

# Security Analysis of Simpel Desa using Mobile Security Framework and ISO 27002:2013

**Received:**

2 October 2022

**Accepted:**

2 January 2023

**Published:**

10 February 2023

<sup>1\*</sup>**Khairunnisak Nur Isnaini**, <sup>2</sup>**Didit Suhartono**

<sup>1-2</sup>*Informatika, Universitas Universitas Amikom Purwokerto*

*E-mail: <sup>1</sup>nisak@amikompurwokerto.ac.id,*

*<sup>2</sup>didit@amikompurwokerto.ac.id*

\*Corresponding Author

**Abstract**— The Personal Identification Number or KTP is prone to be stolen and used by unwanted parties, this is also a possibility for the Simpel Desa, a village administration application that also contain and use the Personal Identification Number. This study aims to detect information security vulnerabilities. This study aims to analyze security vulnerabilities in applications using MobSF and ISO 27002:2013. MobSF is used for penetration testing for malware in applications. In MobSF the Simpel Desa application is analyzed in two ways, namely static and dynamic. ISO 27002:2013 is used to map the findings of vulnerabilities and potential misuse of information so that they get accurate analysis results. The control used is domain 9 (access control) and 10 (cryptography). The results obtained in the static analysis found the existence of vulnerabilities in aspects of cryptography and permission access. The dynamic analysis found that Root Detection and Debugger Check Bypass had not been implemented. Overall, based on ISO 27002:2013 information security has not been maximally implemented. The recommendations given focus on the aspects of application permissions and access rights, user authentication, and the implementation of information security.

**Keywords**— Information Security; Vulnerabilities; MobSF; ISO 27002:2013

This is an open access article under the CC BY-SA License.



---

**Corresponding Author:**

Khairunnisak Nur Isnaini,  
Informatika,  
Universitas Universitas Amikom Purwokerto,  
Email: [nisak@amikompurwokerto.ac.id](mailto:nisak@amikompurwokerto.ac.id)



## I. INTRODUCTION

Simpel Desa application is an android application that contains village administration services and village businesses used by the people of Cingebul Village, Lumbr District, Banyumas Regency. It is integrated with the Management and Service Information System [1]. The use of population numbers in applications is not new and has become part of the development of information technology. Using population data is risky because its value in information can become a vulnerable target. The Population Identification Number contains important and sensitive information related to the owner's personal data and is connected to various government and banking administration services.

The Simpel Desa application was originally intended to be exclusive to villagers with a residence number as access to the application. Still in practice, it can be accessed by other residents who are not residents. These problems can happen because not all villagers are aware of the importance of maintaining personal data and the potential for misuse of information. In addition, the application needs to implement the right user authentication system to deal with these events. Exposing of important data in the application can be easily misused by irresponsible people, which becomes a series of very high-risk security holes. Lack of security implementation and user negligence are also factors that can increase information security risks.

The existence of access rights issues and the potential for misuse of information on population numbers are used by attackers to steal important information on smartphone devices. Information is an important and valuable asset that is presented in various formats in the form of notes, verbal, electronic, postal, and audio-visual [2]. In addition, the impact users feel, for developers, the existence of this security hole is also a big loss in terms of good name and reputation. So, to maintain stability and integrity, developers or organizations must be able to protect important and sensitive information [3].

Data theft perpetrators use various ways to collect important information through cybercrime [4] and malicious programs inserted into the user's device. One of the malicious programs in an application is malware. Malicious software or malware is explicitly designed to carry out malicious activities or other malware such as trojans, viruses, spyware, and exploits [5]. The existence of this malicious program is difficult to detect for users with technology. The way the program works in the background, along with other applications, adds to the difficulty of detecting the presence of malware. In addition, malicious programs such as malware inserted through the applications used, theft or loss of personal data can also occur due to negligence on the part of the user. The existence of non-technical factors like this makes information security increasingly threatened and adds to the long record of security risks faced.

Actions to secure information need to be taken to avoid unauthorized threats or incoming authorities, as well as guard against damage caused by outsiders [6]. In addition to threats and impacts for users, security risks are also felt for organizations related to a good name, business continuity to reputation [3]. Information security protection aims to prevent harm to the organization's continuity of activities [7]. Information security can be used as a human firewall to save organizational assets [8]. Some security aspects are privacy and confidentiality, integrity, authentication, and availability [4]. Confidentiality of data or information is a complete service that keeps stored information from being read or opened by unauthorized parties [9]. The process guides information security to protect confidential and sensitive information and aims to avoid misuse and modification of information [10]. Organizations must be able to guarantee the security of their information so that the information is guaranteed confidentiality (confidentiality), can be authenticated (integrity) and can always be available when needed (availability) [11].

The need for information security measures can implement a framework. The framework aims to reduce or avoid threats to existing information security risks. Based on the problems that arise, the framework that will use is a framework related to applications, namely the Mobile Security Framework and a framework related to application users, namely ISO 27002:2013. Testing on the application side using penetration testing is also important to detect vulnerabilities such as the intentional exploitation of applications, operating systems, and networks [12].

Mobile security best practices it is applied to be either reference or standard to secure mobile device or user's data [13]. A Mobile security framework is a tool or framework that can use to analyze security in mobile able to provide information about security vulnerabilities in an application [14]. The mobile security framework abbreviated as MobSF, is also used for penetration testing of mobile applications starting from Android, IOS, and windows [15]. MobSF is an open-source application test [16]. MobSF can analyze statically and dynamically against applications that are vulnerable to being penetrated by hackers to detect malware applications that are packaged into other applications. [17]. In dynamic testing, MobSF can fuzz Web API, which functions to find software errors using random input [18]. In addition, dynamic testing does not check the source code but focuses on application activities when used. [19].

Information security is also to be implemented by applying reference standards to reduce and avoid information security vulnerabilities. ISO 27002 is a form of the 27000 series, which manages and controls three aspects of information security: confidentiality, integrity, and availability [20]. ISO 27002:2013 is a framework, as a reference for security standards that contain guidelines for applying standards and information management [21], is one form of security implementation that can use to maximize the application of information security. ISO 27002 improves security based on standard rules in compliance checking basic knowledge [22].

ISO 27002:2013 provides controls to measure logical information security, and the available controls are also used to measure physical information security [23]. Control selection in ISO 27002:2013 can be selected optionally implementing a risk-based information security management system [24]. Document ISO 27002:2013 which consists of 14 major divisions, 35 subdivisions, and 114 controls provides governance of security guidelines that can be select according to information control needs.

Previous research related to the information security process was implemented by [23] on a library application's implementation and authentication process. The tests aim to ensure the validity of the processed data and the encryption performance used. Information security analysis is also implemented by [24], which focuses on the login security policy process, such as password requirements, password length, and type of data Security used by several online banking. This study produces an alternative secure password based on the type of encryption used and the login mechanism using an encrypted password. Research conducted by [25] uses ISO 27002:2013 to determine the level of information security and provides recommendations based on the evaluation results of 3 clauses, namely access control, physical and environmental security and information system acquisition, development, and maintenance. The recommendations given are only on the user side and have yet to reach the technical side of the system developer. ISO 27002:2013 is used by [26] to propose an evaluation of an information security system in a school with the results of finding various important assets that fall into the high-risk category according to their respective clauses. ISO 27002:2013, implemented with COBIT 5 is used by [26] for information technology security audits in an institution. The results provide data that the institution is still at level 2, namely managed process, without providing recommendations for further improvement. MobSF is used by [27] for vulnerability assessment testing by comparing two applications downloaded on Playstore and modified applications. This study tested the authenticity of the data based on the results of the security score and hash value without classifying the static test or dynamic test on the application. Research by [28] evaluates the possibility of attacks or penetration testing using MobSF. The results conclude that vulnerability testing needs to implement to improve information security regularly.

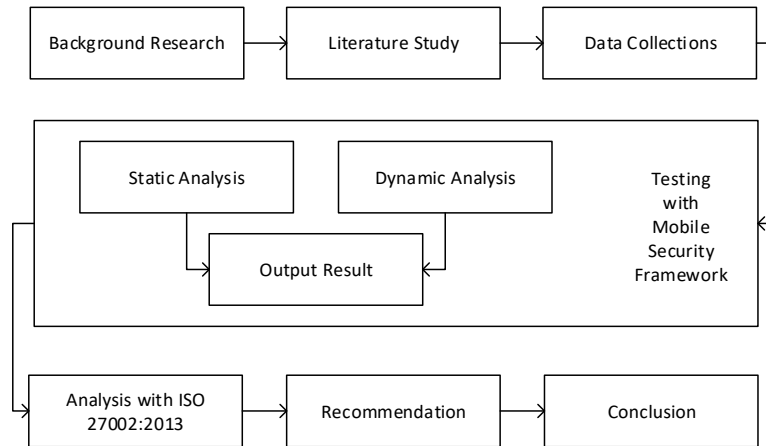
This study aims to analyze security holes in applications using MobileSF and ISO 27002 and provide recommendations for users and application developers. The previous research was limited to analyzing the result of static and dynamic analysis without recommending maintenance, and it only addressed the user. Thus this research not only focuses on the result of static and dynamic analysis from mobile SF but also maps both analysis results into the control according to ISO to extract the recommendation of maintenance. The shared recommendation can help the user to increase security awareness. Besides, a recommendation **is also advised** to application

developers, which can be addressed through the operator of local officials to increase the patch security in each updated version of the application.

## II. RESEARCH METHOD

### A. Research Flow

The stages or flow of research are described in the following figure 1:



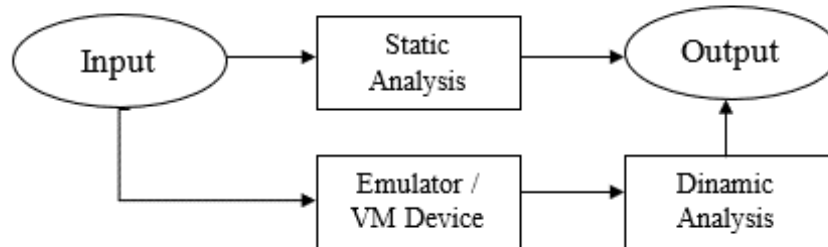
**Figure 1.** RESEARCH FLOW

Figure 1 shows the flow of the research, and this paper uses a qualitative case research method, which means the assessment is based on the results obtained from ongoing case studies and data at the observation and interview stages [14]. This information is through a literature study regarding using Simpel Desa applications among the people of Cingebul Village, Lumbr District, Banyumas Regency. The data collected is divided into two types of data, namely primary data and secondary data. Primary data is information from interviews and observations, while secondary data is obtained indirectly, such as publications, documentation, and notes. In this case, the data is obtained through the access of one of the application users to find out how the application works and other supporting data. In addition, interviews are also conducted with these users. All this data is used as a tool and material for analysis on the Simpel Desa application using a mobile security framework. The test using MobSF is divided into two, namely static analysis and dynamic analysis. Security analysis using static analysis is done using MobSF Framework accessed via the local network, then the Simpel Desa application is uploaded with the APK version (Application Package Kit), and the results will be presented on that page. At the same time, dynamic analysis is done by installing an emulator in the MobSF Framework and producing some application security test results. The results obtained are then used as evaluation material using ISO 27002:2013 controls based on MobSF static and dynamic analysis reports and information at the data collection stage to draw conclusions can use as evaluation material. The

results of this evaluation will be compiled as recommendation material for users of the Sempel Desa application and application developers to improve the security of their applications.

## B. Mobile Security Framework

A mobile security framework is a tool capable of performing static and dynamic analysis [8]. The flow of the use of the mobile security framework is generally described as follows:



**Figure 2.** THE FLOW OF MOBILE SECURITY FRAMEWORK

Figure 2 shows the flow of the mobile security framework, which starts from the input files on MobSF. Application files uploaded will be automatically tested for static analysis and displayed after completion. The static test process on the mobile security framework includes cryptographic analysis, SSL pinning, dangerous permissions and hardcoded secrets [2]. This static process is utilized by checking the source code of the application that has gone through the *decompile* file; after the static analysis process is complete, the stages are continued with the application dynamic analysis process. Dynamic analysis is performed on a virtual device emulator to create a more secure environment during testing. Dynamic analysis tests include SSL pinning bypass, API monitoring, root detection, and debugger check bypass. The output generated from the dynamic tests is in the form of reports from the tests.

## C. Control ISO 27002:2013

ISO 27002:2013 contains policies, processes, procedures, organizational structures, and software to hardware on information security [10]. The ISO 27002:2013 document provides security standard controls that can tailor to the needs of the organizations that use them. Domain 9 (access control controls) and domain 10 (cryptographic controls) are used based on the relevance of the results of the mobile security framework and existing primary data. Domain 9 (access control) provides policies, implementations, and procedures that can use in applying information standards. Access control is a mechanism used to secure and ensure the confidentiality of user data. The instrument in question checked the rights of the user based on the

specified authorization [16]. Domain 10 (cryptography) governs the policy and implementation of cryptography on a system. Cryptographic methods secure important information for user data organizations [17].

### III. RESULT AND DISCUSSION

#### A. Testing Mobile Security Framework

A mobile security framework is a collection of tools required to perform the testing process. MobSF **cannot be implemented** if the needed tools or programs **are not met**. The tools required are:

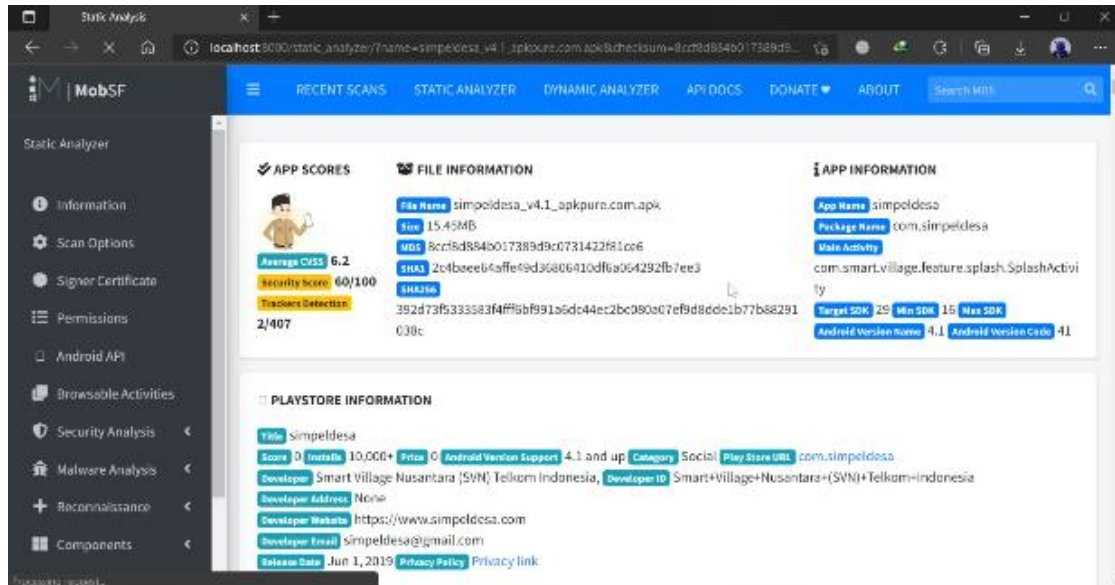
1. Git : is a version control software for source code management that **is needed** and **is used** to make clones or copies of the GitHub mobile security framework repository.
2. Python : as a programming language, python is needed a mobile security framework in the configuration and use process.
3. Java Development Kit (JDK): The JDK is used as the compiler and debugger needed by the mobile security framework.
4. Microsoft Visual C++ Build Tools: Mobile security framework requires visual c++ build tools to use libraries and program integrations.
5. OpenSSL : MobSF requires OpenSSL during dynamic analysis to secure the connection.
6. Wkhtmltopdf : It is a tool used by MobSF to render analysis results from HTML into PDF format.
7. Genymotion : An android emulator that is used to run the program under test when performing dynamic analysis processes.

#### B. Analysis of Mobile Security Framework

Mobile security framework can perform static and dynamic analysis tests on applications. The Simpel Desa application file with .Apk format is used in the testing process.

##### 1. Static Analysis

After the required programs' installation process is complete, the framework can be accessed by the local network on a web browser. The static analysis process by the mobile security framework is implemented after the application file has been uploaded automatically. MobSF will immediately start the static test process. The results of the static analysis on the Simpel Desa application are shown in Figure 3.



**Figure 3. STATIC ANALYSIS RESULTS**

Figure 4 shows the initial display of the report on the results of the analysis test on the Sempel Desa application. There is some information on the results of the static analysis. Still, we cannot describe all of it because it is not the scope of the research problem, namely information security vulnerabilities. One of the general information provided is the application score. In the application score, Sempel Desa is measured by the CVSS technique (or Common Vulnerability Scoring System), a standard for assessing the severity or vulnerability of system security. The score is calculated based on several metrics that estimate the impact of existing exploits. These include CVSS base, CVSS temporal, and CVSS environmental metrics. CVSS has the lowest score of 0, and 10 is the highest [29]. The score rating is shown in table 1.

**Table 1. CVSS SCORE RATING**

CVSS Score Qualitative Rating	
0.0	None
0.1-3.9	Low
4.0-6.9	Medium
7.0-8.9	High
9.0-10.0	Critical

Table 1 describes the types of vulnerability ratings, ranging from undetected vulnerabilities to the highest level, critical vulnerabilities. In the Sempel Desa application, the CVSS results are shown in Figure 4.



File Name:	simpeldesa_v4.1_apkpure.com.apk
Package Name:	com.simpeldesa
Average CVSS Score:	<b>6.2</b>
App Security Score:	<b>60/100 (MEDIUM RISK)</b>
Trackers Detection:	<b>2/407</b>
Scan Date:	Jan. 2, 2022, 4:01 p.m.

**Figure 4.** THE RESULT OF CVSS SCORE

From the static analysis results in Figure 4, the Sempel Desa application gets the medium category on the CVSS with an average score of 6.2. The value of 6.2 is included in the medium risk category, indicating the possibility of security holes encountered during the analysis process. In addition, there is general information, such as the assessment of the application of the static analysis results obtained, which is contained in table 2

**Table 2.** STATIC ANALYSIS RESULTS

<b>Possible Vulnerability</b>	<b>Simpel Desa</b>
Weak Crypto	Yes
Dangerous Permission	Yes
Network Security	Yes
Hardcode Secret	Yes
SSL Pinning	No

Table 2 summarises the static analysis results on the Sempel Desa application. Possible security vulnerabilities in these applications allow security risks to occur.

a. Weak Crypto

The use of cryptography in Sempel Desa applications **is listed** in the signature certificate section. The purpose cryptographic method is data security. The results obtained are in Figure 5.

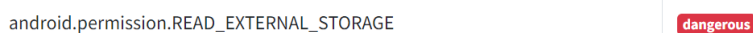


**Figure 5.** THE RESULT OF SIGNATURE CERTIFICATE

In Figure 5, MD5 and SHA-1 cryptographic methods for Sempel Desa applications fall into weak cryptographic security techniques. These results are reinforced by findings quoted from the book Security+ guide to network security fundamentals that defects were found in the compression function that could cause collisions. The collision is that hackers can create files with the same hash, so it cannot be confirmed that the file has not been modified [30].

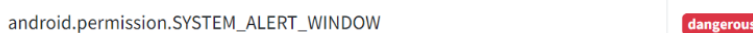
b. Application Permission

In testing on the mobile security framework, it is known that several permissions are categorized as dangerous, some of which are reading data and warning systems. In Figure 6, there are results from the results of the analysis of reading data.



**Figure 6.** THE RESULT OF EXTERNAL STORAGE

The Sempel Desa application has access to external storage on the user's device, as shown in Figure 6. This action makes the application capable of reading data on the user's device. This capability has the potential for the theft of critical data stored by users on the device. In addition, the warning system results are shown in Figure 7.




**Figure 7.** THE RESULT OF ALERT WINDOW

Figure 8 shows that the Sempel Desa application has system alert or warning window permissions. This action causes suspicious applications to take over all access to the user's device screen with these permissions.

c. Network Security

The mobile security framework detects network activity in a Sempel Desa application. This activity is indicated in Figure 8 by discovering a domain categorized as high severity by MobSF.

 NETWORK SECURITY

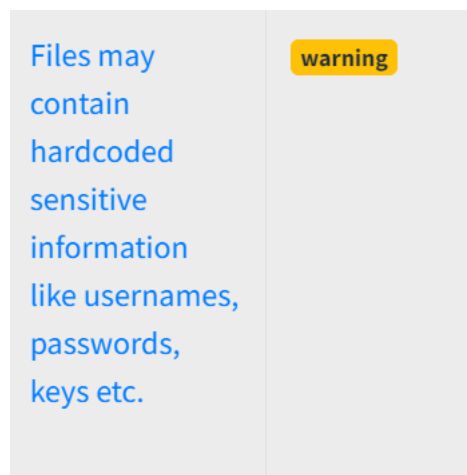
NO ↑↓	SCOPE ↑↓	SEVERITY ↑↓
1	api-staging.simpeldesa.com assets.simpeldesa.com.s3-id-jkt- 1.kilatstorage.id assets.simpeldesa.com	high

**Figure 8.** THE RESULT OF NETWORK SECURITY

The results in Figure 9 indicate that this domain is categorized as high severity because it is insecurely configured to allow text traffic on the domain.

d. Hardcode Secret

Hardcode secret displays information about application fields, including sensitive data, as shown in figure 9.



**Figure 9.** THE RESULT OF HARDCODE INFORMATION

Figure 9 shows a warning from the code analysis that it is known that the Sempel Desa application has files containing sensitive information such as usernames, passwords, and others. Hardcode is meant to embed data directly into the source code.

e. SSL Pinning

SSL Pinning is to detect and use a valid certificate or public key, making application developers use this technique as an additional layer of security on application traffic. The use of SSL Pinning is illustrated in Figure 10.

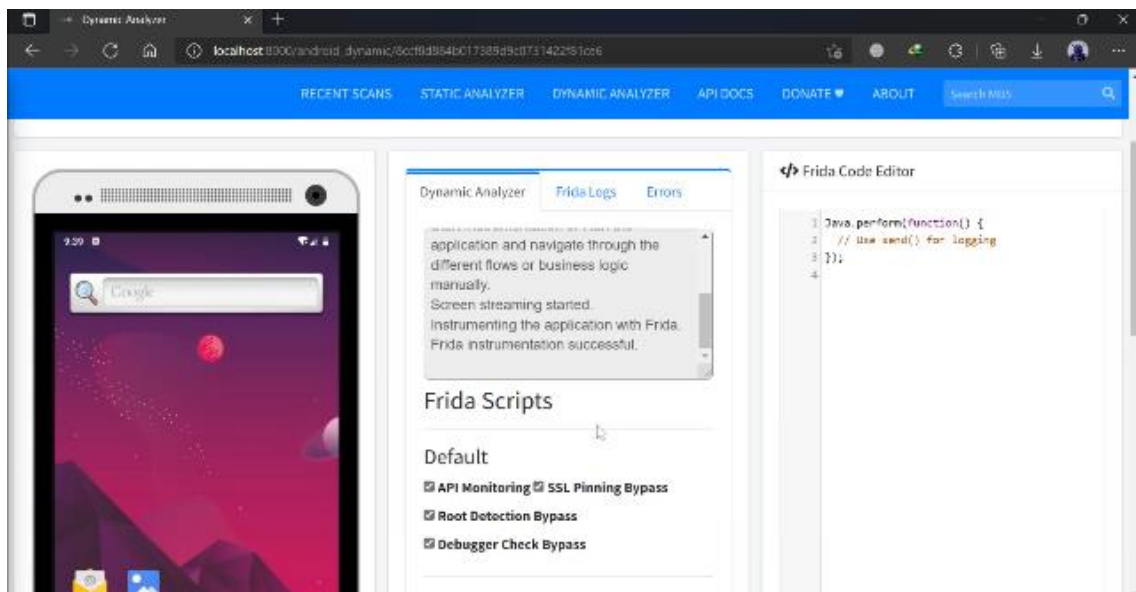
This App uses  
SSL certificate  
pinning to  
detect or  
prevent MITM  
attacks in  
secure  
communication  
channel.

secure

**Figure 10.** THE RESULT OF SSL PINNING

In Figure 10, a Simpel Desa application uses SSL certificate pinning to prevent attacks on communication lines. The running process, namely the application, will detect custom certificates and intercept traffic or traffic that occurs to reduce existing security risks.

2. Dynamic Analysis : The dynamic analysis process can be implemented after completing the analysis test.



**Figure 11.** THE RESULTS OF DYNAMIC ANALYSIS

Figure 11 shows the initial view of dynamic analysis testing on the mobile security framework. The genymotion emulator installed and run will be detected by MobSF and displayed on the analysis page through a proxy on genymotion. Several tests were used in the dynamic analysis test: API Monitoring, SSL Pinning Bypass, Root Detection, and Debugger Check Bypass.

a) API Monitoring

API Monitoring is the process of collecting and analyzing data related to the performance of an API (Application Programming Interface) to identify the impact felt by users. In general, the results in the Sempel Desa application have not found a performance weakness in the application programming interface. Still, in some activities, it is found that the application response has decreased performance and caused a force close during the test.

b) SSL Pinning Bypass

SSL pinning bypass is a test to find out whether there is an invalid certificate in the application, as illustrated in figure 12.

```
Frida Logs - com.simpeldesa

Data refreshed in every 10 seconds.

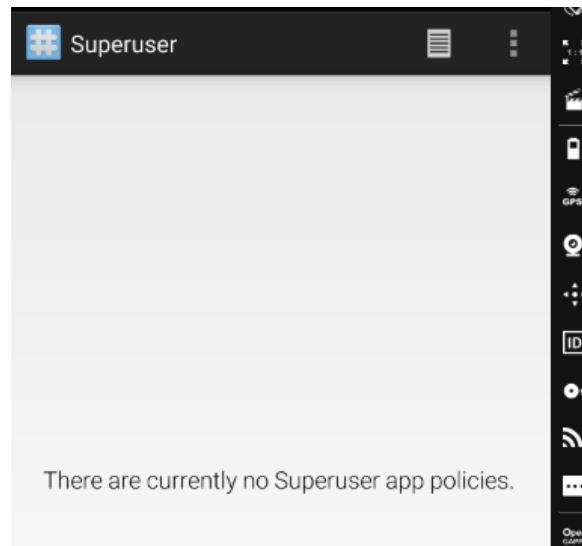
Loaded Frida Script - debugger_check_bypass
Loaded Frida Script - root_bypass
Loaded Frida Script - ssl_pinning_bypass
[Debugger Check Bypass] isDebuggerConnected() bypassed
[Debugger Check Bypass] isDebuggerConnected() bypassed
[SSL Pinning Bypass] okhttp CertificatePinner not found
[RootDetection Bypass] test-keys check
[RootDetection Bypass] return value for binary: Superuser.apk
[RootDetection Bypass] return value for binary: su
[SSL Pinning Bypass] DataTheorem trustkit not found
[SSL Pinning Bypass] Appcelerator PinningTrustManager not found
[SSL Pinning Bypass] Apache Cordova SSLCertificateChecker not found
[SSL Pinning Bypass] Wultra CertStore.validateFingerprint not found
[RootDetection Bypass] test-keys check
[RootDetection Bypass] return value for binary: Superuser.apk
[RootDetection Bypass] return value for binary: su
[Debugger Check Bypass] isDebuggerConnected() bypassed
[SSL Pinning Bypass] Xutils not found
[SSL Pinning Bypass] httpclientandroidlib not found
[SSL Pinning Bypass] Cronet not found
[SSL Pinning Bypass] certificatetransparency.CTInterceptorBuilder not found
```

**Figure 12.** THE RESULTS OF DYNAMIC ANALYSIS

Figure 12 shows the results that there are no loopholes in the SSL Pinning Bypass process that have the potential to become a security threat in Sempel Desa applications. This result indicates that the developer has implemented SSL Pinning to secure traffic on the application and avoid damaging the certificate or critical validation by third parties.

c) Root Detection

Root detection bypass is a technique applied to applications to detect the use of super users on devices that have access to the entire system. Root Detection results can be seen in Figure 13.



**Figure 13.** THE RESULTS OF ROOT DETECTION

Figure 13 shows the super user application used during the dynamic analysis process. The application does not detect any extraordinary user requests from an application.

d) Debugger Check Bypass

A debugger is a tool that allows users to see what is being executed and view data from an application while it is running. The results can be seen in Figure 14.

```
[Frida] [Debugger Check Bypass] isDebuggerConnected() bypassed  
[Frida] [Debugger Check Bypass] isDebuggerConnected() bypassed
```

**Figure 14.** THE RESULTS OF DEBUGGER CHECK BYPASS

The Simpel Desa application is bypassed when testing dynamic analysis, as shown in Figure 14, which is a log when the investigation is run. When the dynamic analysis is run and checking with the debugger connected state is performed, the log shows that the debugger status is bypassed, which means the Simpel Desa application does not detect any debugger attached to the application.

**C. ISO 27002:2013 Framework**

Control The use of ISO 27002:2013 control is implemented to determine the security standards applied to the application. Not all control domains are included in the discussion and adjusted to the results of the mobile security framework and the primary data obtained. Then who will further analyze only controls related to access control and cryptography.

**Table 2.** LIST OF ISO 27002:2013 CONTROLS (PROCESSED BY THE AUTHOR)

<b>Domain</b>	<b>ID</b>	<b>Control</b>	<b>Object</b>
9 Access Control	9.1.1	Access control policy	Policy
	9.1.2	Access to network and network services	Policy
	9.2.1	User registration and de-registration	Implementation
	9.2.2	User access provisioning	Implementation
	9.2.3	Management of privileged access rights	Implementation
	9.2.4	Management of secret authentication information of users	Implementation
	9.2.5	Review of user access rights	Implementation
	9.2.6	Removal or adjustment of access rights	Implementation
	9.3.1	Use of secret authentication information	Implementation
	9.4.1	Information access restriction	Policy
	9.4.2	Secure log-on procedures	Procedure
	9.4.3	Password management	Implementation
	9.4.4	Use of privileged utility programs	Policy
	9.4.5	Access control to program source code	Policy
	10 Cryptography	10.1.1	on the use of cryptographic controls
10.1.2		Key management	Implementation

Table 2 shows the two control domains in ISO 27002:2013 that have gone through the processing stage based on the relevance and results obtained in the static and dynamic analysis processes and the existing primary data.

1. Aspect 9 – Access Control

a. 9.1.1 – Access Control Policy

Sets to a policy to be defined, documented and reviewed regularly. It is known that the Simpel Desa application regulates application usage policies and user information written in the privacy policy and application terms and conditions.

b. 9.1.2 – Access to Network

Sets that users are only granted access to networks and services they are expressly authorized to use. It is known that the Simpel Desa application has a service that uses a network to access content on the application.

- c. 9.2.1 – User Registration and De-Registration  
Arrange for a formal user registration and de-registration process to assign access rights. It is known that the Sempel Desa application implements a registration process using a residence number as an entry requirement. However, no de-registration process has been found in the application.
- d. 9.2.2 – User Access Provisioning  
A formal user access provisioning process should be implemented to revoke user access rights. It is known that the Sempel Desa application applies controls to assign or revoke access rights to the services used.
- e. 9.2.3 – Management of Privileged Access Rights  
The review of the allocation and use of special access rights shall be limited and controlled. It is known that the Sempel Desa application has special access rights for village officials according to their respective positions.
- f. 9.2.4 – Management of Secret Authentication Information of Users  
An overview of the allocation of confidential authentication information is controlled through a formal management process. It is known that the Sempel Desa application applies information authentication in the application login process with a one-time password and temporary password.
- g. 9.2.5 – Review of User Access Rights  
Reviews are conducted periodically on users who have special access rights. It is known that the Sempel Desa application has not found any studies and reviews of user access rights on a regular basis.
- h. 9.2.6 – Removal or Adjustment of Access Rights  
Employee and external user reviews of information and processing facilities should be removed upon termination of employment, contract or agreement. There is no known information regarding the review of deletion or changes in access rights for users who have an employment relationship.
- i. 9.3.1 – Use of Secret Authentication Information  
Users are required to follow the organization's practice of using information. It is known that the Sempel Desa application documents, the practice of using information on the terms and conditions page and the privacy policy used.
- j. 9.4.1 – *Information Access Restriction Information*  
Access and application functions are restricted according to access control policies. It is known that the Sempel Desa application imposes restrictions on access to information by ordinary users and those with special access rights.



- k. 9.4.2 – Secure Log-On Procedures  
Secure login procedures must control access to systems and applications. It is known that the Simpel Desa application will check whether the population number used is registered in the system before it can proceed with one-time password.
  - l. 9.4.3 – Password Management  
Password management should be interactive and ensure password quality. It is known that the Simpel Desa application recommends that users replace temporary files password with password consisting of 8 unique characters to increase the quality of security.
  - m. 9.4.4 – Use of Privileged Utility Programs  
Review the use of utility programs capable of taking over system and application control. It is known that the Simpel Desa application has not found any supervision for utility programs that are able to obtain special access rights such as *super users* on the device.
  - n. 9.4.5 – Access Control to Program Source Code  
Access to program code should be limited. It is known that users cannot access the Simpel Desa application code.
2. Aspect 10 – Cryptography
    - a. 10.1.1 – Policy on the Use of Cryptographic Controls  
The application of a policy on using of cryptographic controls for information protection. It is known that the Simpel Desa application has not found a policy that regulates the use of cryptography.
    - b. 10.1.2 – Key Management  
User, protection and lifecycle policies of cryptographic keys shall be developed and implemented. It is known that the Simpel Desa application applies a cryptographic key that has a valid validity period or life cycle for the next few years.

The analysis and evaluation results on the application of security standards for Simpel Desa applications based on domain 9 access control and 10 cryptography in ISO 27002:2013 found several control domains that were appropriate or not in accordance with their application. The following is an illustration of the implementation of security in the Simpel Desa application.

**Table 3.** IMPLEMENTATION STATUS

Domain	ID	Status	
		Appropriate	Not Appropriate
9.1.1	Access control policy	✓	
9.1.2	Access to network and network services		✓
9.2.1	User registration and de-registration		✓
9.2.2	User access provisioning		✓
9.2.3	Management of privileged access rights	✓	
9.2.4	Management of secret authentication information of users	✓	
9.2.5	Review of user access rights		✓
9.2.6	Removal or adjustment of access rights		✓
9.3.1	Use of secret authentication information	✓	
9.4.1	Information access restriction	✓	
9.4.2	Secure log-on procedures	✓	
9.4.3	Password management	✓	
9.4.4	Use of privileged utility programs		✓
9.4.5	Access control to program source code		✓
10.1.1	Policy on the use of cryptographic controls		✓
10.1.2	Key management	✓	

Table 4 is the status of the application of 9 access control application and 10 cryptography aspects of domain security in the Simpel Desa application. The Simpel Desa application has generally met the information security aspect based on the controls used. However, it found that several control domains needed to be implemented according to the standards for the two controls, such as the lack of de-registration in control 9.2.1 and policies on using cryptography in control 10.1.1.

If viewed based on ISO 27002:2013 controls, in general, the Simpel Desa application has implemented security standards according to these controls, such as in sub-control 9.4.2 secure log-on procedures, Simpel Desa application has implemented login controls and access-to-use applications only for residents village whose NIK has been registered in collaboration with the developer. In addition, there is also OTP (One Time Password) verification for users as part of the application login security process. However, in Control 9, some applications are not appropriate, such as in sub-controls 9.2.5 and 9.2.6, where users of the Simpel Desa application are residents as ordinary users and village officials as users who have special rights. There has yet to be a periodic review of access rights for residents and village officials who no longer have the right to access the application.

Then, in controlling 10 cryptography of Simpel Desa applications, cryptographic methods have been applied to secure user data. Still, the results of the analysis carried out on control 10.1.1 policy on the use of cryptographic control have not found a policy that regulates the use of cryptography in Simpel Desa applications. From the two analytical test results, namely static analysis and dynamic analysis, it can be seen if there are vulnerabilities or weaknesses in the Simpel Desa application, primarily if it is associated with three main aspects of information security.

#### 1. Data confidentiality

Static analysis on Application permissions: the ability of applications to read data can potentially be modified and even stolen by hackers. In addition, the alert window allows malicious applications to access the device's screen access.

Static analysis on Network Security: The presence of unsafe data traffic on the network can cause data theft during the process.

Static analysis on Hardcode secret: This analysis found sensitive information in the form of user account information ranging from usernames, passwords, and others that could potentially become access to data theft/illegal use of data.

#### 2. Data integrity

Static analysis on Weak Crypto: Using a weak safety technique due to a defect in the compression function that resulted in a collision. The collision in question is that the hacker can create the same hash value, so it cannot ascertain that the data has been changed or valid data.

#### 3. Availability

In dynamic analysis, API Monitoring, the results show that we have not found a performance weakness in the application programming interface. However, in some cases, the application is forced to close, causing data to be lost during the data input process.

Risk Management that the developer has made can be seen in static and dynamic analysis, namely SSL Pinning, with the result that an SSL certificate has been applied to prevent attacks on data communication lines. Second, on the results of the dynamic analysis: of root detection, there is no super-user activity in the application, so it can conclude that the risk of data theft has been planned for prevention.

### **D. Recommendations**

Information security recommendations are obtained based on the evaluation of the two ISO controls used and the results of the two analyzes on MobSF.

1. Pay attention to the use of required permits and authorizations
2. Review of application permissions and de-registration of users
3. Tighten user authentication procedures

4. There is a periodic review and improvement of the competence of users who have special access rights.
5. Improved security standards for user information assets.
6. Improved implementation of data security used.

The recommendations generated are intended for application developers through the village employee or the suggestion box provided on the website. This recommendation can use to maximize the application of information security by existing standards.

#### IV. CONCLUSION

The research shows that the Sempel Desa application still has vulnerabilities, as evidenced by penetration testing using MobSF. The static analysis found vulnerabilities in the application of Weak Crypto, Dangerous Permission, and Hardcode Secrets. Results of dynamic analysis found that it has not implemented Root Detection and Debugger Check Bypass. In addition, the results of mapping and analysis using ISO 27002:2013 are known if there is a weak implementation of cryptography as evidenced by the discovery of permits, services and activities that are categorized as dangerous. In the control subdomain, it is mapped that the application has the "not yet appropriate" status. In domain 9, seven things are not by access control security standards. In domain 9, one thing was found that did not comply with cryptographic standards, namely the policy of using cryptographic controls.

#### REFERENCES

- [1] Admin, "Sempel Desa," 2022. [www.simpeldesa.com](http://www.simpeldesa.com)
- [2] C. Hanifurohman and D. DurbinHutagalung, "Analisa Keamanan Aplikasi Mobile E-Commerce Berbasis Android Menggunakan Mobile Security Framework," *Prosiding Universitas Pamulang*, Vol. 1, No. 1, PP. 1–7, 2020, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/Proceedings/article/view/5195>
- [3] S. A. Sholikhatin and K. N. Isnaini, "Analysis of Information Security Using ISO 27001 and Triangular Fuzzy Number Weighting," *Jurnal Ilmiah Informatika*, Vol. 6, No. 1, PP. 43–49, Jun. 2021, doi: 10.35316/jimi.v6i1.1224.
- [4] I. A. Dianta and E. Zusrony, "Analisis Pengaruh Sistem Keamanan Informasi Perbankan Pada Nasabah Pengguna Internet Banking," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, Vol. 3, No. 1, P. 1, Feb. 2019, doi: 10.29407/intensif.v3i1.12125.
- [5] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Justindo, Jurnal Sistem & Teknologi Informasi Indonesia*, Vol. 2, No. 1, PP. 19–30, 2017, doi: 10.32528/justindo.v2i1.1037.
- [6] Fina Triana, Jon Endri, and Irma Salamah, "Implementation of CAESAR CIPHER Cryptography Techniques for Android Based Information Data Security," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, Vol. 4, No. 4, PP. 627–634, Aug. 2020, doi: 10.29207/resti.v4i4.1984.

- [7] K. N. Isnaini and D. Suhartono, "Evaluation of Basic Principles of Information Security at University Using COBIT 5," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, Vol. 21, No. 2, PP. 317–326, Mar. 2022, doi: 10.30812/matrik.v21i2.1311.
- [8] A. Mahfuth, "Security Knowledge Required To Improve Employee Security Behavior in Information Security Culture," *International Journal of Computer Science and Information Security*, Vol. 20, No. 2, 2022, [Online]. Available: <https://www.researchgate.net/publication/359187687>
- [9] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," *Eksplora Informatika*, Vol. 8, No. 1, P. 52, Sep. 2018, doi: 10.30864/eksplora.v8i1.139.
- [10] N. Matondang, I. N. Isnainiyah, and A. Muliawatic, "Analisis Manajemen Risiko Keamanan Data Sistem Informasi (Studi Kasus: RSUD XYZ)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, Vol. 2, No. 1, PP. 282–287, Apr. 2018, doi: 10.29207/resti.v2i1.96.
- [11] D. P. Agustino, "Information Security Management System Analysis Menggunakan ISO/IEC 27001 (Studi Kasus: STMIK STIKOM Bali)," *Eksplora Informatika*, Vol. 8, No. 1, P. 1, Sep. 2018, doi: 10.30864/eksplora.v8i1.130.
- [12] N. Kohli and M. Mohaghegh, "Security Testing Of Android Based Covid Tracer Applications," in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, Dec. 2020, PP. 1–6. doi: 10.1109/CSDE50874.2020.9411579.
- [13] P. Weichbroth and Ł. Łysik, "Mobile Security: Threats and Best Practices," *Mobile Information Systems*, Vol. 2020, PP. 1–15, Dec. 2020, doi: 10.1155/2020/8828078.
- [14] A. Kartono, A. Sularsa, and S. J. I. Ismail, "Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan Mobsf," 2019, Vol. 5, No. 1, PP. 146–151. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/8563/8431>
- [15] C. Hanifurohman and D. D. Hutagalung, "ANALISIS STATIS MENGGUNAKAN MOBILE SECURITY FRAMEWORK UNTUK PENGUJIAN KEAMANAN APLIKASI MOBILE E-COMMERCE BERBASIS ANDROID," *Sebatik*, Vol. 24, No. 1, PP. 22–28, Jun. 2020, doi: 10.46984/sebatik.v24i1.920.
- [16] E. Tansen and D. W. Nurdiarto, "Analisis dan Deteksi Malware dengan Metode Hybrid Analysis Menggunakan Framework MOBSF," *Jurnal Teknologi Informasi*, Vol. 4, No. 2, PP. 191–201, Dec. 2020, doi: 10.36294/jurti.v4i2.1338.
- [17] H. Shahriar, C. Zhang, M. A. Talukder, and S. Islam, *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, Vol. 919. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-57024-8.
- [18] F. Nurindahsari and B. Parga Zen, "ANALISIS STATIK KEAMANAN APLIKASI VIDEO STREAMING BERBASIS ANDROID MENGGUNAKAN MOBILE SECURITY FRAMEWORK (MOBSF)," *Cyber Security dan Forensik Digital*, Vol. 4, No. 2, PP. 63–80, Apr. 2022, doi: 10.14421/csecurity.2021.4.2.3373.
- [19] M. Zeybek, E. N. Yilmaz, and I. Alper Dogru, "A Study on Security Awareness in Mobile Devices," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, Nov. 2019, No. November, PP. 1–6. doi: 10.1109/UBMYK48245.2019.8965476.
- [20] D. Sulistyowati, F. Handayani, and Y. Suryanto, "Comparative Analysis and Design of Cybersecurity Maturity Assessment Methodology Using NIST CSF, COBIT, ISO/IEC 27002 and PCI DSS," *JOIV : International Journal on Informatics Visualization*, Vol. 4, No. 4, P. 225, Dec. 2020, doi: 10.30630/joiv.4.4.482.
- [21] S. R. Musyarofah and R. Bisma, "Pembuatan Standard Operating Procedure ( SOP ) Keamanan Informasi Berdasarkan Framework ISO / IEC 27001 : 2013 dan ISO / IEC

- 27002 : 2013 pada Dinas Komunikasi dan Informatika Pemerintah Kota Madiun,” JEISBI: Journal of Emerging Information Systems and Business Intelligence, Vol. 01, PP. 43–50, 2020, [Online]. Available: <https://ejournal.unesa.ac.id/index.php/JEISBI/article/view/36860>
- [22] S. Fenz and T. Neubauer, “Ontology-based information security compliance determination and control selection on the example of ISO 27002,” *Information & Computer Security*, Vol. 26, No. 5, PP. 551–567, Nov. 2018, doi: 10.1108/ICS-02-2018-0020.
- [23] I Made Sukarsa, I Made Rama Pradana, and Putu Wira Buana, “Implementasi Enkripsi dan Otentikasi Transmisi Data ZeroMQ Menggunakan Advanced Encryption Standard,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, Vol. 4, No. 6, PP. 1149–1156, Dec. 2020, doi: 10.29207/resti.v4i6.2581.
- [24] A. Syahir and C. C. Wen, “Secure Login Mechanism for Online Banking,” *JOIV : International Journal on Informatics Visualization*, Vol. 2, No. 3–2, P. 179, Jun. 2018, doi: 10.30630/joiv.2.3-2.136.
- [25] E. Kurniawan and I. Riadi, “Analisis Tingkat Keamanan Sistem Informasi Akademik Berdasarkan Standard ISO/IEC 27002:2013 Menggunakan SSE-CMM,” *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, Vol. 2, No. 1, P. 12, Feb. 2018, doi: 10.29407/intensif.v2i1.11830.
- [26] A. David Purba, I. K. Adi Purnawan, and I. P. Agus Eka Pratama, “Audit Keamanan TI Menggunakan Standar ISO/IEC 27002 dengan COBIT 5,” *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, Vol. 6, No. 3, P. 148, Dec. 2018, doi: 10.24843/JIM.2018.v06.i03.p01.
- [27] I. K. A. O. Ardita, I. G. N. Anom Cahyadi Putra, M. R. Kustiadie, G. N. M. Dika Varuna, and M. Y. Eka Prananda, “Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment,” *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, Vol. 10, No. 3, P. 279, Apr. 2022, doi: 10.24843/JLK.2022.v10.i03.p04.
- [28] B. Yankson, J. V. K, P. C. K. Hung, F. Iqbal, and L. Ali, “Security Assessment for Zenbo Robot Using Drozer and mobSF Frameworks,” in *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Apr. 2021, PP. 1–7. doi: 10.1109/NTMS49979.2021.9432666.
- [29] Admin, “What are CVSS Scores,” 2022. <https://www.balbix.com/insights/understanding-cvss-scores/> (accessed Nov. 24, 2022).
- [30] Gregory Manley, “What Is MD5 and Why Is It Considered Insecure?,” 2022. <https://www.section.io/engineering-education/what-is-md5/> (accessed Nov. 24, 2022).