Analysis and Enhancement of Website Security using Anti-CSRF Token, CSP, and Anti-Clickjacking Approaches

Muhammad Arif Putra Wibowo
Departement of Information System
Universitas Ahmad Dahlan
Yogyakarta of Indonesia

Imam Riadi
Departement of Information System
Universitas Ahmad Dahlan
Yogyakarta of Indonesia

ABSTRACT

Information security is a crucial aspect for any organization or company, whether private or governmental, as exemplified by the website lekamandiri.web.id. As part of a private company entity, PT Leka Mandiri utilizes a website application to manage financial data. This website handles sensitive client information. This research was conducted to identify and address security vulnerabilities on the lekamandiri.web.id website against CSRF attacks, Clickjacking, and Content Security Policy (CSP) violations. In light of these vulnerabilities, a security evaluation is required to provide proper mitigation and improvements for the lekamandiri.web.id website. This study employed the Penetration Testing method to identify and exploit existing security gaps on the PT Leka Mandiri Cikarang website. The testing phases included Information Gathering, Planning Analysis, Vulnerability Detection, Penetration Testing, Maintenance (Patching), and Reporting. Several tools were used during the testing process, such as OWASP ZAP for vulnerability scanning, Nmap and Whois for information gathering, Burp Suite for exploitation testing, and Visual Studio Code for maintenance (patching). The results of the study showed that the website had several vulnerabilities with varying risk levels. The initial testing indicated potential vulnerabilities to CSRF attacks, Clickjacking, and CSP violations. After remediation efforts, these three vulnerabilities were successfully mitigated through the implementation of Anti-CSRF tokens, the configuration of CSP headers, and the addition of Anti-Clickjacking headers. Overall, the implemented security measures proved to be effective in enhancing the website's security level, as evidenced by the improved security score from the retesting process.

Keyword

Information Security, Anti-CSRF Token, Content Security Policy (CSP), Anti-Clickjacking.

1. INTRODUCTION

In today's digital era, information has become a crucial asset for individuals, organizations, and nations. While advancements in information and communication technology have facilitated rapid data exchange, they also bring risks to data confidentiality, integrity, and availability. Threats like hacking, data theft, malware, and cyberattacks pose significant challenges. The rise of social media has increased concerns about data privacy, as it often becomes a source of confidential data leaks [1]. Technological progress, including internet growth [2], has expanded online access in Indonesia [3], supported by TCP/IP-based connectivity [4]. Alongside advances in hardware and software, especially web-based applications, information exchange has become more efficient [5]. Penetration testing plays a vital role in evaluating system security by simulating unauthorized attacks [6], addressing technical, administrative, legal, and political dimensions [6]. Web applications are frequent attack targets due to their rapid growth [7]. This study used penetration testing to identify vulnerabilities in the web application lekamandiri.web.id. Findings revealed weaknesses in the Anti-CSRF token, CSP implementation, and Anti-Clickjacking Header, increasing the site's exposure to attacks. The main objective is to fix flaws in the Anti-CSRF mechanism, improve CSP settings, and implement proper Anti-Clickjacking protections. Given the sensitivity of the data managed by lekamandiri.web.id, these vulnerabilities pose serious risks, including data breaches, service disruptions, and reputational damage. Therefore, this research aims to conduct a thorough security assessment, evaluate the severity of vulnerabilities, and recommend effective mitigation strategies to improve the website's security and user trust.

2. STUDY LITERATURE

2.1 Information System Concept

2.1.1 System

A system is a collection of elements that interact and relate to each other to achieve a specific goal. It consists of two or more interconnected components that work together to accomplish an objective [9]. Some systems are made up of smaller subsystems that support the larger system [8].

2.1.2 Information

Information is data or facts that have been processed in such a way that they transform into meaningful information. Additionally, information can reduce uncertainty and holds value in decision-making because, with information, we can choose actions that carry the least risk [10]. According to [11], information is defined as "a set of data that is temporary, time-dependent, and capable of delivering surprises or unexpected elements to its recipient. The identity and duration of these surprises are referred to as the value of information." Information is also temporary and possesses several characteristics such as relevance, accuracy, timeliness, completeness, and reliability.

2.1.3 Information System

A system consists of five interconnected components that collect, process, store, and distribute data to support organizational control and decision-making [12]. Information systems help managers and employees in coordination, problem-solving, and product development, while also processing daily transactions and generating reports to support operational activities [13].

2.1.4 Information Security

Data security aims to protect against various threats to ensure business continuity, reduce risks, and enhance investment and business opportunities. It involves preventing and detecting data theft, software alterations, or physical damage to information systems, all of which can lead to business losses. Data security focuses on three main aspects.

(a) Secrecy alludes to the measures taken to guarantee that data is open as it were to authorized parties [12].

(b) Keenness relates to keeping up the genuineness and validity of information. In this setting, astuteness includes activities pointed at anticipating unauthorized alteration, harm, or modification of information [14].

(c) Accessibility alludes to the measures taken to guarantee that data is available as it were to authorized parties [15].

2.2 Website

2.2.1 Website Explanation

A website is a digital platform consisting of one or more web pages accessible online, commonly used for information access and communication within specific communities or organizations [16]. Its importance lies in its cross-platform accessibility.

2.2.2 Website Attacks

Site attacks are actions by individuals or groups that exploit vulnerabilities in websites or servers. Many websites, including those of businesses, governments, and institutions, store sensitive data but often lack adequate protection [17]. These attacks can alter content, disrupt operations, deny access, degrade performance, or damage files [18]. Web-based applications are frequent targets for hackers, with varying motivations but potentially serious consequences for organizations relying on their websites [18].

2.2.3 Database Attacks

The database is a crucial component of a website system, serving as a storage for essential data that supports web application operations. Ensuring database security is vital to maintaining data integrity, confidentiality, and availability, especially since it holds sensitive information like user identities and financial records. Databases are common targets for cyberattacks such as SQL injection, data exfiltration, and malware. SQL injection, a prevalent method, exploits input validation flaws in web applications to execute malicious SQL commands [19]. Therefore, securing databases is essential to protect an organization’s information assets.

2.3 Security Vulnerability Analysis

This process aims to identify security vulnerabilities in networks to minimize threats like unauthorized data breaches. The author used penetration testing (pentest), which simulates attacks on systems, web applications, and networks to detect exploitable weaknesses [20].

2.4 OWASP

The Open Web Application Security Project (OWASP) is a global community dedicated to improving software security by providing free, impartial resources such as guidelines, tools, and documentation to help developers and organizations identify and address web application vulnerabilities. OWASP also establishes security testing standards, highlighting the importance of security in information systems [21].

2.5 OWASP Zap

OWASP ZAP is an open-source vulnerability scanner developed by OWASP, continuously updated and widely used for penetration testing [22]. It detects issues like SQL injection, Cross-Site Scripting (XSS), private IP disclosure, and application error exposure. Designed for easy identification of security flaws during web application testing, OWASP ZAP is user-friendly, can generate reports, and is freely available [23].

3. METHODOLOGY

3.1 Data Collection Method

The information collection strategies utilized in this consider incorporate different strategies planned to efficiently, precisely, and pertinently get data concurring to the investigate needs. These methods incorporate

3.1.1 Vulnerability Assesment

Vulnerability Assessment is a thorough evaluation of various security aspects to identify potential critical vulnerabilities [24]. This includes information security, system configurations, and physical security. In this stage, OWASP ZAP is used to automatically scan website security, focusing on vulnerabilities like Anti-CSRF, CSP, and Anti-clickjacking. Detected issues are then analyzed for their severity through risk analysis.

3.1.2 Literature Study

Collecting information from various sources about information security, identified attacks, and security testing methodologies.

3.1.3 Interview

Conducting interviews with the site chairmen to get it the security approaches executed and the challenges confronted in keeping up the website's security.

3.1.4 Observation

The method used involves collecting data by observing and recording through direct visits. In this study, observation was conducted to obtain information about the information system services at PT Leka Mandiri. The observation took place in the IT department by asking questions about the information system service processes within the organization. The questions asked served as the basis for observation related to the information system services in the respective institution.

3.2 Research Stage

This stage begins with a literature review and continues through to the analysis of the test results. These stages can be seen in Figure 1

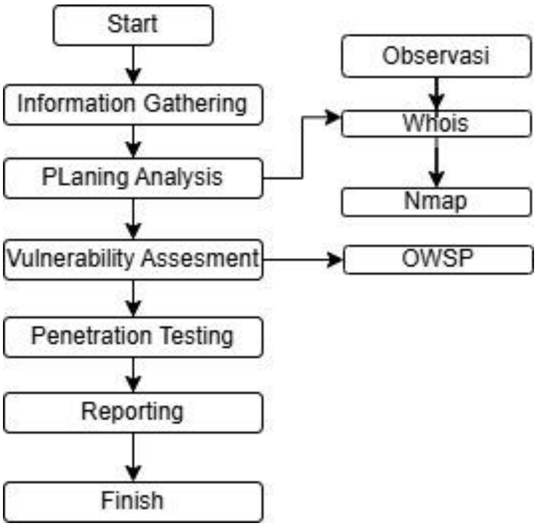


Figure 1. Research Stage

Figure 1 outlines a arrangement of stages pointed at distinguishing security vulnerabilities. These stages start with

Data Gathering and conclude with Announcing.

3.2.1 Information Gathering

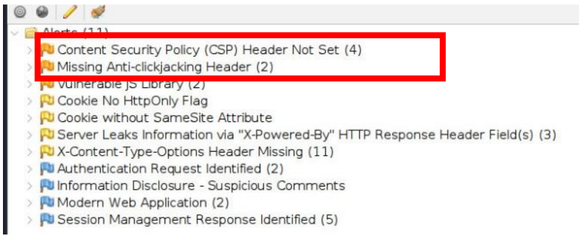
At this stage, key information about PT Lekamandiri web application infrastructure and potential vulnerabilities was gathered. This included direct observation of the system and its security policies, as well as the use of the Whois tool to collect domain data for lekamandiri.web.id, such as registrant details and registration dates. Additionally, NMAP scanning was performed to identify open ports and active services, helping to detect security gaps and determine the next steps in the testing process.

3.2.2 Planning Analysis

At this stage, planning and analysis are conducted to identify potential vulnerabilities based on previously collected data. This involves reviewing network configurations, active services, and open port risks. A testing strategy is then developed, including attack simulations and the use of tools like OWASP ZAP for comprehensive vulnerability scanning.

3.2.3 Vulnerability Detection

This phase identifies potential vulnerabilities in the web application using OWASP ZAP, focusing on critical aspects such as Anti-CSRF Token, CSP, and Anti-Clickjacking headers. The initial scan results are shown in Figure 2.



Alerts (11)
- Content Security Policy (CSP) Header Not Set (4)
- Missing Anti-clickjacking Header (2)
- Vulnerable JS Library (2)
- Cookie No HttpOnly Flag
- Cookie without SameSite Attribute
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (3)
- X-Content-Type-Options Header Missing (11)
- Authentication Request Identified (2)
- Information Disclosure - Suspicious Comments
- Modern Web Application (2)
- Session Management Response Identified (5)

Figure 2. Scan result using Zap

Figure 2 shows the results of the first scan using OWASP ZAP, where alerts were found with details of 4 medium-risk, 2 low-risk, and 6 informational alerts. These details can be seen in Table 1.

Table 1 OWASP ZAP Vulnerability List

No	*Vulnerability*	*Level Risk*
1	*Absence of Anti-CSRF Tokens*	*Medium*
2	*Content Security Policy (CSP) Header Not set*	*Medium*
3	*Missing Anti-Clickjacking Header*	*Medium*
4	*Vulnerbility JS Library*	*Medium*
5	*Strict-Transport-Security header Not Set*	*Low*
6	*X-Content-Type-Option Header Missing*	*Low*
7	*Content-Type Header Missing*	*Infromational*
8	*Information Disclosure-Suspicious Comments*	*Infromational*
9	*Modern Web Aplication*	*Informational*
10	*Re-examine Cache-control Directives*	*Informational*
11	*User Agent Fuzzer*	*Informational*
12	*User Controllabel HTML Element Attribute (Potential XXS*	*Infromational*

Table 1 describes the results of the scanning process using the OWASP ZAP tool, which are classified into two categories: Vulnerabilities and Risk Levels.

3.2.4 Penetration Testing

Penetration testing involves assessing IT assets to find security vulnerabilities that attackers might exploit, using automated tools or manual methods [25]. This method effectively uncovers web application security gaps by exploiting identified weaknesses to evaluate their impact. The results are then analyzed to determine vulnerability severity and guide remediation efforts.

3.2.5 Reporting

At the reporting and security enhancement stage, all test results are compiled into a detailed report that includes findings, risk analysis, and security improvement recommendations. The report details vulnerabilities, proof of exploits, and risk assessments, along with suggested measures such as implementing or improving the Anti-CSRF token, optimizing CSP settings, and applying the Anti-Clickjacking header.

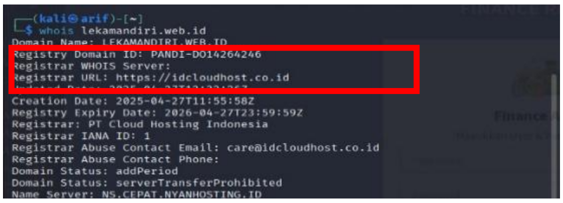
4. RESULTS AND DISCUSSION

4.1 Results

4.1.1 Information Gathering

This initial research stage involves collecting data on the infrastructure, services, and potential vulnerabilities of lekamandiri.web.id using Whois and Nmap tools.

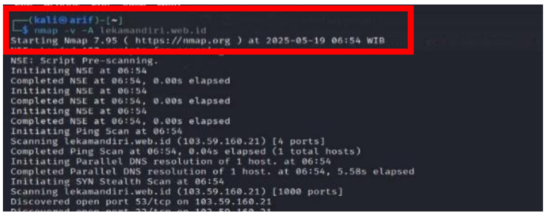
(a) Whois is an information-gathering tool for lekamandiri.web.id, used to retrieve domain, IP, and system data to protect intellectual property, prevent domain misuse, and enhance internet security. The scan results are shown in Figure 3.



```
┌──(kali㉿arif)-[~]
└─$ whois lekamandiri.web.id
Domain Name: LEKAMANDIRI.WEB.ID
Registry Domain ID: PANDI-DO14264246
Registrar WHOIS Server:
Registrar URL: https://idcloudhost.co.id
Creation Date: 2025-04-27T11:55:58Z
Registry Expiry Date: 2026-04-27T23:59:59Z
Registrar: PT Cloud Hosting Indonesia
Registrar IANA ID: 1
Registrar Abuse Contact Email: care@idcloudhost.co.id
Registrar Abuse Contact Phone:
Domain Status: addPeriod
Domain Status: serverTransferProhibited
Name Server: NS.CEPAT.NYANHOSTING.ID
```

Figure 3. Whois Results

(b) The next step in data gathering involves using the Nmap tool for network scanning, specifically for port scanning on the lekamandiri.web.id website. The results of this data gathering are shown in Figure 4.



```
┌──(kali㉿arif)-[~]
└─$ nmap -v -A lekamandiri.web.id
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-19 06:54 WIB
NSE: Script Pre-scanning.
Initiating NSE at 06:54
Completed NSE at 06:54, 0.00s elapsed
Initiating NSE at 06:54
Completed NSE at 06:54, 0.00s elapsed
Initiating NSE at 06:54
Completed NSE at 06:54, 0.00s elapsed
Initiating Ping Scan at 06:54
Scanning lekamandiri.web.id (103.59.160.21) [4 ports]
Completed Ping Scan at 06:54, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:54
Completed Parallel DNS resolution of 1 host. at 06:54, 5.58s elapsed
Initiating SYN Stealth Scan at 06:54
Scanning lekamandiri.web.id (103.59.160.21) [1000 ports]
Discovered open port 53/tcp on 103.59.160.21
```

Figure 4. Nmap Result

(c) A comparison of WHOIS and Nmap results based on filtering is presented in Table 2

Table 2. Comparison of WHOIS and Nmap

Aspek	*WHOIS*	*Nmap*
Information Domain	Provides details about domain registration	None
Ports and Services	None	Provides a list of open ports and services
Domain Security	Identifying potential problems in domain management	None
Network Analysis	None	Provides operating system, and service data

4.1.2 Planning Analysis

The following arrange is Arranging Investigation or the arranging stage. In this stage, the information collected amid the Data Gathering arrange is analyzed to distinguish potential vulnerabilities that might be abused. An assessment is conducted on the organize arrangement, recognizable proof of running administrations, and chance appraisal of open ports. The flow of the arranging investigation arrange is outlined within the flowchart appeared in Figure 5.

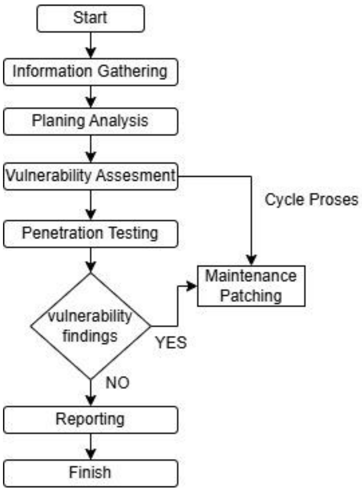
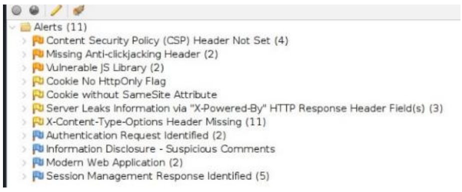


Figure 5. Flowchart Planning Analysis

4.1.3 Vulnerability Assesment

During the vulnerability detection stage, a list of vulnerabilities was compiled from scanning results using the OWASP ZAP 2.16.1 application. This stage was divided into two phases: before and after remediation. The results of the vulnerability detection can be seen in Figure 6.



```
Alerts (11)
  Content Security Policy (CSP) Header Not Set (4)
  Missing Anti-clickjacking Header (2)
  Vulnerable JS Library (2)
  Cookie No HttpOnly Flag
  Cookie without SameSite Attribute
  Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (3)
  X-Content-Type-Options Header Missing (11)
  Authentication Request Identified (2)
  Information Disclosure - Suspicious Comments
  Modern Web Application (2)
  Session Management Response Identified (5)
```

Figure 6. OWASP First Scan

Figure 6 shows the OWASP scan results of PT. Leka Mandiri's website (https://lekamandiri.web.id/), which took approximately 8 minutes and identified 12 vulnerabilities, including 4 medium-risk alerts, 2 low-risk alerts, and 6 informational alerts.

(a) Absence of Anti-CSRF Tokens. The PT. Leka Mandiri Cikarang website's HTML forms lack Anti-CSRF tokens, exposing them to CSRF attacks. Implementing Anti-CSRF tokens on all authenticated forms is strongly recommended to enhance security.

(b) Content Security Policy (CSP) Header Not set. PT. Leka Mandiri Cikarang's website lacks a Content Security Policy (CSP) header, increasing the risk of attacks like reflected Cross-Site Scripting (XSS). Implementing CSP restricts resource loading to trusted sources, enhancing protection against content-based attacks.

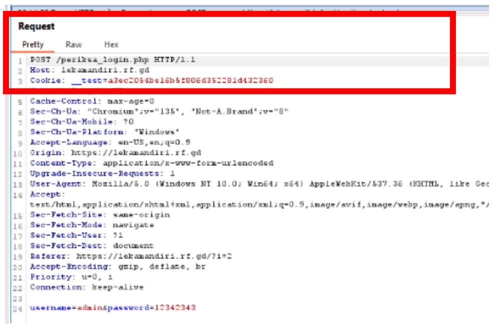
(c) Missing Anti-Clickjacking Header. The PT. Leka Mandiri Cikarang website lacks security headers like X-Frame-Options and Content-Security-Policy with frame-ancestors rules, which help prevent Clickjacking attacks. Without these headers, pages can be embedded in iframes controlled by external parties. Adding these headers is essential to strengthen protection against such attacks.

4.1.4 Penetration Testing

This phase aims to exploit vulnerabilities identified from the OWASP ZAP scanning results using tools appropriate to each vulnerability.

(a) Anti-CSRF Token Testing

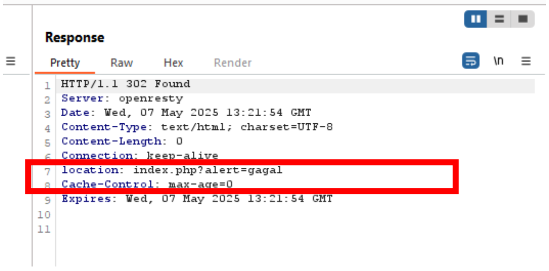
The Anti-CSRF token protects web applications from Cross-Site Request Forgery (CSRF) attacks by generating a unique, random string for each user session, which must be included in sensitive requests. CSRF attacks exploit a logged-in user's credentials to send unauthorized requests without their knowledge. Testing for CSRF vulnerabilities on http://lekamandiri.web.id/ was conducted using Burp Suite by capturing and analyzing requests, as shown in Figure 7.



```
POST /periksa_login.php HTTP/1.1
Host: lekamandiri.rf.gd
Cookie: __test=a3ec2054be16b5f806d352281d432360

Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="135", "Not-A.Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin: https://lekamandiri.rf.gd
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://lekamandiri.rf.gd/?i=2
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive

username=admin&password=12342343
```

Figure 7. Request results

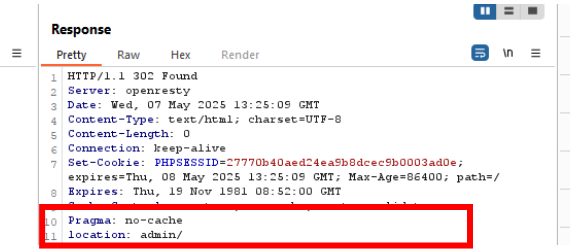
The next step is sending the request to Repeater to attempt exploiting the CSRF attack by entering a username and password. In this test, incorrect credentials were used, as shown in Figure 8.



```
HTTP/1.1 302 Found
Server: openresty
Date: Wed, 07 May 2025 13:21:54 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
location: index.php?alert=gagal
Cache-Control: max-age=0
Expires: Wed, 07 May 2025 13:21:54 GMT
```

Figure 8. attempted login with wrong user

Figure 8 shows that the server response returns a "Area Alert failed" message, indicating an incorrect account entry. Subsequently, a valid user account is entered as shown in Figure 9.



```
HTTP/1.1 302 Found
Server: openresty
Date: Wed, 07 May 2025 13:25:09 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
Set-Cookie: PHPSESSID=27770b40aed24ea9b8dcec9b0003ad0e;
expires=Thu, 08 May 2025 13:25:09 GMT; Max-Age=86400; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
location: admin/
```

Figure 9. User with correct account

Figure 9 shows the server responding with "Location:admin/", indicating the entered account is valid. This reveals the site's vulnerability to Cross-Site Request Forgery (CSRF) attacks due to inadequate Anti-CSRF token protection. Attackers can exploit this to access the system and perform unauthorized actions like password changes. Studies show CSRF attacks succeed frequently without proper token validation, leading to significant losses such as data theft and unauthorized system modifications.

(b) Content Security Policy (CSP) Testing
Content Security Policy (CSP) is a security layer that helps prevent attacks like Cross-Site Scripting (XSS) and data injection by controlling which resources a browser can load. During penetration testing, a reflected XSS vulnerability was found on the Leka Mandiri site, detected by injecting test content during the login and transaction processes and analyzed using Chrome's inspection tool see Figure 10.

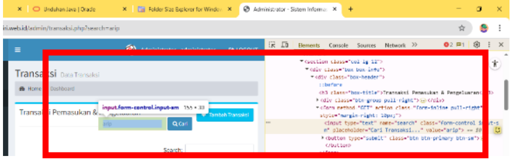


Figure 10. inspect chrome features

Figure 10 shows text injection detected in the URL. The next step is to check if the injection appears in the page's text elements by injecting URL-encoded HTML code using Burp Suite's Decoder, as illustrated in Figure 11.

Figure 11. Encode to url result

Figure 11 shows the URL input results, which were then encoded and pasted into the target website as shown in Figure 12.

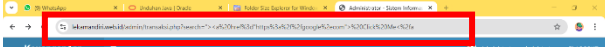


Figure 12. Encode the url pasted into the main page

Figure 12 shows the URL used for a search, which, when executed, reveals that the injected script is defined and can run, as illustrated in Figure 13.

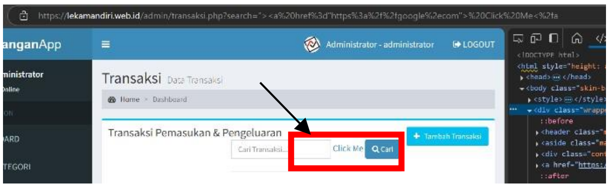
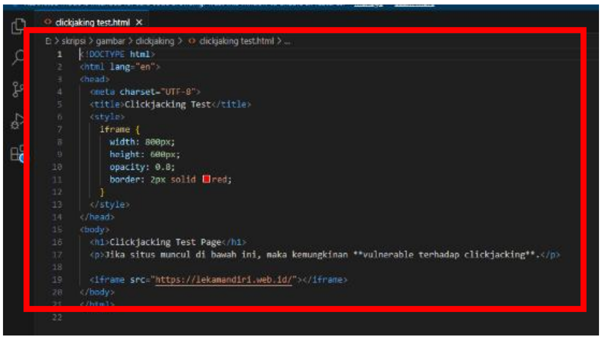

https://lekamandiri.web.id/admin/transaksi.php?search="><a%20href%3d"https%3a%2f%2fgoogle%2ecom">%20Click%20Me<%2fa
anganApp
Administrator - administrator
LOGOUT
Transaksi Data Transaksi
Home › Dashboard
Transaksi Pemasukan & Pengeluaran
Cari Transaksi...
Click Me
Cari
Tambah Transaksi

Figure 13. Defined Url result

Figure 13 shows that a Reflected XSS attack was successfully executed via the search feature, as the injected HTML element appeared and redirected the user to an external site (Google). This occurred due to insufficient input filtering or sanitization, allowing encoded HTML in the URL to be executed by the browser. The application's lack of protection against script injection via URL parameters makes it vulnerable to Reflected XSS attacks.

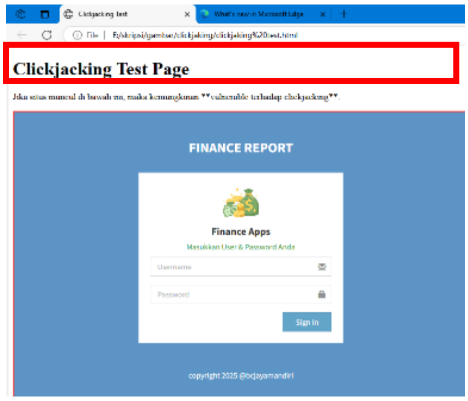
(c) Anti-Clickjacking Header Testing
Anti-Clickjacking headers, such as X-Frame-Options, are utilized as a assurance component to avoid unauthorized parties from stacking pages in iframe components, which can lead to clickjacking assaults. In this test, the defenselessness was distinguished with a moo seriousness rating. To test the misuse, an iframe script was utilized. The starting step was to put the target location address into an iframe component in an HTML record, as appeared in Figure 14.


clickjaking test.html
E: > skripsi > gambar > clickjaking > clickjaking test.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Clickjacking Test</title>
  <style>
    iframe {
      width: 800px;
      height: 600px;
      opacity: 0.8;
      border: 2px solid red;
    }
  </style>
</head>
<body>
  <h1>Clickjacking Test Page</h1>
  <p>Jika situs muncul di bawah ini, maka kemungkinan **vulnerable terhadap clickjacking**.</p>

  <iframe src="https://lekamandiri.web.id/"></iframe>
</body>
</html>
```

Figure 14. Html Inframe Clickjaking

Figure 14 shows the URL entered into the iframe, followed by executing the script on Google, as illustrated in Figure 15.


Clickjacking Test
What's new in Microsoft Edge
File | E:/skripsi/gambar/clickjaking/clickjaking%20test.html
FINANCE REPORT
Finance Apps
Masukkan User & Password Anda
Username
Password
Sign In
copyright 2025 @bcjayamandiri

Clickjacking Test Page

Jika situs muncul di bawah ini, maka kemungkinan **vulnerable terhadap clickjacking**.

Figure 15. Run the script that has been created

Figure 15 shows that the website https://lekamandiri.web.id/ is vulnerable to clickjacking attacks, as the iframe script runs successfully and displays the embedded URL. This vulnerability is due to missing or incorrect Anti-Clickjacking headers like X-Frame-Options:SAMEORIGIN. Such weaknesses allow attackers to trick users into clicking hidden elements, potentially leading to credential theft, unauthorized data changes, or malicious command execution.

4.1.5 Maintenance (Patching)

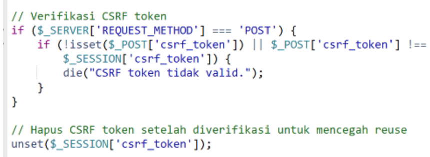
This stage focuses on maintenance or patching, including system repairs, configuration updates, code modifications, and implementation of additional policies. After completion, the process returns to vulnerability detection to ensure security gaps are addressed or critical risks minimized.

(a) Penetration testing improvements

After conducting the investigation from the Helplessness Discovery and Entrance Testing stages, the following step is to settle the site based on the OWASP Destroy comes about. The check appeared a few vulnerabilities with a medium caution level, to be specific within the Anti-Clickjacking header, Anti-CSRF Token, and Substance Security Arrangement (CSP). Suggestions for settling these issues will be sketched out as takes after.

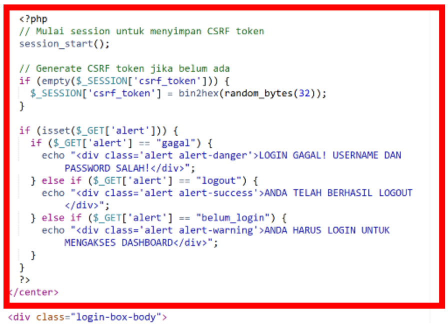
1. Token Anti-CSRF

The previous penetration test revealed that lekamandiri.web.id is vulnerable to CSRF attacks due to insufficient security measures, allowing attackers to perform unauthorized actions on behalf of logged-in users. To address this, implementing an Anti-CSRF token on login forms is essential. This can be configured in the index.php and cek_login.php files, enabling the system to automatically add a unique token to each form, as illustrated in Figures 16 and 17.



```
// Verifikasi CSRF token
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (!isset($_POST['csrf_token']) || $_POST['csrf_token'] !==
        $_SESSION['csrf_token']) {
        die("CSRF token tidak valid.");
    }
}

// Hapus CSRF token setelah diverifikasi untuk mencegah reuse
unset($_SESSION['csrf_token']);
```

Figure 16. *editing periksa_login.php file*



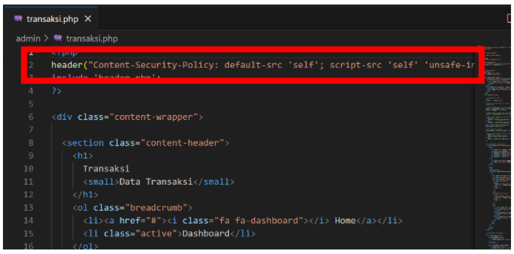
```
<?php
// Mulai session untuk menyimpan CSRF token
session_start();

// Generate CSRF token jika belum ada
if (empty($_SESSION['csrf_token'])) {
  $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}

if (isset($_GET['alert'])) {
  if ($_GET['alert'] == "gagal") {
    echo "<div class='alert alert-danger'>LOGIN GAGAL! USERNAME DAN
        PASSWORD SALAH!</div>";
  } else if ($_GET['alert'] == "logout") {
    echo "<div class='alert alert-success'>ANDA TELAH BERHASIL LOGOUT
        </div>";
  } else if ($_GET['alert'] == "belum_login") {
    echo "<div class='alert alert-warning'>ANDA HARUS LOGIN UNTUK
        MENGAKSES DASHBOARD</div>";
  }
}
?>
</center>
<div class="login-box-body">
```

Figure 17. editing *Index.Php File*

Figures 16 and 17 show the use of Anti-CSRF tokens on the login form, ensuring every form request includes a valid token verified by the server.

2. Content Security Policy (CSP) Header Not Set

Content Security Policy (CSP) is a security feature that controls which resources the browser can load to prevent attacks like Reflected XSS. Testing revealed the website lacked CSP, allowing untrusted content. To mitigate this, the CSP header can be set to allow only same-origin content, as shown in Figure 18.



transaksi.php ×

admin > transaksi.php

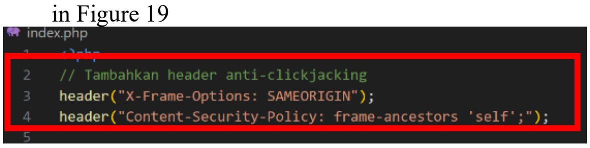
```
<?php
header("Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-ir
include 'header.php';
?>

<div class="content-wrapper">

  <section class="content-header">
    <h1>
      Transaksi
      <small>Data Transaksi</small>
    </h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
      <li class="active">Dashboard</li>
    </ol>
```

Figure 18. Maintenance Header CSP

Figure 18 shows the Content Security Policy (CSP) header configured to allow only resources from the same domain to be loaded by the browser.

3. Anti-Clickjacking Header

To fix the Anti-Clickjacking Header vulnerability, the X-Frame-Options header is set to SAMEORIGIN and the frame-ancestors 'self' directive is added to the Content Security Policy (CSP). This prevents the website from being embedded in iframes from untrusted sources, protecting against clickjacking attacks. If iframe use is necessary, additional configurations are required as shown in Figure 19



index.php

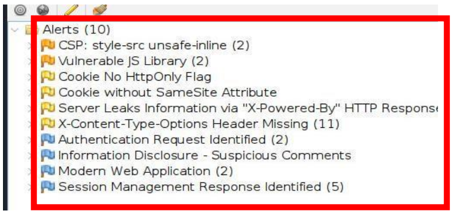
```
<?php
// Tambahkan header anti-clickjacking
header("X-Frame-Options: SAMEORIGIN");
header("Content-Security-Policy: frame-ancestors 'self';");
```

Figure 19. index.php X-Frame-Options Header Configuration

Figure 19 shows the configuration of the X-Frame-Options and Content Security Policy headers on index.php, which help prevent the lekamandiri.web.id site from clickjacking attacks by blocking page loading in iframes from other domains.

(b) Second Vulnerability Detection Analysis

After completing the repair process, vulnerability detection is repeated using OWASP ZAP to evaluate the effectiveness of the fixes and ensure the security risks on lekamandiri.web.id are significantly reduced, as shown in Figure 20.



Alerts (10)
- CSP: style-src unsafe-inline (2)
- Vulnerable JS Library (2)
- Cookie No HttpOnly Flag
- Cookie without SameSite Attribute
- Server Leaks Information via "X-Powered-By" HTTP Respons
- X-Content-Type-Options Header Missing (11)
- Authentication Request Identified (2)
- Information Disclosure - Suspicious Comments
- Modern Web Application (2)
- Session Management Response Identified (5)

Figure 20. Second OWASP ZAP Scan Results

Figure 20 shows the second scan results with 10 vulnerabilities (2 medium, 4 low, 4 informational). After patching and security improvements like enhanced headers, Anti-CSRF tokens, and CSP configuration, both the number and severity of vulnerabilities decreased significantly, indicating effective mitigation.

4.1.6 Reporting

The report presents the security testing and patching results on PT. Leka Mandiri Cikarang, identifying vulnerabilities classified as high, medium, and low risk. Tools used included OWASP ZAP, Burp Suite, Nmap, and Whois. The main issues addressed were the Anti-CSRF Token, Content Security Policy (CSP), and Anti-Clickjacking Header. The report covers findings before and after the maintenance.

(a) Results of Security Testing Prior to Maintenance (Patching). The vulnerability analysis before patching using OWASP ZAP identified security gaps in the Anti-CSRF Token, Content Security Policy (CSP), and Anti-Clickjacking Header. These vulnerabilities were further tested, with results detailed below.

1. There were 4 Token Anti-CSRF vulnerabilities, all classified with a Medium Alert Level. The test results showed success by attempting to exploit the vulnerability using the 'Send to Repeater' feature, followed by attempts to gain access, which resulted in a server response of 'Location: admin/
2. The vulnerability related to Content Security Policy (CSP) was identified six times, each classified with a medium alert level. The testing was successful in revealing that the website lacked CSP protection against Reflected XSS vulnerabilities. This was demonstrated by attempting to exploit the transaction page through a modified URL, edited using the decoder feature in Burp Suite.
3. For the Anti-Clickjacking Header, there are 4 issues with a Medium Alert Level. The test results showed a successful attack because the website does not use the X-Frame-Options header set to “Same Origin.”.

The test results identified several moderate-risk security vulnerabilities on PT. Leka Mandiri Cikarang’s website, including the Anti-CSRF Token, Content Security Policy (CSP), and Anti-Clickjacking Header. These vulnerabilities could be exploited to hijack user sessions, inject malicious content, and steal credentials. Immediate remediation is required to significantly enhance the website's security.

(b) Following the maintenance (patching) process, a second round of testing was conducted using OWASP ZAP to verify the fixes applied to previously identified vulnerabilities. The results confirmed that issues related to the Anti-CSRF Token, Anti-Clickjacking Header, and Content Security Policy (CSP) had been effectively resolved. Specifically.

1. The Anti-CSRF vulnerability, previously rated as a medium alert, was mitigated by integrating a unique Anti-CSRF token for each login session and validating it on the server side.
2. The CSP vulnerability, also rated medium, was addressed by configuring a security policy with directives such as default-src 'self', script-src 'self' 'nonce-random123', and others.
3. The Anti-Clickjacking issue was resolved by implementing the X-Frame-Options: SAMEORIGIN header and the CSP frame-ancestors 'self' directive.

(c) Comparison of Scan Results Pre- and Post-Fix From the scanning results before and after the improvements, we observed a difference where vulnerabilities such as the Anti-CSRF Token, Content Security Policy (CSP), and Anti-Clickjacking Header were reduced after the fixes were applied. The comparison of these reductions can be seen in Table 3 as follows.

Table 3. Comparasion Table Before and After Improvemens

Risk Level	Before Repair	Vulnerability Score (CVSS)	Status
Medium	*Token Anti-CSRF*	5.8 → 0.0	Done
Medium	*Anti-Clickjacking Header*	4.1 → 0.0	Done
Medium	*Content Security Policy (CSP)*	6.1 → 0.9	Done

Table 3 appears a comparison with the Common Powerlessness Scoring Framework (CVSS) v3.1 score evaluation. the checking comes about prepare some time recently and after repairs to vulnerabilities within the framework. The vulnerabilities are centered on the Anti-CSRF Token, Substance Security Arrangement (CSP), and Anti-Clickjacking Header vulnerabilities. The Common Helplessness Scoring System (CVSS) v3.1 score may be a standard utilized to evaluate the seriousness of security vulnerabilities. CVSS scores run from 0.0 (no chance) to 10.0 (exceptionally basic chance) [26]. Medium-rated vulnerabilities such as (Anti-CSRF Token, Anti-Clickjacking Header, Substance Security Arrangement (CSP):

These 3 vulnerabilities had CVSS scores of 6.5 and 6.1 some time recently the settle since they may lead to assaults such as Cross-Site Scripting (XSS) or clickjacking. After the fix, the score dropped to 0.0 since these vulnerabilities were not found. Some time recently the support (fixing), the vulnerabilities had medium chance and moo chance levels, which had the potential to imperil application security that might hurt PT. Leka Mandiri Cikarang. After the settle, these vulnerabilities were nearly totally settled. This appears that the relief steps executed, such as altering the expansion of interesting tokens, executing security arrangements, and actuating extra security highlights, have been compelling on the PT Leka Mandiri Cikarang site.

5. CONCLUSION

The research on the website http://lekamandiri.web.id utilized penetration testing with tools like OWASP ZAP to identify and mitigate security vulnerabilities, aiming to improve the overall protection of PT Leka Mandiri Cikarang’s site. Initial assessments revealed risks related to CSRF, Clickjacking, and improper Content Security Policy (CSP) implementation due to missing CSRF tokens, insufficient CSP configuration, and the absence of iframe protection. After applying necessary fixes, these issues were resolved by adding CSRF tokens to login forms, enforcing a strict CSP to allow only trusted sources, and implementing the X-Frame-Options header to prevent Clickjacking. The effectiveness of these improvements was confirmed through CVSS v3.1 scores, where medium-level vulnerabilities (6.1–6.5) were reduced to 0.0, indicating a significant enhancement in website security based on OWASP ZAP re-testing.

6. REFERENCES

[1] B H. Gunawan, “Pengukuran Kesadaran Keamanan Informasi Dan Privasi Dalam Sosial Media,” J. Muara Sains, Teknol. Kedokt. dan Ilmu Kesehat., vol. 5, no. 1, p. 1, 2021, doi:10.24912/jmstkik.v5i1.3456.

[2] D Fauzan, F. Y., & Syukhri, S. (2021). Analisis Metode Web Security PTES (Penetration Testing Execution And Standart) Pada Aplikasi E-Learning Universitas Negeri Padang. Voteteknika (Vocational Teknik Elektronika dan Informatika), 9(2), 105–111. https://doi.org/10.24036/voteteknika.v9i2.111778.

[3] Hidayatulloh, S., & Saptadiaji, D. (2021). Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP). Jurnal Algoritma, 18(1), 77–86. https://doi.org/10.33364/algoritma/v.18-1.827

[4] Dewi, B. T. K. & Setiawan, M. A. (2022). Kajian Literatur: Metode dan Tools Pengujian Celah Keamanan Aplikasi Berbasis Web. Automata, 3(1), 1–8. https://journal.uii.ac.id/Automata/article/view/21883/12030.

[5] R. Hafsari, R. Rahmadani Saputra, and M. Afin Wirdyansah, "Perancangan Absensi Berbasis Web Dengan Metode Waterfall (Studi Kasus: PT. GlobalRiau Data Solusi)," vol. 4, no. 1, pp. 306–312, 2023, doi: 10.37859/coscitech.v4i1.5400.

[6] A. Rochman, R. Rohian Salam, dan Sandi Agus Maulana Sekolah Tinggi Manajemen Ilmu Komputer, and S. Likmi, "DI RUMAH SAKIT XYZ," ANALISIS KEAMANAN WEBSITE DENGAN INFORMATION SYSTEM SECURITY ASSESSMENT FRAMEWORK (ISSAF) DAN OPEN WEB APPLICATION SECURITY PROJECT, vol. 2, no. 4, 2021.

[7] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," Jurnal Informasi dan Teknologi, pp. 70–75, Mar. 2022, doi: 10.37034/jidt.v4i1.190.

[8] Elinda Revita, Intan Puspita, & Raimon Efendi. (2023). Sistem Informasi Pembayaran SPP Berbasis Web Pada MTS Al-Ihsan Tugu Rejo. INNOVATIVE: Journal Of Social Science Research, 3, 5053–5063.

[9] Widiawaty, V., & Irmanda, H. N. (2021). Website-Based Tuition Payment Information System at SMP Strada St. Fransiskus Xaverius II. In National Seminar of Computer Science Students and Its Applications (SENAMIKA) Jakarta-Indonesia

[10] Ayu Sahdilla. (2021). Design of a Web-Based Drug Sales Information System at Dian Pharmacy. 9(2).

[11] Oktaviyana, A., Mercedes Br Aritonang, M., & Saputri br Sembiring, E. (2023). Analysis and Development of Management Information Systems.

[12] Ningsih, N. F., & Riadi, I. (2021). Risk Assessment Analysis on Library Information System using OCTAVE Allegro Framework. Internasional Journal of Computer Applications, 183(28), 6–13. https://doi.org/10.5120/ijca2021921620

[13] Ikhsan, N., & Ramadhani, S. (2020). Correspondence Administration Information System of the Regional Office of the Ministry of Religion of Riau Province. Journal of Technology and Business Information Systems, 2(2), 141–151. https://doi.org/10.47233/jteksis.v2i2.126

[14] C. Juandy, "Penilaian Risiko pada Layanan Sistem Informasi Desa Berdaya Menggunakan OCTAVE Allegro," 2024.

[15] Rosita R, "Penilaian Keamanan Informasi Pada Layanan Surat Menggunakan Indeks KAMI 4.2," 2022.

[16] C. A. Prawastiyo and I. Hermawan, "Pengembangan Front-End Website Perpustakaan Politeknik Negeri Jakarta Dengan Menggunakan Metode User Centered Design," Inf. Sci. Libr., vol. 1, no. 2, pp. 50–60, 2022

[17] I. G. Arya Kukuh Y, Geraldo Alfarenb, "Analisis Serangan Sistematik Penetration Testing : Sebuah Review," J. Ilm. Inform. Komput., vol. 1 no. 2, pp. 21–26, 2022.

[18] M. A. Suharto and M. N. Apriyani, "Konsep Cyber Attack , Cyber Crime , Dan Cyber Warfare Dalam Aspek Hukum Internasional," vol. 17, pp. 98–107, 2021.

[19] Al Fajar, F. (2020). Analisis Keamanan Aplikasi Web Prodi Teknik Informatika Uika Menggunakan Acunetix Web Vulnerability. Inova-tif, 3(2). https://doi.org/10.32832/inovatif.v3i2.4127

[20] M. A. Adiguna and B. W. Widagdo, "Analisis Keamanan Jaringan WPA2-PSK Menggunakan Metode Penetration Testing (Studi Kasus : Router Tp-Link Mercusys Mw302r)," J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan), vol. 5, no. 2, pp. 1–8, 2022, doi: 10.47970/siskom-kb.v5i2.268.

[21] A. Aliefyan, "Penetration Testing Untuk Mengetahui Kerentanan Keamanan Aplikasi Web Menggunakan Standar OWASP 10 pada domain Web Perusahaan Penetration Testing Untuk Mengetahui Kerentanan Keamanan Aplikasi Web," ResearchGate, no. July, 2020.

[22] Yudiana, Y., Elanda, A., & Buana, R. L. 2021. Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIK Rosma Dengan Menggunakan OWASP Top 10. CESS (Journal of Computer Engineering, System and Science), 6(2), 185

[23] Riadi, I., Umar, R., Lestari, T., Informasi, S., Dahlan, U. A., Informatika, T., & Ahmad, U. (2020). Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP. 5(3), 146– 152.

[24] Darojat, E. Z., Sediyono, E., & Sembiring, I. (2022). Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner. Jurnal Sistem Informasi Bisnis, 12(1), 36–44. https://doi.org/10.21456/vol12iss1pp36-44

[25] H. Sofyan, M. Sugiarto, and B. M. Akbar, "Implementation of Penetration testing on Websites to Improve Security of Information Assets UPN 'Veteran' Yogyakarta," Jurnal Informatika dan Teknologi Informasi, vol. 20, no. 2, pp. 1–10, 2023, doi: 10.31515/telema tika .v20i2.7757

[26] FIRST. (2023). Common Vulnerability Scoring System v3.1: Specification Document. Retrieved from https://www.first.org/cvss/v3.1/specification-document.