Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Analysis of Local Government Website Vulnerabilities Using the PTES Framework

Muh. Ariq Hasabi[1]*, Muhammad Azwar[2], Lilik Widyawati[3]

[1,2,3]Fakultas Teknik, Universitas Bumigora
muhammadariqhasabi@gmail.com [1*]

Abstract

This study aims to analyze and validate security vulnerabilities on the official website of a local government agency, which is a crucial public service portal. The study adopts an adapted Penetration Testing Execution Standard (PTES) methodology, focusing on non-invasive techniques to ensure ethical and responsible assessment of active government systems. Key stages include information gathering, vulnerability scanning using tools such as Nessus, and manual validation using Metasploit and SQLMap. Post-validation analysis confirmed several significant vulnerabilities, with the most critical findings being the exposure of development configuration files to the public and the presence of an outdated version of phpMyAdmin. The study also identified systemic issues such as weak cipher suite support (SWEET32) and configurations that enable DNS amplification attacks. The manual validation process critically succeeded in uncovering false positives from the automated scanner, highlighting the importance of verification by experts. This website exhibits significant security weaknesses due to inadequate patch management and insecure configurations. These findings underscore the urgent need for government agencies to adopt proactive security audits and structured remediation cycles to protect public data and maintain trust in digital services.

Keywords: *Cyber Security; Government Websites; PTES; phpMyAdmin; Web Vulnerabilities*

1. Introduction

In today's digital age, digital transformation has become a key pillar in the modernization of governments around the world. The Indonesian government, in its efforts to improve efficiency, transparency, and the reach of public services, has massively adopted digital technology [1]. Official government websites have evolved from mere information portals into crucial interactive platforms for citizens. However, this progress is a double-edged sword. On one hand, it brings convenience and efficiency; on the other, it opens the door to new and increasingly complex threats in the digital realm.

However, every step toward greater connectivity inherently opens up a new attack surface. Ease of access for citizens means potential ease of access for malicious actors. The accompanying escalation of cyber threats is not only in terms of quantity but also sophistication. Modern attacks are no longer limited to simple defacement but rather planned operations targeting sensitive data, aiming to paralyze services (ransomware) or even cyber espionage [2].

Digital transformation in the government sector has made official websites the frontline for delivering information and public services. However, this increase in connectivity has been accompanied by an escalation in increasingly sophisticated cyber threats. Indonesia's national cybersecurity context shows a significant level of threat. A report from the National Cyber and Cryptography Agency (BSSN) underscores this urgency, noting that throughout 2023, 403,990,813 cyber traffic anomalies were identified and 2,860 security vulnerabilities were found in 586 government electronic systems [3]. This data confirms that government digital assets are the primary targets of attacks, making security audits an urgent necessity.

At the local government level, this challenge becomes even more apparent. A local government agency manages the website [domain_target].go.id as a vital channel for public services. Based on initial observations, there has never been any documented penetration testing of the site. The absence of security assessments creates a gap between the national threat landscape and the actual security posture at the local level. This phenomenon is not an isolated case; various studies have reported similar vulnerabilities on other government websites in Indonesia, indicating a pattern of systemic weaknesses [4][5][6].

To address this gap, a systematic approach is needed. This study adopts the Penetration Testing Execution Standard (PTES) framework, a globally recognized methodology for conducting penetration tests [7][8][9]. Choosing the Penetration Testing Execution Standard (PTES) as a framework is a sound methodological decision. PTES is not simply trying to hack but rather a systematic approach that ethically mimics the work of professional hackers. This ensures that testing is comprehensive, no important steps are overlooked, and the results are accountable. By applying PTES, this study aims to identify and analyze vulnerabilities on the agency's website, validate the exploitability of vulnerabilities in a non-invasive manner, and formulate technical recommendations for risk mitigation. The ultimate goal of this research is not simply to compile a list of vulnerabilities, but to generate actionable recommendations.

2. Research Method

This study uses a qualitative approach with a case study method that focuses on in-depth technical analysis. The framework used is an adaptation of the Penetration Testing Execution Standard (PTES) to ensure a structured and ethical testing process [10]. Given that the research object is an active government system, the intrusive Exploitation and Post-Exploitation stages were modified into non-invasive validation phases to confirm potential exploitation without causing disruption.

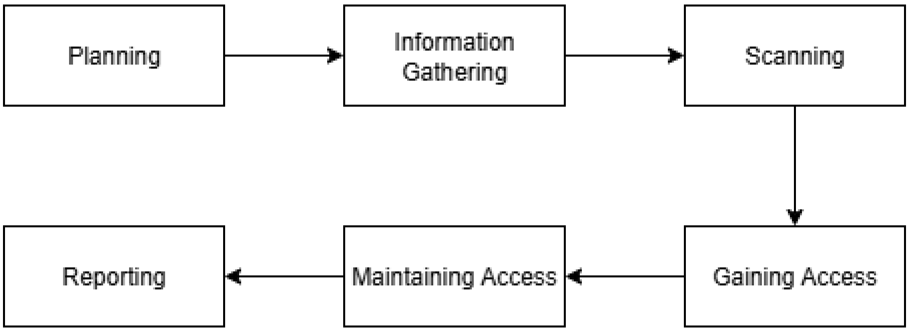


Fig. 1: Stages in the PTES method

The adaptation of the PTES framework is then implemented through five main stages, as illustrated in the figure above. The process begins with the planning stage to establish a clear scope and boundaries, followed by information gathering and scanning to map and identify potential security gaps. In accordance with the modifications described, the Gaining Access stage does not involve destructive exploitation but focuses on validating vulnerabilities through safe and controlled Proof of Concept (PoC). By eliminating the Maintaining Access phase, this testing cycle concludes directly with the Reporting phase, where all findings and validation evidence are documented in detail along with recommendations for improvements. This five-phase approach ensures that security testing can be conducted comprehensively and ethically without posing risks to the operational stability of active government systems.

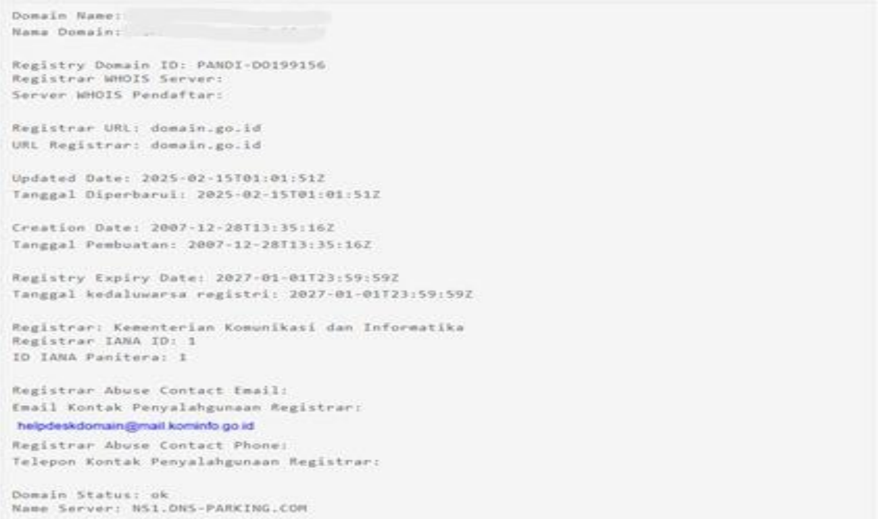
2.1. Planning

The planning phase serves as the ethical and legal foundation for the entire security testing process. During this critical phase, initial interactions are not merely formalities but strategic dialogues with relevant government agencies. The objective is to align perceptions and build trust, ensuring that the testing team and stakeholders (such as system administrators and security officers) share a common understanding.

The scope of testing is clearly defined and strictly limited to the [domain_target].go.id domain and its directly related assets. This clarification is essential to prevent scope creep or unintended testing of other government systems. Furthermore, testing limitations are established as key rules. Commitment to a non-invasive approach is a top priority. This means that the testing team is explicitly prohibited from performing active exploitation that could alter, delete, or damage system data and configurations. Potentially disruptive attacks, such as Denial of Service (DoS) or stress testing, are also strictly prohibited. All these agreements are then formalized in a document (Rules of Engagement) that serves as the operational foundation to ensure the testing process is conducted safely, controlled, and does not disrupt ongoing public services.

2.2. Information Gathering

The information-gathering stage is conducted passively using Open-Source Intelligence (OSINT) techniques to map the attack surface without directly interacting with the target infrastructure. This approach utilizes publicly available data to build an initial understanding of the agency's digital assets. One of the fundamental tools used in this phase is WHOIS, which serves to obtain administrative and technical data related to domain registration.



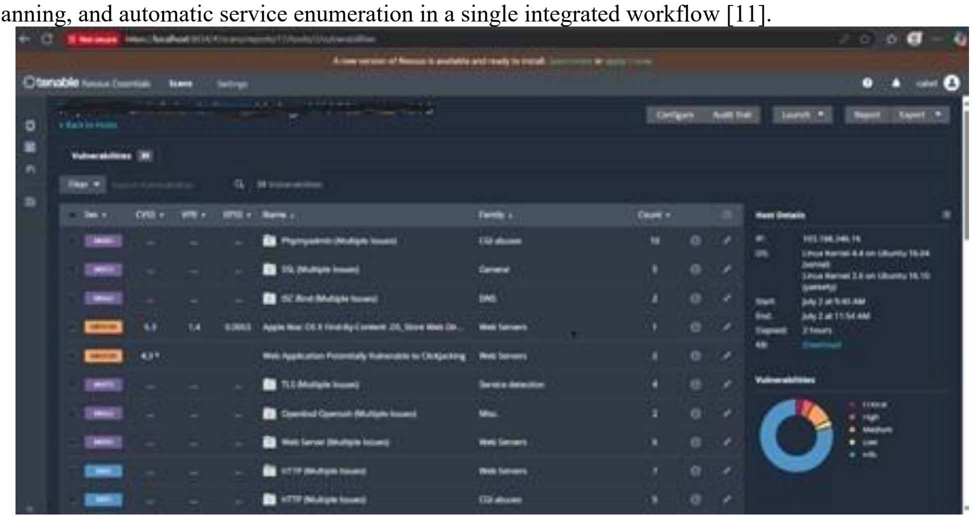
```
Domain Name: [redacted]
Nama Domain:

Registry Domain ID: PANDI-DO199156
Registrar WHOIS Server:
Server WHOIS Pendaftar:

Registrar URL: domain.go.id
URL Registrar: domain.go.id

Updated Date: 2025-02-15T01:01:51Z
Tanggal Diperbarui: 2025-02-15T01:01:51Z

Creation Date: 2007-12-28T13:35:16Z
Tanggal Pembuatan: 2007-12-28T13:35:16Z

Registry Expiry Date: 2027-01-01T23:59:59Z
Tanggal kedaluwarsa registri: 2027-01-01T23:59:59Z

Registrar: Kementerian Komunikasi dan Informatika
Registrar IANA ID: 1
ID IANA Panitera: 1

Registrar Abuse Contact Email:
Email Kontak Penyalahgunaan Registrar:
 helpdeskdomain@mail.kominfo.go.id
Registrar Abuse Contact Phone:
Telepon Kontak Penyalahgunaan Registrar:

Domain Status: ok
Name Server: NS1.DNS-PARKING.COM
```

Fig. 2: Results of the whois query

From the WHOIS query results for the domain [domain_target].go.id as shown in the image above, several key pieces of information were obtained. It was revealed that this domain is officially registered under the Ministry of Communication and Information Technology, confirming its status as a legitimate government asset. Historical data shows that this domain was created on December 28, 2007, indicating that it is a long-standing domain.

2.3. Scanning

Active scanning is performed to identify potential technical and comprehensive vulnerabilities. For this phase, the scanning process uses the Nessus Essentials tool. This tool was chosen because its capabilities are not limited to vulnerability identification but also include host discovery, port scanning, and automatic service enumeration in a single integrated workflow [11].



Sev	CVSS	VPR	EPSS	Name	Family	Count
MIXED	–	–	–	Phpmyadmin (Multiple Issues)	CGI abuses	10
MIXED	–	–	–	SSL (Multiple Issues)	General	5
MIXED	–	–	–	ISC Bind (Multiple Issues)	DNS	2
MEDIUM	5.3	1.4	0.0053	Apple Mac OS X Find-By-Content .DS_Store Web Dir...	Web Servers	1
MEDIUM	4.3 *			Web Application Potentially Vulnerable to Clickjacking	Web Servers	2
MIXED	–	–	–	TLS (Multiple Issues)	Service detection	4
MIXED	–	–	–	OpenBSD OpenSSH (Multiple Issues)	Misc.	2
MIXED	–	–	–	Web Server (Multiple Issues)	Web Servers	9
INFO	–	–	–	HTTP (Multiple Issues)	Web Servers	7
INFO	–	–	–	HTTP (Multiple Issues)	CGI abuses	5

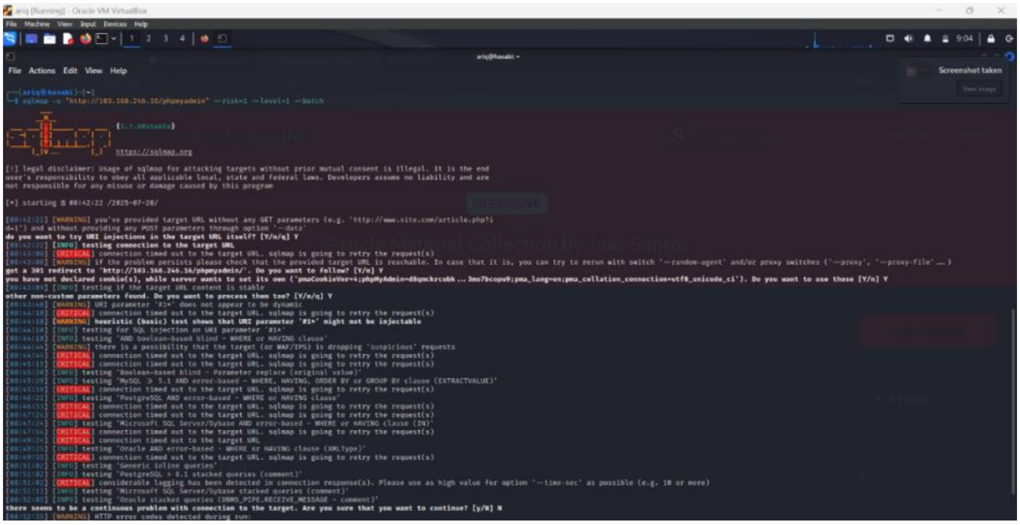
IP: 103.168.246.16
OS: Linux Kernel 4.4 on Ubuntu 16.04 (xenial)
Linux Kernel 3.x on Ubuntu 16.10 (yakkety)
Start: July 2 at 9:40 AM
End: July 2 at 11:54 AM
Elapsed: 2 hours
KB: Download

Fig. 3: Results of the Nessus scan

The results of the scanning conducted by Nessus Essentials successfully identified several important findings. Quantitatively, the scan found 2 vulnerabilities with a critical risk level, 2 vulnerabilities with a high risk level, and 5 vulnerabilities with a medium risk level. The presence of findings at the critical and high levels indicates significant security gaps. Some specific examples of identified findings include issues with phpMyAdmin (multiple issues), potential vulnerabilities to clickjacking attacks on web applications, and several weaknesses in SSL/TLS configuration. The list of identified vulnerabilities serves as the basis for further verification and validation in the next phase.

2.4. Gaining Access

This stage is a verification phase that aims to validate findings from automated scanners without gaining illegal access or causing disruption. This approach is essential to ensure that every reported vulnerability is genuine and not a false positive, while still complying with agreed non-invasive testing restrictions. Validation is conducted carefully using the Metasploit Framework (specifically the auxiliary module) to verify weaknesses in SSL/TLS versions and SQLMap with safe parameters (--risk=1 --level=1) to detect potential SQL Injection vulnerabilities. The use of the --risk=1 and --level=1 parameters in SQLMap ensures that the testing only uses the safest payloads and will not attempt techniques that could potentially damage the database. This validation process specifically targets the phpMyAdmin interface, which was previously flagged by Nessus as high risk.



ariq [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
9:04
ariq@hasabi: ~
File Actions Edit View Help
Screenshot taken
View image

```
┌──(ariq㉿hasabi)-[~]
└─$ sqlmap -u "http://103.168.246.16/phpmyadmin" --risk=1 --level=1 --batch

        {1.7.8#stable}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end
user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are
not responsible for any misuse or damage caused by this program

[*] starting @ 08:42:22 /2025-07-28/

[08:42:22] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?i
d=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] Y
[08:42:22] [INFO] testing connection to the target URL
[08:43:00] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:43:00] [WARNING] if the problem persists please check that the provided target URL is reachable. In case that it is, you can try to rerun with switch '--random-agent' and/or proxy switches ('--proxy', '--proxy-file' ...)
got a 301 redirect to 'http://103.168.246.16/phpmyadmin/'. Do you want to follow? [Y/n] Y
you have not declared cookie(s), while server wants to set its own ('pmaCookieVer=4;phpMyAdmin=d8qmckrcubk...3mo7bcopu9;pma_lang=en;pma_collation_connection=utf8_unicode_ci'). Do you want to use those [Y/n] Y
[08:43:09] [INFO] testing if the target URL content is stable
other non-custom parameters found. Do you want to process them too? [Y/n/q] Y
[08:43:48] [WARNING] URI parameter '#1*' does not appear to be dynamic
[08:44:18] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:44:18] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[08:44:18] [INFO] testing for SQL injection on URI parameter '#1*'
[08:44:18] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[08:44:44] [WARNING] there is a possibility that the target (or WAF/IPS) is dropping 'suspicious' requests
[08:44:44] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:45:15] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:45:20] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[08:45:29] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[08:45:59] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:46:22] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[08:46:53] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:47:24] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:47:24] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[08:47:54] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:49:24] [CRITICAL] connection timed out to the target URL
[08:49:25] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[08:49:55] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[08:51:02] [INFO] testing 'Generic inline queries'
[08:51:02] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[08:51:02] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for option '--time-sec' as possible (e.g. 10 or more)
[08:51:12] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[08:52:05] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
there seems to be a continuous problem with connection to the target. Are you sure that you want to continue? [y/N] N
[08:52:35] [WARNING] HTTP error codes detected during run:
```

Fig. 4: Results from the sqlmap test

As shown in Figure 4 above, the validation process using SQLMap was run according to secure parameters. However, the test results showed that the tool experienced continuous connection problems (connection timed out) to the target URL. Although this test did not definitively confirm the existence of an SQL injection vulnerability due to connectivity issues, this finding itself is an important note. The instability of the service when receiving light requests from the testing tool may indicate issues with the server, network configuration, or the presence of overly aggressive protection systems (such as IPS/WAF). All observation and validation results, both successful and blocked, will be fully documented in the reporting phase.

2.5. Reporting

The reporting stage is the final and most crucial phase in a penetration testing cycle, where all data, findings, and analysis from the testing are compiled into a formal document. The main objective is to clearly communicate the security status of the system to the system owner, both to management and the technical team. This report serves as a bridge between technical findings and concrete corrective actions.

3. Result and Discussion

3.1. Post-validation report

Each potential vulnerability is manually tested using a documented set of tools and techniques to confirm its existence and assess the actual risk level. Table 1 summarizes each finding, the validation method used, the final status of the verification, and the adjusted risk level.

Table 1: Summary of vulnerability findings

Vulnerability Type	Location/Target	Payload	Validation Status	Risk Level
Directory Listing	Metasploit dir_listing	Metasploit Framework	Not Confirmed	Low
phpMyAdmin 4.x < 4.8.5	http://103.168.246.16/phpmyadmin	SQLMap --risk=1 --level=1 --batch	Not Confirmed (403 Forbidden)	Medium
TLS 1.0 Protocol Detection	Nessus+ Metasploit	Nessus + Metasploit ssl_version	False Positive (TLS 1.2 is used)	Low
Development Configuration Files	/etc/, /CSCOT/, /git/	Dirsearch enumeration tool	Confirmed	High
TLS 1.1 Protocol Detection	https://103.168.246.16	Metasploit TLS fingerprint	Not Validated (TLS 1.2 is used)	Low
SSL Certificate Expiry	bakesbangpol.lomboktengahkab.go.id	Metasploit SSL Certificate Module	Not Relevant (other domain)	Unclassified
SWEET32 (Weak Cipher)	Port 443 (HTTPS)	Metasploit SSL scan (blocked)	Not Validated (Blocked Connection)	Low
DNS Amplification DDoS	103.168.246.16 (DNS)	Metasploit dns_amp module	Confirmed	Medium

3.2. Mitigation recommendations

Based on the results of the security scan and validation that has been carried out, several vulnerabilities were found in the target system. These findings have been classified according to their level of risk to facilitate the repair process.

In summary, there is one critical high-risk vulnerability, namely the exposure of development configuration files that are accessible to the public. In addition, there are two medium-risk vulnerabilities related to DNS server misuse and an outdated version of phpMyAdmin. The remaining findings are low-risk and are recommendations for general system hardening. Table 2 provides details of each vulnerability along with mitigation recommendations.

Table 2: Vulnerability mitigation recommendations

Vulnerability	Mitigation Recommendations	Priority	Category
Development Configuration Files	Sensitive configuration files such as /git/ are exposed. Immediately remove them from public access and audit potential data leaks to prevent system compromise.	High	Critical Vulnerability
DNS Amplification DDoS	DNS servers can be used to attack other targets. Disable open recursion so that the server only serves trusted networks.	Medium	Risk of Abuse
Old Version of phpMyAdmin	An old, vulnerable version of phpMyAdmin has been found. Please update to the latest version immediately, or remove it if it is not in use to close the vulnerability.	Medium	Security Vulnerability
Directory Listing	This feature can expose the file structure of the server. Disable directory listing in the web server configuration as a standard security practice.	Low	Information Leakage
SWEET32 (Weak Cipher)	The server may support outdated and weak encryption. Update the SSL/TLS configuration to use only strong, modern ciphers.	Low	Weak Encryption
TLS 1.0 & 1.1 Detection	Potential use of insecure TLS 1.0/1.1 protocols detected. Ensure that the server only enables TLS 1.2 and 1.3.	Low	Obsolete Protocol

4. Conclusion

This study successfully applied an ethically adapted PTES framework to analyze the security of a local government agency's website. The test results revealed a number of significant vulnerabilities, with the most critical finding being the exposure of sensitive development configuration files to the public. The main contribution of this research is the demonstration of the practical value of structured testing methodologies on government digital assets and the emphasis on the crucial role of manual validation to avoid false positives. These findings underscore that cybersecurity is an ongoing process that requires proactive assessment and timely remediation. The implementation of the proposed recommendations is expected to strengthen the security of the relevant agency's website and serve as a model for other government agencies.

References

[1] S. W. Gusman, “Development of the Indonesian Government’s Digital Transformation,” *Dinasti Int. J. Educ. Manag. Soc. Sci.*, vol. 5, no. 5, pp. 1128–1141, 2024, doi: 10.38035/dijemss.v5i5.2868.

[2] Ashish Dewakar Pandey and Shakil Saiyad, “Emerging Threats in Cybersecurity : A Deep Analysis of Modern Attack,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 10, no. 2, pp. 693–697, 2024, doi: 10.32628/cseit2410297.

[3] BADAN SIBER DAN SANDI NEGARA RI, “Laporan Keamanan Siber Indonesia (Bssn),” 2023. [Online]. Available: https://csirt.kemenpora.go.id/wp-content/uploads/2025/02/keamanan.pdf

[4] S. W. Ningsih, “Analisis Pengujian Kerentanan Situs Pemerintahan XYZ dengan PTES,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 3, pp. 1543–1556, 2021, doi: 10.35957/jatisi.v8i3.1224.

[5] E. Z. Darojat, E. Sediyono, and I. Sembiring, “Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner,” *J. Sist. Inf. Bisnis*, vol. 12, no. 1, pp. 36–44, 2022, doi: 10.21456/vol12iss1pp36-44.

[6] Z. Faizi, Puwantori, and A. Ali Ridha, “Analisis Web Security Hole Menggunakan Metode Penetration Testing Execution and Standard (Studi Kasus : Universitas Singaperbangsa Karawang),” *J. Inf. dan Komput.*, vol. 11, no. 2, p. 2023, 2023.

[7] M. Tahir and M. Risky, “Analisis Keamanan Website Dinas Pemerintahan Yogyakarta Dengan Metode PTES (Penetration Testing Execution Standard),” *J. Tek. Inform. UNIKA ST.Thomas*, vol. 9, pp. 2657–1501, 2024, [Online]. Available: https://ejournal.ust.ac.id/index.php/JTIUST/article/view/3334

[8] D. A. Andhika, Slamet, and N. Ningsih, “Pengujian Penetrasi pada Windows 10 menggunakan Model Penetration Testing Execution Standard (PTES),” *J. Technol. Informatics*, vol. 3, no. 2, pp. 55–61, 2022, doi: 10.37802/joti.v3i2.222.

[9] M. Noval, R. Darmawan, Y. Muhyidin, and D. Singasatia, “Analisis Keamanan Web Sman 1 Wanayasa Menggunakan Metode Pentration Testing Execution Standard (Ptes),” vol. 2, pp. 110–121, 2024.

[10] Muhammad Risky Ardiansyah *et al.*, “Analisis Kerentanan Keamanan Website Menggunakan Metode PTES (Penetration Testing Execution And Standart),” *Nuansa Inform.*, vol. 18, no. 2, pp. 145–153, 2024, doi: 10.25134/ilkom.v18i2.119.

[11] M. Muin, K. Kapti, and T. Yusnanto, “Campus Website Security Vulnerability Analysis Using Nessus,” *Int. J. Comput. Inf. Syst.*, vol. 3, pp. 79–82, 2022, doi: 10.29040/ijcis.v3i2.72.