Atlantis Highlights in Computer Sciences, volume 4
Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)

Penetration Testing Analysis with Standardized Report Generation

Kousik Barik[1,*], A Abirami[2],Saptarshi Das[3],Karabi Konar[4],Archita Banerjee[5]

[1,3,4,5]*JIS Institute of Advanced Studies & Research, JIS University, Kolkata, India*
[2] *Bannari Amman Institute of Technology, Erode, India*
[*]*Corresponding author. Email: kousikbarik@gmail.com*

ABSTRACT
Penetration testing is a mirrored cyber-attack defined for identifying vulnerabilities and flaws in a computer system/Network/Web application— the organization appoints experts to conduct the test and present the details for deeper interpretation. One of the critical components of securing the network is to perform penetration tests of the network and web applications. In this paper, the industry-known OWASP (Open Web Application Security Project) vulnerability tool and three vulnerable web applications in a lab setup are explored and presented with a detailed analysis. Further, three penetration test reports are selected, and comprehensive analysis and reports are generated from the proposed setup. After the observation, it's understood that there is a lack of standardization format of the penetration testing reports. Therefore, this paper presents a format that will cater to the understanding of domain knowledge experts, decision-making bodies, and board members of the top executives of an organization for making further decisions on improving the robustness of their network and web applications.

Keywords: *Penetration testing, Penetration testing report, Automated testing, Web application security.*

1. INTRODUCTION

With the endless demands of information networks, daily life and work increasingly depend on information network systems. Cyber-attacks have developed into one of the biggest threats worldwide. The magnitude of damage due to cyber-attacks is increasing exponentially, and hackers perform organizational data breaches. Penetration testing is a security evaluation process for Information Technology infrastructure originating from an attack by the penetration tester. The testing process requires generating many attacks on the intended system to determine security gaps. The significant difference between a penetration tester and a hacker is that a penetration test works with a license and signs an agreement with the organization and outcomes produced as reports. Figure 1 represents the amount of monetary damage by the reported cybercrime [1] from 2011 to 2020.

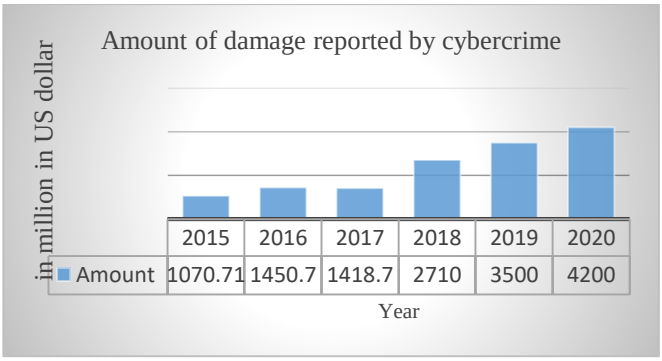

Figure 1 Damaged caused due to cybercrime.

In India [2], the reported number of registered cybercrime cases increased to 44564 in 2019 from 9622 in 2014, shown in Table 1.

Table 1. Cyber Crimes cases in India

Year	Total No of Cases Reported
2014	9622
2015	11592
2016	12317
2017	21796
2018	27248
2019	44546

As per the penetration testing survey conducted in 2020 [3], 72% of the responder noted they use open source/ freeware tools. Nearly 50% of responders reported commercial tools that open source tools may not offer. Reporting was the most prevalent factor [3] while selecting tools for penetration testing. As per survey [3], 69% of the responder listed reporting as an essential factor, maybe due to compliance requirements, which often require extensive documentation to comply with regulation and industry needs. 64% of the respondents listed multi-vector testing capabilities as the second important factor in the penetration testing report.

Expert security consultants conduct penetration testing as complex manual processes. However, the manual penetration testing process is time-consuming, expensive, and dependent on the skill set of the penetration tester. Therefore, expert professionals develop an automated tool so that non-expert users can supplant the penetration team to view the current security status. Table 2 shows the comparison between manual and automatic penetration testing [4,5].

Table 2. Comparison between manual and automated penetration testing

Phases	Manual	Automated
Testing Phases	Manual nonstandard process, high cost in customization. process, high cost in customization.	Standard process, fast in time.
Attack Database Management	Dependencies on the public database and essential to maintain the database manually.	Updates are available for the attack database.
Reporting	Manual process of collection of data.	Customization of reports based on requirements and available centrally.
Network Identification	No change is needed in systems.	Different systems modifications are required.
Auditing	Slow, complex to manage, and often inaccurate process.	Records all activity automatically.
Exploit Development and Management	Maintaining an exploit database is very difficult, and public exploits can be unsafe to run.	Product vendors maintain exploits and are easier to manage. In addition, exploits are developed by professional experts and thoroughly tested.
Training	More straightforward, to train users compared to manual testing.	Training can be customized but time-consuming.

In this context, the motivation of this work deals with studying different open source tools, web vulnerable websites online available for freeware testing. A lab environment has been set up to experiment with various attacks, generate reports on a real-time basis. Further, different penetration testing reports are studied, available online. Based on common gaps identified in reporting formats, comparative analysis and a standardized penetration testing report have been proposed.

The remaining paper is formulated as follows. Section I discusses the related works. In section II discussed methodology, set up a lab environment. Section III performed detailed analysis, captured and presented this paper. A standard penetration report based on findings has been proposed. Finally, in IV, the paper has been concluded and provided various directions for future research.

2. RELATED WORK

Kwiatkowska *et al.* [6] highlighted the significance of each tired of the web, mainly security with its service point and sufficient security check at the service point. Touseef *et al.* [7] proposed a complete study of web vulnerability measuring and recognizing relevant datasets. Khera *et al.* [8] examined and presented the lifecycle of the VAPT process and VAPT tools. They concentrate on several organization levels for adaptation requirements of security standards to defend various cybersecurity threats. Alanda *et al.* [9] highlighted vulnerability and techniques used to find an exposure in mobile-based penetration testing using the OWASP. Yulianton *et al.* [10] suggested a framework for identifying web vulnerabilities using taint analysis and black-box testing. Jinfeng *et al.* [11], a case study is conducted for vulnerability scanners using the OWASP tool. Helmiawan *et al.* [12] examined the security of the web using Open Web Application Security Project 10. Wijaya *et al.* [13] suggested a web-based dashboard for monitoring penetration testing based on OWASP standards. The dashboard is made using PHP programming languages tools and can display application vulnerabilities based on their frequency of occurrence. Lala *et al.* [14] highlighted the mitigation process of vulnerabilities in web applications using configuration changes, coding, and security updates.

Shebli *et al.* [15] explained the importance of penetration testing, components considered, the survey of tools and procedures resulted while conducting penetration testing. Zakaria *et al.* [16] proposed a penetration testing format to understand the organization's security professional and upper management needs. Shebli *et al.* [15] explained the importance of penetration testing, components considered, the survey of tools and procedures resulted while conducting penetration testing. Zakaria *et al.* [16] suggested a penetration testing format to understand the

organization's security professional and upper management needs.

3. METHODOLOGY

This study begins with selecting tools for this work—first, popular tools are chosen in the cybersecurity domain. Second, popular open source tools are studied based on Gartner Magic Quadrant for application security testing [17,18,19]. Unfortunately, due to the lack of a licensed version of the tool, it was not feasible to continue research. Therefore, only freeware tools, i.e., OWASP ZAP [20], are used in this work. Table 3 represents the version and other parameters of the OWASP tool [31-34].

Table 3. Parameter of OWASP tool

Tool Name	Availability	License	Version	Tool type	Last update
OWASP ZAP	Freeware	Apache	2.10.0	Proxy	June 2021

Two computers are used, one for the attacker and another one for the server. Both connected physically through wire via a switch, shown in Figure 2. The server computer uses Windows 10 Professional operating system with Intel (R) Core (TM.) i7 5GHz, 16 GB RAM processor. In addition, three virtual machines are installed for running web servers (named web server 1, web server 2, and web server 3). The attacker computer uses Windows 10 Professional operating system with Intel Core (R) Core (TM.) i5 2.20GHz, 8 GB. of RAM processor. Table 4 represents each component in the proposed method.

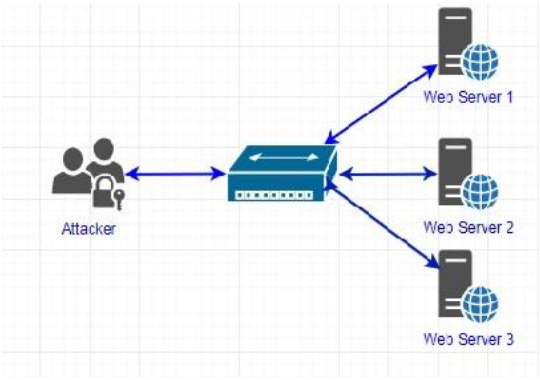

Figure 2 Proposed lab setup.

Table 4. Test environment features

	Attacker	Web Server 1	Web Server 2	Web Server 3
OS	Windows 10 Professional	Ubuntu Server 14.04.6 LTS x86	Ubuntu Server 14.04.6 LTS x86	Ubuntu Server 14.04.6 LTSx86
Hardware	8 GB RAM, Intel Core i5 2.7 GHz	1GB Ram, 1 CPU,(Virtual)	1GB Ram, 1 CPU,(Virtual)	1GB Ram, 1 CPU,(Virtual
Software	OWASP	DVWA	Google Gruyere	BWAPP

The process followed while detecting attacker patterns and evaluating vulnerable web applications is shown in Figure 3.

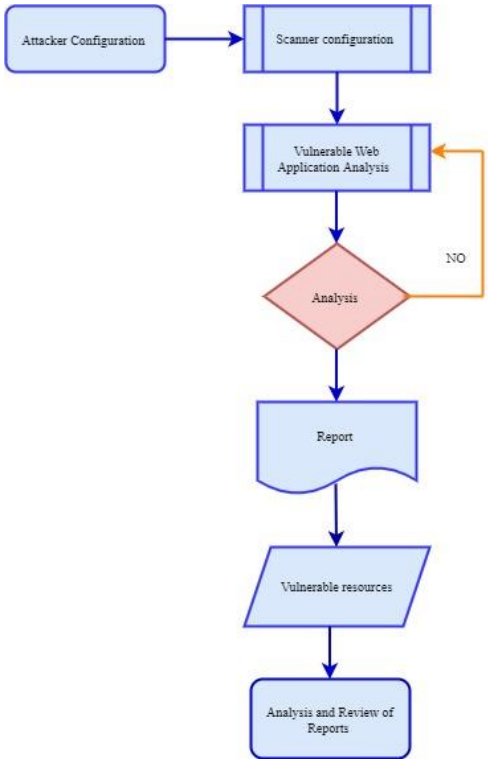

Figure 3 Process flow for detection and reporting.

3.1. Tool

OWASP ZAP: It is a freeware tool extensively accepted concerning automated vulnerability detection in web applications. It is a non-profit organization, which serves to enhance security in software. The 2.10.0 version of OWASP ZAP is utilized for this work, with default scan profiles [35-38].

3.2. Vulnerable Web Application

Three types of vulnerable web applications, namely DVWA [21], Google Gruyere [22], and BWAPP [23], are employed in this work. DVWA (Damn Vulnerable Web Application) is a web application based on vulnerability used by security professionals to test skillsets and tools in a legal environment. Google Gruyere is a web application that is thoroughly discovering bugs and learning ways to fix them. Finally, BWAPP (Buggy Web Application) is an open source web application and uses the penetration tester to find and prevent web

vulnerabilities. The analyses discussed above are performed for education purposes only.[39-42]

4. RESULT AND DISCUSSION

The experimental assessment is detailed and showed the web vulnerability identified in Table 5, Table 6, and Table 7.

Table 5. Web vulnerability detected inside DVWA

Vulnerability	No of instances
Cross-Domain Misconfiguration	19
CSP: style-src unsafe-inline	3
CSP: Wildcard Directive	3
Vulnerable JS Library	1
X-Frame-Options Header Not Set	2
Incomplete or No Cache-control and Pragma HTTP Header Set	6
X-Content-Type-Options Header Missing	16
Information Disclosure - Suspicious Comments	2

After evaluating DVWA vulnerable web applications utilizing the OWASP ZAP tool, the Cross-Domain Misconfiguration and X-content-type-options header missing amount to the maximum number of instances. Therefore, the shared vulnerability revealed by DVWA is command injection, JavaScript, domain misconfiguration, CSS, information disclosure, etc.

Table 6. Web vulnerability detected inside Google Gruyere

Vulnerability	No of instances
Cross Site Scripting (Reflected)	1
X-Frame-Options Header Not Set	60
Absence of Anti-CSRF Tokens	3
Cookie No HttpOnly Flag	2
Cookie Without SameSite Attribute	2
Cookie Without Secure Flag	2
Incomplete or No Cache-control and Pragma HTTP Header Set	62
X-Content-Type-Options Header Missing	72
Charset Mismatch (Header Versus Meta Content-Type Charset)	7
Information Disclosure - Suspicious Comments	1
Timestamp Disclosure - Unix	1

After assessing Google Gruyere's vulnerable web applications utilizing the OWASP ZAP tool, the common vulnerabilities identified are cross-site scripting, charset mismatch, CSRF, time disclosure, etc.

Table 7. Web vulnerability detected inside BWAPP

Vulnerability	No of instances
X-Frame-Options Header Not Set	4
X-Content-Type-Options Header Missing	4

After evaluating BWAPP, vulnerable web applications employing the OWASP ZAP tool, two types of vulnerability are identified, i.e., X-Frame-Options Header Not Set and X-Content-Type-Options Header Missing.

Figure 4 represents the overview of the OWASP ZAP tool, and Figure 5 illustrates the analysis of Vulnerability in DVWA web application; Figure 6 shows analysis of Vulnerability in Google Gruyere web application.

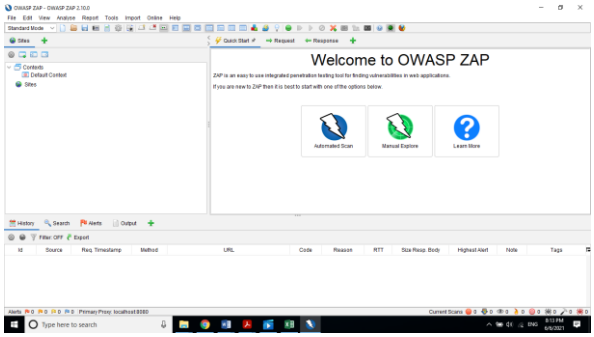

Figure 4 Overview of OWASP.

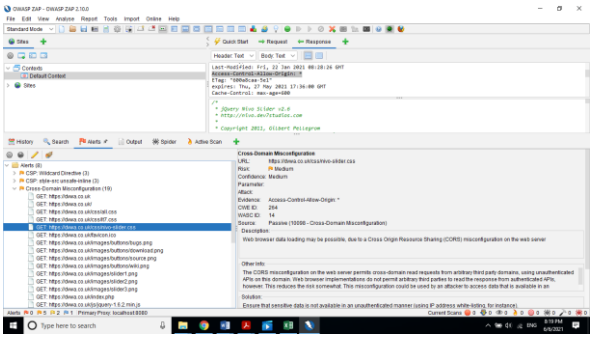

Figure 5 Analysis of Vulnerability in DVWA.

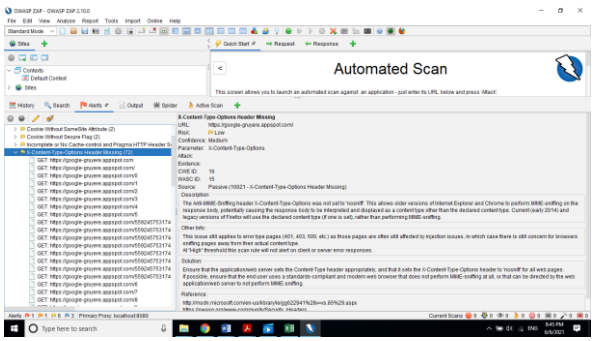

Figure 6 Analysis of Vulnerability in Google Gruyere.

In this section, the discovery of web vulnerability using open source penetration testing tools is highlighted. The primary intention is to aid security professionals in testing their skill set and tools in a legal environment.

Additionally, two online open source penetration testing tools are tested, i.e., pentest tools [24] and Forgenix [25], with three existing web-based vulnerable databases. Reports generated using the OWASP tool are represented in Figure 7.

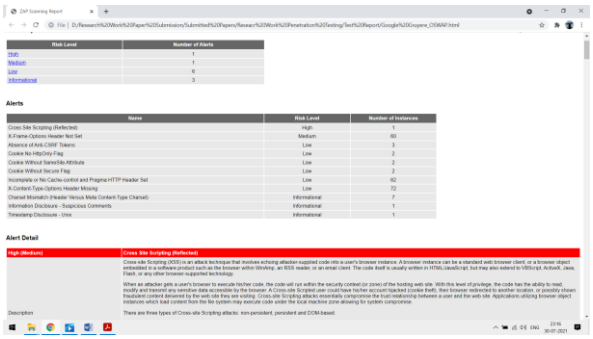

Figure 7 Generation of the report using OWASP.

Figure 8 - 9 presents reports generated subsequently from tools using three existing vulnerability applications used in the current setup.

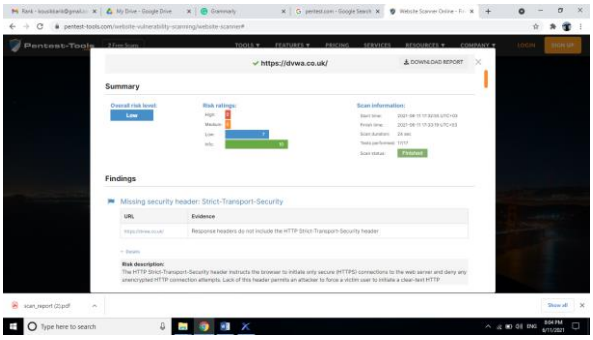

Figure 8 Generation of the report using pentest.

Figure 9 Generation of the report using Forgenix.

There is no standard format of penetration testing report from the study as mentioned above. Further, three different penetration test reports available online [26], [27], [28] are studied. No standardized report format for the above studies is discovered, shown in Annexure 1. According to the SANS (SysAdmin, Audit, Network, and Security) institute [29], four primary penetration report writing phases are shown in Fig 10. An ancient proverb in the consulting business: "If you do not document it, it did not happen" [30].

Figure 10 Penetration testing report-writing phases.

Based on the online penetration report and experimental setup study, a penetration test report has been proposed that should comprise the below-specified objectives.

Executive Summary

This section represents a comprehensive outline of the report. In addition, it contains a high-level meeting of the penetration test, findings, observations, and forms for the executive members.

1.Scope of work: This section defines the overall scope of work.

2.Objectives of work: The aim is to connect the penetration test report outcomes based on the scope of the work and direction.

3.Timeline: It shows the penetration testing start and end date, which should be included.

4.Summary of findings: It represents the overall identified risks based on priorities in Summary.

5.Summary of recommendation: This section represents the high level of advice for the target organizations based on identified risks.

Methodology

This section outlines the steps to be pursued to collect information, analysis, and the method used to measure the risk of vulnerability.

1.Planning: In this phase, the requirements of penetration testing are identified.

2.Exploitation: The penetration tester maps all the possible vulnerabilities and sees to what extent they can get into the environment.
3.Reporting: In this phase, the penetration tester reports all the detected vulnerabilities.

Detailed Findings

This section represents complete information concerning each finding; results can be described as a table, pie chart, bar graph, diagrams, etc.

1.Details of vulnerability: For all vulnerabilities, the description should be conferred about the source of the vulnerabilities, impact, and the likelihoods to be exploited.

2.Impact: It outlines the effects of vulnerability by threats.
3.Likelihood: It represents the probability of occurrence of the threat.
4.Recommendations: The penetration tester presents good recommendations on the risk rating and severity of the asset based on the outcome and discoveries. [42-44]

5. CONCLUSION

Several state-of-the-art penetration testing tools have been studied and explored in their report formats. This paper highlights the differences in reports produced by various tools. A new standard format of report generation for pen testing is proposed based on the survey carried out. This standardized format will facilitate understanding of the network vulnerabilities by the security professionals irrespective of the pen testing tool used. The tool collects the data in any format and converts it to the standardized format as per the proposal. The future scope is to construct a penetration testing tool and generate the report as the proposed idea.

REFERENCES

[1] TheStatisca, https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/

[2] National Crime Investion Bureau, The government of India, https://ncrb.gov.in/

[3] Penetration Testing Report, 2020, Coresecurity. https://www.coresecurity.com/

[4] Y. Stefinko, A. Piskozub and R. Banakh, "Manual and automated penetration testing. Benefits and drawbacks. Modern tendency," 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, 2016, pp. 488-491. DOI: 10.1109/TCSET.2016.7452095.

[5] Mirjalili, Mahin, Alireza Nowroozi, and Mitra Alidoosti. "A survey on web penetration test."International Journal in Advances in Computer Science 3.6 (2014). ISSN: 2322-5157.

[6] Kachhwaha R., Purohit R. (2019) Relating Vulnerability and Security Service Points for Web Application Through Penetration Testing. In: Panigrahi C., Pujari A., Misra S., Pati B., Li KC. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 714. Springer, Singapore.DOI:10.1007/978-981-13-0224-4_4.

[7] Touseef, P., Alam, K. A., Jamil, A., Tauseef, H., Ajmal, S., Asif, R., ... & Mustafa, S. (2019, July). Analysis of automated web application security vulnerabilities testing. In Proceedings of the 3rd International Conference on Future Networks and Distributed Systems (pp.1-8), DOI: 10.1145/3341325.3342032.

[8] Khera, Y., Kumar, D., & Garg, N. (2019, February). Analysis and Impact of Vulnerability Assessment and Penetration Testing. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) (pp.525-530),IEEE.DOI:10.1109/COMITCon.2019.886222.4.

[9] Alanda, A., Satria, D., Mooduto, H. A., & Kurniawan, B. (2020, May). Mobile Application Security Penetration Testing Based on OWASP. In IOP Conference Series: Materials Science and Engineering (Vol. 846, No. 1, p. 012036). IOP Publishing.DOI:10.1088/1757-899X/846/1/012036.

[10] Yulianton, H., Trisetyarso, A., Suparta, W., Abbas, B. S., & Kang, C. H. (2020, July). Web Application Vulnerability Detection Using Taint Analysis and Black-box Testing. In IOP Conference Series: Materials Science and Engineering (Vol. 879, No. 1, p. 012031). IOP Publishing.DOI:10.1088/1757-899X/879/1/012031.

[11] Li, Jinfeng. "Vulnerabilities mapping based on OWASP-SANS: a survey for static application security testing (SAST)." Annals of Emerging Technologiesin Computing(AETiC). DOI:10.33166/AETiC.2020.03.001.

[12] Helmiawan, M. A., Firmansyah, E., Fadil, I., Sofivan, Y., Mahardika, F., & Guntara, A. (2020, October). Analysis of Web Security Using Open Web Application Security Project 10. In 2020 8th International Conference on Cyber and IT Service Management(CITSM) (pp.1-5),IEEE.DOI:10.1109/CITSM50537.2020.9268856

[13] Wijaya, Y. S., & Ramadhani, I. (2020). Web-Based Dashboard for Monitoring Penetration Testing Activities Based on OWASP Standards. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 6(1),36-41.DOI: 10.26555/jiteki.v6i1.17019.

[14] Lala, S. K., Kumar, A., & Subbulakshmi, T. (2021, May). Secure Web development using OWASP Guidelines. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp.323-332).IEEE.DOI: 10.1109/ICICCS51141.2021.9432179.

[15] Al Shebli, H. M. Z., & Beheshti, B. D. (2018, May). A study on penetration testing process and tools. In 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-7). IEEE.DOI: 10.1109/LISAT.2018.8378035.

[16] Zakaria, M. N., Phin, P. A., Mohmad, N., Ismail, S. A., Kama, M. N., & Yusop, O. (2019, December). A Review of Standardization for Penetration Testing Reports and Documents. In 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS) (pp. 1-5),DOI: 10.1109/ICRIIS48246.2019.9073393.

[17] Gartner Application Security Testing (AST) Review and Ratings.

[18] The 2020 Gartner Magic Quadrant for Application Security Testing.

[19] A Tirosh, M. Horvath and D. Zumerle, "Magic Quadrant for Application Security Testing," Gartner, 18 April 2019. OWASP ZAP.

[20] https://owasp.org/www-project-zap/

[21] https://dvwa.co.uk/

[22] https://google-gruyere.appspot.com/

[23] http://www.itsecgames.com/

[24] https://pentest-tools.com

[25] https://foregenix.com

[26] Pen Test Report, January 1,2020, PurpleSec.

[27] Real VNC, "Penetration Test Report and Response", 2019.

[28] Offensive Security Services, LLC, Penetration Test Report : Mega Corp One,2013.

[29] SANS Institute, "Writing a Penetration Teting Report", 2010.

[30] Lam,K.,Smith, B., & LeBlanc, D. (2004). Assessing network security. Microsoft Press.

[31] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi and T. Sato, "Deep-Learning-Empowered Breast Cancer Auxiliary Diagnosis for 5GB Remote E-Health," IEEE Wireless Communications, vol. 28, no. 3, pp. 54-61, June 2021, doi: 10.1109/MWC.001.2000374.

[32] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, T. Sato, "Secure Artificial Intelligence of Things for Implicit Group Recommendations", IEEE Internet of Things Journal, 2021, doi: 10.1109/JIOT.2021.3079574.

[33] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin and G. Srivastava, "An Efficient Ciphertext-Policy Weighted Attribute-Based Encryption for the Internet of Health Things," IEEE Journal of Biomedical and Health Informatics, 2021, doi: 10.1109/JBHI.2021.3075995.

[34] L. Zhen, A. K. Bashir, K. Yu, Y. D. Al-Otaibi, C. H. Foh, and P. Xiao, "Energy-Efficient Random Access for LEO Satellite-Assisted 6G Internet of Remote Things", IEEE Internet of Things Journal, doi: 10.1109/JIOT.2020.3030856.

[35] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi and Y. Xie, "Early Collision Detection for Massive Random Access in Satellite-Based Internet of Things," IEEE Transactions on Vehicular Technology, vol. 70, no. 5, pp. 5184-5189, May 2021, doi: 10.1109/TVT.2021.3076015.

[36] L. Tan, K. Yu, A. K. Bashir, X. Cheng, F. Ming, L. Zhao, X. Zhou, "Towards Real-time and Efficient Cardiovascular Monitoring for COVID-19 Patients by 5G-Enabled Wearable Medical Devices: A Deep Learning Approach", Neural Computing and Applications, 2021, https://doi.org/10.1007/s00521-021-06219-9

[37] Puttamadappa, C., and B. D. Parameshachari. "Demand side management of small scale loads in a smart grid using glow-worm swarm optimization technique." Microprocessors and Microsystems 71 (2019): 102886.

[38] Parameshachari, B. D., H. T. Panduranga, and Silvia liberata Ullo. "Analysis and computation of encryption technique to enhance security of medical images." In IOP Conference Series: Materials Science and Engineering, vol. 925, no. 1, p. 012028. IOP Publishing, 2020.

[39] Subramani, Prabu, Ganesh Babu Rajendran, Jewel Sengupta, Rocío Pérez de Prado, and Parameshachari Bidare Divakarachari. "A block bi-diagonalization-based pre-coding for indoor multiple-input-multiple-output-visible light communication system." Energies 13, no. 13 (2020): 3466.

[40] Hu, Liwen, Ngoc-Tu Nguyen, Wenjin Tao, Ming C. Leu, Xiaoqing Frank Liu, Md Rakib Shahriar, and SM Nahian Al Sunny. "Modeling of cloud-based digital twins for smart manufacturing with MT connect." Procedia manufacturing 26 (2018): 1193-1203.

[41] Nguyen, Tu N., Bing-Hong Liu, Nam P. Nguyen, and Jung-Te Chou. "Cyber security of smart grid: attacks and defenses." In ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2020.

[42] Pham, Dung V., Giang L. Nguyen, Tu N. Nguyen, Canh V. Pham, and Anh V. Nguyen. "Multi-topic misinformation blocking with budget constraint on online social networks." IEEE Access 8 (2020): 78879-78889.

[43] Arun, M., E. Baraneetharan, A. Kanchana, and S. Prabu. "Detection and monitoring of the asymptotic COVID-19 patients using IoT devices and sensors." International Journal of Pervasive Computing and Communications (2020).

[44] Kumar, M. Keerthi, B. D. Parameshachari, S. Prabu, and Silvia liberata Ullo. "Comparative Analysis to Identify Efficient Technique for Interfacing BCI System." In IOP Conference Series: Materials Science and Engineering, vol. 925, no. 1, p. 012062. IOP Publishing, 2020.

Annexure 1.

Penetration Testing Report	Executive Summary					Methodology			Details of findings					Appendices			
	Objective	Scope of work	Timeline	Summary of outcomes	Summary of recommendation	Planning	Exploitation	Reporting	Details of vulnerabilities	Impact factor	Likelihood	Risk evaluation	Recommendations	Scanning Results	Vulnerability finding results	Other useful information if any	References
26	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
27	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓				
28	✓	✓		✓					✓	✓	✓	✓	✓				