Website Security Analysis of Department and Integrated Services One Door of XYZ Regency using Kali Linux

Poltak Andreas Marbun[1], Ahmad Almaarif[1] and Avon Budiono[1]
[1]*Information System, School of Industrial and System Engineering, Telkom University, Bandung 40257, Indonesia*

Keywords: Vulnerability assessment, Web application, Kali Linux, Security, Web vulnerability scanners

Abstract: Protecting the system from an attack is important in the information system security process. Information system security in a company or organization must be given special attention because this is very important for confidentiality, integrity, and availability data on the system. Therefore, we need a method to maintain the security and confidentiality of data, so that the data can only be accessed by certain people. Weaknesses on a website will never be known if there is no audit of the system. The website security audit process is carried out with the aim of getting a system vulnerability gap from the website. The weakness from the website will be used as a parameter to provide solutions or suggestions to improve the system. The purpose of this research is to conduct a security level audit on Investment Department and Integrated Services One Door of XYZ Regency. There are three tools carried out for the audit, namely Nmap, Paros, and Vega with Kali Linux operating system. The method used in this research is Vulnerability Assessment Methodology. The scope of this research is to find the vulnerability of a website and then provide a solution only to that vulnerability.

1 INTRODUCTION

The issue of security is one of the important aspects of an information system. Often the order of security in a system is ranked second, or even last in the list of things that are considered important. As security at the network perimeter has improved, attackers have transferred their efforts to the easier target of web applications (Walden, 2008). In web application, an attacker can exploit the data by attacking the target website that can cause damage or steal any information on the system (Mahmoud et al., 2017). A report that has been collected highlights that 86% of 30,000 websites were tested has at least one serious vulnerability, and most have more than one (Subedi et al., 2016). Web applications are difficult to secure because they are open to the public and can be accessed by everyone, including malicious users (BAYKARA, 2018). Web applications vulnerability has been widely recognized as a serious problem because data breaches due to this vulnerability have been repeated in recent years and will likely continue to be a major problem in the future (Bau et al., 2012). Attacks on a network connected to the Internet can be done by anonymous (Montieri et al., 2019). One of the software that can be used in anonymously attacking a network is Tor (Montieri et al., 2019). Gartner Group estimates that more than 70% of attacks on a company's website or web apps are in the application layer, not a network or system layer therefore, network vulnerability scanners, network firewalls, and use of Secure Socket Layer (SSL) do not guarantee the security of a website (BAYKARA, 2018).

Investment Department and Integrated Services One Door of XYZ Regency is one form of service provided by the local government in providing services to the community. This department provides online services in the form of web applications to facilitate the public in accessing information and to take care of matters relating to investment and the process of fulfilling business licensing in XYZ Regency. The department's web application certainly stores a lot of sensitive public and government data. Therefore, we need a good security for this web application in order to minimize the hackers who want to retrieve the data.

This research aims to conduct a security analysis on the department's web application using Kali Linux. After the security analysis has been completed, it will be given recommendation or proposal in terms of security development of the website system. This can contribute to help the Investment Department and Integrated Services One Door of XYZ Regency find out the weaknesses of their web applications so that security can be improved in the future.

Marbun, P., Almaarif, A. and Budiono, A.
Website Security Analysis of Department and Integrated Services One Door of XYZ Regency using Kali Linux.
DOI: 10.5220/0009908302270233
In *Proceedings of the International Conferences on Information System and Technology (CONRIST 2019)*, pages 227-233
ISBN: 978-989-758-453-4
Copyright © 2020 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

2 LITERATURE REVIEW

2.1 Web Application Vulnerabilities

Web application can be described as a program that is developed in order to perform specific processes and normally handles the user's input in a script and includes database data collection (Charpentier Rojas, 2013). Web application vulnerabilities are security vulnerabilities that are present on the system. Attackers have the potential to use many different loopholes in different attacks on the system, so that each vulnerability has its own risks in the security of the web application (M. Sevri, 2016). Some major threats to the web server layer are SQL injection, unauthorized server access, and password hacking attacks. Most SQL injection vulnerabilities are caused by weak input validation (BAYKARA, 2018).

2.1.1 Injection Vulnerabilities

These types of attacks include data injection, command injection, resource injection, and SQL injection attacks (BAYKARA, 2018). SQL Injection does not filter correctly user input of web applications and places them directly into SQL statements (Tajpour et al., 2011). This allows data in the database to be stolen or modified. Another possibility that can occur is an executable script that is forced to do things that were not anticipated by the author.

2.1.2 Cross Site Scripting (XSS)

Cross site scripting is an attack in the form of malicious script code that is injected into a web application that wants to be attacked (M. Sevri, 2016). Cross site scripting exploits vulnerabilities in web applications in the form of inputs and outputs that are not validated or coded. Hackers inject malicious script code into the web pages and when a user visits a website the evil script can exploiting user credentials such as hijacking session ID, password, credit card number or cookie (Mahmoud et al., 2017).

2.1.3 Security Misconfiguration

Misconfiguration in the web application server can be fatal for the security of the system. The reason is because Apache, MySQL and PHP (AMP) are web application server environment is the most widely used, and these components are open source (Eshete et al., 2011). It can cause vulnerabilities in Web applications that hackers can exploit

2.1.4 Authorization, Authentication, and Access Control

This vulnerability can allow hackers to control applications or back-end servers. These attacks include weak password management, use of weak encryption methods, authentication errors, and cryptographic errors (BAYKARA, 2018). This threat can endanger integrated applications or systems related to stolen identity data.

2.2 Web Application Attacks

Crimes in cyber security for now have been very difficult to avoid. One of the events that often happens now is hacking a web application of a company or organization (Joshi and Singh, 2016). Web application attacks can occur due to several possibilities, such as security misconfiguration, session management and authentication, or other problems (Mitropoulos et al., 2017). Hackers usually first conduct a vulnerability assessment to find out the weaknesses of the target web application. Weaknesses with the highest level of web applications will be used by hackers to carry out attacks (M. Sevri, 2016).

Figure 1: Web Application Attacks Model.

Figure 1 describe that the attack can be carried out by the attacker for each vulnerability. Each vulnerability also has its own impact.

2.3 Confidentiality, Integrity, Availability (CIA)

Confidentiality, Integrity, Availability (CIA) is the center of information security. The three aspects above are very important because it is very influential on the level of information system security (Sumra et al., 2015). If one aspect fails, this can provide a way for hackers to compromise with your network and data. However, the mixture of the three sections depends on each company, project or asset used. Some companies may value confidentiality above all, others may give the highest value on availability (Qadir and Quadri, 2016). Effective cyber security is knowing what vulnerabilities are to the system and can protect them from hacker attacks.

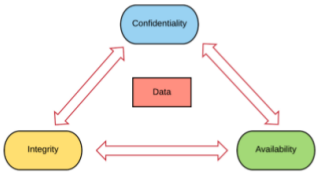

Figure 2: CIA Model.

Figure 2 explains that a good data must have confidentiality, integrity, and availability.

2.3.1 Confidentiality

Confidentiality is all about privacy and is a user's privilege. Only those who need access to specific information must be provided, and steps need to be taken to ensure sensitive data is prevented from falling into the wrong hands (Qadir and Quadri, 2016). The more critical the information, the stronger security measures need to be applied. This ensures that confidentiality support can include data encryption, ID and password, and two-factor authentication.

2.3.2 Integrity

Integrity refers to trust in a data or information, the trust includes consistency, completeness, and accuracy of the data or information available. Information integrity is very important, and organizations need to take the steps necessary to ensure that information remains accurate throughout its entire life cycle. In this case also need to be considered that all data or information must be backed up so that it can be recovered if there is an attack from hackers.

2.3.3 Availability

Availability is the availability of data or information, which means that data or information is always available when needed for people who have permission to that information. So that when needed by the user, data or information can be quickly accessed and used. To guarantee the availability of data or information, it is necessary to backup data to anticipate unexpected things.

2.4 Kali Linux Operating System

Kali Linux is a very popular operating system in terms of digital forensics, penetration testing, and security audits (Gunawan et al., 2018). Kali Linux was first released on March 13, 2013 based Debian files and systems in accordance with the File system Hierarchy Standard (FHS) (Babincev and Vuletić, 2016). Kali Linux Operating System can be downloaded freely and for free on its official website on the internet. Kali Linux can also be installed in a virtual environment or as a main operating system. In addition, there is also the possibility of using two operating systems in one computer with a dual boot method, it depends on the user's comfort in its application. Kali Linux already has more than 600 penetration testing tools and wideranging wireless device support(Gunawan et al., 2018). The many choices of security tools owned by Kali Linux can be utilized by users in conducting security analysis on web or network applications.

2.4.1 Kali Linux Tools

Kali Linux consists of hundreds of tools that exist on the operating system. These tools are grouped into sections based on their functionality and utility. Each section performs a different task but with the same goal to do penetration testing and security analysis both on the network or web application (Gunawan et al., 2018). The following are categorizing tools in Kali Linux based on their functions (Babincev and Vuletić, 2016):

1. Information Gathering
   The reconnaissance tools used for collecting data on the target network and devices. Tool identify the device until the protocol is used.
2. Vulnerability Analysis
   This tool focuses on evaluate and analyze vulnerabilities in a system.
3. Web Applications
   These tools are used to analyze vulnerabilities in web servers. However, this tool does not always refer to attacks on web servers, but can also audit web applicationbased network services.
4. Password Attacks
   This tool is useful for performing Brute force password attacks used for authentication on a system.
5. Wireless Attacks
   This tool is used to analyze vulnerabilities on wireless networks.
6. Exploitation Tools
   This tool is used which is used to exploit vulnerabilities found in the system.
7. Sniffing and Spoofing
   This tool is used for network packet capture and manipulate network packets.
8. Database Assessment
   This tool is used to look for vulnerabilities and database attacks on web applications.

9. Reverse Engineering
   This tool functions to analyze how a program is developed so that it can be copied, modified, or so that it can lead to the development of other programs. This tool is lso used for malware analysis or in finding vulnerabilities in software applications.
10. Forensics
    Forensic tools are used to monitor and analyze computer network traffic and applications on a system.
11. Reporting Tools
    This tool serves to provide information found after doing penetration testing.
12. System Services In this section we can enable or disable the kali linux service.

3 METHODOLOGY

Vulnerability Assessment is one methodology used in analyzing or auditing the security system of a network or web application (Nath, 2011). The Vulnerability Assessment aims to measure and prioritize risks associated with network and host-based systems or devices to enable rational technology planning and activities that manage data or information from an organization (Bairwa et al., 2014). Periodic vulnerability checking is very necessary, it serves to find out the state of the security of the system owned, so as to minimize all forms of threats that might occur. The output of this method is a report of security conditions of the web application that can be used as a parameter in improving the security system of the web application. The security report will greatly assist an organization or company in closing vulnerabilities contained in the object to be examined. Vulnerability Assessment methodology have 5 phase for implementation. Start from Reconnaissance, Vulnerability Scanning, Vulnerability Detection, Risk Assessment Value, and Reporting.

Figure 3: Vulnerability Assessment Methodology.

Figure 3 describes these phases of Vulnerability Assessment Methodology in this research which is illustrated in a flow diagram. The following is an explanation of each phase:

1. Reconnaissance
   This phase is the stage of gathering as much information as possible about the target object that will be carried out Vulnerability Assessment.
2. Vulnerability Scanning
   In this phase the Vulnerability Scanning process begins with various tools. In this research, the tools used for Vulnerability scanning are Paros and Vega.
3. Vulnerability Detection
   After completing vulnerability scanning, in this phase we can find out the vulnerability of the web application.
4. Risk Assessment Value
   At this stage is an assessment of the threat level of the vulnerability owned by system.

5. Reporting
   In this phase, the results of the examination of each tool will be presented together with a comparison.

In this research all phases of the Vulnerability Assessment methodology will be used in its implementation.

4 EXPERIMENTAL RESULTS AND ANALYSIS

The focus of this research is to find any vulnerabilities that may occur on Investment Department and Integrated Services One Door of XYZ Regency web application, so that later security recommendations can be given to improve the current web application system. The first step taken is to collect as much information as possible about the web application that will be the target of vulnerability assessment or we can also call it a security analysis on the web application. In this research, the security analysis of the Investment Department and Integrated Services One Door of XYZ Regency web application is three times the experiment with 3 different tools. The aim is to get more information about the vulnerabilities that exist in the web application. Tools used in this research are Nmap, Paros, and Vega. The Vulnerability Assessment process is carried out in a virtual environment with Linux times as the operating system.

4.1 Vulnerability Assessment with Nmap Tools

Nmap is a tool used to perform scan on ports (Bairwa, 2014). To do a port scan, we can use the IP address or hostname as a scope for checking. In this research, port inspection and analysis will be carried out on the One Door of XYZ Regency Investment and Integrated Services web application. This should be done to find out the vulnerability of existing ports in the web application.

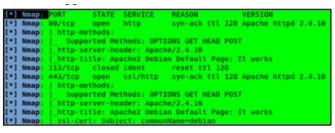

Figure 4: Port scan results.

Figure 4 is the result of port scanning on the Investment Department and Integrated Services One Door of XYZ Regency web application. From the results of these experiments, the following is information that can be obtained:

1. Port 80/tcp with http services has open port status.
2. Port 443/tcp with ssl/http services has open port status.
3. Support methods used are OPTION, GET, HEAD, and POST.
4. The web server used in the web application is Apache / 2.4.10 Debian.

4.2 Vulnerability Assessment with Paros Tools

Paros is an application that is installed by default on Kali Linux. Paros can be used to perform an overall security scan of web applications. In this research, a vulnerability assessment trial was carried out on the Investment Department and Integrated Services One Door of XYZ Regency web application, to determine the existing vulnerabilities.

Risk Level	Number of Alerts
High	1
Medium	4
Low	1
Informational	0

Figure 5: Summary of Alerts Paros Tools.

Figure 5 is a report about the vulnerability information found after scanning the Integrated Services One Door of XYZ Regency web application. The total number of vulnerability alerts on this web application is 6. The results of scanning the web application indicate that the security system is still insecure because there are still many weaknesses and some even have a high level of threat level.

Risk Level	Detail of Alerts
High	SQL Injection
Medium	Password Autocomplete in browser
	Secure page browser cache
	Cross site scripting
	Cross site scripting without brackets
Low	Obsolete file extended check

Figure 6: Detail of Alerts.

Figure 6 contains detailed information of each threat that exists in the web application. The results of the report can be used as parameters to minimize all forms of attack from hackers. In improving the security of web applications it is advisable to first improve the threat with the highest level.

4.3 Vulnerability Assessment with Vega Tools

Vega is a free tool designed to test security and find out web application vulnerabilities. Tools can be used

to check the vulnerability of web applications such as SQL Injection, Cross Site Scripting (XSS), Counterfeiting Cross Site Requests Forgery (CSRF), and many others (Babincev and Vuletić, 2016). The Tools Vega is very user friendly and very easy to use. In Kali Linux to run Vega there are two ways namely by typing the command in the terminal and directly open the application. In this study also conducted experiments on the application of web applications using Vega tools.

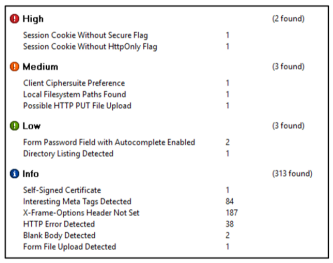

Figure 7: Vega Scanning Tool Reports.

Figure 7 is a report on the results of a web application security check on the Integrated Services One Door of XYZ Regency web application.

Reports from the results of the scan can be used as a benchmark to fix any gaps in the web application.

Risk Level	Number of Alerts
High	2
Medium	3
Low	3
Informational	313

Figure 8: Summary of Alerts Vega Tools.

Figure 8 describes the report on the web application security check on the Integrated Services One Door of XYZ Regency web application using the Vega tool. Vulnerability levels in the table are sorted by threat level with the highest level to the lowest.

4.4 Vulnerability Analysis

Based on the vulnerability assessment experiment that has been carried out with 3 tools, various information about the security conditions on the Integrated Services One Door of XYZ Regency web application has been obtained. This information can be used as material for analysis in correcting any existing weaknesses.

Risk Level	Number of Alerts
High	3
Medium	7
Low	4
Informational	313

Figure 9: Total Summary of Alerts.

Figure 9 lists the total reports of vulnerabilities based on threat level. From this report it can be seen that the security conditions of web applications are not good. That's because there are 3 security holes with the highest threat level.

Vulnerabilities	Nmap	Paros	Vega
SQL Injection		✓	
Session Cookie			✓
Cross Site Scripting		✓	
Open Port	✓		
Secure page browser cache		✓	
HTTP Put File Upload			✓
Server Identification	✓		

Figure 10: Comparative of Vulnerabilities Detected by Tools.

Figure 10 contains a comparison of the ability of tools to detect every possible vulnerability.

5 CONCLUSIONS

The conclusion of this research is that Investment Department and Integrated Services One Door of XYZ Regency web application still has many security weaknesses, so there are many loopholes that can be exploited by hackers to attack this web application. Vulnerability Assessment has a very important role in developing a network system. Without doing Vulnerability Assessment we will never know the weakness of the web application that we have in terms of security unless there is already a hacker who is attacking. In this research, it can be seen that scanning of web applications using different tools will produce different types of vulnerabilities. In this research Figure 5: Vega Scanning Tool Reports.

found 3 vulnerabilities with the highest threat level, 7 threats with medium level, and 4 threats with low level. This indicates that the Investment Department and Integrated Services One Door of XYZ Regency web application has a very big risk to be attacked by people who are not responsible or commonly called hackers. Therefore, it is recommended that the Web application immediately upgraded its security on SQL injection section, session cookie, password management, cross site scripting, client ciphersuite preference, local filesystem paths, and possible HTTP PUT file upload. This is done to minimize all possible threats related to weaknesses that are owned by the web application.

For further research it is recommended to use other methods and tools in conducting security audit experiments on a web application. It aims to compare the features and capabilities of each tool in conducting Vulnerability Assessment. This research can also be a reference for everyone in conducting a security audit of a web application, both in terms of methods or tools.

REFERENCES

Babincev, I. M. and Vuletić, D. V. (2016). Web application security analysis using the kali linux operating system. *Vojnotehnički glasnik*, 64(2):513–531.

Bairwa, S., Mewara, B., and Gajrani, J. (2014). Vulnerability scanners-a proactive approach to assess web application security. *arXiv preprint arXiv:1403.6955*.

Bau, J., Wang, F., Bursztein, E., Mutchler, P., and Mitchell, J. C. (2012). Vulnerability factors in new web applications: Audit tools, developer selection & languages. *Stanford, Tech. Rep*.

BAYKARA, M. (2018). Investigation and comparison of web application vulnerabilities test tools.

Charpentier Rojas, J. E. (2013). Web application security.

Eshete, B., Villafiorita, A., and Weldemariam, K. (2011). Early detection of security misconfiguration vulnerabilities in web applications. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 169–174. IEEE.

Gunawan, T. S., Lim, M. K., Zulkurnain, N. F., and Kartiwi, M. (2018). On the review and setup of security audit using kali linux. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(1):51–59.

Joshi, C. and Singh, U. K. (2016). Performance evaluation of web application security scanners for more effective defense. *International Journal of Scientific and Research Publications (IJSRP)*, 6(6):660–667.

M. Sevri, N. T. (2016). An infrastructure model to detect and prevent web attacks.

Mahmoud, S. K., Alfonse, M., Roushdy, M. I., and Salem, A.-B. M. (2017). A comparative analysis of cross site scripting (xss) detecting and defensive techniques. In *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 36–42. IEEE.

Mitropoulos, D., Louridas, P., Polychronakis, M., and Keromytis, A. D. (2017). Defending against web application attacks: approaches, challenges and implications. *IEEE Transactions on Dependable and Secure Computing*, 16(2):188–203.

Montieri, A., Ciuonzo, D., Bovenzi, G., Persico, V., and Pescapé, A. (2019). A dive into the dark web: Hierarchical traffic classification of anonymity tools. *IEEE Transactions on Network Science and Engineering*.

Nath, H. V. (2011). Vulnerability assessment methods–a review. In *International Conference on Network Security and Applications*, pages 1–10. Springer.

Qadir, S. and Quadri, S. (2016). Information availability: An insight into the most important attribute of information security. *Journal of Information Security*, 7(3):185–194.

Subedi, B., Alsadoon, A., Prasad, P., and Elchouemi, A. (2016). Secure paradigm for web application development. In *2016 15th RoEduNet Conference: Networking in Education and Research*, pages 1–6. IEEE.

Sumra, I. A., Hasbullah, H. B., and AbManan, J.-l. B. (2015). Attacks on security goals (confidentiality, integrity, availability) in vanet: a survey. In *Vehicular Ad-Hoc Networks for Smart Cities*, pages 51–61. Springer.

Tajpour, A., Ibrahim, S., and Masrom, M. (2011). Sql injection detection and prevention techniques. *International Journal of Advancements in Computing Technology*, 3(7):82–91.

Walden, J. (2008). Integrating web application security into the it curriculum. In *Proceedings of the 9th ACM SIGITE conference on Information technology education*, pages 187–192.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS