Penetration Test Report

Nikhil Tyagi

May 23th, 2024

Email: Nikhilcyber610@gmail.com
Website scanned:https://download.vulnhub.com/matrix/Machine_Matrix.zip

Table of Contents

Table of Contents

Executive Summary

Objective:

Our objective in conducting this penetration test was to thoroughly evaluate the security of the target website, https://download.vulnhub.com/matrix/Machine_Matrix.zip. By simulating real-world attack scenarios, we aimed to identify vulnerabilities that could potentially be exploited by malicious actors.

Scope:

We focused our assessment on the target website's infrastructure and web application, employing a combination of automated tools and manual techniques. Our goal was to provide a comprehensive evaluation of the website's security controls and defenses.

Findings:

Our examination uncovered several critical vulnerabilities within the target website. These vulnerabilities ranged from misconfigurations, which could expose sensitive data, to outdated software versions susceptible to known exploits. Additionally, we identified potential weaknesses in authentication mechanisms, raising concerns about unauthorized access to sensitive areas of the site.

Recommendations:

To address these vulnerabilities and enhance the overall security posture of the target website, we recommend several measures. These include implementing a robust patch management process to ensure timely software updates, enhancing authentication mechanisms with multi-factor authentication where applicable, providing comprehensive security training for staff to raise awareness of potential risks, and conducting regular security assessments to proactively identify and address emerging threats.

Conclusion:

In conclusion, addressing the vulnerabilities identified during this penetration test and implementing the recommended measures are critical steps in fortifying the security of the target website. By prioritizing security and adopting proactive measures, organizations can effectively mitigate the risk of cyber threats and safeguard their assets and sensitive information.

Project Objective

The primary objective of the penetration test is to assess the security posture of the website located at https://download.vulnhub.com/matrix/Machine_Matrix.zip. Specifically, the assessment aims to identify vulnerabilities, weaknesses, and misconfigurations within the target system that could be exploited by malicious actors. Additionally, the assessment seeks to evaluate the effectiveness of existing security controls, assess web application security, validate compliance with relevant security standards, and provide actionable recommendations for enhancing the overall security posture of the target system. By achieving these objectives, the penetration test aims to provide stakeholders with valuable insights into security risks and empower them to make informed decisions regarding security improvements and risk mitigation strategies.

Project Scope

For my internship project as a penetration tester, I'll be conducting a security assessment of the website Machine_Matrix.zip, accessible at https://download.vulnhub.com/matrix/Machine_Matrix.zip. This assessment will involve examining the website's setup, functionality, and potential vulnerabilities. I'll focus on identifying any security weaknesses that could allow unauthorized access or compromise sensitive information. Additionally, I'll evaluate the effectiveness of existing security measures, such as firewalls or encryption, to determine if they adequately protect the website. To assist in my assessment, I'll utilize common penetration testing tools like nmap, dirbuster, burpsuite, and sqlmap. Throughout the project, I'll document my findings, noting any security issues discovered and their potential impact. Finally, I'll compile a comprehensive report summarizing my findings and providing recommendations for addressing identified security risks.

Summary of Findings

Throughout the penetration test, several vulnerabilities and weaknesses were identified within the target system. These vulnerabilities ranged from misconfigurations in the infrastructure to flaws in the web application's security controls. Critical vulnerabilities, such as default credentials and SQL injection vulnerabilities , were discovered, posing significant risks to the confidentiality, integrity, and availability of the system. Additionally, findings included inadequate patch management practices, insecure network configurations, and insufficient access controls. Each vulnerability was assessed for severity and potential impact, with recommendations provided for remediation. The findings highlight the importance of addressing security gaps and implementing robust security measures to protect the target system from potential exploitation.

Summary of Recommendations

The summary of recommendations outlines actionable steps to address the vulnerabilities identified during the penetration test. Each recommendation is tailored to mitigate specific risks and improve the overall security posture of the target system. Prioritization is given to critical vulnerabilities with a high potential for exploitation, ensuring that resources are allocated effectively to address the most pressing security concerns. Recommendations include implementing software patches, reconfiguring security settings, enhancing access controls, and improving monitoring and detection capabilities. Additionally, recommendations emphasize the importance of ongoing security awareness training for personnel and regular security assessments to identify and address emerging threats. By implementing these recommendations, stakeholders can enhance the resilience of the target system against potential security threats and mitigate the risk of unauthorized access or data breaches.

Testing Methodology

The testing methodology describes the approach, techniques, and tools used during the penetration test to assess the security posture of the target system. It outlines the various phases of the assessment, including reconnaissance, vulnerability scanning, exploitation, and post-exploitation activities. Both automated scanning tools and manual testing techniques are employed to ensure comprehensive coverage and depth of analysis. Tools such as nmap, dirbuster, burpsuite, and sqlmap are utilized to identify vulnerabilities and weaknesses within the target system. Throughout the assessment, a structured testing approach is followed to systematically uncover security flaws and assess the effectiveness of existing security controls. Findings are meticulously documented and reported, providing stakeholders with clear insights into the security risks identified and

recommendations for remediation. The testing methodology serves as a guide for understanding the assessment process and ensures transparency and rigor in the evaluation of the target system's security posture.

Appendix A: Vulnerability Detail and Mitigation

Risk Rating Scale

mod_dav_svn module for the Apache HTTP Server

Rating: **High**

CVE Number **CVE-2011-0715**

Description: The mod_dav_svn module for the Apache HTTP Server, as distributed in Apache Subversion before 1.6.16, allows remote attackers to cause a denial of service (NULL pointer dereference and daemon crash) via a request that contains a lock token.

Impact: Denial of Service (DoS): Exploiting this vulnerability could lead to a denial of service

condition on the affected Apache Subversion server. By sending a specially crafted request containing a lock token, an attacker could cause the server daemon to crash, rendering the service unavailable to legitimate users , Depending on the nature of the crash and the state of the server at the time of the attack, there is a risk of potential data loss or corruption. Unsaved changes or transactions in progress may not be properly completed or recorded, leading to inconsistencies or data integrity issues within the version control repository .

Remediation: The most effective remediation is to upgrade Apache Subversion to version

1.6.16 or later. This version contains the necessary patches and fixes to address the vulnerability in the mod_dav_svn module. Users should ensure that they are using a version of Apache Subversion that is not affected by the vulnerability.

Rating: High

Description: McAfee ePolicy Orchestrator agent allows remote attackers to cause a denial of service (memory consumption and crash) and possibly execute arbitrary code via an HTTP POST request with an invalid Content-Length value, possibly triggering a buffer overflow.

Impact:

Potential Arbitrary Code Execution: Successful exploitation of the buffer overflow could allow remote attackers to execute arbitrary code on the compromised system. By leveraging this capability, attackers may gain unauthorized access to sensitive information, escalate privileges, or deploy additional malicious payloads. This scenario presents a significant security risk, as it enables attackers to achieve persistent access to the compromised system and potentially pivot to other systems within the network.

Remediation: Provide security awareness training to system administrators and users responsible for managing and maintaining ePO agents. Educate them about the importance of applying security patches promptly and following best practices to enhance the security of ePO deployments.

Conduct regular penetration testing and vulnerability assessments on ePO deployments to identify and address security weaknesses proactively. This can help uncover potential vulnerabilities before they are exploited by malicious actors.

Multiple cross-site scripting (XSS) vulnerabilities .

Rating: High

Description:

Multiple cross-site scripting (XSS) vulnerabilities in Phorum 3.1 through 5.0.3 beta allow remote attackers to inject arbitrary web script or HTML via the (1)

HTTP_REFERER parameter to login.php, (2) HTTP_REFERER parameter to register.php, or (3) target parameter to profile.php.

Impact:

1. Data Theft: By exploiting XSS vulnerabilities, attackers can steal sensitive data stored within the web application or transmitted between users and the server. This includes personal information, financial data, or any other confidential data .
2. Reputation Damage: Exploitation of XSS vulnerabilities can tarnish the reputation of the affected website or web application. Users may lose trust in the security of the platform, resulting in decreased user engagement, loss of customers, and damage to the organization's brand reputation.

Remediation:

1. Apply patches or updates provided by the Phorum developers to address the XSS vulnerabilities. Ensure that the web application is upgraded to a version that includes the necessary security fixesConfigure HTTP security headers, such as Content Security Policy (CSP), to mitigate the impact of XSS vulnerabilities. CSP allows you to define and enforce a whitelist of trusted sources for content, scripts, and other resources, reducing the risk of XSS attacks by blocking unauthorized scripts from executing. to mitigate the risk of exploitation.

CVE_SCORE: **CVE-2004-1822**

Rating: High

Description: admin.htm in Geo++ GNCASTER 1.4.0.7 and earlier does not properly enforce HTTP Digest Authentication, which allows remote authenticated users to use HTTP Basic Authentication, bypassing intended server policy.

Impact:

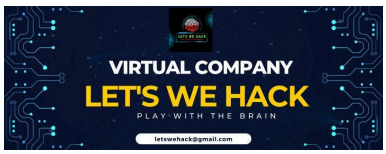
Unauthorized Access: Remote authenticated users can bypass intended server policy and gain unauthorized access to sensitive administrative functionality provided by the admin.htm page. This could allow attackers to view, modify, or delete critical system configurations, settings, or data.

Remediation: Conduct regular security audits and assessments of the GNCASTER application to identify and address any security weaknesses or misconfigurations. This includes reviewing authentication mechanisms and access controls to ensure they are properly configured.

Implement monitoring and logging mechanisms to detect and alert on unauthorized access attempts or suspicious activity related to administrative functionality. Monitor access logs and audit trails for signs of unauthorized access or misuse

CVE_NUMBER

CVE-2010-0550



HTTP File Server

Rating: High

Description: .HTTP File Server (HFS) before 2.2c tags HTTP request log entries with the username sent during HTTP Basic Authentication, regardless of whether authentication succeeded, which might make it more difficult for an administrator to determine who made a remote request.

Impact:

The inaccurate tagging of HTTP request log entries with usernames, regardless of successful authentication, can create challenges for administrators in accurately tracing and attributing remote requests within the logs.

The inability to distinguish between successful and unsuccessful authentication attempts may lead to reduced accountability and hinder incident response efforts. Administrators may find it difficult to determine the true origin of requests, impacting their ability to investigate security incidents or track user activity effectively.

Remediation: To remediate the vulnerability in HTTP File Server (HFS) versions before 2.2c, where HTTP request log entries are incorrectly tagged with the username sent during HTTP Basic Authentication regardless of whether authentication succeeded, the following steps can be taken:

1. Update to the Latest Version: Upgrade the HFS server to version 2.2c or later, which includes a fix for the vulnerability. Ensure

that you download the latest stable release from the official HFS website or repository.

2. Implement Proper Logging Mechanism: Modify the logging mechanism in HFS to accurately log HTTP request information, including the authentication status. Ensure that log entries correctly reflect the outcome of authentication attempts, distinguishing between successful and unsuccessful authentication.

CVE_NUMBER **CVE-2008-0407**

stat.php in AuraCMS 1.62

Rating: Informational

Description: stat.php in AuraCMS 1.62, and Mod Block Statistik for AuraCMS, allows remote attackers to inject arbitrary PHP code into online.db.txt via the X-Forwarded-For HTTP header in a stat action to index.php, and execute online.db.txt via a certain request to index.php.

Impact:

1. Arbitrary Code Execution: Attackers can inject malicious PHP code into the online.db.txt file, enabling them to execute arbitrary commands or scripts on the server. This can lead to

unauthorized access, data theft, system compromise, and further exploitation of the server.
2. Server Compromise: Exploiting this vulnerability may result in full compromise of the server hosting AuraCMS. Attackers can gain unauthorized access to sensitive data, manipulate system files, install backdoors, or carry out other malicious activities, giving them complete control over the server.
3. Data Breach: Attackers can access and manipulate data stored within online.db.txt, potentially compromising sensitive information such as user credentials, personal details, or financial records. This could lead to data breaches, privacy violations, and reputational damage for the affected organization.

Remediation: To remediate the vulnerability in AuraCMS 1.62 and Mod Block Statistik for AuraCMS, which allows remote attackers to inject arbitrary PHP code into online.db.txt and execute it via certain requests, several steps can be taken:

1. Patch or Update: Apply patches or updates provided by AuraCMS or the relevant module developer to address the vulnerability. Ensure that the AuraCMS installation and any associated modules are updated to the latest secure versions.
2. Input Validation: Implement strict input validation mechanisms to sanitize user-supplied data, especially in HTTP headers like X-Forwarded-For. Validate and sanitize input to prevent the injection of malicious PHP code into the online.db.txt file.
3. Secure Configuration: Review and configure the AuraCMS and web server settings to enhance security. Ensure that appropriate access controls are in place to restrict unauthorized access to sensitive files and directories, including online.db.txt.

CVE_NUMBER

CVE-2008-0390

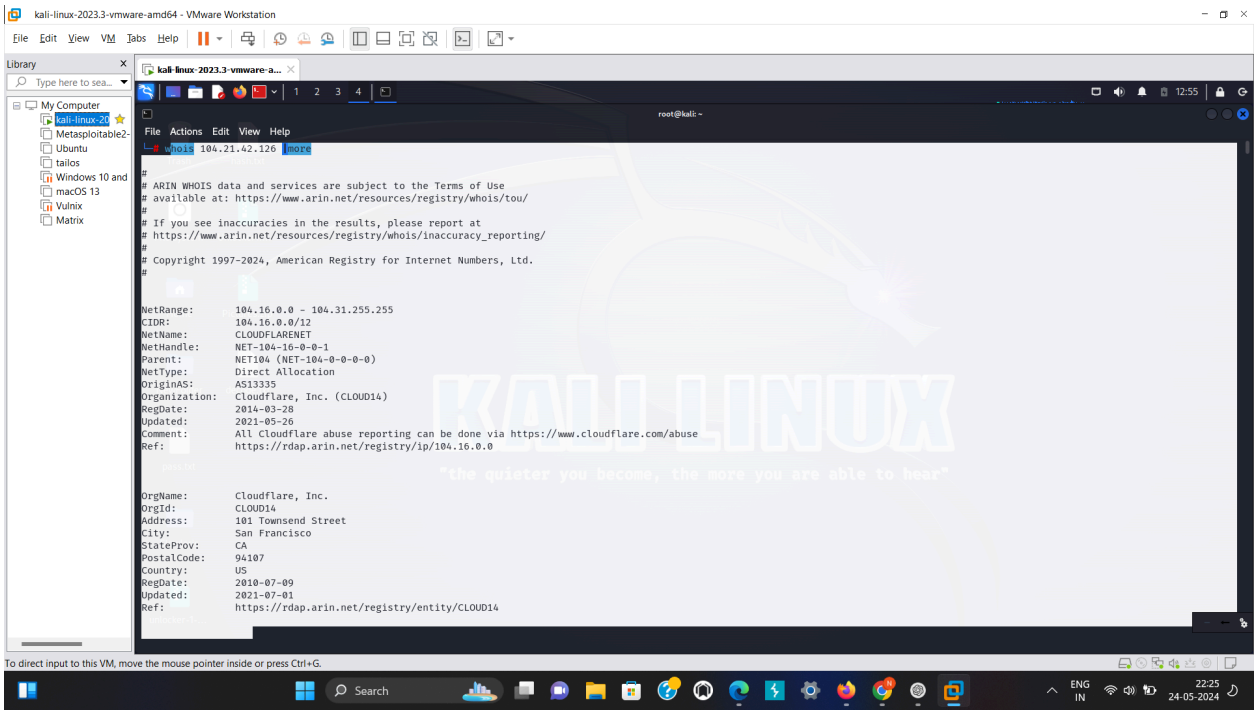
TOOLS USED —---> Nmap

Here is a Proof of concept

Step 1- Open a

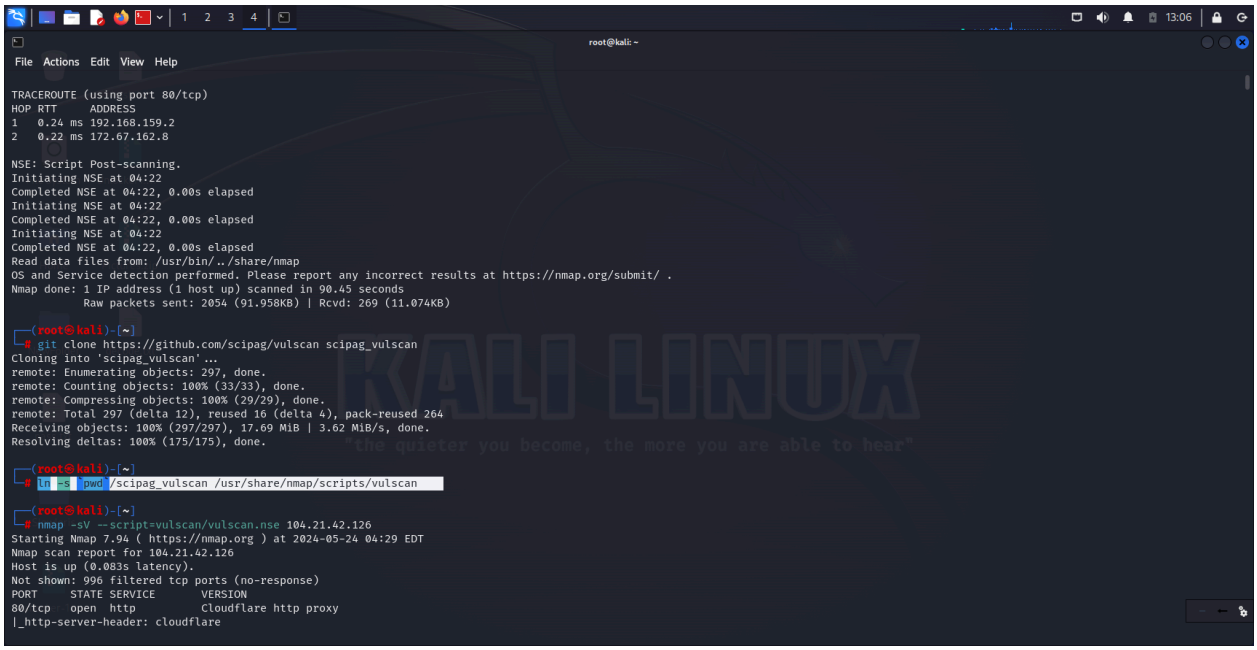
Figure 1 – Information gathering for megacorpone.com reveals three active name server

With the name servers identified, we attempted to conduct a zone transfer. We found that was vulnerable to a full DNS zone transfer misconfiguration. This provided us with a listing of hostnames and associated IP addresses, which could be used to further target the organization. (Figure 2) Zone transfers can provide attackers with detailed information about the capabilities of the organization. It can also leak information about the network ranges owned by the organization. Please see Appendix A for more information ..

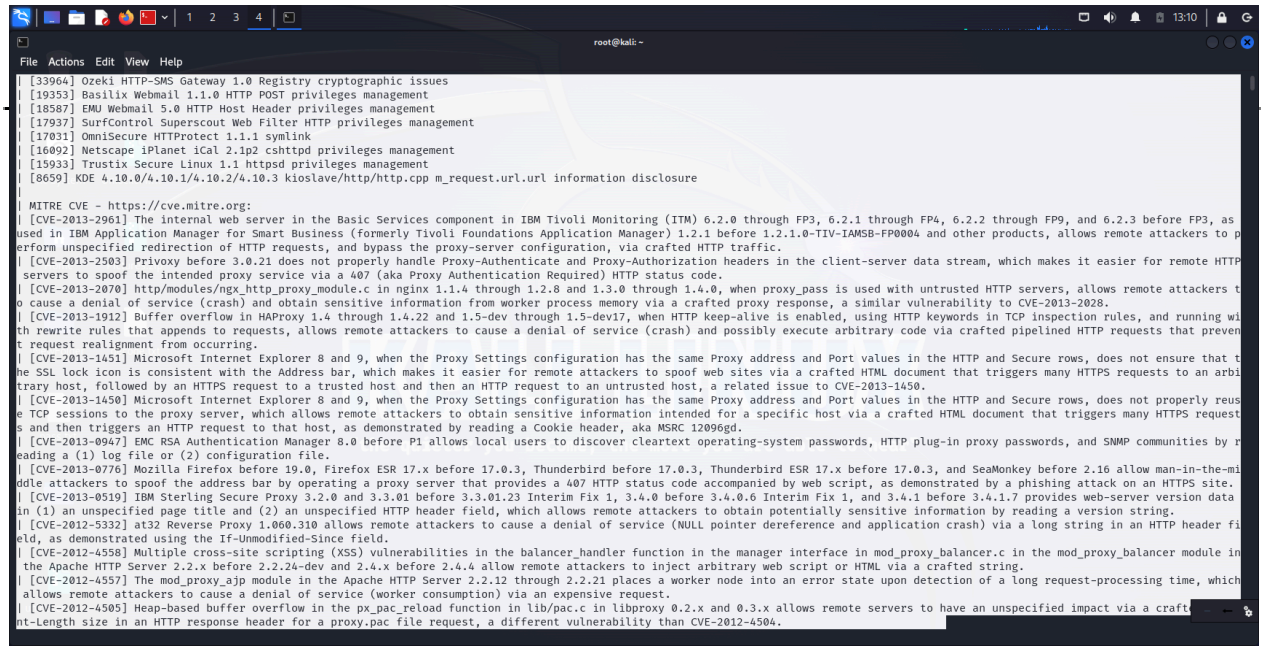


Step 3 - At step 3 we had simply installed vulnix tools which will help us to analyze the types of vulnerabilities in our system , works with the nmap.

Step 5 - Simply type the following command in order to perform for the vulnix tool .



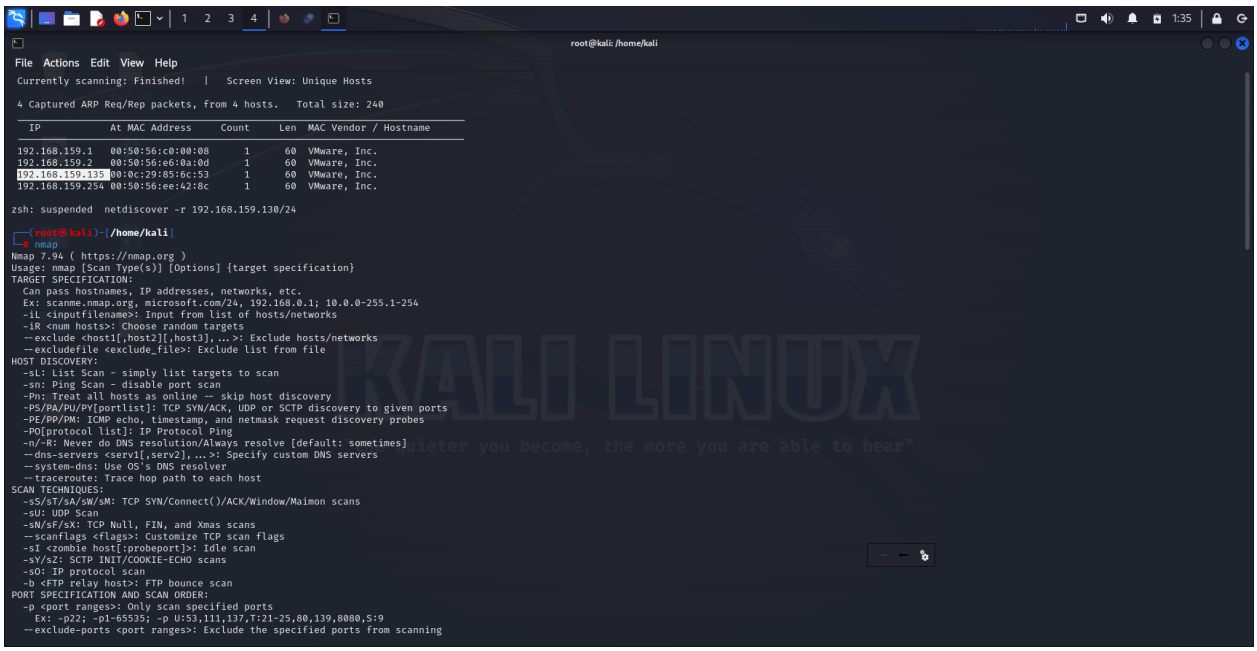
Step 6 - Now finally start the Vulscan tool with the help of nmap tools .



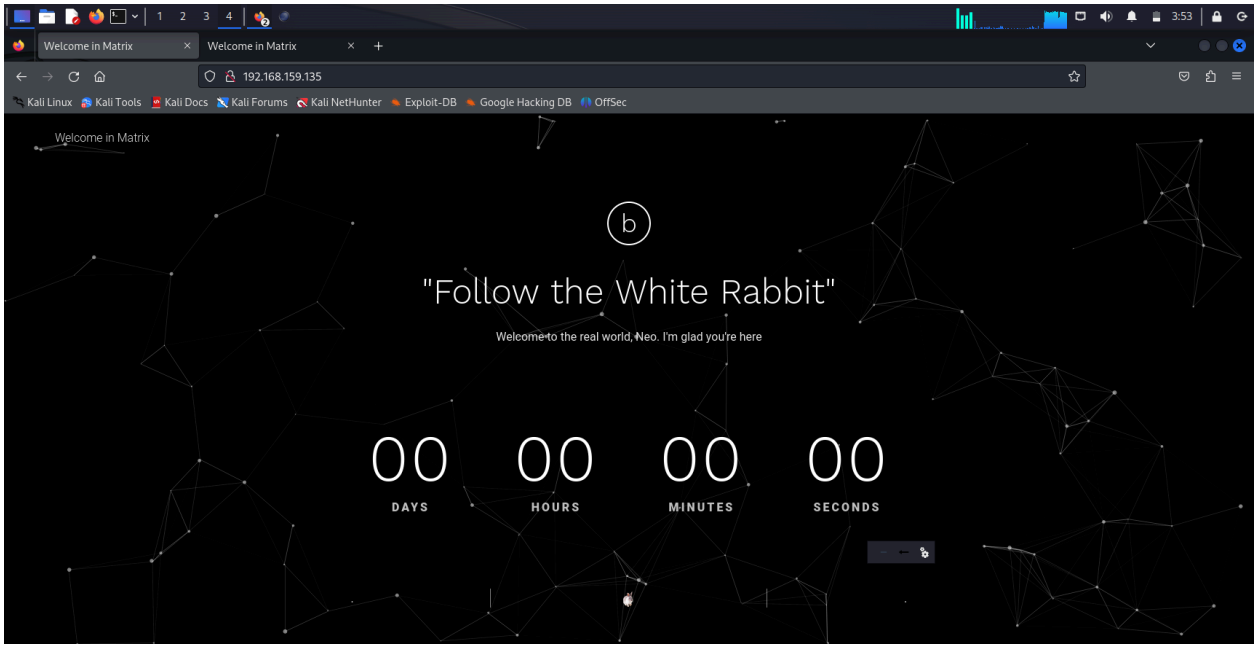

VIRTUAL COMPANY
LET'S WE HACK
PLAY WITH THE BRAIN
letswehack@gmail.com

Now Accessing the Ip address of the particular .

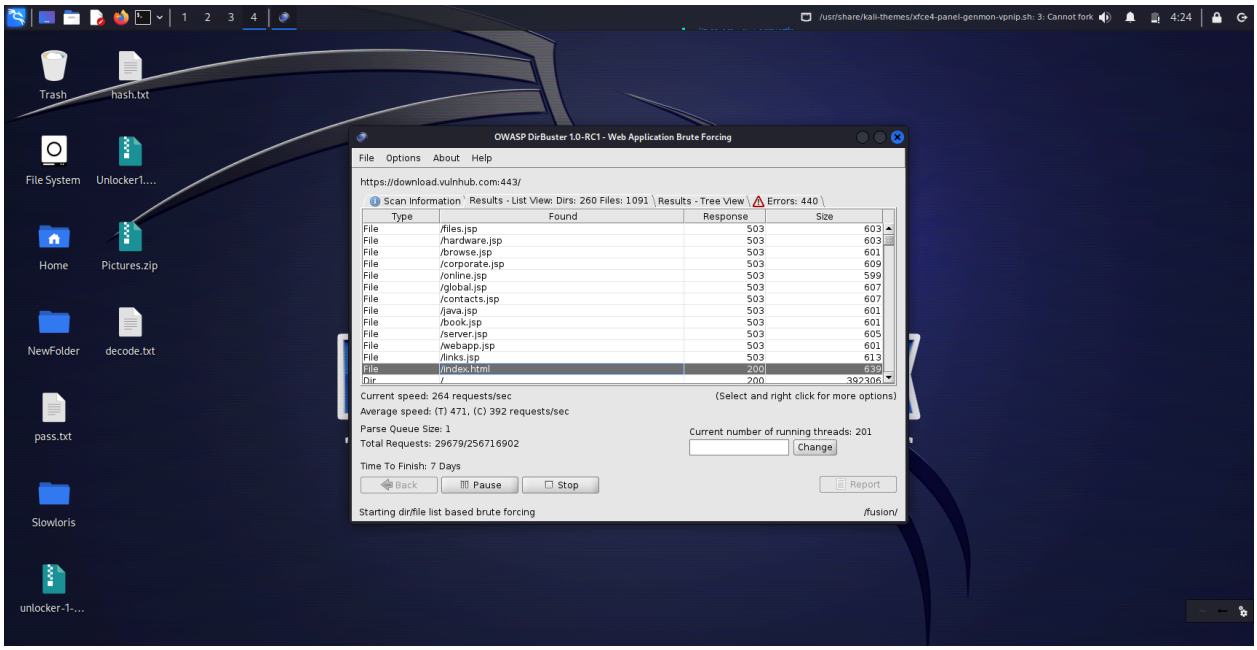
Step1 - Type the command we had analyzed



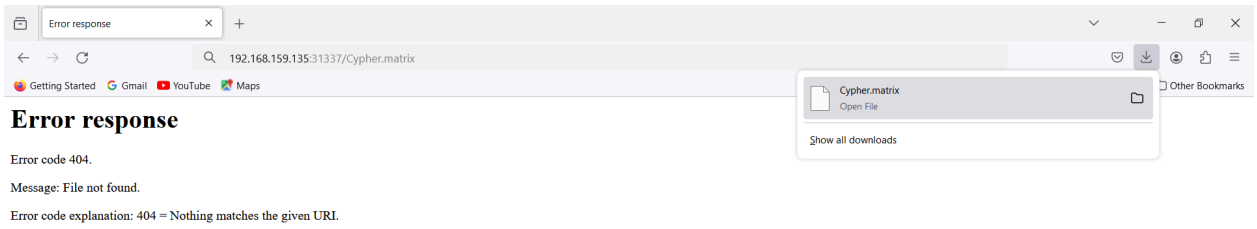
Step 11 - We had also used dirbuster unable to gathered more information about the target site .



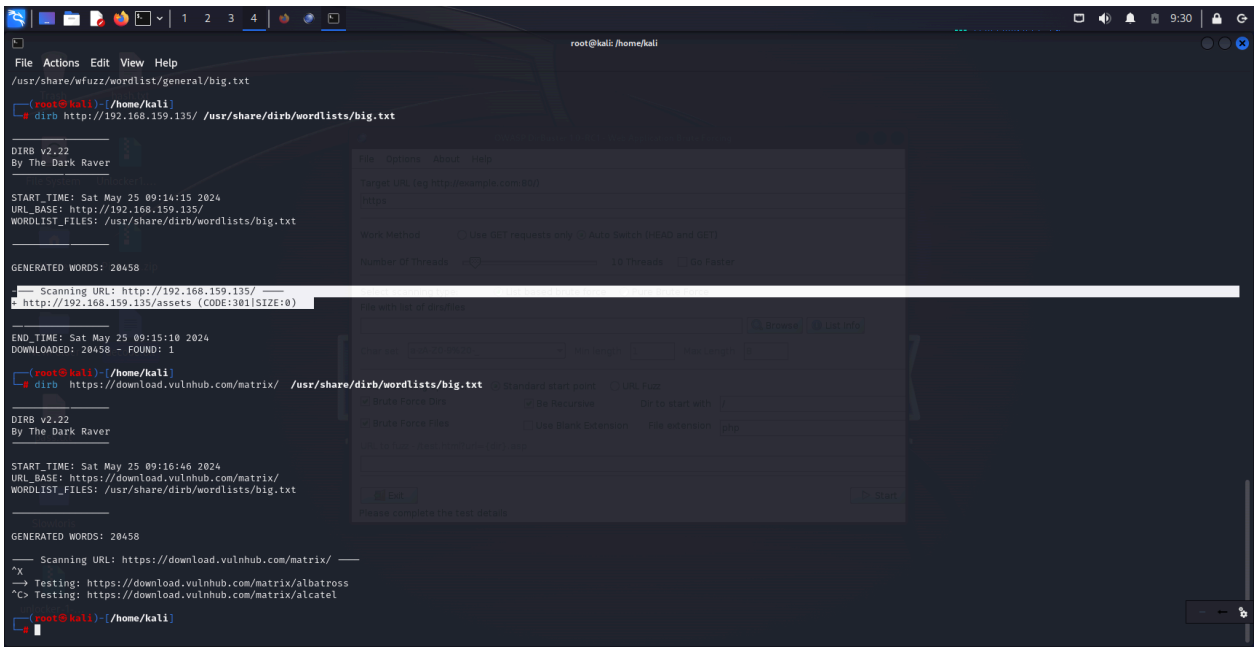
Step 12 - Using dir buster it shows that Website has only one directory which is merely the index.html



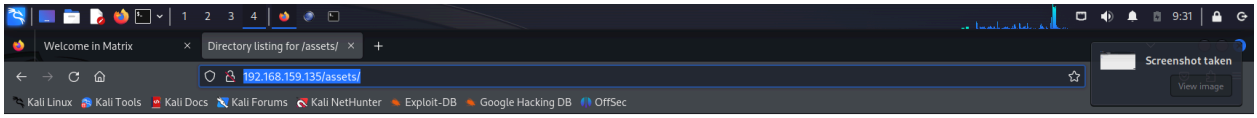
Step 13- By getting the information of directories we had noticed that there is file with the name of the cypher.matrix is totally a malware which can perform various attacks or gain unauthorized access inside the system .



Step 15 - Now run dirb tool to find more in order to find the location .

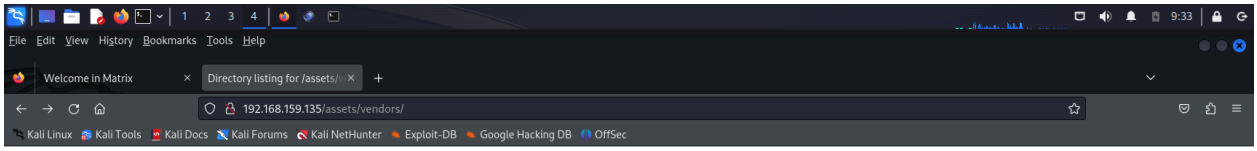


Step 3 - Now put the url in browser in order to gather more information .



Directory listing for /assets/

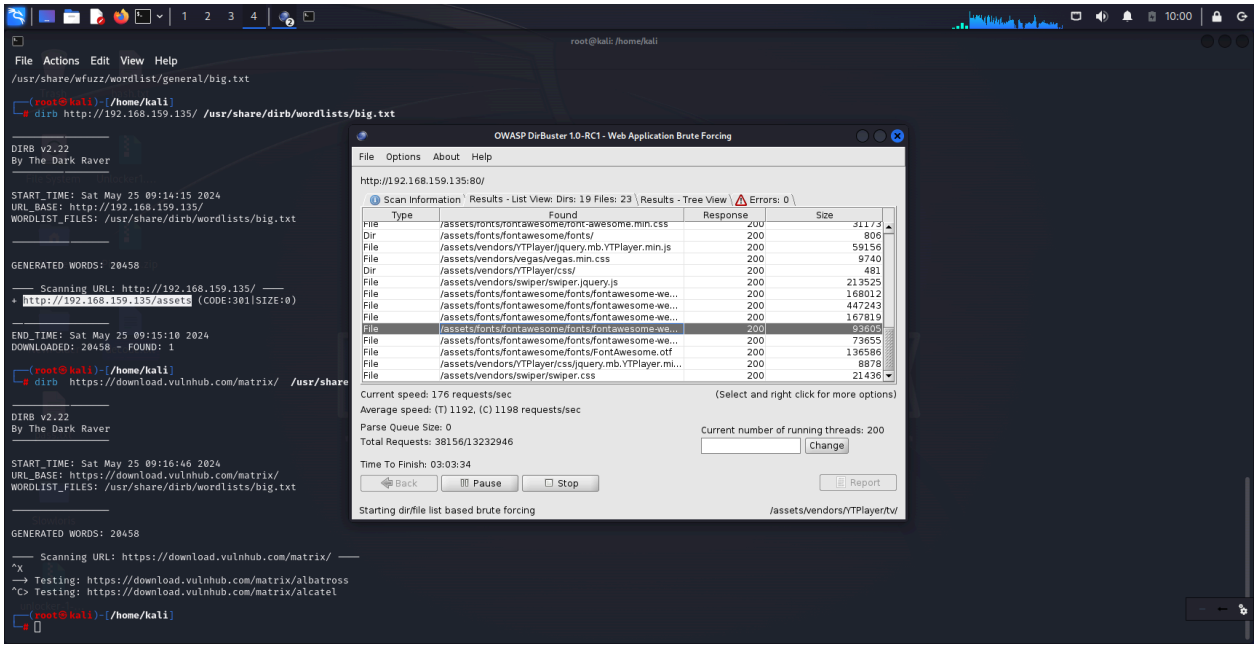
- css/
- fonts/
- img/
- js/
- vendors/



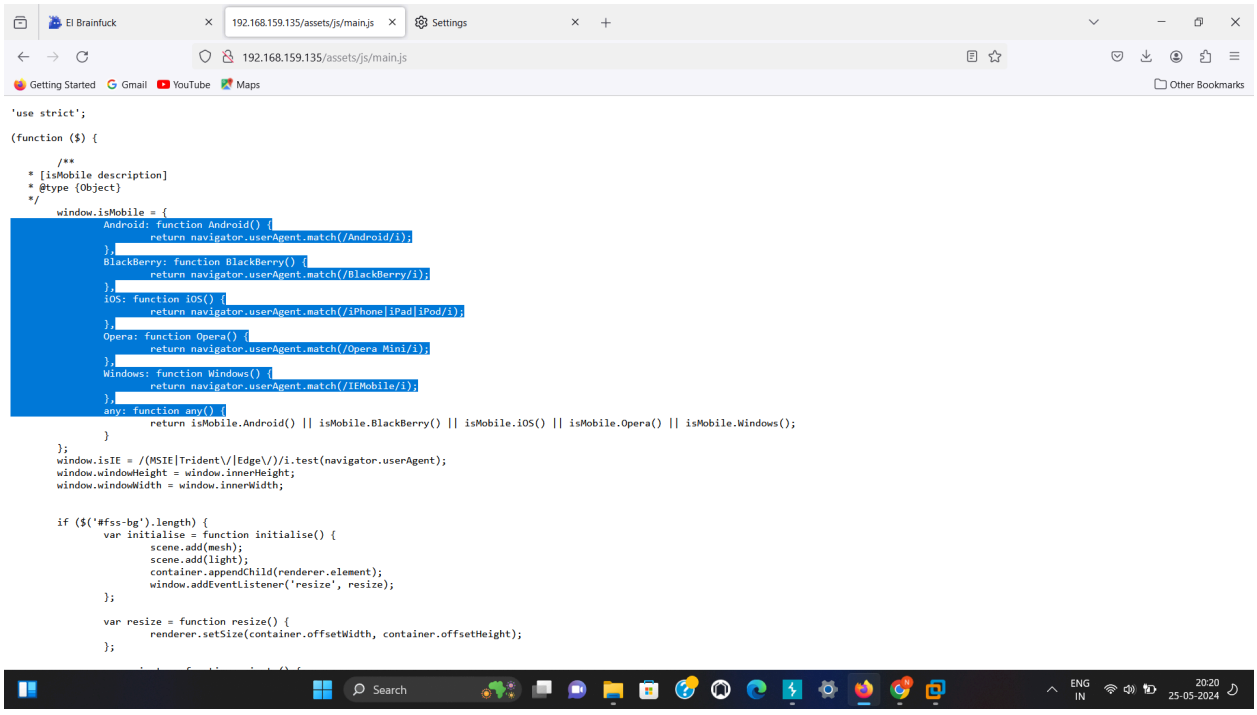
Directory listing for /assets/vendors/

- jquery/
- bootstrap/
- flat-surface-shader/
- jquery.countdown/
- particles.js/
- quietflow/
- swiper/
- vegas/
- waterpipe/
- YTPlayer/

Step 11 - Find more information about the site using dirbuster .

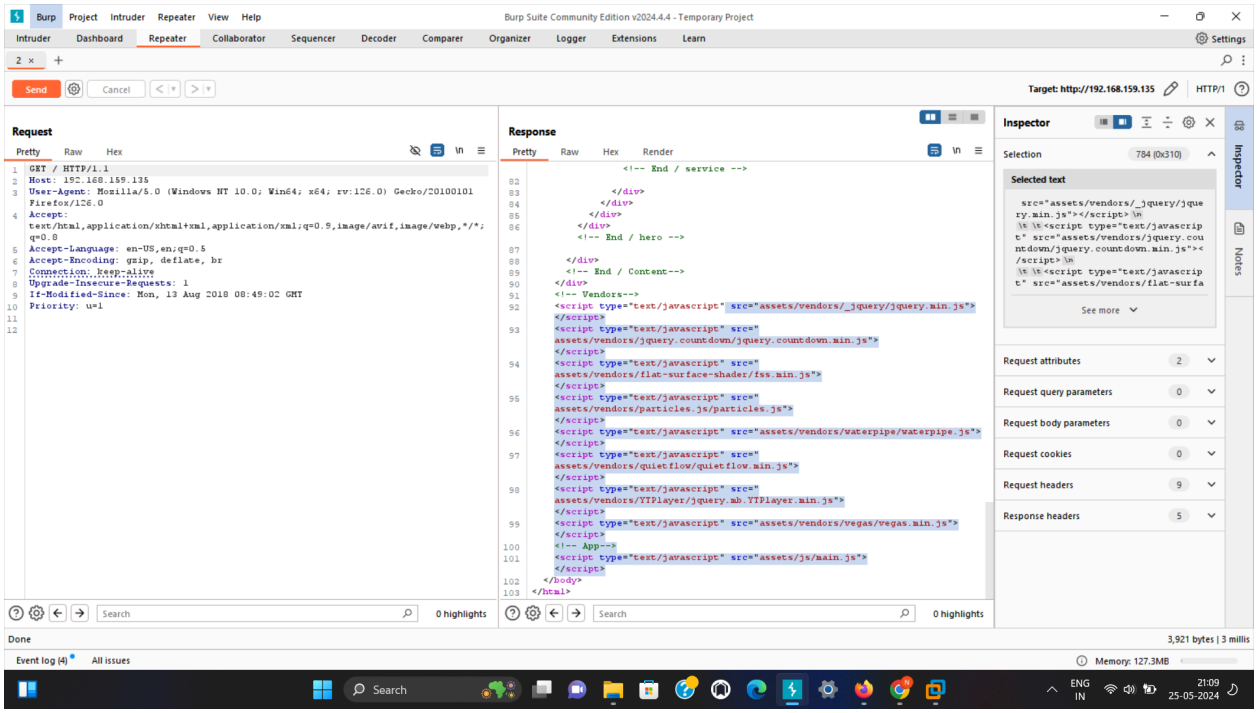


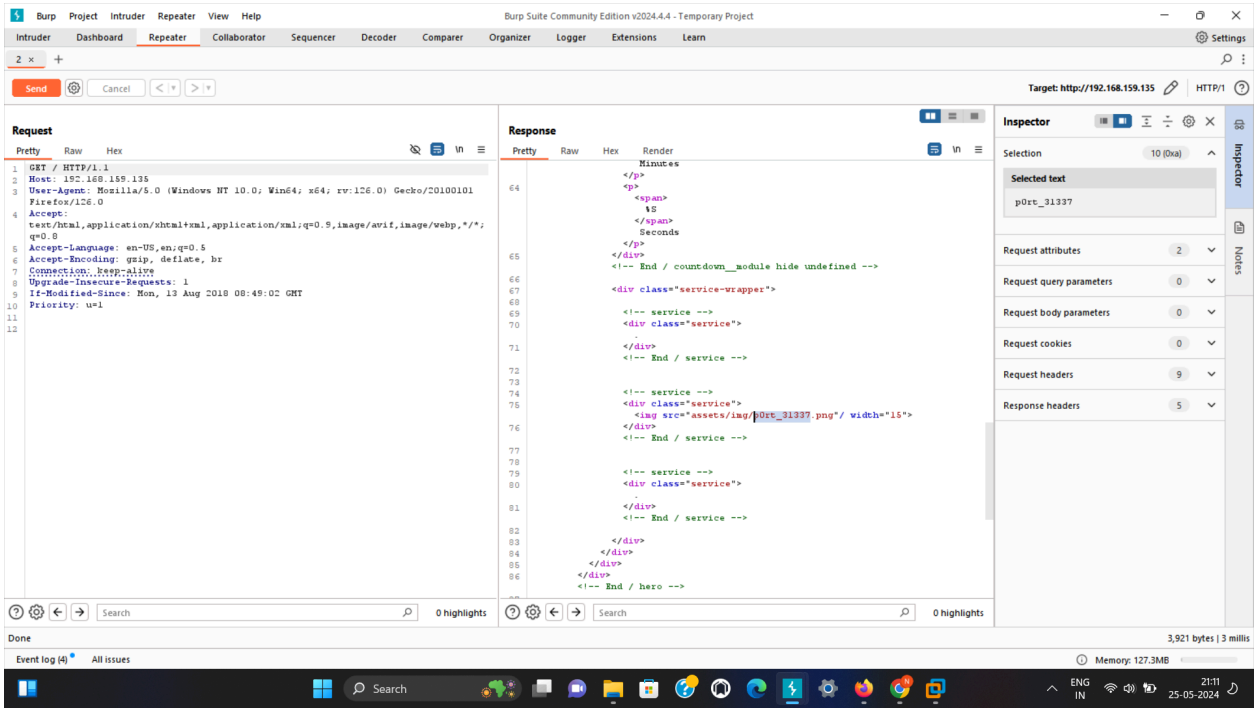
As it shows that there are directories exists in the website which can provide more information about the target .It can found more vulnerable and machine can be easily compromised , it can also more information about the admin page , organization as well . So sensitive information like content inside the file must be encrypted , less use of directories ways must be followed .



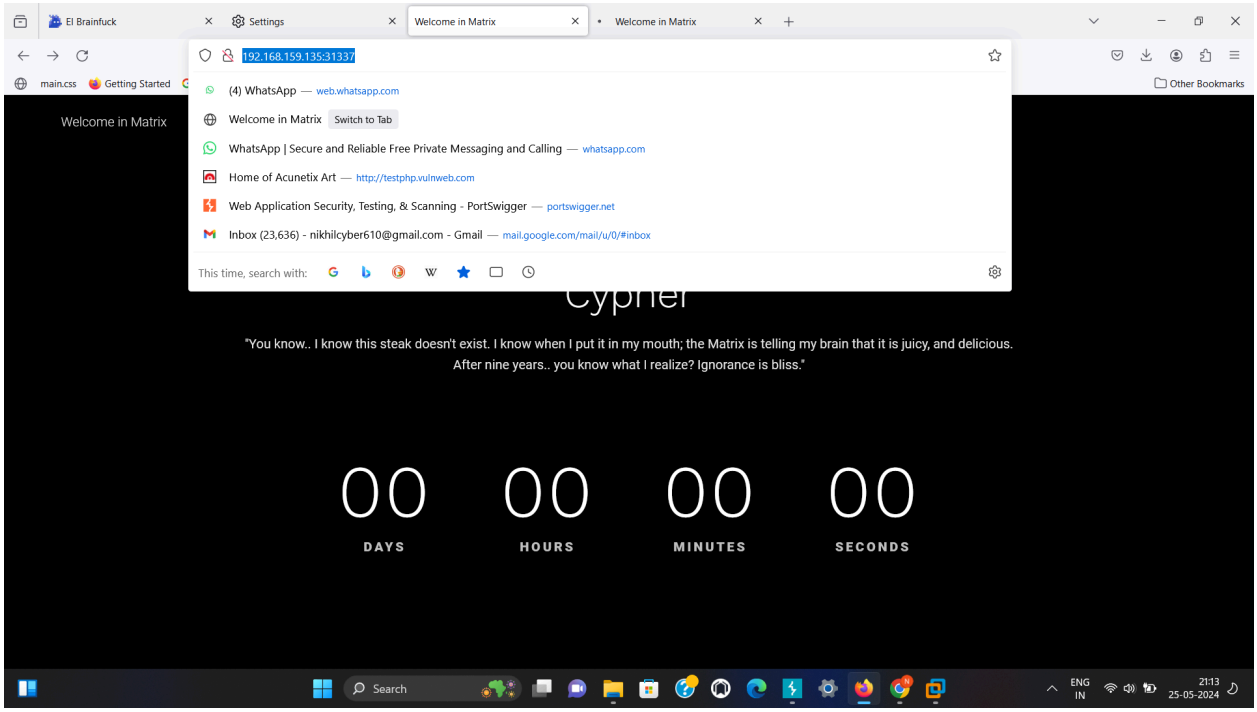
We had gathered information about the devices used by the victim .this can be helpful and useful for the attacker in order to gain unauthorized access .

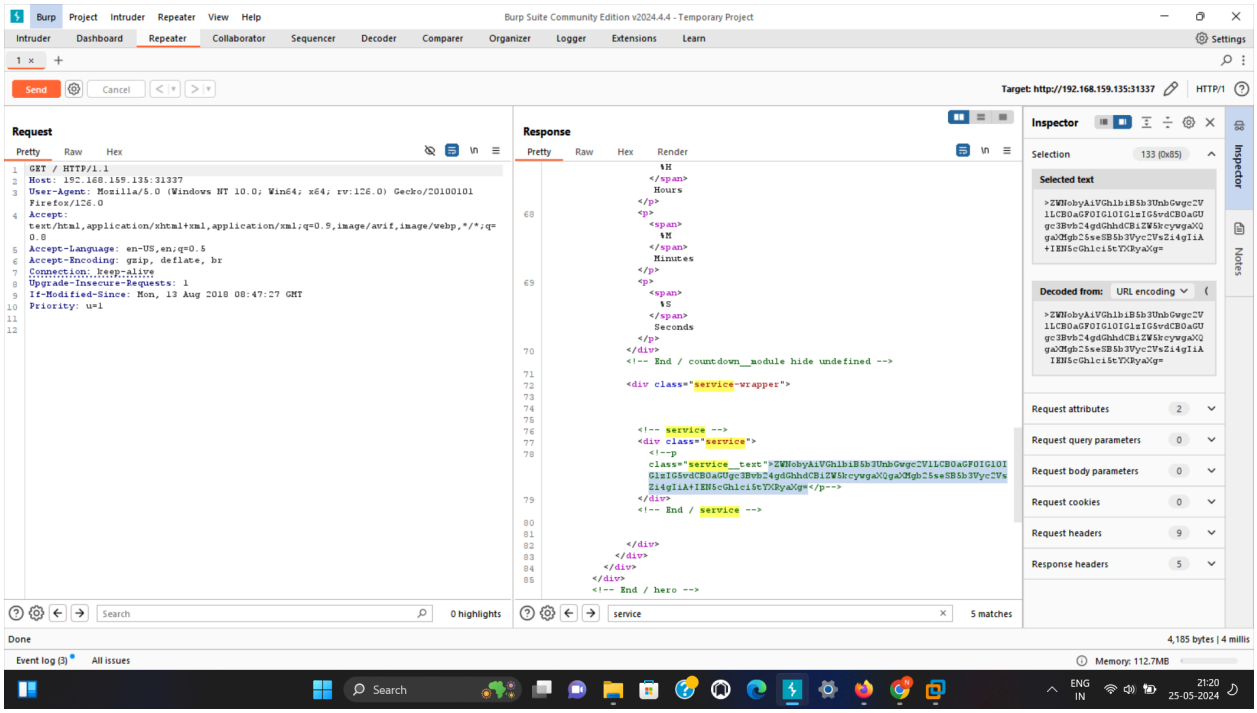
Step 12 - We had also analyzed through the burp suite in order to anyalze more information for the exploitation process .



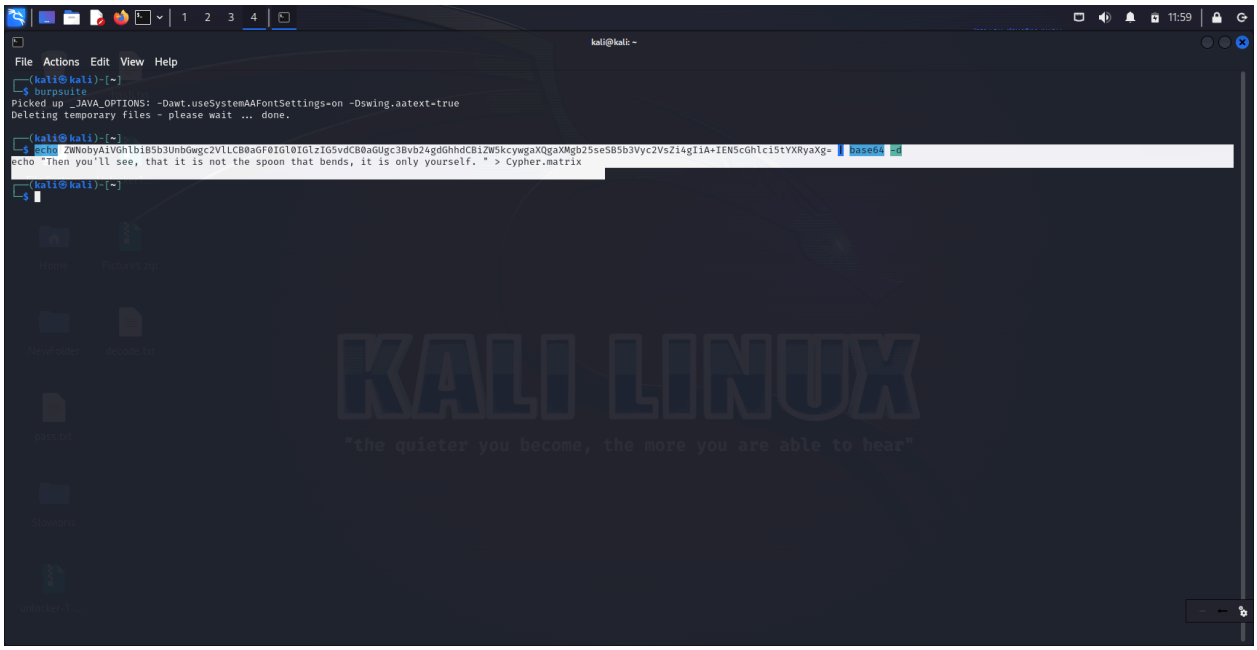


We got an footprint of 31337 through burp suite . Which can be a port number .

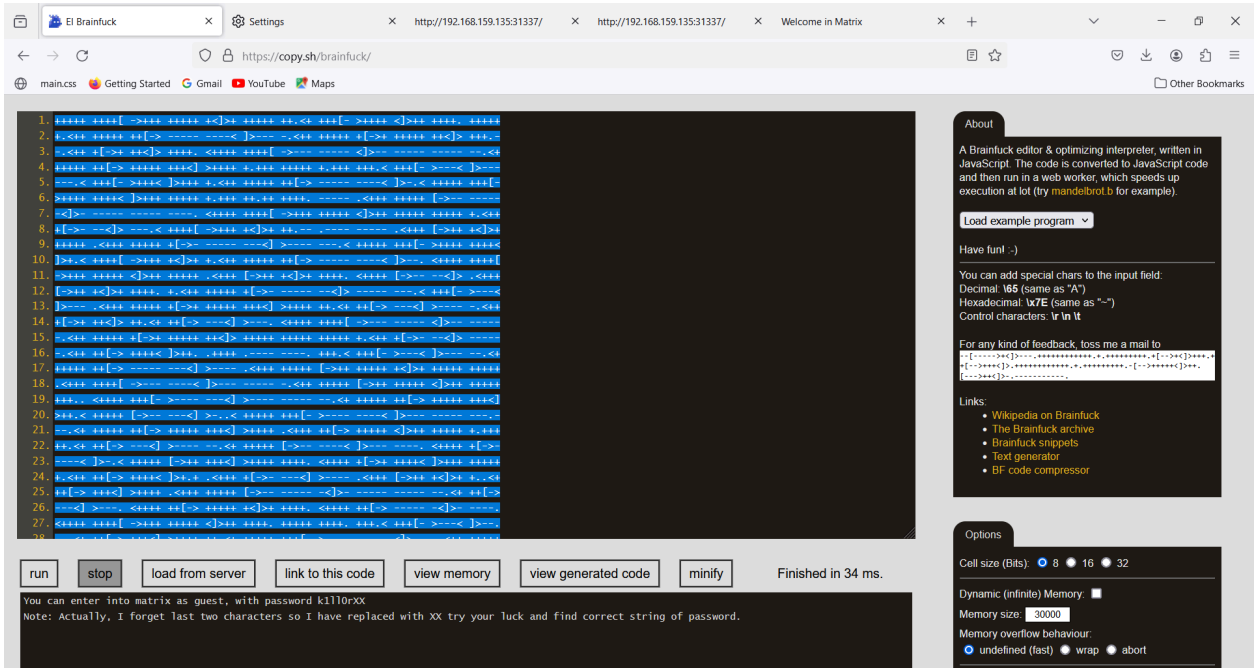




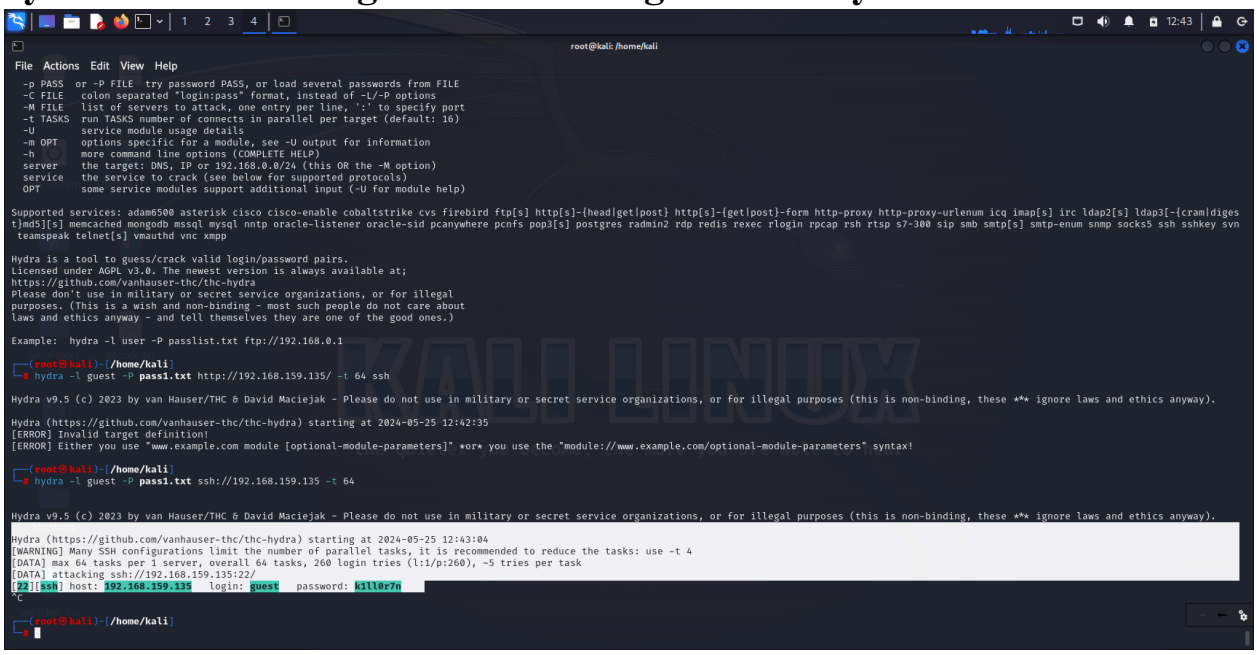
Step 13 - simply the convert base64 code into plain text .



Step 15 - Simply convert the base64 code into plain text .



Step 16 - Finally we had sueccesffuly exploit the vuberbilites using hydra tool now we gain access using dictionary attack .



Finally we comprise the machine , and Also solved the challenge as well .

—----------END of Proof of concept —----------

The interface gave us direct access to the data and the ability to extract a list of users on the system with the associated password hash values (Figure 8).

Action	id	name	type	pw	last_login	wysiwyg
X	1	admin	1	a7d114b3072535f10a201aa8b1d6f073f848c6725e3c0667d5	1366376562	0
X	2	joe	0	0af12a0c93eba9edf940ad455df837b5afaaa510501424ccae	1366375461	0
X	3	mike	0	8e0ab72cecbe72c9e3f56adb3a909ffa655fc480e5480a2d3a	1366375306	0
X	4	alan	0	06dda79ec74207e73454bfa477c302ef88214f2905331e04ee	1366632889	0

» Create a View with name from this query.

Figure 8 – Lack of additional access controls allows an attacker to retrieve usernames and password hashes from the "userdata" database.

After examination of the values, we found that the hashes did not conform to any standard format. Using a copy of the "**phpselitecms**" software, we examined the source code to determine exactly how this value is produced. Through this process we were able to identify the function responsible for hashing of the account passwords.

```
function generate_pw_hash($pw)
{
 $salt = random_string(10,'0123456789abcdef');
 $salted_hash = sha1($pw.$salt);
 $hash_with_salt = $salted_hash.$salt;
 return $hash_with_salt;
}
```

Figure 9 – Source code review leads to the discovery of the password hash generation algorithm.

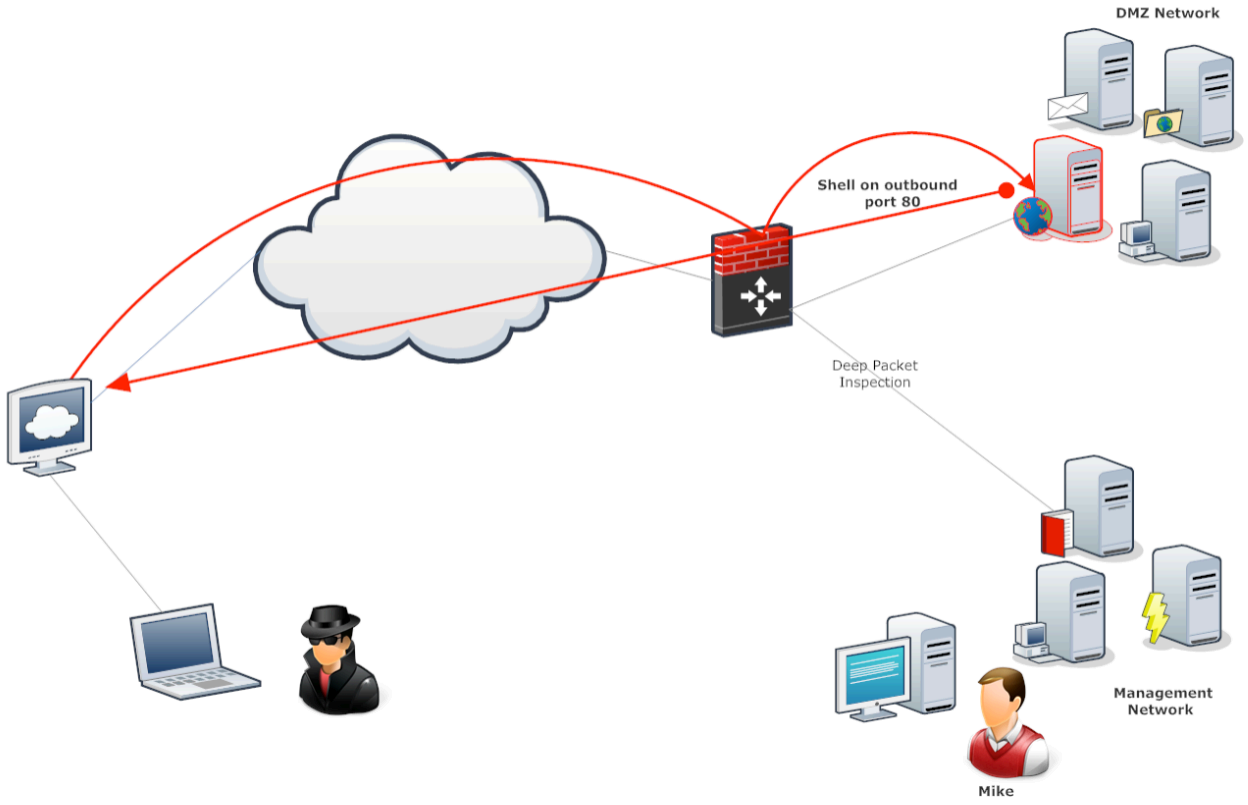
With the newly-acquired knowledge of the password hashing format and the use of a randomly generated 10 character salt value, we were able to easily convert the recovered hashes into their salted SHA1 equivalent and conduct a brute-force attack.

This effort resulted in the recovery of two plaintext passwords. Although these values were not immediately useful, they were retained in hope that they may have been re-used on other systems within the organization.

Interactive Shell to Admin Server

The previously discovered SQLite Manager software was found to be vulnerable to a well-known code injection vulnerability[3]. Successful exploitation of this vulnerability results in shell access to the underlying system in the context of the webserver user. Using a modified public exploit, we were able to obtain limited interactive access to the webserver. Please see Appendix A for more information.

3 http://www.exploit-db.com/exploits/24320



DMZ Firewell must be maintained in order to capture the traffic around to track the logs and tracks .

Figure 12 - Web Server Compromise

Administrative Privilege Escalation

With interactive access to the underlying operating system of the administrative webserver obtained, we continued with the examination of the system searching for ways to escalate privileges to the administrative level. We found that the system was vulnerable to a local privilege escalation exploit[4], which we were able to utilize successfully. Please see Appendix A for more information.

4 http://www.exploit-db.com/exploits/18411


VIRTUAL COMPANY
LET'S WE HACK
PLAY WITH THE BRAIN
letswehack@gmail.com

It was determined that while these steps would be possible, they would be considered outside the scope of the current engagement. It was demonstrated that a total compromise of the domain had been accomplished with a complete loss of integrity for all local systems.

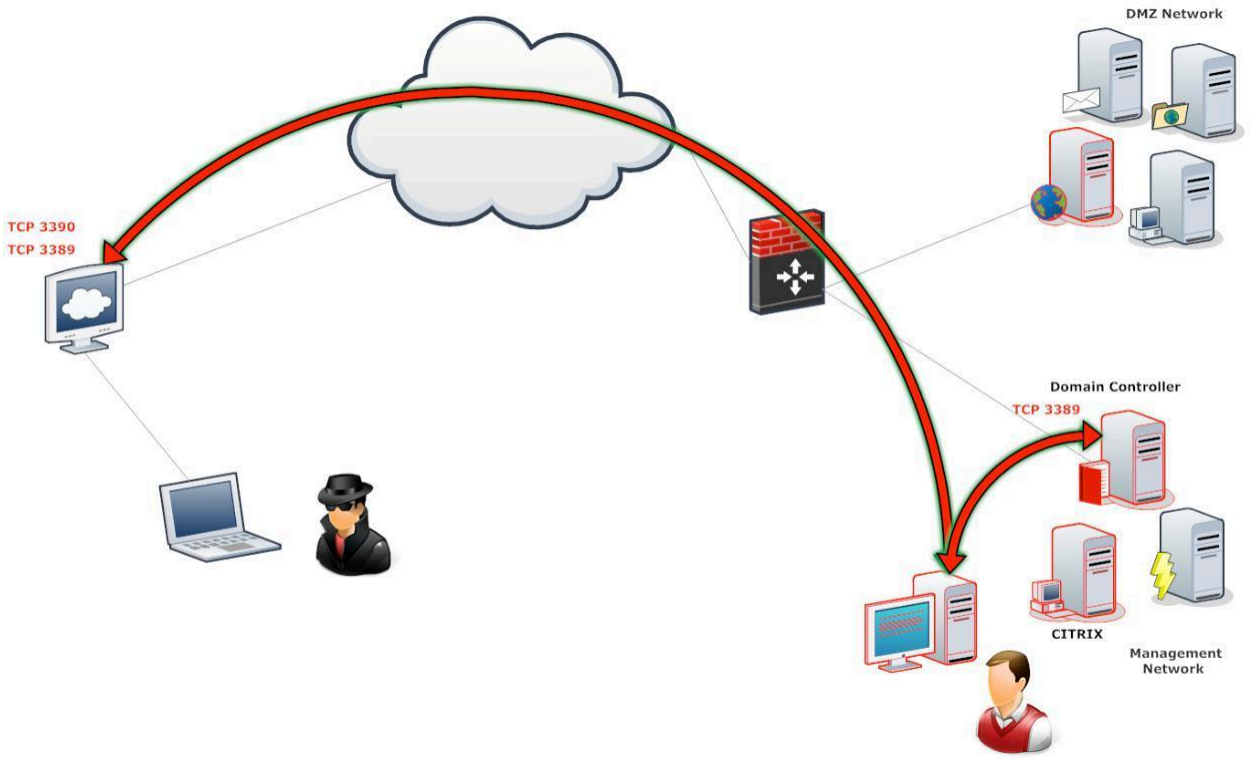


Figure 36 - Full Domain Compromise

Conclusion

Following a comprehensive penetration test, several crucial conclusions emerge regarding the security landscape of the project. Throughout the assessment, a myriad of vulnerabilities were unearthed, ranging from inadequate authentication protocols to insufficient input validation mechanisms, as evidenced in components like HTTP File Server (HFS), AuraCMS, and Geo++ GNCASTER. These vulnerabilities pose a substantial risk to the organization, exposing it to a multitude of potential threats, including unauthorized access, data breaches, and denial of service (DoS) attacks. The inherent exploitation potential of these vulnerabilities is alarming, as they could be leveraged by malicious actors to execute arbitrary code, compromise sensitive data, or disrupt critical services. However, amidst these challenges lies an opportunity for improvement and fortification. Recommendations have been meticulously crafted to address these vulnerabilities and bolster the project's security posture. From implementing robust authentication mechanisms like HTTP Digest Authentication to fortifying input validation measures against injection attacks, the path to resilience is clear. Moreover, the journey towards enhanced security is not a destination but a continuous evolution. By adopting a proactive stance towards security, including regular assessments, vigilant monitoring, and ongoing training initiatives, the organization can fortify its defenses, mitigate emerging threats, and safeguard its assets and reputation. In conclusion, by embracing these recommendations and committing to a culture of security excellence, the project can chart a course towards a safer, more resilient future, ensuring the protection of its invaluable resources and the trust of its stakeholders.

Recommendations

Due to the impact to the overall organization as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high level items are important to mention.

Offensive Security recommends the following:

1. **Ensure that strong credentials are use everywhere in the organization**. The compromise of MegaCorp One system as drastically impacted by the use of weak passwords as well as the reuse of passwords across systems of differing security levels. NIST SP 800-11[9] is recommended for guidelines on operating an enterprise password policy. While this issue was not widespread within MegaCorp One, it was still an issue and should be addressed.
2. **Patch Management**: Establish a robust patch management process to promptly apply security patches and updates for all software components, including web servers, applications, and third-party modules. Regularly monitor vendor advisories and security mailing lists for patch releases.
3. **Authentication Mechanisms**: Ensure that proper authentication mechanisms, such as HTTP Digest Authentication, are enforced consistently across all web server endpoints. Implement strong password policies, multi-factor authentication (MFA), and session management controls to mitigate the risk of unauthorized access.
4. **Input Validation**: Implement strict input validation mechanisms to sanitize and validate user-supplied data, especially in HTTP headers and request parameters. Use secure coding practices and input validation libraries/frameworks to prevent injection attacks, such as XSS and SQL injection.
5. **Access Controls**: Review and strengthen access controls for sensitive resources and administrative functionality. Utilize role-based access control (RBAC) to enforce the principle of least privilege and restrict access to authorized users based on their roles and responsibilities.
6. Logging and Monitoring: Enhance logging and monitoring capabilities to detect and respond to security incidents effectively. Implement centralized logging solutions and intrusion detection systems (IDS) to monitor for suspicious activities, anomalous behavior, and unauthorized access attempts.
7. **Conduct regular vulnerability assessments**. As part of an effective organizational risk management strategy, vulnerability assessments should be conducted on a regular basis. Doing so will allow the organization to determine if the installed security controls are properly installed, operating as intended, and producing the desired outcome. Please consult NIST SP 800-30[11] for guidelines on operating an effective risk management program.

---

9 http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf

10 http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf

11 http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-30-Rev.%201

Risk Rating

The overall risk identified to https://download.vulnhub.com/matrix/Machine_Matrix.zip as a result of the penetration test is **High and Informational .** It is reasonable to believe that a malicious entity would be able to successfully execute an attack against download.vulnhub.com through targeted attacks.

Appendix A: Vulnerability Detail and Mitigation

Risk Rating Scale

In accordance with NIST SP 800-30, exploited vulnerabilities are ranked based upon likelihood and impact to determine overall risk.

Default or Weak Credentials

Rating: High

Description: An externally exposed administrative interface is only protected with a weak password.

Impact: Using common enumeration and brute-forcing techniques, it is possible to retrieve the administrative password for the SQLite Manager web interface. Due to the lack of any additional authentication mechanisms, it is also possible to retrieve all user password hashes in the underlying database. Successful retrieval of plaintext passwords could allow further compromise of the target environment if password reuse is found to exist.

Remediation: Ensure that all administrative interfaces are protected with complex passwords or passphrases. Avoid use of common or business related words, which could be found or easily constructed with the help of a dictionary.

Patch Management

Rating: High

Description: MegaCorp One's external and internal environments contain a number of unpatched systems and application.

Impact: A combination of weak authentication and unpatched hosts, which contain known vulnerabilities with publicly available exploits, allows an attacker to gain unauthorized access to a large number of MegaCorp One's assets. Specifically, discovered instance of SQLite Manager is vulnerable to a remote code execution vulnerability and the underlying host also contains a local privilege escalation vulnerability, which can easily be leveraged to compromise the externally exposed host entirely. This appears to be an indication of an insufficient patch management policy and its implementation.

Remediation: All corporate assets should be kept current with latest vendor-supplied security patches. This can be achieved with vendor-native tools or third-party applications, which can provide an overview of all missing patches. In many instances, third-party tools can also be used for patch deployment throughout a heterogeneous environment.

Public IP : 192.168.215.207
Host machine's IP :**192.168.247.1**
Kali Machine's IP:192.168.159.130
Seattle VM's IP : 104.21.42.126 , 172.67.162.8 , 192.168.159.135

Thank You

Nikhil Tyagi