**2/c Ishimwe Vivine**

**Computer and Network Security**

**11 December 2020**

**Penetration Testing**

**Introduction**

I am Vivine Ishimwe, a security professional. I am conducting this penetration test at the request of director of Ishimwe pioneer organization to validate security requirements and understand the security posture of this organization. The purpose of this test is to use different tools to discover vulnerabilities in Windows XP, and Linux machines and remediation steps moving towards NIST 800-53 compliance in accordance with the open source security Testing Methodology Manual.

**Objective**

The objective of this assessment is to perform an internal penetration test against Windows7 XP, and Linux virtual machines by evaluating the network and security issues associated with them, identifying systems, exploiting vulnerabilities and developing the countermeasures to mitigate them while reporting the findings back to Ishimwe Pioneers Organization.

**Requirements**

The security professional will be required to fill out this penetration testing report fully and to include the following sections:

* Overall High-Level Summary and Recommendations
* Methodology walkthrough and detailed outline of steps taken
* Each finding with included screenshots
* Any additional items that were not included

**High-Level Summary**

I am a security professional, tasked with performing an internal penetration test on win 7, Linux, and win7 XP virtual machines. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to use kali Linux is to perform attacks, similar to those of a hacker and attempt to infiltrate the virtual machines systems. My overall objective is to evaluate the network, identify systems, and exploit vulnerabilities while reporting the findings back to Ishimwe Pioneers Organization.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Win XP, and Linux virtual machines. When performing the attacks, I was able to gain access to machines, primarily due to outdated patches and poor security configurations. During the testing, I was able to gain administrative level access through some vulnerabilities. some systems were successfully exploited, and access granted, however some machines prevented me from gaining the access. These systems as well as a brief description on how access was obtained are listed below:

**Windows XP**

* 192.168.0.58 (cns-c622dafb194.hitronhub.home) – Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
* 192.168.0.58 (cns-c622dafb194.hitronhub.home) - Microsoft Windows system vulnerable to remote code execution (MS08-067)
* 192.168.0.58(cns-c622dafb194.hitronhub.home) - SSL POODLE information leak
* 192.168.0.58(cns-c622dafb194.hitronhub.home) - SSL/TLS MITM vulnerability (CCS Injection)

**Linux**

* 192.168.0.59 (Ubuntu.hitronhub.home) - Apache byte range filter DoS
* 192.168.0.59 (Ubuntu.hitronhub.home) - ftp-vsFTPd-backdoor
* 192.168.0.59(Ubuntu.hitronhub.home) - Slowloris DOS attack

**Recommendations**

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date. For instance, continuous updating of Ubuntu software and packages will likely protect the organization against the attackers. For each vulnerability found, I provided the way to fix it. Thus, I recommend using vulnerability fixes provided to maintain the security of these machines against the attackers.

**Methodologies**

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Win XP and Linux machines are secured. I utilized internal vulnerability scanning which was done by using Nmap in kali Linux. In addition to Nmap, I also used Nessus essentials in order to scan the external ports. By doing so, I was able to determine numerous vulnerabilities that were present on Win XP and Linux machines with which the attackers can take advantage of. After determining the vulnerabilities present on these machines, I utilized Metasploit in order to gain access and provided the solution to each vulnerability found. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

**Information Gathering**

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting two virtual machines. I was able to exploit them by using Metasploit and information gathered from Nmap about the IPs of these machines. The specific IP addresses were:

192.168.0.58 for Windows XP host

192.168.0.59 for Linux host

**Penetration**

The penetration testing portions of the assessment focus heavily on gaining access to Win7 XP and Linux Virtual machines by using Metasploit and Nmap tools. During this penetration test, I was able to successfully gain access to **2 machines (Win7 XP and Linux)** out of the **3** systems.

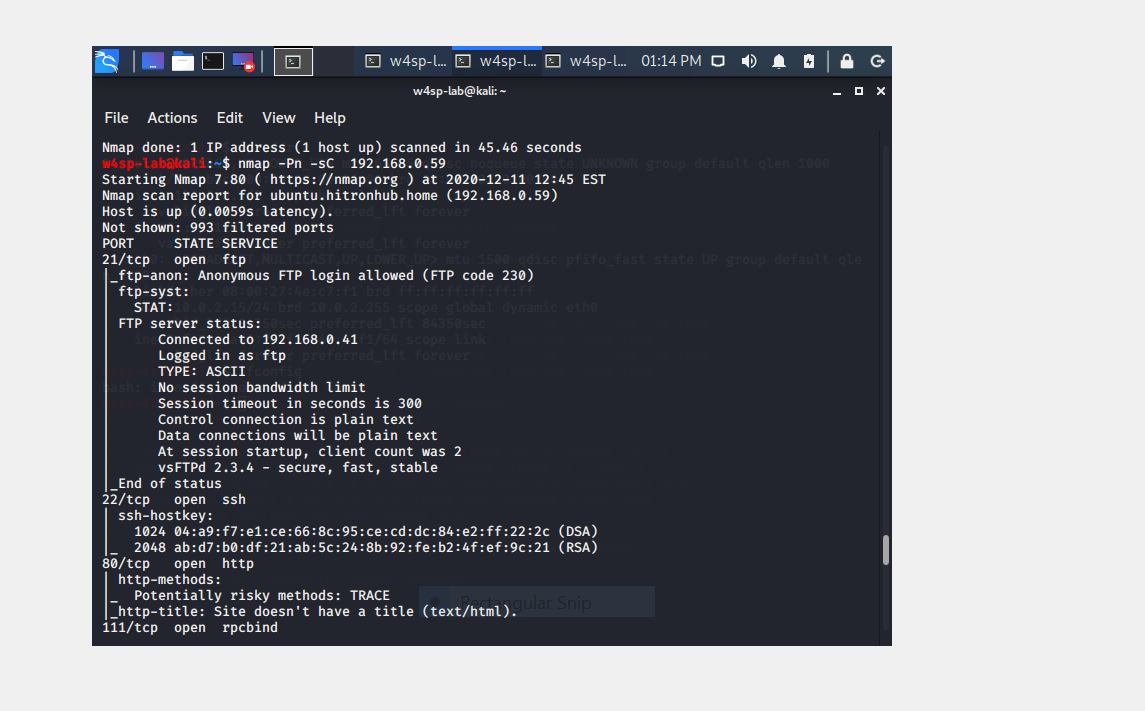
**System IP: 192.168.0.58, 192.168.0.59**

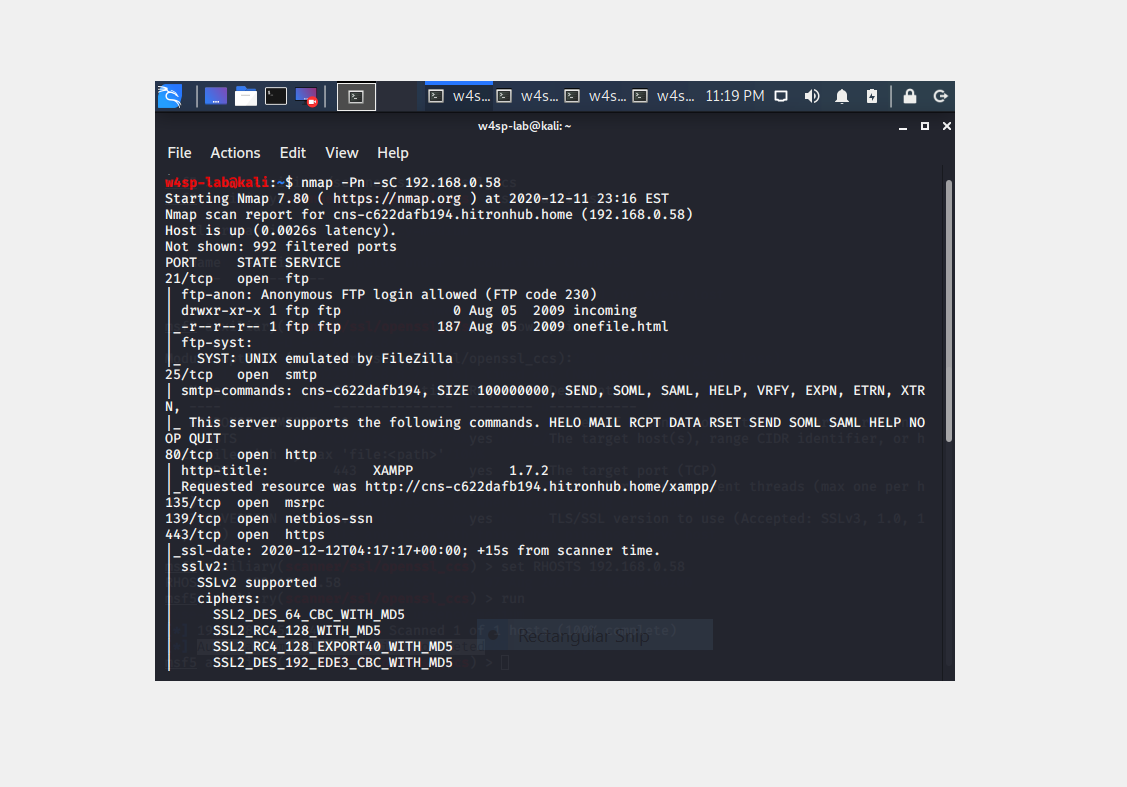
**Service Enumeration**

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

|  |  |
| --- | --- |
| Service IP address | Ports Open |
| 192.168.0.58 | TCP: 21,25,80, 135,139,443, 445,3306  UDP: 123,137 |
| 192.168.0.59 | TCP: 21,22,80,111,139,445,2049  UDP: 111,137,2049,5353 |

**Nmap Scan Results:**





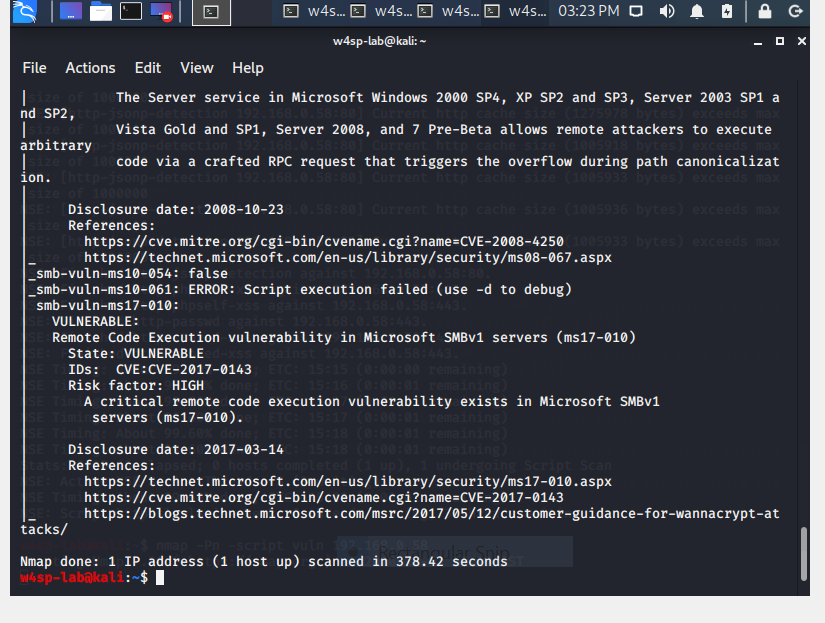
Information about the system: By using Nmap, I tried to exploit both machines. on port 80 on Linux Machine, I found out that there is potentially risky HTTP TRACE method. This could be a threat to the company because an attacker can use it to steal secrets from legitimate users. I tried to gain access on other open ports on this machine however I was not able to which means this machine is secure from certain attacks.

**System IP: 192.168.0.58**

**Service Enumeration**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| 192.168.0.58 | **TCP**: 21,25,80,135,139,443,445,3306 |
| **UDP**: | 123,137 |

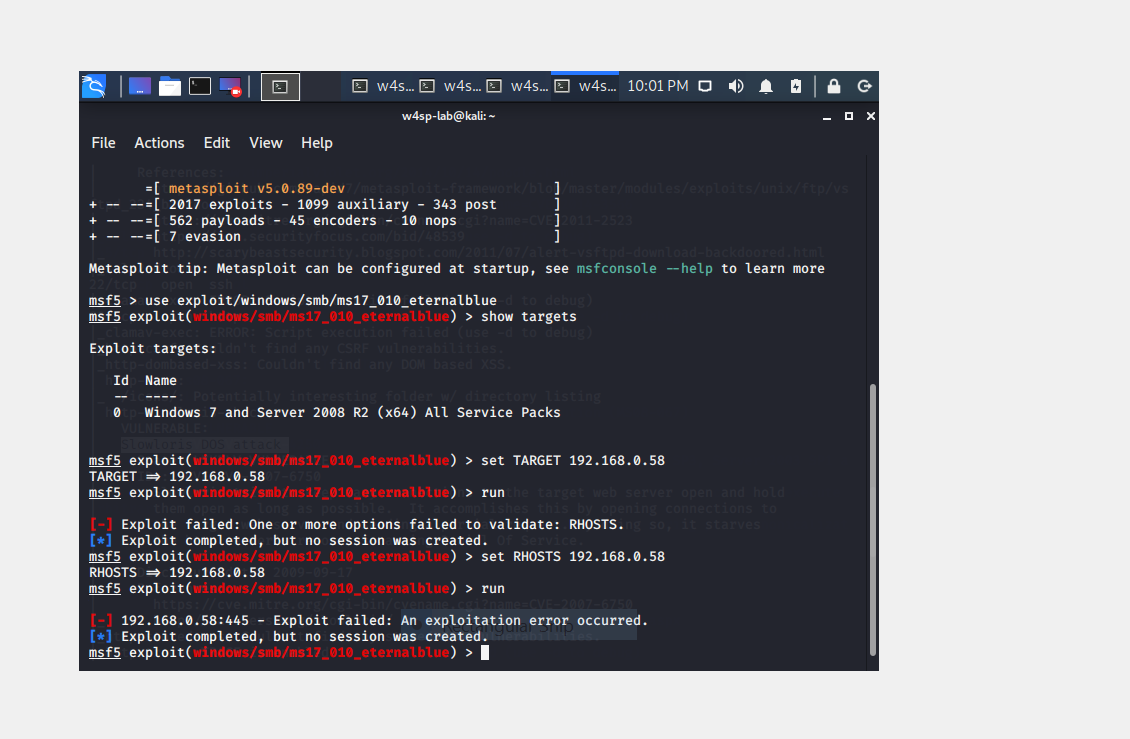
**Nmap Scan Results:**



Initial Shell Vulnerability Exploited: Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)

Additional info about where the initial shell was acquired from: This vulnerability was acquired from Windows XP machine by using Nmap and then I used Metasploit to gain access into the system.

**Vulnerability Explanation:** This vulnerability exists in Windows XP due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code.

**Proof of Exploit:** 

**Vulnerability Fix:** In order to fix this, you have to download the security update(a set of patches for Windows) on Microsoft Windows’s website. See <https://support.microsoft.com/en-us/help/4013389/title>

**Severity: Critical**

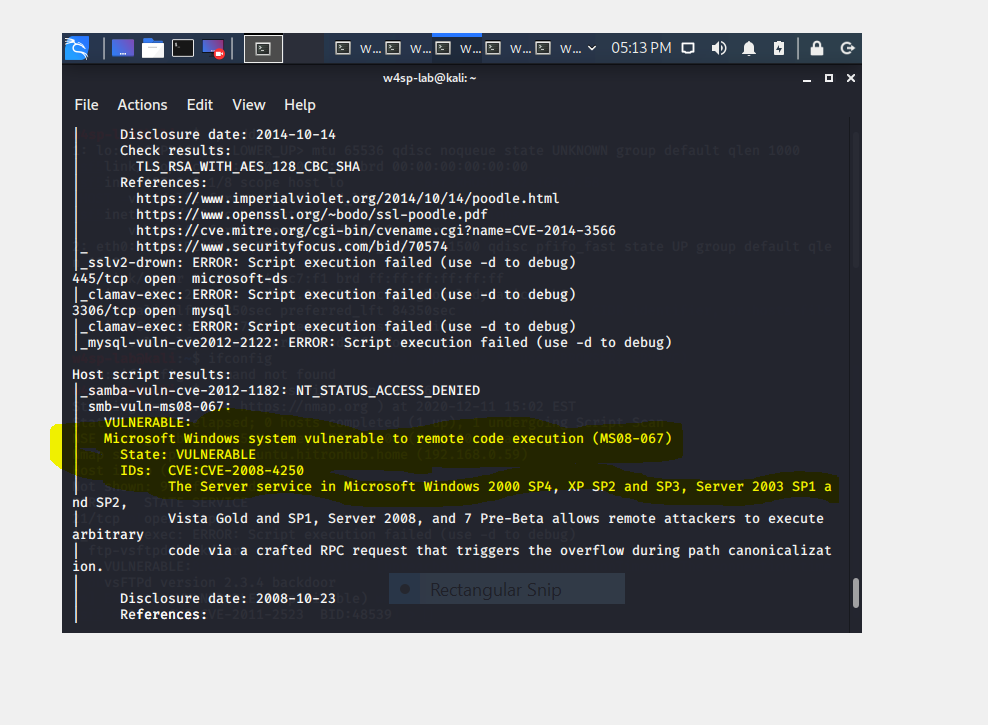
**Privilege Escalation: I was not able to gain access on Windows XP through this vulnerability.**

**System IP: 192.168.0.58**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| **192.168.0.58** | **TCP: 21,25,80,135,139,443,445,3306** |
| **UDP:** | **123,137** |

**Vulnerability Exploited: MS08-067 Microsoft Windows Server service crafted RPC Request Handling Remote code execution**

**Nmap Scan:**

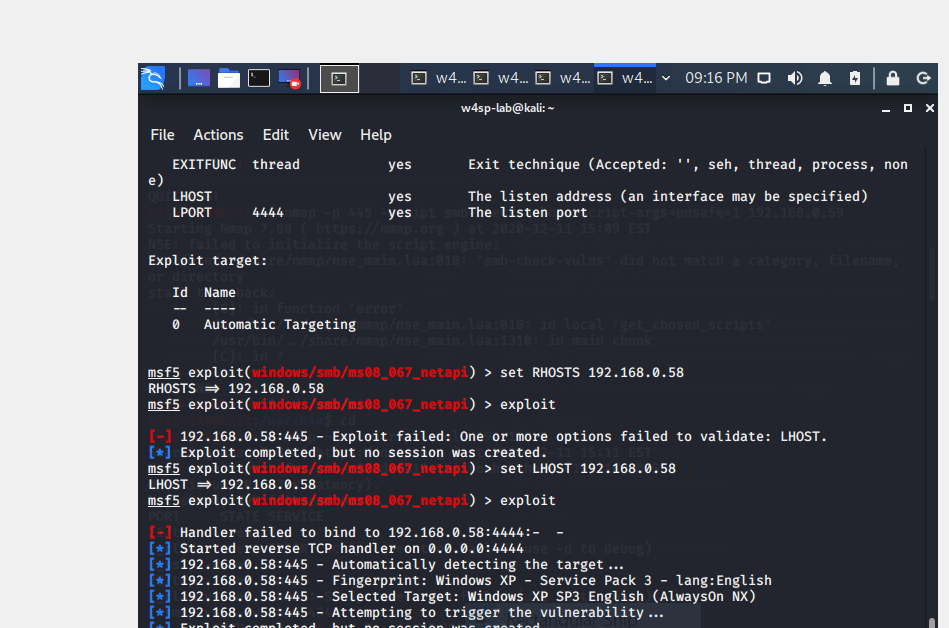


**Vulnerability Explanation:** The vulnerability could allow remote code execution if an affected system received a specially crafted RPC request. On Windows XP, an attacker could exploit this vulnerability without authentication to run arbitrary code.

**Vulnerability Fix:** Firewall best practices and standard default firewall configurations can help protect network resources from this kind of attack. You can also download the latest patch for the ms08-067 on Microsoft Window XP website. <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2008/ms08-067>

**Severity:** Critical

**Exploit Proof Screenshot Here:**

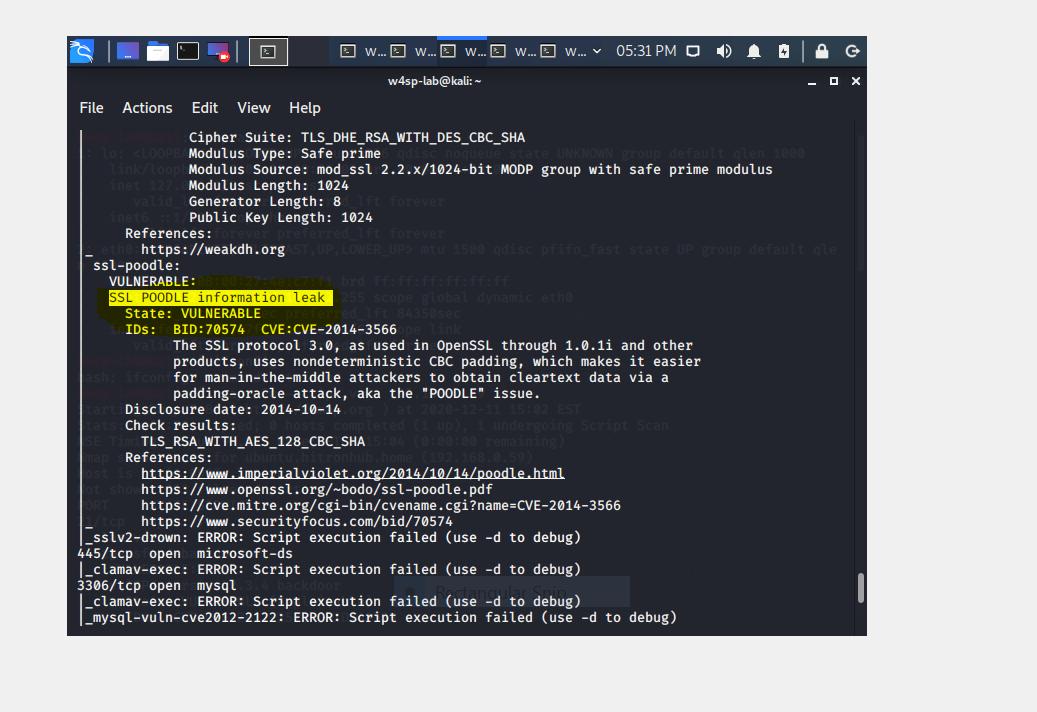


**Privilege Escalation: I was able to gain a remote access into windows XP.**

**System IP: 192.168.0.58**

**Service Enumeration:**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| **192.168.0.58** | **TCP: 21,25,80,135,139,443,445,3306** |
| **UDP:** | **123,137** |

**Nmap Scan Results:** 

Initial Shell Vulnerability Exploited : SSL POODLE Information Leak

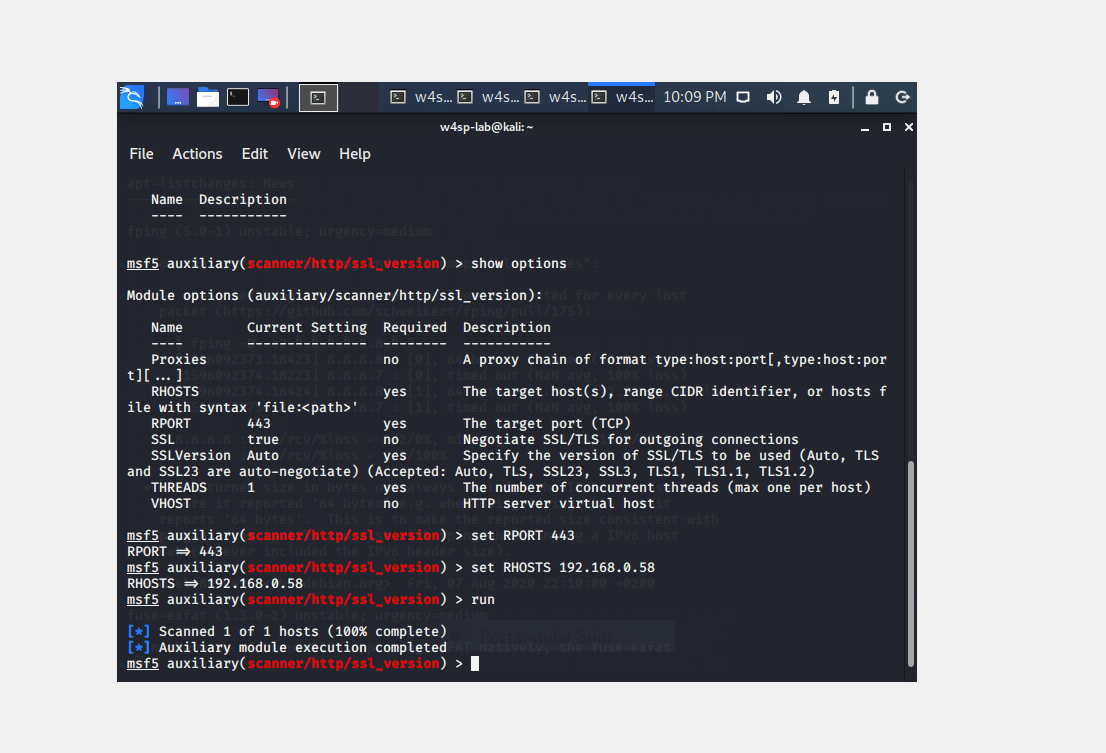
Additional info about where the initial shell was acquired from: This vulnerability was required from Nmap

**Vulnerability Explanation:** This flaw is found in Win XP, it uses nondeterministic CBC padding, which makes it easier for man in the middle attackers to obtain clear text data via a padding-oracle attack, aka the “POODLE” issue.

**Vulnerability Fix:** There is currently no fix for the vulnerability SSL 3.0 itself, as the issue is fundamental to the protocol; however, disabling SSL 3.0 support in system/application configurations is the most viable solution currently available. See <https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-poodle-sslv3-vulnerability#:~:text=On%20October%2014th%2C%202014%2C%20a,in%2Dthe%2Dmiddle%20attack>.

**Severity:** Critical

**Proof of exploit:**



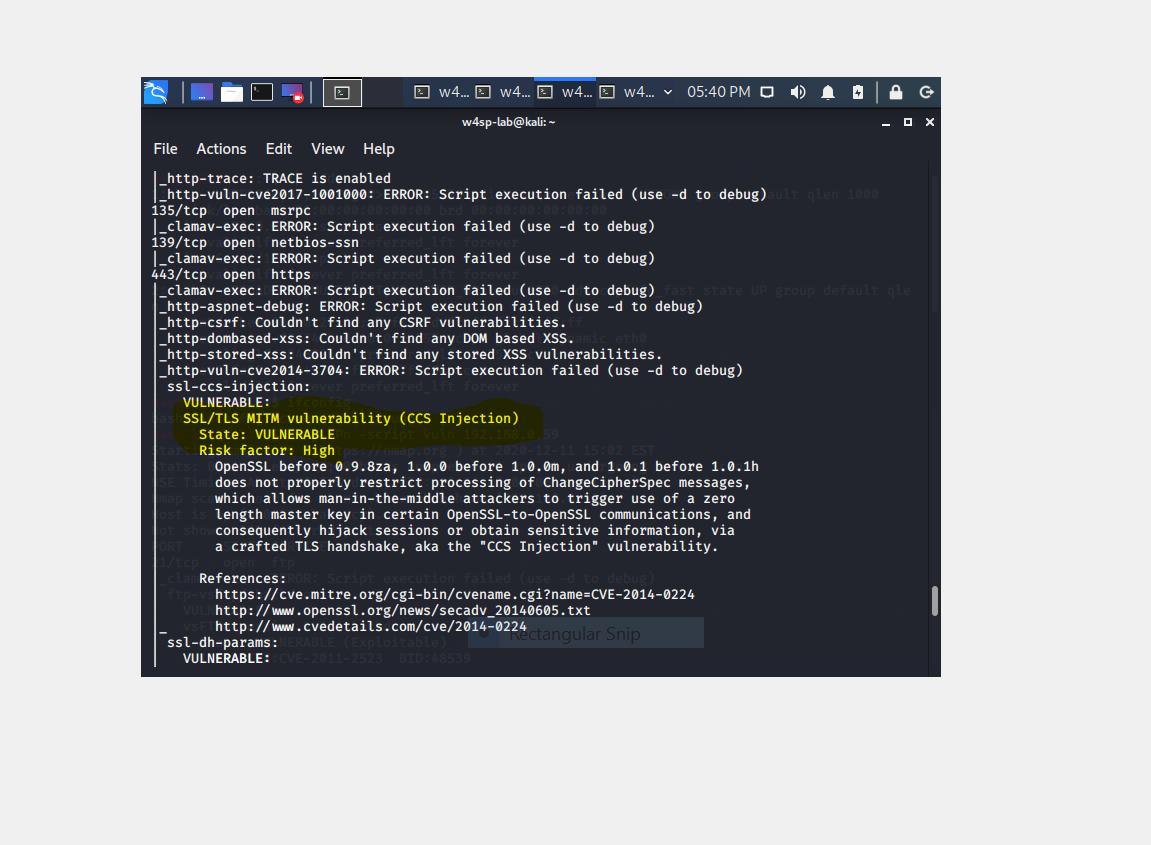
**Privilege Escalation: Even though, this is not an exploit, one can be able to attack Window XP through this vulnerability.**

**System IP: 192.168.0.58**

**Service Enumeration**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| **192.168.0.58** | **TCP: 21,25,80,135,139,443,445,3306** |
| **UDP:** | **123,137** |

**Nmap Scan Results:**



**Initial Shell Vulnerability Exploited:** SSL/TLS MITM vulnerability (CCS Injection)

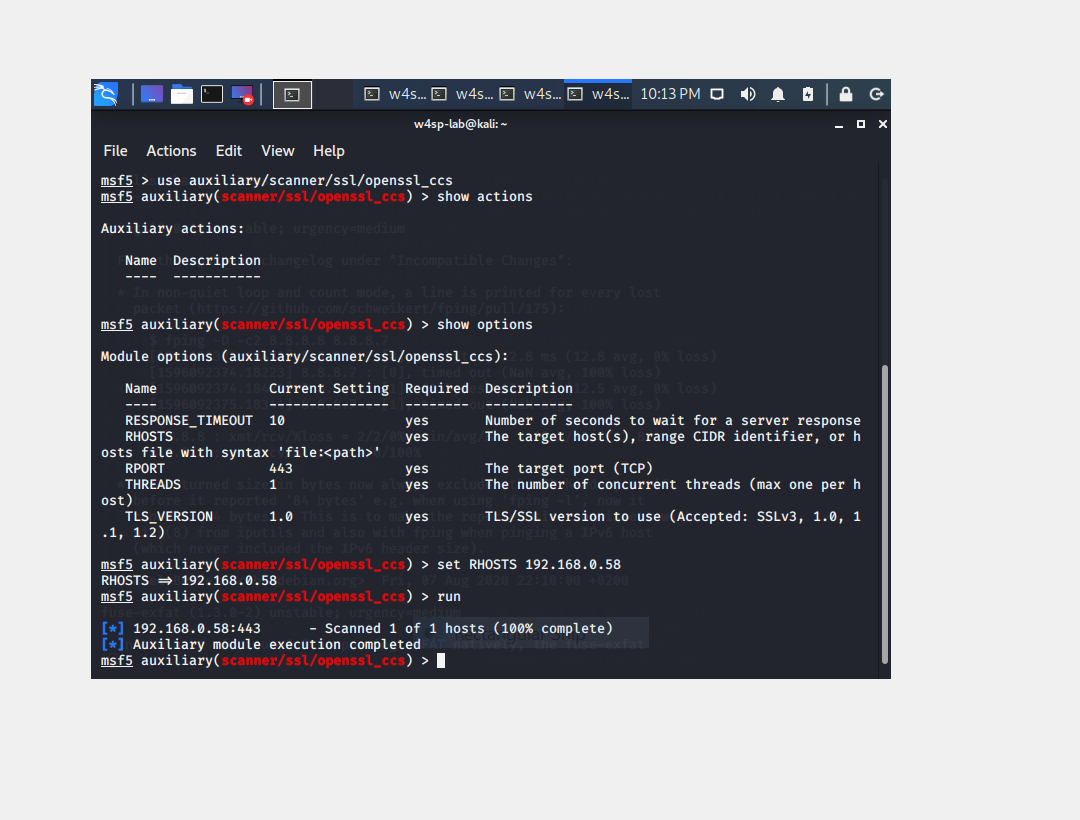
**Additional info about where the initial shell was acquired from:** This vulnerability was acquired by using Nmap.

**Vulnerability Explanation:** CertainOpenSSL does not properly restrict processing of Change Cipher Spec messages which allows man-in-the-middle attackers to trigger use of a zero length master key in certain OpenSSL-to-OpenSSL communications, and consequently hijack sessions or obtain sensitive information, via crafted TLS handshake, aka the "CCS Injection" vulnerability. Attackers can predict temporal encryption key materials of any communication by sending invalid signals in the handshake sessions. If attackers grasp the key materials, they can eavesdrop the encrypted communication or steal your identity.

**Vulnerability Fix:** You can apply software updates from each software vendors. Also, if you are using Windows, Mac or iPhone, there are no risks regarding to this vulnerability. <https://www.rapid7.com/db/vulnerabilities/http-openssl-cve-2014-0224/>

**Severity:** High

**Proof of exploit:**



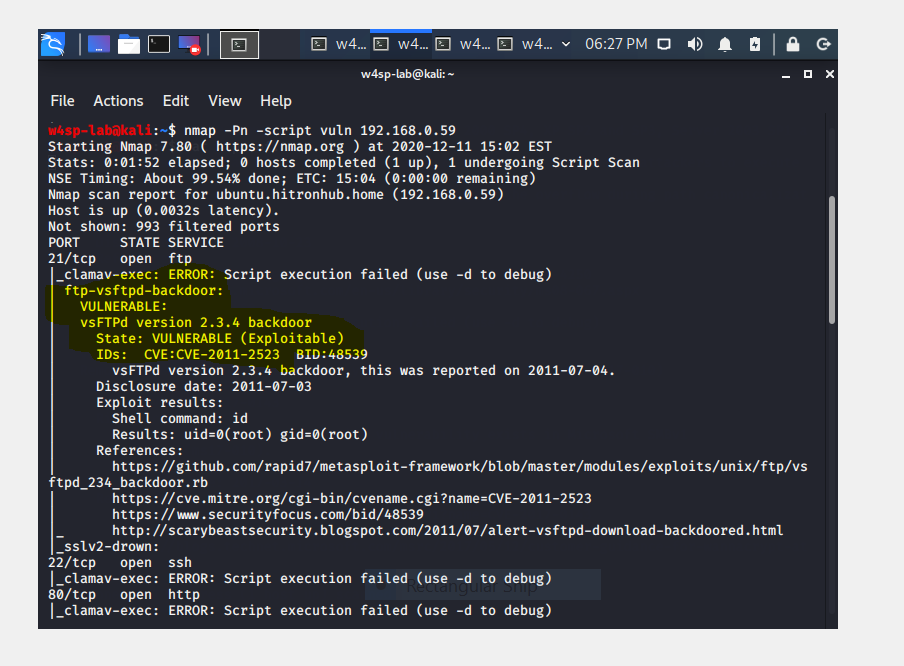
**Privilege Escalation:** Even Though this is not an exploit**,** this auxiliary module allows you to pass credentials in a number of ways.

**System IP: 192.168.0.59**

**Service Enumeration**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| **192.168.0.58** | **TCP: 21,22,80,111,139,445,2049** |
| **UDP:** | **111,137,2049,5353** |

**Nmap Scan Results:**



**Initial Shell Vulnerability Exploited: ftp-vsFTPd-backdoor**

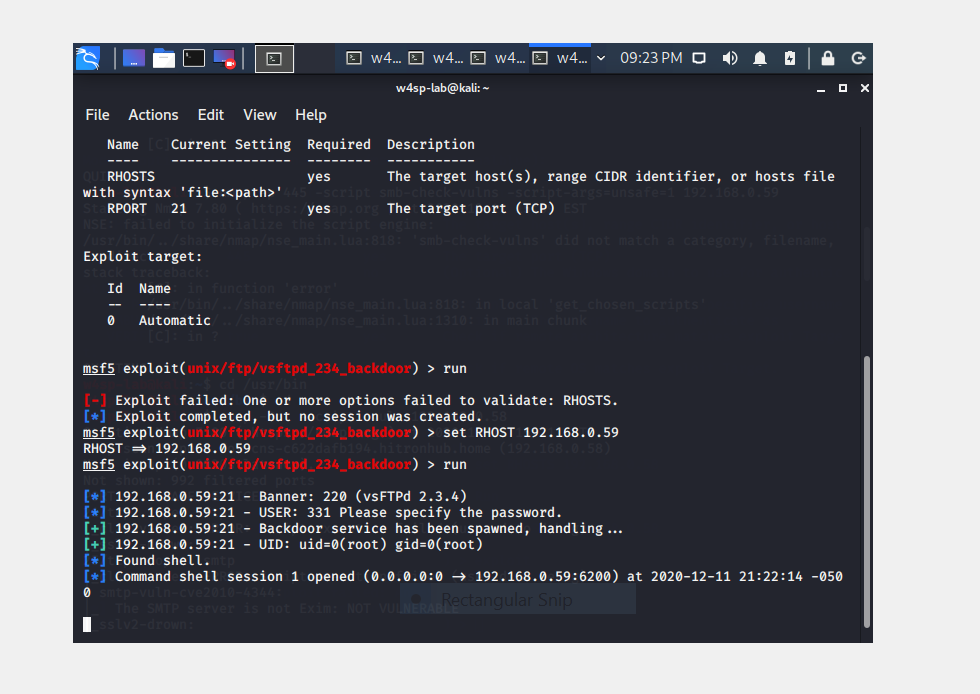
**Additional info about where the initial shell was acquired from: This vulnerability was acquired from Linux virtual machine by using Nmap.**

**Vulnerability Explanation:** vsFTPd is prone to a backdoor vulnerability. Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected application.

**Vulnerability Fix:** The repaired package can be downloaded from

[**https://security.appspot.com/vsftpd.html**](https://security.appspot.com/vsftpd.html)**.**

**Severity:** Critical

**Proof of exploitation:** 

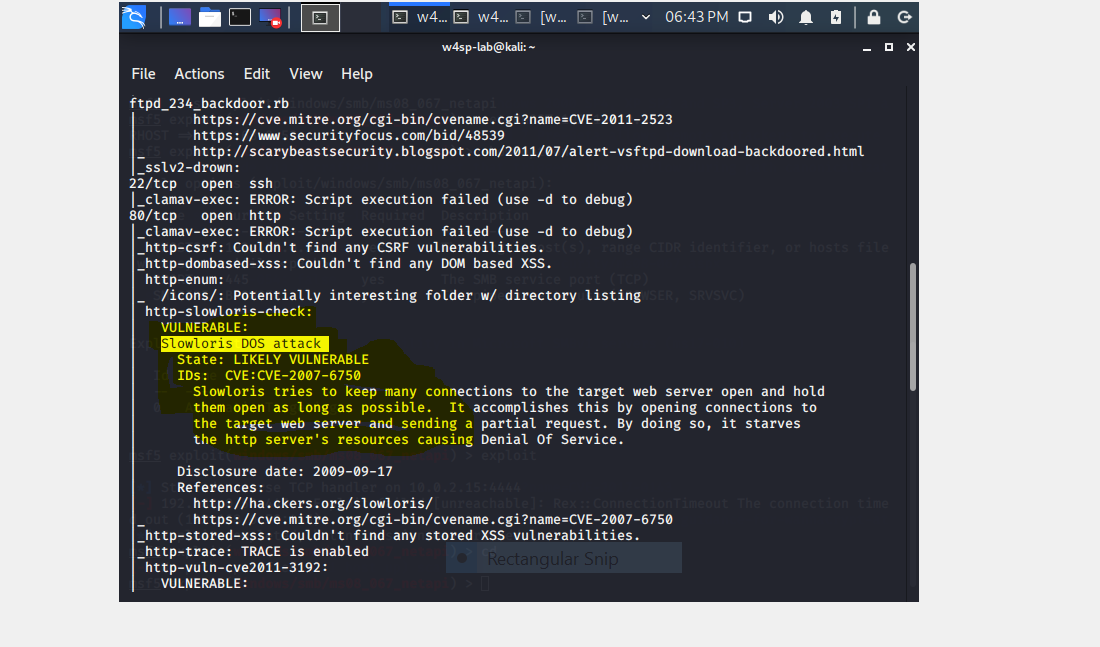
**Privilege Escalation: I was able to gain access to this Linux Machine.**

**System IP: 192.168.0.59**

**Service Enumeration**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| **192.168.0.58** | **TCP: 21,22,80,111,139,445,2049** |
| **UDP:** | **111,137,2049,5353** |

**Nmap Scan Results:**



**Initial Shell Vulnerability Exploited:** Slowloris DOS attack

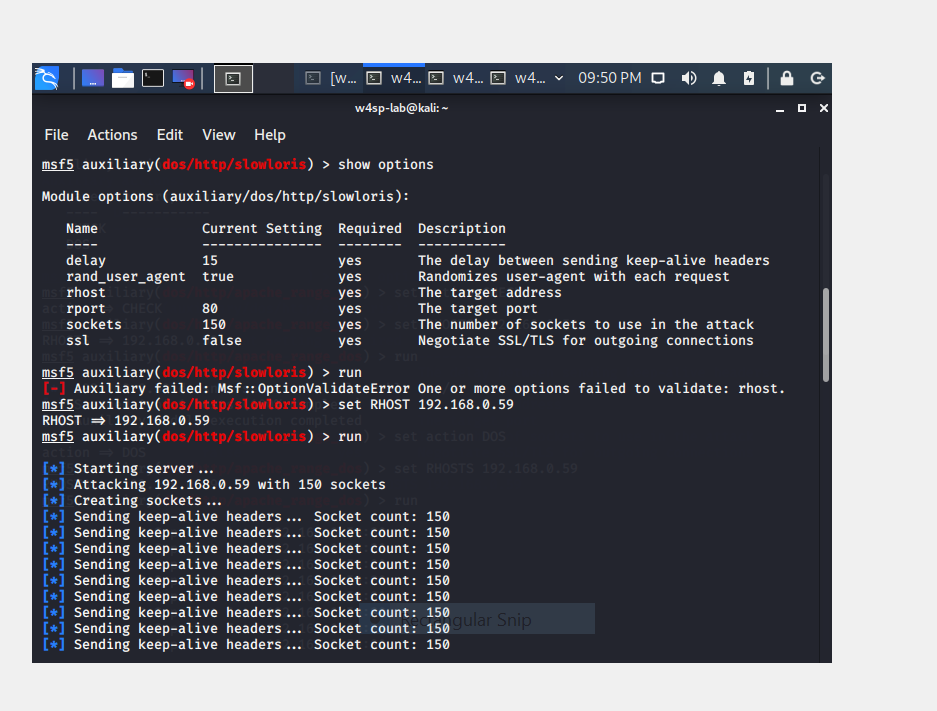
**Additional info about where the initial shell was acquired from:** This vulnerability was acquired from Linux machine by using Nmap.

**Vulnerability Explanation:** Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial of Service.

**Vulnerability Fix:** Update the system**,** updates are available at <https://blogs.oracle.com/sunsecurity/cve-2007-6750-resource-management-errors-vulnerability-in-apache>

**Severity:** Medium

**Proof of exploit:**



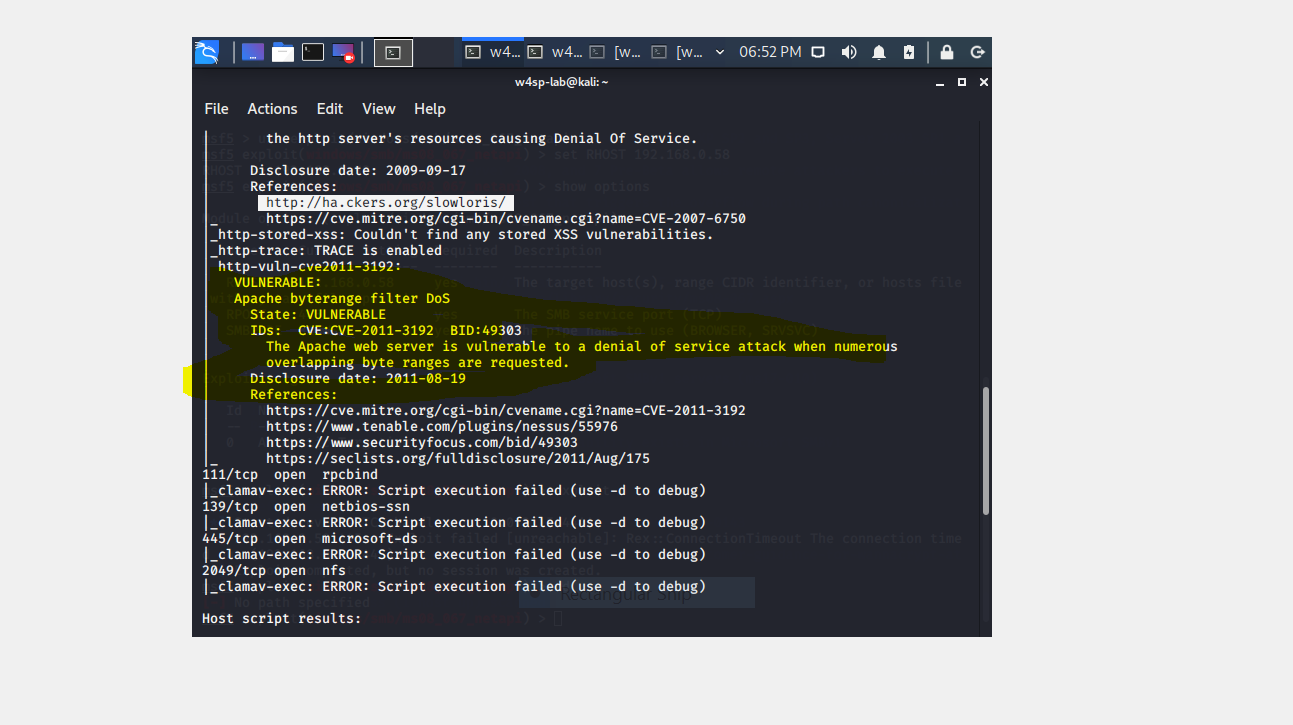
**Privilege Escalation:** Was able to attack this machine through Slowloris DOS attack vulnerability**.**

**System IP: 192.168.0.59**

**Service Enumeration**

| **Server IP Address** | **Ports Open** |
| --- | --- |
| **192.168.0.58** | **TCP: 21,22,80,111,139,445,2049** |
| **UDP:** | **111,137,2049,5353** |

**Nmap Scan Results:**



**Initial Shell Vulnerability Exploited:** Apache byte range filter DoS

**Additional info about where the initial shell was acquired from:** This vulnerability was acquired from Linux machine by using Nmap.

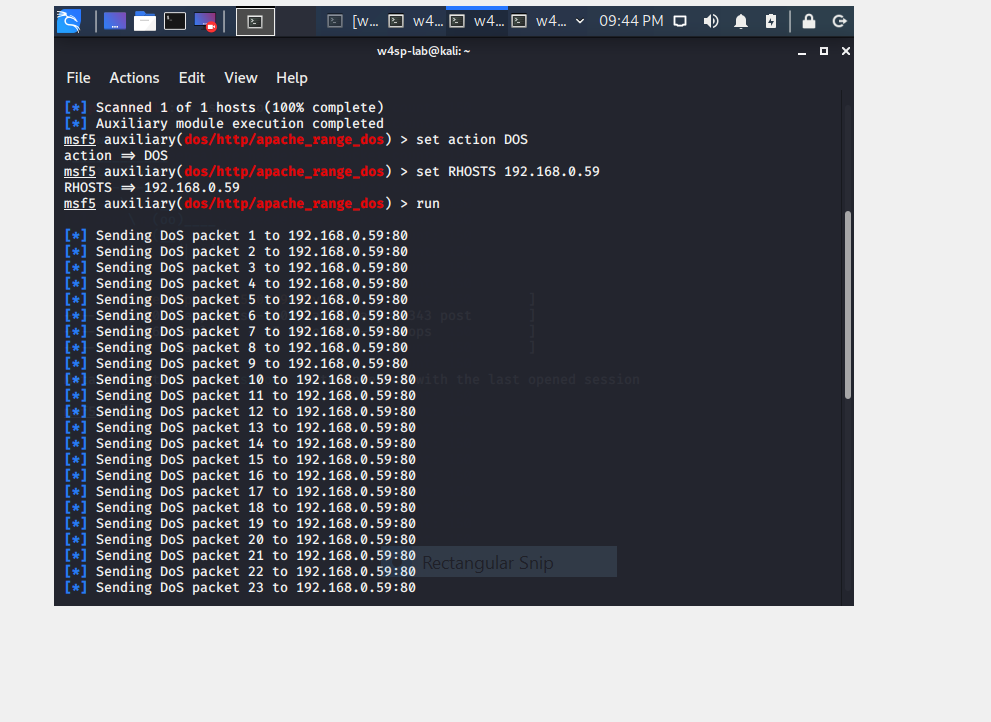
**Vulnerability Explanation:** There exists a denial of service vulnerability in the Apache HTTP Server byte-range filter. The vulnerability occurs during the handling of requests that include a byte-range value, when HTTP Server is configured to act as a proxy. This flaw can cause the excess consumption of memory resources. A remote attacker can exploit the vulnerability by sending specially crafted HTTP requests to the target server, potentially causing denial of service. A successful attack leveraging this vulnerability can cause Apache httpd to allocate significant amounts of memory. If a malicious user keeps sending the crafted HTTP request to the target host, a denial-of-service occurs.

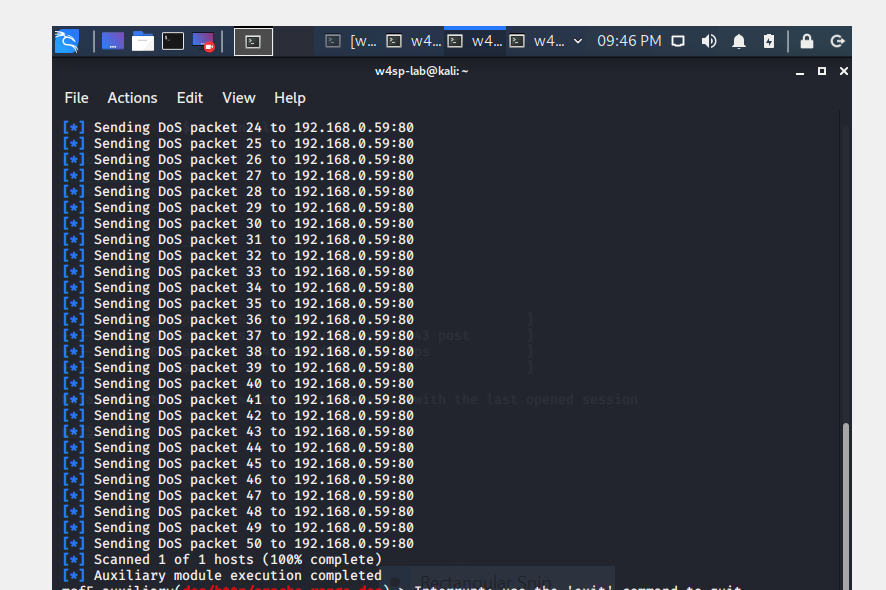
**Vulnerability Fix:** Upgrade your Apache server to a currently stable version. Please see the following reference for more information. **Slackware Linux 12.2**

* **Slackware httpd-2.2.20-i486-1\_slack12.2.tgz**[**ftp://ftp.slackware.com/pub/slackware/slackware-12.2/patches/packages/ httpd-2.2.20-i486-1\_slack12.2.tgz**](ftp://ftp.slackware.com/pub/slackware/slackware-12.2/patches/packages/httpd-2.2.20-i486-1_slack12.2.tgz)

**Severity:** Medium

**Proof of exploit:**





**Privilege Escalation:** Gained access to resources

**Maintaining Access**

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

**House Cleaning**

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous, and no remnants of our penetration test are left over is important.

After collecting trophies from the machines’ network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Ishimwe Pioneer organization should not have to remove any user accounts or services from the system.

**Additional Items**

For the exam, I used my Metasploit/Meterpreter allowance on the following machine: 192.168.0.58 and 192.168.0.59