The Importance of Integrating Security Testing into Software Development Life Cycle to Secure Government Services

Assiya Yeraly
School of Engineering and Digital Sciences
Nazarbayev University
Astana, Kazakhstan
assiya.yeraly@nu.edu.kz

***Abstract*—One of the important things in software development is ensuring a safe user experience. This is especially relevant when the main purpose of the software being developed is the digitalization of governmental services for increasing accessibility and comfort for local citizens. A great responsibility comes along with the increased convenience and productivity that is brought by modern technological solutions. Although government representatives attempt to take the necessary measures to provide software security to their platform users, the existing methods still need to be improved and constantly updated. Incorporating security testing into the software development life cycle (SDLC) is one of such progressive improvements in enhancing cyber security.**

This survey paper found that adopting methodologies such as the Information System Security Assessment Framework (ISSAF) can help government authorities to decrease the number of cyber attacks on critical infrastructure. This methodology defines the process of penetration testing, which is aimed at identifying vulnerabilities of web applications before they are exploited by cyber attackers. It was also established that integrating the penetration testing before or after the deployment stage of the software development process is the most suitable stage. The main focus of the paper is to review the existing literature related to the topic and analyze the importance and process of integrating penetration testing into the development of governmental service platforms.

***Index Terms*—security testing, software security, cyber security, secure code, penetration testing, confidentiality, government software, software development life cycle, cyber threats**

I. Introduction

Cyber threats are one of the most acute issues in the modern software development field. Foreseeing and addressing such threats ensures their early prevention and guarantees proper integration and availability of software and confidentiality preservation of personal and sensitive information. This is not only a concern of the end users, but also of software developers and different tech organizations. However, most of the time the security of software is underprioritized since software engineers perceive the integration of security measures as an external procedure to the main development process [1]. Providing a secure user experience must be one of the highest priorities in software development. It is important to enhance trust and confidence in users, increase the quality of the software being developed, and protect against both emerging and unknown threats.

Security testing is an essential part of activities related to software testing. Its main purpose is to determine and handle issues in software application security, such as unauthorized access, malicious attacks on the software, and breaches of data security. Over the last decade, it became a common tendency for governments all over the world to digitize their services, and documentation and integrate software solutions into public health, finances, and education. Hence, there emerged a need to incorporate security testing into government software platforms, as the government possesses and manages all the personal information of its citizens. This requires a proactive approach to its security, which is integrating security testing into the software development life cycle.

Safitra et al. [2] differentiate several types of threats to information system security: unauthorized access and modification, illegal destruction, and data thievery. She suggests penetration testing execution standards as a technique to test the security susceptibility of governmental websites. The importance of applying such techniques can be observed from real-life incidents when governmental software underwent a cyber attack. An example of such a cyber attack on a governmental entity is the ransomware attack on the Metropolitan Police Department in Washington D.C. As a result, the attackers gained access to 250 gigabytes of data from the largest police departments in the USA [3].

This survey paper will first focus on the definition of security testing and available technologies and procedures of security testing. Then we will discuss the state of security level of government platforms worldwide, examples of recent cyber attacks on governmental online services, and the importance of security integration into the software development life cycle to increase the efficiency of measures taken to ensure secure user experience. The limitations of this paper include not covering all the existing security testing approaches, such as risk assessment, security audit and vulnerability scanning. This paper also did not research the integration of security testing techniques into all the SDLC stages.

II. Security testing and software development

A. A brief overview of security testing

Software security testing is a procedure of making the software correspond to the expected behavior when exposed to malicious attacks both intentional and unintentional. Security testing used to be solely viewed as network port scanning. However, the concept of software security testing has significantly evolved since then, and it is not restricted by black box or functional testing techniques. These testing techniques require following specific testing requirements, and while black box testing examines the output and input of the software without any prior programming knowledge, functional testing requires the tester to be acquainted with the internal workings of the software, such as databases, integration of the software with other systems and application programming interfaces.

According to Potter and McGraw [4], in security testing the main focus of testers is to employ strategies based on risk assessment, considering both the structure of the system and how attackers might exploit the existing vulnerabilities. This helps in assessing software security more effectively and ensures full test coverage of the software elements with high cyber threat risks. By identifying risks within the system and designing tests around them, testers can concentrate on parts of the code where attacks are more likely to occur. This method offers greater assurance of software security compared to traditional testing approaches.

It is a common tendency for software applications to have security-related issues after being released for use. Most of the time developers choose to not pay attention to securing their software products during the development cycle, or attempt to after launching the applications. Moreover, the security measurement has been viewed as a separate process that has to be handled by professionals independent from the developer's team. This is considered as a serious issue in the software development field because securing software products at the development stage is more efficient in terms of both cost and success rate in cyber attack prevention.

According to the Open Web Application Security Project foundation, two-thirds of web apps have serious security issues [5]. This could be prevented if security testing was part of the software development life cycle.

B. Frameworks and Practices for Software security

There exist several frameworks that focus on assisting information technology companies and organizations with available and effective software security practices. Such frameworks include the Building Security in Maturity Model, the Open Web Application Security Project, The US National Institute of Standards and Technology, and the Cybersecurity Framework. These frameworks follow a 5-purpose model: determine, secure, locate, acknowledge, and revive. As Weir et al. [6] state, by following this model, these frameworks increase the awareness of cyber security risks and promote approaches that can improve the software security level in different tech companies. It was also found that security testing is one of the most studied software security practices [7].

In his research, Mark Kreitz [8] discusses security by design as one of the applications of security testing during the software development process. The author believes that his definition of security by design is different from the one that is commonly accepted in a testing community. He argues that the previous comprehension is not strong enough, as it does not cover the implementation phase of the software, which is known to be the most time-consuming stage of the SDLC. Considering that these days the agile methodology is increasingly applied by developers, there is a high possibility of overlooking new security breaches. The overlooks are particularly expected as a result of the main scope of agile methodology: continuous change integration at any stage of software development. The author also suggests introducing Integrated development environment features that will assist the software developers with security testing throughout the development activities. This requires the development of security code analysis tools, which can be created by using existing technologies. These technologies are based on two types of code analysis: dynamic and static analysis.

Penetration testing is a popular dynamic analysis approach. After assessing the possible vulnerabilities of a software, the security tester tries to act as a cyber attacker, while being aware of the vulnerabilities. As Shah and Mehtre [9] state, the software vulnerabilities are grouped into similar sets and used as inputs during the testing. As a result, the tester identifies the extent to which a potential hacker is able to successfully gain access to the system. This testing process consists of several steps, which include planning and preparation, identification and intrusion, subsequent exploitation and information extraction, and lastly, documentation and cleanup.

Static code analysis is believed to be more effective than dynamic analysis, as it can be used even in situations when the code can not be compiled. A widespread tool that represents the static code analysis approach is FindBugs. This tool is mainly used to test software written in Java programming language. Its functionality is based on comparing the program code with predefined bug patterns and providing the tester with warnings. Despite its efficiency in early security issue detection and popularity among developers, it is not the most reliable tool due to high rates of false warning generations [10].

III. Security in governmental software platforms

A. The importance of security testing in governmental online platforms

The modern system of government was created well before modern information technologies arose. The general public not only trusts the government with managing their tax money, international security, health, and education but also holds it accountable for the cyber protection of these critical infrastructure and public institutions. The government is not the only authority that possesses the rights and power to influence and control cyber security all over the country. Big private technological companies and institutions have similar or sometimes even larger amounts of resources to affect the level of cyber protection of the essential software in the country. In spite of that, people still perceive the government as the only entity that has to take preventative measures against any cyber-attacks that can disturb the functionality of the software. Besides, any user of the software is capable of increasing the risks of cyber attacks by not being careful while using the services that the government provides them online.

As Shandler and Gomez [11] point out, the main goal of cyber attackers aiming at governmental infrastructures might be specifically to destroy the trust that people have in government and demonstrate that they are not protected against any cyber threats. Such actions cause panic and unrest in society, which makes the country vulnerable to destruction. People are worried about their safety, the safety of their finances, personal information, and confidentiality. The ways that cyber attackers can abuse critical software when they gain access to them is able to bring serious harm to the general public. Integrating security testing into SDLC of the governmental infrastructures can help to prevent such situations and improve the current security policies and measures.

The importance of integrating security testing into the lifecycle of software development can be seen in the increasing number of cyber attacks on government software applications. For example, there was a serious incident of malware attack in the City of Riviera located in the USA in 2019. As reported by Hassanzadeh et al. [12] a computer intruder tricked an employee in the police department into opening a harmful email. The consequences of this incident affected the work of the police department, the local city government, and the water service department by leading to the encryption of their data in the shared computer system. This cyber attack also interfered with the water service department from supervising the water pumping stations and water quality testing activities.

Another example of such disturbing cyber attacks is the incident that happened to the largest and the most significant police department in the USA. The D.C. Metropolitan Police Department reported that they experienced a severe ransomware attack on April 26th, 2021. As a result of this cyber attack, an organization that calls itself "Babuk" gained access to 250 gigabytes of sensitive data that contained all the information about the police department's working processes. After encrypting the personal information about the department, the organization demanded $4 million for returning it and not posting it for the general public online. The police department could not pay such an amount of money, so the Babuk hackers leaked the data on May 13th, 2021 [3].

According to Safitra et al., [2], the State Cipher and Cyber Agency's National Cyber Security Operations Center documented 88,414,296 cyber-attacks between the period of January 1st and April 12th, 2020. These incidents clearly show the importance of finding solutions to improve the cyber security of government computer systems and demonstrate that governments do not take enough measures to prevent such situations.

B. *Appropriate security testing techniques and their efficiency*

A sophisticated penetration testing methodology that was specifically developed to assess the security level of systems, application controls, and networks is the Information System Security Assessment Framework (ISSAF). This methodology consists of three main components: planning and preparation, assessment and reporting. During the planning and preparation step, the software testers arrange the necessary testing tools, official document agreements with the customers, and time constraints of the testing period, and identify the obligations of the engagement team. The next step is assessment, which defines the actual process of penetration testing [13]. This step consists of nine stages, which are described in detail by Sanjaya et al. [14].

The first stage is Information Gathering, where information about the software being tested is collected by investigation. This investigation helps identify the necessary set of procedures to be taken to follow the provided requirements. During this stage of assessment, the Domain Information Groper command line tool is used to acquire information about Domain Name System servers. The next stage of network mapping is required for gathering information about the website's physical network and accessible network devices. Vulnerability identification is carried out after network mapping to find the possible vulnerabilities of the website. Such identification is necessary for the fourth, penetration stage, where software testers reproduce cyber attacks to locate the security breaches. Gaining access and privilege escalation is another stage, where the process of accessing the website's critical systems is tested. Password requirements of the website are evaluated during the enumerating stage to find more vulnerable elements of the website. By compromising remote users in the system, the security weaknesses of the remote access system are determined in the seventh stage. The last stages include the testers attempting to maintain the access they gained to the website without being detected by the existing security measures and finally, getting rid of the evidence of penetration testing activities.

The last major component of the penetration testing methodology is reporting. The reporting step is dedicated to writing a comprehensive report about the whole testing process and eliminating all the artifacts that were created during the assessment step. This penetration testing framework excludes random cyber attack selection, which saves time and resources during the testing. However, the authors also highlight the downside of this approach, and state that the reporting step is not clearly defined in terms of specific requirements for writing and structuring the final report [13].

Hassan and Ahmad [15] suggest two phases that are the most suitable for integrating this testing technique into the SDLC, which are the stage before the software deployment phase and the stage after the software deployment. The complete SDLC stages include 7 phases: arrangement, conditions, software design, development, testing, deployment and maintenance. They also found that the most useful operating system for identifying cyber security vulnerabilities during penetration testing in SDLC is Kali Linux, as it contains more than 600 factory-integrated tools for penetration testing. These authors conducted a further research and determined the top three web scanners that can be used to assist with penetration testing: OWASP Zed Attack, Netsparker, and Acunetix. Lastly, the authors also identified four hacking approaches that are the most productive during penetration testing. These hacking methods include phishing, which is defined as using spam emails and links to gain the user's personal information. The second hacking technique is malware, or harmful files, which are used to infect the computer system with viruses. SQL injection enables the cyber attacker to interfere with the website's database queries. Finally, session hijacking allows the cyber attacker to acquire control over the user's web session.

Thus, ISSAF is an elaborate penetration testing methodology that covers considerable aspects of security testing activities. Integrating this procedure after or before the deployment phase of SDLC will ensure the detection of substantial part of the software vulnerabilities.

IV. Ethical Considerations

Penetration testing conducted to identify vulnerabilities in web applications has to be adapted for ethical hacking purposes only. Government online services mostly work with people's personal, sensitive, and financial data related to their jobs and property. Testers that gain unauthorized access during an authorized activity must not abuse it with harmful intentions. They also have to strictly follow the testing process structure and regulations specified before the start of the testing activities. The information about the detected vulnerabilities has to be fully reported to the corresponding security team and kept confidential without disclosing it to a third party. The testing team has to also address their work with professionalism, continuously learn new technologies and frameworks, and be aware of any changes in legal regulations of cyber security.

Conclusion

As government entities from all over the world started using software products to automatize their working processes and digitize citizen services, it became of great significance to improve the present cyber security measures. Security testing is used to establish the behavior of the software when it is subjected to cyber attacks. The complications emerge when software developers refuse to integrate security testing measures into the software development life cycle. Although it was found that users of two-thirds of web applications experience personal data thievery, improper functionality of the software, and other types of cyber security issues, integrating security testing into the SDLC is not a widespread practice in government online platforms design.

This paper also identified that most frameworks that are focused on promoting cyber security among digital firms suggest using security testing as a way to enhance the cyber security level of their software. Penetration testing is one of the noteworthy security testing techniques that was described as a popular dynamic analysis method. The ISSAF framework provides its users with an efficient application of the testing technique, which consists of three main steps: planning and preparation, assessment, and reporting. Integrating the penetration testing after or before the deployment step of SDLC was found to be the most appropriate. It is important for the penetration testers not to misuse the access to the critical computer systems that are granted to them. It is recommended for further research to study the perspectives of integrating the security testing into other SDLC phases besides deployment.

References

[1] K. Rindell, K. Bernsmed, and M. G. Jaatun, "Managing security in software: Or: How i learned to stop worrying and manage the security technical debt," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pp. 1–8, August 2019.

[2] M. F. Safitra, M. Lubis, and A. Widjajarto, "Security vulnerability analysis using penetration testing execution standard (ptes): Case study of government's website," in *Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, (Fukuoka Institute of Technology, Fukuoka, Japan), pp. 139–145, March 2023.

[3] E. Caroscio, J. Paul, J. Murray, and S. Bhunia, "Analyzing the ransomware attack on dc metropolitan police department by babuk," in *2022 IEEE International Systems Conference (SysCon)*, pp. 1–8, April 2022.

[4] B. Potter and G. McGraw, "Software security testing," *IEEE Security Privacy*, vol. 2, pp. 81–85, Sept.-Oct. 2004.

[5] R. Cope, "Strong security starts with software development," *Network Security*, vol. 2020, no. 7, pp. 6–9, 2020.

[6] C. Weir, S. Migues, M. Ware, and L. Williams, "Infiltrating security into development: Exploring the world's largest software security study," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1326–1336, August 2021.

[7] S. Wen, "Software security in open source development: A systematic literature review," in *2017 21st Conference of Open Innovations Association (FRUCT)*, pp. 364–373, 2017.

[8] M. Kreitz, "Security by design in software engineering," *ACM SIGSOFT Software Engineering Notes*, vol. 44, no. 3, pp. 23–23, 2019.

[9] S. Shah and B. M. Mehtre, "An overview of vulnerability assessment and penetration testing techniques," *Journal of Computer Virology and Hacking Techniques*, vol. 11, pp. 27–49, 2015.

[10] H. J. Kang, K. L. Aw, and D. Lo, "Detecting false alarms from automatic static analysis tools: How far are we?," in *Proceedings of the 44th International Conference on Software Engineering*, pp. 698–709, May 2022.

[11] R. Shandler and M. A. Gomez, "The hidden threat of cyber-attacks–undermining public confidence in government," *Journal of Information Technology Politics*, vol. 20, no. 4, pp. 359–374, 2023.

[12] A. Hassanzadeh, A. Rasekh, S. Galelli, M. Aghashahi, R. Taormina, A. Ostfeld, and M. K. Banks, "A review of cybersecurity incidents in the water sector," *Journal of Environmental Engineering*, vol. 146, no. 5, p. 03120003, 2020.

[13] M. Prandini and M. Ramilli, "Towards a practical and effective security testing methodology," in *The IEEE Symposium on Computers and Communications*, pp. 320–325, IEEE, June 2010.

[14] I. G. A. S. Sanjaya, G. M. A. Sasmita, and D. M. S. Arsa, "Information technology risk management using iso 31000 based on issaf framework penetration testing (case study: Election commission of x city)," *International Journal of Computer Network and Information Security*, vol. 12, no. 4, pp. 30–40, 2020.

[15] S. Z. ul Hassan and S. Z. Ahmad, "The importance of ethical hacking tools and techniques in software development life cycle," *International Journal*, vol. 10, no. 3, 2021.