Passive and Active Information Gathering
Vulnerability testing exercises mostly begin with collecting publicly and readily available information to understand how the target system works and where security-related challenges could exist. The information-gathering phase, sometimes called the reconnaissance phase in the Cyber-Kill chain model (Liu et al., 2020), weighs crucial importance in defining the appropriate goals and objectives of the tests. We apply passive and active information-gathering methodologies to pursue the best approaches to exploit the target system.

Passive information-gathering (PIG) is one of the less evasive methods that "does not communicate directly to the targeted system" (Thion, 2007), often proceeding before Active information-gathering (AIG). PIG collects the whois command outputs, the web application host lists on the same IP addresses, mail servers' information, and social network services. AIG requires more preparation than PIG as it sends packets and leaves traces to the target systems, which could trigger alerts on the target system. AIG collects information on open ports, running services, the version of running web applications and operating systems, and their components.

Security Challenges
Generic websites could be subject to security challenges such as (Distributed) Denial of Services, Man-in-the-middle and Data Breach. When identifying the most prominent ones for E-health themed websites, the medical fitness personal data breach is the most critical security challenge to address. Edwards et al. (2016) reported a similar incident in which Anthem Inc. breached 80 million healthcare records, including personal information, in 2015. Data breach directly impacts reputational loss, and insufficient data protection leads to even a regulatory fine. The Portugal Data Protection Authority sanctioned the Barreiro Montijo Hospital Center with a fine of 400,000 euros for improper access control, and the Hague Hospital got fined 460,000 euros against art. 32 GDPR for the insufficient internal security of patient records (GH., 2019). Vulnerability tests identify vulnerabilities before any external threats' exploitation, strengthen the client's cybersecurity, and contribute to complying with security regulations such as GDPR.

A list of the vulnerability tools with justifications and selection criteria
OWASP (N.D.) introduced 73 vulnerability testing tools as of 2021. The selection criteria to determine the tools are to examine and find critical vulnerabilities, including those related to the security challenges mentioned in the previous sections, free to use, and less evasive or less impact, meaning that the tools should provide the safe mode options instead of the attack mode, on the target web application. All methodologies in the Unit 6 guideline that are automated, manual, remote, and local are preferable to be supported by the selected tools. In this regard, we have selected OWASP Zed Attack Proxy (ZAP) and Tenable Nessus as the vulnerability testing tools.

OWASP ZAP
OWASP ZAP is a Dynamic Application Security Testing (DAST) tool that supports automated/manual testing methodologies in a local environment. It is a straightforward, actively maintained, and open-source tool used by various people with a wide range of experience in cybersecurity, acquiring valuable information about the target. There are add-ons to extend broader coverage on different

platforms and systems. The tool scans various potential vulnerabilities, including OWASP Top 10 Threats such as SQL injection and cross-site scripting that directly lead to a data breach. As the tool supports manual scan, it is possible to cover the complex test scenarios if the web application is a large, complicated system. OWASP ZAP provides a safe mode and complies with the selection criteria.

Tenable Nessus

As DAST tends to have false positives, using multiple vulnerability testing tools is essential. Tenable Nessus is another DAST tool to provide a second opinion to assure the current state. It applies automated remote testing methodologies. It is an actively maintained commercial tool that supports many useful sans, including checking the compliance maturity of security standards such as PCI-DSS, CIS and HIPAA. It finds the open ports and the version of software running on the machine and identifies vulnerabilities, classifying severity using Common Vulnerability Scoring System version 3. The tool is famous for fewer false positives, which mitigates alarm fatigue of the security staff. The justification for using this tool is that it supports various scans, including data breach-related scanning, and provides a free plan.

A list of any potential impacts on normal operations caused by using the tools and methods

OWASP Foundation (N.D.) refers to the vulnerability scanning tools as DAST. DAST tools may cause some damage to the web application and its server. Asking the client to provide a staging environment and, as OWASP Foundation (2020) recommended, avoid running in the production environment is reasonable. The tools may increase the latency and cause timeout events if the target server is low on spec. In addition, the monitoring team may detect unusual request spikes or traffic, alerting anomaly events during the tests. Hence, avoiding business hours is an option if production testing is required.

A list of limitations of the tools and outputs produced

While Tenable Nessus supports only automated testing, OWASP ZAP supports both manual and automated testing. If the web application has a complex structure, the tools cannot cover whole scenarios without structuring the manual testing, leading to failure of vulnerability discovery. Another limitation of the tools is the fact that the scanning is time-consuming. As DAST is known for its slow speed, it is necessary to plan vulnerability testing properly if the web application is large and complex. Nessus takes an hour to scan even a small website. Some other DAST scans can last 5 to 7 days (Peterson, 2021). Therefore, running vulnerability testing before deploying to the production environment is recommended to avoid exposing the vulnerabilities in public. As the nature of the tools, they cannot avoid false positives. While Tenable Nessus generated 10% false positives in the report, OWASP ZAP did 27 % (Qasaimeh et al., 2018). Hence, scrutinising the reported vulnerabilities is required, and the client needs additional resources to check if the reported vulnerabilities are the actual vulnerabilities or the false positives.

Assumption and Timeline

Due to the size of the e-health web application, we assume that the vulnerability testing will demand about an hour. Including passive and active information gathering, the scanning task will complete within a day or two.

References

Edwards, B. Hofmeyr, S. & Forrest, S. (2016) Hype and heavy tails: A closer look at data breaches. *Journal of Cybersecurity* 2(1): 3-14. DOI: 10.1093/cybsec/tyw003 [Accessed 12 September 2021].

GH., A.E.I.C. (2019) The new data protection regulation claims under GDPR. *Journal of Information Systems & Operations Management* 13(2): 100-115. Available from: http://web.rau.ro/websites/jisom/Vol.13%20No.2%20-%202019/JISOM-WI19-A07.pdf [Accessed 13 September 2021].

Liu, C., Shinghal, A. & Wijesekera, D. (2020) 'Forensic Analysis of Advanced Persistent Threat Attacks in Cloud Environments', *IFIP International Conference on Digital Forensics: Advances in Digital Forensics XVI*. New Delhi, 6-8 January. New Delhi: Springer, Cham: 161-180.

OWASP Foundation (2020) OWASP Vulnerability Management Guideline (OVMG). Available from https://owasp.org/www-project-vulnerability-management-guide/OWASP-Vuln-Mgm-Guide-Jul23-2020.pdf [Accessed: 11 September 2021].

OWASP Foundation (N.D.) OWASP Vulnerability Scanning tools. Available from: https://owasp.org/www-community/Vulnerability_Scanning_Tools [Accessed: 11 September 2021].

Peterson, J. (2021) Dynamic Application Security Testing: DAST Basics. Available from: https://www.whitesourcesoftware.com/resources/blog/dast-dynamic-application-security-testing/ [Accessed: 11 September 2021].

Qasaimesh, M., Shamlawi, A. & Khairallah, T. (2018) Black Box evaluation of web application scanners: standards mapping approach. *Journal of Theoretical and Applied Information Technology* 96(14): 4584-4596.

Thion, R. (2007) Network-Based Passive Information Gathering. *Cyber Warfare and Cyber Terrorism.* 120-128. DOI: 10.4018/978-1-59140-991-5.ch016 [Accessed 14 September 2021].

Bibliography

Daud, M. Bakar, K. & Hasan, M. (2014) 'A Case Study on Web Application Vulnerability Scanning Tools', *Science and Information Conference 2014*. London, UK, 27-29 August. 595-596.

Doupé, A., Cavedon, L., Kruegel, C. & Vigna, G. (2012) 'Enemy of the state: A state-aware black-box web vulnerability scanner', *In 21st {USENIX} Security Symposium ({USENIX} Security 12).* Bellevue, August 8-10. Bellevue: 523-538.

Fonseca, J., Vieira, M. & Madeira, H. (2007) 'Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks', *In 13th Pacific Rim international symposium on dependable computing (PRDC 2007).* Melbourne, VIC, 17-19 December. Melbourne: IEEE. 365-372.

Kushe, R. (2017) Comparative Study of Vulnerability Scanning Tools: Nessus vs Retina. *Security & Future* 1(2): 69-71.

Petukhov, A. & Kozlov, D. (2008) 'Detecting security vulnerabilities in web applications using dynamic analysis with penetration testing', *OWASP Application Security Conference.* Ghent, Belgium, 19-22 May. Ghent: OWASP. 1-16.