Journal of
Cybersecurity and Privacy

Review

Bridging the Gap: A Survey and Classification of Research-Informed Ethical Hacking Tools

Paolo Modesti *, Lewis Golightly *, Louis Holmes, Chidimma Opara * and Marco Moscini

Department of Computing and Games, Teesside University, Middlesbrough TS1 3BX, UK;
b1445121@tees.ac.uk (L.H.); m.moscini@tees.ac.uk (M.M.)
* Correspondence: p.modesti@tees.ac.uk (P.M.); l.golightly@tees.ac.uk (L.G.); c.opara@tees.ac.uk (C.O.)

Abstract: The majority of Ethical Hacking (EH) tools utilised in penetration testing are developed by practitioners within the industry or underground communities. Similarly, academic researchers have also contributed to developing security tools. However, there appears to be limited awareness among practitioners of academic contributions in this domain, creating a significant gap between industry and academia's contributions to EH tools. This research paper aims to survey the current state of EH academic research, primarily focusing on research-informed security tools. We categorise these tools into process-based frameworks (such as PTES and Mitre ATT&CK) and knowledge-based frameworks (such as CyBOK and ACM CCS). This classification provides a comprehensive overview of novel, research-informed tools, considering their functionality and application areas. The analysis covers licensing, release dates, source code availability, development activity, and peer review status, providing valuable insights into the current state of research in this field.

Keywords: ethical hacking; tools and techniques; research-informed; classification; PTES; Mitre ATT&CK; CyBOK; ACM CCS

check for updates

Citation: Modesti, P.; Golightly, L.; Holmes, L.; Opara, C.; Moscini, M. Bridging the Gap: A Survey and Classification of Research-Informed Ethical Hacking Tools. *J. Cybersecur. Priv.* **2024**, *4*, 410–448. https://doi.org/10.3390/jcp4030021

Academic Editors: Carlos Rabadão and Mario Antunes

Received: 11 April 2024
Revised: 3 June 2024
Accepted: 23 June 2024
Published: 16 July 2024

1. Introduction

In the domain of Ethical Hacking (EH), developing innovative tools is essential to tackle emerging threats and vulnerabilities. Ethical Hacking tools are designed mainly by industry practitioners, occasionally by underground communities [1], and sometimes even by state actors [2]. However, even experienced security developers may overlook critical requirements for such applications. An intriguing example is provided by Valenza et al. [3], challenging the conventional belief that remote scanning carries negligible risk. Their methodology, which transformed the scanning system into a target for counterattacks, revealed vulnerabilities in widely deployed tools, including Metasploit Pro. Overall, the researchers identified weaknesses in 36 out of 78 scanning applications.

The existing divide between industry and academia in developing EH tools reflects differing goals and approaches, highlighting a significant awareness gap. Industry practitioners are often insufficiently informed about the outcomes and insights generated by academic research in this field. Driven by immediate operational requirements, the industry tends to favour established tools and practices that promptly address real-time threats. However, this emphasis on practical application can result in a lack of awareness regarding significant academic contributions, such as novel methodologies and solutions for emerging threats or advancements in theoretical frameworks. Consequently, research findings may remain underutilised by industry practitioners.

One way to bridge the gap between industry and academia in developing EH tools is by producing in-depth survey papers that detail the tools created by both communities. However, existing surveys primarily assess and compare tools used by industry practitioners, with only occasional consideration of research-informed tools [4–6]. This focus overlooks the innovative contributions of the research community.

Additionally, the quantity and breadth of tools reviewed by state-of-the-art surveys in EH tools are limited. For instance, the work by Altulaihan et al. [7] covered 15 papers for web application penetration testing, while Yaacoub et al. [5] reviewed 13 tools specifically applied to IoT. This limited scope restricts the comprehensive evaluation of EH tools. Moreover, existing surveys that classify EH methodologies or frameworks compare existing frameworks such as PTES or other industry methodologies like the Information Systems Security Assessment Framework (ISAF) [8]. However, they do not discuss the specific tools that fall under each category. This narrow focus fails to provide a holistic view of the EH tools landscape.

1.1. Research Contributions

In light of these limitations, this paper makes two significant contributions:

1. *Survey of Research-informed EH Tools*: This study surveys 100 research-informed EH tools developed in the last decade. It highlights key areas such as licensing, release dates, source code availability, development activity level, and peer review status. This analysis aims to provide insights into the state-of-the-art EH tools developed by the research community.
2. *Alignment with Recognised Frameworks*: This study categorises the tools into *process-based* frameworks, such as the *Penetration Testing Execution Standard* (PTES) [9], and the *Mitre ATT&CK framework* [10] and *knowledge-based* frameworks like the *National Cyber Security Centre's Cyber Security Body Of Knowledge* (CyBOK) [11] and the *Association for Computing Machinery's Computing Classification System* (ACM CCS) [12]. Combining these four classifications offers an informative view of the landscape of novel and research-informed ethical tools, their functionality, and application domain for the benefit of scholars, researchers, and practitioners.

This comprehensive approach not only bridges the gap between industry and academia but also ensures that Ethical Hacking tools evolve in tandem with the ever-changing cyber threat landscape.

1.2. Outline of the Paper

Section 2 introduces the background of EH and the methodologies used by practitioners, Section 3 presents our research methodology, and Section 4 discusses the classification criteria applied to the EH tools. Section 5 discusses the tool categorisation into process-based and knowledge-based frameworks. Section 6 presents the systematic evaluation of research-informed EH tools, while Section 7 concludes the paper.

2. Background

In this section, we discuss the background and fundamentals of Ethical Hacking, including the motivations behind hacking systems and the different motivations of hackers categorised and represented using *hats*. Additionally, we introduce methodologies used in EH.

2.1. (Unethical) Hacking Landscape and Motivations

Cyberattacks, intrusion techniques, social engineering, and information manipulation are increasingly becoming more sophisticated, targeting individuals and organisations. The objective of each attack, regardless of its nature, is to circumvent the three primary principles of security: *confidentiality*, *integrity*, and *availability* [13]. There is a wide range of motivations for cyberattacks, and many factors interplay. According to [14–17], the motivations for these attacks can be grouped into:

- *Economic Gain*: Cybercriminals often target individuals, businesses, or organisations to extort money through ransomware [18] or financial fraud. Financial institutions such as banks and related services can be a target, as in the case of the attack on the SWIFT international transaction system [19].

- *Competitive Advantage and Sabotage*: Competing companies, state-sponsored actors, and individuals can steal and reveal industrial secrets and intellectual properties to gain a competitive edge and compromise the data integrity and accessibility in businesses. While the WannaCry ransomware was used primarily to extort money from the victims, the attack on the UK National Health Service (NHS) could have also been conducted to demonstrate the business complacency and lack of digital transformation [18].
- *Personal Revenge*: Cyberattacks driven by personal revenge are often perpetrated by disgruntled insiders or individuals with a vendetta against specific targets. These attacks leverage insider knowledge or access to inflict damage, disrupt operations, or steal sensitive data.
- *Political*: The attack is carried out as groups of hackers engaged in politics, sponsored-stated hacking teams aiming at damaging specific targets. This includes governmental institutions, political parties, social society organisations and other public subjects. Examples are the alleged interference in the US presidential elections by Russian state-sponsored cyber actors in 2016 [20], and the *Operation Socialist* in 2010–2013 against Belgacom attributed to the UK's GCHQ [21], a case of an attack perpetrated by a NATO member state against another one.

The activities described above broadly fall into the category of cybercrime and involve hacking, data theft, identity theft, financial fraud, and malware distribution. However, when cyberattacks are carried out by state-sponsored actors against other nations or entities, they are often called cyber warfare. The distinction can be blurred in some cases, as the direct involvement of government organisations can have surprising ramifications and side effects.

The *EternalBlue* exploit [22] was developed by the United States National Security Agency (NSA) targeting a vulnerability in Microsoft's Windows operating system, specifically in the Server Message Block (SMB) protocol. The NSA utilised the exploit for years without reporting the vulnerability to Microsoft. However, it became widely known when a hacking group called the Shadow Brokers leaked the exploit in April 2017. The most notorious incident involving EternalBlue was the aforementioned WannaCry attack in May 2017. The exploit allowed the rapid spread of malware across networks, affecting hundreds of thousands of computers in over 150 countries and causing hundreds of millions of USD of damage worldwide.

This demonstrates the potential for unintended consequences and collateral damage as malicious actors can weaponise an offensive security tool developed by a government agency for large-scale cybercrime. This case also highlights the importance of responsible handling and disclosure of vulnerabilities by any entity, including government intelligence agencies.

2.2. Ethical Hacking

Ethical Hacking, also known as penetration testing, aims to identify vulnerabilities in computer systems, networks, and software applications before real-world attackers can exploit them. By uncovering weaknesses and providing recommendations for mitigation, EH helps organisations enhance their defences, protect sensitive data, and prevent unauthorised access. Ethical hackers utilise their skills to simulate potential cyberattacks and assess the security of a system. In fact, such specialists essentially utilise the same techniques as cyber attackers, with the important difference being that the system's owner authorises them and agrees on the scope of the penetration testing exercise. As individuals capable of compromising systems, any misuse of their skills is criminally punishable according to the laws of various countries. The Budapest Convention on Cybercrime of 2001 (Article 6) [23], the EU Directive 2013/40 (Article 7) [24], and the UK Computer Misuse Act of 1990 (Section 3.1) [25] are some of the legislations that regulate cybersecurity activities in terms of the improper use of personal capabilities, software, and hardware dedicated to unauthorised access to third-party information.

The key issue in determining the legality of hacking activities is avoiding actions that contravene the law. Hacking professionals must prioritise legal compliance to avoid prosecution. While the term *Legal Hacking* may more accurately describe this focus on legality, it is essential to recognise that legality does not always equate to ethical behaviour. Nevertheless, the term *Ethical Hacking* remains widely used, emphasising the importance of both legal compliance and ethical conduct in the profession.

2.3. Ethical Hackers

Traditional media often portrays hackers as mysterious figures, typically depicted wearing hoodies in dimly lit rooms, perpetuating a stereotype prevalent in pop culture. Hackers are commonly seen as computer pirates who infiltrate systems for personal or financial gain. However, historical context reveals a more nuanced understanding. According to the classic definition reported by Gehring [26], hackers enjoy the intellectual challenge of overcoming programming limitations and seeking to extend their capabilities. This definition, prevalent until the 1980s and intertwined with notions of democracy and freedom, has evolved, as discussed by Jaquet-Chiffelle and Loi [27]. *Hats* of different colours are broadly used as a symbolic representation of individuals based on their intentions and actions related to hacking [15]:

- *White Hat* (ethical): Embodies the principles of hacker culture by employing technical skills to proactively enhance system security measures. These individuals focus on identifying vulnerabilities and developing defensive strategies to mitigate potential risks.
- *Black Hat* (malicious): Represents individuals who maliciously exploit vulnerabilities within systems for personal gain or disruptive purposes. Their actions typically involve unauthorised access, data theft, and system manipulation, often resulting in financial losses or reputational damage for targeted entities.
- *Grey Hat* (undecided): Occupies an intermediary role, engaging in activities that blur the line between ethical and malicious hacking. These individuals engage in operations as both Black Hat and White Hat, depending on the circumstances [27].

In recent years, cybersecurity and privacy protection have emerged as central themes for all organisations, and professional roles have arisen to address these needs. Penetration testing and malware analysis are among the most sought-after roles in the cybersecurity job market, falling under the main category of EH.

2.4. Ethical Hacking Methodologies

Penetration testing takes different forms and can cover various areas. Yaacoub et al. [14] describe the process of conducting an attack in five main phases:

- *Reconnaissance*: The hacker gathers information on systems and users through passive or active techniques. This includes physical methods like social engineering and analysing network packets to identify details such as network configuration, hardware, and security measures.
- *Scanning*: The hacker searches for vulnerabilities in systems through simulated tests, including identifying open ports, active hosts, and weak firewall configurations. Enumeration is then carried out to gather further information while maintaining an active connection.
- *Gaining Access*: The hacker attempts to access the system using penetration testing tools and techniques, aiming to bypass security measures.
- *Maintaining Access*: The hacker establishes backdoors or rootkits to maintain remote access with elevated privileges.
- *Covering Tracks*: The hacker eliminates evidence that could reveal their identity or traces of the attack.

Each phase is complex and crucial for the success of a cyber attack. Due to the unique nature of each system, there are no strict rules for systematically executing an attack or penetration test. However, various frameworks and methodologies have been developed to guide the penetration testing process in planning and executing cyber attack simulations.

These frameworks can be categorised into three main areas: open source, maintained by non-profit organisations or security institutes; industrial/governmental, maintained by government entities such as the National Institute of Standards and Technology (NIST); and proprietary, maintained by private companies and accessible through payment of a usage license.

2.4.1. PTES

The Penetration Testing Execution Standard [9] was created in 2009 by a group of practitioners who developed this framework to provide both businesses and security service providers with a common language for conducting penetration tests. It comprises seven phases: *Pre-engagement Interactions*, *Intelligence Gathering*, *Threat Modelling*, *Vulnerability Analysis*, *Exploitation*, *Post-Exploitation*, and *Reporting*. The methodology is presented in detail in Section 4.1.

2.4.2. Mitre ATT&CK

Mitre Adversarial Tactics, Techniques, and Common Knowledge [10] is a matrix that describes the behaviour of attackers throughout the life cycle of an operation. It covers *Tactics, Techniques, and Procedures (TTP)* used by threat actors to achieve their objectives. Mitre is a non-profit American company involved in numerous cybersecurity standards and frameworks, such as *CVE (Common Vulnerabilities and Exposures)* [28] to identify and classify disclosed security vulnerabilities. *CWE (Common Weakness Enumeration)* [29] is used as a common language for weakness identification, prevention and mitigation. See Section 4.2 for the details of this methodology.

2.4.3. PCI DSS Penetration Testing Guidance

Payment Card Industry Data Security Standard (PCI DSS) [30] is a set of security requirements designed to protect payment card information during transactions. Developed by the PCI Security Standards Council, it applies to all organisations that accept, process, store, or transmit payment card data. PCI DSS establishes requirements for data security, network management, application protection, and other measures to prevent credit card fraud. The Penetration Testing Guide [31] is divided into four parts: *Penetration Tester Components*, *Qualification of a Penetration Tester*, *Methodology* and *Reporting and Documentation*. Organisations must comply with these requirements to ensure the security of credit card transactions and protect cardholders' sensitive data.

2.4.4. ISSAF

The *Information Systems Security Assessment Framework* [32] is a standard supported by the Open Information System Security Group (OISSG). It incorporates all possible attack domains, and the main feature is that the penetration testing activity is divided into three phases: *Planning and Preparation*, *Assessment* and *Reporting, Cleanup, and Artefact Destruction*.

2.4.5. OSSTMM

The *Open Source Security Testing Methodology Manual* [33] is a set of guidelines and procedures for conducting security tests and assessing the security of information systems. Developed by the Institute for Security and Open Methodologies (ISECOM), OSSTMM aims to provide an open-source standardised methodology for cybersecurity professionals. It focuses on testing operational security through five channels: *Human Security*, *Physical Security*, *Wireless Communications*, *Telecommunications*, and *Data Networks*.

2.4.6. NIST800-115

NIST800-115 [34] is a methodology published by the National Institute of Standards and Technology (NIST) in the United States. This standard provides detailed guidelines for conducting tests and assessments of information security in computer environments. It covers a wide range of security testing and assessment activities. The standard includes planning, information gathering, vulnerability analysis, test execution, risk assessment, and documentation. It offers practical and detailed recommendations for performing various tests, including tool selection and management of information collected during the testing process. Although published by NIST, the standard is designed to be adopted in both the public and private sectors, providing a flexible framework that can be applied to different environments.

2.4.7. OWASP

The *Open Worldwide Application Security Project* [35] was launched in 2001 as an open-source project that provides guidelines, tools, and methodologies to improve the security of applications, collected into a guide named OTG 4.0 (*Owasp Testing Guide*) [36]. This document is divided into five parts: *Before development begins*, *During definition and design*, *During development*, *During deployment* and *Maintenance and operations*. By completing these procedures, developers can significantly reduce the risk of data breaches caused by attacks facilitated by poor code quality.

3. Survey Methodology

A three-step approach was devised to investigate Ethical Hacking tools developed by the research community over the past decade (Figure 1). First, clear guidelines were established to determine the inclusion of tools in the survey. Second, relevant papers and tools satisfying the above criteria were identified. Finally, these tools were categorised based on established cybersecurity frameworks.

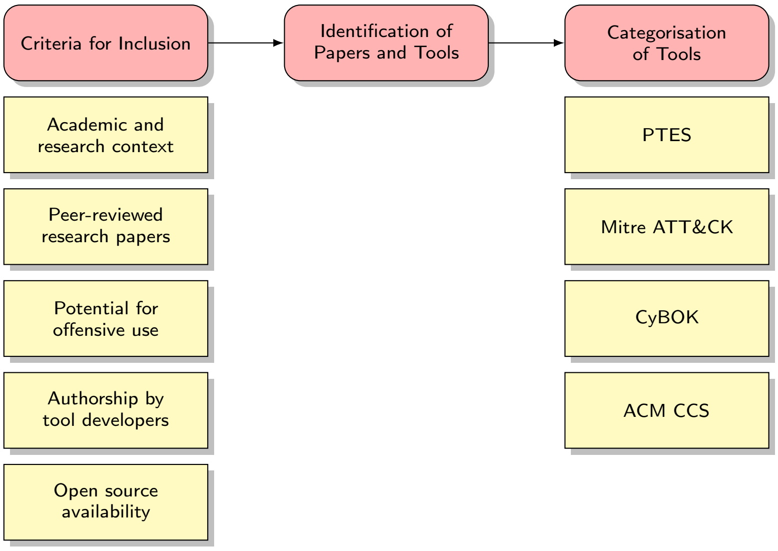
Criteria for Inclusion
Identification of Papers and Tools
Categorisation of Tools
Academic and research context
Peer-reviewed research papers
Potential for offensive use
Authorship by tool developers
Open source availability
PTES
Mitre ATT&CK
CyBOK
ACM CCS

Figure 1. Survey methodology.

3.1. Criteria for Inclusion of Ethical Hacking Tools in the Paper

This survey established specific conditions to determine the inclusion of state-of-the-art EH tools in the paper. The following criteria were adhered to:

- *Academic and research context*: The tool has been developed within an academic/research project: this excludes any tools developed primarily as practitioner tools (e.g., they are included in a popular EH distribution, like Kali Linux).
- *Peer-reviewed research papers*: Each EH tool included in the survey must be published in a peer-reviewed venue. Peer review validates the tool's architecture, functionalities, and relevance.
- *Potential for offensive use*: The tool has at least the potential to be used in an offensive context even if authors do not state that explicitly, as the tool could have been

developed for another purpose (e.g., software testing, supporting software or system development).

- *Authorship by tool developers*: The survey also requires that the authors of the papers have designed/developed the tool. This criterion ensures credibility and depth of insight, as the creators are directly involved in its conception and development.
- *Open source availability*: The tool should be open source, and the source code (or distribution package) should be freely available. However, this requirement was relaxed throughout the research as we acknowledged that some tools may not be open-source for various reasons, such as their proprietary nature, pending patents, or limited accessibility.

3.2. Collating Research-Informed Ethical Hacking Tools

The inclusion of 100 EH tools was driven by the aim of achieving a balance between depth and breadth in our analysis. The selection process was systematic and rigorous, inspired by the PRISMA methodology [37], with identification, screening and inclusion phases.

Initially, we initiated a collaborative research project involving cohorts of students from the MSc Cybersecurity program at Teesside University (UK). Despite the absence of a dedicated module on EH in their curriculum, the students showed considerable interest in working within this domain. This project allowed them to integrate EH professionalism with their research interests.

The students' initial submissions yielded over 200 academic references. However, after a careful review process conducted by the authors, approximately 30 tools aligned with the research scope and were included in this paper. Many tools found by the students were excluded for these reasons:

- Difficulty of the students in distinguishing between research-informed and practitioner tools.
- Confusion between papers describing the design and implementation of a tool and those describing its application.
- The approach of identifying tools first and then searching for papers to support the findings leads to the above misconceptions.

Following the criteria outlined in Section 3.1, the authors expanded the total count of EH tools to 185. These tools were then resampled, and 100 tools were finally selected. The final selection was based not only on adherence to the criteria in Section 3.1 but also on the fact that these tools were not merely applications of existing methodologies, frameworks, or aggregations of practitioner tools.

In fact, among the 85 candidates excluded in the final round, 28 focused on mitigation tools and techniques, 18 on methodologies and frameworks, 14 on the application of practitioner tools, 13 were surveys, 5 addressed socio-technical aspects, 4 were simulation tools, and 3 focused on education.

For details of the 100 tools surveyed in this paper, see Table 1.

Moreover, as discussed in Section 6, to include a significant number of tools that could reflect the current state-of-the-art, we had to relax on the criteria of availability of the source code. Therefore, in this survey, we considered 41 tools that satisfy all other criteria, but no source code has been published.

3.3. Classification of Identified Ethical Hacking Tools

In the second phase of the research, the identified EH tools were classified according to established cybersecurity frameworks. This task was undertaken by the authors, who have extensive expertise in EH from years of teaching, research, and professional experience in the field.

All 100 identified tools were categorised according to the following classifications (Figure 2):

1. Penetration Testing Execution Standard (PTES) [9].
2. Mitre ATT&CK framework [10].
3. NCSC CyBOK [11].
4. ACM Computing Classification System (CCS) [12].

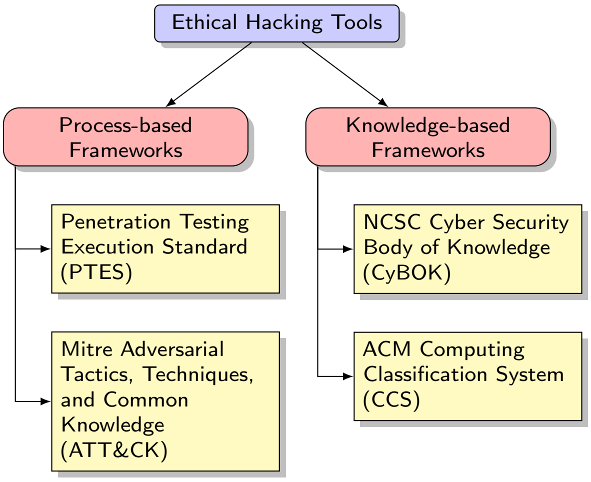
Ethical Hacking Tools
Process-based Frameworks
Knowledge-based Frameworks
Penetration Testing Execution Standard (PTES)
Mitre Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK)
NCSC Cyber Security Body of Knowledge (CyBOK)
ACM Computing Classification System (CCS)

Figure 2. Classification criteria applied in this survey.

Incorporating *process-based* classifications, such as the PTES and the Mitre ATT&CK framework, ensures that the survey covers the practical aspects of EH tools. On the other hand, *knowledge-based* classifications such as NCSC CyBOK and ACM CCS focus on the theoretical and conceptual aspects of computing and cybersecurity domains, thereby exploring the underlying theoretical bases of EH tools.

The next section provides an in-depth discussion of these frameworks.

4. Cybersecurity Frameworks Used for Tools Classification

This section delves into a detailed examination of the cybersecurity frameworks used to categorise the Ethical Hacking tools surveyed in this paper. These frameworks include PTES, Mitre ATT&CK framework, CyBOK, and ACM CCS.

4.1. Penetration Testing Execution Standard

PTES [9] is a standardised methodology for planning, executing, and reporting security tests and it is widely used within the cybersecurity industry as one of the most significant standards for conducting penetration tests. PTES was proposed by a group of penetration testers and security professionals to provide guidance and best practices for conducting effective penetration tests within legal and ethical boundaries. It consists of seven phases (Figure 3):

1. *Pre-engagement Interactions*: In this phase, the scope and rules of engagement are defined through an agreement between the penetration testing team and the system's owner. The system's owner must provide permissions and authorisations, and communication lines must be established between the testers and the target organisation.
2. *Intelligence Gathering*: Information about the target organisation or system is collected using techniques such as open-source intelligence (OSINT) gathering, reconnaissance, and network scanning. Active and passive information-gathering methods are distinguished based on direct interaction with the target system.
3. *Threat Modelling*: This phase identifies potential vulnerabilities and threats specific to the target organisation or system. It involves analysing collected information, understanding infrastructure and architecture, prioritising attack vectors, and assigning risks to threats to inform vulnerability mitigation.
4. *Vulnerability Analysis*: Vulnerabilities and weaknesses in the target's systems and applications are identified and assessed, typically using classification systems like

the *Common Vulnerability Scoring System* (CVSS). Manual and automated testing, configuration analysis, and examination of insecure application design may be involved.

5. *Exploitation*: Vulnerabilities previously identified are exploited to compromise the target system, gain unauthorised access, or execute malicious activities. The goal is to demonstrate the impact of vulnerabilities and their potential exploitation, bypassing security mechanisms.
6. *Post-Exploitation*: After successful exploitation, the focus shifts to determining the value of the compromised system, maintaining access, escalating privileges, and pivoting to other systems within the network. This simulates an attacker's post-compromise activities, considering the data's importance and the advantage provided for further attacks.
7. *Reporting*: The final phase involves documenting the findings, including identified vulnerabilities, their impact, and recommendations for remediation. The report should be clear, concise, and actionable for the target organisation, tailored to various audiences ranging from senior managers to technical staff.

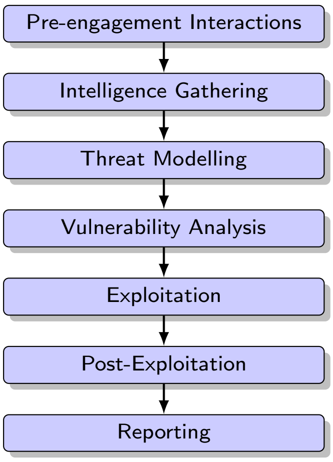

Figure 3. Penetration testing phases according to the PTES methodology.

The PTES methodology can be applied to various systems to assess their security, including networks and critical infrastructures [38].

4.2. Mitre ATT&CK Framework

Mitre ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) [10] is a framework that categorises the tactics, techniques, and procedures (TTPs) used by real-world threat actors during cyberattacks (Figure 4). It provides a standardised and comprehensive mapping of the various stages of an attack and consists of a matrix that categorises adversary behaviours across different stages of the attack lifecycle. Within each tactic, specific techniques and procedures are listed, which outline the specific actions and methods used by adversaries to accomplish their objectives.

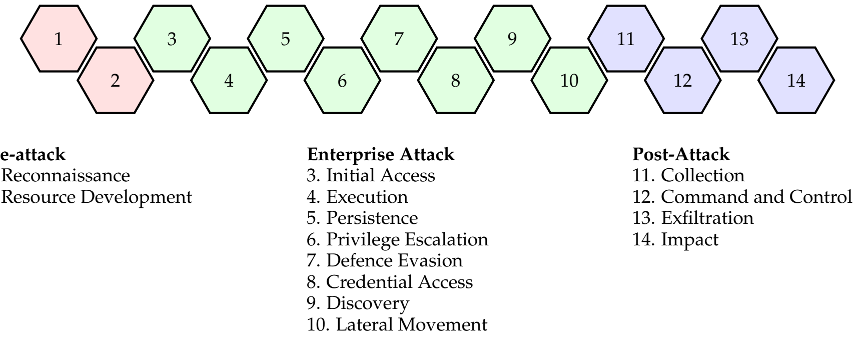

Figure 4. Mitre ATT&CK framework: phases and tactics (inspired by [1]).

Mitre ATT&CK consists of fourteen phases [39]:

1. *Reconnaissance*: Collecting information on the target to plan and execute attacks. Methods include: *Active Scanning*, *Passive Scanning*, *Social Engineering* and *OSINT*.
2. *Resource Development*: Acquiring resources required for further exploitation and maintaining access. Methods include: *Developing Tools* and *Developing and Executing Malware*.
3. *Initial Access*: Techniques performed to gain access to the target environment. Methods to achieve this include: *Spear-Phishing*, *Exploiting Vulnerabilities* and *Stolen Credentials*.
4. *Execution*: Techniques performed executing Malicious Software (Malware) on a target system. Methods include: *Executing Binaries*, *Scripts* and *System Tools*.
5. *Persistence*: Techniques performed around maintaining system access over a significant period of time. Methods include: *Backdoor Creation* and *Scheduled Tasks*.
6. *Privilege Escalation*: Increasing the access control levels in the compromised environment. Methods include: *Vulnerability Exploitation*, *Configuration Manipulation* and *Credential Theft*.
7. *Defence Evasion*: Techniques to avoid detection or target defensive mechanisms. Methods include: *Anti-Virus Evasion*, *Obfuscation* and *Living-off-the-land Techniques*.
8. *Credential Access*: Techniques for stealing credentials for unauthorised access. Methods include: *Credential Dumping*, *Keylogging* and *Brute-Force Attacks*.
9. *Discovery*: Techniques for identifying information about the target system. Methods include: *Network Scanning*, *System Enumeration* and *Querying Systems*.
10. *Lateral Movement*: Methods for moving through the network for accessing additional systems by using *RDP*, *Trust Relationships* and *Lateral Tool Transfer*.
11. *Collection*: Acquiring and consolidating target system information. Methods include: *Data Mining*, *Scraping* and *Information Capture*.
12. *Command and Control*: Creating and Maintaining communication channels between the attacker and compromised systems. Methods include: *Command and Control (C2)*, *Covert Channels* and *Network Protocols*.
13. *Exfiltration*: Techniques around the unauthorised data transfer external to the target environment. Methods include: *Network Data Exfiltration*, *Encryption Channels* and *Scheduled Transfer*.
14. *Impact*: Achieving the desired outcome or effect could involve damaging a target. Methods include: *Destroying Data*, *System Operation Disruption* and *Deploying Malware*.

The Mitre ATT&CK framework can be applied to broad kinds of targets, including, financial systems [40], healthcare [41] and Industrial Control Systems (ICS) [42].

4.3. NCSC CyBOK

The *Cyber Security Body of Knowledge* (CyBOK) [11] is a project developed by the United Kingdom's National Cyber Security Centre (NCSC), a child agency of the Government Communications Headquarters (GCHQ), in collaboration with academia, industry, and government partners. CyBOK aims to provide a comprehensive and authoritative reference for the knowledge, skills, and competencies on which various educational programmes and job descriptions may be aligned. The CyBOK is divided into 21 top-level *Knowledge Areas* (KAs) and five broad categories, along with the introductory concepts, as shown in Figure 5. These categories, while orthogonal, are not entirely separate from each other, reflecting the interdisciplinary nature of cyber security.

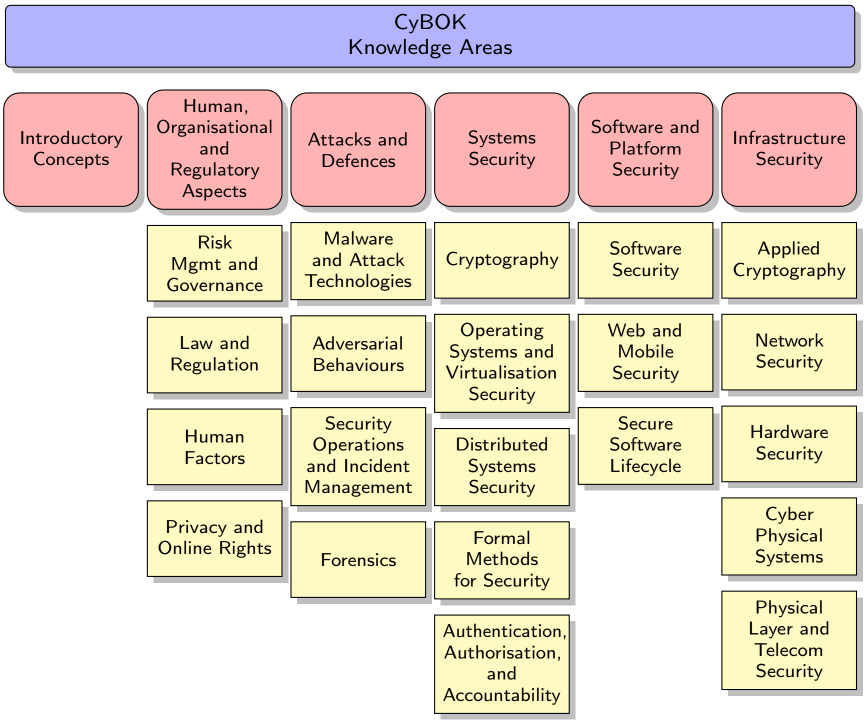
CyBOK
Knowledge Areas
Introductory Concepts
Human, Organisational and Regulatory Aspects
Attacks and Defences
Systems Security
Software and Platform Security
Infrastructure Security
Risk Mgmt and Governance
Malware and Attack Technologies
Cryptography
Software Security
Applied Cryptography
Law and Regulation
Adversarial Behaviours
Operating Systems and Virtualisation Security
Web and Mobile Security
Network Security
Human Factors
Security Operations and Incident Management
Distributed Systems Security
Secure Software Lifecycle
Hardware Security
Privacy and Online Rights
Forensics
Formal Methods for Security
Cyber Physical Systems
Authentication, Authorisation, and Accountability
Physical Layer and Telecom Security

Figure 5. Cyber Security Body Of Knowledge (CyBOK) Knowledge Areas.

In summary, the Knowledge Areas in CyBOK version 1.1 [43] are organised as follows:

1. Introductory Concepts: *Introduction to CyBOK*.
2. Human, Organisational and Regulatory Aspects: (a) *Risk Management and Governance*, (b) *Law and Regulation*, (c) *Human Factors* and (d) *Privacy and Online Rights*.
3. Attacks and Defences: (a) *Malware and Attack Technologies*, (b) *Adversarial Behaviours*, (c) *Security Operations and Incident Management* and (d) *Forensics*.
4. Systems Security: (a) *Cryptography*, (b) *Operating Systems and Virtualisation Security*, (c) *Distributed Systems Security*, (d) *Formal Methods for Security* and (e) *Authentication, Authorisation, and Accountability*.
5. Software and Platform Security: (a) *Software Security*, (b) *Web and Mobile Security* and (c) *Secure Software Lifecycle*.
6. Infrastructure Security: (a) *Applied Cryptography*, (b) *Network Security*, (c) *Hardware Security*, (d) *Cyber Physical Systems* and (e) *Physical Layer and Telecommunications Security*.

1. Introductory Concepts: *Introduction to CyBOK*.
2. Human, Organisational and Regulatory Aspects
   (a) *Risk Management and Governance*: Asset assessment, identification and management.
   (b) *Law and Regulation*: Regulatory Compliance with national and international legislation.
   (c) *Human Factors*: Physical and Digital Social Engineering techniques targeting the human state vulnerability characteristics and exploiting these in a cybersecurity context.
   (d) *Privacy and Online Rights*: Purpose limitation, data transparency, and minimisation.

3. Attacks and Defences
    (a) *Malware and Attack Technologies*: Attack techniques, analysis, and detection of malware, including response using evasive countermeasures and disruption of malware operations.
    (b) *Adversarial Behaviours*: Characterising cybercriminals based on their motivation (e.g., financial, political, etc.), types of cyber offences (cyber-enabled and cyber-dependent crimes), and the activities performed in a cyber attacks.
    (c) *Security Operations and Incident Management*: The management of secure systems, including the setup, operation, maintenance, incident response, and using threat intelligence for detection and security measures.
    (d) *Forensics*: Data acquisition, file system and block device analysis, as well as data recovery and file content carving, including SaaS.
4. Systems Security
    (a) *Cryptography*: Techniques for securing data and communications: encryption algorithms, cryptographic protocols, key management, and others.
    (b) *Operating Systems and Virtualisation Security*: Authentication and identification, Access Control Lists (ACL), memory protection and address spaces, and physical access and secure deletion.
    (c) *Distributed Systems Security*: Access and identity management, data transportation, resource management and coordination of services, and data security.
    (d) *Formal Methods for Security*: Analysis and verification of security properties of systems using formal specification languages and mathematical models.
    (e) *Authentication, Authorisation, and Accountability*: Mechanisms for verifying the identities of users, controlling access to resources, and maintaining audit trails for accountability purposes.
5. Software and Platform Security
    (a) *Software Security*: Language-based security techniques aimed at preventing vulnerabilities applied to system design and implementation: type systems, memory management, code generation, and others.
    (b) *Web and Mobile Security*: Security challenges specific to web and mobile applications, including secure communication protocols and protections against common threats such as CSRF, XSS, and SQL Injection.
    (c) *Secure Software Lifecycle*: Ensuring software security by integrating security software engineering techniques throughout the development lifecycle.
6. Infrastructure Security
    (a) *Applied Cryptography*: Cryptographic techniques applied in securing infrastructure components.
    (b) *Network Security*: Securing network infrastructure and communications, SDN and NFV security, network access control, and zero trust networking.
    (c) *Hardware Security*: Secure element, smart card, and trusted platform module (TPM).
    (d) *Cyber Physical Systems*: Securing industrial control systems, electrical power and smart grids, autonomous vehicles, robotics, medical devices, and IoT.
    (e) *Physical Layer and Telecommunications Security*: Securing telecommunications networks and physical communication channels, NFC, air traffic communication networks, cellular networks, and others.

The CyBOK can be applied in various ways to enhance the security posture of businesses. It can be used to assess skills, develop workforces, design curricula in higher education, and for certification programs [44,45].

4.4. ACM Computing Classification System (CCS)

The *Computing Classification System* (CCS) [12] is a taxonomy developed by the Association for Computing Machinery (ACM). It is designed to categorise and organise the various areas of research and practice within the field of computing. The CCS provides a hierarchical structure that classifies research papers, articles, conference proceedings, and other scholarly works in computing. Authors use appropriate CSS categories when submitting publication manuscripts to journals and conferences for classification and organisation. This system helps to locate relevant literature, understand the structure of the field, and facilitate communication within the computing community. The root concepts of ACM CSS include [12]:

1. *General and Reference:* Fundamental concepts and cross-disciplinary topics in computing.
2. *Hardware:* Physical components and architecture of computing systems.
3. *Computer Systems Organisation:* Organisation and structure of computer systems.
4. *Networks:* Communication and connectivity in computing environments.
5. *Software and its Engineering:* Development, design, and maintenance of software systems.
6. *Theory of Computation:* Mathematical and theoretical aspects of computation.
7. *Mathematics of Computing:* Mathematical foundations of algorithms and computation.
8. *Information Systems:* Management, retrieval, and processing of information in computing.
9. *Security and Privacy:* Protection of computing systems and data privacy concerns.
10. *Human-Centred Computing:* Interaction between humans and computing technologies.
11. *Computing Methodologies:* Methodological approaches in computing research and practice.
12. *Applied Computing:* Application of computing techniques in various domains.
13. *Social and Professional Topics:* Ethical, legal, and social aspects of computing.

5. Classification

The complete list of identified tools is available in Table 1. We include the availability of the source code, the license type, the source code repository and the year of publication. For readability reasons, we put the other classification tables in Appendix A and the review of the tools in the Supplementary Material.

Table 1. Classified tools, licence type and source code availability.

Tool Name	*Year*	*License Type*	*Source Code Repository*
ADaMs [46]	2021	MIT License	https://github.com/TheAdamProject/adams
AIBugHunter [47]	2023	MIT License	https://github.com/awsm-research/aibughunter
ARMONY [48]	2013	Not Available	Not Available
Autosploit [49]	2020	Not Available	Not Available
AVAIN [50]	2019	MIT License	https://github.com/ra1nb0rN/Avain
Bbuzz [51]	2017	MIT License	https://github.com/lockout/Bbuzz
Black Ostrich [52]	2023	Not Available	Not Available
Black Widow [53]	2021	Not Specified	https://github.com/SecuringWeb/BlackWidow
Bleem [54]	2023	Not Available	Not Available
Cairis [55]	2020	Apache 2.0	https://github.com/cairis-platform/cairis
Censys [56]	2015	Apache 2.0 + ISC	https://github.com/zmap/zgrab2
Chainsaw [57]	2016	Not Available	Not Available
Chucky [58]	2013	GPLv3	https://github.com/a0x77n/chucky-ng/
Commix [59]	2019	GPLv3	https://github.com/commixproject/commix
CryptoGuard [60]	2019	GPLv3	https://github.com/CryptoGuardOSS/cryptoguard
CuPerFuzzer [61]	2021	Not Specified	https://github.com/little-leiry/CuPerFuzzer
Deemon [62]	2017	GPLv3	https://github.com/tgianko/deemon
Delta [63]	2017	Not Specified	https://github.com/seungsoo-lee/DELTA
DFBC [64]	2021	Not Available	Not Available
Diane [65]	2021	Not Specified	https://github.com/ucsb-seclab/diane

Table 1. *Cont.*

Tool Name	*Year*	*License Type*	*Source Code Repository*
EBF [66]	2021	MIT License	https://github.com/fatimahkj/EBF
ELAID [67]	2020	Not Available	Not Available
ESASCF [68]	2023	Available upon request	Available upon request
ESRFuzzer [69]	2021	Not Available	Not Available
ESSecA [70]	2022	Not Specified	https://github.com/DanieleGranata94/SlaGenerator
Firmaster [71]	2018	Not Available	Not Available
FUGIO [72]	2022	Not Specified	https://github.com/WSP-LAB/FUGIO
FUSE [73]	2020	Not Specified	https://github.com/WSP-LAB/FUSE
Gail-PT [74]	2023	Not Specified	https://github.com/Shulong98/GAIL-PT/
GNPassGAN [75]	2022	MIT License	https://github.com/fangyiyu/GNPassGAN/
HARMer [76]	2020	MIT License	https://github.com/whistlebee/harmat
HILTI [77]	2014	Not Specified	https://github.com/rsmmr/hilti
IoTFuzzer [78]	2018	Not Specified	https://github.com/zyw-200/IOTFuzzer_Full
JCOMIX [79]	2019	Not Specified	https://github.com/SERG-Delft/JCOMIX
LAID [80]	2018	Not Available	Not Available
Link [81]	2022	Not Specified	https://github.com/WSP-LAB/Link
Lore [82]	2023	Not Available	Not Available
LTESniffer [83]	2023	Not Specified	https://github.com/SysSec-KAIST/LTESniffer
Mace [84]	2014	Not Available	Not Available
MAIT [85]	2021	Not Available	Not Available
MAL [86]	2018	Apache 2.0	https://github.com/mal-lang/malcompiler/
MaliceScript [87]	2018	Not Available	Not Available
Masat [88]	2015	Not Available	Not Available
Mirage [89]	2019	MIT License	https://github.com/RCayre/mirage
Mitch [90]	2019	Not Specified	https://github.com/alviser/mitch
MoScan [91]	2021	UPL 1.0	https://github.com/baigd/moscan
NAUTILUS [92]	2023	Apache 2.0	https://github.com/chenleji/nautilus
NAVEX [93]	2018	GPLv3	https://github.com/aalhuz/navex
NetCAT [94]	2020	Not Available	Not Available
NeuralNetworkCracking [95]	2016	Apache 2.0	https://github.com/cupslab/neural_network_cracking
No Name (CSRF) [96]	2020	Not Available	Not Available
No Name (TTCN-3) [97]	2018	Not Available	Not Available
NoCrack [98]	2015	MIT License	https://github.com/rchatterjee/nocrack
NodeXP [99]	2021	Not Specified	https://github.com/esmog/nodexp
ObjectMap [100]	2019	MIT License	https://github.com/georlav/objectmap
OMEN [101]	2015	MIT License	https://github.com/RUB-SysSec/OMEN
OSV [102]	2017	GPLv3	https://github.com/Emoform/OSV
Owfuzz [103]	2023	GPLv3	https://github.com/alipay/Owfuzz
PassGAN [104]	2019	MIT License	https://github.com/brannondorsey/PassGAN
PassGPT [105]	2023	CC BY-NC 4.0	https://github.com/javirandor/passgpt
PasswordCrackingTraining [106]	2022	MIT License	https://github.com/focardi/PasswordCrackingTraining
PenQuest [107]	2020	Proprietary	https://www.pen.quest/
PentestGPT [108]	2023	MIT License	https://github.com/GreyDGL/PentestGPT
PhpSAFE [109]	2015	GPLv2	https://github.com/JoseCarlosFonseca/phpSAFE
PJCT [110]	2015	Not Available	Not Available
Project Achilles [111]	2019	LGPLv3	https://github.com/secure-software-engineering/achilles-benchmark-depscanners
PURITY [112]	2015	Proprietary	Not Available
Pyciuti [113]	2023	Not Available	Not Available
RAT [114]	2022	Available upon request	Available upon request
Revealer [115]	2021	GPLv2	https://github.com/cuhk-seclab/Revealer
RiscyROP [116]	2022	Not Available	Not Available
Robin [117]	2020	Not Specified	https://github.com/olmps/Robin
ROSploit [118]	2019	MIT License	https://github.com/seanrivera/rosploit
RT-RCT [119]	2021	Not Available	Not Available
Scanner++ [120]	2023	Not Available	Not Available
SemanticGuesser [121]	2014	Not Specified	https://github.com/vialab/semantic-guesser
SerialDetector [122]	2021	Not Specified	https://github.com/yuske/SerialDetector
ShoVAT [123]	2016	Not Available	Not Available
Snout [124]	2019	Not Specified	https://github.com/nislab/snout/
SOA-Scanner [125]	2013	Not Available	Not Available
Spicy [126]	2016	MIT License	https://github.com/zeek/spicy/
SuperEye [127]	2019	Not Available	Not Available
SVED [128]	2016	Not Available	Not Available
TAMELESS [129]	2023	Not Specified	https://github.com/FulvioValenza/TAMELESS

Table 1. *Cont.*

Tool Name	*Year*	*License Type*	*Source Code Repository*
TChecker [130]	2022	Not Available	Not Available
TORPEDO [131]	2015	Not Available	Not Available
UE Security Reloaded [132]	2023	Not Available	Not Available
Untangle [133]	2023	Not Specified	https://github.com/untangle-tool/untangle
VAPE-BRIDGE [134]	2022	Not Available	Not Available
VERA [135]	2013	Not Available	Not Available
VUDDY [136]	2017	MIT License	https://github.com/squizz617/vuddy
Vulcan [137]	2013	Not Available	Not Available
VulCNN [138]	2022	Not Specified	https://github.com/CGCL-codes/VulCNN
VulDeePecker [139]	2018	Apache 2.0	https://github.com/CGCL-codes/VulDeePecker
Vulnet [140]	2019	Not Available	Not Available
Vulnsloit [141]	2020	Available upon request	Available upon request
VulPecker [142]	2016	Not Specified	https://github.com/vulpecker/Vulpecker
WAPTT [143]	2014	Not Available	Not Available
WebFuzz [144]	2021	GPLv3	https://github.com/ovanr/webFuzz
WebVIM [145]	2020	Not Available	Not Available

5.1. Process-Based Classification: PTES and Mitre ATT&CK

Table A1 shows the tools identified and classified for the different PTES phases. The tool distribution according to steps in the Ethical Hacking process is reported in Table 2 (a). The absence of tools in the *Pre-engagement Interactions* phase aligns with expectations, considering its non-technical nature, which typically involves scoping, planning, and agreement on the terms of engagement between the penetration tester and the client. This may potentially explain the lack of interest from the research community.

Table 2. Tool counts for process-based classification.

(**a**) Number of tools identified according to PTES phases

PTES Phase	*No.*
Vulnerability Analysis	80
Exploitation	39
Post-Exploitation	21
Intelligence Gathering	20
Threat Modelling	6
Reporting	4
Pre-engagement Interactions	0

(**b**) Number of tools identified according to Mitre ATTA&CK phases

Mitre ATTA&CK Phase	*No.*
Reconnaissance	84
Initial Access	48
Resource Development	21
Discovery	11
Execution	9
Credential Access	9
Collection	2
Impact	1
Persistence	0
Privilege Escalation	0
Defense Evasion	0
Lateral Movement	0
Command and Control	0
Exfiltration	0

The significant presence of tools in the *Vulnerability Analysis* phase (80 tools) reflects the importance of identifying and assessing vulnerabilities within target systems, which is essential for any security assessment activity. In particular, many scanners were developed.

Additionally, 20 tools possess *Intelligence Gathering* capabilities, primarily because this phase sometimes overlaps with vulnerability analysis as attackers interact with target systems.

Exploitation (39 tools) has a substantial number of tools designed to exploit identified vulnerabilities to gain unauthorised access to systems. Post-Exploitation has slightly fewer tools than other phases (21 tools). We found six tools for *Threat Modelling*. However, other researchers have developed some methodologies which are not implemented as tools that we discuss in Section 5.3.

The Mitre ATT&CK classification table (Table A2) shows tools associated with different stages of the attack process. The *Reconnaissance* (84 tools) and *Initial Access* (48 tools) stages exhibit a higher concentration of tools (Table 2 (b)), indicating the significance of these phases. This aligns with PTES findings, where most research effort seems to be put into vulnerability analysis. *Resource Development* (21) and *Discovery* (11) are also well represented. In contrast, stages such as *Persistence* and *Privilege Escalation* appear to have no tools directly associated with them, implying potential areas of development of novel research-informed tools. Further details on the classification, with sub-areas, are presented in Table A3.

Overall, researchers seem to have focused more on the technical aspects of the penetration testing process, and most of the tools have vulnerability analysis capability.

5.2. Knowledge-Based Classification: NCSC CyBOK and ACM CCS

Table A4 presents the classification of tools according to NCSC CyBOK. The distribution of tools across different *Knowledge Areas* (KAs) reflects the range of cybersecurity domains and disciplines covered by penetration testing activities. From the categorisation in Table 3, it is evident that certain areas, such as *Software and Platform Security* and *Networks Security*, are unsurprisingly more prominent, indicating areas of emphasis within cybersecurity practice. It should also be noted that while each category addresses specific aspects of cybersecurity, many tools may span multiple categories.

Table 3. Number of tools identified according to CyBOK, for KAs with at least one tool.

CyBOK Knowledge Area	*No.*
Software and Platform Security: Software Security	77
Software and Platform Security: Web and Mobile Security	38
Infrastructure Security: Network Security	26
Attacks and Defences: Adversarial Behaviours	9
Systems Security: Authentication, Authorisation and Accountability	9
Systems Security: Distributed Systems Security	3
Infrastructure Security: Applied Cryptography	2
Human, Organisational and Regulatory Aspects: Human Factors	2
Attacks and Defences: Malware and Attack Technology	1
Infrastructure Security: Physical Layer and Telecommunications Security	1
Human, Organisational and Regulatory Aspects: Privacy and Online Rights	1

Most of the tools are classified under *Software and Platform Security: Software Security: Detection of Vulnerabilities* (57 tools), which is a subcategory of *Software and Platform Security: Software Security*. The significant number of tools in this area reflects the recognition of software as a primary attack vector and demonstrates the research community's effort. Moreover, 38 tools are classified in the *Software and Platform Security: Web and Mobile Security* subcategory, highlighting the research work conducted to address the challenges posed by the development and deployment of web and mobile applications.

The 26 tools falling under *Infrastructure Security: Network Security* demonstrate the academic efforts in this area, ranging from network traffic monitoring and anomaly detection to implementing robust encryption protocols.

Nine tools classified under the *Attacks and Defences: Adversarial Behaviours* subcategory indicate research aimed at understanding and simulating the techniques used by attackers.

Among the various categories within the ACM CCS, EH tools in Table A5 predominantly fall into the *Security and Privacy* root category, specifically within the subcategories of *Systems Security*, *Software and Application Security*, and *Network Security*. In general, ACM CSS categories are too coarse to capture certain peculiarities of the tools. As CyBOK is specific to cybersecurity, it is more granular than ACM CSS for our purpose. We discuss the limitations in the classification in Section 5.4.

Software And Application Security: Vulnerability Management: Vulnerability Scanners has the highest number of tools at 73. This indicates the proactive measures the research community is taking to detect various issues, from misconfigurations and missing patches to software flaws and weak passwords. The categorisation of 35 tools under the *Software and Application Security: Web Applications Security* subcategory highlights the focus on developing specialised tools designed to test and secure web applications. These tools analyse web applications for vulnerabilities like SQL injection, cross-site scripting (XSS), and security misconfigurations.

Twenty-two tools were categorised under the *Network Security* domain and subdomains, focusing on protecting the data during its transmission across networks. These tools are essential for detecting intrusions, monitoring network traffic for suspicious activities, and implementing preventive measures such as firewalls and encryption. EH tools within this category enable security professionals to simulate attacks on the network to identify vulnerabilities and assess the network's resilience against cyber threats.

5.3. A Note on Threat Modelling Tools and Methodologies

The PTES classification shows that the number of tools identified for *Threat Modelling* is relatively small. However, the research field is somewhat active, but some contributions only propose new methodologies without implementing specific tools, so we did not include them in the classification. Some threat modelling methodologies discussed here cover different frameworks, each designed to improve security in cyber-physical systems (CPS), information technology, and critical infrastructure areas.

Ding et al.'s [146] framework integrates vulnerability assessment with reliability and threat analysis (both external and internal) within a unified model focused on critical infrastructures integrated with CPS. Similarly, Agadakos et al. [147] present a novel method for modelling cyber and physical interactions within IoT networks. The study emphasises the identification of unexpected event chains that could lead to security vulnerabilities. Additionally, Castiglione et al. [148] proposed a hazard-driven threat modelling methodology tailored for CPS, focusing on the interplay between security, reliability, and safety.

To highlight the critical role of human factors in information security, Evans et al. [149] introduce a methodology that systematically evaluates information security incidents caused by human error, adopting the HEART methodology from high-reliability sectors like aviation and energy. Also, David et al. [150] propose using timed automata to model socio-technical attacks, offering a method that incorporates time and cost into analysing socio-technical systems and attacks.

Furthermore, using formal methods for security analysis, Malik et al. [151] introduce an algorithm that transforms Attack Trees into Markov Decision Process models, aiming to address the limitations of scalability, state explosion, and manual interaction inherent in Attack Trees.

Collectively, these methodologies demonstrate a shift towards integrating diverse analytical tools and perspectives, from human factors to formal methods and system theory, to address the increasingly complex and interconnected nature of modern systems.

5.4. Limitations Surrounding the Classification of Tools

The four classification systems identified in this study were chosen due to their overall topic coverage and relevance to computing topics and concepts. When combined, the classifications can give a precise idea of what a tool can do and the cybersecurity field it falls under.

Although these four classifications are fit for purpose when considered within the scope and the goal of this survey and its goals, there are some challenges and potential limitations in classifying tools in this manner. The first issue is an inconsistency within the specificity of the tools.

For example, within *Mitre* [10], despite having hundreds of individual attack techniques and vectors, such as *Enterprise: Privilege Escalation: Access Token Manipulation: SID-History Injection* and *Enterprise: Defence Evasion: Hijack Execution Flow: Path Interception by PATH Environment Variable*, which are very specific vectors down to the operating system architecture within individual target systems, Mitre lumps the entire concept of compromising a web application under *Enterprise: Initial Access: Exploit Public-Facing Application*. There are other means of gaining specificity within this classification system, such as the *Reconnaissance: Vulnerability Scanning* field. All together, these give a more specific view of the tool. However, both fields do not offer the specificity of the *SID-History Injection* or *Path Interception by PATH Environment Variable* fields.

Another issue relates to tools that can be used for activities potentially unintended by the application designer. Deciding whether to include the ACM CSS *Security and Privacy: Network Security: Denial-of-Service Attacks* field when considering a web application fuzzer that could potentially crash the target web application (and making similar decisions throughout the classification of the many tools within this survey) posed a significant challenge, as opinions on whether to include the field may vary between researchers.

Another issue was found when trying to discern the exact scope of any given tool. There were many instances when, in the paper, a tool would present itself as having one function, for example, being capable of completing a specific task within the abstract and majority of the discussion in the associated paper, only to reveal that the tool itself is a *proof-of-concept*, with more limited capabilities than initially assumed.

To fully understand each tool's potential, an in-depth evaluation involving running the tools and testing their capabilities would be necessary. However, this is beyond the scope of this survey. Future work could focus on specific subject clusters to provide an in-depth comparison of the tools.

6. Evaluation

This section evaluates 100 research-informed Ethical Hacking tools developed within the past decade and included in this study. The discussion will focus on several key aspects: their licensing, release dates, availability of source code, the activity level of their development, whether the papers publishing them underwent peer review and their alignment with recognised cybersecurity frameworks.

6.1. Peer Review Analysis and Date of Publication

Of the 100 tools discussed in this study, 96% were disseminated through peer-reviewed journals and conferences (Table 4 (a)).

Table 4. Classification.

(**a**) Peer Reviewed		
Peer Reviewed	*No.*	*%*
Y	96	96.00%
N	4	4.00%
Total	*100*	*100.00%*

(**b**) Source Code Available		
Source Code Avail.	*No.*	*%*
Y	59	59.00%
N	41	41.00%
Total	*100*	*100.00%*

This indicates that the proposed tools have undergone rigorous validation, guaranteeing their effectiveness and reliability. This emphasis on peer-reviewed tools in our study reflects our commitment to ensuring readers have confidence in the credibility and utility of the tools presented. The remaining tools, which were not yet peer-reviewed at the time of our survey but are available as pre-print (e.g., [49,108]), are potentially under review or to be submitted in the near future.

Furthermore, the distribution of tool releases over the years, as illustrated in Figure 6, shows an increase in development activities in recent years, with 16 tools released in 2023, 14 in 2021, and 13 in 2019. This trend mirrors the evolution of cybersecurity threats and the response from the research community to address these problems.

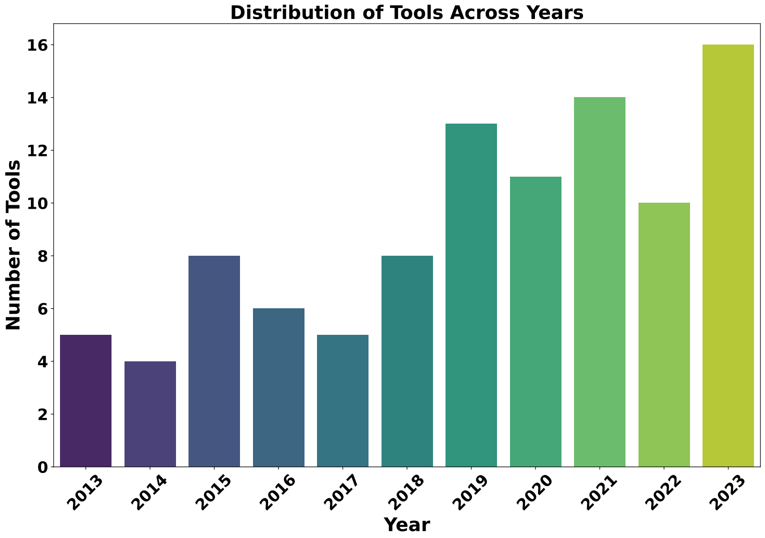

Figure 6. Distribution of tool released over the last decade.

6.2. Types of Licensing and Source Code Availability

This section delves into the licensing types, development activity, and source code availability for the tools discussed in this paper. The evaluation found that out of the 100 tools included in this study, 59 have their source code readily available on GitHub, while 41 are unavailable (Table 4 (b)).

Overall, the fact that source code is available for more than half of the tools (59 tools) demonstrates the cybersecurity researchers' dedication to openness and active community participation. However, the 41 inaccessible tools in this study highlight an ongoing debate: the need to balance transparency with security, privacy, and commercial interests.

The types of licenses for the tools available on GitHub vary widely (Figure 7), mostly open source licences, ranging from the *MIT License* (17 tools), to *GPLv3* (8 tools). However, for 23 tools, the licence is *Not Specified*. The lack of clear licensing information could be an oversight by developers regarding the importance of transparent communication of usage rights. This ambiguity may potentially hinder adoption and adaptability. In the absence of a license, default copyright laws apply, meaning that authors retain all rights to their source code and reproduction, distribution, or creation of derivative works is prohibited.

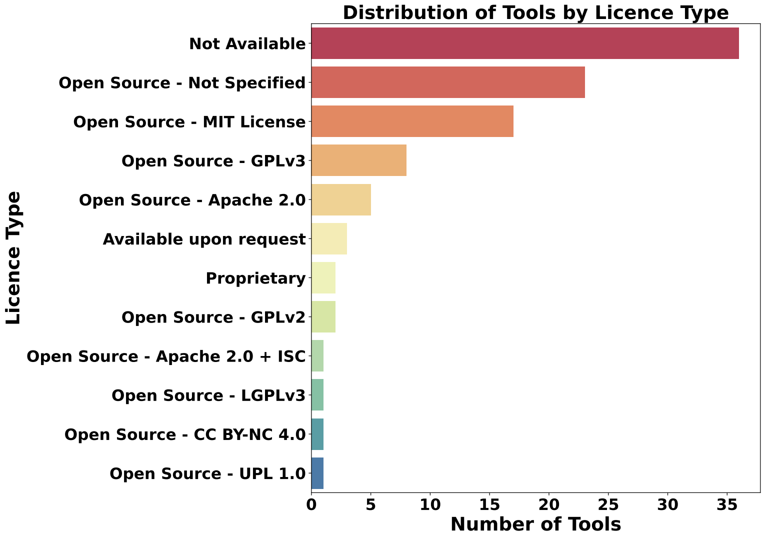

Figure 7. Distribution of tools by license type.

6.3. Tool Development and Maintenance

We examined the GitHub repositories of the 59 publicly available tools to understand specific features related to tool development and maintenance. Specifically, we collected data on the number of *commits* and the dates of the first and the last *commits*. We believe analysing *commit* activity in GitHub repositories provides insights into the development intensity and duration. However, we must consider that projects can move from one repository to another and that a private repository is used alongside a private one, and the public one is used only for dissemination purposes. Therefore, we can only attempt to capture some trends with this analysis, but we must be cautious about making statements regarding specific projects.

Table 5 (a) provides an overview of the distribution of project activity periods. We measured the difference in months between the first and last *commit* for the considered projects. The data shows that approximately one-third of the projects have a relatively short activity span of less than 3 months. This may indicate that the publication of the source code has likely been instrumental to the publication of the paper.

Comparing the year of publication of the paper and the date of the first *commit*, we can see that around 90% of the projects have been active sometime in the year before or after the publication. This is not surprising. However, fewer than 10% of the projects have been active 2 or more years before the publication, according to publicly available data.

Another parameter we considered is the number of *commits* (Table 5 (b)). The data highlights the diversity in project engagement and development intensity within the examined dataset. Around 30% of the projects have just up to 10 *commits*. As the *commit* ranges increase, the percentage of projects gradually decreases. Projects with a higher number of *commits* (500+) account for 13.33% of the total, indicating a smaller but notable proportion of projects with an extensive development history.

Table 6 represents the activity level as a percentage of the time between a project's release and the present. Additionally, *100–100* indicates continuous activity throughout the period, while *0–0* signifies no activity at all. For example, *26–50* means that the project was active for at least 26% and up to 50% of the considered time interval (9–18 months).

Table 5. Distribution of project activity and *commits*.

(**a**) Distribution of Period of Project Activity (last—first *commit*)		
Period (Months)	*No.*	*%*
0–3	19	32.20%
4–6	3	5.08%
6–12	6	10.17%
13–24	6	10.17%
25–36	10	16.95%
37–60	8	13.56%
61-inf	7	11.86%
Total	*59*	*100.00%*
(**b**) Distribution of Number of Project's *Commits*		
Commits Range	*No.*	*%*
1–10	17	28.81%
11–50	19	32.20%
51–100	4	6.78%
101–250	6	10.17%
251–500	5	8.47%
501-inf	8	13.56%
Total	*59*	*100.00%*

The data shows that one-third of the projects remained active within three years after their release. However, over half of these projects ceased activity after just 1.5 years. This trend sheds light on the development lifecycle of these projects, indicating a high initial engagement that tends to taper off relatively quickly for a significant number of projects.

Table 6. Distribution of project activity within 3 years after release.

% of Project Activity	***No***	***%***
0–0	5	8.47%
1–25	12	20.34%
26–50	15	25.42%
51–75	5	8.47%
76–99	2	3.39%
100–100	20	33.90%
Total	*59*	*100.00%*

6.4. Recommendations

To improve the dissemination and enhance the impact of research on the wider community of practitioners, we suggest that researchers should:

- Distribute the software as open source without exception and keep the software repository alive. Otherwise, it would be impossible for any dissemination within the practitioner community [152].
- Clearly specify the licence type and adopt standard FOSS licences [153], like GNU GPLv3, so that users may know precisely what they can do with the tools.
- Produce comprehensive documentation and tutorials on how to use the tools. Currently, this is partially conducted, but the existing documentation is primarily intended to support the peer-review process, as noted by Mirhosseini (2020) [154].
- Try to maintain the software by implementing bug fixes and improvements after publishing the paper. This is particularly challenging for academic projects as they

operate with limited availability of human resources and funding. Once the project ends or the paper is published, the interest of the researcher tends to move to new projects [152].
- Some tools may become obsolete for several reasons: incompatibility with more recent versions of other software (OSs, libraries, applications, etc.) or the vulnerability covered by the tool being patched. In those cases, the authors should update the documentation and clearly specify the requirements, scope, context and limitations of the tool.
- Try to implement their solutions in modular tools utilised by practitioners like Metasploit and Nmap. While this can be possible for certain solutions, in general, some tools are so different and innovative that they cannot fit into the API of existing tools.
- Consider that public dissemination mitigates the risk of weaponising tools by promoting a level-playing field approach.

Another question is: what can practitioners and industry do? We cannot expect many individual practitioners to engage directly with the research outputs except when driven by intellectual curiosity. However, the IT and cybersecurity industry should try to incentivise collaboration with academia. Industry and venture capitalists likely monitor academic research to understand the state-of-the-art and gain inspiration for new ideas. However, more effective engagement from the industry may help academic research to enhance its impact.

For example, the industry currently invests in bug bounty programs, providing monetary incentives to security researchers to identify and report vulnerabilities. This informs bug fixes and improves the quality of the product overall. However, this process is typically ex-post and not something an academic researcher would be directly involved in, except if finding a bug is a by-product of the research work. However, in many cases, academic researchers would likely engage in a responsible disclosure process.

Concretely, the industry could redirect some funding from bug bounty programs [155] to grant schemes supporting open source projects, for example, the Google Summer of Code, which could enable researchers to develop and enhance their tools. This is likely something that medium to large companies could be interested in. Still, it requires a shift in perspective beyond the immediate rewards and limited risks and commitments of current bug bounty programs.

Finally, an important aspect is that most research-informed tools are developed by small teams, sometimes even by a single individual, for non-profit reasons. Given the working conditions in many higher education institutions, especially in countries where the sector is very competitive and commercialised [156], it is often the case that, unless a research grant supports the project, the developers end up working significant hours during their own free time [157].

6.5. Related Work

Existing reviews of Ethical Hacking tools typically focus on industry practitioner tools, with occasional consideration of research-informed tools. Many popular practitioner tools included in these reviews (e.g., [1,14,158,159]) are recurrent: Nmap, Metasploit Framework, OpenVAS/GVM, Nessus, Burp Suite, OWASP ZAP, SQLMap, BeEF, Nikto, W3AF, and others.

Yaacoub et al. [14] survey and classify around 40 practitioner tools and OSs (e.g., Kali Linux and ParrotOS), focusing on challenges and issues associated with EH activities. The paper maps the tools and techniques for vulnerability assessment, network scanning tools, crimeware toolkits, etc., considering different attack types and application domains. Duque Anton et al. [1] include in their review around 25 popular practitioner tools, and their capabilities are evaluated using criteria such as active maintenance, licensing, commercial aspects (paid vs. free), and technical elements like programming language and interaction with other technology.

Moreover, Alhamed et al. [158] analyse around 20 mostly practitioner tools, with good coverage of network vulnerability and exploitation in particular. However, they consider existing research proposals for mitigating techniques. Additionally, Sarker et al. [159] reviewed penetration testing frameworks, processes, tools, and scoring methods, encompassing around 15 practitioner EH tools.

In some cases, authors restrict their focus to a specific domain. For example, Yaacoub et al. [5] provide good coverage of practitioner commercial and open-source solutions for EH in IoT, while Altulaihan et al. [7] review and compare industry practitioner tools for web application penetration testing. Similarly, Shahid et al. [160] provide a comparative analysis of commercial and open-source tools for Web Application Security with a focus on accuracy and precision. Alzahrani et al. [161] and Ravindran et al. [162] compare many EH tools, including both industry practitioner tools and a few research-informed tools for web vulnerability assessment and exploitation, e.g., XSS and SQL injection. Kowta et al. [163] analysed a variety of reconnaissance and information-gathering tools and techniques including Google Dorking, Shodan, Web Crawler, Recon-ng, Photon, Final Recon, and Sherlock. The tools are compared with criteria such as update frequency, languages used, and supported OSs, with some research-informed tools also included in the review.

In a few cases, authors systematically classify the tools according to some methodology or taxonomy. Duque Anton et al. [1] compared and classified practitioner tools, mapping them to the Mitre ATT&CK framework. Moreover, Zilberman et al. [164] provide a review of threat emulators while mapping to the Mitre ATT&CK matrix tactics.

Shanley et al. [8] review and compare several methodologies and frameworks, including PTES, Building Security in Maturity Model (BSIMM), Metasploit Framework (MSF), OWASP Testing Guide (OTG), Information Systems Security Assessment Framework (ISSAF), and the Open Source Security Testing Methodology Manual (OSSTMM). However, no tools are reviewed; therefore, no classification is attempted.

Our study significantly differs from previously published papers in both the number of tools covered and its exclusive focus on research-informed EH tools. By categorising the tools into process-based and knowledge-based classifications, we organise them according to specific phases, demonstrating where and when they are utilised in EH processes. While other reviews include classifications, the main contribution of our work is a more comprehensive and unique exploration. We surveyed 100 tools and classified them according to four different frameworks: PTES, Mitre ATT&CK, CyBOK, and ACM CCS. Additionally, we identify and analyze trends in developing, maintaining, and disseminating novel research-informed tools.

7. Conclusions and Future Work

Addressing emerging cyber security threats requires developing Ethical Hacking tools to identify vulnerabilities in networks, systems, and applications. While practitioners design most EH tools for immediate use in the industry, academic researchers have also significantly contributed to developing security tools. However, there is a noticeable gap in awareness among practitioners about academic contributions in this domain. This paper evaluates 100 research-informed tools, examining aspects such as licensing, release dates, source code availability, development activity, and peer review status. These tools are then aligned with established frameworks like PTES, the Mitre ATT&CK framework, CyBOK, and ACM CCS.

Key findings indicate that 96% of these tools originate from peer-reviewed research, with 59% having their source code readily accessible on GitHub. Activity analysis shows that 90% of projects were active around their publication year, yet activity dwindles significantly within 1.5 years post-release. Under the PTES framework classification, most tools are designed for vulnerability analysis, whereas threat modelling tools are relatively few. The CyBOK and ACM CCS classifications emphasise tools for detecting vulnerabilities, particularly under the *Software and Platform Security* and *Security And Privacy* categories, respectively. For the Mitre ATT&CK framework, most tools primarily focus on reconnaissance, highlighting the vital role of information gathering in identifying network and system details. Future directions involve experimental evaluations and comparisons of specific tools, integration of existing practitioner tools, and exploration of using large language models in penetration testing. This approach aims to bridge the gap between industry and academia, enhancing the development and effectiveness of Ethical Hacking tools.

Supplementary Materials: The supporting information, containing the review of the research-informed EH tools covered in this survey, can be downloaded at https://www.mdpi.com/article/10.3390/jcp4030021/s1. Refs [165–171] are cited in Supplementary Materials.

Author Contributions: Conceptualization, P.M.; methodology, P.M .; software, P.M.; validation, P.M. and L.H.; formal analysis, P.M.; investigation, P.M., L.G., L.H, C.O and M.M.; resources, P.M.; data curation, P.M and L.H.; writing—original draft preparation, P.M., L.G., L.H, C.O and M.M.; writing—review and editing, P.M., L.G., L.H, C.O and M.M.; visualization, P.M., L.G., and C.O.; project administration, P.M., L.G., and C.O.; funding acquisition, N/A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data was created, apart from the one presented in this paper and the supplementary material.

Acknowledgments: The authors express their gratitude to Zia Ush Shamszaman, Sachin Manakkil Jolly, and the Advanced Practice students of the MSc Cybersecurity course at Teesside University for engaging in constructive discussions and recommending certain tools for inclusion in this survey.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABAC	Attribute-Based Access Control
ACL	Access Control Lists
AE	Authenticated Encryption
APT	Advanced Persistent Threats
AP	Access Point
ATT&CK	(Mitre) Adversarial Tactics, Techniques, and Common Knowledge
C2	Command and Control
CBAC	Code-Based Access Control
CI	Continuous Integration
CLI	Command Line Interface
CPE	Common Platform Enumeration
CSRF	Cross Site Request Forgery
CSS	(ACM) Computing Classification System
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CyBOK	Cyber Security Body of Knowledge
DFBC	Digital Footprint and Breach Check
DFD	Data Flow Diagrams
DPI	Deep Packet Inspection
DRL	Deep Reinforcement Learning
DoS	Denial of Service
E2E	End-to-End
EH	Ethical Hacking
ETSI	European Telecommunications Standards Institute

FTP	File Transfer Protocol
GAIL	Generative Adversarial Imitation Learning
GAN	Generative Adversarial Network
GUI	Graphical User Interface
HARM	Hierarchical Attack Representation Model
ICS	Industrial Control Systems
IO2BO	Integer-Overflow-to-Buffer-Overflow
ISAAF	Information System Security Assessment Framework
IoMT	Internet of Medical Things
IoT	Internet of Things
LFA	Link Flooding Attacks
MAC	Message Authentication Code
MITM	Man-In-The-Middle
NFC	Near-Field Communications
NHS	National Health Service
NVD	National Vulnerability Database
OSINT	Open-Source INTelligence
OSPF	Open Shortest Path First
OSSTMM	Open-Source Security Testing Methodology Manuel
OS	Operating System
OWASP	Open Web Application Security Project
PCI DSS	Payment Card Industry Data Security Standard
POI	PHP Object Injection
PTES	Penetration Testing Execution Standard
RBAC	Role-Based Access Control
RDP	Remote Desktop Protocol
RL	Reinforcement Learning
SDN	Software Defined Networking
SDR	Software Defined Radio
SET	Social Engineering Toolkit
SOHO	Small Office and Home Office
SP	Special Publication
SQLIA	SQL Injection Attacks
SSJI	Server-Side Javascript Injection
TPM	Trusted Platform Module
TTP	Tactics, Techniques, and Procedures
UEFU	Unrestricted Executable File Upload
UFU	Unrestricted File Upload
VAPT	Vulnerability Assessment and Penetration Testing
VM	Virtual Machine
WCMS	Web Content Management Systems
XMLi	XML injection
XSS	Cross Site Scripting

Appendix A. Classification

Table A1. PTES classification.

PTES Phase	*Tools*
Pre-Engagement Interactions	
Intelligence Gathering	Bbuzz [51], DFBC [64], ESASCF [68], ESRFuzzer [69], Firmaster [71], IoTFuzzer [78], LTESniffer [83], Lore [82], MaliceScript [87], Owfuzz [103], Pyciuti [113], RT-RCT [119], SVED [128], Scanner++ [120], ShoVAT [123], SuperEye [127], TORPEDO [131], UE Security Reloaded [132], Vulcan [137], Vulnsloit [141]
Threat Modelling	Cairis [55], ESSecA [70], HARMer [76], MAL [86], PenQuest [107], TAMELESS [129]
Vulnerability Analysis	AIBugHunter [47], ARMONY [48], AVAIN [50], Autosploit [49], Bbuzz [51], Black Ostrich [52], Black Widow [53], Bleem [54], Censys [56], Chainsaw [57], Chucky [58], Commix [59], CryptoGuard [60], CuPerFuzzer [61], Deemon [62], Delta [63], Diane [65], EBF [66], ELAID [67], ESASCF [68], ESRFuzzer [69], FUGIO [72], FUSE [73], Firmaster [71], Gail-PT [74], HILTI [77], IoTFuzzer [78], JCOMIX [79], LAID [80], Link [81], Lore [82], Mace [84], MaliceScript [87], Masat [88], Mirage [89], Mitch [90], MoScan [91], NAUTILUS [92], NAVEX [93], No Name (CSRF) [96], No Name (TTCN-3) [97], NodeXP [99], OSV [102], ObjectMap [100], Owfuzz [103], PJCT [110], PURITY [112], PentestGPT [108], PhpSAFE [109], Project Achilles [111], Pyciuti [113], RAT [114], ROSploit [118], RT-RCT [119], Revealer [115], RiscyROP [116], Robin [117], SOA-Scanner [125], SVED [128], Scanner++ [120], SerialDetector [122], ShoVAT [123], Snout [124], Spicy [126], SuperEye [127], TChecker [130], TORPEDO [131], UE Security Reloaded [132], VAPE-BRIDGE [134], VERA [135], VUDDY [136], VulCNN [138], VulDeePecker [139], VulPecker [142], Vulcan [137], Vulnet [140], Vulnsloit [141], WAPTT [143], WebFuzz [144], WebVIM [145]
Exploitation	Chainsaw [57], Commix [59], ELAID [67], ESASCF [68], FUGIO [72], Firmaster [71], Gail-PT [74], LAID [80], LTESniffer [83], Lore [82], MAIT [85], Mace [84], MaliceScript [87], Mirage [89], Mitch [90], NAUTILUS [92], NAVEX [93], NetCAT [94], No Name (TTCN-3) [97], NodeXP [99], OSV [102], Owfuzz [103], PURITY [112], PentestGPT [108], Pyciuti [113], ROSploit [118], Revealer [115], RiscyROP [116], Robin [117], SOA-Scanner [125], SVED [128], SerialDetector [122], Snout [124], TORPEDO [131], Untangle [133], VAPE-BRIDGE [134], Vulnsloit [141], WAPTT [143], WebVIM [145]
Post-Exploitation	ADaMs [46], AVAIN [50], Delta [63], Diane [65], ESRFuzzer [69], GNPassGAN [75], HILTI [77], IoTFuzzer [78], Mirage [89], NeuralNetworkCracking [95], NoCrack [98], OMEN [101], OSV [102], PassGAN [104], PassGPT [105], PasswordCrackingTraining [106], Pyciuti [113], SemanticGuesser [121], Snout [124], Spicy [126], Untangle [133]
Reporting	ESASCF [68], Firmaster [71], No Name (TTCN-3) [97], Pyciuti [113]

Table A2. Mitre ATT&CK classification.

Mitre ATT&CK	***Tools***
Reconnaissance	AIBugHunter [47], ARMONY [48], AVAIN [50], AVAIN [50], Autosploit [49], Bbuzz [51], Black Ostrich [52], Black Widow [53], Bleem [54], Cairis [55], Censys [56], Chainsaw [57], Chucky [58], Commix [59], CryptoGuard [60], CuPerFuzzer [61], DFBC [64], Deemon [62], Delta [63], Delta [63], Diane [65], EBF [66], ELAID [67], ESASCF [68], ESRFuzzer [69], ESSecA [70], FUGIO [72], FUSE [73], Firmaster [71], Gail-PT [74], Gail-PT [74], HILTI [77], HILTI [77], IoTFuzzer [78], JCOMIX [79], LAID [80], LTESniffer [83], Link [81], Lore [82], Mace [84], MaliceScript [87], Masat [88], Mirage [89], Mirage [89], Mitch [90], MoScan [91], NAUTILUS [92], NAVEX [93], No Name (CSRF) [96], No Name (TTCN-3) [97], No Name (TTCN-3) [97], NodeXP [99], OSV [102], ObjectMap [100], Owfuzz [103], PURITY [112], PenQuest [107], PentestGPT [108], PhpSAFE [109], Pyciuti [113], RAT [114], ROSploit [118], RT-RCT [119], RT-RCT [119], Revealer [115], RiscyROP [116], Robin [117], SOA-Scanner [125], SVED [128], Scanner++ [120], SerialDetector [122], ShoVAT [123], ShoVAT [123], Snout [124], Snout [124], Spicy [126], Spicy [126], SuperEye [127], TAMELESS [129], TChecker [130], TORPEDO [131], UE Security Reloaded [132], VAPE-BRIDGE [134], VERA [135], VUDDY [136], VulCNN [138], VulDeePecker [139], VulPecker [142], Vulcan [137], Vulnet [140], Vulnsloit [141], WAPTT [143], WebFuzz [144], WebVIM [145]
Resource Development	AIBugHunter [47], Autosploit [49], Chucky [58], CuPerFuzzer [61], ELAID [67], ESASCF [68], HARMer [76], HILTI [77], LAID [80], MAIT [85], MAL [86], Owfuzz [103], PJCT [110], PJCT [110], Project Achilles [111], Revealer [115], Spicy [126], UE Security Reloaded [132], Untangle [133], VUDDY [136], VulCNN [138], VulPecker [142]
Initial Access	Black Ostrich [52], Black Widow [53], Censys [56], Chainsaw [57], Commix [59], Deemon [62], ESASCF [68], ESSecA [70], FUGIO [72], FUSE [73], Firmaster [71], Gail-PT [74], JCOMIX [79], Link [81], Lore [82], MAL [86], Mace [84], MaliceScript [87], Masat [88], Mitch [90], NAUTILUS [92], NAVEX [93], NetCAT [94], No Name (CSRF) [96], NodeXP [99], OSV [102], ObjectMap [100], PURITY [112], PentestGPT [108], PhpSAFE [109], Pyciuti [113], RAT [114], Revealer [115], Robin [117], SOA-Scanner [125], SVED [128], Scanner++ [120], SerialDetector [122], ShoVAT [123], TChecker [130], TORPEDO [131], VAPE-BRIDGE [134], VERA [135], Vulcan [137], Vulnet [140], WAPTT [143], WebFuzz [144], WebVIM [145]
Execution	Bbuzz [51], ESASCF [68], Lore [82], Mirage [89], PentestGPT [108], ROSploit [118], RiscyROP [116], SVED [128], Vulnsloit [141]
Persistence	
Privilege Escalation	
Defense Evasion	
Credential Access	ADaMs [46], Firmaster [71], GNPassGAN [75], LTESniffer [83], NeuralNetworkCracking [95], NoCrack [98], OMEN [101], PassGAN [104], PassGPT [105], PasswordCrackingTraining [106], SemanticGuesser [121]
Discovery	AVAIN [50], Cairis [55], Firmaster [71], HILTI [77], Masat [88], PenQuest [107], RT-RCT [119], Snout [124], Spicy [126], TAMELESS [129], Vulcan [137]
Lateral Movement	
Collection	HILTI [77], Spicy [126]
Command And Control	
Exfiltration	
Impact	Revealer [115], TORPEDO [131]

Table A3. Mitre ATT&CK classification (details).

Mitre ATT&CK	*Tools*
Collection: Adversary-In-The-Middle	HILTI [77], Spicy [126]
Credential Access: Brute Force: Password Cracking	GNPassGAN [75], PassGAN [104], PasswordCrackingTraining [106]
Discovery: Cloud Infrastructure Discovery	MASAT [88], VULCAN [137]
Discovery: Network Service Discovery	AVAIN [50], Firmaster [71], HILTI [77], RT-RCT [119], Snout [124], Spicy [126]
Enterprise: Credential Access: Brute Force	Firmaster [71]
Enterprise: Credential Access: Network Sniffing	LTESniffer [83]
Enterprise: Impact: Service Stop	TORPEDO [131]
Enterprise: Initial Access: External Remote Services	NetCAT [94]
Execution	Bbuzz [51], Lore [82], Mirage [89], PentestGPT [108], ROSploit [118], SVED [128], Vulnsloit [141]
Execution: Inter-Process Communication	RiscyROP [116]
Gather Victim Network Information	Lore [82], PentestGPT [108], SVED [128]
Impact: Endpoint Denial Of Service	Revealer [115]
Initial Access	Gail-PT [74], Lore [82], OSV [102], PentestGPT [108], SVED [128]
Initial Access: Exploit Public-Facing Application	Commix [59], JCOMIX [79], Mitch [90], No Name (CSRF) [96], PURITY [112], Puciuty [113], Robin [117], Vulnet [140], WebVIM [145], ZGrab [56]
Initial Access: Exploit Public-Facing Application	WAPTT [143]
Initial Access: Exploit Public-Facing Application	Black Ostrich [52], Black Widow [53], Chainsaw [57], Deemon [62], FUGIO [72], FUSE [73], Firmaster [71], Link [81], MASAT [88], Mace [84], MaliceScript [87], NAUTILUS [92], NAVEX [93], NodeXP [99], ObjectMap [100], PhpSAFE [109], Revealer [115], SOA-Scanner [125], Scanner++ [120], SerialDetector [122], ShoVAT [123], TChecker [130], TORPEDO [131], VAPE-BRIDGE [134], VERA [135], VULCAN [137], WebFuzz [144]
Reconnaissance: Active Scanning	LTESniffer [83], TORPEDO [131]
Reconnaissance: Active Scanning: Vulnerability Scanning	NodeXP [99]
Reconnaissance: Active Scanning: Vulnerability Scanning	AIBugHunter [47], ARMONY [48], AVAIN [50], Autosploit [49], Bbuzz [51], Black Ostrich [52], Black Widow [53], Chainsaw [57], Chucky [58], Commix [59], CryptoGuard [60], CuPerFuzzer [61], DELTA [63], DIANE [65], Deemon [62], EBF [66], ELAID [67], ESRFuzzer [69], FUGIO [72], FUSE [73], Firmaster [71], Gail-PT [74], HILTI [77], IoTFuzzer [78], JCOMIX [79], LAID [80], Link [81], Lore [82], MASAT [88], Mace [84], MaliceScript [87], Mirage [89], Mitch [90], NAUTILUS [92], NAVEX [93], No Name (CSRF) [96], No Name (TTCN-3) [97], OSV [102], ObjectMap [100], Owfuzz [103], PURITY [112], PentestGPT [108], PhpSAFE [109], Puciuty [113], ROSploit [118], RT-RCT [119], Revealer [115], RiscyROP [116], Robin [117], SOA-Scanner [125], SVED [128], Scanner++ [120], SerialDetector [122], ShoVAT [123], Snout [124], Spicy [126], SuperEye [127], TChecker [130], UE Security Reloaded [132], VAPE-BRIDGE [134], VERA [135], VUDDY [136], VULCAN [137], VulCNN [138], VulDeePecker [139], VulPecker [142], Vulnet [140], Vulnsloit [141], WAPTT [143], WebFuzz [144], WebVIM [145], ZGrab [56]
Reconnaissance: Gather Victim Identity Information	DFBC [64]
Reconnaissance: Gather Victim Network Information	AVAIN [50], DELTA [63], Gail-PT [74], HILTI [77], MaliceScript [87], Mirage [89], Puciuty [113], RT-RCT [119], ShoVAT [123], Snout [124], Spicy [126]
Reconnaissance: Gather Victim Network Information: Network Topology	No Name (TTCN-3) [97]
Reconnaissance: Resource Development	HARMer [76]
Resource Development: Develop Capabilities	HILTI [77], PICT [110], Spicy [126]
Resource Development: Develop Capabilities: Exploits	ELAID [67], LAID [80], Owfuzz [103], Project Achilles [111], UE Security Reloaded [132], VulCNN [138]
Resource Development: Develop Capabilities: Malware	MAIT [85]
Resource Development: Obtain Capabilities: Exploits	AIBugHunter [47], Autosploit [49], Chucky [58], CuPerFuzzer [61], PICT [110], Revealer [115], VUDDY [136], VulPecker [142]

Table A4. CyBOK classification.

CyBOK	***Tools***
Attacks & Defences: Adversarial Behaviours	Cairis [55], ESASCF [68], ESSecA [70], HARMer [76], Lore [82], MAL [86], PenQuest [107], PenQuest [107], SVED [128], TAMELESS [129]
Attacks & Defences: Malware & Attack Technology: Malware Analysis: Analysis Techniques: Static Analysis/Dynamic Analysis	MAIT [85]
Human, Organisational & Regulatory Aspects: Human Factors	ESSecA [70], TAMELESS [129]
Human, Organisational & Regulatory Aspects: Privacy & Online Rights: Privacy Engineering: Privacy Evaluation	DFBC [64]
Infrastructure Security: Applied Cryptography: Cryptographic Implementation: Api Design For Cryptographic Libraries	CryptoGuard [60]
Infrastructure Security: Applied Cryptography: Cryptographic Implementation: Cryptographic Libraries	Firmaster [71]
Infrastructure Security: Cyber Physical Systems	ESSecA [70], TAMELESS [129]
Infrastructure Security: Network Security	AVAIN [50], Cairis [55], Delta [63], ESASCF [68], Gail-PT [74], HARMer [76], HILTI [77], Lore [82], Masat [88], NetCAT [94], SVED [128], Spicy [126]
Infrastructure Security: Network Security: Network Protocols And Their Security	OSV [102], SuperEye [127], Vulnsloit [141]
Infrastructure Security: Network Security: Network Protocols And Their Security: Security At The Internet Layer	Bbuzz [51]
Infrastructure Security: Network Security: Network Protocols And Their Security: Security At The Internet Layer: Ipv6 Security	No Name (TTCN-3) [97]
Infrastructure Security: Network Security: Networking Applications	Vulcan [137]
Infrastructure Security: Network Security: Networking Applications: Local Area Networks	ESRFuzzer [69], Firmaster [71], HILTI [77], No Name (TTCN-3) [97], Pyciuti [113], Spicy [126]
Infrastructure Security: Network Security: Networking Applications: Wireless Networks	ESRFuzzer [69], Firmaster [71], LTESniffer [83], Owfuzz [103], RT-RCT [119], Snout [124], UE Security Reloaded [132]
Infrastructure Security: Network Security: Other Network Security Topics: Cloud And Data Center Security	Masat [88], Vulcan [137]
Infrastructure Security: Network Security: Software-Defined Networking And Network Function Virtualization	Delta [63]
Infrastructure Security: Physical Layer & Telecommunications Security: Identification: Attacks On Physical Layer Identification	Snout [124]
Operating Systems & Virtualization Security: Operating System Hardening	ROSploit [118]
Physical Layer & Telecommunications Security: Physical Layer Security Of Selected Communication Technologies: Cellular Networks: 4G (Lte)	LTESniffer [83]
Physical Layer & Telecommunications Security: Physical Layer Security Of Selected Communication Technologies: Cellular Networks: 5G	UE Security Reloaded [132]
Resource Development: Develop Capabilities: Exploits	ESASCF [68]
Software And Platform Security: Software Security: Categories Of Vulnerabilities: Memory Management Vulnerabilities	ARMONY [48], ELAID [67], IoTFuzzer [78], LAID [80], WAPTT [143]
Software And Platform Security: Software Security: Detection Of Vulnerabilities	ARMONY [48], AVAIN [50], Autosploit [49], Bbuzz [51], Black Ostrich [52], Black Widow [53], Cairis [55], Censys [56], Chainsaw [57], Commix [59], CryptoGuard [60], Deemon [62], EBF [66], ESASCF [68], FUGIO [72], FUSE [73], Firmaster [71], HILTI [77], JCOMIX [79], Link [81], Mace [84], MaliceScript [87], Mirage [89], Mitch [90], MoScan [91], NAUTILUS [92], NAVEX [93], No Name (CSRF) [96], No Name (TTCN-3) [97], NodeXP [99], OSV [102], ObjectMap [100], Owfuzz [103], PJCT [110], PURITY [112], PentestGPT [108], Project Achilles [111], Pyciuti [113], RAT [114], ROSploit [118], RT-RCT [119], Revealer [115], SOA-Scanner [125], Scanner++ [120], SerialDetector [122], ShoVAT [123], Snout [124], Spicy [126], SuperEye [127], TChecker [130], TORPEDO [131], UE Security Reloaded [132], VAPE-BRIDGE [134], VERA [135], VulDeePecker [139], Vulcan [137], Vulnet [140], Vulnsloit [141], WAPTT [143], WebFuzz [144], WebVIM [145]
Software And Platform Security: Software Security: Detection Of Vulnerabilities: Dynamic Detection	Bbuzz [51], Black Ostrich [52], CuPerFuzzer [61], Diane [65], EBF [66], Project Achilles [111]

Table A4. *Cont.*

CyBOK	***Tools***
Software And Platform Security: Software Security: Detection Of Vulnerabilities: Dynamic Detection: Black-Box Fuzzing	Bleem [54], Delta [63], IoTFuzzer [78], Owfuzz [103]
Software And Platform Security: Software Security: Detection Of Vulnerabilities: Dynamic Detection: Generating Relevant Executions: Dynamic Symbolic Execution	RiscyROP [116]
Software And Platform Security: Software Security: Detection Of Vulnerabilities: Static Detection	AIBugHunter [47], Chucky [58], ELAID [67], LAID [80], PhpSAFE [109], Untangle [133], VUDDY [136], VulCNN [138], VulPecker [142]
Software And Platform Security: Software Security: Dynamic Detection	WebFuzz [144]
Software And Platform Security: Software Security: Side-Channel Vulnerabilities	NetCAT [94]
Software And Platform Security: Web & Mobile Security	Black Ostrich [52], EBF [66], Mace [84], MoScan [91], NAUTILUS [92], NAVEX [93], RAT [114], Revealer [115], Robin [117], Scanner++ [120], ShoVAT [123], VAPE-BRIDGE [134]
Software And Platform Security: Web & Mobile Security: Client Side Vulnerabilities And Mitigations	MaliceScript [87]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations	Censys [56], PURITY [112], Pyciuti [113], Robin [117], SOA-Scanner [125], TORPEDO [131], VERA [135], Vulnet [140]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities	Commix [59], FUGIO [72], ObjectMap [100], SerialDetector [122]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities: Command Injection	JCOMIX [79], NodeXP [99]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities: Cross-Site Request Forgery (Csrf)	Deemon [62], Mitch [90], No Name (CSRF) [96]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities: Cross-Site Scripting (Xss)	Black Widow [53], Chainsaw [57], PhpSAFE [109], TChecker [130], WAPTT [143], WebFuzz [144]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities: Cross-Site Scripting (Xss): Reflected Xss	Link [81]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities: Sql-Injection	Chainsaw [57], PhpSAFE [109], TChecker [130], WAPTT [143], WebVIM [145]
Software And Platform Security: Web & Mobile Security: Server Side Vulnerabilities And Mitigations: Injection Vulnerabilities: User Uploaded Files	FUSE [73]
Systems Security: Authentication, Authorisation & Accountability: Authentication: Passwords	ADaMs [46], GNPassGAN [75], NeuralNetworkCracking [95], NoCrack [98], OMEN [101], PassGAN [104], PassGPT [105], PasswordCrackingTraining [106], SemanticGuesser [121]
Systems Security: Distributed Systems Security	Cairis [55], MAL [86], PenQuest [107]

Table A5. ACM CCS classification.

ACM CCS	*Tools*
Hardware: Emerging Technologies: Analysis And Design Of Emerging Devices And Systems: Emerging Architectures	AVAIN [50], Diane [65], EBF [66], IoTFuzzer [78], Mirage [89], ROSploit [118], RT-RCT [119], Snout [124]
Human-Centered Computing: Human Computer Interaction (Hci): Interactive Systems And Tools	TAMELESS [129]
Networks: Network Components: Intermediate Nodes: Routers	ESRFuzzer [69]
Networks: Network Protocols: Network Layer Protocols: Routing Protocols	No Name (TTCN-3) [97], OSV [102]
Security And Privacy: Cryptography	EBF [66]
Security And Privacy: Human And Societal Aspects Of Security And Privacy	DFBC [64]
Security And Privacy: Intrusion/Anomaly Detection And Malware Mitigation: Malware And Its Mitigation	MAIT [85]
Security And Privacy: Network Security	AVAIN [50], Bbuzz [51], Censys [56], NetCAT [94], No Name (TTCN-3) [97], OSV [102], Pyciuti [113], RT-RCT [119], SuperEye [127], Vulcan [137], Vulnsloit [141]
Security And Privacy: Network Security: Mobile And Wireless Security	ESRFuzzer [69], Firmaster [71], LTESniffer [83], Owfuzz [103], Scanner++ [120], Snout [124], UE Security Reloaded [132]
Security And Privacy: Network Security: Security Protocols	HILTI [77], No Name (TTCN-3) [97], Spicy [126]
Security And Privacy: Network Security: Web Protocol Security	Bbuzz [51]
Security And Privacy: Security Services: Authorisation	ADaMs [46], GNPassGAN [75], NeuralNetworkCracking [95], NoCrack [98], OMEN [101], PassGAN [104], PassGPT [105], PasswordCrackingTraining [106], SemanticGuesser [121]
Security And Privacy: Software And Application Security	CryptoGuard [60], VulDeePecker [139]
Security And Privacy: Software And Application Security: Domain-Specific Security And Privacy Architectures	ESSecA [70], MAL [86], PenQuest [107]
Security And Privacy: Software And Application Security: Software Reverse Engineering	RiscyROP [116], VulCNN [138]
Security And Privacy: Software And Application Security: Software Security Engineering	AIBugHunter [47], Chucky [58], CuPerFuzzer [61], ELAID [67], LAID [80], PJCT [110], Project Achilles [111], Untangle [133], VUDDY [136], VulPecker [142]
Security And Privacy: Software And Application Security: Web Applications Security	Black Ostrich [52], Black Widow [53], Censys [56], Chainsaw [57], Commix [59], Deemon [62], FUGIO [72], FUSE [73], JCOMIX [79], Link [81], Mace [84], MaliceScript [87], Mitch [90], MoScan [91], NAUTILUS [92], NAVEX [93], No Name (CSRF) [96], NodeXP [99], ObjectMap [100], PURITY [112], PhpSAFE [109], Pyciuti [113], RAT [114], Robin [117], SOA-Scanner [125], SerialDetector [122], ShoVAT [123], TChecker [130], TORPEDO [131], VAPE-BRIDGE [134], VERA [135], Vulnet [140], WAPTT [143], WebFuzz [144], WebVIM [145]
Security And Privacy: Systems Security: Denial Of Service Attacks	Revealer [115]
Security And Privacy: Systems Security: Distributed Systems Security	MAL [86], PenQuest [107]
Security And Privacy: Systems Security: Vulnerability Management: Penetration Testing	Cairis [55], Diane [65], ESASCF [68], ESSecA [70], Gail-PT [74], HARMer [76], Lore [82], MAL [86], Mirage [89], PenQuest [107], PentestGPT [108], Pyciuti [113], SVED [128], TAMELESS [129]
Security And Privacy: Systems Security: Vulnerability Management: Vulnerability Scanners	AIBugHunter [47], ARMONY [48], AVAIN [50], Autosploit [49], Black Ostrich [52], Black Widow [53], Bleem [54], Censys [56], Chainsaw [57], Chucky [58], Commix [59], CryptoGuard [60], CuPerFuzzer [61], Deemon [62], Delta [63], EBF [66], ELAID [67], ESSecA [70], FUGIO [72], FUSE [73], Firmaster [71], HILTI [77], IoTFuzzer [78], JCOMIX [79], LAID [80], Link [81], Mace [84], MaliceScript [87], Masat [88], Mitch [90], MoScan [91], NAUTILUS [92], NAVEX [93], No Name (CSRF) [96], No Name (TTCN-3) [97], NodeXP [99], OSV [102], ObjectMap [100], Owfuzz [103], PJCT [110], PURITY [112], PhpSAFE [109], Project Achilles [111], Pyciuti [113], RAT [114], ROSploit [118], RT-RCT [119], Revealer [115], RiscyROP [116], Robin [117], SOA-Scanner [125], Scanner++ [120], SerialDetector [122], ShoVAT [123], Snout [124], Spicy [126], SuperEye [127], TChecker [130], TORPEDO [131], UE Security Reloaded [132], Untangle [133], VAPE-BRIDGE [134], VERA [135], VUDDY [136], VulCNN [138], VulDeePecker [139], VulPecker [142], Vulcan [137], Vulnet [140], Vulnsloit [141], WAPTT [143], WebFuzz [144], WebVIM [145]

References

1. Duque Anton, S.D.; Fraunholz, D.; Schneider, D. Investigating the Ecosystem of Offensive Information Security Tools. *arXiv* **2020**, arXiv:2012.08811. [CrossRef]
2. Leal, M.M.; Musgrave, P. Backwards from zero: How the U.S. public evaluates the use of zero-day vulnerabilities in cybersecurity. *Contemp. Secur. Policy* **2023**, *44*, 437–461. [CrossRef]
3. Valenza, A.; Costa, G.; Armando, A. Never Trust Your Victim: Weaponizing Vulnerabilities in Security Scanners. In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID), Online, 14–16 October 2020; pp. 17–29. [CrossRef]
4. Denis, M.; Zena, C.; Hayajneh, T. Penetration testing: Concepts, attack methods, and defense strategies. In Proceedings of the 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 29 April 2016; pp. 1–6. [CrossRef]
5. Yaacoub, J.P.A.; Noura, H.N.; Salman, O.; Chehab, A. Ethical hacking for IoT: Security issues, challenges, solutions and recommendations. *Internet Things -Cyber-Phys. Syst.* **2023**, *3*, 280–308. [CrossRef]
6. Aarya, P.S.; Rajan, A.; Sachin, K.P.S.; Gopi, R.; Sreenu, G. Web Scanning: Existing Techniques and Future. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; pp. 123–128. [CrossRef]
7. Altulaihan, E.A.; Alismail, A.; Frikha, M. A Survey on Web Application Penetration Testing. *Electronics* **2023**, *12*, 1229. [CrossRef]
8. Shanley, A.; Johnstone, M.N. *Selection of Penetration Testing Methodologies: A Comparison and Evaluation;* SRI Security Research Institute: Menlo Park, CA, USA, 2015. [CrossRef]
9. PTES Working Group. *The Penetration Testing Execution Standard*; PTES Working Group: London, UK, 2011.
10. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. *Mitre att&ck: Design and Philosophy*; Technical Report; The MITRE Corporation: McLean, VA, USA, 2018.
11. Rashid, A.; Chivers, H.; Danezis, G.; Lupu, E.; Martin, A. (Eds.) *Cyber Security Body of Knowledge*, version 1.1.0; The National Cyber Security Centre: London, UK, 2021.
12. Rous, B. Major update to ACM's Computing Classification System. *Commun. ACM* **2012**, *55*, 12. [CrossRef]
13. Bishop, M. *Computer Security: Art and Science*; Pearson Education: London, UK, 2018.
14. Yaacoub, J.A.; Noura, H.N.; Salman, O.; Chehab, A. A Survey on Ethical Hacking: Issues and Challenges. *arXiv* **2021**, arXiv:2103.15072. [CrossRef]
15. Walker, M. *CEH Certified Ethical Hacker All-in-One Exam Guide*, 5th ed.; McGraw Hill LLC: New York, NY, USA, 2021.
16. Hald, S.L.N.; Pedersen, J.M. An updated taxonomy for characterizing hackers according to their threat properties. In Proceedings of the 2012 14th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Republic of Korea, 19–22 February 2012; pp. 81–86.
17. Oliver, D.; Randolph, A.B. Hacker Definitions in Information Systems Research. *J. Comput. Inf. Syst.* **2022**, *62*, 397–409. [CrossRef]
18. Aljaidi, M.; Alsarhan, A.; Samara, G.; Alazaidah, R.; Almatarneh, S.; Khalid, M.; Al-Gumaei, Y.A. NHS WannaCry Ransomware Attack: Technical Explanation of The Vulnerability, Exploitation, and Countermeasures. In Proceedings of the 2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI), Zarqa, Jordan, 6–8 December 2022; pp. 1–6. [CrossRef]
19. Qin, M.; Mogos, G. Cyber-attacks on SWIFT Systems of financial institutions. In Proceedings of the 5th International Conference on Computer Science and Software Engineering, Guilin, China, 21–23 October 2022; pp. 596–599. [CrossRef]
20. Fidler, D.P. The US election hacks, cybersecurity, and international law. *Am. J. Int. Law* **2016**, *110*, 337–342.
21. Steffens, T. *Attribution of Advanced Persistent Threats—How to Identify the Actors Behind Cyber-Espionage*; Springer: Berlin/Heidelberg, Germany, 2020. [CrossRef]
22. Liu, Z.; Chen, C.; Zhang, L.Y.; Gao, S. Working Mechanism of Eternalblue and Its Application in Ransomworm. In *Lecture Notes in Computer Science*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 178–191. [CrossRef]
23. Concil of Europe. *Convention on Cybercrime*; European Treaty Series—No. 185; Council of Europe: Strasbourg, France, 2001.
24. *Directive 2013/40/EU of the European Parliament and of the Council of 12 August 2013 on Attacks against Information Systems and Replacing Council Framework Decision 2005/222/JHA*; Council of the European Union; European Parliament: Brussels, Belgium, 2013.
25. *Computer Misuse Act 1990*; United Kingdom Parliament: London, UK, 1990.
26. Gehring, V.V. *The Internet in Public Life*; Rowman & Littlefield: Lanham, ML, USA, 2004.
27. Jaquet-Chiffelle, D.O.; Loi, M. Ethical and unethical hacking. *Ethics Cybersecur.* **2020**, *21*, 179–204. [CrossRef]
28. MITRE. *Common Vulnerabilities and Exposures*; The MITRE Corporation: McLean, VA, USA, 2024.
29. MITRE. *Common Weakness Enumeration*; The MITRE Corporation: McLean, VA, USA, 2024.
30. *Payment Card Industry Data Security Standard: Requirements and Testing Procedures*, v4.0; PCI Security Standards Council: Wakefield, MA, USA, 2022.
31. *PCI Data Security Standard—Penetration Testing Guidance*; PCI Security Standards Council: Wakefield, MA, USA, 2017.
32. Rathore, B.; Brunner, M.; Dilaj, M.; Herrera, O.; Brunati, P.; Subramaniam, R.; Raman, S.; Chavan, U. Information systems security assessment framework (issaf). *Draft 0.2 B* **2006**, *1*, 2006.
33. Herzog, P. *The Open Source Security Testing Methodology Manual*; ISECOM: Barcelona, Spain, 2010.

34. Scarfone, K.A.; Souppaya, M.P.; Cody, A.; Orebaugh, A.D. *SP 800-115*; Technical guide to information security testing and assessment. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2008.
35. Cerullo, F.E. OWASP TOP 10 2009. In *Web Application Security*; Serrão, C., Aguilera Díaz, V., Cerullo, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; p. 19. [CrossRef]
36. Meucci, M.; Muller, A. *OWASP Testing Guide*, v4.0; OWASP: Wakefield, MA, USA, 2014.
37. Moher, D.; Shamseer, L.; Clarke, M.; Ghersi, D.; Liberati, A.; Petticrew, M.; Shekelle, P.; Stewart, L.A.; Group, P.P. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Syst. Rev.* **2015**, *4*, 1. [CrossRef] [PubMed]
38. Astrida, D.N.; Saputra, A.R.; Assaufi, A.I. Analysis and Evaluation of Wireless Network Security with the Penetration Testing Execution Standard (PTES). *Sink. J. Dan Penelit. Tek. Inform.* **2022**, *7*, 147–154. [CrossRef]
39. Rajesh, P.; Alam, M.; Tahernezhadi, M.; Monika, A.; Chanakya, G. Analysis of cyber threat detection and emulation using mitre attack framework. In Proceedings of the 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), San Antonio, TX, USA, 5–7 September 2022; pp. 4–12. [CrossRef]
40. Georgiadou, A.; Mouzakitis, S.; Askounis, D. Assessing mitre att&ck risk using a cyber-security culture framework. *Sensors* **2021**, *21*, 3267. [CrossRef] [PubMed]
41. Messinis, S.; Temenos, N.; Protonotarios, N.E.; Rallis, I.; Kalogeras, D.; Doulamis, N. Enhancing Internet of Medical Things security with artificial intelligence: A comprehensive review. *Comput. Biol. Med.* **2024**, 108036. [CrossRef] [PubMed]
42. Alexander, O.; Belisle, M.; Steele, J. *MITRE ATT&CK for Industrial Control Systems: Design and Philosophy*; The MITRE Corporation: Bedford, MA, USA, 2020; Volume 29.
43. Williams, L. *The Cyber Security Body of Knowledge*, v1.1.0; Chapter Secure Software Lifecycle; KA Version 1.0.2; University of Bristol: Bristol, UK, 2021.
44. Nautiyal, L.; Rashid, A.; Hallett, J.; Shreeve, B. *The UK's Cyber Security Degree Certification Programme: A CyBOK Case Study*; Technical Report; University of Essex: Colchester, UK, 2020.
45. Attwood, S.; Williams, A. Exploring the UK Cyber Skills Gap through a mapping of active job listings to the Cyber Security Body of Knowledge (CyBOK). In Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, Oulu, Finland, 14–16 June 2023; pp. 273–278. [CrossRef]
46. Pasquini, D.; Cianfriglia, M.; Ateniese, G.; Bernaschi, M. Reducing Bias in Modeling Real-world Password Strength via Deep Learning and Dynamic Dictionaries. In Proceedings of the 30th USENIX Security Symposium, USENIX Security 2021, Online, 11–13 August 2021; Bailey, M.D., Greenstadt, R., Eds.; USENIX Association: Berkeley, CA, USA, 2021; pp. 821–838.
47. Fu, M.; Tantithamthavorn, C.; Le, T.; Kume, Y.; Nguyen, V.; Phung, D.; Grundy, J. AIBugHunter: A Practical tool for predicting, classifying and repairing software vulnerabilities. *Empir. Softw. Eng.* **2023**, *29*, 4. [CrossRef]
48. Chen, L.H.; Hsu, F.H.; Hwang, Y.; Su, M.C.; Ku, W.S.; Chang, C.H. ARMORY: An automatic security testing tool for buffer overflow defect detection. *Comput. Electr. Eng.* **2013**, *39*, 2233–2242. [CrossRef]
49. Moscovich, N.; Bitton, R.; Mallah, Y.; Inokuchi, M.; Yagyu, T.; Kalech, M.; Elovici, Y.; Shabtai, A. Autosploit: A Fully Automated Framework for Evaluating the Exploitability of Security Vulnerabilities. *arXiv* **2020**, arXiv:2007.00059. [CrossRef]
50. Egert, R.; Grube, T.; Born, D.; Mühlhäuser, M. AVAIN—A Framework for Automated Vulnerability Indication for the IoT in IP-based Networks. In Proceedings of the 2019 International Conference on Networked Systems, NetSys 2019, Munich, Germany, 18–21 March 2019; pp. 1–3. [CrossRef]
51. Blumbergs, B.; Vaarandi, R. Bbuzz: A bit-aware fuzzing framework for network protocol systematic reverse engineering and analysis. In Proceedings of the 2017 IEEE Military Communications Conference, MILCOM 2017, Baltimore, MD, USA, 23–25 October 2017; pp. 707–712. [CrossRef]
52. Eriksson, B.; Stjerna, A.; De Masellis, R.; Rüemmer, P.; Sabelfeld, A. Black Ostrich: Web Application Scanning with String Solvers. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, Copenhagen, Denmark, 26–30 November 2023. [CrossRef]
53. Eriksson, B.; Pellegrino, G.; Sabelfeld, A. Black Widow: Blackbox Data-driven Web Scanning. In Proceedings of the 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24–27 May 2021; pp. 1125–1142. [CrossRef]
54. Luo, Z.; Yu, J.; Zuo, F.; Liu, J.; Jiang, Y.; Chen, T.; Roychoudhury, A.; Sun, J. Bleem: Packet Sequence Oriented Fuzzing for Protocol Implementations. In Proceedings of the 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, 9–11 August 2023; Calandrino, J.A., Troncoso, C., Eds.; USENIX Association: Berkeley, CA, USA, 2023; pp. 4481–4498.
55. Faily, S.; Scandariato, R.; Shostack, A.; Sion, L.; Ki-Aries, D. Contextualisation of Data Flow Diagrams for Security Analysis. In Proceedings of the Graphical Models for Security—7th International Workshop, GraMSec 2020, Boston, MA, USA, 22 June 2020; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12419, pp. 186–197. [CrossRef]
56. Durumeric, Z.; Adrian, D.; Mirian, A.; Bailey, M.; Halderman, J.A. A Search Engine Backed by Internet-Wide Scanning. In Proceedings of the 22nd ACM Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 542–553. [CrossRef]

57. Alhuzali, A.; Eshete, B.; Gjomemo, R.; Venkatakrishnan, V.N. Chainsaw: Chained Automated Workflow-based Exploit Generation. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S., Eds.; ACM: New York, NY, USA, 2016; pp. 641–652. [CrossRef]
58. Yamaguchi, F.; Wressnegger, C.; Gascon, H.; Rieck, K. Chucky: Exposing missing checks in source code for vulnerability discovery. In Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, 4–8 November 2013; Sadeghi, A., Gligor, V.D., Yung, M., Eds.; ACM: New York, NY, USA, 2013; pp. 499–510. [CrossRef]
59. Stasinopoulos, A.; Ntantogian, C.; Xenakis, C. Commix: Automating evaluation and exploitation of command injection vulnerabilities in Web applications. *Int. J. Inf. Sec.* **2019**, *18*, 49–72. [CrossRef]
60. Rahaman, S.; Xiao, Y.; Afrose, S.; Shaon, F.; Tian, K.; Frantz, M.; Kantarcioglu, M.; Yao, D.D. CryptoGuard: High Precision Detection of Cryptographic Vulnerabilities in Massive-sized Java Projects. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, 11–15 November 2019; Cavallaro, L., Kinder, J., Wang, X., Katz, J., Eds.; ACM: New York, NY, USA, 2019; pp. 2455–2472. [CrossRef]
61. Li, R.; Diao, W.; Li, Z.; Du, J.; Guo, S. Android Custom Permissions Demystified: From Privilege Escalation to Design Shortcomings. In Proceedings of the 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24–27 May 2021; pp. 70–86. [CrossRef]
62. Pellegrino, G.; Johns, M.; Koch, S.; Backes, M.; Rossow, C. Deemon: Detecting CSRF with Dynamic Analysis and Property Graphs. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, 30 October–3 November 2017; Thuraisingham, B., Evans, D., Malkin, T., Xu, D., Eds.; ACM: New York, NY, USA, 2017; pp. 1757–1771. [CrossRef]
63. Lee, S.; Yoon, C.; Lee, C.; Shin, S.; Yegneswaran, V.; Porras, P.A. Delta: A security assessment framework for software-defined networks. In Proceedings of the NDSS, San Diego, CA, USA, 26 February–1 March 2017. [CrossRef]
64. Ng, C.K.; Yusof, Y.; Ab Aziz, N.S.N. DFBC Recon Tool: Digital Footprint and Breach Check Reconnaissance Tool. In Proceedings of the 2021 14th International Conference on Developments in eSystems Engineering (DeSE). Sharjah, United Arab Emirates, 7–10 December 2021; pp. 526–530. [CrossRef]
65. Redini, N.; Continella, A.; Das, D.; De Pasquale, G.; Spahn, N.; Machiry, A.; Bianchi, A.; Kruegel, C.; Vigna, G. Diane: Identifying fuzzing triggers in apps to generate under-constrained inputs for iot devices. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; pp. 484–500. [CrossRef]
66. Aljaafari, F.; Menezes, R.; Mustafa, M.A.; Cordeiro, L.C. Finding Security Vulnerabilities in IoT Cryptographic Protocol and Concurrent Implementations. *arXiv* **2021**. [CrossRef]
67. Xu, L.; Xu, M.; Li, F.; Huo, W. ELAID: Detecting integer-Overflow-to-Buffer-Overflow vulnerabilities by light-weight and accurate static analysis. *Cybersecurity* **2020**, *3*, 1–19. [CrossRef]
68. Ghanem, M.C.; Chen, T.M.; Ferrag, M.A.; Kettouche, M.E. ESASCF: Expertise Extraction, Generalization and Reply Framework for Optimized Automation of Network Security Compliance. *IEEE Access* **2023**, *11*, 129840–129853. [CrossRef]
69. Zhang, Y.; Huo, W.; Jian, K.; Shi, J.; Liu, L.; Zou, Y.; Zhang, C.; Liu, B. ESRFuzzer: An enhanced fuzzing framework for physical SOHO router devices to discover multi-Type vulnerabilities. *Cybersecurity* **2021**, *4*, 24. [CrossRef]
70. Rak, M.; Salzillo, G.; Granata, D. ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems. *Comput. Electr. Eng.* **2022**, *99*, 107721. [CrossRef]
71. Visoottiviseth, V.; Jutadhammakorn, P.; Pongchanchai, N.; Kosolyudhthasarn, P. Firmaster: Analysis Tool for Home Router Firmware. In Proceedings of the 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Pathom, Thailand, 11–13 July 2018; pp. 1–6. [CrossRef]
72. Park, S.; Kim, D.; Jana, S.; Son, S. FUGIO: Automatic Exploit Generation for PHP Object Injection Vulnerabilities. In Proceedings of the 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, 10–12 August 2022; Butler, K.R.B., Thomas, K., Eds.; USENIX Association: Berkeley, CA, USA, 2022; pp. 197–214.
73. Lee, T.; Wi, S.; Lee, S.; Son, S. FUSE: Finding File Upload Bugs via Penetration Testing. In Proceedings of the 27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, CA, USA, 23–26 February 2020. [CrossRef]
74. Chen, J.; Hu, S.; Zheng, H.; Xing, C.; Zhang, G. GAIL-PT: An intelligent penetration testing framework with generative adversarial imitation learning. *Comput. Secur.* **2023**, *126*, 103055. [CrossRef]
75. Yu, F.; Martin, M.V. GNPassGAN: Improved Generative Adversarial Networks For Trawling Offline Password Guessing. In Proceedings of the 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Genoa, Italy, 6–10 June 2022. [CrossRef]
76. Enoch, S.Y.; Huang, Z.; Moon, C.Y.; Lee, D.; Ahn, M.K.; Kim, D.S. HARMer: Cyber-Attacks Automation and Evaluation. *IEEE Access* **2020**, *8*, 129397–129414. . [CrossRef]
77. Sommer, R.; Vallentin, M.; De Carli, L.; Paxson, V. HILTI: An Abstract Execution Environment for Deep, Stateful Network Traffic Analysis. In Proceedings of the 2014 Conference on Internet Measurement Conference, Vancouver, BC, Canada, 5–7 November 2014. [CrossRef]
78. Chen, J.; Diao, W.; Zhao, Q.; Zuo, C.; Lin, Z.; Wang, X.; Lau, W.C.; Sun, M.; Yang, R.; Zhang, K. IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing. In Proceedings of the 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, CA, USA, 18–21 February 2018. [CrossRef]

79. Stallenberg, D.M.; Panichella, A. JCOMIX: A search-based tool to detect XML injection vulnerabilities in web applications. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia, 26–30 August 2019; pp. 1090–1094. [CrossRef]
80. Xu, M.; Li, S.; Xu, L.; Li, F.; Huo, W.; Ma, J.; Li, X.; Huang, Q. A Light-Weight and Accurate Method of Static Integer-Overflow-to-Buffer-Overflow Vulnerability Detection. In Proceedings of the Information Security and Cryptology—14th International Conference, Inscrypt 2018, Fuzhou, China, 14–17 December 2018; Guo, F., Huang, X., Yung, M., Eds.; Revised Selected Papers; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11449, pp. 404–423. [CrossRef]
81. Lee, S.; Wi, S.; Son, S. Link: Black-box detection of cross-site scripting vulnerabilities using reinforcement learning. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 743–754. [CrossRef]
82. Holm, H. Lore a Red Team Emulation Tool. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 1596–1608. [CrossRef]
83. Hoang, T.D.; Park, C.; Son, M.; Oh, T.; Bae, S.; Ahn, J.; Oh, B.; Kim, Y. LTESniffer: An Open-Source LTE Downlink/Uplink Eavesdropper. In Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Guildford, UK, 29 May–1 June 2023; pp. 43–48. [CrossRef]
84. Monshizadeh, M.; Naldurg, P.; Venkatakrishnan, V.N. Mace: Detecting privilege escalation vulnerabilities in web applications. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 690–701. [CrossRef]
85. Yucel, C.; Lockett, A.; Chalkias, I.; Mallis, D.; Katos, V. MAIT: Malware Analysis and Intelligence Tool. *Inf. Secur.* **2021**, *50*, 49–65. [CrossRef]
86. Johnson, P.; Lagerström, R.; Ekstedt, M. A meta-language for threat modelling and attack simulations. In Proceedings of the 13th International Conference on Availability, Reliability and Security, University of Hamburg, Germany, 27–30 August 2018; pp. 1–8. [CrossRef]
87. Liu, C.; Cui, X.; Wang, Z.; Wang, X.; Feng, Y.; Li, X. Malicescript: A novel browser-based intranet threat. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18–21 June 2018; pp. 219–226. [CrossRef]
88. Mjihil, O.; Kim, D.S.; Haqiq, A. Masat: Model-based automated security assessment tool for cloud computing. In Proceedings of the 2015 11th International Conference on Information Assurance and Security (IAS), Marrakech, Morocco, 14–16 December 2015; pp. 97–103. [CrossRef]
89. Cayre, R.; Nicomette, V.; Auriol, G.; Alata, E.; Kaaniche, M.; Marconato, G. Mirage: Towards a metasploit-like framework for iot. In Proceedings of the 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), Berlin, Germany, 28–31 October 2019; pp. 261–270. [CrossRef]
90. Calzavara, S.; Conti, M.; Focardi, R.; Rabitti, A.; Tolomei, G. Mitch: A Machine Learning Approach to the Black-Box Detection of CSRF Vulnerabilities. In Proceedings of the IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, 17–19 June 2019; pp. 528–543. [CrossRef]
91. Wei, H.; Hassanshahi, B.; Bai, G.; Krishnan, P.; Vorobyov, K. MoScan: A model-based vulnerability scanner for web single sign-on services. In Proceedings of the ISSTA '21: 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, Virtual Event, Denmark, 11–17 July 2021; Cadar, C., Zhang, X., Eds.; ACM: New York, NY, USA, 2021; pp. 678–681. [CrossRef]
92. Deng, G.; Zhang, Z.; Li, Y.; Liu, Y.; Zhang, T.; Liu, Y.; Yu, G.; Wang, D. NAUTILUS: Automated RESTful API Vulnerability Detection. In Proceedings of the 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, 9–11 August 2023; Calandrino, J.A., Troncoso, C., Eds.; USENIX Association: Berkeley, CA, USA, 2023.
93. Alhuzali, A.; Gjomemo, R.; Eshete, B.; Venkatakrishnan, V.N. NAVEX: Precise and Scalable Exploit Generation for Dynamic Web Applications. In Proceedings of the 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, 15–17 August 2018; Enck, W., Felt, A.P., Eds.; USENIX Association: Berkeley, CA, USA, 2018; pp. 377–392.
94. Kurth, M.; Gras, B.; Andriesse, D.; Giuffrida, C.; Bos, H.; Razavi, K. NetCAT: Practical Cache Attacks from the Network. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020. [CrossRef]
95. Melicher, W.; Ur, B.; Komanduri, S.; Bauer, L.; Christin, N.; Cranor, L.F. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In Proceedings of the 2017 USENIX Annual Technical Conference, USENIX ATC 2017, Santa Clara, CA, USA, 12–14 July 2017; Silva, D.D., Ford, B., Eds.; USENIX Association: Berkeley, CA, USA, 2017.
96. Rankothge, W.H.; Randeniya, S.M.N. Identification and Mitigation Tool For Cross-Site Request Forgery (CSRF). In Proceedings of the 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC), Kuching, Malaysia, 1–3 December 2020; pp. 1–5. [CrossRef]
97. Leal, A.G.; Teixeira, I.C. Development of a suite of IPv6 vulnerability scanning tests using the TTCN-3 language. In Proceedings of the 2018 International Symposium on Networks, Computers and Communications, ISNCC 2018, Rome, Italy, 19–21 June 2018; pp. 1–6. [CrossRef]
98. Chatterjee, R.; Bonneau, J.; Juels, A.; Ristenpart, T. Cracking-Resistant Password Vaults Using Natural Language Encoders. In Proceedings of the 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, 17–21 May 2015; IEEE: Piscataway, NJ, USA, Computer Society; 2015; pp. 481–498. [CrossRef]
99. Ntantogian, C.; Bountakas, P.; Antonaropoulos, D.; Patsakis, C.; Xenakis, C. NodeXP: NOde. js server-side JavaScript injection vulnerability DEtection and eXPloitation. *J. Inf. Secur. Appl.* **2021**, *58*, 102752. [CrossRef]

100. Koutroumpouchos, N.; Lavdanis, G.; Veroni, E.; Ntantogian, C.; Xenakis, C. ObjectMap: Detecting insecure object deserialization. In Proceedings of the 23rd Pan-Hellenic Conference on Informatics, PCI 2019, Nicosia, Cyprus, 28–30 November 2019; Manolopoulos, Y., Papadopoulos, G.A., Stassopoulou, A., Dionysiou, I., Kyriakides, I., Tsapatsoulis, N., Eds.; ACM: New York, NY, USA, 2019; pp. 67–72. [CrossRef]
101. Dürmuth, M.; Angelstorf, F.; Castelluccia, C.; Perito, D.; Chaabane, A. OMEN: Faster password guessing using an ordered Markov enumerator. In Proceedings of the Engineering Secure Software and Systems: 7th International Symposium, ESSoS 2015, Milan, Italy, 4–6 March 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 119–132. [CrossRef]
102. Kasemsuwan, P.; Visoottiviseth, V. OSV: OSPF vulnerability checking tool. In Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Si Thammarat, Thailand, 12–14 July 2017; pp. 1–6. [CrossRef]
103. Cao, H.; Huang, L.; Hu, S.; Shi, S.; Liu, Y. Owfuzz: Discovering Wi-Fi Flaws in Modern Devices through Over-The-Air Fuzzing. In Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Guildford, UK, 29 May–1 June 2023; WiSec '23, pp. 263–273. [CrossRef]
104. Hitaj, B.; Gasti, P.; Ateniese, G.; Pérez-Cruz, F. PassGAN: A Deep Learning Approach for Password Guessing. In Proceedings of the Applied Cryptography and Network Security—17th International Conference, ACNS 2019, Bogota, Colombia, 5–7 June 2019; Proceedings; Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11464, pp. 217–237. [CrossRef]
105. Rando, J.; Pérez-Cruz, F.; Hitaj, B. PassGPT: Password Modeling and (Guided) Generation with Large Language Models. In Proceedings of the Computer Security—ESORICS 2023—28th European Symposium on Research in Computer Security, The Hague, The Netherlands, 25–29 September 2023; Tsudik, G., Conti, M., Liang, K., Smaragdakis, G., Eds.; Proceedings, Part IV; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2023; Volume 14347, pp. 164–183. [CrossRef]
106. Campi, A.M.D.; Focardi, R.; Luccio, F.L. The Revenge of Password Crackers: Automated Training of Password Cracking Tools. In Proceedings of the Computer Security—ESORICS 2022—27th European Symposium on Research in Computer Security, Copenhagen, Denmark, 26–30 September 2022; Atluri, V., Pietro, R.D., Jensen, C.D., Meng, W., Eds.; Proceedings, Part II; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2022; Volume 13555, pp. 317–336. [CrossRef]
107. Luh, R.; Temper, M.; Tjoa, S.; Schrittwieser, S.; Janicke, H. PenQuest: A gamified attacker/defender meta model for cyber security assessment and education. *J. Comput. Virol. Hacking Tech.* **2020**, *16*, 19–61. [CrossRef]
108. Deng, G.; Liu, Y.; Vilches, V.M.; Liu, P.; Li, Y.; Xu, Y.; Zhang, T.; Liu, Y.; Pinzger, M.; Rass, S. PentestGPT: An LLM-empowered Automatic Penetration Testing Tool. *arXiv* **2023**, arXiv:2308.06782. [CrossRef]
109. Nunes, P.J.C.; Fonseca, J.; Vieira, M. phpSAFE: A Security Analysis Tool for OOP Web Application Plugins. In Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2015, Rio de Janeiro, Brazil, 22–25 June 2015; IEEE Computer Society: Piscataway, NJ, USA, 2015; pp. 299–306. [CrossRef]
110. Jain, S.; Johari, R.; Kaur, A. PJCT: Penetration testing based JAVA code testing tool. In *Proceedings of the International Conference on Computing, Communication & Automation*; IEEE: Piscataway, NJ, USA, 2015; pp. 800–805. [CrossRef]
111. Saccente, N.; Dehlinger, J.; Deng, L.; Chakraborty, S.; Xiong, Y. Project Achilles: A Prototype Tool for Static Method-Level Vulnerability Detection of Java Source Code Using a Recurrent Neural Network. In Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASE Workshops 2019, San Diego, CA, USA, 11–15 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 114–121. [CrossRef]
112. Bozic, J.; Wotawa, F. PURITY: A Planning-based secURITY testing tool. In Proceedings of the 2015 IEEE International Conference on Software Quality, Reliability and Security-Companion, Vancouver, BC, Canada, 3–5 August 2015; pp. 46–55. [CrossRef]
113. Muralidharan, M.; Babu, K.B.; Sujatha, G. Pyciuti: A Python Based Customizable and Flexible Cybersecurity Utility Tool for Penetration Testing. In Proceedings of the 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Dehradun, India, 14–15 March 2023; pp. 679–683. [CrossRef]
114. Amouei, M.; Rezvani, M.; Fateh, M. RAT: Reinforcement-Learning-Driven and Adaptive Testing for Vulnerability Discovery in Web Application Firewalls. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 3371–3386. [CrossRef]
115. Liu, Y.; Zhang, M.; Meng, W. Revealer: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities. In Proceedings of the 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24–27 May 2021; pp. 1468–1484. [CrossRef]
116. Cloosters, T.; Paaßen, D.; Wang, J.; Draissi, O.; Jauernig, P.; Stapf, E.; Davi, L.; Sadeghi, A.R. RiscyROP: Automated Return-Oriented Programming Attacks on RISC-V and ARM64. In Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses, Limassol, Cyprus, 26–28 October 2022; pp. 30–42. [CrossRef]
117. Girotto, G.; Zorzo, A.F. Robin: A Web Security Tool. *arXiv* **2020**, arXiv:2007.06629. [CrossRef]
118. Rivera, S.; Lagraa, S.; State, R. ROSploit: Cybersecurity Tool for ROS. In Proceedings of the 3rd IEEE International Conference on Robotic Computing, IRC 2019, Naples, Italy, 25–27 February 2019; pp. 415–416. [CrossRef]
119. Fagroud, F.Z.; Toumi, H.; Baddi, Y.; El Filali, S. RT-RCT: An online tool for real-time retrieval of connected things. *Bull. Electr. Eng. Inform.* **2021**, *10*, 2804–2810. [CrossRef]
120. Yin, Z.; Xu, Y.; Ma, F.; Gao, H.; Qiao, L.; Jiang, Y. Scanner++: Enhanced Vulnerability Detection of Web Applications with Attack Intent Synchronization. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 7. [CrossRef]

121. Veras, R.; Collins, C.; Thorpe, J. On Semantic Patterns of Passwords and their Security Impact. In Proceedings of the 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, CA, USA, 23–26 February 2014. [CrossRef]
122. Shcherbakov, M.; Balliu, M. SerialDetector: Principled and Practical Exploration of Object Injection Vulnerabilities for the Web. In Proceedings of the 28th Annual Network and Distributed System Security Symposium, NDSS 2021, Virtually, 21–25 February 2021. [CrossRef]
123. Genge, B.; Enachescu, C. ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services. *Secur. Commun. Netw.* **2016**, *9*, 2696–2714. [CrossRef]
124. Mikulskis, J.; Becker, J.K.; Gvozdenovic, S.; Starobinski, D. Snout: An Extensible IoT Pen-Testing Tool. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, 11–15 November 2019; Cavallaro, L., Kinder, J., Wang, X., Katz, J., Eds.; ACM: New York, NY, USA, 2019; pp. 2529–2531. [CrossRef]
125. Antunes, N.; Vieira, M. SOA-Scanner: An integrated tool to detect vulnerabilities in service-based infrastructures. In Proceedings of the 2013 IEEE International Conference on Services Computing, Santa Clara, CA, USA, 28 June–3 July 2013; pp. 280–287. [CrossRef]
126. Sommer, R.; Amann, J.; Hall, S. Spicy: A unified deep packet inspection framework for safely dissecting all your data. In Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, 5–9 December 2016; Schwab, S., Robertson, W.K., Balzarotti, D., Eds.; ACM: New York, NY, USA, 2016; pp. 558–569. [CrossRef]
127. Li, Z.; Yu, X.; Wang, D.; Liu, Y.; Yin, H.; He, S. SuperEye: A Distributed Port Scanning System. In Proceedings of the Artificial Intelligence and Security—5th International Conference, ICAIS 2019, New York, NY, USA, 26–28 July 2019; Sun, X., Pan, Z., Bertino, E., Eds.; Proceedings, Part IV; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11635, pp. 46–56. [CrossRef]
128. Holm, H.; Sommestad, T. SVED: Scanning, Vulnerabilities, Exploits and Detection. In Proceedings of the 2016 IEEE Military Communications Conference, MILCOM 2016, Baltimore, MD, USA, 1–3 November 2016; Brand, J., Valenti, M.C., Akinpelu, A., Doshi, B.T., Gorsic, B.L., Eds.; IEEE: Piscataway, NJ, USA, 2016; pp. 976–981. [CrossRef]
129. Valenza, F.; Karafili, E.; Steiner, R.V.; Lupu, E.C. A Hybrid Threat Model for Smart Systems. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4403–4417. [CrossRef]
130. Luo, C.; Li, P.; Meng, W. TChecker: Precise Static Inter-Procedural Analysis for Detecting Taint-Style Vulnerabilities in PHP Applications. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, 7–11 November 2022; Yin, H., Stavrou, A., Cremers, C., Shi, E., Eds.; ACM: New York, NY, USA, 2022, pp. 2175–2188. [CrossRef]
131. Olivo, O.; Dillig, I.; Lin, C. Detecting and exploiting second order denial-of-service vulnerabilities in web applications. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 616–628. [CrossRef]
132. Bitsikas, E.; Khandker, S.; Salous, A.; Ranganathan, A.; Piqueras Jover, R.; Pöpper, C. UE Security Reloaded: Developing a 5G Standalone User-Side Security Testing Framework. In Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Guildford, UK, 29 May–1 June 2023; pp. 121–132. [CrossRef]
133. Bertani, A.; Bonelli, M.; Binosi, L.; Carminati, M.; Zanero, S.; Polino, M. Untangle: Aiding Global Function Pointer Hijacking for Post-CET Binary Exploitation. In Proceedings of the Detection of Intrusions and Malware, and Vulnerability Assessment—20th International Conference, DIMVA 2023, Hamburg, Germany, 12–14 July 2023; Proceedings; Gruss, D., Maggi, F., Fischer, M., Carminati, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2023; Volume 13959, pp. 256–275. [CrossRef]
134. Vimala, K.; Fugkeaw, S. VAPE-BRIDGE: Bridging OpenVAS Results for Automating Metasploit Framework. In Proceedings of the 2022 14th International Conference on Knowledge and Smart Technology (KST), Chon Buri, Thailand, 26–29 January 2022. [CrossRef]
135. Blome, A.; Ochoa, M.; Li, K.; Peroli, M.; Dashti, M.T. Vera: A flexible model-based vulnerability testing tool. In Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, Luxembourg, 18–22 March 2013; pp. 471–478. [CrossRef]
136. Kim, S.; Woo, S.; Lee, H.; Oh, H. VUDDY: A Scalable Approach for Vulnerable Code Clone Discovery. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, 22–26 May 2017; IEEE Computer Society: Piscataway, NJ, USA, 2017; pp. 595–614. [CrossRef]
137. Kamongi, P.; Kotikela, S.; Kavi, K.; Gomathisankaran, M.; Singhal, A. Vulcan: Vulnerability assessment framework for cloud computing. In Proceedings of the 2013 IEEE 7th International Conference on Software Security and Reliability, Gaithersburg, MD, USA, 18–20 June 2013; pp. 218–226. [CrossRef]
138. Wu, Y.; Zou, D.; Dou, S.; Yang, W.; Xu, D.; Jin, H. VulCNN: An Image-Inspired Scalable Vulnerability Detection System. In Proceedings of the 44th International Conference on Software Engineering, Pittsburgh, PA, USA, 25–27 May 2022; pp. 2365–2376. [CrossRef]
139. Li, Z.; Zou, D.; Xu, S.; Ou, X.; Jin, H.; Wang, S.; Deng, Z.; Zhong, Y. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection. In Proceedings of the 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, CA, USA, 18–21 February 2018. [CrossRef]
140. Cigoj, P.; Blazic, B.J. An Intelligent and Automated WCMS Vulnerability-Discovery Tool: The Current State of the Web. *IEEE Access* **2019**, *7*, 175466–175473. [CrossRef]

141. Castiglione, A.; Palmieri, F.; Petraglia, M.; Pizzolante, R. Vulsploit: A Module for Semi-automatic Exploitation of Vulnerabilities. In *Proceedings of the Testing Software and Systems*; Casola, V., De Benedictis, A., Rak, M., Eds.; Springer: Cham, Switzerland, 2020; pp. 89–103. [CrossRef]
142. Li, Z.; Zou, D.; Xu, S.; Jin, H.; Qi, H.; Hu, J. VulPecker: An automated vulnerability detection system based on code similarity analysis. In Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, 5–9 December 2016; Schwab, S., Robertson, W.K., Balzarotti, D., Eds.; ACM: New York, NY, USA, 2016; pp. 201–213. [CrossRef]
143. Đurić, Z. WAPTT-Web application penetration testing tool. *Adv. Electr. Comput. Eng.* **2014**, *14*, 93–102. [CrossRef]
144. van Rooij, O.; Charalambous, M.A.; Kaizer, D.; Papaevripides, M.; Athanasopoulos, E. webFuzz: Grey-Box Fuzzing for Web Applications. In Proceedings of the Computer Security—ESORICS 2021—26th European Symposium on Research in Computer Security, Darmstadt, Germany, 4–8 October 2021; Bertino, E., Schulmann, H., Waidner, M., Eds.; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12972, pp. 152–172. [CrossRef]
145. Rankothge, W.H.; Randeniya, M.; Samaranayaka, V. Identification and Mitigation Tool for Sql Injection Attacks (SQLIA). In Proceedings of the 15th IEEE International Conference on Industrial and Information Systems, ICIIS 2020, Rupnagar, India, 26–28 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 591–595. [CrossRef]
146. Ding, J.; Atif, Y.; Andler, S.F.; Lindström, B.; Jeusfeld, M. CPS-based threat modeling for critical infrastructure protection. *ACM Sigmetrics Perform. Eval. Rev.* **2017**, *45*, 129–132. [CrossRef]
147. Agadakos, I.; Chen, C.Y.; Campanelli, M.; Anantharaman, P.; Hasan, M.; Copos, B.; Lepoint, T.; Locasto, M.; Ciocarlie, G.F.; Lindqvist, U. Jumping the air gap: Modeling cyber-physical attack paths in the Internet-of-Things. In Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy, Dallas, TX, USA, 3 November 2017; pp. 37–48. [CrossRef]
148. Castiglione, L.M.; Lupu, E.C. Hazard driven threat modelling for cyber physical systems. In Proceedings of the 2020 Joint Workshop on CPS&IoT Security and Privacy, Virtual Event, 9 November 2020; pp. 13–24. [CrossRef]
149. Evans, M.; He, Y.; Maglaras, L.; Janicke, H. HEART-IS: A novel technique for evaluating human error-related information security incidents. *Comput. Secur.* **2019**, *80*, 74–89. [CrossRef]
150. David, N.; David, A.; Hansen, R.R.; Larsen, K.G.; Legay, A.; Olesen, M.C.; Probst, C.W. Modelling social-technical attacks with timed automata. In Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats, Denver, CO, USA, 12–16 October 2015; pp. 21–28. [CrossRef]
151. Malik, S.U.R.; Anjum, A.; Moqurrab, S.A.; Srivastava, G. Towards enhanced threat modelling and analysis using a Markov Decision Process. *Comput. Commun.* **2022**, *194*, 282–291. [CrossRef]
152. Kalliamvakou, E.; Gousios, G.; Blincoe, K.; Singer, L.; German, D.M.; Damian, D. An in-depth study of the promises and perils of mining GitHub. *Empir. Softw. Eng.* **2016**, *21*, 2035–2071. [CrossRef]
153. Metzger, A. *Free and Open Source Software (FOSS) and Other Alternative License Models: A Comparative Analysis*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 12.
154. Mirhosseini, S.; Parnin, C. Docable: Evaluating the executability of software tutorials. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Virtual Event, 8–13 November 2020; pp. 375–385. [CrossRef]
155. Walshe, T.; Simpson, A. An empirical study of bug bounty programs. In Proceedings of the 2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF), London, ON, Canada, 18 February 2020; pp. 35–44. [CrossRef]
156. Lynch, K.; Ivancheva, M. Academic freedom and the commercialisation of universities: A critical ethical analysis. *Ethics Sci. Environ. Politics* **2015**, *15*, 71–85. [CrossRef]
157. University College and Union. Workload Survery 2021 Data Report. 2022. Available online: https://www.ucu.org.uk/media/12905/UCU-workload-survey-2021-data-report/pdf/WorkloadReportJune22.pdf (accessed on 10 April 2024).
158. Alhamed, M.; Rahman, M.M.H. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. *Appl. Sci.* **2023**, *13*, 6986. [CrossRef]
159. Sarker, K.U.; Yunus, F.; Deraman, A. Penetration Taxonomy: A Systematic Review on the Penetration Process, Framework, Standards, Tools, and Scoring Methods. *Sustainability* **2023**, *15*, 10471. [CrossRef]
160. Shahid, J.; Hameed, M.K.; Javed, I.T.; Qureshi, K.N.; Ali, M.; Crespi, N. A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Appl. Sci.* **2022**, *12*, 4077. [CrossRef]
161. Alzahrani, A.; Alqazzaz, A.; Zhu, Y.; Fu, H.; Almashfi, N. Web application security tools analysis. In Proceedings of the 2017 IEEE 3rd International Conference on Big Data Security on Cloud (Bigdatasecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), Beijing, China, 26–28 May 2017; pp. 237–242. [CrossRef]
162. Ravindran, U.; Potukuchi, R.V. A Review on Web Application Vulnerability Assessment and Penetration Testing. *Rev. Comput. Eng. Stud.* **2022**, *9*. [CrossRef]
163. Kowta, A.S.L.; Bhowmick, K.; Kaur, J.R.; Jeyanthi, N. Analysis and overview of information gathering & tools for pentesting. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–13. [CrossRef]
164. Zilberman, P.; Puzis, R.; Bruskin, S.; Shwarz, S.; Elovici, Y. Sok: A survey of open-source threat emulators. *arXiv* **2020**, arXiv:2003.01518.

165. Durumeric, Z.; Wustrow, E.; Halderman, J.A. {ZMap}: Fast internet-wide scanning and its security applications. In Proceedings of the 22nd USENIX Security Symposium (USENIX Security 13), Washington, DC, USA, 14–16 August 2013; pp. 605–620.
166. Lattner, C.; Adve, V.S. LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In Proceedings of the 2nd IEEE/ACM International Symposium on Code Generation and Optimization (CGO 2004), San Jose, CA, USA, 20–24 March 2004; IEEE Computer Society: Piscataway, NJ, USA, 2004; pp. 75–88. [CrossRef]
167. Juliet Test Suites. NSA Center for Assured Software. Available online: https://samate.nist.gov/SARD/test-suites/112 (accessed on 10 April 2024)
168. Bojinov, H.; Bursztein, E.; Boyen, X.; Boneh, D. Kamouflage: Loss-resistant password management. In Proceedings of the Computer Security–ESORICS 2010: 15th European Symposium on Research in Computer Security, Athens, Greece, 20–22 September 2010; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2010; pp. 286–302. [CrossRef]
169. Narayanan, A.; Shmatikov, V. Fast dictionary attacks on passwords using time-space tradeoff. In Proceedings of the 12th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 16–18 October 2005; pp. 364–372. [CrossRef]
170. Weir, M.; Aggarwal, S.; De Medeiros, B.; Glodek, B. Password cracking using probabilistic context-free grammars. In Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, Oakland, CA, USA, 17–20 May 2009; pp. 391–405. [CrossRef]
171. Pale, P.C. *Mastering the Nmap Scripting Engine*; Packt Publishing Ltd.: Birmingham, UK, 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.