A Taxonomy for Tsunami Security Scanner Plugins

Gonçalo Lima
Instituto Politécnico de Viana do Castelo
Viana do Castelo, Portugal
Email: goncalola@ipvc.pt

Vitor Hugo Gonçalves
Instituto Politécnico de Viana do Castelo
Viana do Castelo, Portugal
Email: vigoncalves@ipvc.pt

Pedro Pinto
Instituto Politécnico de Viana do Castelo
UMaia and INESC TEC
Viana do Castelo and Porto, Portugal
Email: pedropinto@estg.ipvc.pt

2023 IEEE International Conference on Cyber Security and Resilience (CSR) | 979-8-3503-1170-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSR57506.2023.10224998

***Abstract*—Vulnerability scanning tools are essential in detecting systems weaknesses caused by vulnerabilities in their components or wrong configurations. Corporations may use these tools to assess a system in advance and fix its vulnerabilities, thus preventing or mitigating the impact of real attacks.**

A set of these tools are organized by plugins, each intended to check a specific vulnerability, such as the case of the Tsunami Security Scanner tool released in 2020 by Google. Multiple plugins for this tool were proposed in a community-based approach and thus, it is important for the users and research community to have these plugins in a framework consistently categorized across multiple sources and types.

This paper proposes a comprehensive taxonomy for all the 61 plugins available, hierarchically sorted into 2 main categories, 4 categories, 4 subcategories, and 7 types. An analysis and a discussion on statistics by categories and types over time are also provided.

The analysis shows that, so far, there are 4 main contributors, being Google, Community, Facebook, and Govtech. The Google source is still the top contributor counting 39 out of 61 plugins and the highest number of plugins available are in the RCE subcategory. The plugins available are mainly focused on critical and high vulnerabilities.

I. Introduction

Organizations should promptly detect security weaknesses or improper settings in their assets to prevent the successful execution of attacks against their infrastructure. Attackers rely more on automation which imposes limited time on organizations to react to newly discovered vulnerabilities with low or high severity [1]. This is even more challenging for large organizations with hundreds to thousands of systems connected to the Internet [2].

As part of a prevention strategy for attacks targeting vulnerabilities, IT teams can rely on automated vulnerability detectors to detect security threats at scale. A set of Vulnerability Scanners are available, such as Nessus [3], OpenVAS [4], Nexpose [5], or Tsunami [6].

In particular, the Tsunami Security Scanner released in 2020 by Google is a general-purpose security scanner in a pre-alpha stage of development that uses an extensible plugin system to detect vulnerabilities. The release of the Tsunami Security Scanner has been followed by a set of plugin releases, rendering it essential for these plugins to be appropriately classified and documented, enabling its users to better perceive its capabilities and foster the creation of new plugins for existing vulnerabilities already detected.

This paper proposes a taxonomy for classifying all existing plugins of the Tsunami Security Scanner. This taxonomy categorizes all the available plugins from the sources contributing in this context and fits new upcoming plugins. The categorization is based on the organization of the Google Plugins directory in the Tsunami Plugins repository and the arrangement provided by the Nessus Network Monitor plugin families. An analysis and a discussion are also provided on statistics by type and progress over time concerning the currently available plugins.

This paper is organized as follows. Section II provides an overview of related work. Section III details the proposed taxonomy. Section IV presents the analysis and discusses the limitations of the proposed taxonomy. Section V concludes the paper, summarizing the main findings.

II. Related Work

Current research efforts assess and compare the performance of a set of vulnerability scanners.

In [7] the authors present a performance comparative study between Nessus and Retina, regarding the ability to search, Scanning Time, and the ability to detect vulnerabilities. The results show that Nessus performed faster without the active Web Application feature, and slower when the active Web Application module is enabled.

In [8] the authors review and analyze Nessus, OpenVAS, Nexpose, and GFI LanGuard vulnerability scanners in the context of WLANs. Insights are provided following practical utilization. The vulnerability scanners were found to address different vulnerabilities and to present different granularity regarding the output provided to the user. Also, it was verified that the scan duration was not consistent across tools and does not correlate with the number of vulnerabilities detected.

The authors in [9] assess network packet patterns made by penetration test activities. The authors use tools such as Winfingerprint, Superscan, Nmap, SoftPerfect Network Scanner, NeoTrace, Nessus Vulnerability Scanner, and Path Analyzer Pro. This work provides insights into the collection, reduction, and analysis of packet data gathered by penetration test toolkits.

In [10] the authors conduct a large-scale vulnerability assessment of 272 higher education institutions using Nessus. This work detected specific vulnerabilities and it was produced

an automated report mechanism to enable efficient vulnerability remediation. Their enhanced reports address 27.80% of vulnerabilities found in scanned institutions.

Authors in [11] provide a framework for testing the performance of vulnerability scanners applied to small and medium-sized enterprises. The authors compare Nessus, OpenVAS, and Nmap Scripting Engine in regard to how they can be used to automate the process of Risk Assessment in these organizations.

A set of research works have used the Tsunami Security Scanner in their research. In [12], the authors carried out an offensive security assessment of a smart tourism application that uses location proofs, with a set of vulnerability scanners including Tsunami. The results showed that Tsunami failed to detect any vulnerabilities, in contrast to other scanners assessed.

In [13], the authors compared four different security scanners OpenVas, Nessus, Nexpose, and Tsunami, and evaluated their computational, memory, and network resource usage, as well as their ability to detect vulnerabilities. They conducted experiments using a test bed consisting of four virtual machines. The authors conclude that Tsunami did not have enough plugins to be compared with the other assessed scanners.

None of the reviewed works targets the organization or the classification of the plugins of the Tsunami Security Scanner nor aids in fostering the development of new plugins for this scanner.

III. Proposed Taxonomy

According to Tsunami Security Scanner Documentation in [14], this scanner operates following the workflow presented in Fig. 1.

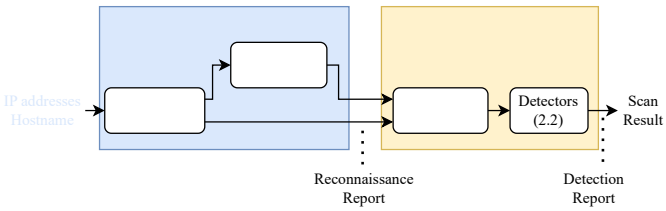

Fig. 1: Tsunami Security Scanner Workflow (based on [14])

The workflow consists of two main phases: Reconnaissance (1) and Vulnerability Verification (2), each with a set of 2 activities. In the phase of reconnaissance, Tsunami receives information about the target (IP addresses and hostname) and gathers more information (such as open ports, protocols, network services, and installed software) with PortScanner (1.1) plugins. Fingerprinter (1.2) plugins may or not be used to supplement the information obtained. At the end of this phase, a reconnaissance report is produced. In the second phase, the Vulnerability Verification (2) phase, the Tsunami Security Scanner performs a Detector Selection (2.1) to select the necessary plugins and executes the detectors (2.2), each detector checking a potential vulnerability identified in the reconnaissance phase. In the end, a detection report is generated.

The proposed taxonomy follows the described workflow and uses these phases as main categories to classify the available plugins. A Dataset of all plugins was created and published in [15]. Fig. 2 presents the main categories, the categories, and the subcategories included in the proposed taxonomy.

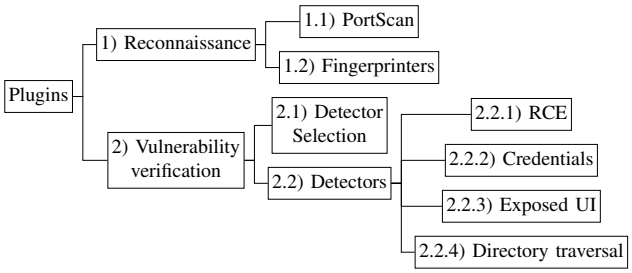

Fig. 2: Proposed main categories, categories, and subcategories

The two main categories are divided into a set of categories (according to the activities); in Reconnaissance the two categories are PortScan and Fingerprinters; in Vulnerability Verification, to categorize the plugins it was adopted the organization of the Tsunami plugin repository, specifically the Google plugins directory, which includes four subcategories RCE, Credentials, Exposed UI, and Directory traversal. These subcategories were already being used by Google, and they were used to categorize all the existing plugins, from all sources. Each subcategory can be described as follows.

A Port scan plugin is used in the Port Scanning Phase to identify open ports, protocols, and network services on the target. These plugins can be considered as wrappers around existing tools such as Nmap [16].

A Fingerprinter plugin is used to identify the specific software type and version running on a host during the Fingerprinting Phase. They may complement the information obtained during the Port Scanning Phase and help to identify all the services running on the target, particularly in cases where multiple services are running on the same port.

A Remote Code Execution (RCE) plugin detects an RCE vulnerability, i.e. a vulnerability where an attacker executes malicious code on a remote device with web applications, operating systems, and network devices. By exploiting this vulnerability, the attacker can gain full control of the system, resulting in data theft, unauthorized access to sensitive information, and disruption of system operations.

A Credentials plugin detects a vulnerability in the authentication credentials of a system or application that enables unauthorized access to sensitive data or systems, e.g. by using weak passwords, default/hard-coded credentials, or insecure storage and transmission of credentials.

The Exposed UI plugins find vulnerabilities in applications with a user interface allowing scheduling tasks and running commands. Some of these applications are used to automate tasks, and if exposed without proper authentication, attackers can execute malicious commands.

The Directory traversal plugins verify the existence of a web security vulnerability that allows attackers to read and write arbitrary files (including sensitive data, credentials, and operating system files) on the server that hosts an application.

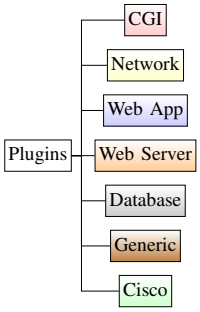

Fig. 3: Proposed types

The proposed taxonomy also classifies the existing plugins into the following seven types. These types are presented in Fig. 3 with distinct colours for each one of the following types: CGI, Network, Web App, Web Server, Database, Generic, and Cisco. These types are derived from the Nessus Network Monitor Plugin Families [17], with the exception of the Cisco type, which is present in other Nessus families and it is suitable for categorizing the Cisco plugins. The types can be described as follows.

A Common Gateway Interface (CGI) type plugin is used to detect vulnerabilities in the interfaces between web servers and clients that use CGI scripts used to execute dynamic content on web pages, such as contact forms, search bars, and e-commerce applications. Vulnerabilities in these interfaces can result in data breaches, website defacement, and other types of cyber attacks.

A Network type plugin detects security vulnerabilities in the software, hardware, or organizational processes that can result in a security breach when compromised by a threat. Attackers exploit these weaknesses to gain unauthorized access, steal sensitive information, disrupt normal system operations, or take control of a compromised system.

A Web App type plugin is intended to detect system vulnerabilities that attackers can use to compromise web applications. These differ from other vulnerabilities as web applications interact with multiple users, making them an attractive target for attackers because of their open nature. They can exploit these vulnerabilities to gain unauthorized access, steal sensitive information, and disrupt normal operations.

Web Server type plugins intend to check vulnerabilities in the front-end applications, databases, or operating systems that can lead to web server vulnerabilities. Examples of web server vulnerabilities include cross-site scripting, SQL injection, and file inclusion vulnerabilities. Attackers can exploit these vulnerabilities to gain unauthorized access, steal sensitive information, and disrupt normal system operations.

A Database type plugin detects a vulnerability in databases that, if exploited, provides attackers unauthorized access, compromises data integrity, or disrupts database operations.

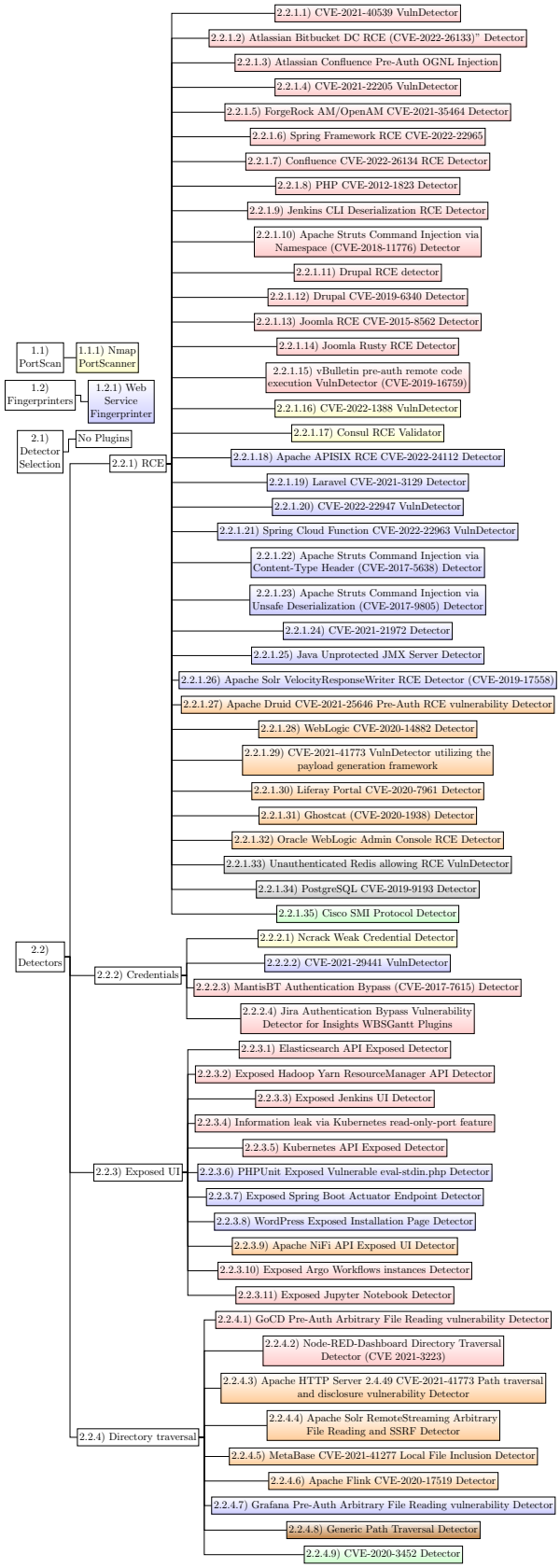

Fig. 4: Proposed Taxonomy

The Generic type plugin provides a wide range of capabilities to detect potential network security issues from SSL/TLS vulnerabilities, File sharing vulnerabilities, Denial of Service (DoS) and more.

Cisco plugins are used to verify the status of vulnerable protocols or services in Cisco devices.

The proposed Taxonomy is presented in Fig. 4, and it includes 4 categories, 4 subcategories, 7 types, and 61 plugins (all existing to date), hierarchically sorted.

IV. Analysis and Discussion

The proposed taxonomy is intended to organize all existing plugins, provide an overview of already existing plugins for reconnaissance and vulnerability verification phases, and is ready to fit new plugins. Until now there are 4 main sources contributing to the range of existing plugins: Google, Community, Facebook, and Govtech, contributing differently to the subcategories and the types of the proposed taxonomy. Table I presents the number of plugins released by Source, Subcategory and Type (empty cells indicate that no plugins were released). The release date is defined as the first commit made in each plugin directory present in the GitHub repository. From this date, any user can import it to their Tsunami tool, even if the plugin is not yet fully finished, as presented in the dataset created [15]. The results show that the majority of plugins for the Tsunami Security Scanner are currently developed by Google, with 39 out of 61 plugins being created by the company, and ExposedUI and PortScan plugins were only released by the company. In the case of Fingerprinters, the existing plugin was released by Google, despite it having later updates from both Google and the Community. Cisco plugins are exclusively released by Govtech or Facebook and these are the only plugins produced by these sources. Google is the sole creator of Databases and Generic plugins. In general, there are more plugins in the subcategory of RCE vulnerability detection (35) and for the CGI type.

TABLE I: Number of plugins by Source, Subcategory and Type

		Sources				
		Google	Community	Facebook	Govtech	Total
Subcategories	PortScan	1				1
	Fingerprinters	1				1
	RCE	21	13	1		35
	Credentials	2	2			4
	ExposeUI	11				11
	DirectotyTraversal	3	5		1	9
	Total	39	20	1	1	
Types	CGI	18	8			26
	Network	3	1			4
	Web App	9	5			14
	Web Server	6	6			12
	Database	2				2
	Generic	1				1
	Cisco			1	1	2
	Total	39	20	1	1	

Fig 5 presents the distribution of the existing plugins per Source by quarter. The results show that the major number of plugins were proposed in Q1/21 only by Google, and only one plugin was released in Q3/20 and Q2/21, created by Govtech and Facebook, respectively, and were the only plugins released by these sources during all quarters. Community-made plugins started to appear in Q4/21, which can be attributed to the rewards program [18] that started in September 2021 that offered monetary rewards ranging from $500 to $3,100 for each plugin developed by the community.

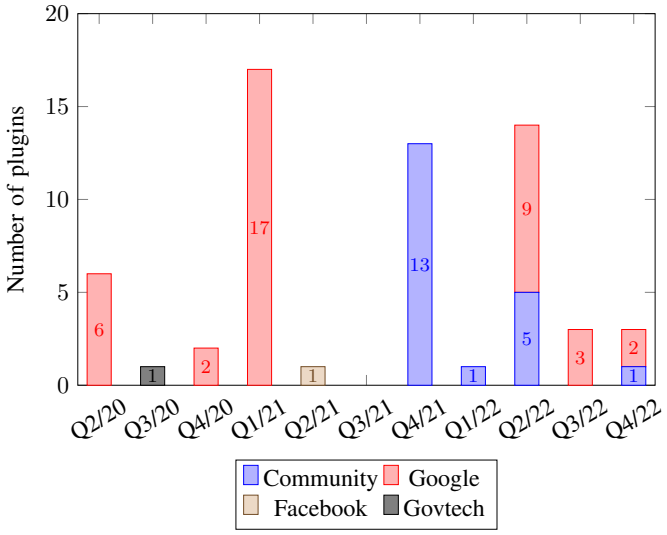

Fig. 5: Number of Plugins per Source by Quarter

Fig. 6 presents the number of plugins by subcategory released by quarter. The results show that in most of the quarters, there is at least one RCE plugin released, following the main goal of focusing on RCE-like vulnerabilities [1]. The results also show 5 plugins in the directory traversal subcategory in Q4/21, which corresponds with the rise of community-made plugins during the same period.

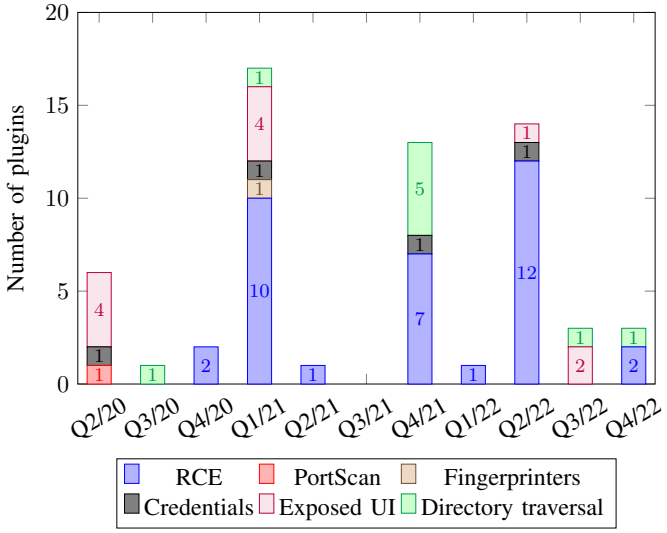

Fig. 6: Number of Plugins per Subcategory by Quarter

Fig. 7 presents the number of plugins by type released by quarter. The results show that the majority of plugins remain

concentrated within the CGI type, with a total of 26 plugins, present in most of the quarters. In contrast, types such as Database and Cisco have only two plugins each, which is comparatively low compared to other types. Furthermore, it is shown that Database plugins did not appear until 2022, similar to Generic plugins.

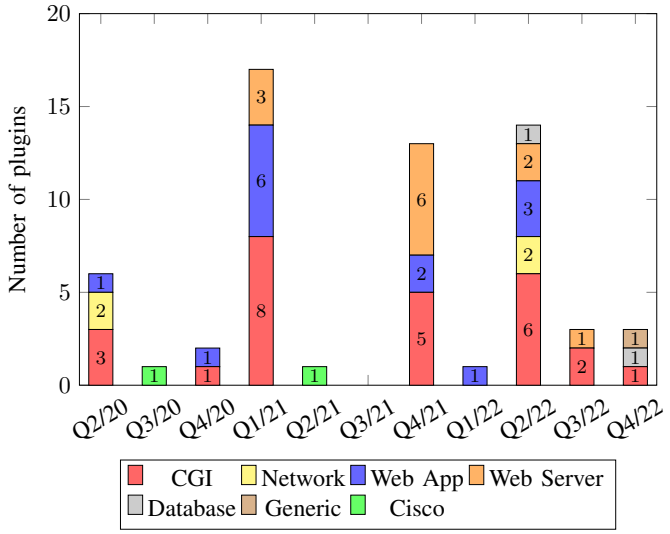

Fig. 7: Number of Plugins per Type by Quarter

From the results presented, it can be seen that on average, 5.5 plugins are released per quarter and even in quarters without any release, there is an activity in the GitHub repository by researchers fixing existing plugins, mainly from Google or from the community.

The existing plugins have different severity grading. Table II presents the number of plugins per Subcategory for each of the used CVSSv2 and CVSSv3 scores. There are no plugins from the reconnaissance phase since they do not have a Common Vulnerabilities and Exposures (CVE). The results show that the majority of the Tsunami plugins have a high CVSS score, which is on average 9.07 for CVSSv3 and 7.29 for CVSSv2. According to [19] and [20] , scores above 9 for CVSSv3 and 7 for CVSSv2 are considered critical and high, respectively. Only 38 plugins in CVSSv3 and 40 plugins in CVSSv2 have a CVE associated. This might be due to the fact that some vulnerabilities have not been publicly disclosed and have not been assigned a CVE, or that the plugin targets a vulnerability that is specific to a particular environment or configuration.

V. Conclusion

The use of security scanners is important for pre-screening existing vulnerabilities or miss configurations. The Tsunami security scanner is one of the options which is still in the ”pre-alpha” stage. This tool has the potential of being a community-driven and plugin-based security scanner.

This paper presents a taxonomy to organize a total of 61 existing plugins hierarchically sorted. The plugins are hierarchically sorted into 4 categories, 4 subcategories, and 7 types. The proposed taxonomy intends to provide an overview

TABLE II: CVSSv2 and CVSSv3 Scores per Subcategory

		Subcategory				
		RCE	Credentials	Exposed UI	Directory Traversal	Total
CVSSv2 Scores	4.3	1			1	2
	4.6	2				2
	5.0				5	5
	6.5		1			1
	6.8	3	1			4
	7.5	17	1	1		19
	9.0	2				2
	9.3	1				1
	10.0	5				5
	Total	31	3	1	6	
CVSSv3 Scores	7.2	1				1
	7.5	2			6	8
	8.1	3				3
	8.8	1	1			2
	9.8	18	2	1		21
	10.0	3				3
	Total	30	3	1	6	

of already existing plugins released, and enable future fitting of new plugins.

From the analysis provided an average of 5.5 plugins are released per quarter and, so far, there are 4 main sources contributing to the range of existing plugins: Google, Community, Facebook, and Govtech. These sources contribute differently to the subcategories and the types of the proposed taxonomy, however, the results show that the Google source is still the top contributor with 39 out of 61 plugins. The highest number of available plugins is reached in the RCE subcategory. The plugins available detect vulnerabilities with different severity ratings, however, they are focused on critical and high vulnerabilities.

Acknowledgment

This study was developed in the context of the Computer Networks and Systems Engineering Graduation at the Instituto Politécnico de Viana do Castelo (IPVC), Portugal.

This work was supported by the Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within the project “Cybers SeC IP” (NORTE-01-0145-FEDER-000044).

References

[1] *Tsunami security scanner plugins*. [Online]. Available: https://github.com/google/tsunami-security-scanner/blob/master/docs/index.md (visited on 03/03/2023).

[2] P. Costa, R. Montenegro, T. Pereira, and P. Pinto, “The security challenges emerging from the technological developments: A practical case study of organizational awareness to the security risks,” *Mobile networks and applications*, vol. 24, pp. 2032–2037, 2019.

[3] *Nessus*. [Online]. Available: https://www.tenable.com/products/nessus (visited on 03/03/2023).

[4] *OpenVAS - Open Vulnerability Assessment Scanner*. [Online]. Available: https://www.openvas.org (visited on 03/03/2023).

[5] *Nexpose Vulnerability Scanner*. [Online]. Available: https://www.rapid7.com/products/nexpose (visited on 03/03/2023).

[6] *Tsunami security scanner*. [Online]. Available: https://github.com/google/tsunami-security-scanner (visited on 03/03/2023).

[7] R. Kushe, "Comparative study of vulnerability scanning tools: Nessus vs Retina," *Security & Future*, vol. 1, no. 2, pp. 69–71, 2017.

[8] A. Kejiou and G. Bekaroo, "A Review and Comparative Analysis of Vulnerability Scanning Tools for Wireless LANs," in *2022 3rd International Conference on Next Generation Computing Applications (NextComp)*, 2022, pp. 1–6. DOI: 10.1109/NextComp55567.2022.9932245.

[9] D.-Y. Kao, Y.-S. Wang, F.-C. Tsai, and C.-H. Chen, "Forensic analysis of network packets from penetration test toolkits," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 363–368. DOI: 10.23919/ICACT.2018.8323758.

[10] C. R. Harrell, M. Patton, H. Chen, and S. Samtani, "Vulnerability Assessment, Remediation, and Automated Reporting: Case Studies of Higher Education Institutions," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2018, pp. 148–153. DOI: 10.1109/ISI.2018.8587380.

[11] I. Chalvatzis, D. A. Karras, and R. C. Papademetriou, "Evaluation of Security Vulnerability Scanners for Small and Medium Enterprises Business Networks Resilience towards Risk Assessment," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2019, pp. 52–58. DOI: 10.1109/ICAICA.2019.8873438.

[12] J. Ferrão, S. Eisa, and M. Pardal, "Offensive security assessment of a REST API for a location proof system," in *INForum - Simpósio de Informática*, Sep. 2021.

[13] R. Araújo, A. Pinto, and P. Pinto, "A performance assessment of free-to-use vulnerability scanners - revisited," in *ICT Systems Security and Privacy Protection*, A. Jøsang, L. Futcher, and J. Hagen, Eds., Cham: Springer International Publishing, 2021, pp. 53–65.

[14] *Tsunami Security Scanner documentation*. [Online]. Available: https://github.com/google/tsunami-security-scanner/blob/master/docs/orchestration.md (visited on 03/03/2023).

[15] G. Lima, V. H. Gonçalves, and P. Pinto, *Details on the Available Plugins for Tsunami Security Scanner*, version Version 1.0, Zenodo, Mar. 2023. DOI: 10.5281/zenodo.7696755. [Online]. Available: https://doi.org/10.5281/zenodo.7696755.

[16] *Nmap*. [Online]. Available: https://nmap.org/ (visited on 03/03/2023).

[17] *Nessus Network Monitor Plugin Families*. [Online]. Available: https://www.tenable.com/plugins/nnm/families (visited on 03/03/2023).

[18] *Google launches new reward program for tsunami security scanner*. [Online]. Available: https://www.zdnet.com/article/google-launches-new-reward-program-for-tsunami-security-scanner (visited on 03/03/2023).

[19] *Common Vulnerability Scoring System v3.0: Specification Document*. [Online]. Available: https://www.first.org/cvss/v3.0/specification-document (visited on 03/03/2023).

[20] *A Complete Guide to the Common Vulnerability Scoring System*. [Online]. Available: https://www.first.org/cvss/v2/guide (visited on 03/03/2023).