**Final VAPT Report**



PREPARED BY: Thirumaal.T

Submitted To: Zero Bank

Submission Date: 10-11-2022

**TABLE OF CONTENTS**

[**EXECUTIVE SUMMARY** 4](#_Toc118889510)

[**HIGH LEVEL ASSESSMENT OVERVIEW** 5](#_Toc118889511)

[Areas for Improvement 5](#_Toc118889512)

[Recommendations 5](#_Toc118889513)

[**SCOPE** 7](#_Toc118889514)

[Project Scope 7](#_Toc118889515)

[Network Information and machines 7](#_Toc118889516)

[**TESTING METHODOLOGY** 8](#_Toc118889517)

[**CLASSIFICATION DEFINITIONS** 9](#_Toc118889518)

[Risk Classifications 9](#_Toc118889519)

[Exploitation Likelihood Classifications 9](#_Toc118889520)

[Business Impact Classifications 10](#_Toc118889521)

[Remediation Difficulty Classifications 10](#_Toc118889522)

[**ASSESSMENT FINDINGS** 11](#_Toc118889523)

[**RISK IDENTIFICATION** 12](#_Toc118889524)

I. Vulnerability Finding for Zero Bank’s Network 12

1. Centralized Directory Service 12

i. Kerberos Information Disclosure 12

ii. SMB Information Disclosure 15

iii. User information extraction from LDAP 17

2. Server Logging Service 19

3. Website 22

4. Remote Access Service 24

II. Vulnerability Finding Mutilldae Webapplication 27

1. Extract Data 27

2. Cross-Site Scripting 30

3. Cross-Site Request Forgery 33

4. Insecure Direct object 36

III. Vulnerability Finding Mutilldae Database 39

Extracting Data SQL injection manual and automated 39

[**APPENDIX A - TOOLS USED** 47](#_Toc118889525)

[**APPENDIX B - ENGAGEMENT INFORMATION** 48](#_Toc118889526)

[Client Information 48](#_Toc118889527)

[Version Information 48](#_Toc118889528)

[Contact Information 48](#_Toc118889529)

# 

# EXECUTIVE SUMMARY

Our team performed a security assessment of the internal corporate network of Zerobank on 10-11-2022. Our team’s penetration test simulated an attack from an external threat actor attempting to gain access to systems within the Zerobank corporate network. The purpose of this assessment was to discover and identify vulnerabilities in Zerobank’s infrastructure and suggest methods to remediate the vulnerabilities. Our team has identified a total of 9 vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITICAL** | **HIGH** | **MEDIUM** | **LOW** |
| **2** | **5** | **2** | **1** |

The highest severity vulnerabilities give potential attackers the opportunity to hack the system, sell company data in black market, etc. In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope. Any changes made to the environment during the period of testing may affect the results of the assessment.

# HIGH LEVEL ASSESSMENT OVERVIEW

## Areas for Improvement

Our team recommends Zerobank takes the following actions to improve the security of the network. Implementing these recommendations will reduce the likelihood that an attacker will be able to successfully attack Zerobank’s information systems and/or reduce the impact of a successful attack. CDA close unused port 88, don’t configure rsyslog or while configuring rsyslog specify server’s IP. Update the OS and software for remote access, use of latest code for web-application.

### Recommendations

Our team recommends ZEROBANK take the following actions as soon as possible to minimize business risk. For **zerobank’s network** in **CDA** I recommend to remove kerberos from windows server 2019. **Sever logging** does not have any misconfiguration but if the attacker knows the ubuntu’s rsyslog configuration file is configured to get logs it is a potential dangerous and this is also possible if the attacker is in same network.

**Web-application** the infrastructure is not responsible that is from the client or user side they should check that we are visiting is it correct or not, easily with typo from user attacker can do Social Engineering attack, the user should check the website is http or https the website the attacker used will be in http so the information is easily readable in plain text. In **remote access** the Windows 7 machine is outdated and it will not prompt the error in the Freesshd 2.1.3 software, this software consists of authbypass vulnerability which will allow any one without entering password to login via ssh I recommend to update latest OS and the software.

For **Mutilldae Website** first exploitation is SQL injection using malicious code attacker can gain access to information, to prevent this train and maintain awareness, don’t trust any user input, use whitelists not blacklists, adopt the latest technologies. Second **Cross-Site Scripting (XSS)** attacks are a type The OWASP ESAPI project has produced a set of reusable security components in several languages, including validation and escaping routines to prevent parameter tampering and the injection of XSS attacks. Sanitize user input, validate to catch potentially malicious user-provided input. Encode output to prevent potentially malicious user-provided data from triggering automatic load and execute behavior by a browser.

**Cross-Site Request Forgery (CSRF)** is an attack that forces an end user to execute unwanted actions on a web application in which they’re currently authenticated. SameSite is a cookie attribute (similar to HTTPOnly, Secure etc.) which aims to mitigate CSRF attacks. Verifying Origin with Standard Headers, Identifying the Target Origin these are the security measurement. **Insecure direct object reference (IDOR)** is a type of access control vulnerability in digital security. This can occur when a web application or application programming interface uses an identifier for direct access to an object in an internal database but does not check for access control or authentication. Avoid Exposing Direct Object References, use an Indirect Reference Map, check user access at the Data-Object level, protecting against Directory Traversal. For **Mutildae database** same information provided to SQL injection is recommended.

# SCOPE

## Project Scope

**1. Network Infrastructure of Zero Bank**

1. Centralized Directory Service
2. Server Logging Service
3. Zero Bank's Website
4. Remote Access Service

**2. Mutillidae Web Application**

1. SQL Injection
2. Cross-site scripting
3. Cross-site Request Forgery
4. Insecure Direct object

**3. Mutillidae Database**

* SQL Injection (SQL map)

## Network Information and machines

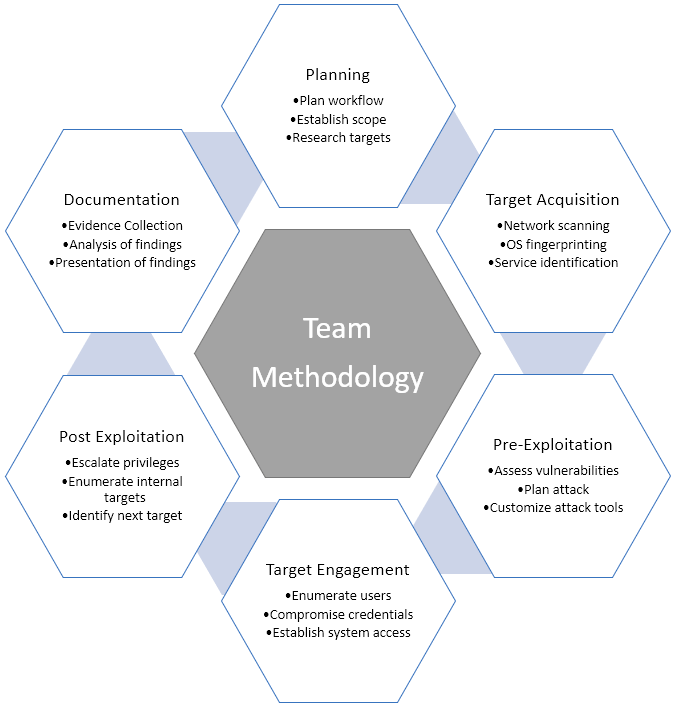
|  |  |  |  |
| --- | --- | --- | --- |
| **Sno.** | **Machines** | **Information** | **IP’s** |
| 1. | Centralized Directory | Windows Server 2019 (domain.local) | 192.168.50.58 |
| 2. | Log Server | Ubuntu | 192.168.50.111 |
| 3. | Website of Zero Bank | http://zero.webappsecurity.com/login.html | **-** |
| 4. | Windows Client (PC: WIN-7-Victim) | Username: STACK-WIN7 | 192.168.50.60 |
| 5. | Meta Sploitable (Multtildae Web application) | http://192.168.137.60/mutillidae/index.php | 192.168.137.63 |
| 6. | Meta Sploitable (Multtildae Web database) | http://192.168.137.60/mutillidae/index.php?page=user-info.php | 192.168.137.63 |

Table 1: Provided Information

# TESTING METHODOLOGY

Our ’s testing methodology was split into three phases: *Reconnaissance*, *Target Assessment*, and *Execution of Vulnerabilities*. During reconnaissance, we gathered information about ZEROBANK’s network systems. Our team used port scanning and other enumeration methods to refine target information and assess target values. Next, we conducted our targeted assessment. Our team simulated an attacker exploiting vulnerabilities in the ZEROBANK network. Our team gathered evidence of vulnerabilities during this phase of the engagement while conducting the simulation in a manner that would not disrupt normal business operations.

The following image is a graphical representation of this methodology.



# CLASSIFICATION DEFINITIONS

## Risk Classifications

|  |  |  |
| --- | --- | --- |
| **Level** | **Score** | **Description** |
| **Critical** | **10** | The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed. |
| **High** | **7-9** | The vulnerability poses an urgent threat to the organization, and remediation should be prioritized. |
| **Medium** | **4-6** | Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible. |
| **Low** | **1-3** | The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible. |
| **Informational** | **0** | These findings have no clear threat to the organization, but may cause business processes to function differently than desired or reveal sensitive information about the company. |

## Exploitation Likelihood Classifications

|  |  |
| --- | --- |
| **Likelihood** | **Description** |
| **Likely** | Exploitation methods are well-known and can be performed using publicly available tools. Low-skilled attackers and automated tools could successfully exploit the vulnerability with minimal difficulty. |
| **Possible** | Exploitation methods are well-known, may be performed using public tools, but require configuration. Understanding of the underlying system is required for successful exploitation. |
| **Unlikely** | Exploitation requires deep understanding of the underlying systems or advanced technical skills. Precise conditions may be required for successful exploitation. |

## Business Impact Classifications

|  |  |
| --- | --- |
| **Impact** | **Description** |
| **Major** | Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage. |
| **Moderate** | Successful exploitation may cause significant disruptions to non-critical business functions. |
| **Minor** | Successful exploitation may affect few users, without causing much disruption to routine business functions. |

## Remediation Difficulty Classifications

|  |  |
| --- | --- |
| **Difficulty** | **Description** |
| **Hard** | Remediation may require extensive reconfiguration of underlying systems that is time consuming. Remediation may require disruption of normal business functions. |
| **Moderate** | Remediation may require minor reconfigurations or additions that may be time-intensive or expensive. |
| **Easy** | Remediation can be accomplished in a short amount of time, with little difficulty. |

## 

# ASSESSMENT FINDINGS

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Finding** | **Risk Score** | **Risk** |
| 1 | Social Engineering | **10** | **Critical** |
| 2 | SSH Authbypass | **10** | **Critical** |
| 3. | Ldap enumeration | **9** | **High** |
| 4. | Cross-site scripting request forgery | **9** | **High** |
| 5. | SMB enumeration | **8** | **High** |
| 6. | SQL Injection | **8** | **High** |
| 7. | Insecure direct object | **8** | **High** |
| 8 | Kerberos enumeration | **7** | **Medium** |
| 9. | Cross-site Scripting | **6** | **Medium** |
| 10. | Server logging | **0** | **Low** |

Table 2: (Sorting by descending risk score)

# RISK IDENTIFICATION

1. **Vulnerability Finding for Zero Bank’s Network**
2. **Centralized Directory Service**
3. Kerberos Information Disclosure

**Description:** Nessus was able to retrieve the realm name and/or server time of the remote Kerberos server.

**Vulnerability:** Kerberos Domain User Enumeration.

|  |  |
| --- | --- |
| **MEDIUM RISK (6/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Mild** |
| **Remediation Difficulty** | **Easy** |

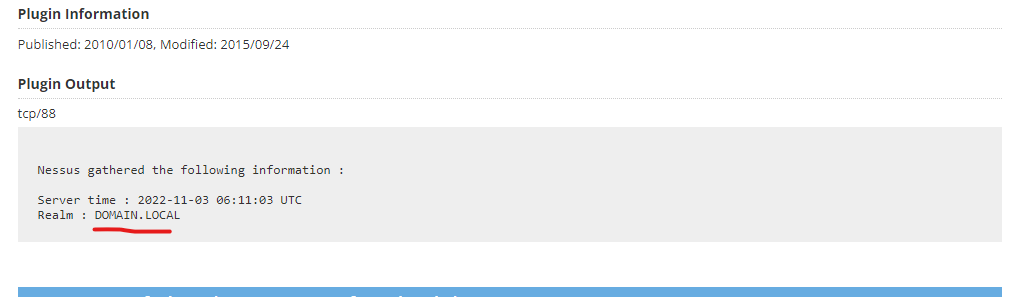
**Security Implications**

This is leaking the domain-name information and username information by that hacker can gain access. This is a huge risk to the company, we can use firewall to filter, IPS or IDS, close the port 88, antivirus can prevent from exploiting.

**Analysis**

1. Nessus

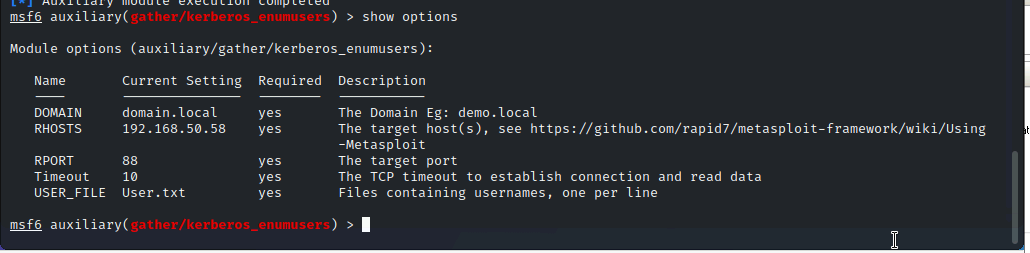
Nessus analysis shows the domain information, server time port. From the port 88 we can know it is used for Kerberos service.



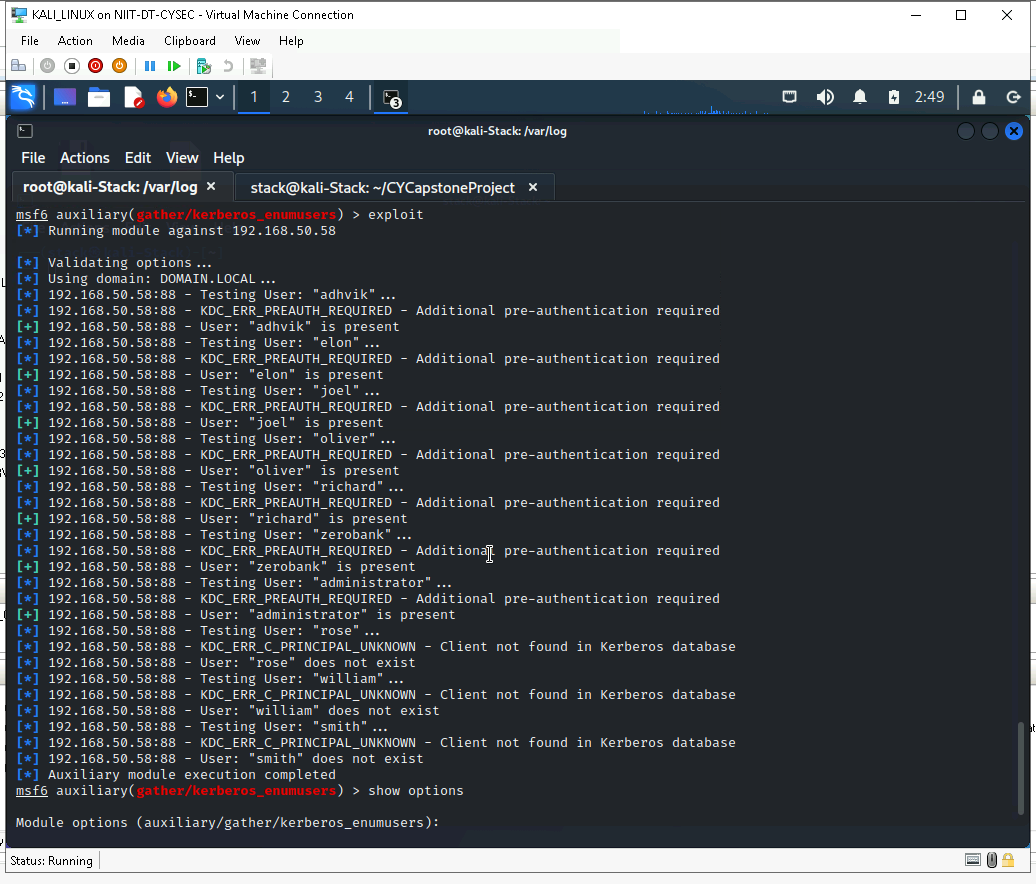
***Figure 1.1.1****: Nessus analysis*

1. Penetration Testing

To gather username information **kerberos\_enumusers** exploit is used. First is to set RHOST IP and the IP obtained from Nmap scan. Then is to set username file and that username file contains usernames. After setting parameters then exploit command is used. By this username of the machine centralized directory can be retrieved.



***Figure 1.1.2:*** *exploit enumusers*



***Figure 1.1.3:*** *Gathering Username by exploitation*

1. SMB Information Disclosure

**Description:** The information of username and password can be obtained by this process.

**Vulnerability:** SMB Enumeration

|  |  |
| --- | --- |
| **HIGH RISK (8/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

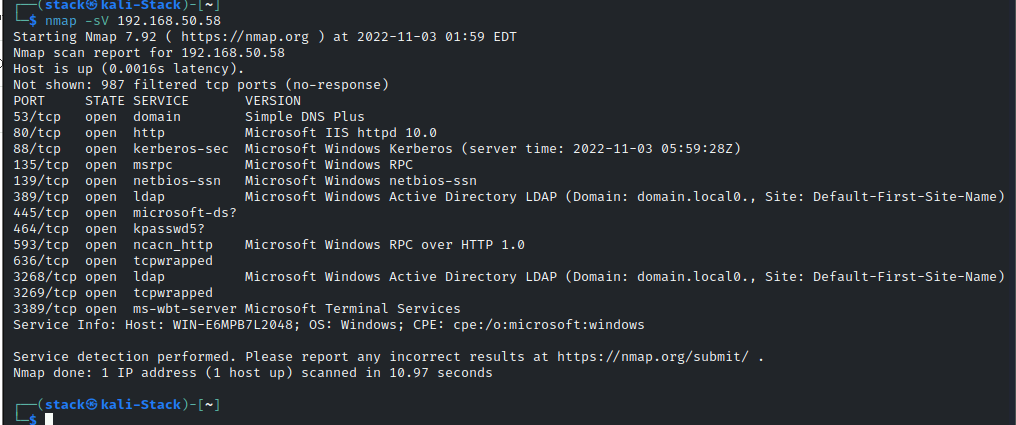
**Security Implications**

By the Kerberos username is obtained, from samba password can be obtained by that hacker can gain access. This is a huge risk to the company, we can use firewall to filter, IPS or IDS, close the port 445, antivirus can prevent from exploiting.

**Analysis**

1. Nmap Scan

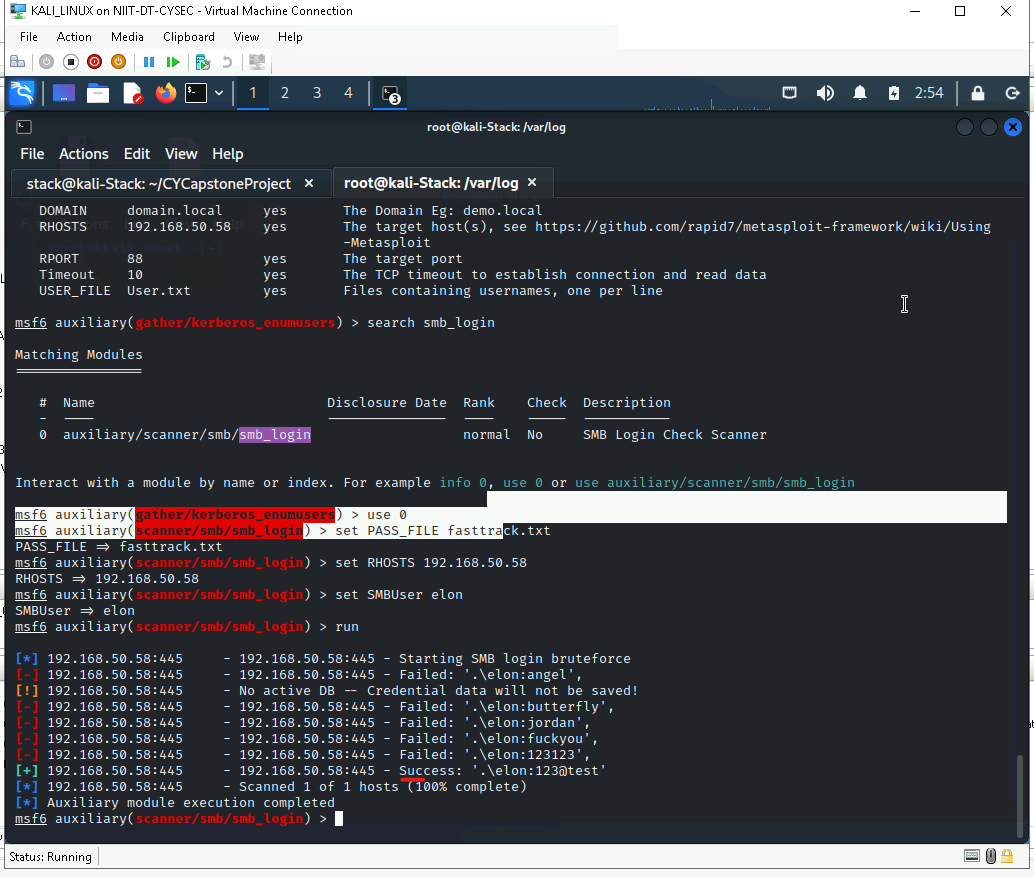
The Nmap scan shows SMB port 445 is opened from this we can do smb enumeration.



***Figure 1.2.1:*** *Nmap Scan*

1. Pen testing

To get the password for gathered username exploit used is smb\_login this will enumerate each and every password provided with username provided. First is to set RHOST IP, set username, set password file and then exploit can be run. Now we can obtain password for the user.

****

***Figure 1.2.2:*** *Password obtained*

1. User information extraction from LDAP

**Description:** I t is possible to discover information about the remote LDAP server. In this user’s information are extracted eg. Email, phone no., etc.

**Vulnerability:** LDAP Enumeration.

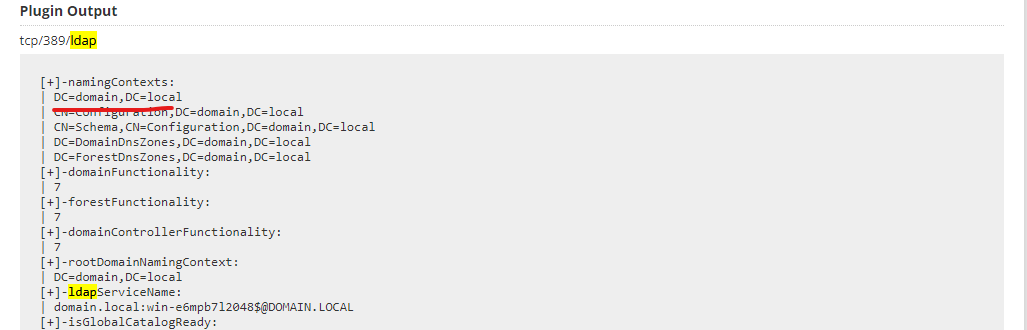
|  |  |
| --- | --- |
| **HIGH RISK (9/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

**Security Implications**

By the Kerberos username is obtained, from samba password is obtained, by ldap domain name is obtained from this information we can get the user’s personal information. This is a huge risk to the company, we can use firewall to filter, IPS or IDS, close the port 389, antivirus can prevent from exploiting.

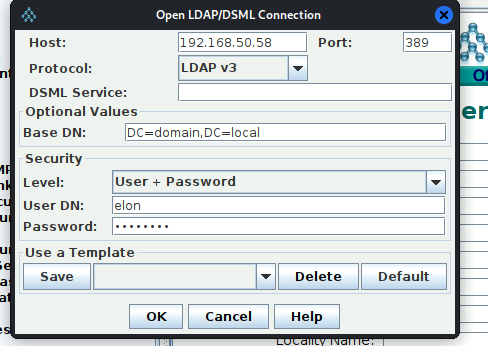
**Analysis**

1. Obtained domain-name from Nessus

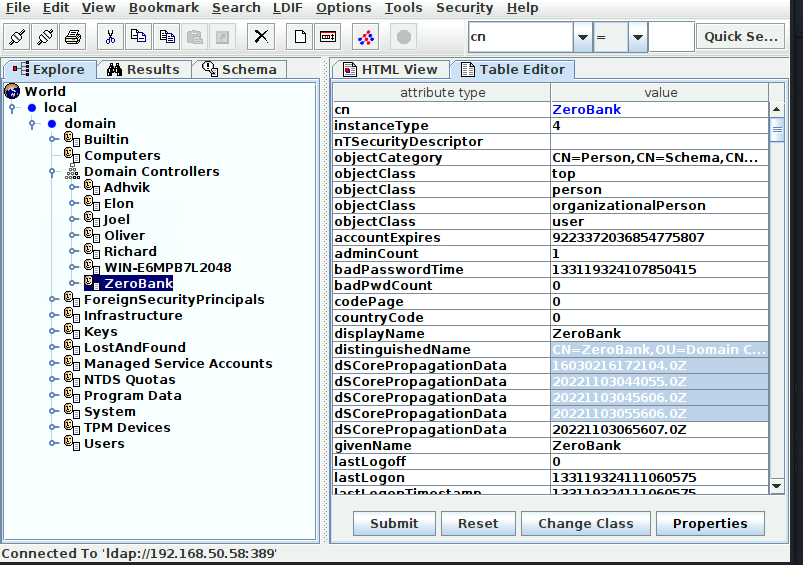
****

***Figure 1.3.1:*** *Domain-Name Obtained*

1. Jxplorer

****

***Figure 1.3.2:*** *Jxplorer configuration*

****

***Figure 1.3.3:*** *User Information*

**2. Server Logging Service**

**Description:** In the server logging service no vulnerability found but we can see the log generating from machine to server.

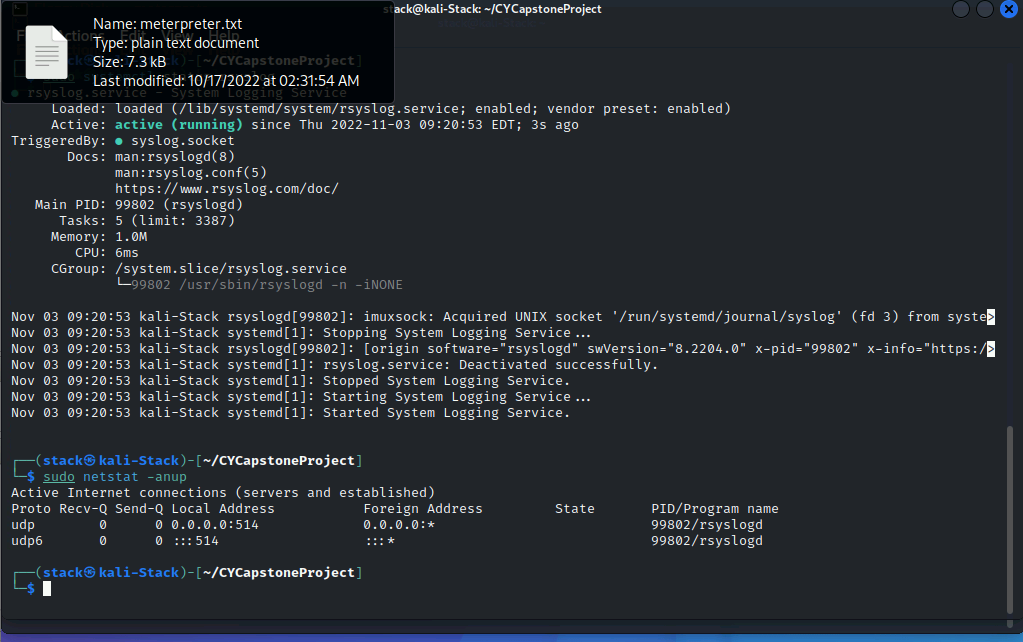
**Vulnerability:** No vulnerability detected.

|  |  |
| --- | --- |
| **NO RISK (0/10)** | |
| **Exploitation Likelihood** | **No** |
| **Business Impact** | **Negligible** |
| **Remediation Difficulty** | **Not needed** |

**Analysis:**

1. Rsyslog status

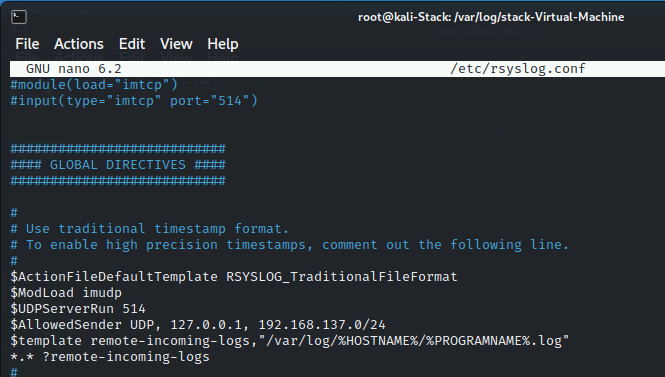
Here rsyslog is running and the port is opened to get the logs. Netstat command is used to check for rsyslog port and systemctl for status.

****

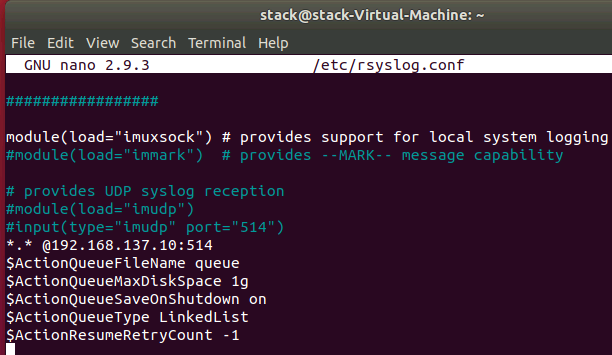
***Figure 2.1:*** *Rsyslog Status*

1. Rsyslog server machine (Kali) and client machine configuration (Ubuntu)

To get logs to server kali we need to add configurations.

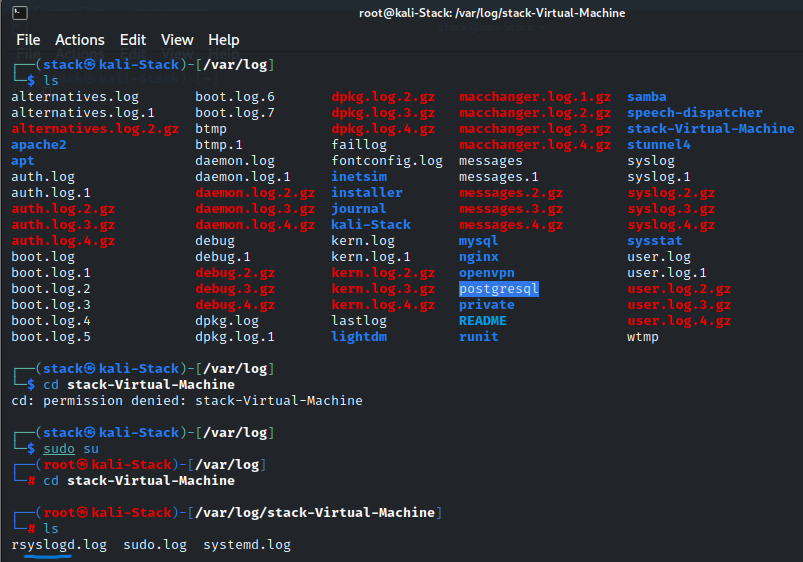


***Figure 2.2:*** *Kali rsyslog.conf*



***Figure 2.3:*** *Ubuntu rsyslog.conf*

1. Logs generated

****

***Figure 2.4:*** *log generated*

**3. Website**

**Credential Harvesting Attack (Phishing Attack)**

**Description:** Phishing attacks are the practice of sending fraudulent communications that appear to come from a reputable source. It is usually done through email. The zero-bank’s website is cloned, the victim will access cloned page from that we can extract credentials.

**Vulnerability:** Social Engineering.

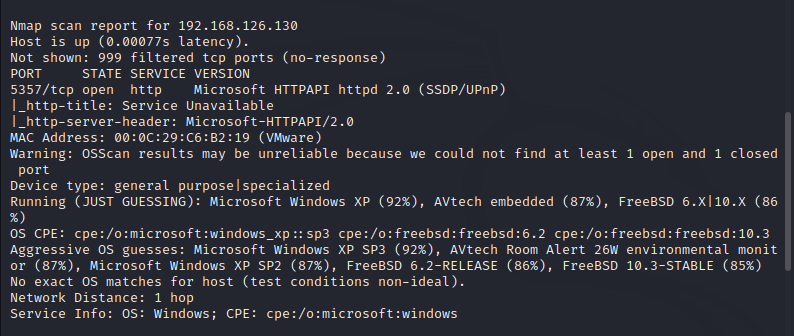
|  |  |
| --- | --- |
| **HIGH RISK (10/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **User Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

**Security Implications**

Check the website is genuine, check the website visiting is https the hackers use http, update browser frequently, don’t install unwanted certificates. Use correct webpage don’t do typo.

**Analysis**

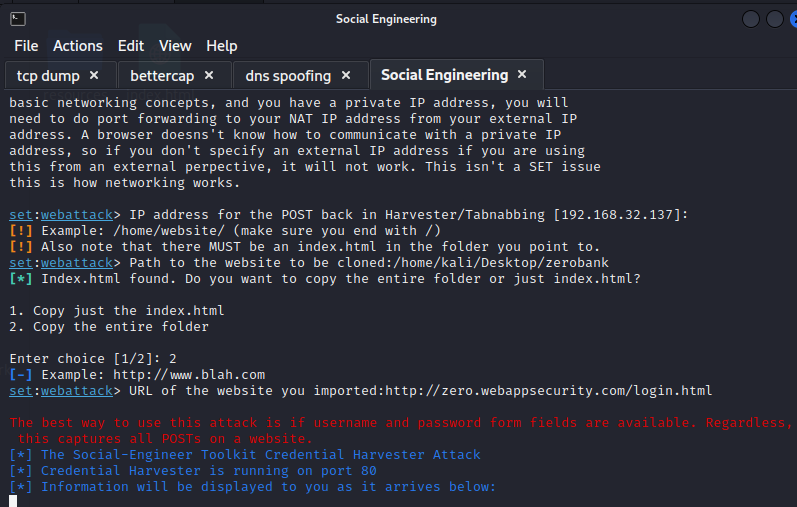
1. Nmap

****

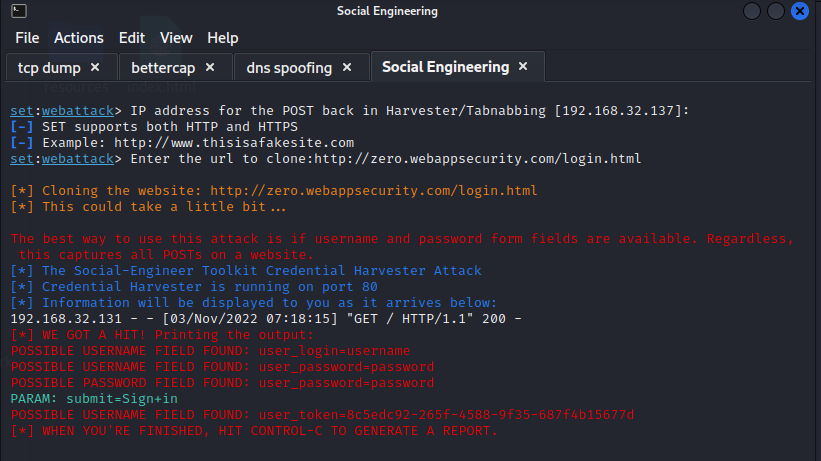
***Figure 3.1:*** *Nmap scan*

1. Setoolkit

Using setoolkit the site is cloned and the website is ready get credentials, additional step DNS spoofing can also be done. The credentials are also obtained.



***Figure 3.2:*** *Setoolkit running*



***Figure 3.3:*** *Obtaining credentials*

**4. Remote Access Service**

**SSH Protocol Authentication Bypass (Remote Exploit Check)**

**Description:** The remote ssh server is vulnerable to an authentication bypass. An attacker can bypass authentication by presenting SSH2\_MSG\_USERAUTH\_SUCCESS message in place of the SSH2\_MSG\_USERAUTH\_REQUEST method that normally would initiate authentication.

**Vulnerability:** SSH AUTH bypass

|  |  |
| --- | --- |
| **HIGH RISK (10/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **User Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

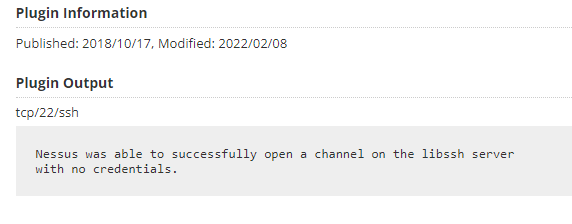
**Security Implications**

Upgrade to libssh 0.7.6 / 0.8.4 or later, if applicable. Otherwise, contact your product vendor. Update free sshd software or use any other ssh software.

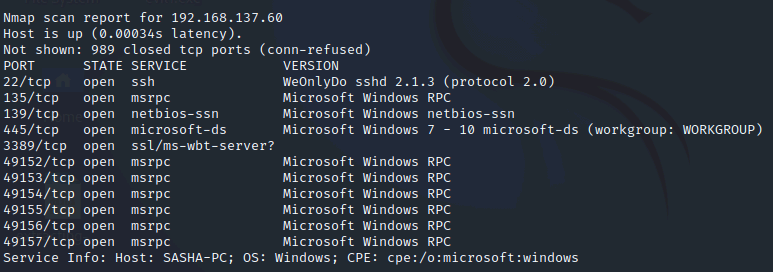
**Analysis**

1. Nessus and Nmap scan

Nessus was able to open channel without any credentials

****

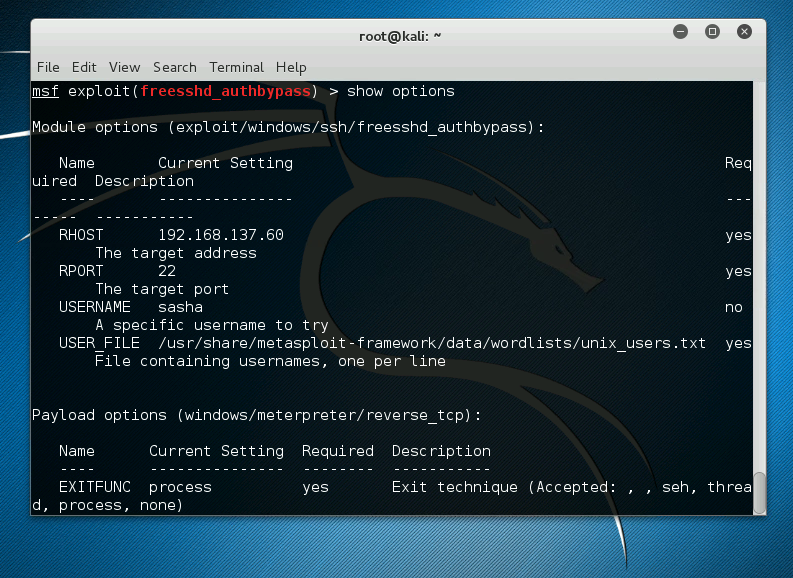
***Figure 4.1:*** *Nessus scan*

****

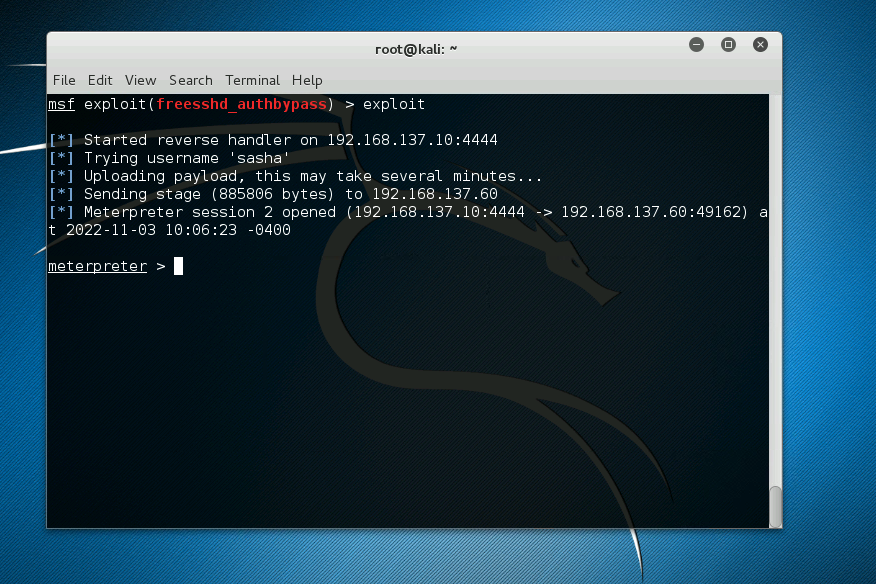
***Figure 4.2:*** *Nmap Scan*

1. Pen testing

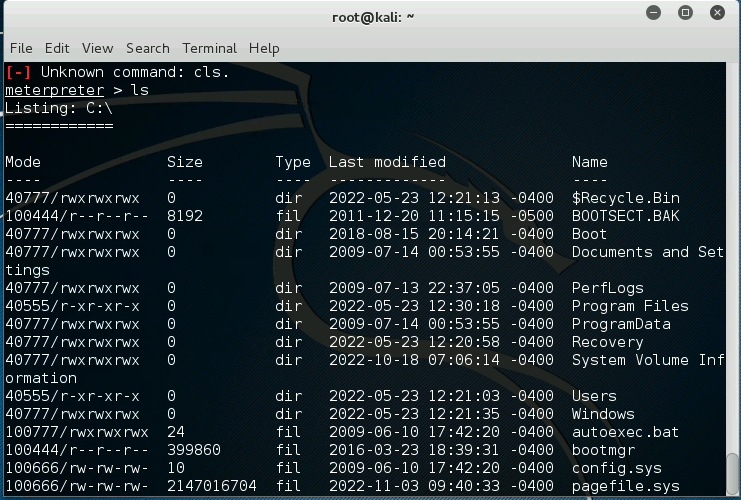
Using the gathered information, we can use AUTH bypass exploit can be used to access the machine. In metasploitable frame work we can use authbybass, now we need to set RHOST, then we can do exploit and we can attain meterpreter shell.



***Figure 4.3:*** *authbypass configuration*



***Figure 4.4:*** *Meterpreter shell*



***Figure 4.5:*** *Listing directories*

1. **Vulnerability Finding Mutilldae Webapplication**
   * 1. Extract Data

**Description:** The user information is exposed in this attack.

**Vulnerability:** SQL Injection

|  |  |
| --- | --- |
| **HIGH RISK (8/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

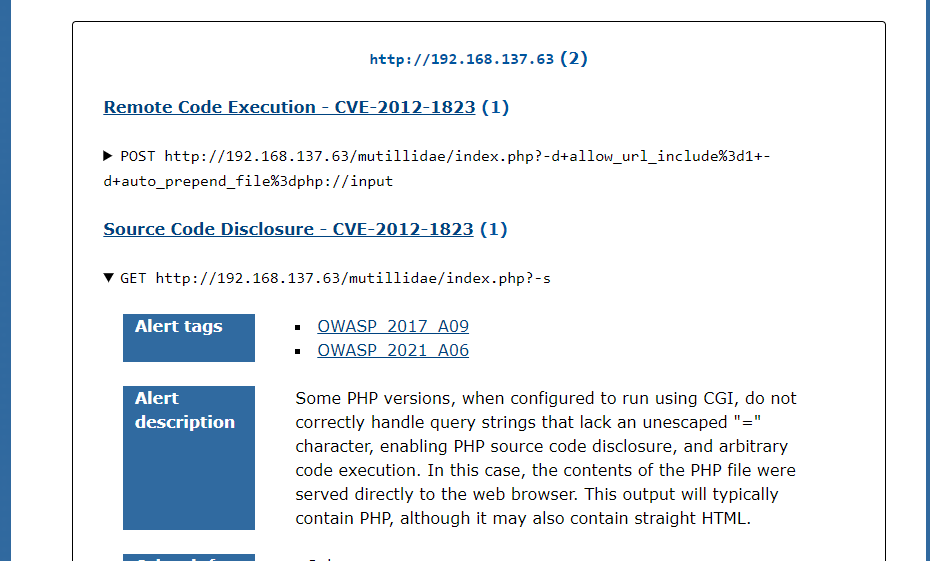
**Security Implications**

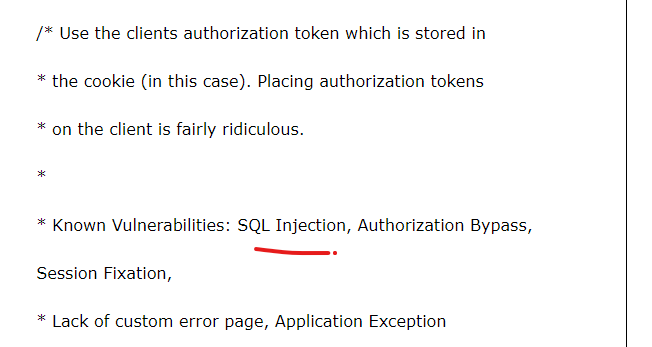
The information of the user is easily accessible by few or single SQL codes. For this user has to enter SQL code in username and password to extract login id and password. Then it can be used to login, even admin username password is exposed and the site settings can be changed. Upgrade to the latest stable version of PHP, or use the Apache web server and the mod\_rewrite module to filter out malicious requests using the "RewriteCond" and "RewriteRule" directives.

**Analysis**

1. Zaproxy Analysis

In the zaproxy report we can find that this application is vulnerable to SQL injection.

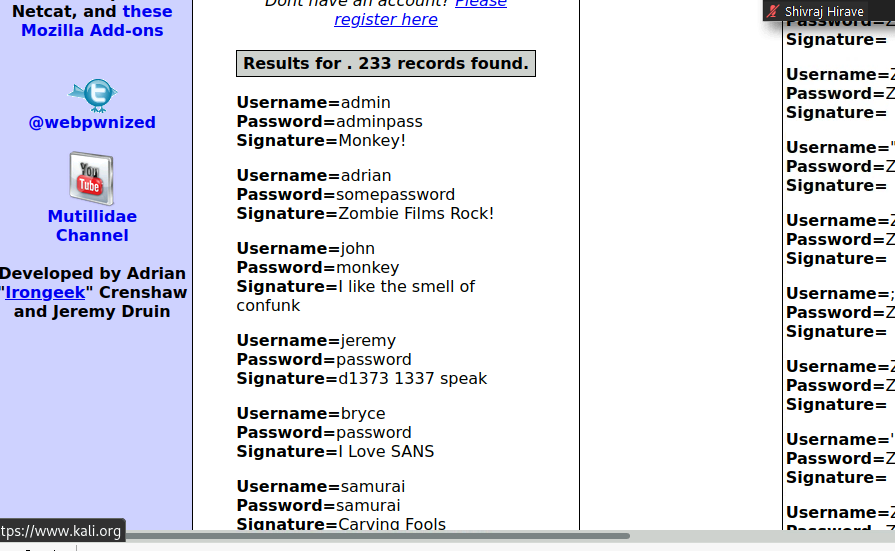




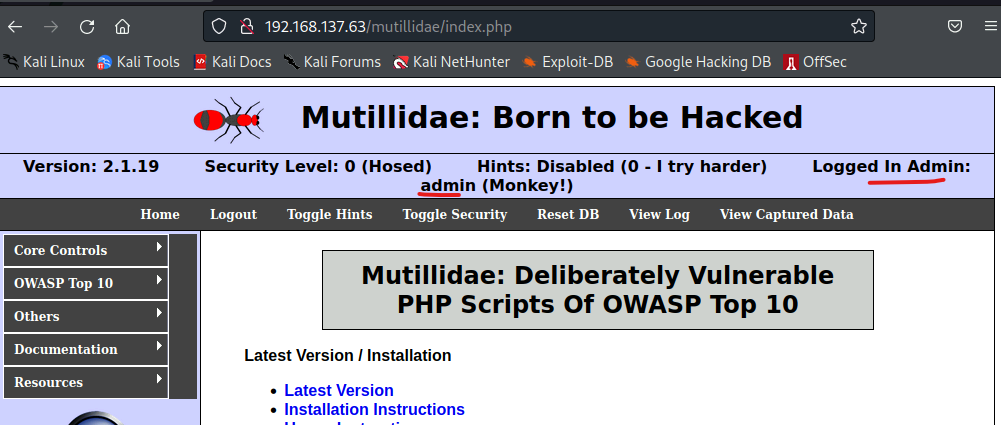
***Figure 1.1****: Zaproxy report*

1. Pen testing

From the report of zaproxy we can use SQL injection. There is so many commands but few will work it differs from one website to another. The query of boolean expression used here is **‘or 1=1 –** from this we can get all user’s login id and password.



***Figure 1.2****: SQL Injection result.*



***Figure 1.3****: Logged in as admin*

* + 1. Cross-Site Scripting

**Description:** The cross-site scripting the attacker can easily pop-up any notification.

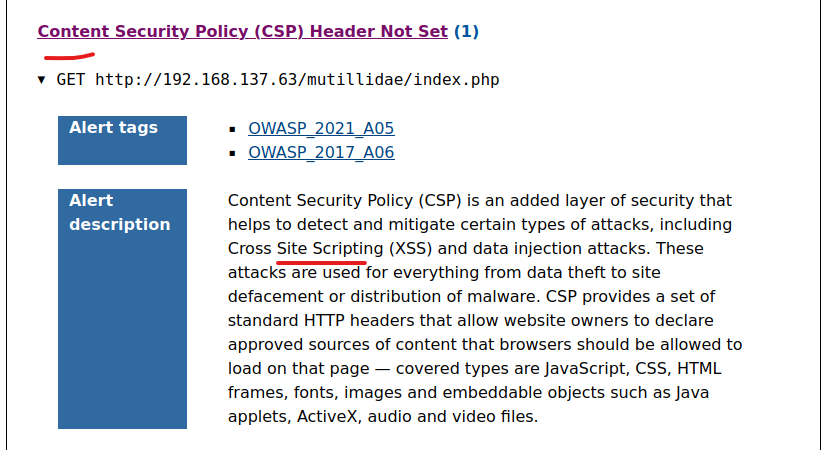
**Vulnerability:** Cross-site scripting XSS

|  |  |
| --- | --- |
| **Medium RISK (6/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Mild** |
| **Remediation Difficulty** | **Easy** |

**Security Implications**: In cross-site scripting the attacker can easily pop-up any notification or easily get document.cookie information. Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support: "Content-Security-Policy" for Chrome 25+, Firefox 23+ and Safari 7+, "X-Content-Security-Policy" for Firefox 4.0+ and Internet Explorer 10+, and "X-WebKit-CSP" for Chrome 14+ and Safari 6+.

**Analysis**

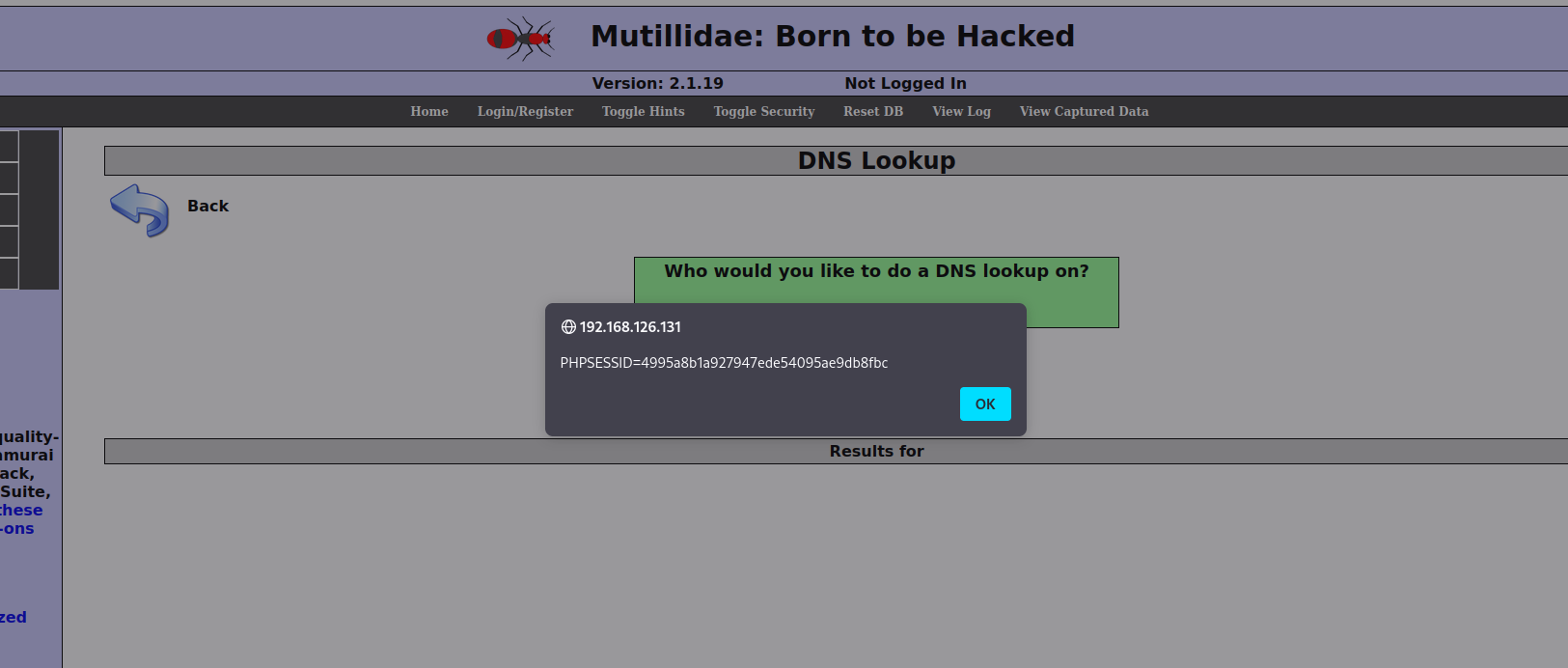
1. Zaproxy analysis



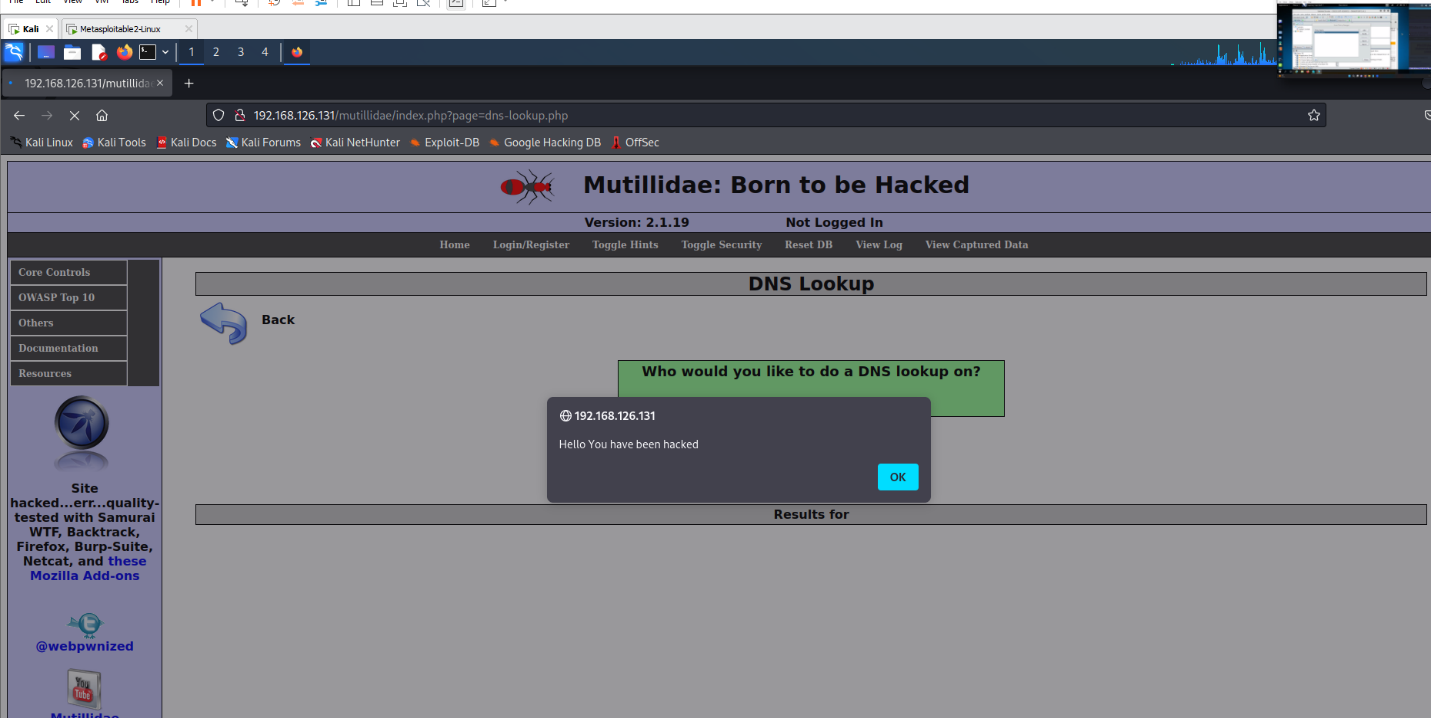
***Figure 2.1****: Xss indication zaproxy*

1. Xss implication.

The xss is java script which can be used to manipulate website. We can prompt messages, view browser details etc. Here **<script>alert(document.cookie)</script>** is used to display cookie information, **<script>alert(“Hello, you have hacked”)</script>** is to prompt user.



***Figure 2.2****: Xss document.cookie*



***Figure 2.3****: Xss document.cookie*

* + 1. Cross-Site Request Forgery

**Description:** The cross-site request forgery the attacker modifies user input like registrations, submissions etc.

**Vulnerability:** Cross-site Request Forgery

|  |  |
| --- | --- |
| **HIGH RISK (9/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

**Security Implications**: Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support: "Content-Security-Policy" for Chrome 25+, Firefox 23+ and Safari 7+, "X-Content-Security-Policy" for Firefox 4.0+ and Internet Explorer 10+, and "X-WebKit-CSP" for Chrome 14+ and Safari 6+.

**Analysis**

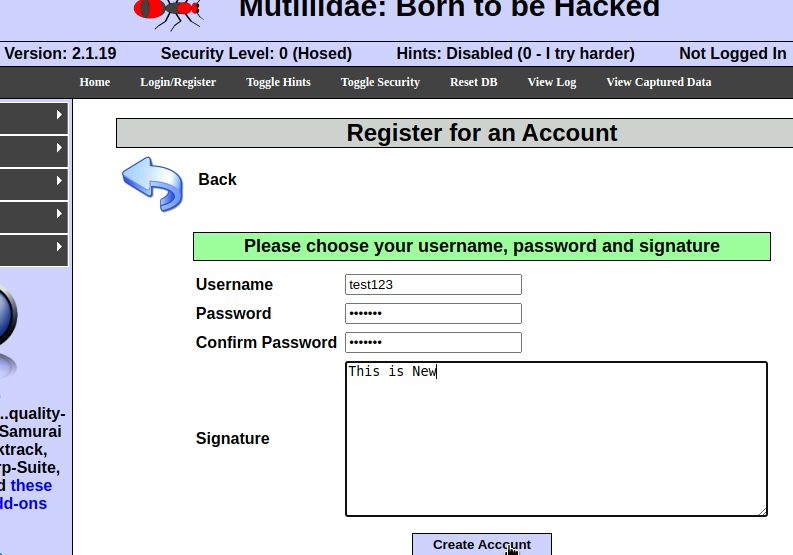
1. Zaproxy analysis



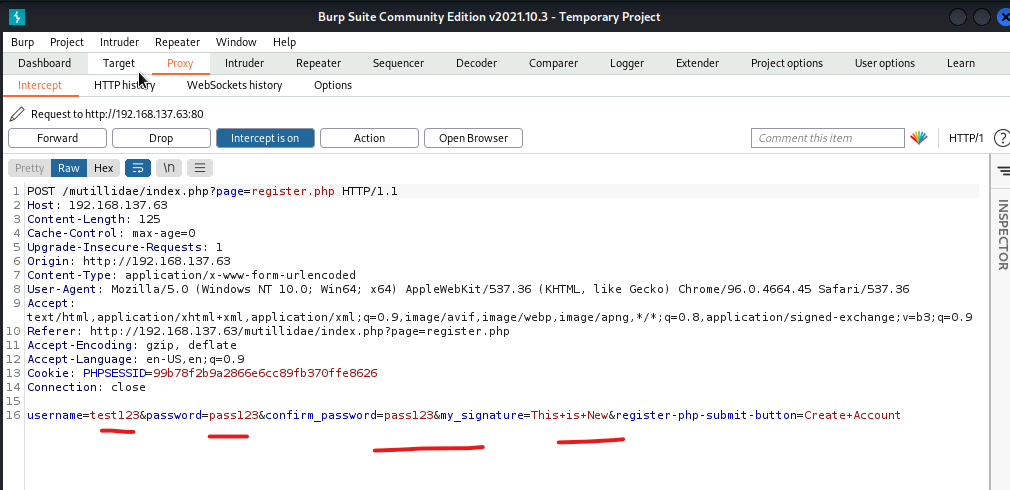
***Figure 3.1*** *zaproxy report*

1. Pentesting

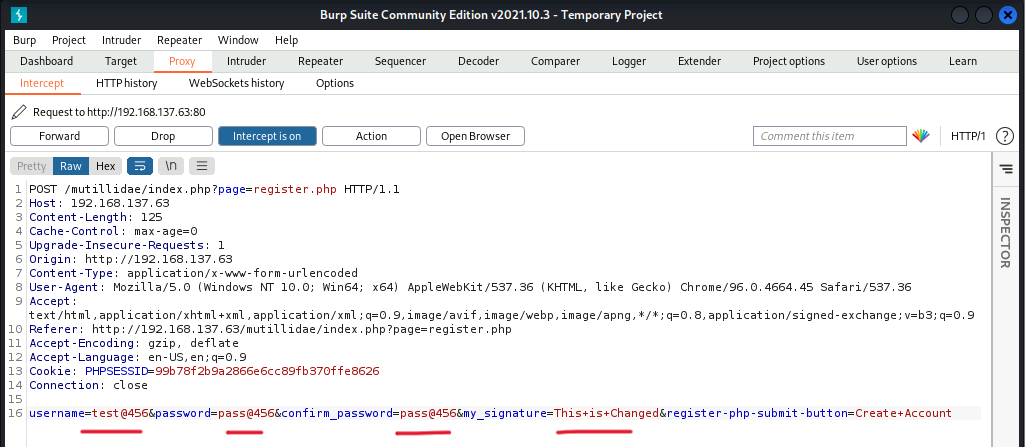
Here we are going to register account in mutilldae and the information will be passed through burpsuite. In burpsuite we are going to alter the values. By this we can achieve request forgery.



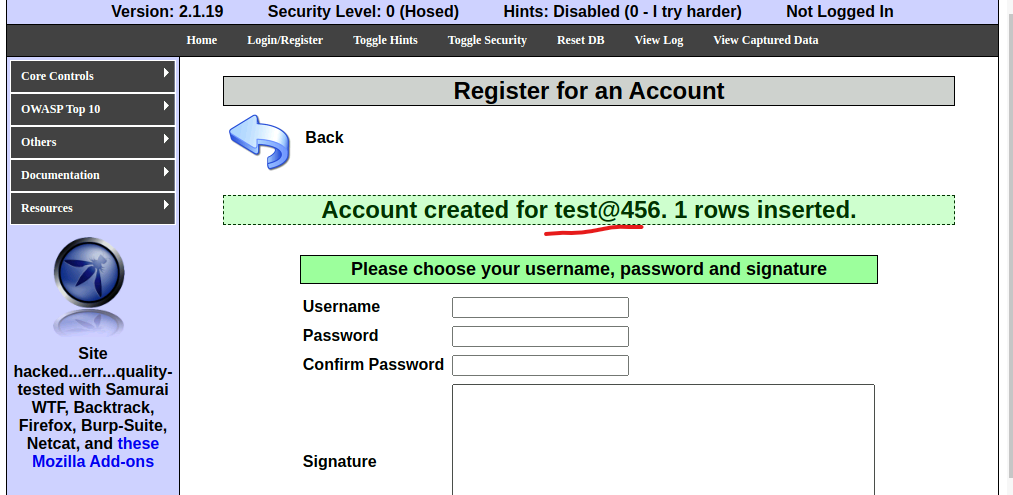
***Figure 3.2****: Registration*

**

***Figure 3.3****: Details in burpsuite*

**

***Figure 3.4****: Details Changed*

**

***Figure 3.5****: Changed details updated in database*

* + 1. Insecure Direct object

**Description:** In the insecure direct object the text files can be viewed. By this system information any readable text file can be accessed.

**Vulnerability:** Insecure direct object

|  |  |
| --- | --- |
| **HIGH RISK (8/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

**Security Implications**:

Update the front-end server due to a loop hole in web application this kind of attack are possible. Insecure direct object reference (IDOR) is a type of access control vulnerability in digital security. This can occur when a web application or application programming interface uses an identifier for direct access to an object in an internal database but does not check for access control or authentication.

**Analysis**

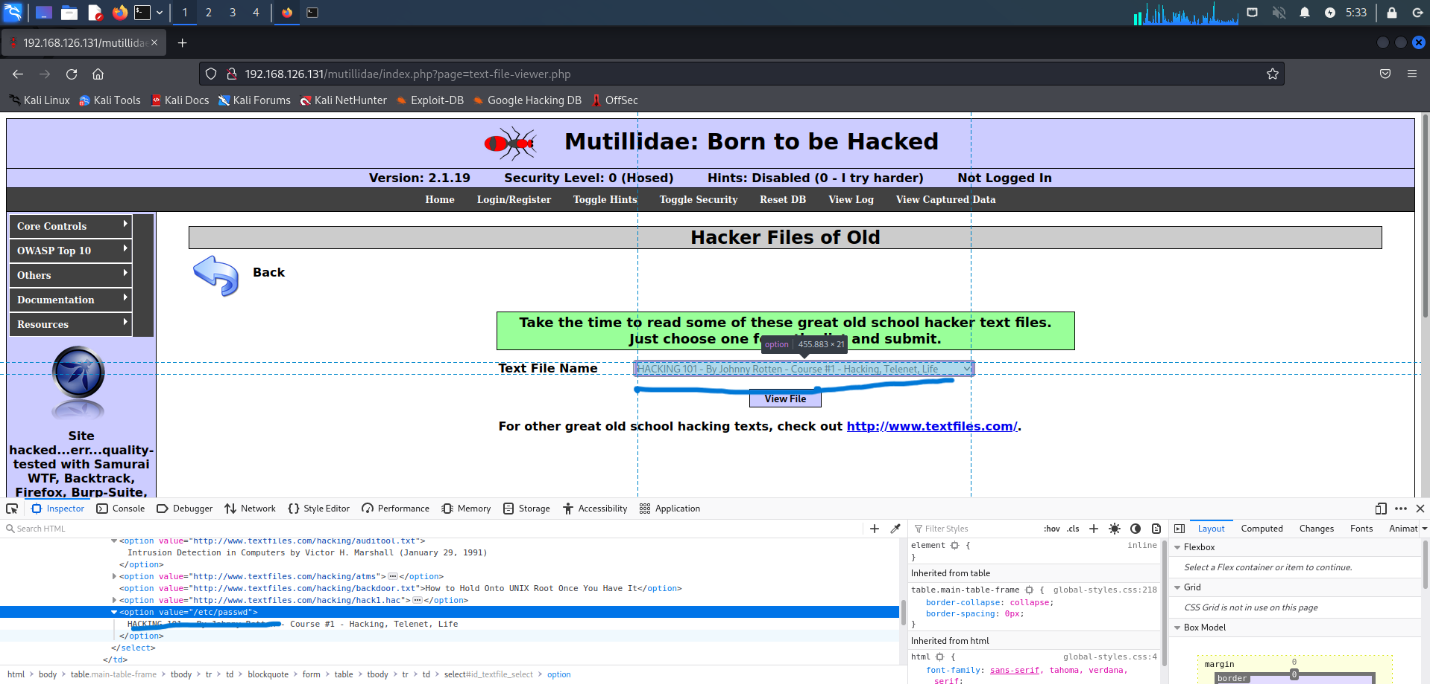
1. Zaproxy analysis

****

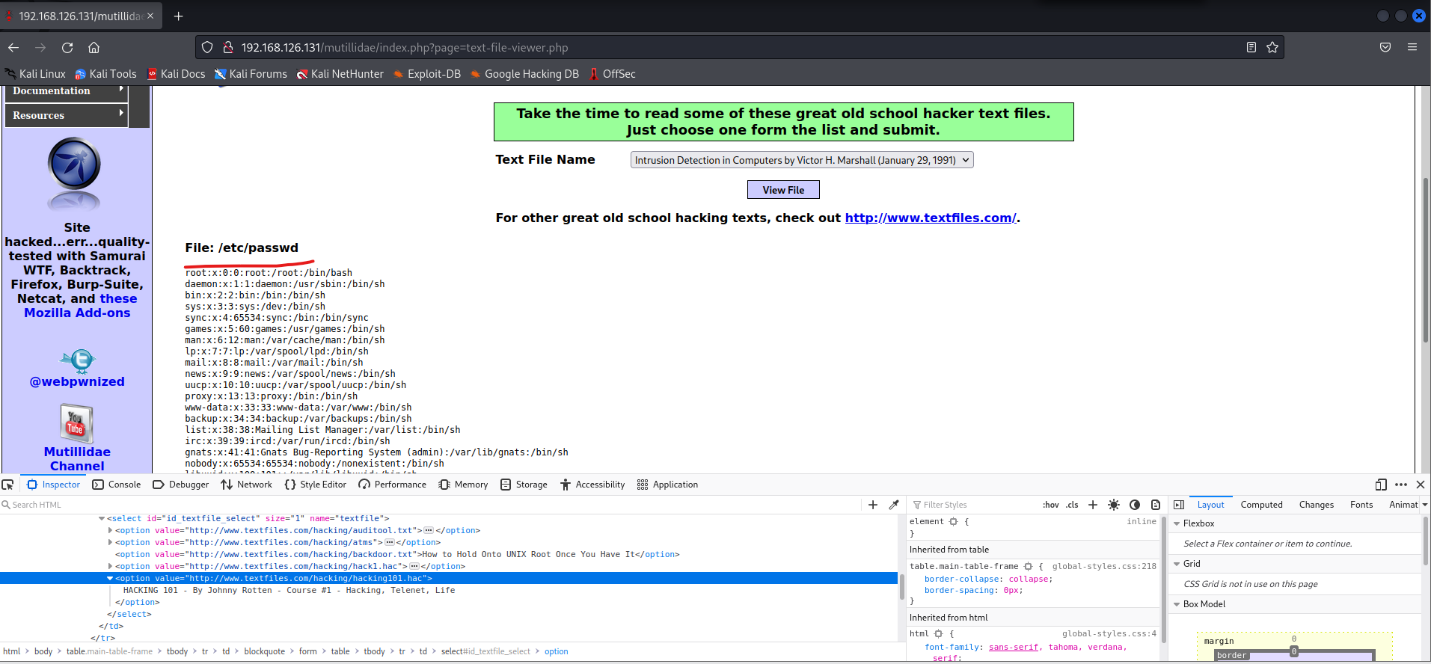
***Figure 4.1****: Zaproxy report*

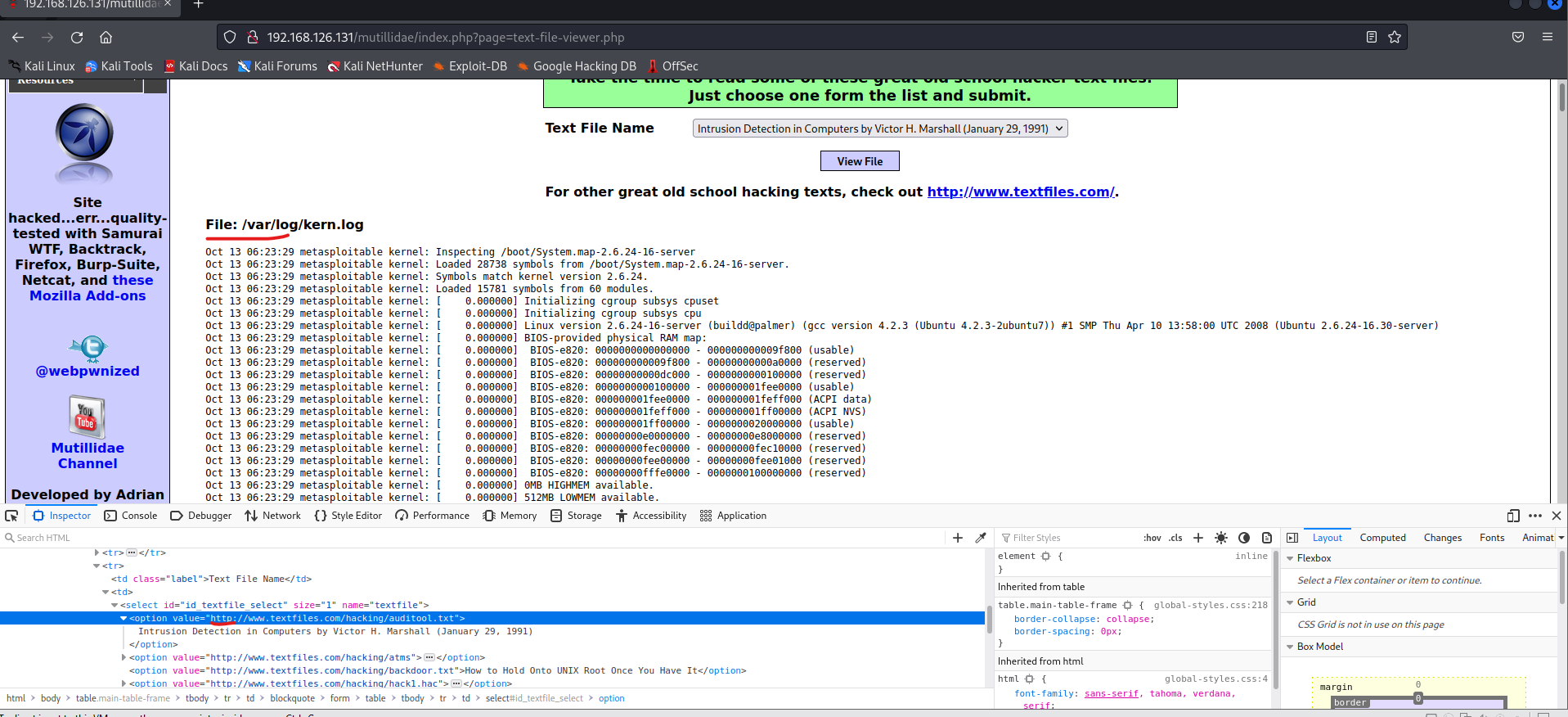
1. Exploiting

The exploitation is done by displaying a text file eg. cat /etc/os-release to see linux version like this we can get the details of logs, diskspace, users, etc. Here there few commands use to display results /etc/passwd, /var/log/kern.log, /etc/fstab, /etc/crontab, /etc/securetty.



***Figure 4.2****: Inspect element of exploiting area*



***Figure 4.3****: Passwd result*

***Figure 4.3****: kern.log result*

1. **Vulnerability Finding Mutilldae Database**

Extracting Data SQL injection manual and automated

**Description:** The user information is exposed in this attack. Data extraction is the process of collecting or retrieving disparate types of data from a variety of sources, many of which may be poorly organized or completely unstructured.

**Vulnerability:** SQL Injection (SQL Map)

|  |  |
| --- | --- |
| **HIGH RISK (8/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

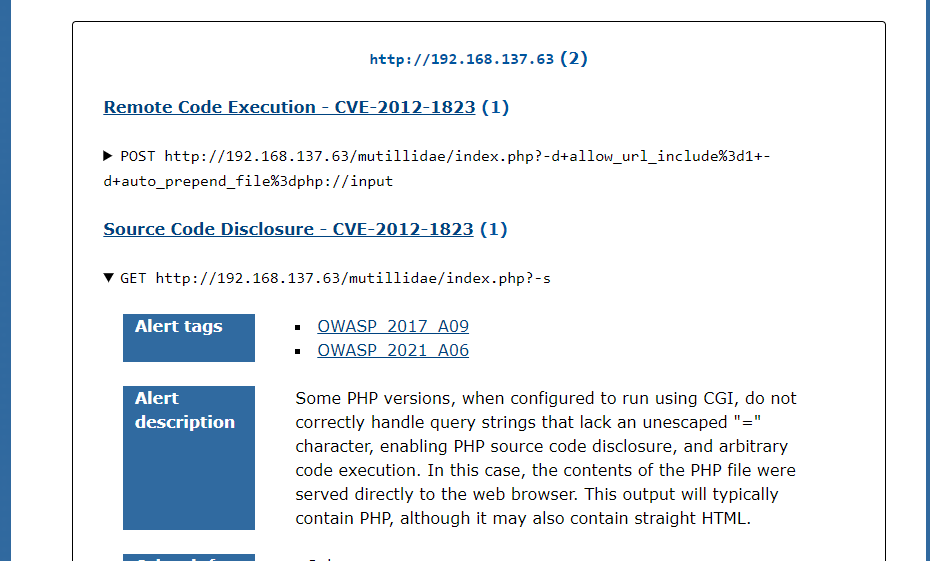
**Security Implications**

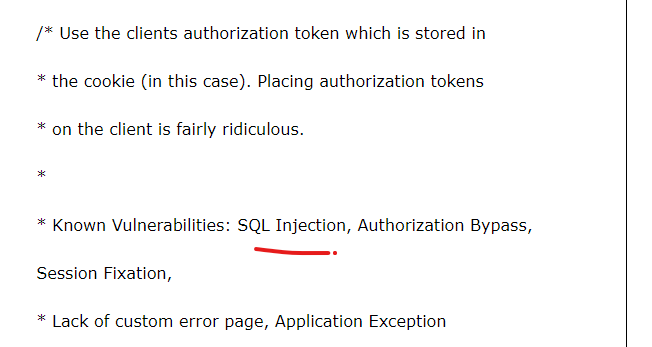
The information of the user is easily accessible by few or single SQL codes. For this user has to enter SQL code in username and password to extract login id and password. The SQLmap is used to get information from databases.Then it can be used to login, even admin username password is exposed and the site settings can be changed. Upgrade to the latest stable version of PHP, or use the Apache web server and the mod\_rewrite module to filter out malicious requests using the "RewriteCond" and "RewriteRule" directives.

**Analysis**

* + - 1. Zaproxy Analysis

In the zaproxy report we can find that this website is vulnerable to SQL injection.

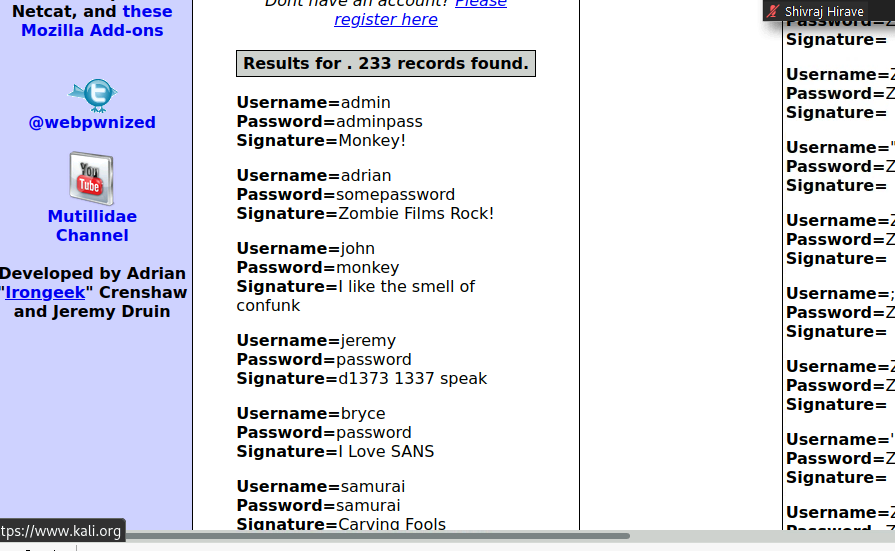




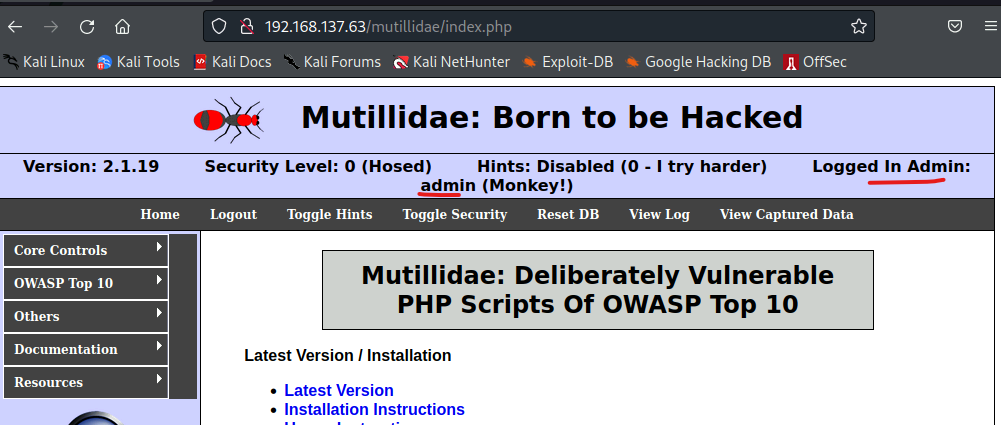
***Figure 1.1****: Zaproxy report*

1. Pen testing
2. Manual SQL Injection

From the report of zaproxy we know that this website is vulnerable to SQL injection. There is so many expressions but only few will work it differs from one website to another. The query of Boolean expression used here is **‘or 1=1 –** from this we can get all user’s login id and password.



***Figure 2.1.1****: SQL Injection result.*



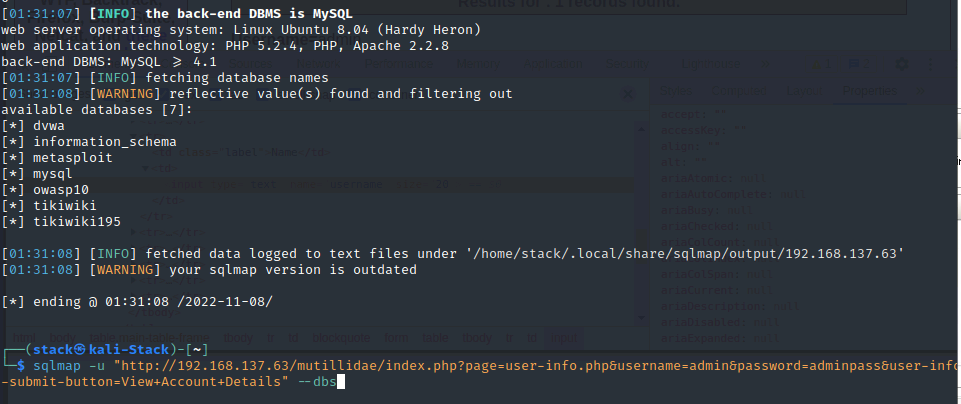
***Figure 2.1.2.****: Logged in as admin*

1. Automated SQL injection

For automated SQL injection the tool used here is **SQLmap** by using some attributes of sqlmap we can retrieve data of database.

1. First step is to get to know the databases available in the mutilldae website. Before getting the databases, I logged in as adminuser from user-info and used the link by login. The command used is

**sqlmap -u “**[**http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details**](http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details)**” --dbs**

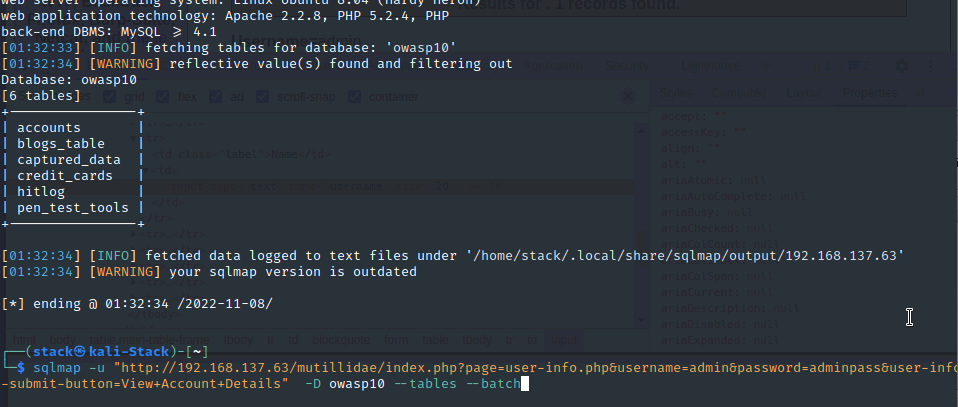


***Figure 2.2.1****: Retrieving databases*

1. Second step is to get the tables from the owasp10 database. The owasp10 is the database of mutilldae. The table details can be fetched using the following command

**sqlmap -u “**[**http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details**](http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details)

**-D owasp10 --tables --batch**



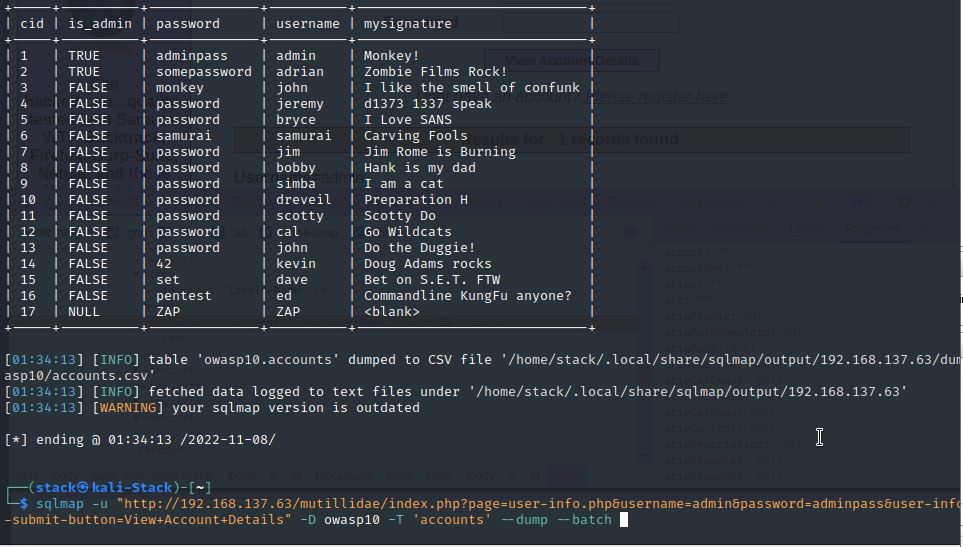
***Figure 2.2.2****: Retrieving tables from databases*

1. Third step is to get information from the table. From before steps we can Accounts, Credit cards, Machine used the website, Log file of database.

The command used to retrieve data from **accounts table** is

**sqlmap -u “**[**http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details**](http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details)

**-D owasp10 -T ‘accounts’ --dump --batch**

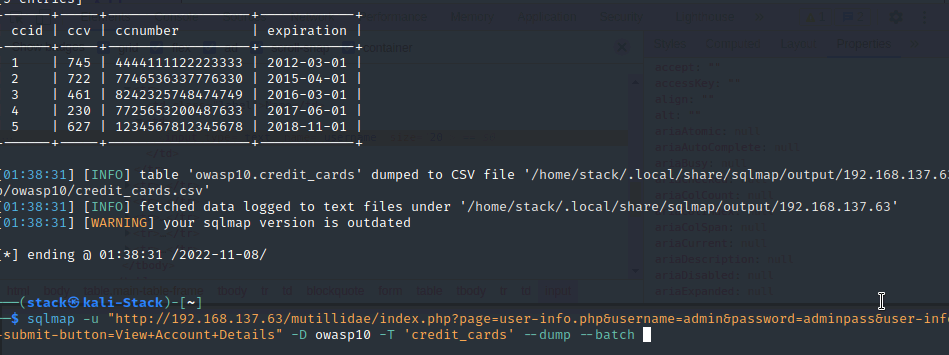


***Figure 2.2.4****: Retrieving information from accounts table*

The command used to retrieve data from **credit\_cards table** is

**sqlmap -u “**[**http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details**](http://192.168.137.63/mutilldae/index.php?page=userinfo.php&username=admin&password=adminpass&user-info-submit-button=View+Account+Details)

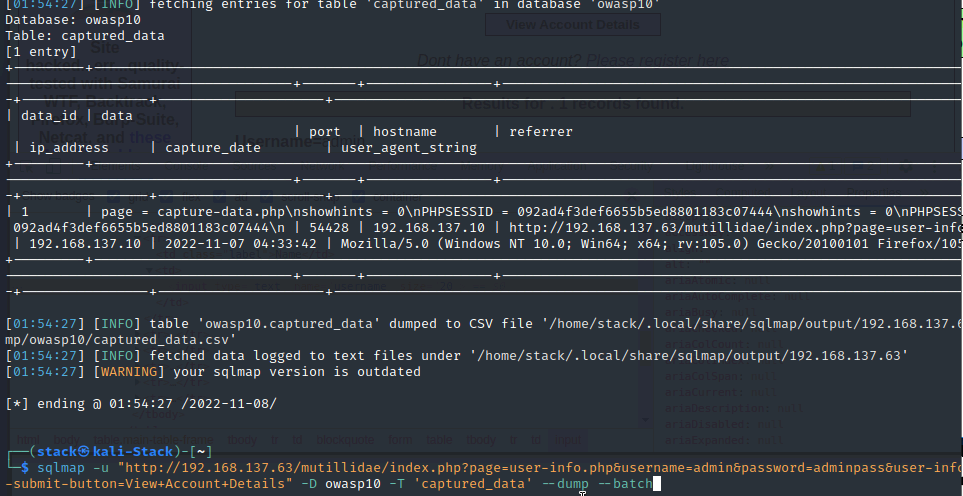
**-D owasp10 -T ‘credit\_cards’ --dump --batch**

****

***Figure 2.2.5****: Retrieving information from credit\_cards table*

The command used to retrieve data from **captured\_data table** is

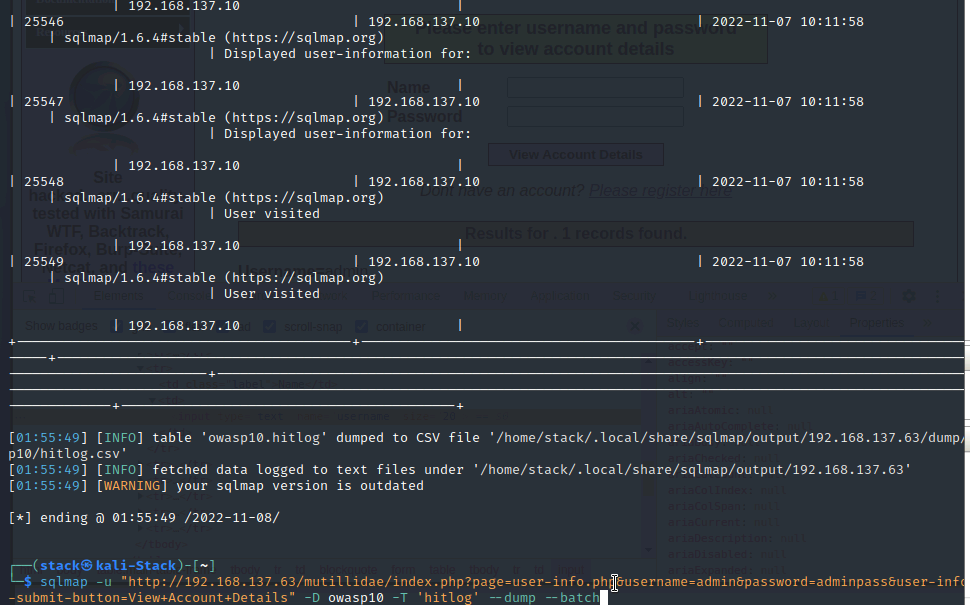
In -T the value changed to **‘captured\_data’**

****

***Figure 2.2.3****: Retrieving information from captured\_data table*

The command used to retrieve data from **hitlog table** is

In -T the value changed to **‘hitlog’**



***Figure 2.2.3****: Retrieving information from hitlog table*

SUGGESTED REMEDIATION

Recommendations

# Secure port 88,445,389 in firewall from windows server 2019 and use Avast antivirus it will prompt if any attack is happening.

#### Multi-Factor Authentication, verify Email Sender’s Identity, check for SSL Certificate, Check and Update your Security Patches, Pay Attention to Your Digital Footprint prevention of SE attacks.

#### Secure port 22 in firewall from Windows 7, update the freesshd software patches have been applied for latest one, update to windows 10 or 11 to get the system protected.

* To prevent from web-site attack, train and maintain awareness, don’t trust any user input, use whitelists not blacklists, adopt the latest technologies.
* To prevent from XSS, use SameSite is a cookie attribute (similar to HTTPOnly, Secure etc.) which aims to mitigate CSRF attacks. Verifying Origin with Standard Headers, Identifying the Target Origin these are the security measurement.
* To prevent from insecure direct object, Avoid Exposing Direct Object References, use an Indirect Reference Map, check user access at the Data-Object level, protecting against Directory Traversal.

# APPENDIX A - TOOLS USED

|  |  |
| --- | --- |
| **TOOL** | **DESCRIPTION** |
| **BurpSuite Community Edition** | Used for testing of web applications. |
| **Metasploit** | Used for exploitation of vulnerable services and vulnerability scanning. |
| **Nmap** | Used for scanning ports on hosts. |
| **Zaproxy** | Used for Vulnerability Assessment on web applications |
| **Nessus** | Used for Vulnerability Assessment on systems |
| **SQLmap** | Used for exploiting database |

***Table A.1:*** *Tools used during assessment*

# APPENDIX B - ENGAGEMENT INFORMATION

## Client Information

|  |  |
| --- | --- |
| **Client** | ZEROBANK |
| **Primary Contact** | Thirumaal |
| **Approvers** | The following people are authorized to change the scope of engagement and modify the terms of the engagement   * Thirumaal |

## Version Information

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | 10-11-2022 | Initial report to client |

## Contact Information

|  |  |
| --- | --- |
| **Name** | Thirumaal Consulting |
| **Address** | 1001 Fake Street, Gotham, NY 11201 |
| **Phone** | 555-185-1782 |
| **Email** | thirumaalg@gmail.com |