

**(Internship Summer 2024)**

|  |  |
| --- | --- |
| **Name**  **Domain**  **Email**  **Date** | **Kamran Bader**  **Cybersecurity**  [**kamranbader@gmail.com**](file:///C:\Users\Administrator\Downloads\kamranbader@gmail.com)  **26/7/2024** |

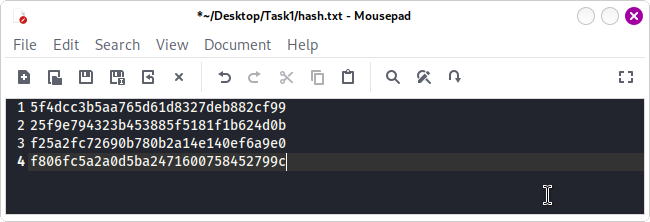
# Task 1: Password Cracking with Hashcat

* **Hashcat Installation:**
  + Download Hashcat from the [Hashcat Website.](https://hashcat.net/hashcat/)
  + Follow the installation instructions provided on the website.
  + Ensure you have a compatible GPU and CUDA or OpenCL drivers installed.
* **Obtain the Target Hash File:**

* + The target hash file (hash.txt) contains the hashed passwords to be cracked. This file was generated using [md5hashgenerator.com.](https://www.md5hashgenerator.com/) ⚫ Hashes generated for the following passwords:

123456789 password iloveyou rockyou monkey

* + Save the hashes in a file named hash.txt.



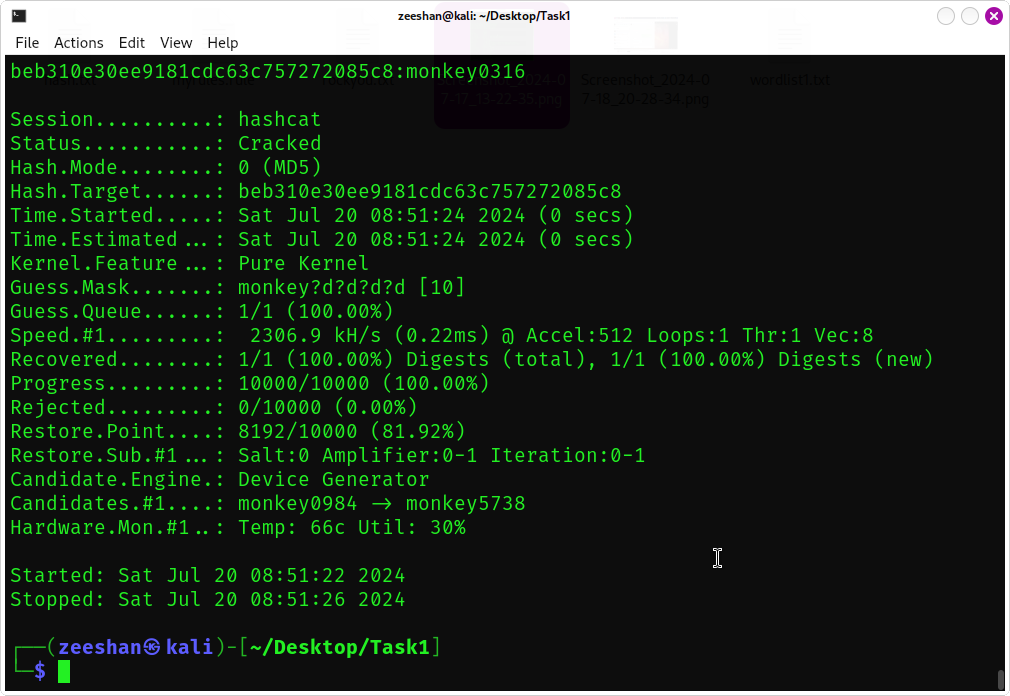
* **Download a Wordlist:**
  + Downloaded the commonly used wordlist **rockyou.txt** from the internet.

* **Identify the Hash Type:**
  + The hashes were generated using the MD5 algorithm.
  + Hashcat supports various hash types represented by different modes. For MD5, the mode is **-m 0**.

* **Run the Basic Command:**

## hashcat -m 0 -a 0 hash.txt rockyou.txt

* **Command Breakdown:**
  + **-m 0**: Specifies the MD5 hash type.
  + **-a 0**: Specifies a straight attack mode.
  + **hash.tx**t: The file containing the hashes. ⚫ **rockyou.txt**: The wordlist.



* **Apply Rules:**
  + To transform words in the wordlist (e.g., leetspeak, adding numbers), use the following command:

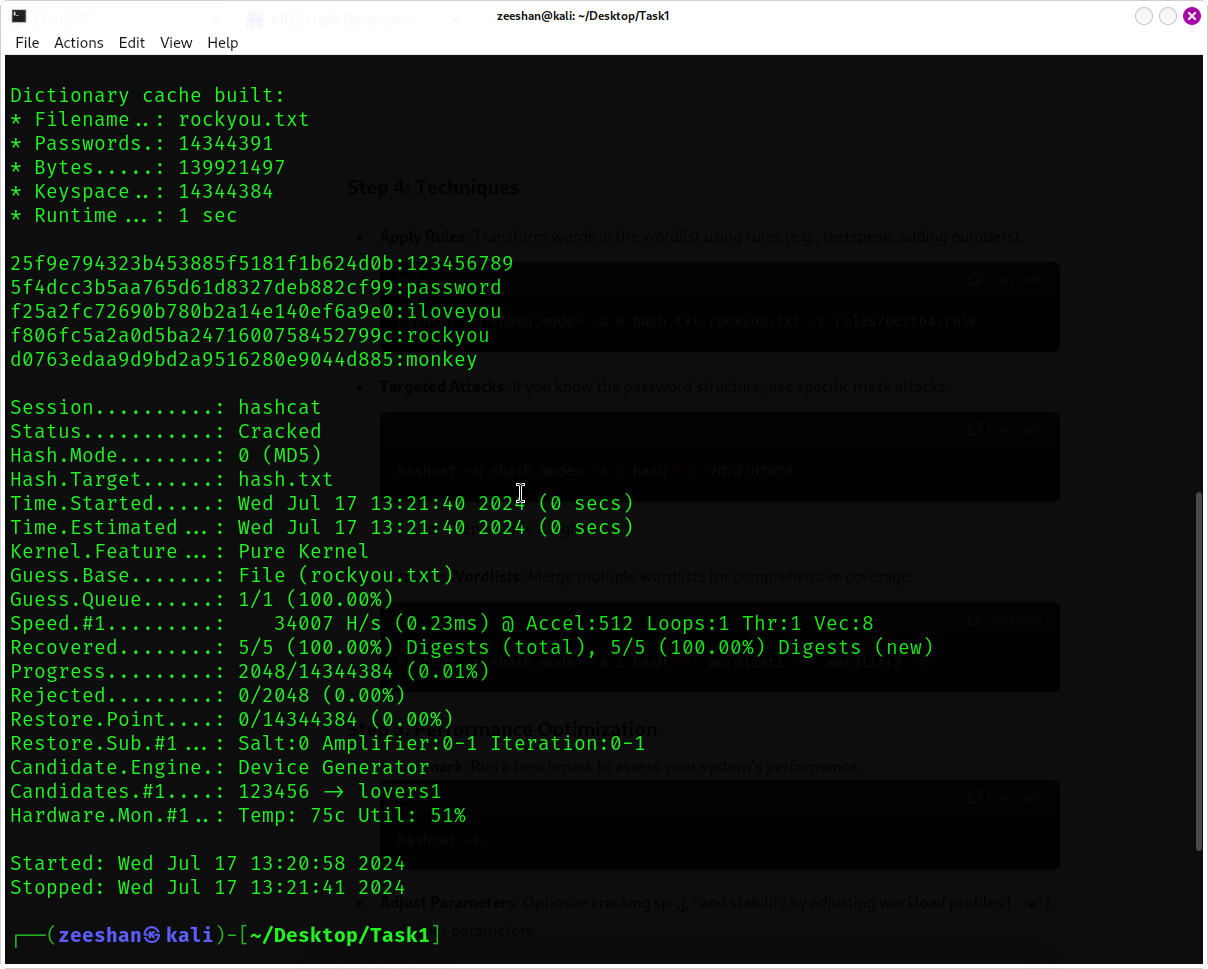
## hashcat -m 0 -a 0 hash.txt rockyou.txt -r rules/best64.rule

➢ **Brute Force Attack:**

⚫ For the password "**monkey0316**", which was not present in the rockyou.txt, a brute force attack was used:

## sudo hashcat -m 0 -a 3 hash.txt ?d?d?d?d

• ?d?d?d?d: Represents four digits (0-9).



# Task 2: Vulnerability Scanning with OpenVAS

* **installation process:**

* + Update Kali Linux with **sudo apt update**.

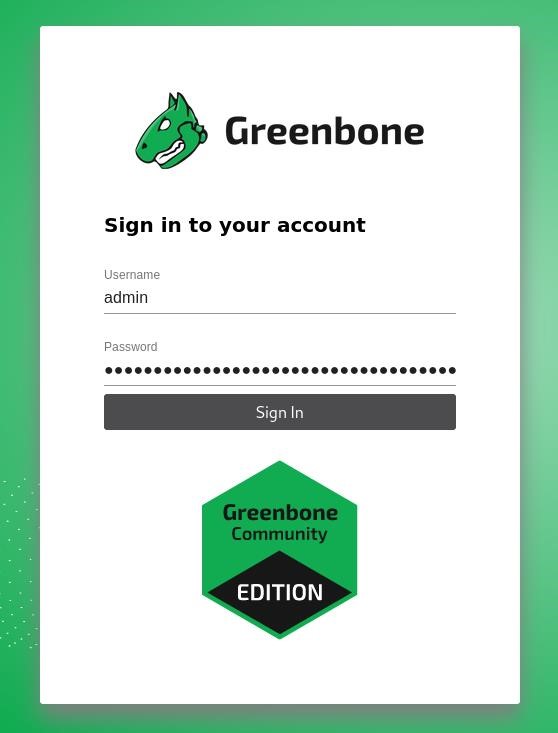
* + Install Greenbone Community Edition with **sudo apt install gvm**.

* + Configure Greenbone Community Edition with sudo gvm-setup and note the provided admin password from the output

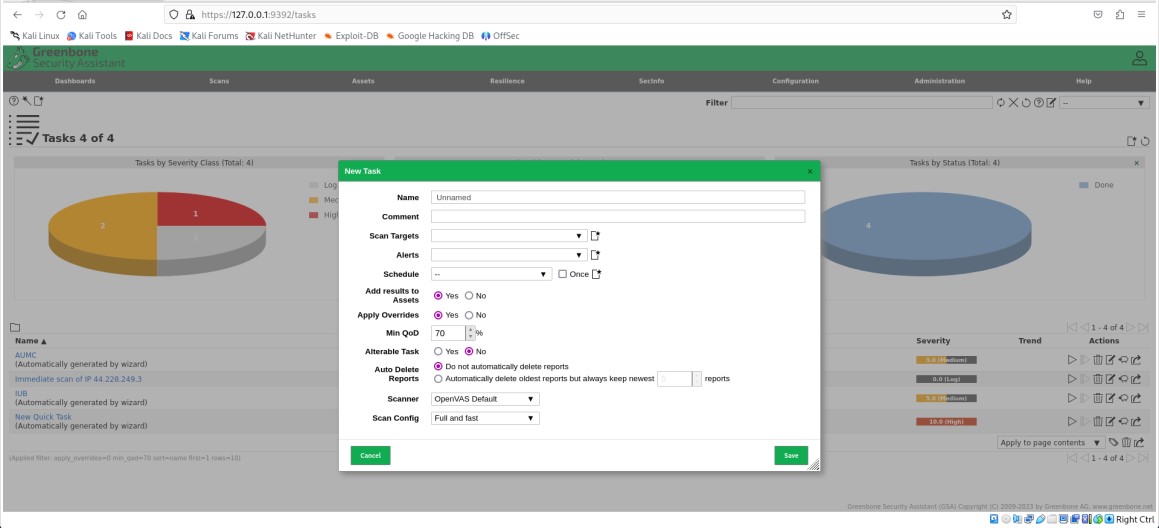
* + Check the install status with **gvm-check-setup**.

* + Visit [**https://127.0.0.1:9392**](https://127.0.0.1:9392/) and log in using the credentials output in step 3.

* + Verify the feed status before starting your first scan.



* **Scaning a Network:** 
  + Input the given data for scaning a network.

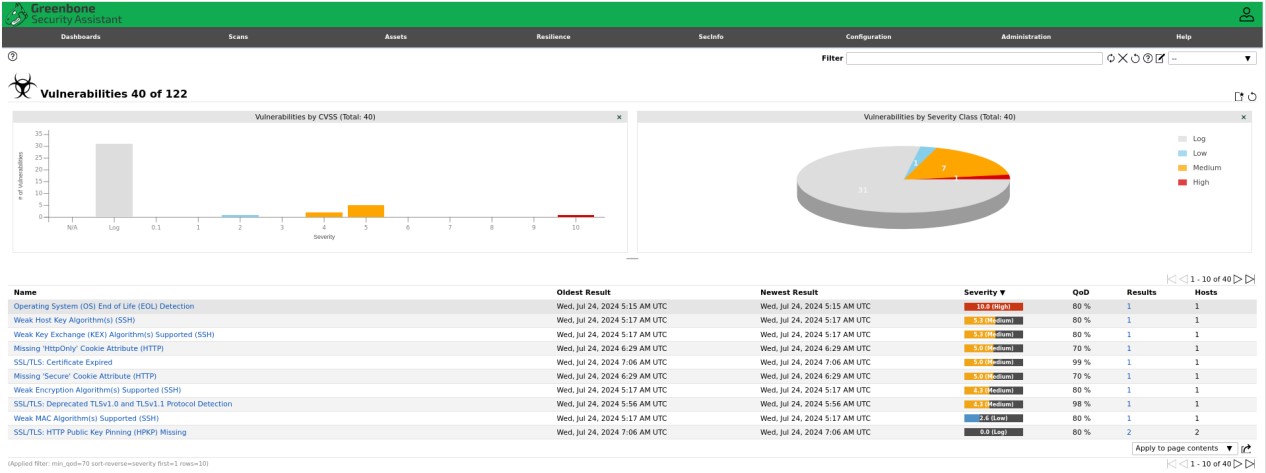


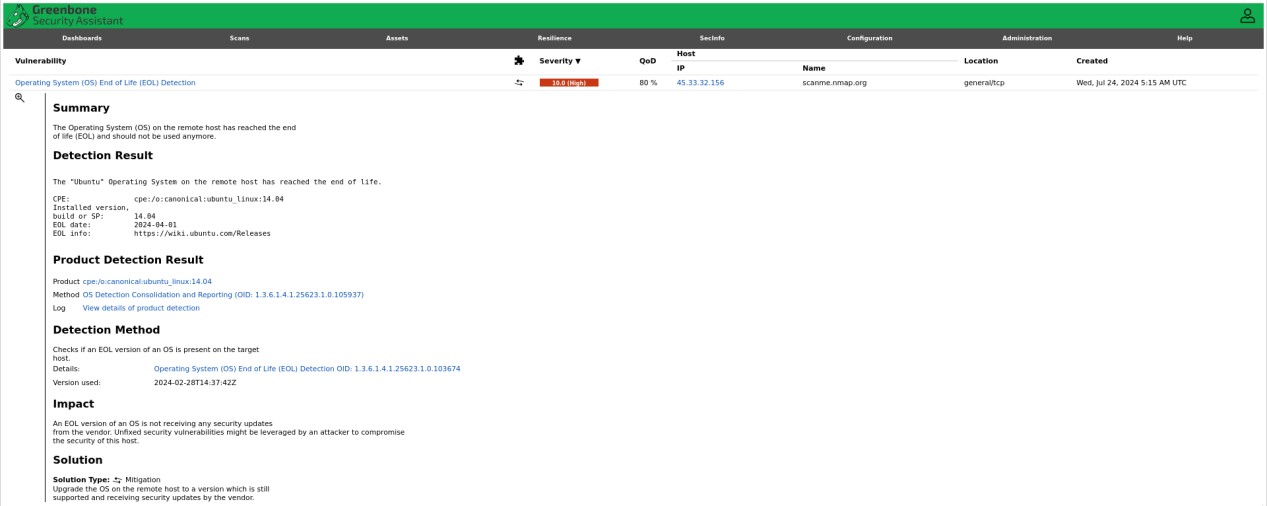
* + i have Scaned

## AUMC [185.151.30.193](https://127.0.0.1:9392/host/2db92ae2-1aa8-4086-a774-695ac4ea4173) IUB [121.52.159.138](https://127.0.0.1:9392/host/9b805015-71aa-47f1-9377-45c3d24e53af) Random Web [45.33.32.156](https://127.0.0.1:9392/host/53d86771-6449-4f89-bd0c-53d26d45982c)



* **Vulnerabilities:**





* **Recommended Solution:**

* + Operating System (OS) End of Life (EOL) Detection
  + Weak Host Key Algorithm(s) (SSH)
  + Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)
  + Missing 'HttpOnly' Cookie Attribute (HTTP)
  + Missing 'Secure' Cookie Attribute (HTTP)
  + Weak Encryption Algorithm(s) Supported (SSH)
  + SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection
  + SSL/TLS: HTTP Public Key Pinning (HPKP) Missing

1. **Operating System (OS) End of Life (EOL) Detection**

**Severity**: High

**Remediation Steps**:

* 1. **Upgrade OS**: Identify the latest stable version of the OS that is currently supported and plan an upgrade.
  2. **Backup Data**: Ensure that all critical data is backed up before starting the upgrade process.
  3. **Test Compatibility**: Verify that all applications and services running on the server are compatible with the new OS version.
  4. **Upgrade and Verify**: Perform the OS upgrade and validate that the system and services are operating correctly post-upgrade.
  5. **Apply Security Patches**: Regularly apply security patches and updates to maintain the system's security.

1. **Weak Host Key Algorithm(s) (SSH)**

**Severity**: High

**Remediation Steps**:

* 1. **Edit SSH Configuration**: Open the SSH configuration file (/etc/ssh/sshd\_config).
  2. **Modify HostKeyAlgorithms**: Update the HostKeyAlgorithms directive to exclude weak algorithms like ssh-dss and include only strong algorithms such as rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2nistp384, and ecdsa-sha2-nistp521.

## HostKeyAlgorithms rsa-sha2-512,rsa-sha2-256,ecdsa-sha2nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521

3. **Restart SSH Service**: Restart the SSH service to apply the new configuration

## sudo systemctl restart sshd

**3. Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)**

**Severity**: High

**Remediation Steps**:

1. **Edit SSH Configuration**: Open the SSH configuration file (/etc/ssh/sshd\_config).
2. **Modify KexAlgorithms**: Update the KexAlgorithms directive to exclude weak algorithms and include only strong ones such as curve25519sha256, diffie-hellman-group-exchange-sha256, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.

## ·KexAlgorithms curve25519-sha256,ecdh-sha2-nistp256,ecdhsha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-groupexchange-sha256

3. **Restart SSH Service**: Restart the SSH service to apply the new configuration:

## sudo systemctl restart sshd

1. **Missing 'HttpOnly' Cookie Attribute (HTTP)**

**Severity**: Medium

**Remediation Steps**:

* 1. **Identify Cookies**: Review the web application code to identify where cookies are set.
  2. **Add HttpOnly Attribute**: Modify the cookie-setting code to include the HttpOnly attribute.
  3. **Test Application**: Test the application to ensure that the cookies are correctly set with the HttpOnly attribute and that functionality is unaffected.

1. **Missing 'Secure' Cookie Attribute (HTTP)**

**Severity**: Medium

**Remediation Steps**:

* + 1. **Enable HTTPS**: Ensure that your web application uses HTTPS for secure communication.
    2. **Identify Cookies**: Review the web application code to identify where cookies are set.
    3. **Add Secure Attribute**: Modify the cookie-setting code to include the Secure attribute.

## · · Set-Cookie: sessionId=abc123; Secure

· · ·

4. **Test Application**: Test the application to ensure that the cookies are correctly set with the Secure attribute and that functionality is unaffected.

1. **Weak Encryption Algorithm(s) Supported (SSH)**

**Severity**: High

**Remediation Steps**:

* + 1. **Edit SSH Configuration**: Open the SSH configuration file (/etc/ssh/sshd\_config).
    2. **Modify Ciphers**: Update the Ciphers directive to exclude weak encryption algorithms and include only strong ones such as chacha20-poly1305@openssh.com, aes256-gcm@openssh.com, and aes128-gcm@openssh.com.
    3. Restart SSH Service

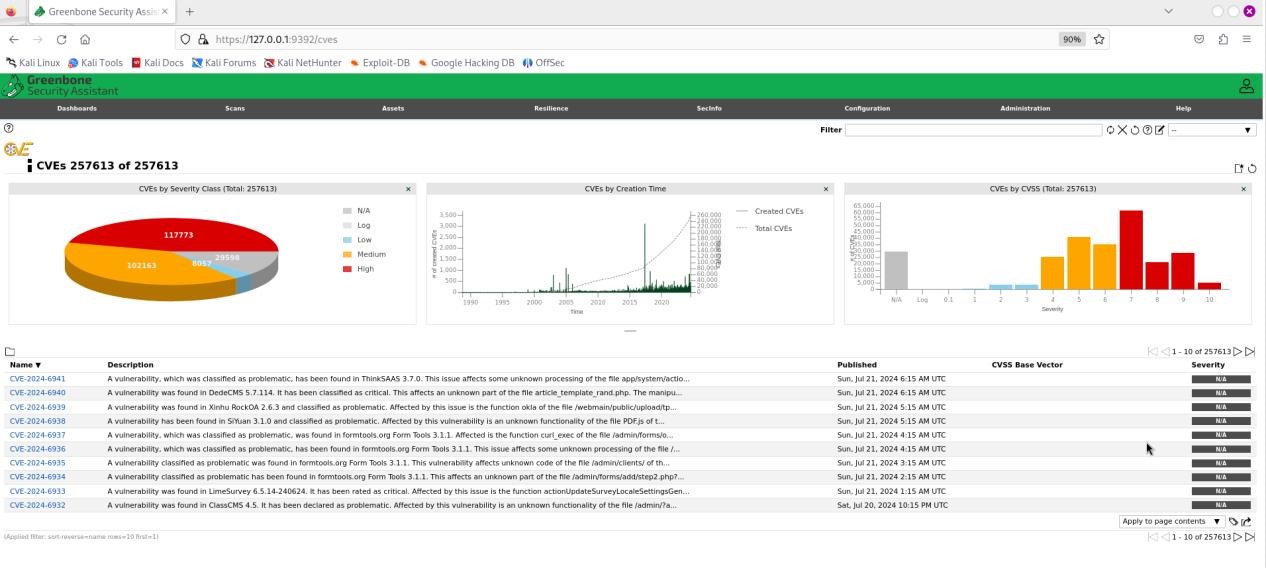
## ssl\_protocols TLSv1.2 TLSv1.3; sudo systemctl restart nginx sudo systemctl restart httpd

1. **SSL/TLS: HTTP Public Key Pinning (HPKP) Missing**

**Severity**: Medium

**Remediation Steps**:

* **Determine Keys**: Identify the public keys you want to pin, typically the Subject Public Key Info (SPKI) of your SSL/TLS certificates.
* **Add HPKP Header**: Add the Public-Key-Pins HTTP header to your web server configuration with the appropriate keys.
* **Deploy and Test**: Deploy the configuration and test it to ensure the HPKP headers are correctly set and that the site remains accessible.
* **Monitor**: Regularly monitor the effectiveness and relevance of HPKP, considering the potential risks of lockout due to misconfiguration.



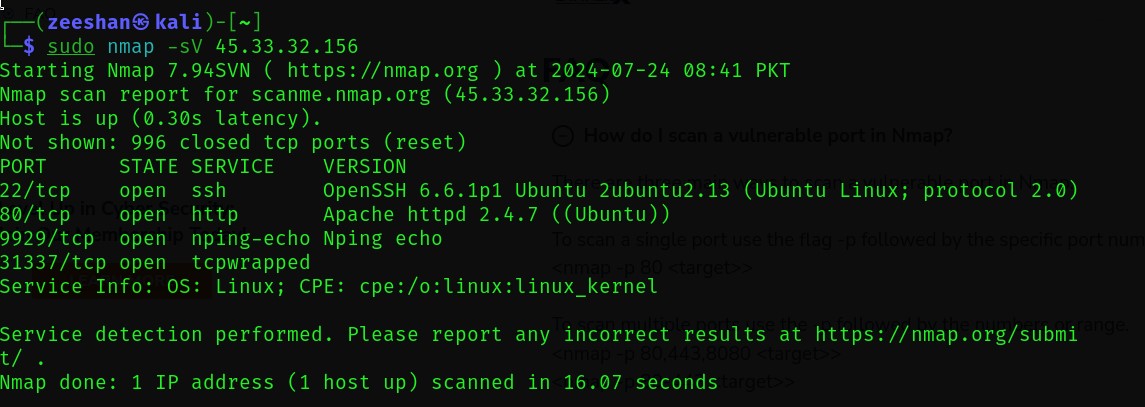
# Task 3: Network Scanning with Nmap

Namp:

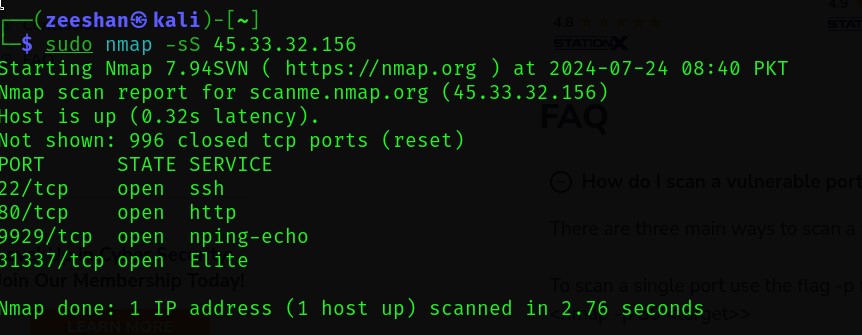
sudo apt install nmap

Network: 45.33.32.156

* **SyN Scan:**



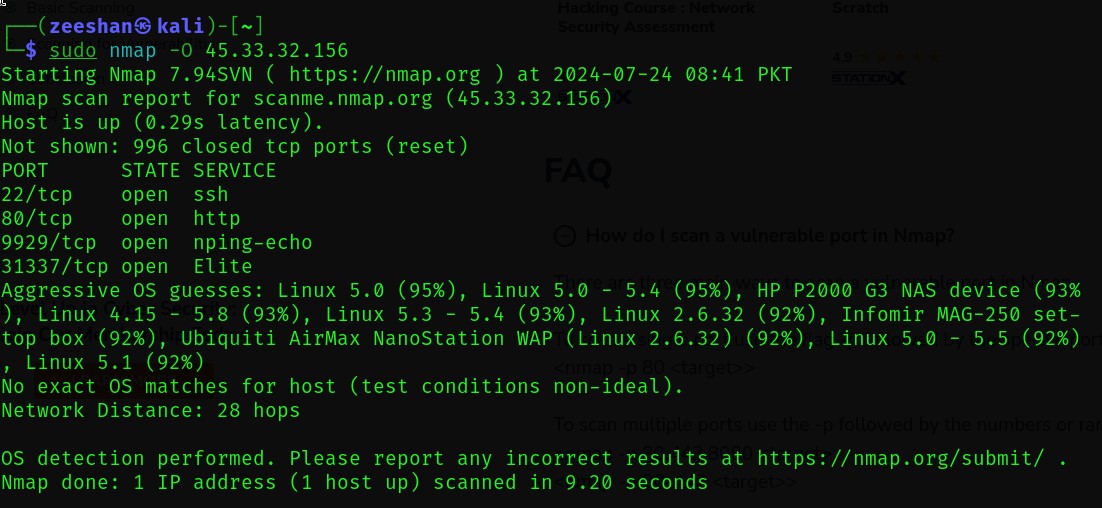
* **Sv service scan:**



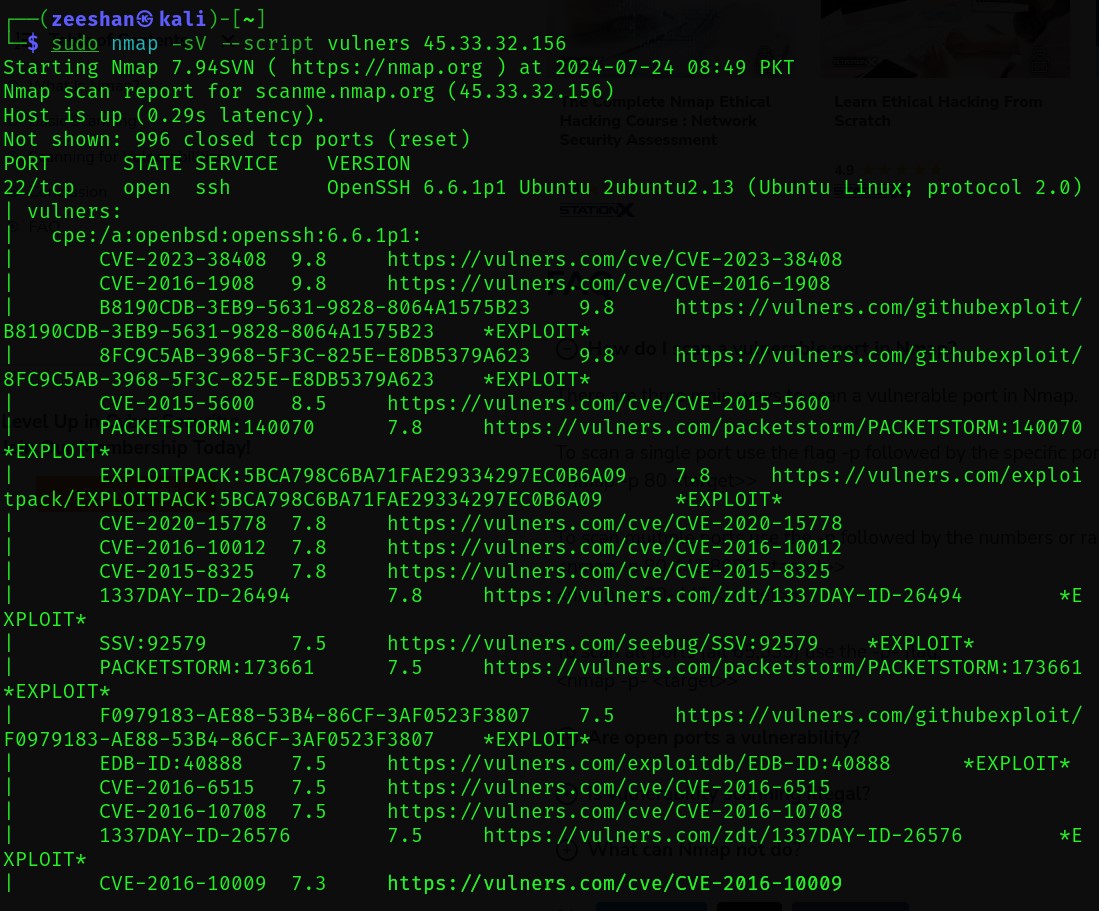
* **OS detection:**

Aggressive OS guesses: Linux 5.0 (95%), Linux 5.0 - 5.4 (95%), HP P2000 G3 NAS device (93%), Linux

4.15 - 5.8 (93%), Linux 5.3 - 5.4 (93%), Linux 2.6.32 (92%), Infomir MAG-250 set-top box (92%), Ubiquiti AirMax NanoStation WAP (Linux 2.6.32) (92%), Linux 5.0 - 5.5 (92%), Linux 5.1 (92%)



* **Vvulnerability Scaning:**



| CVE-2023-51385 6.5 [https://vulners.com/cve/CVE-2023-51385](https://vulners.com/cve/CVE-2023-51385%20)

| EDB-ID:40858 6.4 [https://vulners.com/exploitdb/EDB-ID:40858](https://vulners.com/exploitdb/EDB-ID:40858%20%20%20%20%20%20%20)

| EDB-ID:40119 6.4 [https://vulners.com/exploitdb/EDB-ID:40119](https://vulners.com/exploitdb/EDB-ID:40119%20%20%20%20%20%20%20)

| EDB-ID:39569 6.4 [https://vulners.com/exploitdb/EDB-ID:39569](https://vulners.com/exploitdb/EDB-ID:39569%20%20%20)

| CVE-2016-3115 6.4 [https://vulners.com/cve/CVE-2016-3115](https://vulners.com/cve/CVE-2016-3115%20)

| EDB-ID:40136 5.9 [https://vulners.com/exploitdb/EDB-ID:40136](https://vulners.com/exploitdb/EDB-ID:40136%20%20%20%20%20%20%20)

| EDB-ID:40113 5.9 [https://vulners.com/exploitdb/EDB-ID:40113](https://vulners.com/exploitdb/EDB-ID:40113%20%20%20%20%20%20%20)

| CVE-2023-48795 5.9 [https://vulners.com/cve/CVE-2023-48795](https://vulners.com/cve/CVE-2023-48795%20)

| CVE-2020-14145 5.9 [https://vulners.com/cve/CVE-2020-14145](https://vulners.com/cve/CVE-2020-14145%20)

| CVE-2019-6111 5.9 [https://vulners.com/cve/CVE-2019-6111](https://vulners.com/cve/CVE-2019-6111%20)

| CVE-2016-6210 5.9 [https://vulners.com/cve/CVE-2016-6210](https://vulners.com/cve/CVE-2016-6210%20)

| EDB-ID:45939 5.3 [https://vulners.com/exploitdb/EDB-ID:45939](https://vulners.com/exploitdb/EDB-ID:45939%20%20%20%20%20%20%20)

| EDB-ID:45233 5.3 [https://vulners.com/exploitdb/EDB-ID:45233](https://vulners.com/exploitdb/EDB-ID:45233%20%20%20%20%20)

| CVE-2018-20685 5.3 [https://vulners.com/cve/CVE-2018-20685](https://vulners.com/cve/CVE-2018-20685%20)

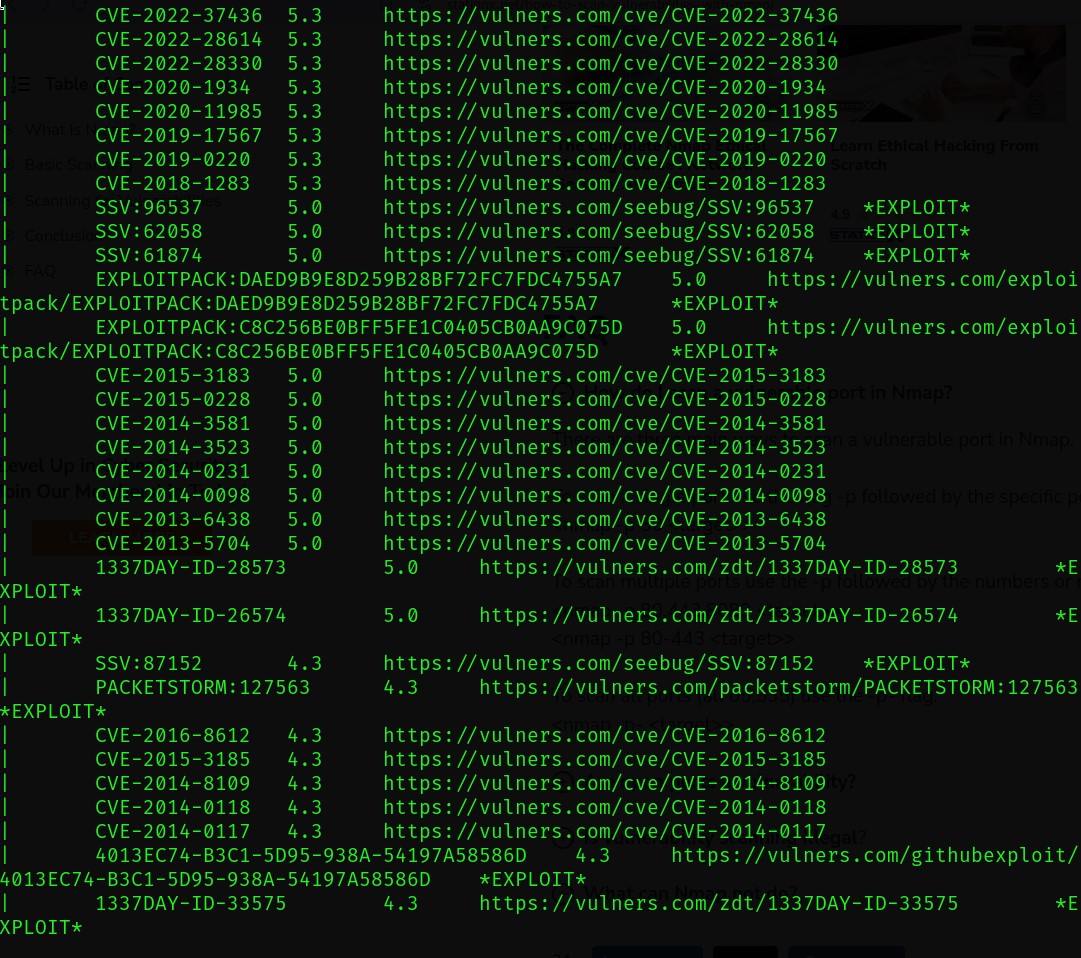
| CVE-2018-15919 5.3 [https://vulners.com/cve/CVE-2018-15919](https://vulners.com/cve/CVE-2018-15919%20)

| CVE-2018-15473 5.3 [https://vulners.com/cve/CVE-2018-15473](https://vulners.com/cve/CVE-2018-15473%20)

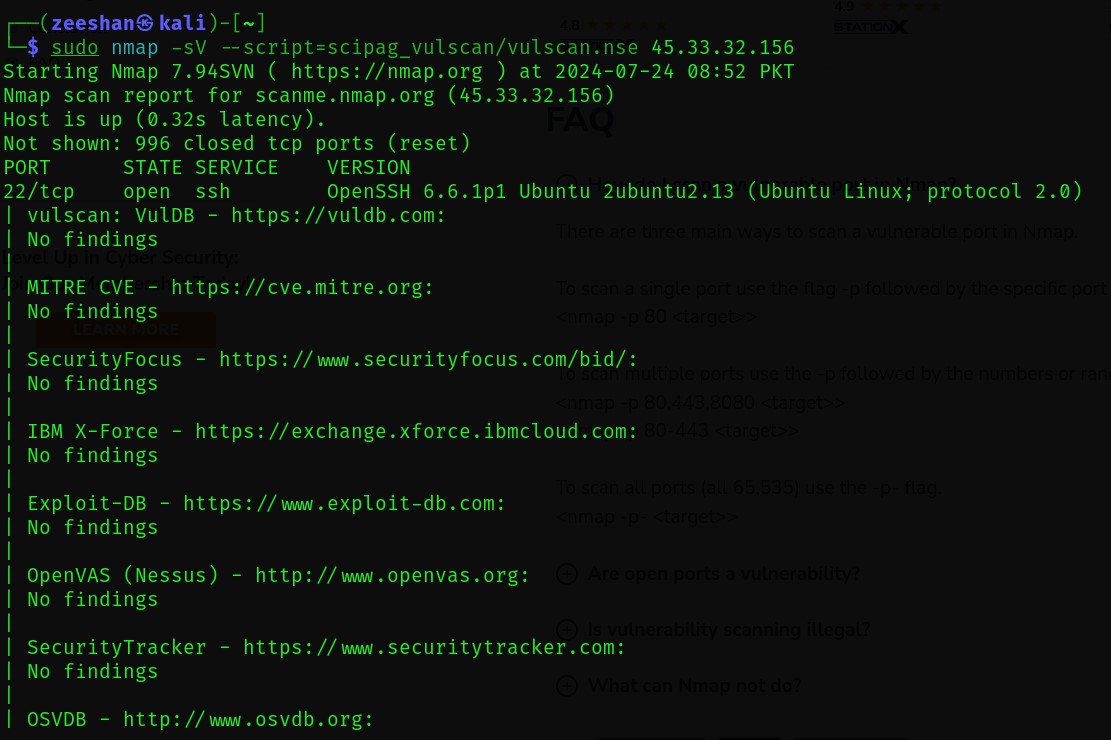
| CVE-2017-15906 5.3 [https://vulners.com/cve/CVE-2017-15906](https://vulners.com/cve/CVE-2017-15906%20)

| CVE-2016-20012 5.3 [https://vulners.com/cve/CVE-2016-20012](https://vulners.com/cve/CVE-2016-20012%20)

| SSH\_ENUM 5.0 [https://vulners.com/canvas/SSH\_ENUM](https://vulners.com/canvas/SSH_ENUM%20%20%20%20%20%20)



➢ **Exploiting the Networking:**



**Task 4: Web App Penetration Testing with OWASP ZAP**

1. **Setting Up OWASP ZAP**

**Step 1: Download and Install OWASP ZAP**

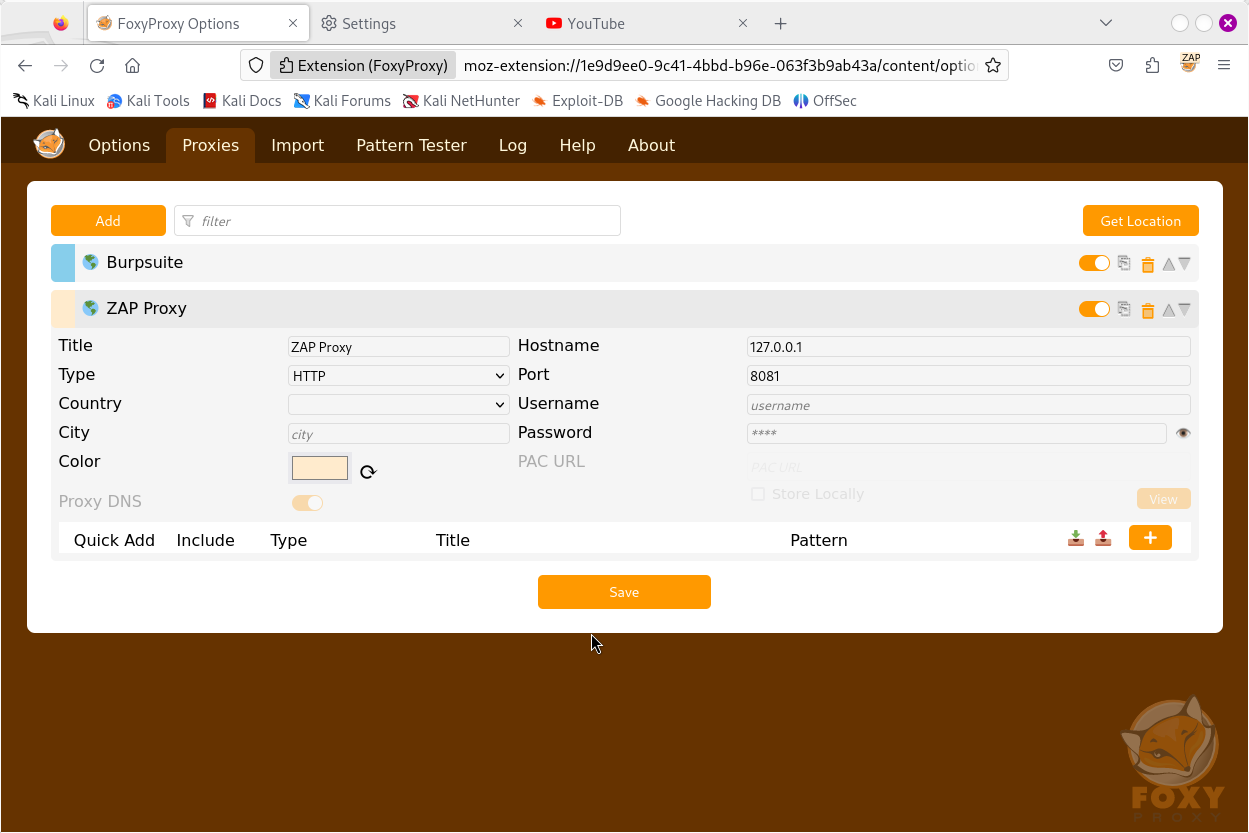
* 1. Visit the [official OWASP ZAP website.](https://www.zaproxy.org/)
  2. Download the appropriate version for your operating system.
  3. Install OWASP ZAP by following the on-screen instructions.

**Step 2: Configure Browser to Use ZAP as a Proxy**

1. Install the FoxyProxy extension in your browser.

* + 1. For Firefox: Go to the [FoxyProxy Firefox Add-ons page](https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/) and install the extension.
    2. For Chrome: Go to the FoxyProxy Chrome Web Store page and install the extension.

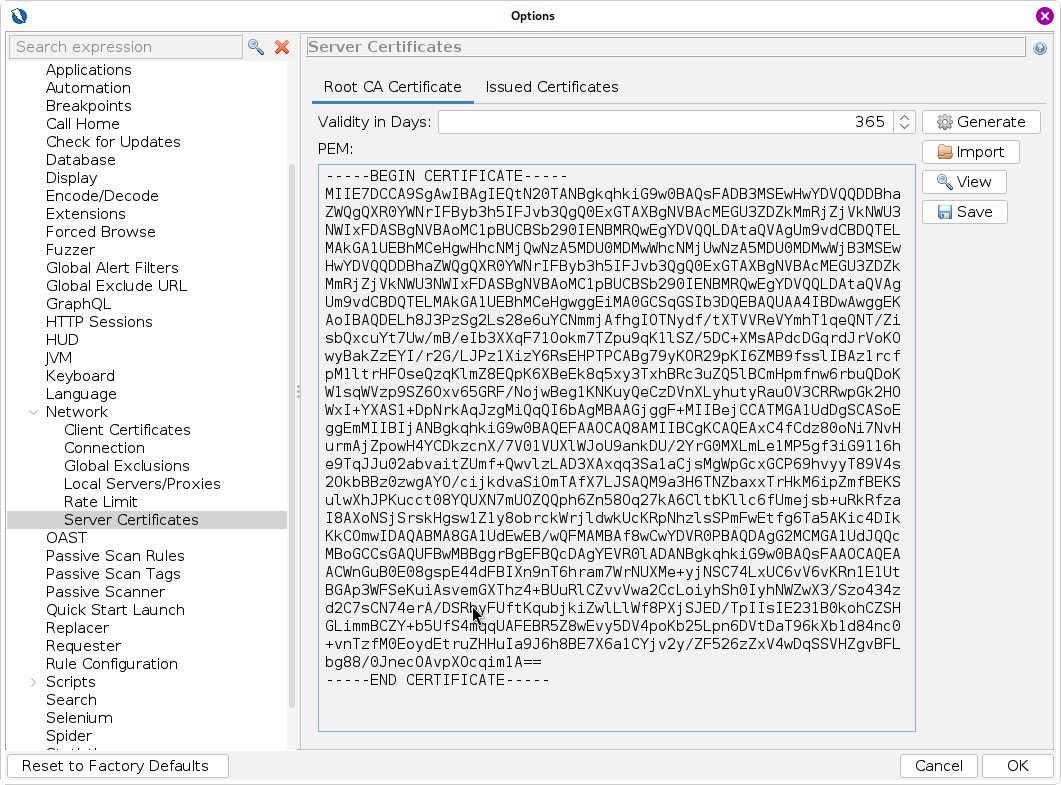
1. Open FoxyProxy and add a new proxy configuration.
   * 1. Proxy IP: **127.0.0.1**
     2. Proxy Port: **8081**
     3. Select HTTP as the protocol.
2. Enable the new proxy configuration in FoxyProxy.



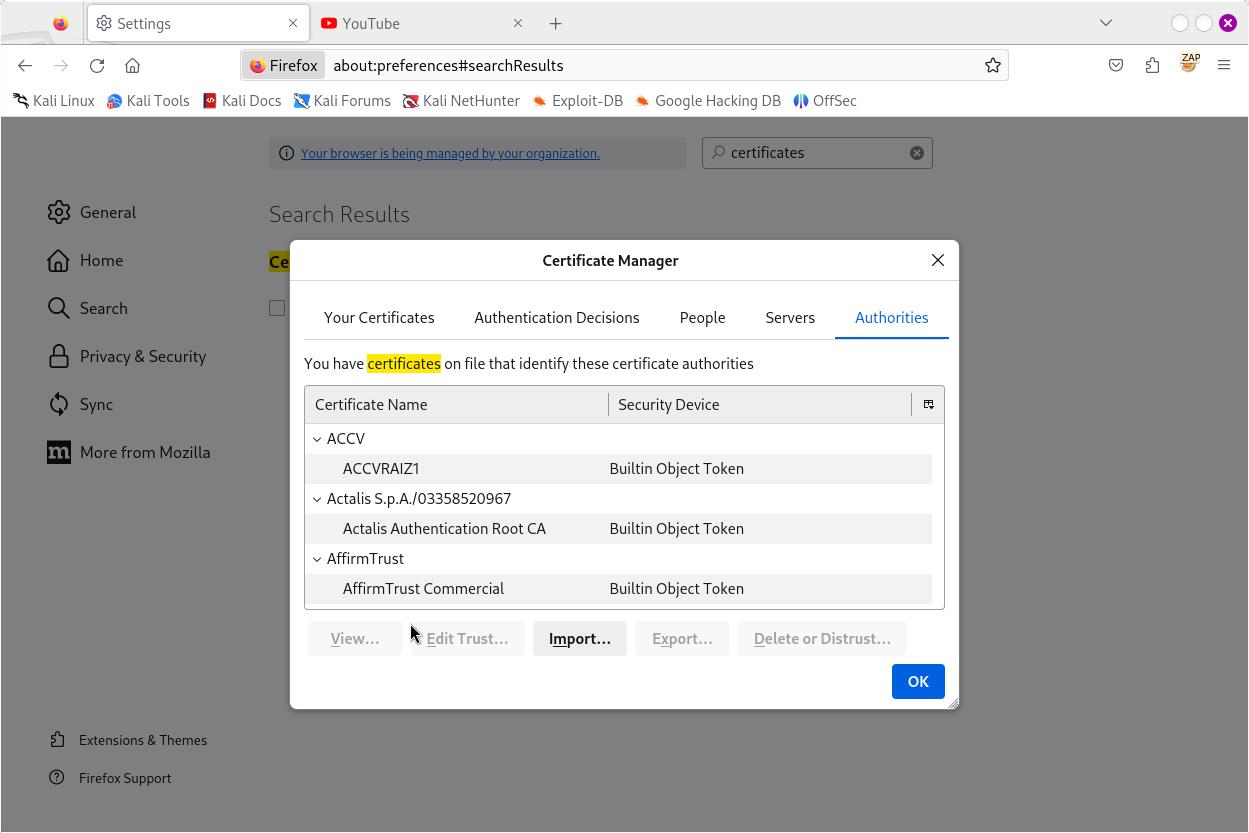
**Step 3: Install ZAP SSL Certificate in Browser**

1. Open OWASP ZAP and navigate to T**ools > Options > Dynamic SSL Certificates**.
2. Click Generate if it’s not already generated, then click Save to export the

certificate.

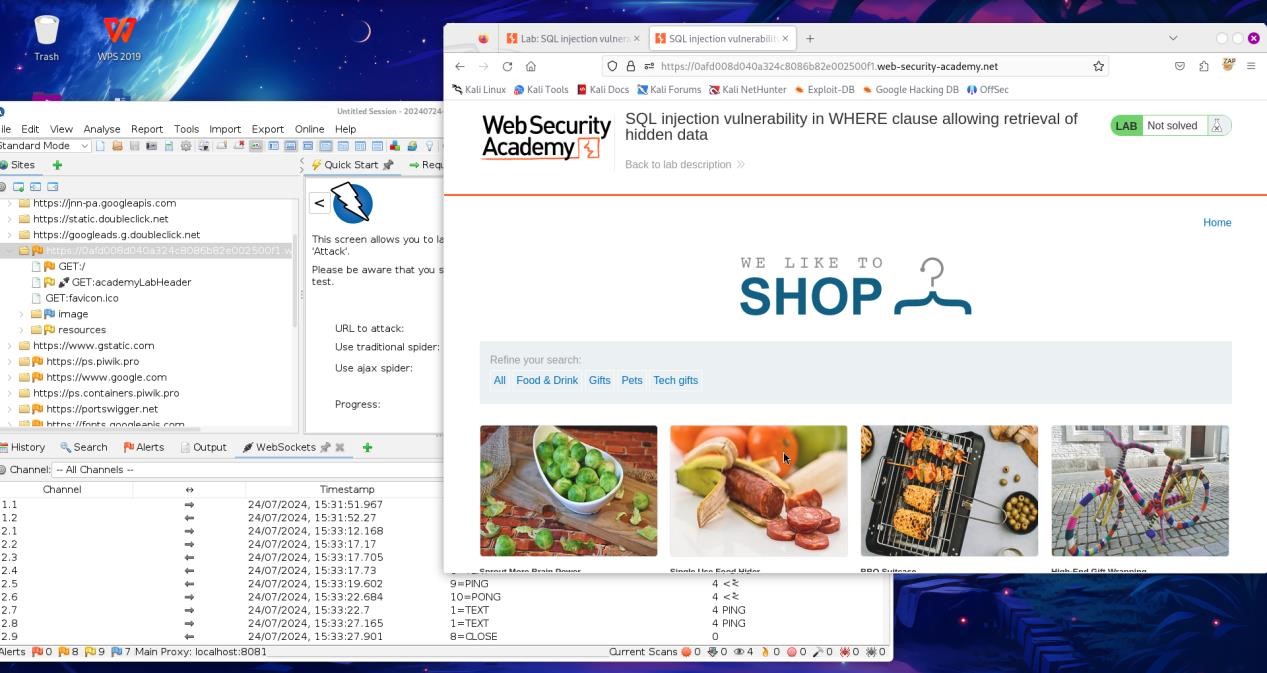


1. In your browser, go to settings and search for "certificates".
   1. For Firefox: Go to **Preferences > Privacy & Security > Certificates > View Certificates**.
   2. For Chrome: Go to **Settings > Privacy and security > Security > Manage certificates**.
2. Import the ZAP certificate and mark it as trusted for web traffic.



**Step 4: Intercept Traffic**

1. With the proxy settings and certificate in place, your browser should now be routing traffic through OWASP ZAP.
2. Open a web application in your browser to see the traffic being intercepted by ZAP.

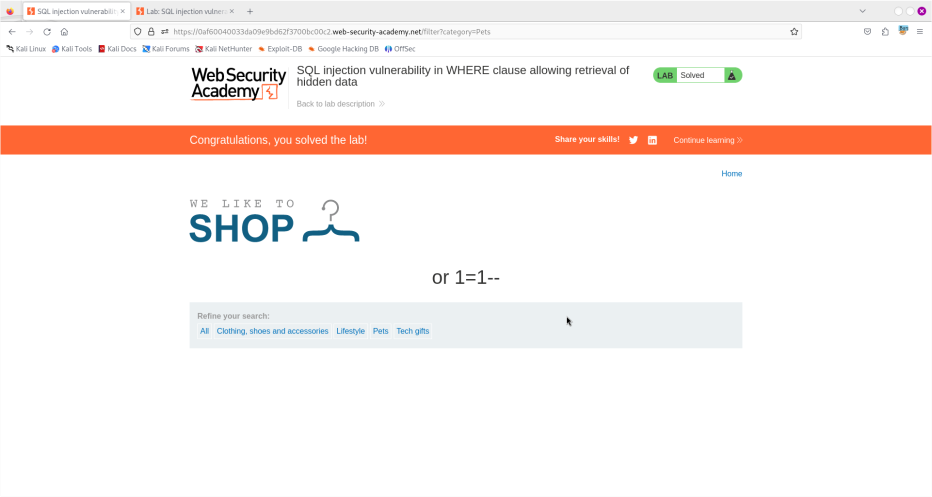


* **SQL injection:** 
  + network:https://0afd008d040a324c8086b82e002500f1.web-securityacademy.net/

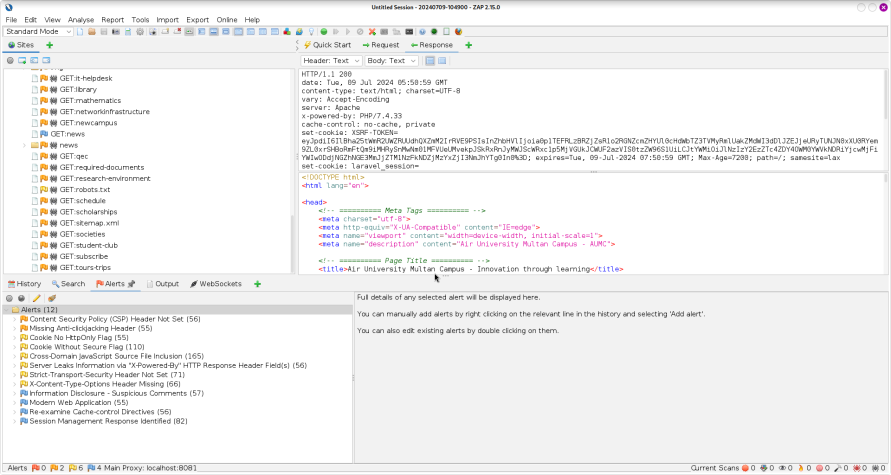
* + portswigger SQL injection practical Lab.



* + in the category button, replace with ‘or+1=1--’ , this unhide all the information and products on the website.



* **Scaning Network:**



**Short Project:**

**Comprehensive Security Assessment and Hardening of a**

**Small Business Network**

Tools used:

* Nmap
* Wireshark
* OpenVas (Greenbone)
* Nessus

* Network: 203.124.43.0/24

* Domain’s info:-

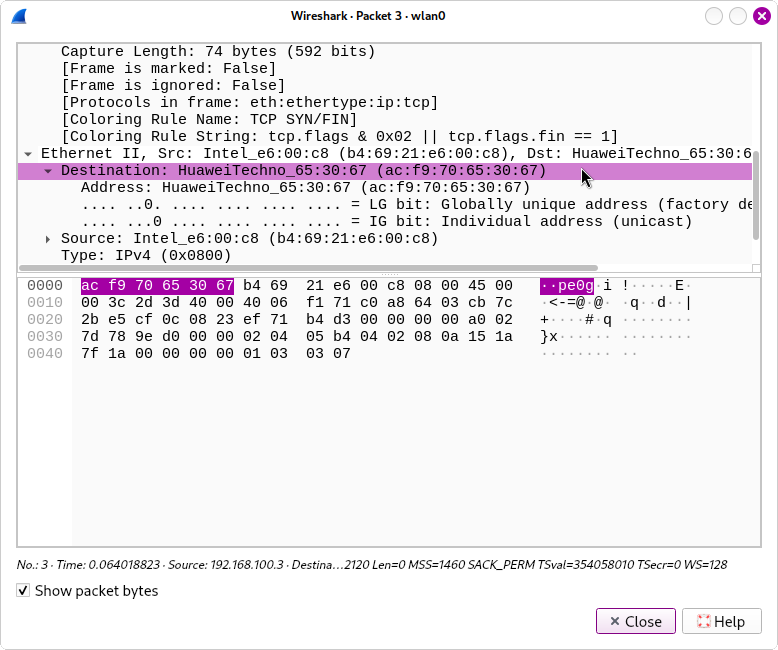


Server’s ip: 203.124.43.229

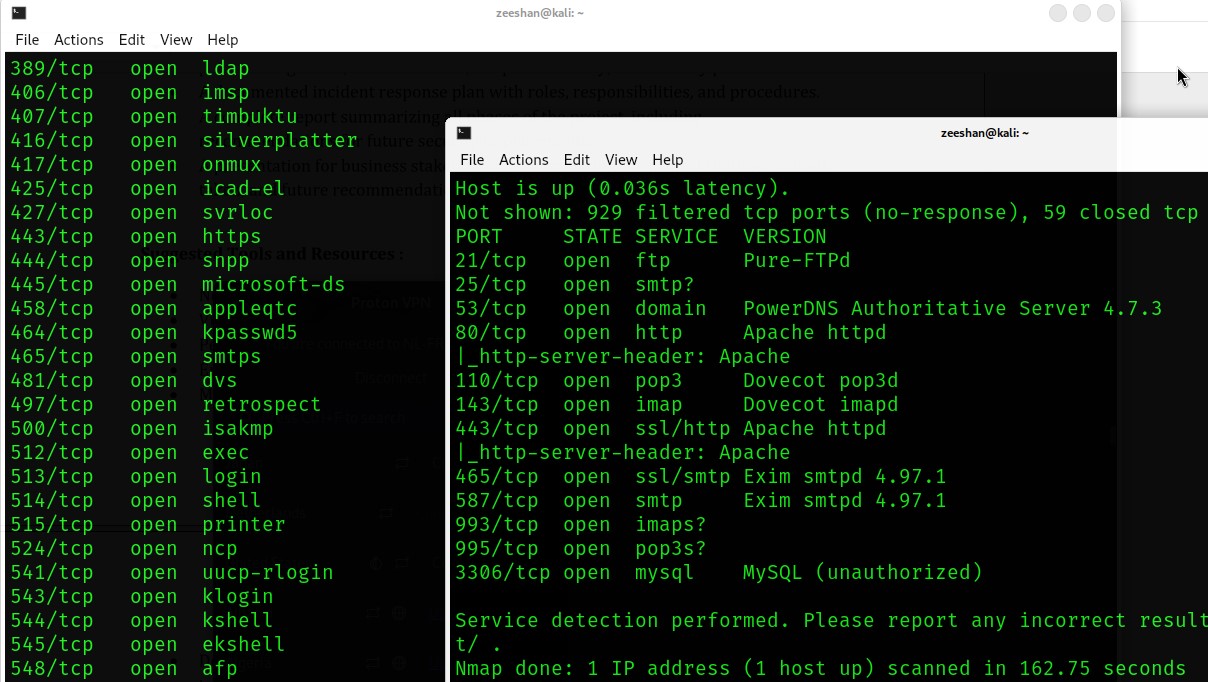
Mac Address: ac:f9:70:65:30:67

Technology: HuaweiTechno

Domain Hosting: host201003.comsatshosting.com



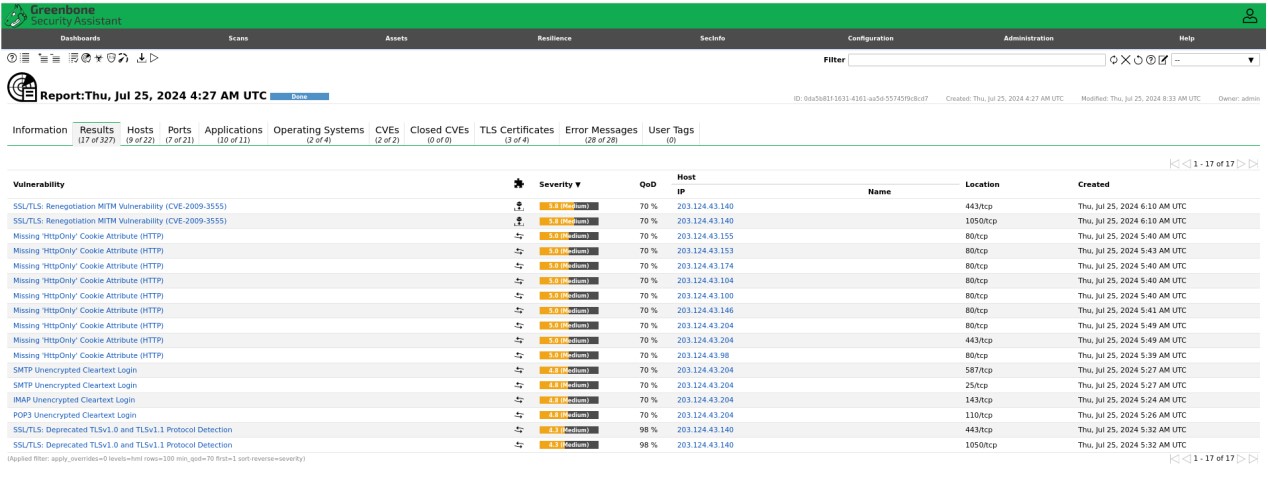
* More than 1000+ ports are opened.



|  |  |
| --- | --- |
| **Ip Address** | **Service** |
| 203.124.43.1 | HP P2000 G3 NAS device |
| 203.124.43.2 | OpenBSD 4.X |
| 203.124.43.21 | Netgear WGR614v7 wireless broadband router |
| 203.124.43.65 | Asus WL-500gP wireless broadband router |
| 203.124.43.66 | Cisco C7200 router (IOS 15) |
| 203.124.43.73 | AVtech Room Alert 26W environmental monitor |
| 203.124.43.80 | D-Link DFL-700 firewall |
| 203.124.43.81 | Cisco Aironet 1250 WAP (IOS 12.4) |
| 203.124.43.87 | D-Link DFL-700 firewall |
| 203.124.43.89 | Cisco Aironet 1250 WAP (IOS 12.4) |
| 203.124.43.96 | D-Link DFL-700 firewall |
| 203.124.43.97 | Cisco Aironet 1250 WAP (IOS 12.4) |
| 203.124.43.98 | Ubiquiti AirMax NanoStation WAP (Linux 2.6.32) |
| 203.124.43.100 | Infomir MAG-250 set-top box |
| 203.124.43.102 | Ubiquiti Pico Station WAP (AirOS 5.2.6) |
| 203.124.43.104 | Ubiquiti Pico Station WAP (AirOS 5.2.6) |
| 203.124.43.107 | Infomir MAG-250 set-top box |
| 203.124.43.108 | Infomir MAG-250 set-top box |
| 203.124.43.111 | D-Link DFL-700 firewall |
| 203.124.43.112 | D-Link DFL-700 firewall |
| 203.124.43.113 | Cisco C7200 router |
| 203.124.43.119 | D-Link DFL-700 firewall |
| 203.124.43.121 | Cisco C7200 router |
| 203.124.43.132 | linux:linux\_kernel |
| 203.124.43.136 | linux:linux\_kernel |
| 203.124.43.137 | linux:linux\_kernel |
| 203.124.43.140 | Huawei embedded |
| 203.124.43.141 | Cyberoam UTM firewall |
| 203.124.43.143 | Adaptec Snap Server 100 NAS device |
| 203.124.43.144 | D-Link DFL-700 firewall |
| 203.124.43.146 | AirMax NanoStation WAP (Linux 2.6.32) |
| 203.124.43.150 | Dropbear sshd (protocol 2.0) |
| 203.124.43.202 | WGR614v7 wireless broadband router |
| 203.124.43.203 | Microsoft Windows |
| 203.124.43.204 | Linux 5.0 - 5.4 |
| 203.124.43.206 | Mail Server Windows |
| 203.124.43.210 | PowerDNS Authoritative Server |
| 203.124.43.225 | AVtech embedded |
| 203.124.43.226 | Server |
| 203.124.43.227 | Server |
| 203.124.43.228 | Server |

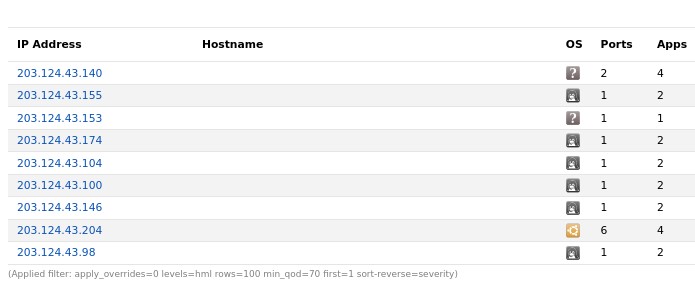
These are the devices connected with the network.

# Vulnerabilities:-



|  |  |  |
| --- | --- | --- |
| **Vulnerability** | **Severity** | **Ip Address** |
| SSL/TLS: Renegotiation MITM  Vulnerability (CVE-2009-3555) | 5.8 | [203.124.43.140](https://127.0.0.1:9392/host/00d3b8d8-92bb-473e-b1e5-820bb46847ae) |
| Missing 'HttpOnly' Cookie Attribute (HTTP) | 5.0 | [203.124.43.155](https://127.0.0.1:9392/host/708c9dbf-2100-4c27-b07f-3a805a67d5ae) |
| Missing 'HttpOnly' Cookie Attribute (HTTP) | 5.0 | [203.124.43.153](https://127.0.0.1:9392/host/da9204eb-fa37-4ba4-aeb0-a8e4be3e9025) |
| Missing 'HttpOnly' Cookie Attribute (HTTP) | 5.0 | [203.124.43.174](https://127.0.0.1:9392/host/79585a8a-43b6-41d0-9fcb-686ddb96c229) |
| Missing 'HttpOnly' Cookie Attribute  (HTTP)  Missing 'HttpOnly' Cookie Attribute (HTTP) | 5.0 | [203.124.43.104](https://127.0.0.1:9392/host/e5c36b6d-906a-4bd2-acda-f574e211f04c) |
| Missing 'HttpOnly' Cookie Attribute (HTTP) | 5.0 | [203.124.43.100](https://127.0.0.1:9392/host/7a34bf03-f94d-4ac6-99f9-ee4728bf7aec) |
| Missing 'HttpOnly' Cookie Attribute (HTTP) | 5.0 | [203.124.43.146](https://127.0.0.1:9392/host/6837e448-f254-4c12-8f77-935302f6cfec) |
| Missing 'HttpOnly' Cookie Attribute | 5.0 | [203.124.43.204](https://127.0.0.1:9392/host/881be3ff-dd19-4efd-a358-14ee57003f36) |
| Missing 'HttpOnly' Cookie Attribute | 5.0 | [203.124.43.204](https://127.0.0.1:9392/host/881be3ff-dd19-4efd-a358-14ee57003f36) |
| (HTTP) |  |  |
| SMTP Unencrypted Cleartext Login | 4.8 | [203.124.43.204](https://127.0.0.1:9392/host/881be3ff-dd19-4efd-a358-14ee57003f36) |
| IMAP Unencrypted Cleartext Login | 4.8 | [203.124.43.204](https://127.0.0.1:9392/host/881be3ff-dd19-4efd-a358-14ee57003f36) |
| POP3 Unencrypted Cleartext Login | 4.8 | [203.124.43.204](https://127.0.0.1:9392/host/881be3ff-dd19-4efd-a358-14ee57003f36) |
| SSL/TLS: Deprecated TLSv1.0 and  TLSv1.1 Protocol Detection | 4.3 | [203.124.43.140](https://127.0.0.1:9392/host/00d3b8d8-92bb-473e-b1e5-820bb46847ae) |

# ➢ Hosts:-



# ➢ TLS certificates:-



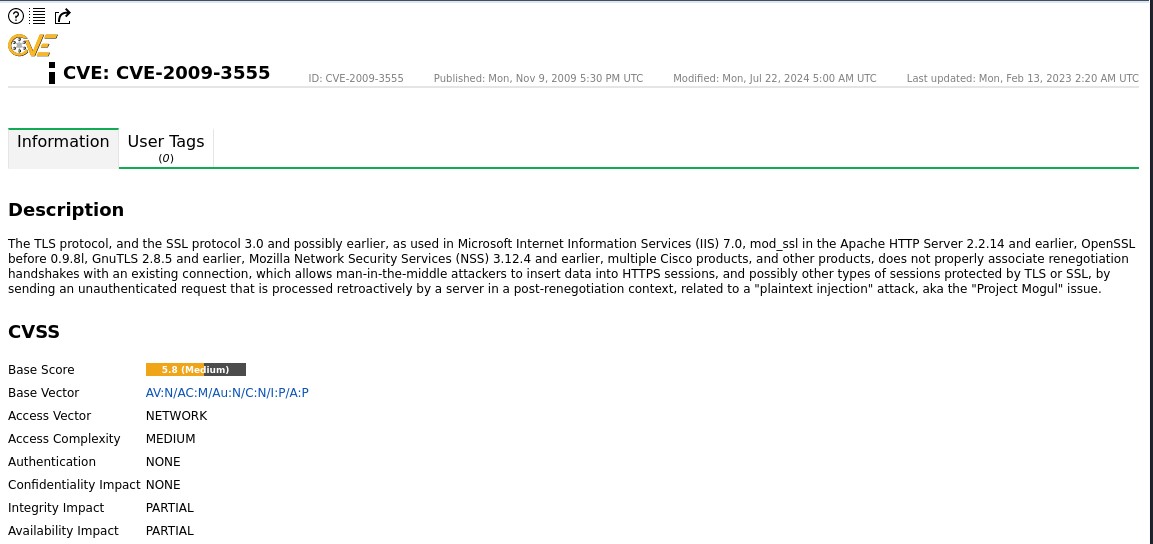
# CVE: CVE-2009-3555

The vulnerability lies in the way the TLS protocol handles renegotiation requests. Renegotiation is a feature that allows a client and server to renegotiate the parameters of an existing TLS connection. However, in the affected versions of the protocol, there was no cryptographic binding between the renegotiation handshake and the original handshake, making it possible for an attacker to inject arbitrary content into the renegotiation process.

## Impact

An attacker could exploit this vulnerability to perform a Man-in-the-Middle (MitM) attack. By intercepting the communication between a client and server, the attacker could inject malicious data into the TLS session, potentially leading to data breaches, session hijacking, or other forms of unauthorized access.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2009-3555>

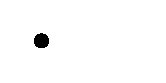


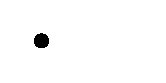
## Detailed Solution for CVE-2009-3555

CVE-2009-3555, the TLS Renegotiation Vulnerability, requires both immediate and long-term solutions to ensure the security of communications using the TLS protocol. Below is a detailed approach to addressing this vulnerability:

### **1. Understanding the Vulnerability**

Before diving into solutions, it’s important to understand the root cause:

 The vulnerability exists because TLS renegotiation requests are not cryptographically tied to the original handshake.

 This allows an attacker to inject data into the session during the renegotiation process, potentially leading to Man-in-the-Middle (MitM) attacks.

### **2. Immediate Mitigation**

While working on a long-term fix, immediate steps should be taken to mitigate the risk:

• **Disable TLS Renegotiation**: Temporarily disable renegotiation on servers. This can usually be configured in the server’s TLS/SSL settings.

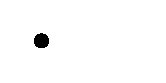
o For example, in Apache HTTP Server, add the following to the configuration file:

### SSLInsecureRenegotiation off

#### **3. Long-term Solution**

The ultimate fix involves updating the TLS implementation to support the Renegotiation Indication Extension as defined in RFC 5746. Here’s a step-by-step guide:

### Step 1: Update SSL/TLS Libraries



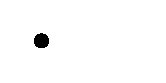
**OpenSSL**: Ensure you are using OpenSSL version 0.9.8m or later, which includes the renegotiation fix.

#### sudo apt-get update sudo apt-get install openssl

**GnuTLS**: Ensure you are using GnuTLS version 2.8.6 or later. sudo apt-get update

sudo apt-get install gnutls-bin

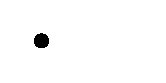
### **Step 3: Verify Renegotiation Fix**

 Use tools to verify that your server properly supports the renegotiation extension. o **Qualys SSL Labs**: Run an SSL Test on your server. Look for the section on renegotiation. It should indicate secure renegotiation support.

o **OpenSSL Command**: Use the openssl s\_client command to check renegotiation:

openssl s\_client -connect yourserver.com:443 -tls1

### **Step 4: Client-Side Updates**

 Ensure that all client software and libraries that connect to your server also support the renegotiation extension.

o Browsers and other client applications typically update automatically, but verify that your internal applications are using updated libraries.

### **Step 5: Testing and Monitoring**

* After implementing the fix, thoroughly test your server to ensure that all services are functioning correctly and securely.
* Continuously monitor for any unusual activity that might indicate attempted exploitation of renegotiation vulnerabilities.

#### **4. Documentation and Training**

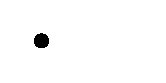
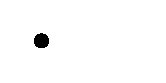
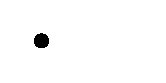
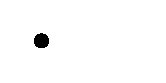
* Document the changes made and ensure that all IT staff are aware of the implications of the vulnerability and the applied fixes.
* Train relevant personnel on the importance of keeping SSL/TLS libraries and server software up to date.

# CVE: CVE-2015-0204

CVE-2015-0204, also known as the "FREAK" (Factoring Attack on RSA-EXPORT Keys) vulnerability, affects several SSL/TLS implementations. Here are the key details:

## Overview

**Vulnerability Name:** FREAK (Factoring Attack on RSA-EXPORT Keys)



**CVE ID:** CVE-2015-0204

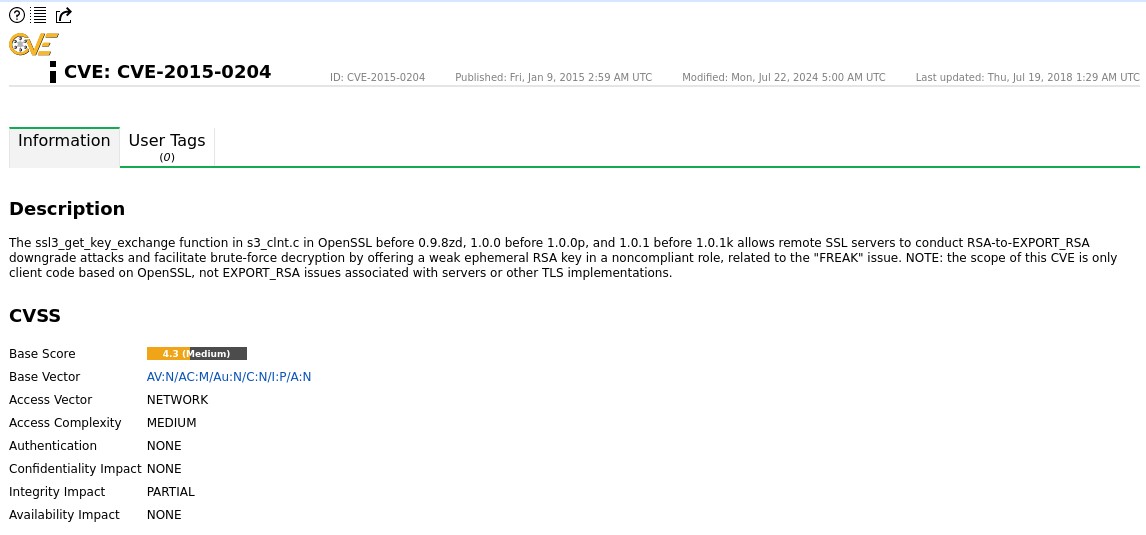
**Published:** January 8, 2015

**Description:** This vulnerability allows a man-in-the-middle (MITM) attacker to downgrade the security of an SSL/TLS connection to use weak RSA "export" keys, which can be easily broken, making it possible to intercept and decrypt encrypted communications.

## Affected Systems

* **Software:** OpenSSL, Apple's Secure Transport, and other SSL/TLS implementations.
* **Devices:** A wide range of devices and software that rely on these SSL/TLS libraries for secure communications.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0204>



## Detailed Solution for CVE-2015-0204 (FREAK Vulnerability)

The solution to CVE-2015-0204 involves several steps to ensure that systems are no longer vulnerable to the FREAK attack. Here is a comprehensive guide:

## Step 1: Identify Affected Systems

**Inventory Assessment**:

* Identify all systems, applications, and devices that use SSL/TLS for secure communications.
* Determine which systems use affected libraries like OpenSSL, Apple's Secure Transport, or others.

**Check Versions**:

* Ensure the versions of SSL/TLS libraries in use. Vulnerable versions of OpenSSL, for example, need to be identified.

## Step 2: Apply Patches and Updates

**Update OpenSSL**:

* Upgrade to a non-vulnerable version of OpenSSL. The following versions include the fix:
  + OpenSSL 1.0.2
  + OpenSSL 1.0.1k ◆ OpenSSL 1.0.0p
  + OpenSSL 0.9.8zd
* Use package managers (like apt-get, yum, etc.) to update OpenSSL:

sudo apt-get update sudo apt-get upgrade openssl

## Step 3: Configuration Changes

**Disable Export Cipher Suites**:

* Modify server configurations to disable support for export-grade cipher suites. For example, in Apache HTTP Server, edit the SSL configuration file:

SSLCipherSuite HIGH:!aNULL:!MD5:!RC4:!EXP SSLHonorCipherOrder on

**Test Configurations**:

* Use tools like *nmap* or *ssllabs.com* to verify that the export cipher suites are disabled.

**Restart Services**:

* Restart the affected services to apply the new configurations

sudo systemctl restart apache2 sudo systemctl restart nginx

## Step 4: Verification and Monitoring

**Verify Fixes**:

* Use vulnerability scanners or security tools to verify that the FREAK vulnerability has been mitigated.
* Perform tests to ensure that the services are functioning correctly after updates and configuration changes.

**Continuous Monitoring**:

* Keep systems updated with the latest security patches.
* Monitor security advisories for new vulnerabilities and fixes.

**THE END**