijmedinf.2018.05.008

Miroshnichenko, M. (2016). *How to Recover Message History, Contacts and Viber Files on Android or Windows*. Hetmanrecovery.Com. https://hetmanrecovery.com/recovery_news/how-to-recover-message-history-contacts-and-viber-files-on-android-or-windows.htm

Mojapelo, M. S. (2015). A legislated School Library Policy: Can Functional School Libraries Be Envisioned Without One? *Mousaion: South African Journal of Information Studies*, *33*(2), 36–55. https://doi.org/10.25159/0027-2639/154

Momen, N., & Fritsch, L. (2020). *App-generated digital identities extracted through Android permission-based data access - a survey of app privacy*. https://doi.org/10.18420/sicherheit2020

Morales, R., Soh, Z., Khomh, F., Antoniol, G., & Chicano, F. (2017). On the use of developers' context for automatic refactoring of software anti-patterns. *Journal of Systems and Software*, *128*, 236–251. https://doi.org/10.1016/j.jss.2016.05.042

Mouton, F., Leenen, L., & Venter, H. S. (2016). Social engineering attack examples, templates and scenarios. *Computers and Security*, *59*, 186–209. https://doi.org/10.1016/j.cose.2016.03.004

Moyo, S., & Mnkandla, E. (2020). A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices. *IEEE Access*, *8*, 33735–33747. https://doi.org/10.1109/ACCESS.2020.2971000

Mujinga, M., Eloff, M., & Kroeze, J. (2013). Towards a Heuristic Model for Usable and Secure Online Banking. *Proceedings of the 24th Australasian Conference on Information Systems*, 1–12. https://researchbank.rmit.edu.au/view/rmit:161134/acis2013_394.pdf

Mujinga, M., Eloff, M. M., & Kroeze, J. H. (2019). Towards a framework for online information security applications development: A socio-technical approach. *South African Computer Journal*, *31*(1), 24–50. https://doi.org/10.18489/sacj.v31i1.587

Napoli, D. (2018). Developing Accessible and Usable Security (ACCUS) Heuristics. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, *2018-April*, 1–6. https://doi.org/10.1145/3170427.3180292

Nash, A. (2020). *How to Backup Viber Messages on PC?* Mobiletrans.Wondershare.Com. https://mobiletrans.wondershare.com/viber/how-to-backup-viber-on-pc.html

National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. (1979). The Belmont Report. *The Commission*. https://rd.mandela.ac.za/rcd/media/Store/documents/RecH/the-belmont-report-508c_FINAL.pdf

National Cyber Security Center (NCSC-NL). (2018). IT Security Guidelines for Mobile Apps. In *National Cyber Security Center Ministry of Justice and Security*. https://english.ncsc.nl/binaries/ncsc-en/documents/publications/2019/juni/01/whitepaper-it-security-guidelines-for-mobile-apps/Whitepaper_IT_Security_Guidelines_for_Mobile_Apps.pdf

National Cyber Security Partnership. (2004). *Improving Security Across the Software Development Lifecycle*. https://www.cyberpartnership.org/SDLCFULL.pdf

National Institute of Standards and Technology. (2017). *Information Technology Laboratory - Computer Security Resource Center*. https://csrc.nist.gov/glossary/term/information_security

Nayebi, F., Desharnais, J. M., & Abran, A. (2012). The state of the art of mobile application usability evaluation. *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering: Vision for a Greener Future, CCECE 2012*, *May*. https://doi.org/10.1109/CCECE.2012.6334930

Neumann, A., Laranjeiro, N., & Bernardino, J. (2018). An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*, *PP*(c), 1. https://doi.org/10.1109/TSC.2018.2847344

Nguyen, D. C., Wermke, D., Acar, Y., Backes, M., Weir, C., & Fahl, S. (2017). A Stitch in Time: Supporting android developers inwriting secure code. *Proceedings of the ACM Conference on Computer and Communications Security*, 1065–1077. https://doi.org/10.1145/3133956.3133977

Nield, D. (2018). *How to send self-destructing messages*. Popsci.Com. https://www.popsci.com/send-self-destructing-messages/

Nielsen, J. (1995). *10 Usability Heuristics for User Interface Design*. https://www.nngroup.com/articles/ten-usability-heuristics/

Nielsen, J. (2016). *The Distribution of Users' Computer Skills: Worse Than You Think*. https://www.nngroup.com/articles/computer-skill-levels/

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the ACM CHI 90 Human Factors in Computing Systems Conference*, *April*, 249–256. https://doi.org/10.1145/97243.97281

Nimgaonkar, A., & Kumbhar, R. (2020). Usable Security: Need of Digital Era. *5th International Conference On "Innovations in IT and Management," 68*(27), 497–501.

NIST SP800-53. (2020). Security and Privacy Controls for Information Systems and Organizations. In *NIST Special Publication* (Vol. 800). https://doi.org/10.6028/NIST.SP.800-53r5

Norgren, A. (2004). *Requirements Engineering and Prototyping in a Legacy Software Setting*. https://www.diva-portal.org/smash/get/diva2:215122/FULLTEXT01.pdf

Nweke, L. O. (2017). Using the CIA and AAA Models to Explain Cybersecurity Activities. *PM World Journal*, *VI*(Xii), 1–3. https://pmworldlibrary.net/wp-content/uploads/2017/05/171126-Nweke-Using-CIA-and-AAA-Models-to-explain-Cybersecurity.pdf

Nyakomitta, P. S., Ogara, D. S., & Abeka, D. S. (2016). Empirical Investigation of Instant Messaging Security in a Virtual Environment. *International Journal of Computer Applications Technology and Research*, *5*(12), 733–747. https://doi.org/10.7753/ijcatr0512.1003

Odukoya, O. H., Adedoyin, O. B., Akhigbe, B. I., Aladesanmi, T. A., & Aderounmu, G. A. (2018). An architectural-based approach to detecting spim in electronic means of communication. *Nigerian Journal of Technology*, *37*(3), 770. https://doi.org/10.4314/njt.v37i3.28

Okereafor, K., & Adelaiye, O. (2020). Randomized Cyber Attack Simulation Model : A Cybersecurity Mitigation Proposal for Post COVID-19 Digital Era. *International Journal of Recent Engineering Research and Development (IJRERD)*, *05*(07), 61–72. https://www.researchgate.net/profile/Kenneth_Okereafor/publication/343318105_Randomized_Cyber_Attack_Simulation_Model_A_Cybersecurity_Mitigation_Proposal_for_Post_COVID-19_Digital_Era/links/5f22ca1a92851cd302c8a4b5/Randomized-Cyber-Attack-Simulation-Model

Otachi, E. (2019). *How to Send Self-Destructing Messages in Facebook Messenger*. Online-Tech-Tips.Com. https://www.online-tech-tips.com/smartphones/how-to-send-self-destructing-messages-in-facebook-messenger/

OWASP. (2016). *OWASP Mobile Top 10 Risks - 2016*. Owasp.Org. https://owasp.org/www-project-mobile-top-10/

OWASP. (2017). *OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks*. https://owasp.org/www-project-top-ten/%0D%0A

OWASP. (2019). *OWASP API Security Top 10 2019: The Ten Most Critical API Security Risks*. https://owasp.org/www-project-api-security/

Pabreja, K., Grover, A., & Sharma, V. (2020). PingMe - a mobile App with a Difference. In M. Kumar, R. Choudhary, & S. K. Pandey (Eds.), *Emerging Trends in Big Data IoT and Cyber Security* (pp. 122–125). Excellent Publishing House. https://msi-ggsip.org/wp-content/uploads/conference2020.pdf#page=135

Pakhomov, V., Bondarenko, O., & Dumchikov, M. (2019). Criminal legal characteristic of social engineering as a way of committing fraud. *Law and Life*, *April*. https://ibn.idsi.md/sites/default/files/imag_file/149-153_4.pdf

Paliszkiewicz, J. (2019). Information Security Policy Compliance: Leadership and Trust. *Journal of Computer Information Systems*, *59*(3), 211–217. https://doi.org/10.1080/08874417.2019.1571459

Pardeshi, A. S., & Pardeshi, A. K. (2020). Inculcate Precise Impressions of Ethical Hacking on Teenagers. *International Journal Of Computer Science And Applications*, *13*(1), 38–42. http://www.researchpublications.org/CKT-2020/IJCSA-13-01-11.pdf

Parente da Costa, R., & Dias Canedo, E. (2019). *A Set of Usability Heuristics for Mobile Applications* (pp. 180–193). https://doi.org/10.1007/978-3-030-22646-6_13

Patton, M. Q. (2002). *Qualitative Research & Evaluation Methods* (Fourth). Sage publications Inc. https://books.google.co.za/books?hl=en&lr=&id=ovAkBQAAQBAJ&oi=fnd&pg=PP1&ots=ZRZ_7qwFzZ&sig=ZFEE_GXoH37fZgphhEaOqufysYA&redir_esc=y#v=onepage&q&f=false

Paul, T., & Hof, H.-J. (2016). An empirical survey on how much security and privacy customers want in instant messengers. *The 10th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*, *July*, 89–94. https://www.researchgate.net/profile/Hans-Joachim_Hof/publication/303341231_An_Empirical_Survey_on_how_Much_Security_and_Privacy_Customers_Want_in_Instant_Messengers/links/5797489b08aeb0ffcd06cf5c.pdf

Pfleeger, S. L., & Caputo, D. D. (2012). Leveraging behavioural science to mitigate cyber security risk. *Computers and Security*, *31*(4), 597–611. https://doi.org/10.1016/j.cose.2011.12.010

Piwek, L., & Joinson, A. (2016). "What do they snapchat about?" Patterns of use in time-limited instant messaging service. *Computers in Human Behavior*, *54*, 358–367. https://doi.org/10.1016/j.chb.2015.08.026

Prasad, R., & Rohokale, V. (2020). *Cyber Security: Communication of Information and The Lifeline Technology*. Springer. https://doi.org/10.1007/978-3-030-31703-4_2

Pribeanu, C. (2017). A Revised Set of Usability Heuristics for the Evaluation of Interactive Systems. *Informatica Economica*, *21*(3/2017), 31–38. https://doi.org/10.12948/issn14531305/21.3.2017.03

Protection of Personal Information Act (POPIA). (2019). *Protection of Personal Information Act - Section 2 Purpose of Act*. Popia.Co.Za. https://popia.co.za/section-2-purpose-of-act/

Ptsecurity.com. (2019). *Vulnerabilities and threats in mobile applications*. https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Mobile-Application-Vulnerabilities-and-Threats-2019-eng.pdf

Quiñones, D., & Rusu, C. (2017). How to develop usability heuristics: A systematic literature review. *Computer Standards and Interfaces*, *53*(September 2016), 89–122. https://doi.org/10.1016/j.csi.2017.03.009

Quiñones, D., Rusu, C., Arancibia, D., González, S., & Saavedra, M. J. (2020). SNUXH: A set of social network user experience heuristics. *Applied Sciences (Switzerland)*, *10*(18). https://doi.org/10.3390/APP10186547

Raber, F., & Krueger, A. (2017). Towards understanding the influence of personality on mobile app permission settings. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10516 LNCS*, 62–82. https://doi.org/10.1007/978-3-319-68059-0_4

Rahman, A., Farhana, E., & Imtiaz, N. (2019). Snakes in Paradise?: Insecure python-related coding practices in stack overflow. *IEEE International Working Conference on Mining Software Repositories*, *2019-May*, 200–204. https://doi.org/10.1109/MSR.2019.00040

Rahman, M. R., Rahman, A., & Williams, L. (2019). Share, but be Aware: Security Smells in Python Gists. *Proceedings - 2019 IEEE International Conference on Software Maintenance and Evolution, ICSME 2019*, 536–540. https://doi.org/10.1109/ICSME.2019.00087

Rai, A., Singh, A. S., & Kumar, A. S. (2020). A Review of Information Security: Issues and Techniques. *International Journal for Research in Applied Science and Engineering Technology*, *8*(5), 953–960. https://doi.org/10.22214/ijraset.2020.5150

Rajamenakshi, R., & Padmavathi, G. (2016). Design and Detection of Network Covert Channels- An Overview. *International Journal of Computer Science and Information Security (IJCSIS)*, *14*(6), 821–828. https://d1wqtxts1xzle7.cloudfront.net/47340143/97_Paper_310516202_IJCSIS_Camera_Ready_B_821-828.pdf?1468905443=&response-content-disposition=inline%3B+filename%3DDesign_and_Detection_of_Network_Covert_C.pdf&Expires=1595063456&Signature=GvkpKFbp30Nz6seFie~

Rajendran, J. A., Baharin, H., & Kamal, F. M. (2019). Understanding Instant Messaging in the Workplace. In H. B. Zaman, A. F. Smeaton, T. K. Shih, S. Velastin, T. Terutoshi, N. M. Ali, & M. N. Ahmad (Eds.), *Advances in Visual Informatics: Vol. 11870 LNCS* (pp. 441–450). Springer. https://doi.org/10.1007/978-3-030-34032-2

Rajivan, P., Moriano, P., Kelley, T., & Camp, L. J. (2017). Factors in an end user security expertise instrument. *Information and Computer Security*, *25*(2), 190–205. https://doi.org/10.1108/ICS-04-2017-0020

Ramakrishnan, U. P., & Tandon, J. . (2018). The Evolving Landscape of Cyber Threats. *The Indian Journal of Management*, *11*(1), 31–35. http://library.capella.edu/login?qurl=https%3A%2F%2Fsearch.proquest.com%2Fdocview%2F2011259745%3Faccountid%3D27965

Rana, M. E., Wei, G., & Hoornaert, P. (2015). An Enterprise Instant Messaging (EIM) solution to cater issues associated with instant messaging (IM) in business. *2015 IEEE Student Conference on Research and Development (SCOReD)*, 187–192. https://doi.org/10.1109/SCORED.2015.7449321

Raymond, A., Schubauer, J., & Madappa, D. (2020). Over-Privileged Permissions: Using Technology and Design to Create Legal Compliance. *Journal of Business and Technology 15.1*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3546518

Rayome, A. D. (2017). *10 bad habits network administrators should avoid at all costs*. Techrepublic.Com. https://www.techrepublic.com/article/10-bad-habits-network-administrators-should-avoid-at-all-costs/

Reardon, J., On, A. E. B., Feal, Á., Vallina-Rodriguez, N., Wijesekera, P., & Egelman, S. (2019). 50 Ways to Leak Your Data: An Exploration of Apps' Circumvention of the Android Permissions System. *Proceedings of the 28th USENIX Security Symposium*, 603–620. https://www.usenix.org/system/files/sec19-reardon.pdf

Reddy, N. (2019). Practical Cyber Forensics. *Practical Cyber Forensics*, 1–28. https://doi.org/10.1007/978-1-4842-4460-9

Renaud, K. V., & Van Biljon, J. (2017). Demarcating Mobile Phone Interface Design Guidelines to Expedite Selection. *South African Computer Journal*, *29*(3), 127–144. https://doi.org/10.18489/sacj.v29i3.438

Reshmi, T. S., & Daniel Madan Raja, S. (2019). A Review on Self Destructing Data: Solution for Privacy Risks in OSNs. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 231–235. https://doi.org/10.1109/ICACCS.2019.8728453

Ricle, J. (2020). *How To Recover Deleted Telegram Posts & Media*. Telegramadviser.Com. https://www.telegramadviser.com/recover-deleted-telegram-posts-media/

Rishika, K. K., & Damodaran, V. (2020). A study on security issues in the cloud. *Journal of Information and Computational Science*, *10*(4). http://www.joics.org/gallery/ics-2885.pdf

Roberts, G. (2016). *IT Incident Criteria*. *January*. https://www.reading.ac.uk/web/files/its/Incident_Criteria_v1.1.pdf

Robinson, N. (2005). Design Methodology. *The Planting Design Handbook*, *2*, 203–247. https://doi.org/10.4324/9781315554648-11

Roesner, F., Gill, B. T., & Kohno, T. (2014). Sex, Lies, Or kittens? investigating the use of snapchat's self-destructing messages. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8437*, 64–76. https://doi.org/10.1007/978-3-662-45472-

5_5

Rotem, L., & Segev, G. (2018). Out-of-band authentication in group messaging: Computational, statistical, optimal. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *483*, 63–89. https://doi.org/10.1007/978-3-319-96884-1_3

Saavedra, M.-J., Rusu, C., Quiñones, D., & Roncagliolo, S. (2019). A Set of Usability and User eXperience Heuristics for Social Networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11578 LNCS* (pp. 128–139). https://doi.org/10.1007/978-3-030-21902-4_10

Sabillon, R., Cano, J., Cavaller, V., & Serra, J. (2016). Cybercrime and Cybercriminals: A Comprehensive Study. *International Journal of Computer Networks and Communications Security*, *4*(6), 165–176. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/78507/1/p1_4-6.pdf

Sahoo, S. R., & Gupta, B. B. (2018). Security Issues and Challenges in Online Social Networks (OSNs) Based on User Perspective: Principles, Algorithm, Applications, and Perspectives. *Computer and Cyber Security*, *July*, 591–606. https://doi.org/10.1201/9780429424878-22

Salahdine, F., & Kaabouch, N. (2019). Social engineering attacks: A survey. *Future Internet*, *11*(4). https://doi.org/10.3390/FI11040089

Samantray, O. P., Tripathy, S. N., & Das, S. K. (2018). A Theoretical Feature-wise Study of Malware Detection Techniques. *International Journal of Computer Sciences and Engineering*, *6*(12), 879–887. https://doi.org/10.26438/ijcse/v6i12.879887

Sapin, J., & Duy, K. T. (2011). On the Types and Roles of Demonstrators. *International Conference on Engineering Design, ICED11*, *August*. https://www.designsociety.org/download-publication/30788/ON+THE+TYPES+AND+ROLES+OF+DEMONSTRATORS+FOR+DESIGNING+MEDICAL+DEVICES

Schnitzler, T., Utz, C., Farke, F. M., Pöpper, C., & Dürmuth, M. (2020). Exploring user perceptions of deletion in mobile instant messaging applications. *Journal of Cybersecurity*, *6*(1), 1–15. https://doi.org/10.1093/cybsec/tyz016

Sharma, A. (2020). *SC Chat Locker: Protecting Your Chats On Snapchat App*. Blogs.Systweak.Com. https://blogs.systweak.com/sc-chat-locker-app-to-lock-chats-on-snapchat/

Sheer, V. C., & Rice, R. E. (2017). Mobile instant messaging use and social capital: Direct and indirect associations with employee outcomes. *Information and Management*, *54*(1), 90–102. https://doi.org/10.1016/j.im.2016.04.001

Shirvanian, M., Saxena, N., & George, J. J. (2017). On the pitfalls of end-To-end encrypted communications: A study of remote key-fingerprint verification. *ACM International Conference Proceeding Series*, 499–511. https://doi.org/10.1145/3134600.3134610

Silas. (2020). *How to sign up WeChat account 2020 (updated)*. Chinahelp4u.Com. https://chinahelp4u.com/how-to-sign-up-wechat-account/

Sinha, P., Rai, A. K., & Bhushan, B. (2019). Information Security threats and attacks with conceivable counteraction. *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2019*, 1208–1213. https://doi.org/10.1109/ICICICT46008.2019.8993384

Snapchat. (2017a). *Forget a Linked Device*. Snapchat. https://support.snapchat.com/en-US/a/forget-devices

Snapchat. (2017b). *Group Chat*. Support.Snapchat.Com. https://support.snapchat.com/en-US/a/group-chat

Snapchat. (2017c). *Send a Snap*. Support.Snapchat.Com. https://support.snapchat.com/en-US/article/send-snap

Snapchat. (2017d). *Set Up Two-Factor Authentication*. Snapchat. https://support.snapchat.com/en-US/a/enable-login-verification

Snapchat. (2017e). *Snapchat Support - Email Address*. Support.Snapchat.Com. https://support.snapchat.com/en-US/article/change-email

Snapchat. (2017f). *Snapchat Support - Mobile Number*. Support.Snapchat.Com. https://support.snapchat.com/en-US/a/mobile-verification

Snapchat. (2017g). *Voice and Video Chat*. Support.Snapchat.Com. https://support.snapchat.com/en-US/a/video-chat

Snapchat. (2019). *Download My Data*. Support.Snapchat.Com. https://support.snapchat.com/en-US/a/download-my-data

Snapchat. (2020a). *Snapchat Privacy Policy*. Snap.Com. https://www.snap.com/en-US/privacy/privacy-policy

Snapchat. (2020b). *When does Snapchat delete Snaps and Chats?* Support.Snapchat.Com.

https://support.snapchat.com/en-US/a/when-are-snaps-chats-deleted

Souppaya, M., & Scarfone, K. (2013). Guidelines for Managing the Security of Mobile Devices in the Enterprise. In *NIST Special Publication 800-124, Revision 1*. https://doi.org/10.6028/NIST.SP.800-124r1

South African National Standard. (2009). *Information technology — Security techniques — Information security management guidelines for telecommunications organizations based on ISO/IEC 27002 SANS 27011:2009* (1st ed.). http://my.mandela.ac.za/sabs/documents/SANS27011.pdf

South African National Standard. (2018). *Information technology — Security techniques — Information security management for inter-sector and inter-organizational communications SANS 27010:2018* (2nd ed.). http://my.mandela.ac.za/sabs/documents/SANS27010_2018_Ed2.pdf

South African National Standard. (2020a). *Information technology — Security techniques — Information security management — Monitoring, measurement, analysis and evaluation SANS 27004:2020* (2nd ed.). http://my.mandela.ac.za/sabs/documents/SANS27004_2020_Ed2.pdf

South African National Standard. (2020b). *Information technology — Security techniques — Information security management systems — Guidance SANS 27003:2020* (2nd ed.). http://my.mandela.ac.za/sabs/documents/SANS27003_2020_Ed2.pdf

StatCounter. (2020). *Mobile Operating System Market Share Worldwide*. StatCounter.Com. https://gs.statcounter.com/os-market-share/mobile/worldwide

Statista. (2021a). *Most popular global mobile messaging apps 2021*. Statista.Com. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps

Statista. (2021b). *Number of available applications in the Apple App Store from 2000 to 2021*. Statista.Com. https://www.statista.com/statistics/268251/number-of-apps-in-the-itunes-app-store-since-2008/

Statista. (2021c). *Number of available applications in the Google Play Store from December 2009 to July 2021*. Statista.Com. https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

Still, J. D., Cain, A., & Schuster, D. (2017). Human-centered authentication guidelines. *Information and Computer Security*, *25*(4), 437–453. https://doi.org/10.1108/ICS-04-2016-0034

Strom, D. (2020). *What is application security? A process and tools for securing software*. Csoonline.Com. https://www.csoonline.com/article/3315700/what-is-application-security-a-process-and-tools-for-securing-software.html

Sullivan, J. (2020). *Messenger Introduces App Lock and New Privacy Settings*. About.Fb.Com. https://about.fb.com/news/2020/07/messenger-app-lock-and-privacy-settings/

Suryanarayana, G., Samarthyam, G., & Sharma, T. (2015). *Refactoring for Software Design Smells Managing Technical Debt*. Elsevier. http://repository.fue.edu.eg/xmlui/bitstream/handle/123456789/3647/10166.pdf?sequence=1&isAllowed=y

Sutikno, T., Handayani, L., Stiawan, D., Riyadi, M. A., & Subroto, I. M. I. (2016). WhatsApp, Viber and Telegram: which is the Best for Instant Messaging? *International Journal of Electrical and Computer Engineering*, *6*(3), 909–914. https://doi.org/10.11591/ijece.v6i3.10271

Symeonidis, I., & Lenzini, G. (2020). Systematization of threats and requirements for private messaging with untrusted servers: The case of e-mailing and instant messaging. *ICISSP 2020 - Proceedings of the 6th International Conference on Information Systems Security and Privacy*, 593–602. https://doi.org/10.5220/0009003805930602

T9gram.com. (2020). *Registration in Telegram. How to Register an Account*. T9gram.Com. https://t9gram.com/f/registration-in-telegram/

Taleqani, A. R., Nygard, K. E., Bridgelall, R., & Hough, J. (2018). Machine Learning Approach to Cyber Security in Aviation. *IEEE International Conference on Electro Information Technology*, *2018-May*, 147–152. https://doi.org/10.1109/EIT.2018.8500165

Tanner, M. (2018). *Steps to Register a WeChat Official Account Under an Overseas Business Entity*. Chinaskinny.Com. https://www.chinaskinny.com/blog/register-overseas-wechat-account/

Taole, D. C. (2020). *Guidelines for the effective use of audio- visual technology in lecture rooms at North-West University*. https://repository.nwu.ac.za/bitstream/handle/10394/34705/Taole_DC_20858337.pdf?sequence=1

Taylor, V. F., & Martinovic, I. (2016). *Quantifying Permission-Creep in the Google Play Store*. http://arxiv.org/abs/1606.01708

Techjunkie. (2020). *How To Tell If Someone Screenshots Your Facebook Messenger Conversation*.

Social.Techjunkie.Com. https://social.techjunkie.com/tell-someone-screenshots-facebook-messenger/

Telegram. (2015a). *Active Sessions and Two-Step Verification*. Telegram.Org. https://telegram.org/blog/sessions-and-2-step-verification

Telegram. (2015b). *Sending Files On Steroids — And More*. Telegram.Org. https://telegram.org/blog/files-on-steroids

Telegram. (2017). *Unsend Messages, Network Usage, and More*. Telegram.Org. https://telegram.org/blog/unsend-and-usage

Telegram. (2018). *Chat Export Tool, Better Notifications and More*. Telegram.Org. https://telegram.org/blog/export-and-more

Telegram. (2019a). *Taking Back Our Right to Privacy*. Telegram.Org. https://telegram.org/blog/unsend-privacy-emoji

Telegram. (2019b). *Telegram F.A.Q.* Telegram. https://telegram.org/faq#login-and-sms

Telegram. (2020a). *Telegram Privacy Policy*. Telegram.Org. https://telegram.org/privacy

Telegram. (2020b). *Video Calls and Seven Years of Telegram*. Telegram.Org. https://telegram.org/blog/video-calls

Telegram. (2020c). *Voice Chats Done Right*. Telegram.Org. https://telegram.org/blog/voice-chats

Thaduri, L. (2020). Detecting Application Anomalies: Machine Learning Approach. *Culminating Projects in Information Assurance*. https://repository.stcloudstate.edu/msia_etds/108

Thomala, L. L. (2020). *Number of monthly active smart device users of Tencent QQ in China from 2014 to 2019*. Statista.Com. https://www.statista.com/statistics/227352/number-of-active-tencent-im-user-accounts-in-china/

Titcomb, J. (2019). *Snapchat adds end-to-end encryption to protect users ' messages*. Telegraph.Co.Uk. https://www.telegraph.co.uk/technology/2019/01/09/snapchat-adds-end-to-end-encryption-protect-users-messages/

TOKOK. (2018). *Install and set Google Authenticator*. Help.Tokok.Io. https://help.tokok.io/hc/en-us/articles/360006528133-Install-and-set-Google-Authenticator

Tripathi, Y. (2020). *Does Facebook Notify When You Screenshot A Story Or Post ? Know Details*. https://www.republicworld.com/technology-news/apps/does-facebook-notify-when-you-screenshot-a-story-or-a-post-know-detail.html 1/4

Van Eemeren, F. H., & Grootendorst, R. (2003). A systematic theory of argumentation: The pragma-dialectical approach. *A Systematic Theory of Argumentation: The Pragma-Dialectical Approach*, 1–216. https://doi.org/10.1017/CBO9780511616389

van Greunen, D., Pottas, D., & Yeratziotis, A. (2011). A three-phase process to develop heuristics. *PROCEEDINGS OF THE 13th ANNUAL CONFERENCE ON WORLD WIDE WEB APPLICATIONS*, *September*. https://www.researchgate.net/profile/Alexandros-Yeratziotis/publication/303159922_A_three-phase_process_to_develop_heuristics/links/5b961250299bf14739380fcc/A-three-phase-process-to-develop-heuristics.pdf

Viber. (2018). *More Protection Than Ever Before: New Privacy Features on Viber For Everyone*. Viber.Com. https://www.viber.com/en/blog/2018-10-14/new-privacy-features-on-viber-for-everyone/

Viber. (2019a). *Back Up and Restore Viber Messages*. Help.Viber.Com. https://help.viber.com/en/article/back-up-and-restore-viber-messages

Viber. (2019b). *Get Started: Viber Setup*. Help.Viber.Com. https://help.viber.com/en/article/get-started-viber-setup

Viber. (2019c). *Viber Account Security and Encryption*. Help.Viber.Com. https://help.viber.com/en/article/viber-account-security-and-encryption

Viber. (2020). *Viber Security*. Viber.Com. https://www.viber.com/en/security/

Viber. (2021a). *Calls & Messages*. Help.Viber.Com. https://help.viber.com/en/calls-messages

Viber. (2021b). *Group Voice and Video Calls on Your Desktop*. Help.Viber.Com. https://help.viber.com/en/article/group-voice-and-video-calls-on-your-desktop

Voskoboinicov, S., & Melnyk, S. (2018). Cyber Security in the Modern Sociation and Improvement of Preparation of Future Factors in the Field of Competent Approach. *Social Work and Education*, *5*(1), 103–112. https://doi.org/10.25128/2520-6230.18.1.10

Walliman, N. (2010). Research Methods: The Basics. In *Research Methods: The Basics*. https://doi.org/10.4324/9780203836071

Warner, M. (2012). Cybersecurity: A pre-history. *Intelligence and National Security*, *27*(5), 781–799.

https://doi.org/10.1080/02684527.2012.708530

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, *26*(2), xiii–xxiii. https://doi.org/10.1.1.104.6570

WeChat. (2015). *Voiceprint: The New WeChat Password*. WeChat: Chatterbox. https://blog.wechat.com/2015/05/21/voiceprint-the-new-wechat-password/

WeChat. (2018). *What's new in WeChat 6.6.2*. Blog.Wechat.Com. https://blog.wechat.com/2018/01/31/whats-new-in-wechat-6-6-2-for-ios/

WeChat. (2020a). *Can messages be cancelled or deleted if already sent?* Wechat.Com. https://www.wechat.co.za/faq/can-messages-be-cancelled-or-deleted-if-already-sent/

WeChat. (2020b). *How do I enable account protection?* Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&lang=en&plat=ios&id=1208117b2mai141024yARrIB

WeChat. (2020c). *How do I protect my WeChat account if it has been hacked?* Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&lang=en&plat=ios&id=190903FfQrq2190903BFNRni&Channel=helpcenter

WeChat. (2020d). *How do I recall a sent message?* Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&plat=2&lang=en&id=120813euEJVf1410236fI7RB&Channel=helpcenter

WeChat. (2020e). *How do I safeguard my account's security?* Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&id=1705236ruuau170523Q7nEZB&lang=en&plat=3&Channel=Q

WeChat. (2020f). *How do I transfer my chat history to a new device?* Help.Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&id=120813euejvf150213fn3uyz&lang=en&plat=2&Channel=WeChatOfficialWebsite

WeChat. (2020g). *How long can I save my chat history?* Help.Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&lang=en&plat=2&pid=1001146&id=120813euejvf141023berq6f&Channel=helpcenter

WeChat. (2020h). *How secure are my chat messages and conversations on WeChat? Can third-parties snoop or read my messages?* Help.Wechat.Com. https://help.wechat.com/cgi-bin/micromsgbin/oshelpcenter?opcode=2&plat=1&lang=en&id=1208117b2mai1410243yyQFZ&Channel=helpcenter

WeChat. (2020i). *Where is my chat history stored?* Help.Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&lang=en&plat=2&pid=1001146&id=120813euejvf150213uynvjm&Channel=helpcenter

WeChat. (2020j). *Why can't I back up my chat history to WeChat's server or a cloud service?* Help.Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?t=help_center/topic_detail&opcode=2&plat=2&lang=en&id=150915VVrUve15091522E3eU&Channel=WeChatOfficialWebsite

WeChat. (2021). *WeChat Help Center*. Help.Wechat.Com. https://help.wechat.com/cgi-bin/micromsg-bin/oshelpcenter?opcode=2&id=1703037JBzqu1703037vue22

Weichbroth, P., & Łysik, Ł. (2020). Mobile Security: Threats and Best Practices. *Mobile Information Systems*, *2020*, 1–15. https://doi.org/10.1155/2020/8828078

WhatsApp Inc. (2020a). *About two-step verification*. https://faq.whatsapp.com/general/verification/about-two-step-verification/?lang=fb

WhatsApp Inc. (2020b). *How to delete messages*. Faq.Whatsapp.Com. https://faq.whatsapp.com/iphone/chats/how-to-delete-messages/?lang=en

WhatsApp Inc. (2020c). *How to log in or out*. Faq.Whatsapp.Com. https://faq.whatsapp.com/general/download-and-installation/how-to-log-in-or-out/?lang=en

WhatsApp Inc. (2020d). *How to make a group video call*. Faq.Whatsapp.Com. https://faq.whatsapp.com/android/voice-and-video-calls/how-to-make-a-group-video-call/?lang=en

WhatsApp Inc. (2020e). *How to make a group voice call*. Faq.Whatsapp.Com. https://faq.whatsapp.com/android/voice-and-video-calls/how-to-make-a-group-voice-call/?lang=en

WhatsApp Inc. (2020f). *How to save your chat history*. Faq.Whatsapp.Com. https://faq.whatsapp.com/android/chats/how-to-save-your-chat-history/?lang=en

WhatsApp Inc. (2020g). *How to send media Send*. Faq.Whatsapp.Com. https://faq.whatsapp.com/web/chats/how-to-send-media/?lang=en

WhatsApp Inc. (2020h). *How to verify your phone number*. Faq.Whatsapp.Com. https://faq.whatsapp.com/kaios/verification/how-to-verify-your-phone-number/?lang=en

WhatsApp Inc. (2020i). *Introducing disappearing messages on WhatsApp*. Blog.Whatsapp.Com. https://blog.whatsapp.com/introducing-disappearing-messages-on-whatsapp

WhatsApp Inc. (2020j). *WhatsApp FAQ - How to restore your chat history*. Faq.Whatsapp.Com. https://faq.whatsapp.com/iphone/chats/how-to-restore-your-chat-history/

WhatsApp Inc. (2020k). *WhatsApp Privacy Policy*. Whatsapp.Com. https://www.whatsapp.com/legal/privacy-policy

WhatsApp Inc. (2021). *WhatsApp Features*. Whatsapp.Com. https://www.whatsapp.com/features/

Wijesekera, P., Baokar, A., Tsai, L., Reardon, J., Egelman, S., Wagner, D., & Beznosov, K. (2018). Dynamically Regulating Mobile Application Permissions. *IEEE Symposium on Security and Privacy*, *February*, 64–71. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8283440&casa_token=uzLRW_oFjSkAAAAA:7hQLbS6SwAyypXJ1dvavjTRhpCxatkjlAJsaMINmhbrNQ5F4kgh2ce9HDXTE8TcoNEfhMIj4kQh82A&tag=1

Wong, W. P., Tan, H. C., Tan, K. H., & Tseng, M. L. (2019). Human factors in information leakage: mitigation strategies for information sharing integrity. *Industrial Management and Data Systems*, *119*(6), 1242–1267. https://doi.org/10.1108/IMDS-12-2018-0546

Woollaston, V. (2016). *How to find and use Facebook's Secret messages*. Wired.Co.Uk. https://www.wired.co.uk/article/messenger-secret-messages-end-to-end-encryption

Xie, M., Wu, Z., & Wang, H. (2012). Secure instant messaging in enterprise-like networks. *Computer Networks*, *56*(1), 448–461. https://doi.org/10.1016/j.comnet.2011.09.006

Yasin, A., Fatima, R., Liu, L., Yasin, A., & Wang, J. (2019). Contemplating social engineering studies and attack scenarios: A review study. *Security and Privacy*, *2*(4), 1–14. https://doi.org/10.1002/spy2.73

Zaharia, A., & Cihodariu, M. (2019). *The Best Encrypted Messaging Apps You Should Use Today [Updated 2019]*. Heimdalsecurity.Com. https://heimdalsecurity.com/blog/the-best-encrypted-messaging-apps/

Zhao, J., Masood, R., & Seneviratne, S. (2020). *A Review of Computer Vision Methods in Network Security*. 1–37. http://arxiv.org/abs/2005.03318

Zimmermann, V., & Renaud, K. (2019). Moving from a 'human-as-problem" to a 'human-as-solution" cybersecurity mindset. *International Journal of Human Computer Studies*, *131*(January), 169–187. https://doi.org/10.1016/j.ijhcs.2019.05.005

Appendix A: Content Analysis – Existing Heuristics

Code	Details	References
H	**Heuristics**	
01	8 usability heuristics for instant messaging applications: - **H.01.01:** Visibility of system status and losability/findability of the mobile device - **H.01.02:** Match between system and the real world - **H.01.03:** Consistency and mapping - **H.01.04:** Good ergonomics and minimalist design - **H.01.05:** Ease of input, screen readability, and glanceability - **H.01.06:** Flexibility, efficiency of use, and personalisation - **H.01.07:** Aesthetic, privacy, and social conventions - **H.01.08:** Realistic error management	(Caro-Alvaro et al., 2018)
02	12 usability heuristics for smartphones and mobile applications: - **H.02.01: Visibility of system status –** The device should keep the user informed about all the processes and state changes through feedback and in a reasonable time. - **H.02.02: Match between system and the real world –** The device should speak the users' language instead of system-oriented concepts and technicalities. The device should follow the real-world conventions and display the information in a logical and natural order. - **H.02.03: User control and freedom –** The device should allow the user to undo and redo his/her actions, and provide clearly pointed 'emergency exits' to leave unwanted states. These options should be available preferably through a physical button or equivalent. - **H.02.04: Consistency and standards –** The device should follow the established conventions, allowing the user to do things in a familiar, standard, and consistent way. - **H.02.05: Error prevention –** The device should hide or deactivate unavailable functionalities, warn users about critical actions, and provide access to additional information. - **H.02.06: Minimise the user's memory load –** The device should offer visible objects, actions, and options in order to prevent users from having to memorise information from one part of the dialogue to another. - **H.02.07: Customisation and shortcuts –** The device should provide basic and advanced configuration options, allow definition and customisation of shortcuts to frequent actions. - **H.02.08: Efficiency of use and performance –** The device should be able to load and display the required information in a reasonable time and minimise the required steps to perform a task. Animations and transitions should be displayed smoothly. - **H.02.09: Aesthetic and minimalist design –** The device should avoid displaying unwanted information overloading the screen. - **H.02.10: Help users recognise, diagnose, and recover from errors –** The device should display error messages in a language familiar to the user, indicating the issue in a precise way and suggesting a constructive solution. - **H.02.11: Help and documentation –** The device should provide easy-to-find documentation and help, centred on the user's current task and indicating concrete steps to follow. - **H.02.12: Physical interaction and ergonomics –** The device should provide physical buttons or the equivalent for main functionalities, located in positions recognisable by the user, which should fit the natural posture (and reach) of the user's dominant hand.	(Inostroza et al., 2016)

03	9 usable security heuristics: • **H.03.01: Informative –** All textual content must be brief, informative, and passable. Demonstrative non-textual artifacts must be described in a way that is meaningful to the user. • **H.03.02: Reliable –** The current state of security/privacy and related functions must be explicitly available. All security information must be described in plain language with no jargon. • **H.03.03: Recognisable –** The interface must be distinguishable and organised in a way that reflects users' expectations. All functionalities are clearly available and traversable. • **H.03.04: Assistive –** Users are guided through decisions to be made. Error prevention conventions are in place. Users can recognise, diagnose, and correct mistakes. Defaults are appropriate and can be modified within reasonable confines. • **H.03.05: Functional –** The site works as expected in a quick and complete manner. No functionalities impede on users' goals nor security/privacy. • **H.03.06: Controllable –** The site is compatible with assistive technology. The interface offers robust and customisable means to protect users with various needs. • **H.03.07: Responsive –** All actions, errors, and threats are effectively communicated without interrupting users' workflow. Users can identify when a task is completed. • **H.03.08: Diverse –** All content and context are communicated in a way that can accommodate various abilities. Satisfactory alternatives, both visually and aurally, are clearly available. • **H.03.09: Memorable –** All system functions and related user actions require a low cognitive load. The system is designed for learnability and evokes high recall abilities.	(Napoli, 2018)
04	10 usability heuristics for user interface design: • **H.04.01: Visibility of system status –** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. • **H.04.02: Match between system and the real world –** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. I should follow real-world conventions, making information appear in a natural and logical order. • **H.04.03: User control and freedom –** Users often choose system functions by mistake and will need a clearly marked 'emergency exit' to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. • **H.04.04: Consistency and standards** – Users should not have to wonder whether different words, situations, or actions mean the same thing. Platform conventions should be followed. • **H.04.05: Error prevention –** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. • **H.04.06: Recognition rather than recall –** Minimise the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. • **H.04.07: Flexibility and efficiency of use –** Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such	(Nielsen, 1995)

	that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. • **H.04.08: Aesthetic and minimalist design –** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. • **H.04.09: Help users recognise, diagnose, and recover from errors –** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. • **H.04.10: Help and documentation –** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.	
05	13 usability heuristics for mobile applications: • **H.05.01: Visibility of System Status –** The application should keep the user informed about all processes and state changes within a reasonable period of time. • **H.05.02: Correspondence between the Application and the Real World –** The application must speak the language of the users and not in technical terms of the system. The application must follow the conventions of the real world and display the information in a logical and natural order. • **H.05.03: User Control and Freedom –** The application should allow the user to undo and redo their actions for clear navigation and should provide the user with an option to exit undesirable system states. • **H.05.04: Consistency and Standards –** The application must follow the established conventions, allowing the user to perform their tasks in familiar, standardised, and consistent manner. • **H.05.06: Error Prevention –** Eliminate error prone conditions and give the user a confirmation option with additional information before committing to the action. • **H.05.07: Minimise the User's Memory Load –** The application should provide visible objects, actions, and options to prevent users from having to memorise information from one interface to another. • **H.05.08: Customisation and Shortcuts** – The application should provide basic and advanced settings for setting and customising shortcuts for frequent actions. • **H.05.09: Efficiency of Use and Performance –** The device must be able to load and display information in a reasonable period of time and minimise the steps required to perform a task (number of steps to be taken by the user to reach a goal). Animations and transitions should display smoothly and smoothly. • **H.05.10: Aesthetic and Minimalist Design –** The application should avoid displaying unwanted information that overwhelms the screen. • **H.05.11: Help Users Recognise, Diagnose, and Recover from Errors –** The application should display error messages in a language familiar to the user, accurately indicating the problem and suggesting a constructive solution. • **H.05.12: Help and Documentation –** The application should provide easy-to-find documentation and help centring on the user's current task and indicating concrete steps to follow. • **H.05.13: Pleasant and Respectful Interaction with the User –** The device should provide a nice iteration with the user so that the user does not feel uncomfortable while using the application. • **H.05.14: Privacy –** The application must protect the user's sensitive data.	(Da Costa & Canedo, 2019)

06	14 usability heuristics: • ***User guidance*** o **H.06.01: Prompting –** Guide users towards taking specific actions. Show the selectable options. Include a title or header for the content (window, web page). Keep the user informed about the system status. Provide associate labels, required formats, and acceptable values for data fields. o **H.06.02: Feedback –** Provide appropriate feedback as a response to user's actions within reasonable time. Provide feedback on user actions (data entries, commands). Inform the user of the current state of processing. Provide immediate feedback. o **H.06.03: Information architecture –** Provide a clear structure of the application. Provide adequate structuring of web pages. Avoid redundant content. Show the navigation history. o **H.06.04: Grouping/distinction –** Provide means to group similar objects and distinguish between different classes of objects. Provide means to understand whether objects belong to a given class. Group similar objects together. Use similar formatting and graphical features for similar objects. Provide a clear distinction between the screen areas having different functions. Mark the currently selected option. • ***User effort*** o **H.06.05: Consistency –** Provide similar meanings and design choices in similar contexts. Provide similar phrasing, text justification, colour, and punctuation. Display similar objects (windows, menus, exit buttons, etc.) in the same way and at the same location. Provide similar procedures for similar functions and tasks. Follow platform conventions. o **H.06.06: Cognitive workload –** Provide means to the users' perceptual and cognitive load. Provide means to facilitate recognition rather than recall. Make the information legible. Reduce the information density. Reduce the demands on the working memory (magical number seven plus or minus two). Allow users short data entries. Provide automated computation of derived data. o **H.06.07: Minimal actions –** Minimise the number of actions needed to accomplish a task goal. Minimise the number of steps for selecting a menu item. Provide shortcuts for advanced users. Provide a search engine on websites. • User control and freedom o **H.06.08: Explicit user actions –** Ensure that only actions requested by the users are processed and only when these are requested. Require an explicit ENTER action to initiate processing. Provide a dual activation when the selection is accomplished by pointing. o **H.06.09: User control –** Provide the means to initiate and control the system processing. Allow users to interrupt, resume or cancel the system processing. Allow users to select and sequence the tasks. Allow users to arrange the windows on the screen. o **H.06.10: Flexibility –** Provide means to customise the interface and select the preferred way to accomplish a goal. Provide different dialogue types for different users. Provide alternative paths to perform a task. Allow experienced users to bypass a menu selection. • ***User support*** o **H.06.11: Compatibility with the user –** Provide means to match the users' characteristics with the characteristics of the user interface. Speak the user language and use real-world conventions. Provide an accessible user interface for users with disabilities. Respect culturally related requirements (calendar, measurement units, design conventions, and language).	(Pribeanu, 2017)

	○ **H.06.12: Task guidance and support –** Provide the user with the procedure and associated support (forms, documents, etc.) needed to perform specific tasks. Provide a procedure describing the steps a user must follow. Provide additional support such as downloadable forms and explanatory notes. Whenever possible, provide a unique entry point, in order to guide the user throughout a lengthy process. ○ **H.06.13: Error management –** Provide means to prevent, diagnose, correct, and recover from errors. Provide means to detect and prevent errors. Provide clearly phrased, polite, and informative error messages. Provide means to correct errors. ○ **H.06.14: Help and documentation –** Provide online help and documentation. Provide contextual help. Provide a user manual. Provide a general presentation of the system.	
07	14 social network user experience heuristics: • **H.07.01: Visual feedback and social network status –** The social network must inform the status of the user application in response to the actions that he/she performs. • **H.07.02: Match between the social network and real world –** The social network should use a language familiar and understandable to the user and use icons that clearly represent their meaning. • **H.07.03: User control and freedom –** The user must feel that he/she manages the social network, being able to undo or redo his/her actions and use the social network freely. • **H.07.04: Consistency and standards in multiplatform –** The social network must be consistent in the several platforms that support it. The same functionalities must be present; there should not be differences (e.g., visual differences, behaviour differences, etc.), and the standards must be followed for each platform. • **H.07.05: Error prevention –** The social network must prevent errors from occurring, providing warning messages to the user with useful information and without technical terms. • **H.07.06: Minimise the user's memory load –** The social network must minimise the user's memory load, without forcing him/her unnecessarily to remember information. • **H.07.07: Aesthetic and minimalist design –** The social network should show only the relevant elements for the user, without overloading the interface with less usual functionalities. • **H.07.08: Flexibility and customisation –** The social network should allow configuring frequent actions and be flexible to adapt its interface based on the users' preferences and their interests. • **H.07.09: Help users recognise, diagnose, and recover from errors –** The social network should help the user to recover from errors by indicating the problem and suggesting a solution. • **H.07.10: Help centre –** The social network should provide help and documentation on how it works, providing accurate information, and oriented on the tasks performed by the user. • **H.07.11: Perception and user status –** The social network should allow the user to perceive if other users are available to interact and/or communicate. In addition, the social network must allow the user to define how and when his/her status is perceived by other users in the network. • **H.07.12: Control the published content –** The social network should control the published content not to affect the sensitivity of users, through filters and regulations. The user should be able to report content published by other users in the network, indicating the reason.	(Quiñones et al., 2020)

	• **H.07.13: Privacy control –** The social network should allow the user to have control over the information that he/she wants to share and who can access it. • **H.07.14: Security and recovery of user account –** The social network must include security measures, account recovery account protection, and personal data of the user.	
08	11 usability and user experience heuristics for social network: • **H.08.01: Visual feedback and system status –** The social network must inform the user of the status of the system after any action taken by the him or her. • **H.08.02: User control and freedom** – The social network should allow the user undo and redo actions; user should always feel in control. • **H.08.03: Consistency and standards in multiplatform** – There should be no visual or functional differences between the various platforms delivered by the same social network, to the extent that user interaction is influenced. • **H.08.04: Prevention and recovery of errors –** The social network must prevent and avoid errors in use of the system through warning messages that deliver the right information, without too much technicality that may confuse the user. • **H.08.05: Minimise user memory load –** User should not have to remember information that he/she already provided. • **H.08.06: Aesthetic and minimalist design** – The social network must show an aesthetic design that includes only the elements relevant to the user in a certain context of use. • **H.08.07: Help centre** – The social network must have a space where users can resolve their doubts about the system; the help information must be brief, accurate, and user-centred. • **H.08.08: User perception and status –** The system must allow the user to configure, at any time, whether he/she is available (or not) to communicate; user must easily perceive other users' availability. • **H.08.09: Control of published content** – The social network must control the content that publishes so as not to affect the sensitivity of users, through filters and regulations; the user must be able to denounce/report content published by other users on the network, indicating the reason. • **H.08.10: Customisation and configuration settings** – The user must be able to adjust the different settings provided by the social network and customise the space it provides. • **H.08.11: Security and user account recovery –** The social network must include security measures, protection of the user's account and personal data; it must also provide an account recovery option.	(Saavedra et al., 2019)

Appendix B: Content Analysis – Existing Guidelines

Code	Details	References
G	**Guidelines**	
01	7 Larger usability guidelines for smartphone applications: • **G.01.01: Design** – To grasp the user attraction an app should be aesthetically pleasant. The use of colour(s) and object(s) should attract the user. An attractive interface will mean more traction, but it is not limited to colours, artwork it is also related to integration of app function with its appearance • **G.01.02: Navigation** – Navigation refers to the mechanism of moving from one screen to another and set of actions to complete a specific task. Navigation includes the usage of buttons, menu tabs, links and images that leads you from one point to another within an app to perform set of actions. Various researchers emphasise on the importance of navigation for making an app useful, but ensuring effective navigation is a challenging task for mobile devices because of display limitations. Many researchers have proposed navigation guidelines to overcome this challenge, which are grouped in following subsections. • **G.01.03: Content –** Content refers to information communicated to user/s. Content includes all expressive material either in the form of text or multimedia. Some apps do not require much content, but few apps are specifically content-based apps, such as internet portal and newspaper apps, etc. Independent of the amount of content on an app, it requires special attention from developers. • **G.01.04: Cognitive Load –** Cognitive load refers to the total amount of mental effort in working memory. Working memory is the system responsible for processing information; it helps in the reasoning, learning, and understanding process. Instructional design should minimise the cognitive load, as higher cognitive load may lead to error/s. Similarly, minimising cognitive load can maximise users' satisfaction and performance. • **G.01.05: Equitable Use –** An app cannot be considered useful if it is not equally usable for all type of users. These differences are attributed either to users themselves or to the mobile devices they are using. An app should be capable of catering to these differences to meet the usability goal. • **G.01.06: Error Handling –** There is always a chance of error in human-developed software. • **G.01.07: Input Method –** Owing to small keyboards it is very difficult for users to provide input in mobile devices. Input methods available for mobile devices are different from desktop devices and require a certain level of aptitude. This problem increases the rate of erroneous input.	(Ahmad et al., 2018)
02	14 user experience guidelines for mobile applications: • **G.02.01:** Learnability • **G.02.02:** Effectiveness • **G.02.03:** Efficiency • **G.02.04:** User satisfaction • **G.02.05:** User error protection • **G.02.06:** Memorability • **G.02.07:** Cognitive Load • **G.02.08:** Demand of user attention	(Bajenaru et al., 2018)

	• **G.02.09:** Human–computer interaction • **G.02.10:** Navigation • **G.02.11:** Mobile context • **G.02.12:** Security • **G.02.13:** Support • **G.02.14:** Installation	
03	6 warning design guidelines: • **G.03.01: Describe the risk comprehensively –** Warnings are meant to alert the user of an impending risk to her information or her identity. Whenever a warning is used, the risk that motivates the usage of a warning should be identified and presented clearly. • **G.03.02: Be concise and accurate –** Warnings always interrupt the user. If too long, overly technical, inaccurate, or ambiguous, a warning will simply be discarded, and its purpose will be lost. • **G.03.03: Offer meaningful options –** Warnings should present understandable choices, and enough information to decide between them. • **G.03.04: Present relevant contextual information –** In most contexts that require a warning to be shown, a computer or software system cannot make a decision on behalf of the user. Warnings should present relevant contextual information that allows the user to make an informed decision. • **G.03.05: Present relevant auditing information –** In some contexts, actions have been performed in the past that may help a user to understand the risks associated with the choice s/he needs to make. In such cases, relevant auditing information should be presented. • **G.03.06: Follow a consistent layout –** Warnings that follow a common visual layout can be recognised faster. We suggest a common layout based on the Human Interface Guidelines (HIG) of the most broadly used operating systems.	(Bauer et al., 2013; Data and Application Security Group TH Köln, 2019b; Gorski et al., 2019)
04	5 computer security dialogue guidelines: • **G.04.01: Follow a visually consistent layout –** Use one icon; do not use a close button; use command links for options; use a primary text to explain the risk; describe the consequences of each option below each button. • **G.04.02: Comprehensively describe the risk –** Describe the risk; describe consequences of not complying; provide instructions on how to avoid the risk. • **G.04.03: Be concise, accurate, and encouraging –** Be brief; avoid technical jargon; provide specific names, locations and values for the objects involved in the risk; do not use strong terms (e.g., abort, kill, fatal). • **G.04.04: Offer meaningful options –** Provide enough information to allow the user to decide; option labels should be answers to explicit question asked to the user; if only one option is available, do not show the warning; the safest option should be the default. • **G.04.05: Present relevant contextual and auditing information –** If the warning was triggered by a known application, describe the application; identify agents involved in the communication by name; if user's information is about to be exposed to risk, describe what the information is and how it will be exposed.	(Bravo-Lillo et al., 2011; Data and Application Security Group TH Köln, 2019a; Gorski et al., 2019)
05	11 usability guidelines for instant messaging applications: • **G.05.01:** Main features should be easy to access. • **G.05.02:** Automatic display of the keyboard at new chats.	(Caro-Alvaro et al., 2018)

	• **G.05.03:** Add a new contact only with the ID. • **G.05.04:** Do not tolerate unrecoverable errors. • **G.05.05:** Keep the top status bar always visible. • **G.05.06:** Provide account recovery features. • **G.05.07:** User interface adapted to and limited by the operating system. • **G.05.08:** Avoid half translations. • **G.05.09:** Provide visual distinction between individual and group chats. • **G.05.10:** Design the interface carefully and accurately. • **G.05.11:** Provide security mechanisms and information to the user.	
06	7 user interface guidelines: • ***Visual Design:*** ○ **G.06.01: Colour and Contrast –** Use colour judiciously; use one primary colour throughout the application and add a secondary colour to highlight important (interactive) elements; avoid using colour as a standalone indicator to communicate the state of an element; avoid red-green and blue-yellow colour combinations. ○ **G.06.02: Typeface –** Use sans-serif instead of serif typefaces; Emphasise important content by applying type variations; apply type variations conservatively; avoid using only uppercase letters in longer texts. ○ **G.06.03: Iconography –** Avoid using icons, if it takes too long to think of an appropriate icon; use icons in size limited display areas over plain text labels; make use of platform-specific icon sets; apply icons in a consistent way; provide small textual labels at the bottom or besides icons. ○ **G.06.04: Terminology –** Keep words and phrases simple and informative; keep the tone of language polite, positive and user-centred; avoid technical or domain specific jargon; write phrases in second person conversational style, avoid first person; use action verbs for labelling interactive elements. • ***Interaction Design:*** ○ **G.06.05: Gestures –** Avoid using too complex gestures, rely on standard gestures; avoid using standard gestures for non-standard actions; Provide discoverable shortcuts to supplement gestures; Make gestures reversible. ○ **G.06.06: Data Entry –** Keep data entry tasks requiring keyboard input to a minimum; provide alternative input forms enabling users to choose from a set of available options; prepopulate input fields where possible; make input fields easily discoverable; provide labels and placeholders to communicate the purpose of an input field; display suitable keyboard layouts for different input types; always communicate the current state of an input field. • ***Navigation:*** ○ **G.06.07: Navigation –** Implement navigation in a way that supports the user in reaching desired content or functionality with ease; avoid the navigation structure to become too deep; always indicate the user's current location inside the application; always equip sub-level views with a 'Back/Up' button to indicate the possibility to return to its parent view.	(Martin, 2018)
07	12 usable security guidelines: • **G.07.01: Visibility –** The visibility of the system status that lets users know exactly what the capabilities of the system are, is one of the most important principles addressed by both usability and usable security scholars.	(Mujinga et al., 2019)

	• **G.07.02: Learnability –** The ability of users to use an application or user interface efficiently and effectively for the first time, as well as subsequent reuse, depends on the ease of learning the system. • **G.07.03: Satisfaction –** Satisfaction is one of the five characteristics of usability identified by Nielsen (2010). Essentially, satisfaction in the use of a system extends beyond usability and into the realm of user experience (UX). • **G.07.04: Errors –** The principle of errors, be it for their prevention in the first place or recovery after they have occurred, is a critical design principle in all systems. • **G.07.05: Availability –** Availability is essential for online applications, which are often marketed as providing convenience by allowing users to access the service 24 hours a day. In the real world, a certain period of system downtime is expected for reasons such as system upgrades and maintenance, but these activities should be scheduled during off-peak times and kept to a minimum in terms of the frequency and duration of downtime. • **G.07.06: Revocability –** Users should be able to undo actions and errors, and a secure and usable system should give warning and confirmation of actions that are irreversible. Although some actions cannot be reversed after a certain stage of processing, developers need to try, by all means possible, to provide support for 'undo' and 'redo' functions. • **G.07.07: Expressiveness –** The system should inform and guide users through security features and yet allow freedom of expression. The system's information security policy must not be too rigid and difficult for users to comply with. • **G.07.08: User language –** User language is another principle that is relevant in both the usability and usable security contexts. The principle requires the system to speak the users' language, using terms and concepts familiar to users, while avoiding the use of technical terms. • **G.07.09: User suitability –** User suitability ensures that the system provides options suitable for users with diverse levels of skill and experience in security. • **G.07.10: Help and documentation –** Users may need assistance, especially for applications that have been developed for a diverse group of users with different levels of skill. Support material that helps new users and system documentation for reference during usage are critical. • **G.07.11: Security –** The system should ensure a trusted path through the communication channel (usually the internet) between the end-user device and trusted servers, addressing fundamental InfoSec principles, such as confidentiality, integrity, and availability, to avoid disclosure and unauthorised access of information assets in storage and in transit. • **G.07.12: Privacy –** Organisations collect personal information about their customers, some of which is sensitive, such as credit card numbers. Hence, the system should protect information provided by users against access by unauthorised parties, and it should be used only for the purposes for which it was collected in the first place.	
08	19 security guidelines for mobile applications: • **G.08.01: Operational policy for mobile apps –** The operational policy for mobile apps describes the way that the organisation deals with designing apps and making them available. The operational policy is a	(National Cyber Security Centre (NCSC-NL), 2018)

more concrete elaboration of the higher-level policy. A solid operational policy is therefore a prerequisite for the safe design of an app and its environment.

- **G.08.02: Secure server-side application –** The app on the mobile device usually forms a chain together with the application on the server side. Safe use of the app is only possible in combination with a secure server-side application. The absence of a secure server-side application or secure server will result in the inability to build a secure chain and thus an environment that is not suitable for storing and exchanging confidential information.
- **G.08.03: Third-party apps –** Apps often work together with other apps such as viewers and keyboards. These apps usually come from other suppliers and are referred to as ‘third-party apps’. Such apps process information outside the app, which means that the protection of the information lies beyond the developer’s control. These apps may have hidden functionality, allowing access to confidential information even if it is protected by the app.
- **G.08.04: Secure code on delivery –** App development can be accelerated by using external code libraries. However, these libraries may contain vulnerabilities or malware. Information about the libraries used and the way that the app works can provide the attacker with information on weaknesses in the app.
- **G.08.05: Secure operation of the app –** Unlike server-side applications, apps are not run in a familiar environment but on the mobile device itself. This situation allows an attacker to obtain the full source code: the binary and the running app. As a result, the attacker is able to control each piece of code during each phase of the app’s program, plus the information stored in the binary files of the app, such as configuration files. Manipulation of the app’s operation by malicious parties can therefore only be limited by electronically protecting the binary and the running app.
- **G.08.06: Storage location –** The choice where to store data will to a large extent determine the options when it comes to protecting the data from unwanted access. The choice of where to store each data item or set must therefore be made consciously. Because better protected locations offer greater security, the basic principle in this security guideline is that the safest location should be chosen for storage, unless it is certain that a less secure location can be demonstrably suitably secured.
- **G.08.07: Storage on the mobile device –** Sensitive data can be protected by using cryptographic techniques. Cryptographic techniques are the only way to protect information efficiently when it is physically accessible. Which data are sensitive or confidential must be determined by the organisation. Confidentiality is determined as part of determining the location of the storage.
- **G.08.08: Unnecessary information in RAM –** The temporary storage of confidential information in the mobile device’s memory is unavoidable for most apps. However, attacks are known where a memory dump is made while the device is locked. Confidential information stored in the memory at that time (such as PINs) can then be obtained by an attacker. Confidential information is also accessible after a crash, for example.
- **G.08.09: User session timeouts –** While the user is accessing the app, a user session is open. During this session, information is accessible via the app. Other people may abuse this accessibility. Session termination

ensures that the session ends after a prescribed time interval of inactivity.

- **G.08.10: Logging –** While logging, user actions and messages about the app's operation can be recorded in log files. Logging can be used to track security incidents and errors in the operation of the app, but sensitive information can also end up in the log files. If this information falls into the wrong hands, not only would this situation result in the privacy of the user being at risk, but the information could also be used to find security weaknesses in the app. Access to this sensitive log information should therefore be prevented.
- **G.08.11: Transport encryption –** Encryption of data transport between the server component of the application and the client component of the application protects confidential data. Mobile devices often use open and therefore unsafe Wi-Fi networks. As a result, the communication is sensitive to man-in-the-middle attacks. Protecting the session by encryption through TLS, even when the information exchanged is not confidential, is nowadays a standard requirement for securing mobile devices. Networks that are deemed insecure must be encrypted. Unsafe networks are networks that are not protected against unauthorised access. This category includes corporate networks in offices that are not demonstrably protected, physically and electronically. As mobile devices are also used over non-secure and public networks, the apps use encryption for secure communication.
- **G.08.12: Certificate pinning –** One of the most important security measures for apps is to secure communication with the server through encryption. Certificates for encryption are issued via a trusted Public Key Infrastructure (PKI), where a certificate is issued by a certificate authority (CA). Usually, there is a root CA and several intermediate CAs that issue the certificates. The final certificate issued contains the specific URL for which the certificate was issued.
- **G.08.13: App hardening –** When the app is hardened, the app's communication capabilities are kept to a minimum (only what is strictly necessary). One of the ways to achieve this situation is by removing or deactivating unnecessary interfaces. By taking stock of the necessary interfaces and then determining the dependencies, a minimum list of interfaces that the app needs to have at its disposal can be compiled. All other interfaces can be removed. Keep in mind that inactive interfaces still present on a system can ultimately lead to a vulnerable app. It is therefore safer to keep the attack surface as small as possible. To this end, unnecessary interfaces and access rights are preferably removed.
- **G.08.14: Principle of least privilege for other apps –** System abuse risks can be significantly reduced by limiting rights on the app. Common policy principles include those based on 'standard no access', 'least privilege' and 'need-to-know'. This measure applies to users but also to apps among themselves. According to the principle of 'least privilege', the rights on an app are limited to the minimum set of rights needed for it to function properly.
- **G.08.15: Input standardisation –** As with all software, apps are highly dependent on a variety of inputs, such as user input, data received from external servers, other apps, and local files. Apps depend heavily on internet standards such as JSON, XML, SQL, HTML, and JavaScript. Input may contain characters or commands that affect the operation of the app. Such input does not comply with the rules for secure input. Just as server-side apps and/or web apps must be protected, so must

	mobile apps. If the app does not handle malicious input correctly, certain input can be used in order to gain access to the data that should be protected by the app. • **G.08.16: Input validation –** The most important rule of thumb for input in an app is that the application may not trust any input and must therefore validate all input for accuracy, completeness and validity. At a minimum, the input should be validated for values outside the valid range (limit values), invalid characters, missing or incomplete data, data that do not conform to the correct format and inconsistency of data with respect to other data within the input or in other data files. Non-trusted input can reach the app from all kinds of different attack vectors, such as intents, services, network traffic, binding interfaces and access to files. Input validation is the most important condition for reliable data processing and invalid input must be rejected by the app. • **G.08.17: HTTP methods** – The back-end web server supports the HTTP protocol. HTTP has methods, headers and error information that may be misused. As a consequence, its use is limited to the minimum necessary for the functioning of the accessible apps. • **G.08.18: XML External Entity injection –** In addition to JSON, XML is a commonly used format to read data into the app. The XML-based input contains data surrounded by codes in a specific structure: in XML, the entities (data) are shown between an opening tag and a closing tag. • **G.08.19: Up to date apps –** Older versions of apps can contain vulnerabilities that are often widely known to attackers and that can be abused. Keeping the apps up to date is therefore an important condition for keeping the app safe.	
09	OWASP Mobile Risks Top 10 – 2016: • **G.09.01: Improper Platform Usage –** This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. • **G.09.02: Insecure Data Storage –** Threat agents include the following: an adversary that has attained a lost/stolen mobile device; malware or another repackaged app acting on the adversary's behalf that executes on the mobile device. • **G.09.03: Insecure Communication –** When designing a mobile application, data is commonly exchanged in a client-server fashion. When the solution transmits its data, it must traverse the mobile device's carrier network and the internet. Threat agents might exploit vulnerabilities to intercept sensitive data while it is traveling across the wire. • **G.09.04: Insecure Authentication –** Threat agents that exploit authentication vulnerabilities typically do so through automated attacks that use available or custom-built tools. • **G.09.05: Insufficient Cryptography –** Threat agents include the following: anyone with physical access to data that has been encrypted improperly, or mobile malware acting on an adversary's behalf. • **G.09.06: Insecure Authorisation –** Threat agents that exploit authorisation vulnerabilities typically do so through automated attacks that use available or custom-built tools. • **G.09.07: Client Code Quality –** Threat Agents include entities that can pass untrusted inputs to method calls made within mobile code. These types of issues are not necessarily security issues in and of themselves but lead to security vulnerabilities. For example, buffer overflows	(OWASP, 2016)

	within older versions of Safari (a poor code quality vulnerability) led to high-risk drive-by Jailbreak attacks. Poor code-quality issues are typically exploited via malware or phishing scams. • **G.09.08: Code Tampering –** Typically, an attacker will exploit code modification via malicious forms of the apps hosted in third-party app stores. The attacker may also trick the user into installing the app via phishing attacks. • **G.09.09: Reverse Engineering –** An attacker will typically download the targeted app from an app store and analyse it within their own local environment using a suite of different tools. • **G.09.10: Extraneous Functionality –** Typically, an attacker seeks to understand extraneous functionality within a mobile app in order to discover hidden functionality in in back-end systems. The attacker will typically exploit extraneous functionality directly from their own systems without any involvement by end-users.	

Appendix C: Content Analysis – Existing Standards

Code	Details	References
S	**Standards**	
01	Information security standards for telecommunication organisations, including but not limited to: • **S.01.01: Addressing security in third-party agreements –** Agreements with third parties involving accessing, processing, communicating, or managing the organisation's information or information processing facilities, or adding products or services to information processing facilities should cover all relevant security requirements. • **S.01.02: Ownership of assets –** Assets maintained in the inventory should be owned. • **S.01.03: Acceptable use of assets control –** Rules for the acceptable use of information and of assets associated with information and information processing facilities should be identified, documented, and implemented. • **S.01.04: Classification guidelines –** Information should be classified in terms of its value, legal requirements, sensitivity, and criticality to the organisation. • **S.01.05: Information labelling and handling –** An appropriate set of procedures for information labelling should be developed and implemented in accordance with the information classification scheme adopted by the organisation. • **S.01.06: Controls against malicious code –** Detection, prevention, and recovery controls to protect against malware should be implemented, combined with appropriate user awareness. • **S.01.07: Controls against mobile code control –** Where the use of mobile code is authorised, the configuration should ensure that the authorised mobile code operates according to a clearly defined security policy, and unauthorised mobile code should be prevented from executing. • **S.01.08: Back-up –** Backup copies of information, software and system images should be taken and tested regularly in accordance with an agreed backup policy. • **S.01.09: Audit logging control –** Audit logs recording user activities, exceptions, and information security events should be produced and kept for an agreed period to assist in future investigations and access control monitoring.	(International Organization for Standardization, 2013a; South African National Standard, 2009)
02	Standards for security and privacy controls, including but not limited to: • **S.02.01: Access enforcement** – Enforce approved authorisations for logical access to information and system resources in accordance with applicable access control policies. • **S.02.02: Information flow enforcement** – Enforce approved authorisations for controlling the flow of information within the system and between connected systems based on organisation-defined information flow control policies. • **S.02.03: Least privilege** – Employ the principle of least privilege, allowing only authorised accesses for users (or processes acting on behalf of users) that are necessary to accomplish assigned organisational tasks. • **S.02.04: Unsuccessful logon attempts -** Enforce a limited organisation-defined number of consecutive invalid logon attempts by a user during an organisation-defined time period; and automatically lock the	(NIST SP800-53, 2020)

	account or node for an organisation-defined time period; lock the account or node until released by an administrator; delay next logon prompt per organisation-defined delay algorithm; notify system administrator; take other organisation-defined action when the maximum number of unsuccessful attempts is exceeded. • **S.02.05: System use notification** – Display organisation-defined system use notification message or banner to users before granting access to the system that provides privacy and security notices consistent with applicable laws, executive orders, directives, regulations, policies, standards, and guidelines. • **S.02.06: Previous logon notification** – Notify the user, upon successful logon to the system, of the date and time of the last logon. • **S.02.07: Concurrent session control** – Limit the number of concurrent sessions for each organisation-defined account and/or account type to organisation-defined number. • **S.02.08: Device lock** – Prevent further access to the system by initiating a device lock after organisation-defined time period of inactivity; requiring the user to initiate a device lock before leaving the system unattended; and retain the device lock until the user re-establishes access using established identification and authentication procedures. • **S.02.09: Session termination** – Automatically terminate a user session after organisation-defined conditions or trigger events requiring session disconnect. • **S.02.10: Security and privacy attributes** – Provide the means to associate organisation-defined types of security and privacy attributes with organisation-defined security and privacy attribute values for information in storage, in process, and/or in transmission. • **S.02.11: Wireless access** – Establish configuration requirements, connection requirements, and implementation guidance for each type of wireless access; and authorise each type of wireless access to the system prior to allowing such connections.	

Appendix D: Content Analysis – Existing Best Practices

Code	Details	References
B	**Best Practices**	
01	12 challenges and best practices in mobile application development: • **B.01.01:** Compatibility with various platforms • **B.01.02:** Incongruity of hardware utilities • **B.01.03:** Improper estimation of requirements • **B.01.04:** Total cost and scheduled time • **B.01.05:** User convenience • **B.01.06:** Front end design • **B.01.07:** Input methods • **B.01.08:** Accessing data • **B.01.09:** Developing worthy applications • **B.01.10:** Difficulty in testing • **B.01.11:** Targeting users • **B.01.12:** Privacy • **B.01.13:** Security	(Ajit Kumar et al., 2016)
02	9 best practices for mobile application development: • **B.02.01: Planning –** Determining the design methodology is a key to mobile application development, particularly in a cross-platform environment where multiple efforts may be ongoing simultaneously. Therefore, it is recommended to design a proper plan before starting programming the application, which may have considered an efficient strategy to attract the potential clients. • **B.02.02: Requirements –** It considers the most important phase in which the discussion regarding the business plan will take place after collecting, analysing, and documenting the client's requirement. The planning strategy must comprise the best method of user interaction, performance, and the utilisation of the limited resource; after that, frequent and rapid iterations of requirements reviews are conducted. • **B.02.03: Design and architecture –** Designing the most proper architecture for mobile applications must be considered by developers. Some studies consider the best way is to develop a layered application, in which the consistency of mobile app functionality is guaranteed across all platforms. It is also recommended to create re-usable platform components that can help to accelerate the development time. • ***User Experience:*** ○ **B.02.04: Guidelines**: Defining and specifying the font usage, colours, layout, pictures, etc., is fundamental to UX. These guidelines must also describe predictable behaviours under some conditions like (user walk-away, network changing, timeouts, etc.). ○ **B.02.05: App branding**: Regular branding inside an application and throughout an organisation's portfolio forms a sense of familiarity and attachment with the client. • **B.02.06: Development –** Despite which mobile platform an organisation selects for mobile application development, a universal set of development process best practices is identified from literature as follows: ○ The appropriate level of code documentation will increase the code readability, and this becomes more important when the development team grows.	(Aldayel & Alnafjan, 2017)

	o The development team must have the proper knowledge and expertise in the exact mobile platforms being targeted. o Iterative development of mobile application by having short iterative cycles with continuous delivery, in order to get user feedback earlier in the development process, which can simplify the process of prioritising and developing these changes. o Peer code review to ensure that developers are following the well-known coding standard and for identifying defects in code. • **B.02.07: Testing –** Simulators and 'On Device' testing: applications should be tested early using simulators to measure the usability and performance, and final testing must be conducted on real mobile devices. • **B.02.08: Deployment –** There are a few aspects that organisations should consider towards the deployment of mobile applications including: o The plan should involve a well-defined release cycle for deploying mobile applications and an explicit description of the hosting environment if needed such as (test, development, production, etc.). o Automating the process of configuring and installing will save time and eventually result in saving significant cost • **B.02.09: Maintenance and support –** Maintenance phase deals with fixing various issues that were faced by the users owing to compatibility or software and hardware constraints, which were not identified during the testing phase; it also involves developing/releasing new features and functionalities. • ***Other aspects:*** o **B.02.10: Security**: Mobile devices are vulnerable, and these devices must incorporate the inherited security capabilities from the mobile platform as well as using appropriate and up-to-date security tools for protecting sensitive data. This can be handled by utilising: access control using enterprise authentication, securing web services/APIs, data encryption (at-rest and in-transit), etc. o **B.02.11: Privacy Policy**: Privacy policy should be clear, transparent, and not ambiguous, covering the data collection, data sharing, and use practices. Mobile developers should present the privacy policy about the components' use across different platforms, to facilitates its maintainability. In order to create effective policy, it is recommended to communicate effectively and openly, use clear and simple language, offer users some controls and choices regarding their acceptances or not, protect users' data, and ensure accountability.	
03	Best practices and examples of risk controls include: • **B.03.01: Writing programs for people rather than for machines wherein:** o a program must not require the users to take in too many facts at once. o names are meaningful, distinctive, and consistent. o the formatting and style of the code are consistent. • **B.03.02: Letting the computer do the job by:** o making the computer able to repeat tasks. o saving in a file the recent commands for reuse. o automating workflows through the use of a build tool. • **B.03.03: Making incremental changes through:** o continuous feedback and correction of course while working in small steps. o the use of a version control system. o putting in version control all that have been manually created.	(Alenezi & Almuairfi, 2019)

	• **B.03.04: Not repeating oneself or others in which:** ○ the system must have a single authoritative representation for each data. ○ codes are modularised instead of being copied and pasted. ○ codes are reused rather than being rewritten. • **B.03.05: Planning for mistakes by:** ○ adding assurance that how programs operate will be checked. ○ utilising an existing and reliable testing unit. ○ conducting test activities for bugs. ○ using a symbolic debugger. • **B.03.06: Making effective use of software only once it is proven that it functions correctly by:** ○ identifying bottlenecks through the use of a profiler. ○ writing code in the best language level possible. • **B.03.07: Documenting the purpose and design rather than the mechanics wherein:** ○ interfaces and reasons are recorded. ○ code is refactored to provide an explanation of how it works. ○ the documentation is embedded for a piece of software. • **B.03.08: Collaborating through:** ○ the use of pre-merge code reviews. ○ the use of pair programming when distinctly tricky problems are tackled and when someone new is being brought up to speed. ○ the use of a problem-tracking tool.	
04	Standards to mitigate the risk of software vulnerabilities: • ***Prepare the Organisation:*** ○ **B.04.01: Define security requirements for software development –** Ensure that security requirements for software development are known at all times so that they can be taken into account throughout the SDLC and duplication of effort can be minimised because the requirements information can be collected once and shared. This includes requirements from internal sources (e.g., the organisation's policies, business objectives, and risk management strategy) and external sources (e.g., applicable laws and regulations). ○ **B.04.02: Implement roles and responsibilities –** Ensure that everyone inside and outside of the organisation involved in the SDLC is prepared to perform their SSDF-related roles and responsibilities throughout the SDLC. ○ **B.04.03: Implement a supporting toolchain –** Use automation to reduce the human effort needed and improve the accuracy, consistency, and comprehensiveness of security practices throughout the SDLC, as well as provide a way to document and demonstrate use of these practices. Toolchains and tools may be used at different levels of the organization, such as organisation-wide or project-specific. ○ **B.04.04: Define criteria for software security checks –** Help ensure that the software resulting from the SDLC meets the organisation's expectations by defining criteria for checking the software's security during development. • ***Protect Software:*** ○ **B.04.05: Protect all forms of code from unauthorised access and tampering –** Help prevent unauthorised changes to code, both inadvertent and intentional, which could circumvent or negate the intended security characteristics of the software. For code that is not intended to be publicly accessible, it helps prevent theft of the	(Dodson et al., 2020)

	software and may make it more difficult or time-consuming for attackers to find vulnerabilities in the software. ○ **B.04.06: Provide a mechanism for verifying software release Integrity –** Help software consumers ensure that the software they acquire is legitimate and has not been tampered with. ○ **B.04.07: Archive and protect each software release –** Help identify, analyse, and eliminate vulnerabilities discovered in the software after release. • ***Produce well-secured Software:*** ○ **B.04.08: Design software to meet security requirements and mitigate security risks –** identify and evaluate the applicable security requirements for the software's design; determine what security risks the software is likely to face during production operation and how those risks should be mitigated by the software's design; and justify any cases where risk-based decisions conclude that security requirements should be relaxed or waived. Addressing security requirements and risks during software design (secure by design) helps to make software development more efficient. ○ **B.04.09: Review the software design to verify compliance with security requirements and risk information –** Help ensure that the software will meet the security requirements and satisfactorily address the identified risk information. ○ **B.04.10: Verify third-party software complies with security requirements –** Reduce the risk associated with using acquired software modules and services, which are potential sources of additional vulnerabilities. ○ **B.04.11: Reuse existing, well-secured software when feasible instead of duplicating functionality –** Lower the costs of software development, expedite software development, and decrease the likelihood of introducing additional security vulnerabilities into the software. These are particularly true for software that implements security functionality, such as cryptographic modules and protocols. ○ **B.04.12: Create source code adhering to secure coding practices –** Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities during source code creation. ○ **B.04.13: Configure the compilation and build processes to improve executable security –** Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs. ○ **B.04.14: Review and/or analyse human-readable code to identify vulnerabilities and verify compliance with security requirements –** Help identify vulnerabilities so that they can be corrected before the software is released to prevent exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Human-readable code includes source code and any other form of code an organisation deems as human readable. ○ **B.04.15: Test executable code to identify vulnerabilities and verify compliance with security requirements –** Help identify vulnerabilities so they can be corrected before the software is released in order to prevent exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Executable code includes binaries, directly executed bytecode, directly executed source code, and any other form of code an organisation deems as executable. ○ **B.04.16: Configure the software to have secure settings by default –** Help to improve the security of the software at the time of installation	

	to reduce the likelihood of the software being deployed with weak security settings that would put it at greater risk of compromise. • ***Respond to Vulnerabilities:*** ○ **B.04.17: Identify and confirm vulnerabilities on an ongoing basis –** Help ensure that vulnerabilities are identified more quickly so they can be remediated more quickly, reducing the window of opportunity for attackers. ○ **B.04.18: Assess, prioritise, and remediate vulnerabilities –** Help to ensure that vulnerabilities are remediated as quickly as necessary, reducing the window of opportunity for attackers. ○ **B.04.19: Analyse vulnerabilities to identify their root causes –** Help reduce the frequency of vulnerabilities in the future.	
05	Secure software development best practices: • **B.05.01: Management buy-in and standards –** Identify key users, get them on board the development process, establish quality and security standards for use. • B.05.02: Functional and security requirements elicitation – Identify functional and associated security requirements. • **B.05.03: Release and sprint planning –** Prioritise sprint tasks; set quality and security test cases for the tasks. • **B.05.04: Development with code and security review –** Produce code using adopted coding standards; review code for quality and security compliance. • **B.05.05: Sprint review and close –** Review sprint deliverable against user and security requirements. • **B.05.06: Evaluation –** Evaluate product against user, quality and security requirements.	(Moyo & Mnkandla, 2020)

Appendix E: Expert Review – Guideline Document

Guideline Document

Purpose of the Study

The purpose of this study is to create a set of usable security heuristics for instant messaging (IM) application development. The implementation of the heuristics will assist developers in the development of more secure instant messaging applications that will provide a usable and favourable user experience to end users of these increasingly popular applications.

Reason for the Individual's Selection and the Role the Individual Will Play in My Study

You have been selected based on your expertise in **Security, Usability and/or Mobile Application Development**. The expertise you possess is required for the validation of the usable security heuristics for instant messaging application development, which will be done in the form of an expert review. The evaluation, provided by you, will be combined with the expert review results provided by other similar experts. This will allow for a full validation of the set of heuristics from the aspects of Security, Usability and Mobile Application Development

Preliminary Usable Security Heuristics for Instant Messaging Application Development

The process followed to attain the proposed set of usable security heuristics was as follows. Various prominent threats to information security were identified and analysed. From the identified information security threats, those which were deemed most relevant to mobile applications were extracted. These extracted threats were further examined and those identified to be the most prominent threats to IM applications are as follows: confidential information leakage, distribution of malicious code, man-in-the-middle attacks, permission system vulnerabilities and social engineering, as indicated in Table 1.

Table 1: Most prominent IM application threats

Threat	Definition
Confidential information leakage	Utilising an IM application for the accidental or deliberate dissemination of confidential information to an unauthorised party.
Distribution of malicious code	Owing to its popularity, IM applications have become one of the most widely used malware attack channels. The broad user base and swiftness of communication is especially optimal for the dissemination of malware.
Man-in-the-middle attacks	The monitoring of an IM application's activities across a network, to acquire confidential information.
Permission system vulnerabilities	An IM application's permission request is a request to access user data or device resources. If granted, an IM application can manipulate user information and device resources to obtain its desired result.
Social engineering	Utilising an IM application to manipulate human weaknesses to achieve a malicious objective.

Having identified the five most prominent IM application threats, the current IM application security controls to address such threats were analysed, from the six most popular IM applications, namely: Facebook Messenger, Snapchat, Telegram, Viber, WeChat, and WhatsApp. This analysis identified the common IM security controls found across the six IM applications and how each IM security control assisted in securing the user from the most prominent IM application threats. Once the study on current IM controls was completed, a content analysis was conducted. The content analysis for this study was conducted based on the steps developed by Bengtsson (2016). Bengtsson (2016) developed the four main steps for a rigorous content analysis, based on the writing of content analysis experts, including: Downe-Wamboldt, 1992; Morse & Richards, 2002; Patton, 2002; Krippendorff, 2004; Silverman, 2015. The aim of the content analysis was to investigate existing security and usability heuristics, guidelines, standards, and best practices for mobile application development. After following the rigorous content analysis process, the identified security and usability heuristics, guidelines, standards, and best practices which were deemed most relevant to mobile application development were identified and mapped against the five previously identified most prominent threats to IM applications. The resulting security and usability heuristics, guidelines, standards and best practices from the mapping were adapted into the proposed usable security heuristics for IM application development. This set of proposed usable security heuristics for IM application development was mapped against the previously identified IM security controls to ensure that the proposed usable security heuristics were relevant to the IM controls and are able to assist developers in implementing the current available IM security controls. The initial set of proposed heuristics went through multiple iterations to ensure that the proposed heuristics, as presented in Table 2, are concise and accurate. Nielsen's usability heuristics largely influenced the proposed set of usable security heuristics.

Take note that heuristics US-H01 to US-H09 presented in Table 2 are all adapted from Nielsen's well-known usability heuristics.

Table 2: Proposed Usable Security Heuristics for Instant Messaging Application Development

Heuristic code	Heuristic name	Definition
D2.US.H.01	**Visibility of security status**	IM applications should always keep users informed about the security status of the application through appropriate feedback within reasonable time.
D2.US.H.02	**Match between security features and the real world**	An IM application's security features should speak the users' language, using real-world standards for terms, phrases, and security ideas with which they are acquainted. This guarantees that the user is well-informed and aware of the influence of the security features on the IM application.
D2.US.H.03	**User security control and freedom**	Users frequently choose IM application security functions by accident, necessitating the presence of a clearly indicated 'emergency escape' that allows them to quit the undesirable state without having to go through a lengthy dialogue. Undo and redo are recommended.
D2.US.H.04	**Security consistency and standards**	When using IM applications security features, users should not have to question whether various security phrases, circumstances, or actions imply the same thing. An IM application's security features should be aligned with other IM applications to ensure that users maintain an understanding of the security features. This ensures that users maintain an understanding of the security features within the IM application environment.
D2.US.H.05	**Security recognition rather than recall**	Make security objects, actions, and choices accessible to reduce IM application user's memory burden. The user should not be required to recall information from one section of the security interaction to the next. When applicable, instructions for using the security features should be visible or easily accessible.
D2.US.H.06	**Flexibility and efficiency of use for security features**	Unseen by the inexperienced user, accelerators may commonly speed up the interaction for the expert user, allowing the security features to accommodate both inexperienced and experienced users. Users should be allowed to customise security-related features that they perform on a regular basis.
D2.US.H.07	**Aesthetic and minimalist security design**	Information that is useless or is seldom used should not be included in security dialogues. In a security dialogue, every additional unit of information, which conflicts with the essential pieces of information, lowers their relative visibility. Ensuring that the security dialogue utilised remains concise and specific to the topic at hand, will improve the user's decision-making with regard to the impact of the related IM application security features.

D2.US.H.08	**Threat prevention and user guidance**	IM applications should present security messages in a plaintext format to the user. IM applications should guide the user during usage by hiding unavailable functions, warning users about their actions, and assisting users to recognise, diagnose, and avoid potential threats.
D2.US.H.09	**Security help and documentation**	Even though it is preferable for the security features to be operated without documentation, assistance and documentation may be required. Any such security information should be simple to find, should concentrate on the user's security duty, should have a list of clear procedures to follow, and should be manageable in size.
D2.US.H.10	**Compliance of security and privacy controls**	IM applications must provide the current industry standard of security and privacy controls with basic plaintext instructions for users on how to implement and utilise these features effectively. The privacy features need to align with international standards, such as the South African Protection of Personal Information Act (POPIA) and the European General Data Protection Regulation (GDPR).
D2.US.H.11	**Encryption of application session and information**	IM applications need to be encrypted to the current industry level of encryption. The encryption level must be made clear to the user. If there is more than one level of encryption available, it must be clear which is active, and the user must be guided in how to select the relevant encryption feature. It is crucial for an IM application to encrypt the application session and the storage and transmission of information.
D2.US.H.12	**Least privilege by default**	IM applications need to be developed with the principle of least privilege, which is to ensure that the permissions requested by the application is limited to the minimum permissions required for functionality. IM applications must not request more permissions than those required. Each permission requested must be clearly and concisely explained to the user, to ensure that an informed decision is made by the user. This will also reduce the cognitive load on the user.
D2.US.H.13	**Secure access control**	No unauthorised access must be given to an IM application. The application must secure itself from all forms of attempted access from unauthorised entities.
D2.US.H.14	**Flexibility of user security expertise**	The security features of IM applications need to provide plaintext options suitable for users with diverse levels of skills and experience in security.
D2.US.H.15	**Notification of security updates**	To ensure optimal security, IM applications need to alert the user about application updates. To mitigate vulnerabilities of older applications, IM applications need to remain updated.
D2.US.H.16	**Secure malware controls**	IM applications need to implement controls to detect, prevent and recover from malware. Such applications should also inform and keep users aware of the situation.
D2.US.H.17	**Secure by default**	IM applications need to ensure that the optimal security settings are active, by default. This will reduce the chances of IM applications being utilised with weaker security.

Refer to Appendix G for Questionnaire.

Appendix F: Expert Review – Information and Informed Consent Form

NELSON MANDELA UNIVERSITY

INFORMATION AND INFORMED CONSENT FORM

RESEARCHER'S DETAILS	
Title of the research project	**Usable Security Heuristics for Instant Messaging Application Development**
Reference number	**216035929**
Principal investigator	**Craig Michael van Niekerk**
Address	
Postal Code	
Contact telephone number **(private numbers not advisable)**	

A. DECLARATION BY OR ON BEHALF OF PARTICIPANT		Initial
I, the participant and the undersigned	(full names)	
ID number		
OR		
I, in my capacity as	(parent or guardian)	
of the participant	(full names)	
ID number		
Address (of participant)		

A.1 HEREBY CONFIRM AS FOLLOWS:		Initial
I, the participant, was invited to participate in the above-mentioned research project		
that is being undertaken by	Craig Michael van Niekerk	
from	Engineering, the Built Environment and Information Technology faculty	
of the Nelson Mandela Metropolitan University.		

THE FOLLOWING ASPECTS HAVE BEEN EXPLAINED TO ME, THE PARTICIPANT:			Initial
2.1	**Aim:**	The purpose of this study is to create a set of usable security heuristics for instant messaging application development. The implementation of the heuristics will assist developers in developing a more secure instant messaging applications that is provide a usable and favourable user experience to end users of these increasingly popular applications.	

		The validation of the proposed set of usable security heuristics for instant messaging application development, from the perspectives of security, usability and mobile development.			
2.2	**Procedures:**	I understand that – participants will be required to fill out the questionnaire provided based on their role in the study, i.e. Security Expert, Usability Expert or Mobile Developer Expert.			
2.3	**Risks:**	No risks have been identified to any of the participants involved in this study.			
2.4	**Possible benefits:**	As a result of my participation in this study – upon completion of the study, participants will be provided with a digital copy of the finalised set of usable security heuristics for instant messaging application development.			
2.5	**Confidentiality:**	My identity will not be revealed in any discussion, description or scientific publications by the investigators.			
2.6	**Access to findings:**	Participants will be provided a digital copy of the final set of usable security heuristics for instant messaging application development.			
2.6	**Voluntary participation / refusal / discontinuation:**	My participation is voluntary	YES	NO	
		My decision whether or not to participate will in no way affect my present or future care / employment / lifestyle	TRUE	FALSE	
2.7	**Future re-usage**	I, the participant, hereby declare that the information provided by me, the participant, can be reused, by the primary investigator, for future publications.	YES	NO	

3.	**THE INFORMATION ABOVE WAS EXPLAINED TO ME/THE PARTICIPANT BY:**							**Initial**
Craig van Niekerk								
in	**Afrikaans**		**English**		**Xhosa**		**Other**	
and I am in command of this language, **or** it was satisfactorily translated to me by								
(name of translator)								
I was given the opportunity to ask questions and all these questions were answered satisfactorily.								
4.	No pressure was exerted on me to consent to participation and I understand that I may withdraw at any stage without penalisation.							
5.	Participation in this study will not result in any additional cost to myself.							

A.2	**I HEREBY VOLUNTARILY CONSENT TO PARTICIPATE IN THE ABOVE-MENTIONED PROJECT**:		
Signed/confirmed at		on	20
Signature or right thumb print of participant		Signature of witness: Full name of witness:	

A. STATEMENT BY OR ON BEHALF OF INVESTIGATOR(S)

I,	Craig Michael van Niekerk	declare that:							
1.	I have explained the information given in this document to	(name of patient/participant)							
	and / or his / her representative	(name of representative)							
2.	He / she was encouraged and given ample time to ask me any questions;								
3.	This conversation was conducted in	**Afrikaans**		**English**		**Xhosa**		**Other**	
	And no translator was used OR this conversation was translated into								
	(language)	by	(name of translator)						
Signed/confirmed at		on		20					
Signature of interviewer		Signature of witness: Full name of witness:							

A. DECLARATION BY TRANSLATOR *(WHEN APPLICABLE)*

I,	(full names)	
ID number		
Qualifications and/or		
Current employment		
confirm that I:		
1.	Translated the contents of this document from English into	(language)
2.	Also translated questions posed by	(name of participant)	as well as the answers given by the investigator/representative;
3.	Conveyed a factually correct version of what was related to me.	
Signed/confirmed at	on	20
I hereby declare that all information acquired by me for the purposes of this study will be kept confidential.		
Signature of translator	Signature of witness: Full name of witness:	

Appendix G: Expert Review – Questionnaire

Questionnaire

Please familiarize yourself with both the proposed usable security heuristics presented in Table 2, and the 5-point Likert scale provided below before answering the questions. The 5-point Likert scale to be utilized for answering questions as follows:

- 1 = Very Low
- 2 = Low
- 3 = Moderate
- 4 = High
- 5 = Very High

Biographical Information

Which best describes your field of expertise?

Answer: Choose an item. – Options: Security, Usability and Mobile Application Development.

Years of experience in your respective field:

Answer: Click or tap here to enter text.

What is your confidence level in understanding heuristics?

Answer: Choose an item. Options: 1=Very Low, 2=Low, 3=Moderate, 4=High and 5=Very High.

What is your confidence level in the implementation of heuristics?

Answer: Choose an item. Options: 1=Very Low, 2=Low, 3=Moderate, 4=High and 5=Very High.

SECURITY SECTION

Indicate the extent to which each of the proposed usable security heuristics satisfies the **security threats and concerns** relating to IM applications. Please indicate this according to the 5-point Likert scale and provide any comments or suggestions for improvement for.

Heuristic code (Refer to Table 2)	**Heuristic name**	**Usable security scale**	**Comment/Suggestion for improvement**
D2.US.H.01	**Visibility of security status**	Choose an item.	Click or tap here to enter text.
D2.US.H.02	**Match between security features and the real world**	Choose an item.	Click or tap here to enter text.
D2.US.H.03	**User security control and freedom**	Choose an item.	Click or tap here to enter text.
D2.US.H.04	**Security consistency and standards**	Choose an item.	Click or tap here to enter text.

D2.US.H.05	**Security recognition rather than recall**	Choose an item.	Click or tap here to enter text.
D2.US.H.06	**Flexibility and efficiency of use for security features**	Choose an item.	Click or tap here to enter text.
D2.US.H.07	**Aesthetic and minimalist security design**	Choose an item.	Click or tap here to enter text.
D2.US.H.08	**Threat prevention and user guidance**	Choose an item.	Click or tap here to enter text.
D2.US.H.09	**Security help and documentation**	Choose an item.	Click or tap here to enter text.
D2.US.H.10	**Compliance of security and privacy controls**	Choose an item.	Click or tap here to enter text.
D2.US.H.11	**Encryption of application session and information**	Choose an item.	Click or tap here to enter text.
D2.US.H.12	**Least privilege by default**	Choose an item.	Click or tap here to enter text.
D2.US.H.13	**Secure access control**	Choose an item.	Click or tap here to enter text.
D2.US.H.14	**Flexibility of user security expertise**	Choose an item.	Click or tap here to enter text.
D2.US.H.15	**Notification of security updates**	Choose an item.	Click or tap here to enter text.
D2.US.H.16	**Secure malware controls**	Choose an item.	Click or tap here to enter text.
D2.US.H.17	**Secure by default**	Choose an item.	Click or tap here to enter text.

In your opinion, is sufficient IM application security and threat exposure addressed by the proposed usable security heuristics?

Answer: Choose an item. Options: Yes and No

If not, please elaborate.

Answer: Click or tap here to enter text.

Do you foresee any challenges when implementing the proposed usable security heuristics from an IM security perspective?

Answer: Choose an item. Options: Yes and No

If not, please elaborate.

Answer: Click or tap here to enter text.

USABILITY SECTION

Indicate the extent to which each of the proposed usable security heuristics satisfies the **usability** concerns relating to the **security of IM applications**. Please indicate according to the 5-point Likert scale and provide any comments or suggestions for improvement for each.

Heuristic code (Refer to Table 2)	**Heuristic name**	**Usable security scale**	**Comment/Suggestion for improvement**
D2.US.H.01	**Visibility of security status**	Choose an item.	Click or tap here to enter text.
D2.US.H.02	**Match between security features and the real world**	Choose an item.	Click or tap here to enter text.
D2.US.H.03	**User security control and freedom**	Choose an item.	Click or tap here to enter text.
D2.US.H.04	**Security consistency and standards**	Choose an item.	Click or tap here to enter text.
D2.US.H.05	**Security recognition rather than recall**	Choose an item.	Click or tap here to enter text.
D2.US.H.06	**Flexibility and efficiency of use for security features**	Choose an item.	Click or tap here to enter text.
D2.US.H.07	**Aesthetic and minimalist security design**	Choose an item.	Click or tap here to enter text.
D2.US.H.08	**Threat prevention and user guidance**	Choose an item.	Click or tap here to enter text.
D2.US.H.09	**Security help and documentation**	Choose an item.	Click or tap here to enter text.
D2.US.H.10	**Compliance of security and privacy controls**	Choose an item.	Click or tap here to enter text.
D2.US.H.11	**Encryption of application session and information**	Choose an item.	Click or tap here to enter text.
D2.US.H.12	**Least privilege by default**	Choose an item.	Click or tap here to enter text.
D2.US.H.13	**Secure access control**	Choose an item.	Click or tap here to enter text.
D2.US.H.14	**Flexibility of user security expertise**	Choose an item.	Click or tap here to enter text.
D2.US.H.15	**Notification of security updates**	Choose an item.	Click or tap here to enter text.
D2.US.H.16	**Secure malware controls**	Choose an item.	Click or tap here to enter text.
D2.US.H.17	**Secure by default**	Choose an item.	Click or tap here to enter text.

Would the proposed usable security heuristics aid in improving the usability of the security within IM applications?

Answer: Choose an item. Options: Yes and No

If not, please elaborate.

Answer: Click or tap here to enter text.

Do you foresee any challenges when implementing the proposed usable security heuristics from an IM **usability perspective**?

Answer: Choose an item. Options: Yes and No

If not, please elaborate.

Answer: Click or tap here to enter text.

MOBILE APPLICATION DEVELOPMENT SECTION

Indicate the extent to which each of the proposed usable security heuristics satisfies the **IM application development concerns** relating to the security of IM applications. Please indicate this according to the 5-point Likert scale and provide any comments or suggestions for improvement for each.

Heuristic code (Refer to Table 2)	**Heuristic name**	**Usable security scale**	**Comment/Suggestion for improvement**
D2.US.H.01	**Visibility of security status**	Choose an item.	Click or tap here to enter text.
D2.US.H.02	**Match between security features and the real world**	Choose an item.	Click or tap here to enter text.
D2.US.H.03	**User security control and freedom**	Choose an item.	Click or tap here to enter text.
D2.US.H.04	**Security consistency and standards**	Choose an item.	Click or tap here to enter text.
D2.US.H.05	**Security recognition rather than recall**	Choose an item.	Click or tap here to enter text.
D2.US.H.06	**Flexibility and efficiency of use for security features**	Choose an item.	Click or tap here to enter text.
D2.US.H.07	**Aesthetic and minimalist security design**	Choose an item.	Click or tap here to enter text.
D2.US.H.08	**Threat prevention and user guidance**	Choose an item.	Click or tap here to enter text.
D2.US.H.09	**Security help and documentation**	Choose an item.	Click or tap here to enter text.
D2.US.H.10	**Compliance of security and privacy controls**	Choose an item.	Click or tap here to enter text.
D2.US.H.11	**Encryption of application session and information**	Choose an item.	Click or tap here to enter text.
D2.US.H.12	**Least privilege by default**	Choose an item.	Click or tap here to enter text.
D2.US.H.13	**Secure access control**	Choose an item.	Click or tap here to enter text.
D2.US.H.14	**Flexibility of user security expertise**	Choose an item.	Click or tap here to enter text.

D2.US.H.15	**Notification of security updates**	Choose an item.	Click or tap here to enter text.
D2.US.H.16	**Secure malware controls**	Choose an item.	Click or tap here to enter text.
D2.US.H.17	**Secure by default**	Choose an item.	Click or tap here to enter text.

Can the proposed usable security heuristics be implemented during the IM mobile application development process?

Answer: Choose an item. Options: Yes and No

If not, please elaborate.

Answer: Click or tap here to enter text.

Do you foresee any challenges when implementing the proposed usable security heuristics from an IM mobile application development perspective?

Answer: Choose an item. Options: Yes and No

If not, please elaborate.

Answer: Click or tap here to enter text.

GENERAL SECTION

What is your overall impression of the proposed usable security heuristics, with regard to their efficacy, utility and quality? Please indicate this according to the 5-point Likert scale and provide any comments or suggestions for improvement for each.

- **Efficacy** – The ability to produce a desired or intended result.
- **Utility** – The state of being useful, or beneficial.
- **Quality** – The standard as measured against other heuristics.

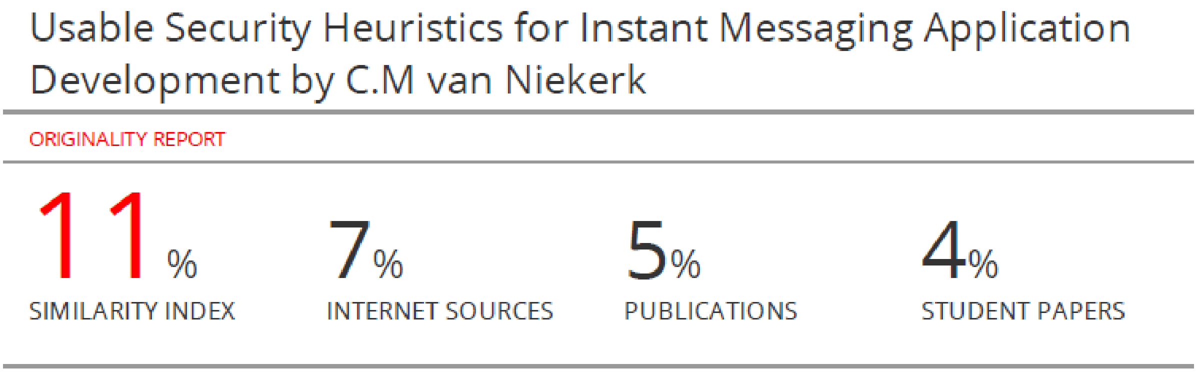
Name	Usable security scale	Comment/Suggestion for improvement
Efficacy	Choose an item.	Click or tap here to enter text.
Utility	Choose an item.	Click or tap here to enter text.
Quality	Choose an item.	Click or tap here to enter text.

Please provide any final comments of the proposed usable security heuristics for IM applications.

Answer: Click or tap here to enter text.

Appendix H: Turnitin Report

Turnitin report for the dissertation titled Usable Security Heuristics for Instant Messaging Application Development, excluding the appendices.



Usable Security Heuristics for Instant Messaging Application Development by C.M van Niekerk

ORIGINALITY REPORT

11%	7%	5%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

Appendix I: Editing Certificate

Editing Certificate

Prof. Lynn Futcher

Faculty of Engineering, the Built Environment and Technology

Nelson Mandela University

Editing of Master's dissertation

I, Marietjie Alfreda Woods, hereby certify that I have completed the editing and correction of the Master's dissertation **Usable Security Heuristics for Instant Messaging Application Development** by **Craig Michael van Niekerk**, submitted in fulfilment of the requirements for the degree **Master of Information Technology** at the Nelson Mandela University. I believe that the dissertation meets with the grammatical and linguistic requirements for a document of this nature.

Name of Editor: Marietjie Alfreda Woods

Qualifications: BA (Hons)(Wits), Practical Copy-editing and Proofreading (UCT)

Signature:

Contact Number: 083 312 6310

Email address: rickywoods604@gmail.com

Date Issued: 7 November 2021