Penetration Testing

Final Report

Mandeep Parihar



Confidential

Table of Contents

[Introduction 3](#_Toc90035192)

[Personnel Involvement 3](#_Toc90035193)

[Contact Information 3](#_Toc90035194)

[Assets Involved 3](#_Toc90035195)

[Objectives of Test 3](#_Toc90035196)

[Scope of Test 3](#_Toc90035197)

[Information Gathering 3](#_Toc90035198)

[Passive Intelligence 3](#_Toc90035199)

[Active Intelligence 4](#_Toc90035200)

[Vulnerability Assessment 4](#_Toc90035201)

[Technical Vulnerabilities 4](#_Toc90035202)

[Scanner Found 4](#_Toc90035203)

[Manually Identified 5](#_Toc90035204)

[Vulnerability Confirmation 6](#_Toc90035205)

[Exploitation Timeline 6](#_Toc90035206)

[Exploitation Activities 6](#_Toc90035207)

[Individual Host Information 6](#_Toc90035208)

[Attacks Conducted 9](#_Toc90035209)

[Remediation 20](#_Toc90035210)

[Post-Exploitation 21](#_Toc90035211)

[Privilege Escalation Path 21](#_Toc90035212)

[Value of Information 21](#_Toc90035213)

[Ability of Persistence 21](#_Toc90035214)

[Countermeasure Effectiveness 22](#_Toc90035215)

[Conclusion 22](#_Toc90035216)

# Introduction

## Personnel Involvement

As directly requested, I was the only active member of the Vanguard Solutions penetration testing team that conducted all tests for this project. No other individuals were including during the process.

## Contact Information

Any further communications with Vanguard Solutions can be done via the following email address: [Vanguard@Solutions.com](mailto:Vanguard@Solutions.com). Any communication with myself can be done via the following email address: [Mandeep.Parihar@edu.sait.ca](mailto:Mandeep.Parihar@edu.sait.ca).

## Assets Involved

VirtualBox was used to create a controlled area for all testing. The boxes used were: Kali, as the sole attacking machine, Windows 7, provided to us on your behalf as the first victim, and a premade Metasploitable2 machine loaded with Ubuntu as the second victim. Inside of the Kali, Nessus, Nmap, and Metasploit were used for information discovery, vulnerability scans, and launching exploits against existing machine vulnerabilities. Both victim machines were left as they came, no changes were required.

## Objectives of Test

Th intent was to discover and generate Nessus reports based on scans of the victim machines, and to exploit one of the high or critical level vulnerability found via Metasploit. Both machines would be exploited, showing that they both have one or more serious vulnerabilities. All of this is done to create this report.

## Scope of Test

The scope was to find and successfully exploit a small number of the found vulnerabilities on the victim machines.

# Information Gathering

## Passive Intelligence

No passive intelligence gathering was required. The machines were provided, as well as their login information, IP addresses, and any internal information that was required could be retrieved from inside of the VMs themselves.

## Active Intelligence

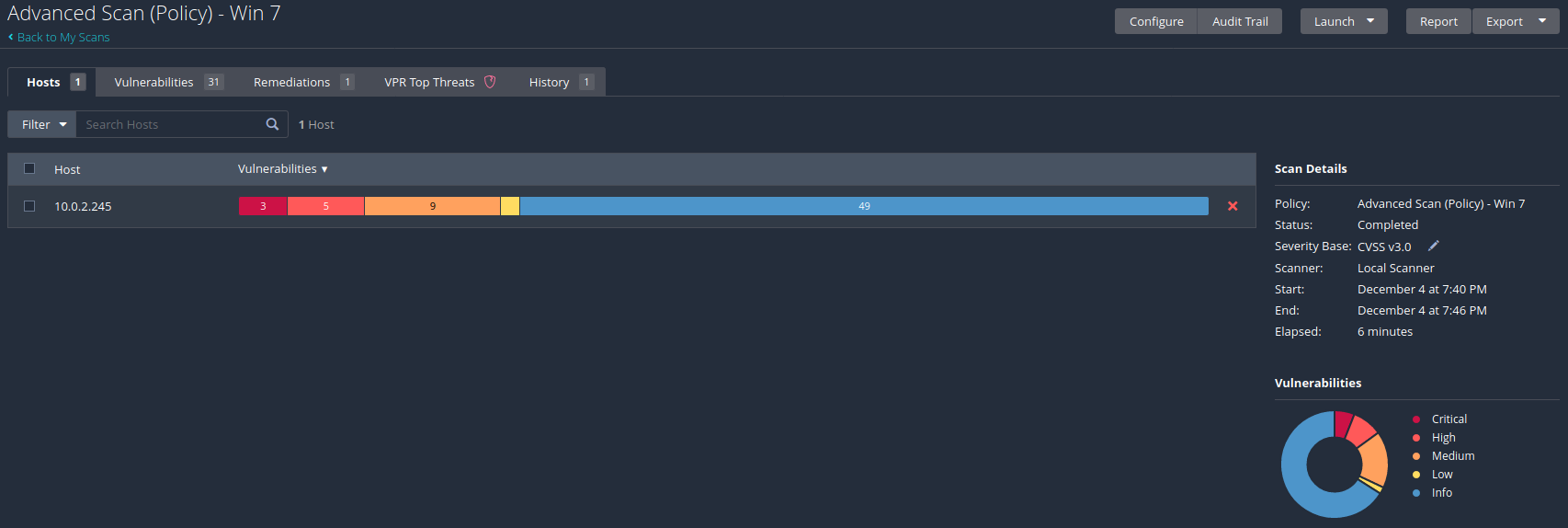
The tests were run under the impression that the attacking machine was in the same subnet as the victim machines, and their IP addresses were known. Machine scans were done using Nessus with the provided IPs. All scans were configured to look for different areas of the machine, such as the status of all ports, so that a full list of vulnerabilities would be on-hand, and accessible through generated reports.

# Vulnerability Assessment

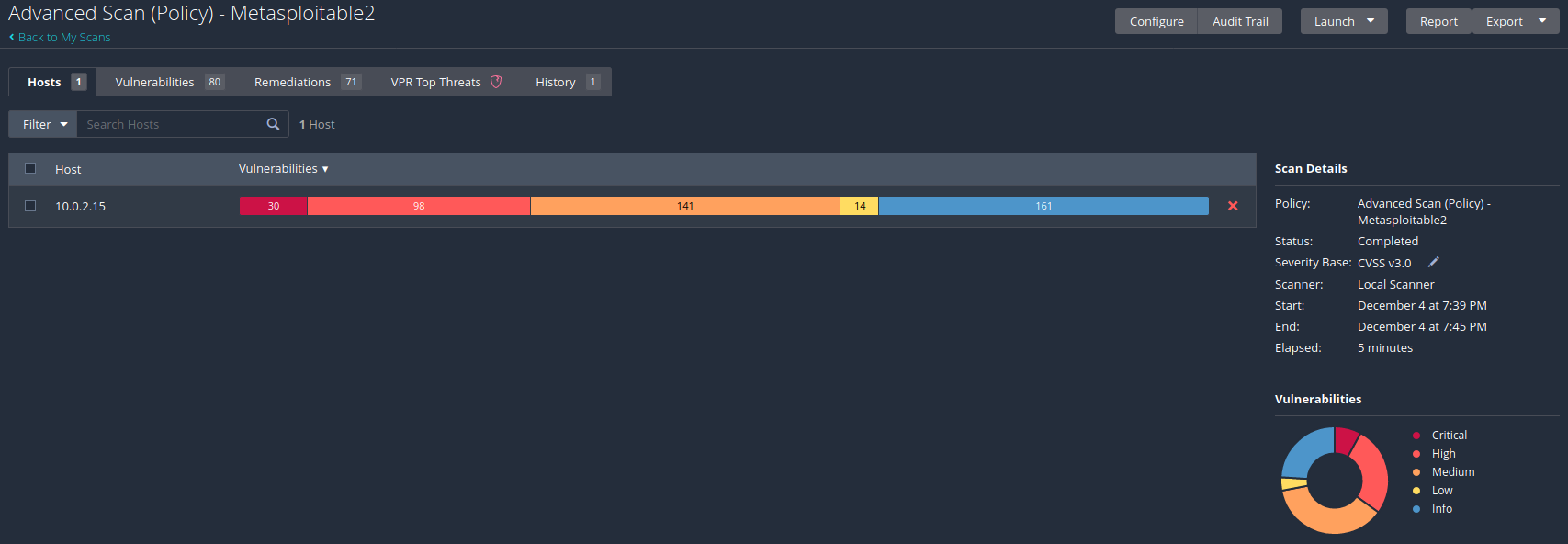
## Technical Vulnerabilities

### Scanner Found

Nessus scans found a noticeably different number of vulnerabilities between both machines. Using an advanced scan on the Windows 7 machine returned 3 critical vulnerabilities and 5 high vulnerabilities:



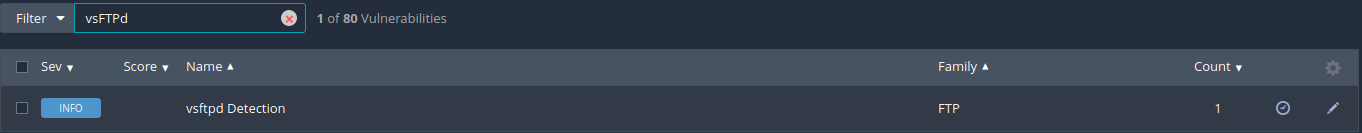
The Metasploitable2 machine returned 30 critical vulnerabilities and 98 high vulnerabilities:



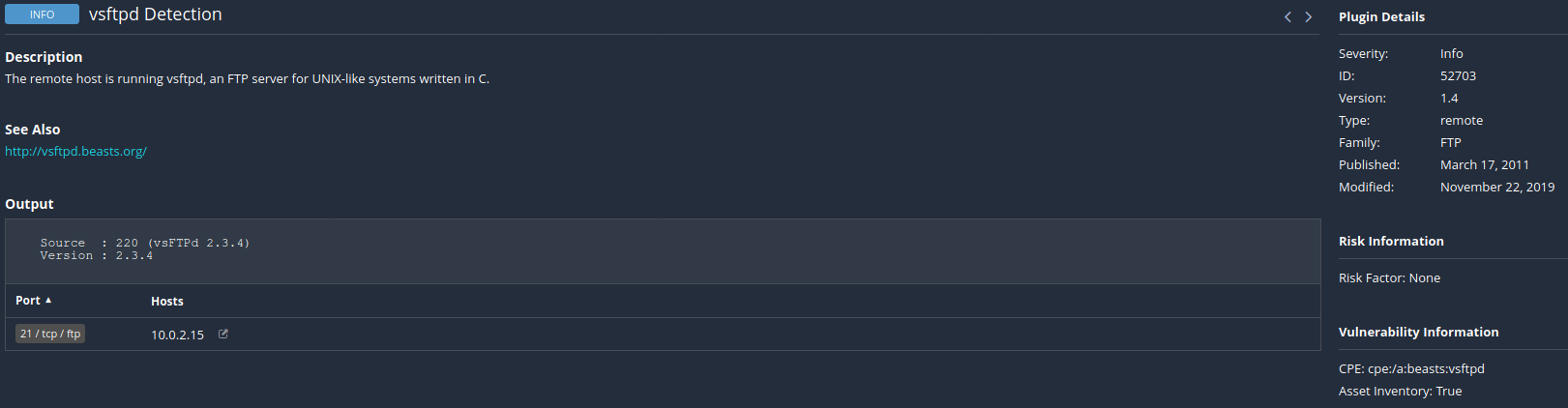
There were other scans run, but of all scans conducted these were the most thorough and yielded the highest number of vulnerabilities.

### Manually Identified

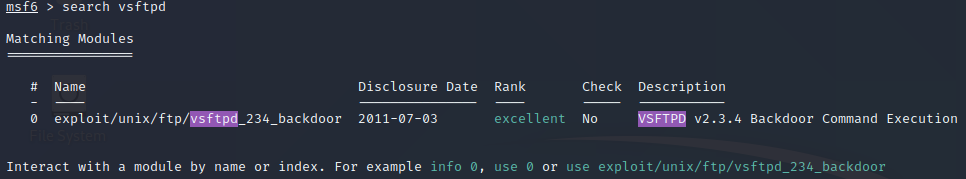
Two vulnerabilities were found manually, and they were with the Metasploitable2 machine. There is a flaw with the version of vsFTPd that is used:



Expanding the potential vulnerability shows additional information:

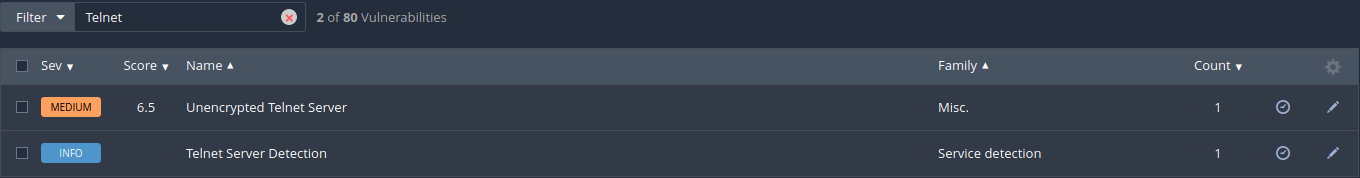


Though the exploit is not promising at a glance, there exists an available exploit in Metasploit ready for deployment:

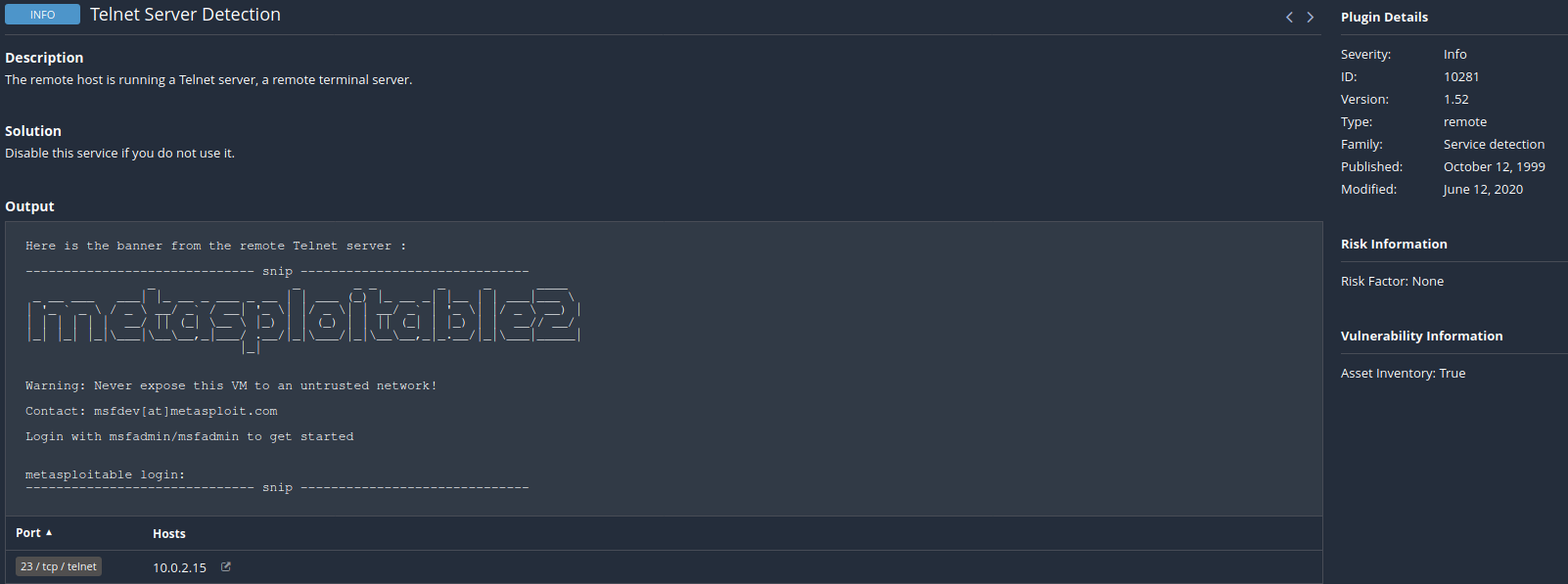


This exploit is described as a backdoor command execution by Metasploit, so it is one worth trying.

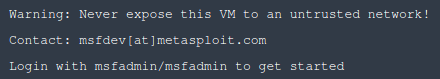
The second exploit that was manually located was with telnet:



Viewing information of this exploit gives general information:



Inside of the output, both the username and the password of the Metasploitable2 machine are visible:



This is a nearly guaranteed vulnerability, and a fatal blow to the security of the victim machine.

# Vulnerability Confirmation

## Exploitation Timeline

All scans and attacks were performed between a three-day period of December 4 – 6, 2021.

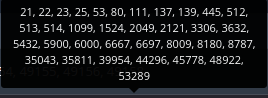
## Exploitation Activities

### Individual Host Information

All ports were scanned on both machines. The open ports found for the Windows 7 machine were plentiful:



However, the Metasploitable2 machine had many more open ports:



Given that both systems are out of date, and sporting out of date software, vulnerable to attack because of critical security updates not being present, these open ports offered avenues of attack.

Should the versions of the operating systems had not been known, they could be discovered using Nmap. Running a scan on the Windows 7 machine returned:



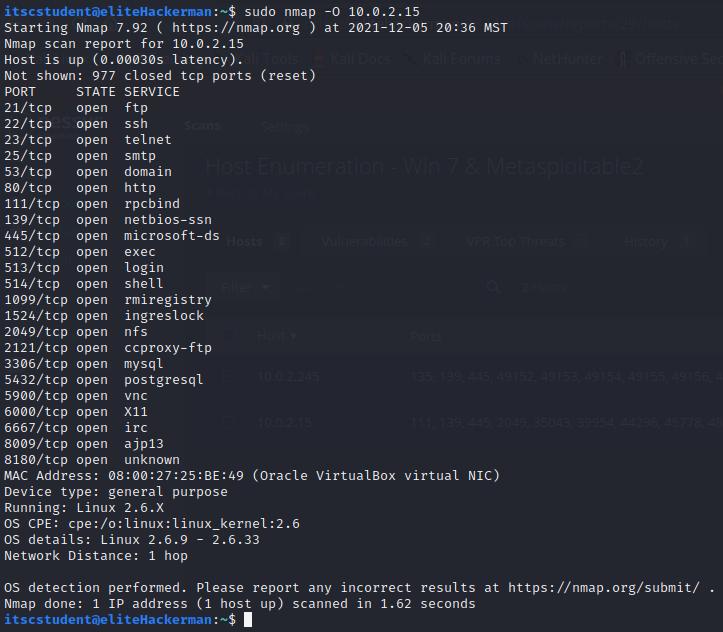
Out of this, the most import area is the OS details:



Nmap sorts the guesses by the most likely first, of which all of the first listings are Windows 7, most likely using service pack 1. Checking the machine itself, the scan is shown to be accurate:



The scan on the Ubuntu machine was not pinpoint, but still extremely close:



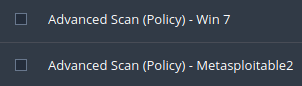
Again, the OS details are most important for this check:



Checking the machine itself, we can find the kernel version:



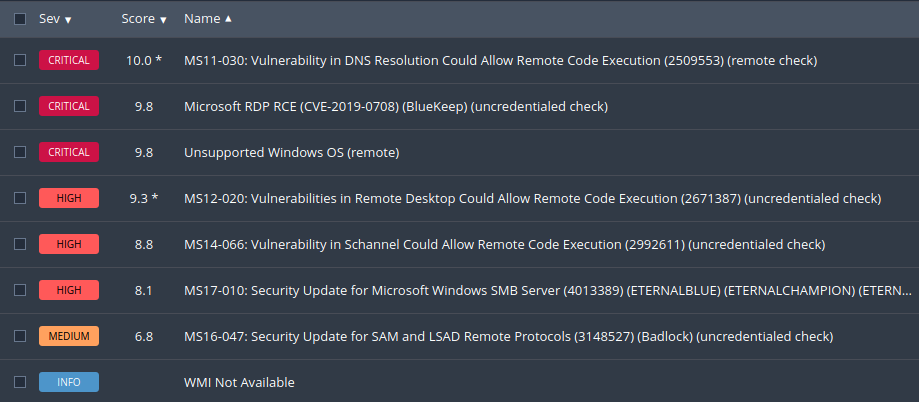
The only issues it the range of the version is sizable, being from x.x.9 – x.x.33, and the machine being listed as general-purpose machine despite it using a server version of Ubuntu. Despite that, the scan is exceptionally close, and the many existing vulnerabilities present many vectors for attack. Knowing the versions of the operating systems, and the ports that are open, an advanced scan can be done inside of Nessus with added plugins to gather a list of vulnerabilities:



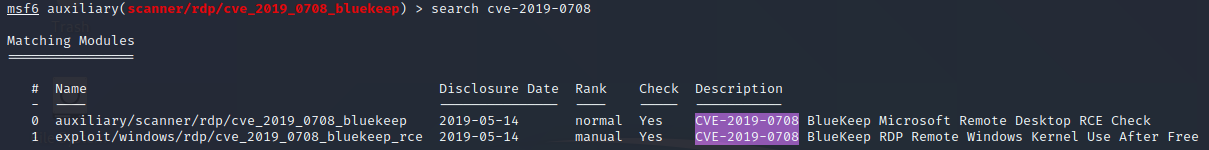
The output of these scans was bountiful – yielding the multiple critical and high-level vulnerabilities mentioned previously.

### Attacks Conducted

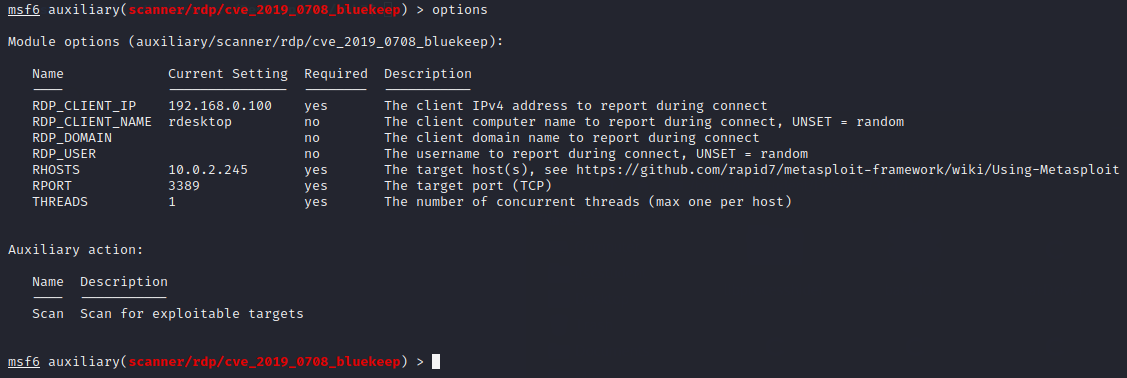
Of all of the listed vulnerabilities, the focus was kept to the ones with a critical and high severity, or seemed promising in both success and results. These would be the fastest way to get elevated privileges or access to the victim machines. Looking at the vulnerabilities listed for the Windows 7 machine, there was one attack conducted out of the list:



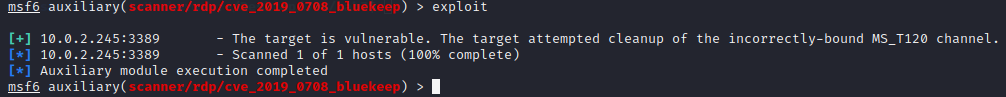
BlueKeep is a straight forward attack to setup and is designed to create a Meterpreter inside of the victim. Information can be gathered from this point, persistence can be established, or privilege escalation can occur. This exploit provides many options. Additionally, there exists two results for this exploit in Metasploit: one that checks the victim to see if it is likely to work, and the exploit itself:



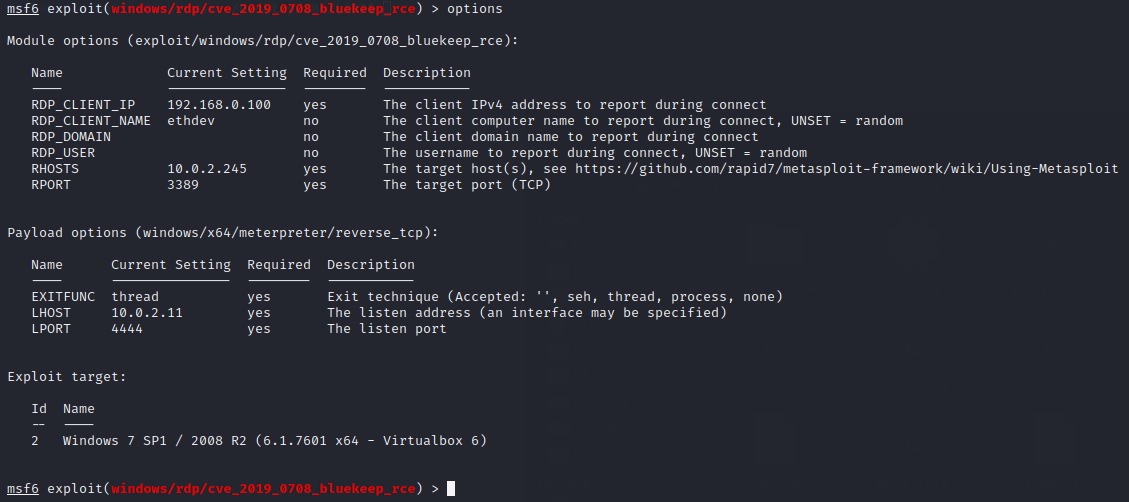
The first step is to run the auxiliary scanner and gauge success. This is done by selecting it and configuring the options to match the victim:



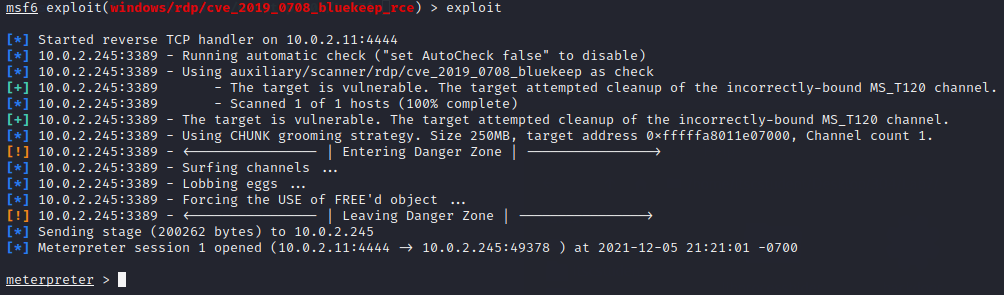
With nothing else required, the scanner can be run:



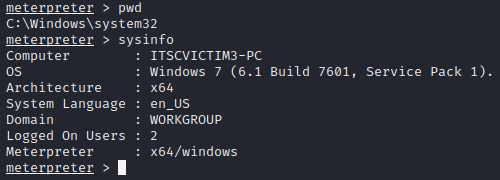
The scanner returns that the machine is vulnerable to the attack, so the actual attack is loaded up for next, and configured to match the victim machine:



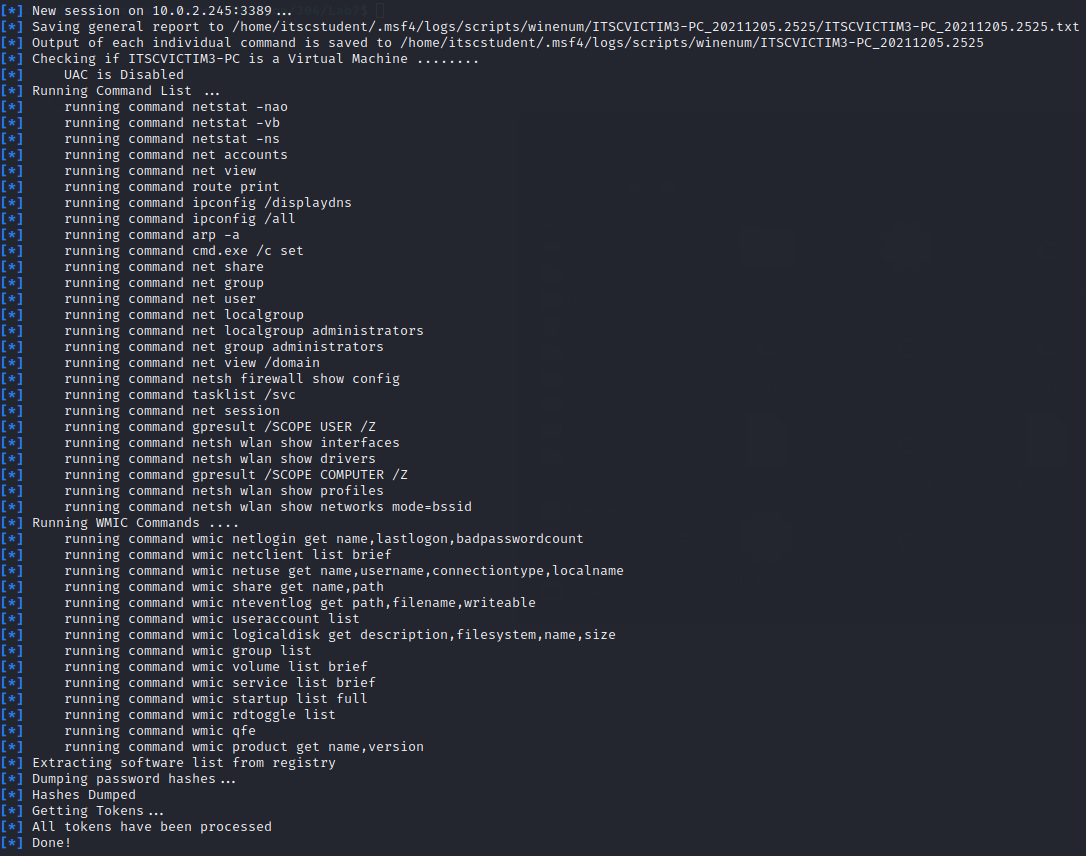
With the options set, the attack can be run:



After a short wait, the attack is successful, and a Meterpreter console is run. To verify the Linux machine has entered the victim machine, a small series of commands are run:



The machine is compromised. Furthering reach inside of the victim machine, the winenum file is run to put Meterpreter inside of a process on the victim:



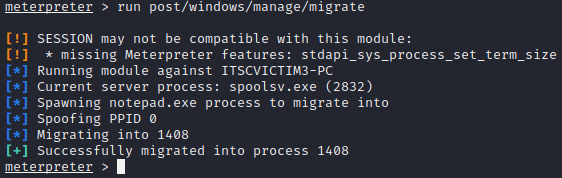
Checking the PID, we can see winenum was successful:



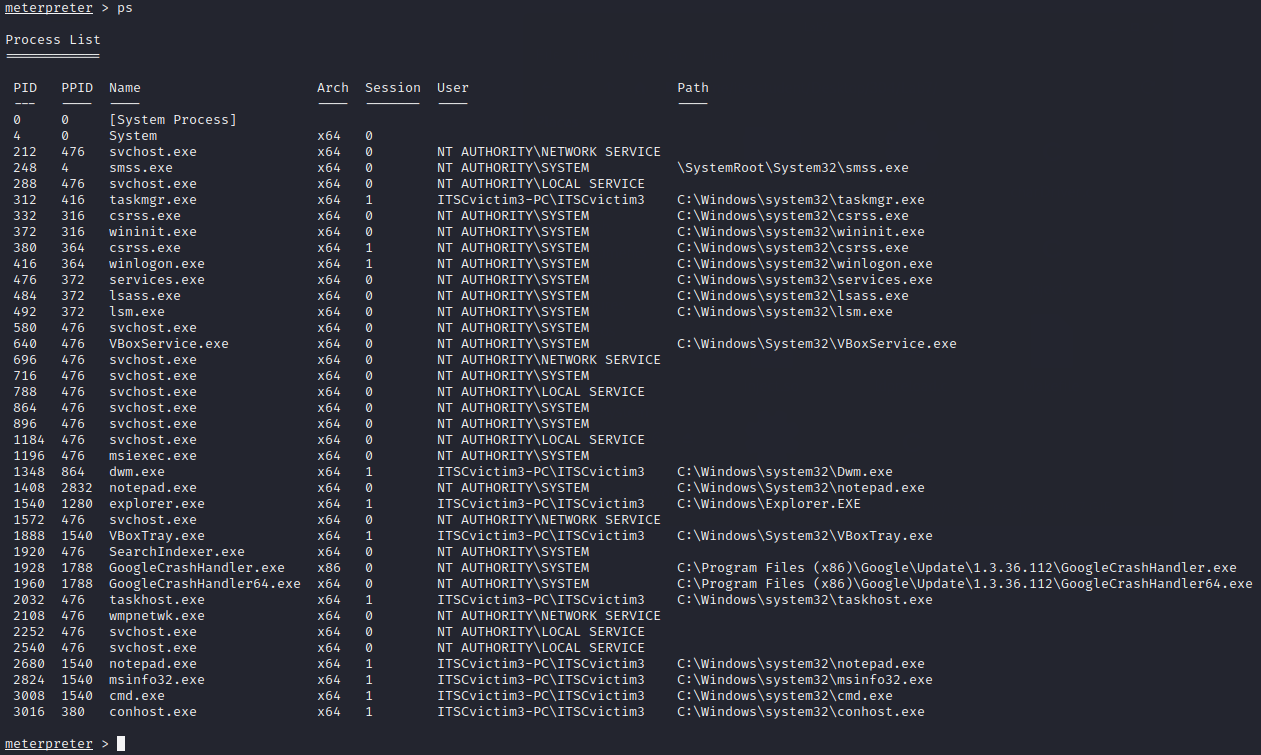
Checking the Windows 7 machine shows this process in task manager as a printer spooler:



To avoid losing the connection, the connection is migrated into something more stable:

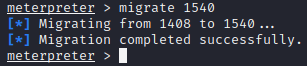


This moves the connection into notepad.exe, which is more stable; however, more stable processes exist, so another migration is to be done. To do so, a PID of a better process is required:

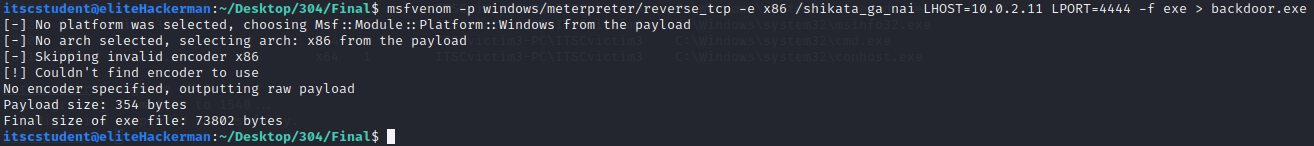


The explorer.exe process is the chosen target for this, so its PID will be used with a migration command:

The migration is successful:



The goal here is to achieve persistence, so additional work must be done. A malicious .exe file is made for establishing persistence:



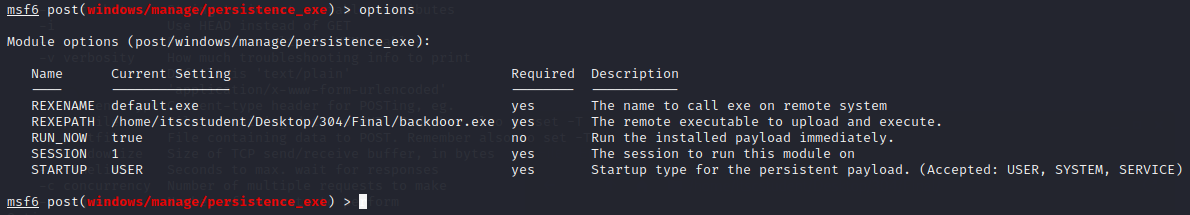
A new exploit is required, so the current Meterpreter is set into the background:



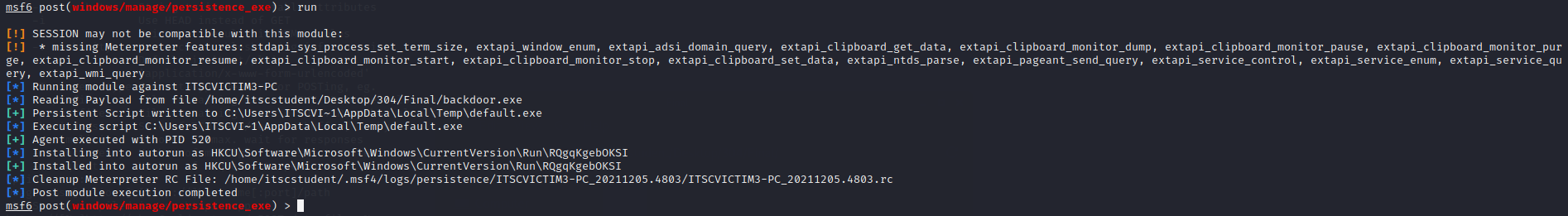
The new exploit is set:



This exploit requires configuration for its options. The backgrounded Meterpreter will be used as the session, and the malicious .exe file will be given as the REXEPATH:



With the settings configured, the attack can be launched:



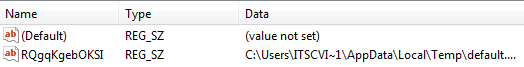
The exploit is successful. Reviewing the output, it can be seen that a persistence script is written to the victim, and executed:



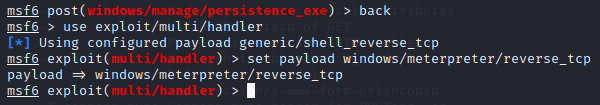
There are also persistence registry keys created in the victim system:



Checking the Windows 7 machine for these shows that they do, in fact, exist under the path listed above:



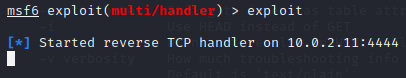
With persistence established, the Kali is to be set to a listener to confirm the persistence works on victim reboot. This requires a new payload:



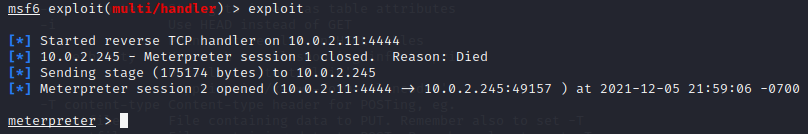
The appropriate configurations need to be made before launching the attack:



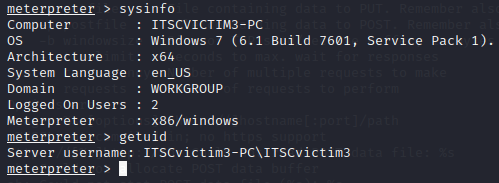
Running this will cause Metasploit to hang, waiting for the reverse TCP connection to be established:



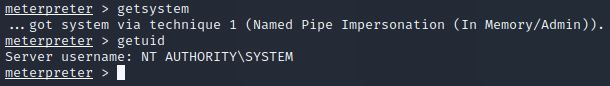
This requires that the victim machine be rebooted. When the system is rebooted, the connection can be seen being established in Metasploit’s output:



To verify the attack worked, a couple commands will be run to get system information:

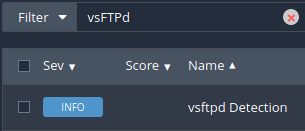


With this, persistence is established and the victim machine can be automatically reconnected to with a basic listener. Lastly, because of the user access control and firewall are disabled by default on the machine, privilege escalation is achieved with a single command:

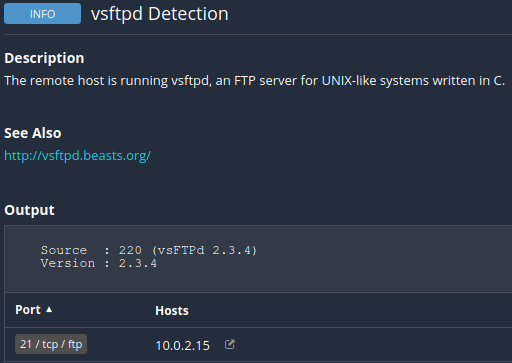


The machine has been compromised.

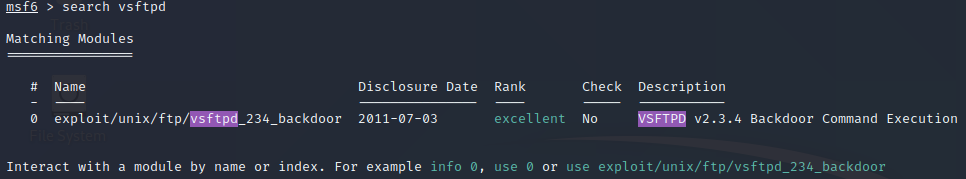
Reviewing the Metasploitable2’s results, there existed many critical and high-level vulnerabilities; however, the conducted attack was one that lacked information. The Ubuntu runs a service called vsFTPd Server, which is heavily out of date. Nessus returned a vulnerability for it:



This exploit contains a very slim description:



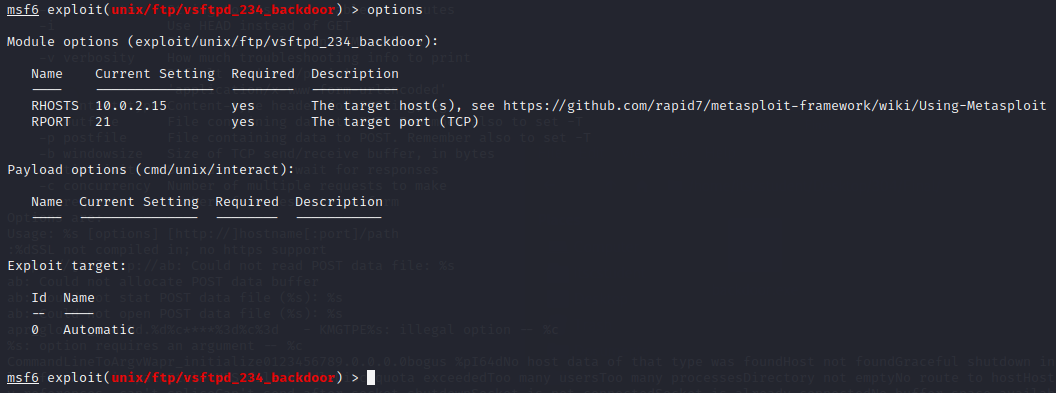
Despite the lack of details, there exists an entry for this in Metasploit:



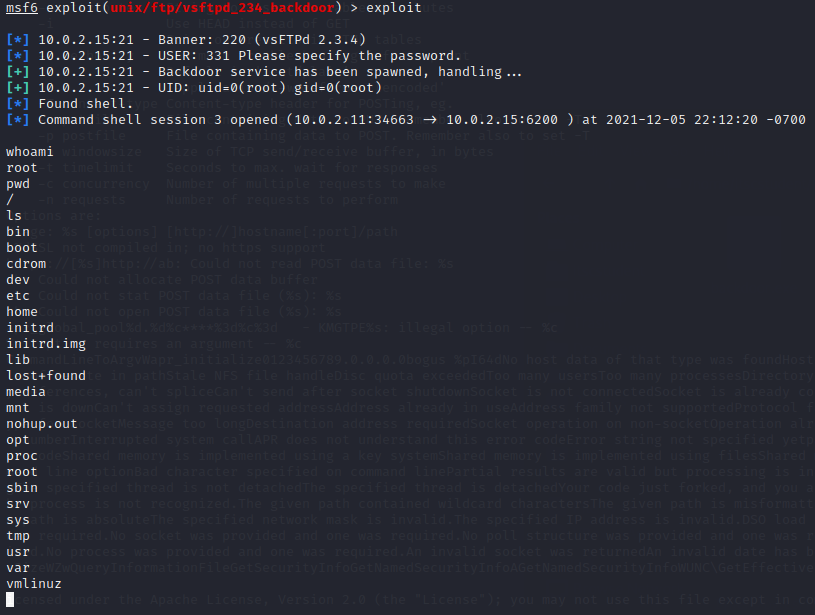
This exploit can be selected:



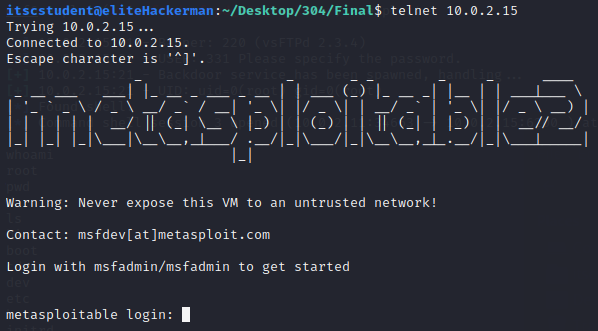
Configuration is required, but little additional information is required for this exploit:



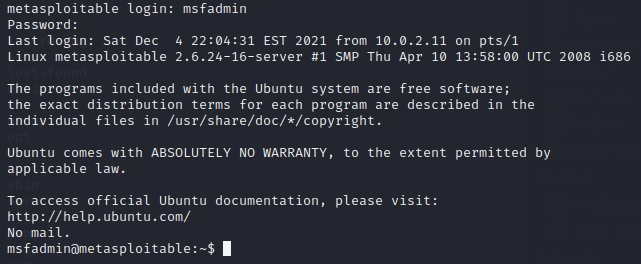
With this, the exploit can be run, and it can be seen that root access has been achieved:



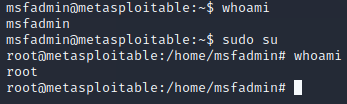
Given how straightforward the attack is, the telnet attack will be done, also achieving root access. Knowing the common telnet port is open, knowing the IP address of the system, and the username and password of the machine, a simple telnet command can be run to begin the process:



Telnet works, and a prompt is brought up for the username and password. These are known, they are “msfadmin”, so those can be entered:



Login is successful, but without root. Knowing the password of the machine, a “sudo su” can be used to get root:



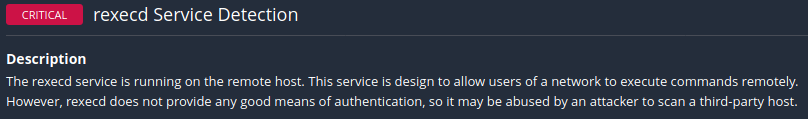
This is all achieved simply, and quickly:



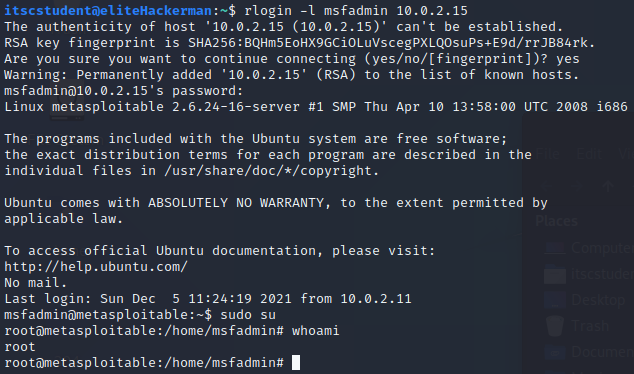
Even quicker than is the rexecd vulnerability listed in Nessus:



According to the description in Nessus, this allows for remote command execution:

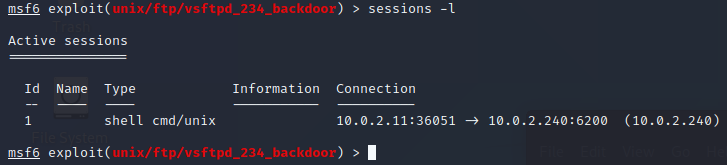


This vulnerability is abusable with the following:

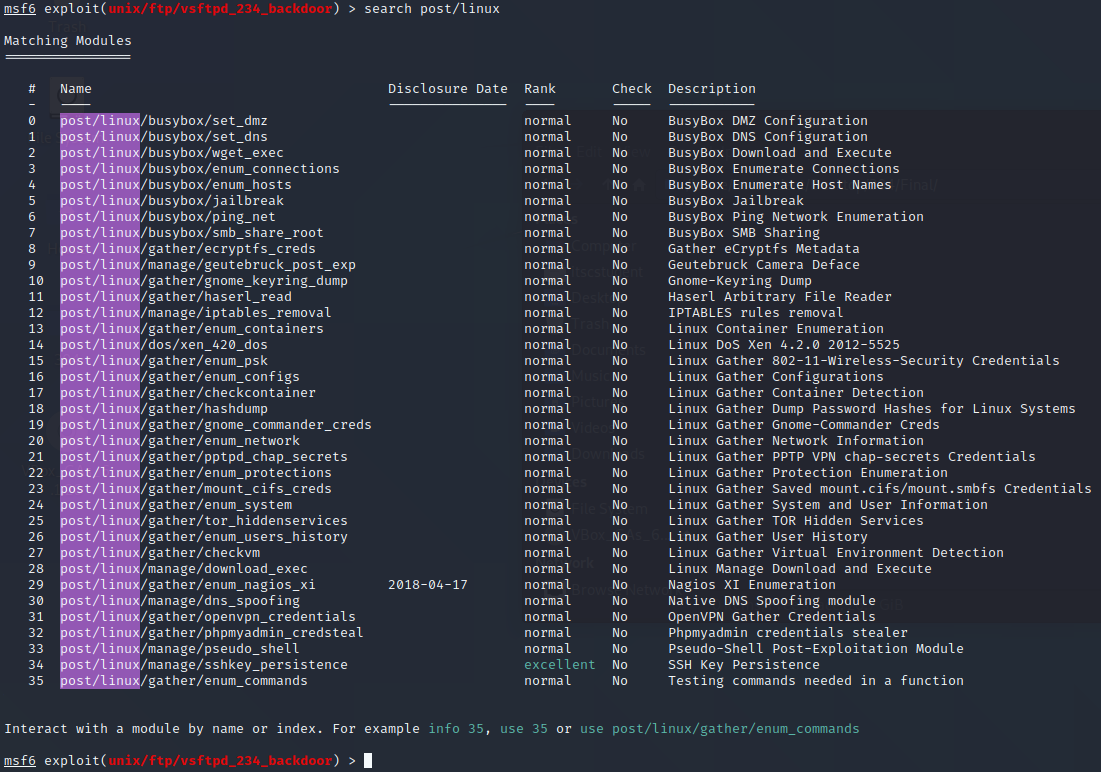


For a third time, root access is rapidly gained, and done so with no resistance.

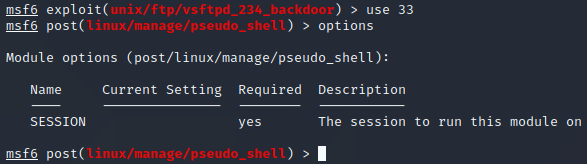
Any of these exploits can be used for post-exploitation, so the exploit against the vsFTPd server will be used. The exploit was run, and root was gained just as before, but a new post-exploitation exploit is required. Firstly, the running session was backgrounded:



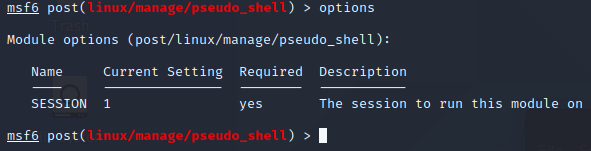
Now a new exploit needs to be found in Metasploit:



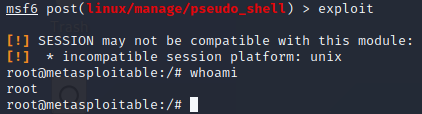
Out of the listed ones, there is one listed specifically as post-exploitation, so that is the one that will be used, and the options for it will be reviewed:



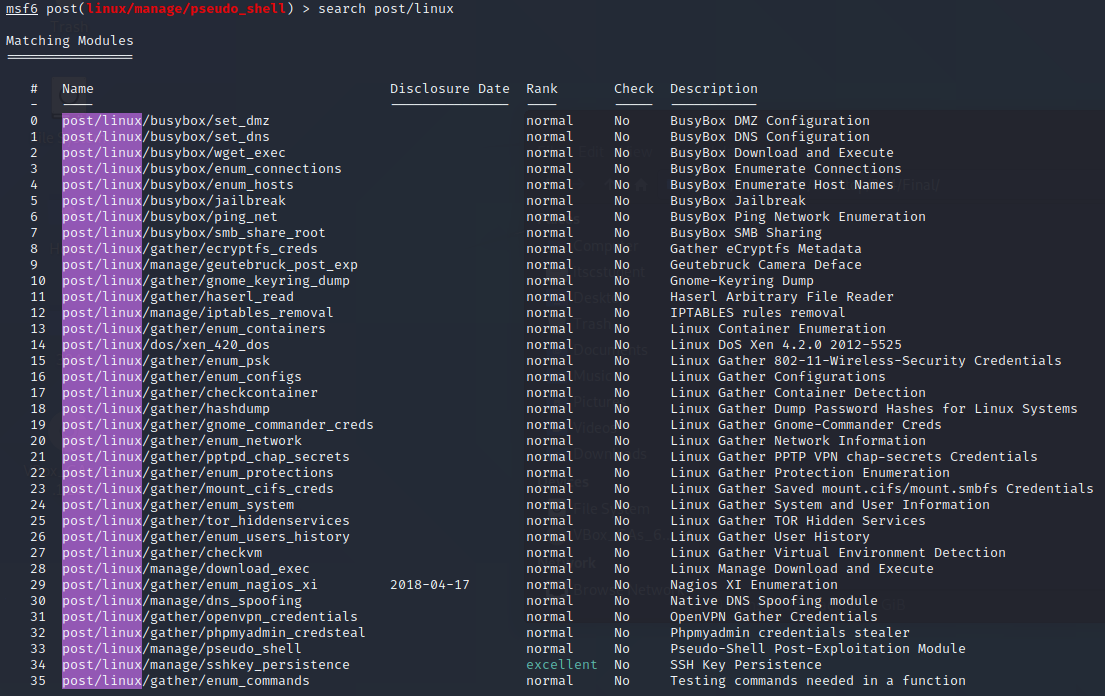
With the session that was created before, it can be utilized for this:



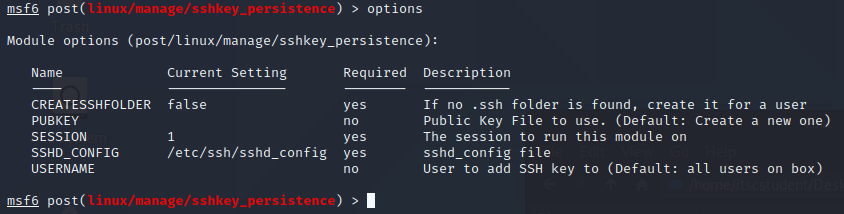
Running the exploit enters into root immediately:



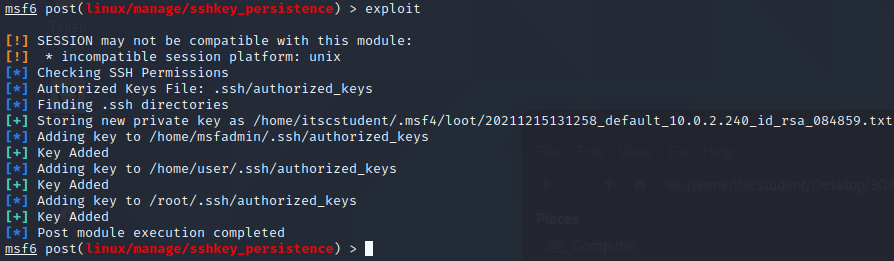
Since the exploit was simple to deploy, another one will be launched as well, this time pursuing persistence:



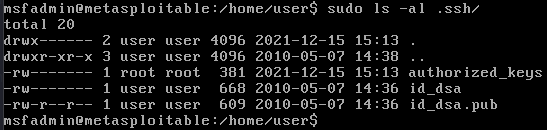
The options were set for the exploit:



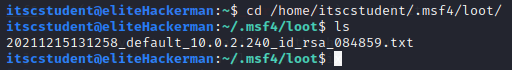
With nothing more required, the exploit is run and returns the desired results:



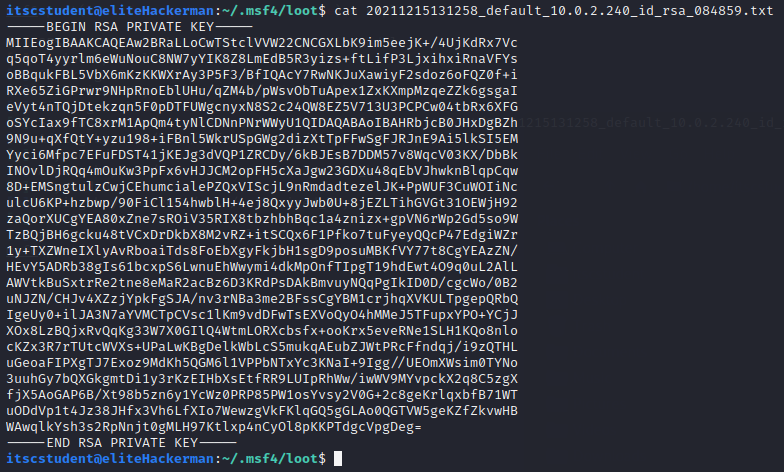
The attack is successful. Navigating to one of the listed locations, /home/user/ folder in the Metasploitable2 machine, the authorized\_keys file can be found:



Checking the attacker machine, the private key file can be seen:



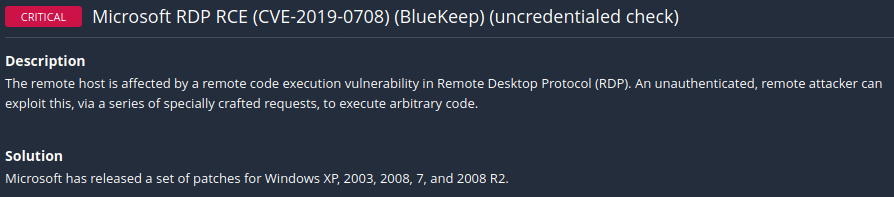
Viewing the private key contents reveals that the file is populated:



Post-exploitation on the Metasploitable2 machine has been completed.

### Remediation

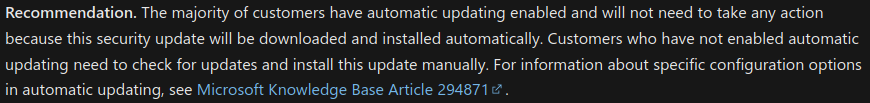
Remediation is straightforward for both systems: they need to be updated to their latest versions, and the installed software needs to be updated to their latest versions. Many critical security patches are missing because the installed versions are so old. The only addendum to be made is that telnet should not be active, and the port should be closed unless it is required. It is a known source of vulnerabilities and a common port that is included in many malicious scanners, such as Nmap or Nessus. Nessus states that the Windows 7 exploit that was run can be prevented with an update:



Microsoft, as well, state that the specific version of Windows 7 is a victim to this attack:



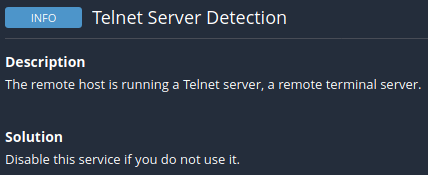
And an update was created to patch this critical vulnerability:



Additional information regarding security measures against the BlueKeep exploit can be found in this Microsoft article:

<https://www.microsoft.com/security/blog/2019/08/08/protect-against-bluekeep/>

For telnet, the solution listed inside of the Nessus scan also states that it should be disabled if not required:



All exploits used here can be prevented easily with updates, disabling any telnet servers, and closing the common telnet port.

# Post-Exploitation

## Privilege Escalation Path

There is no real describable path for privilege escalation. Both machines were missing any security measure to prevent trivial privilege escalation attempts, so all attempts made were successful. All that was required was a previously successful attack to get into the system and then a secondary exploit to get elevated privileges or persistence.

## Value of Information

All information is critical in value as the machines are compromised and fully accessible through these attacks. Transfers can be made the machines, increasing the potential damage considerable. Re-establishing the connection is trivial, only requiring a short few commands. Information is accessible at any time so long as the victim machine is powered on.

## Ability of Persistence

With the BlueKeep attack on the Windows 7 machine, the systems registry is accessed and a new key is added into the …/CurrentVersion/Run pathway to allow for persistence. The registry key written there will run every time the machine is powered on, allowing for a listening attacker to connect automatically. Given the vital role the registry plays in system operation, this is a substantial break in the machines defenses. No persistence is required with the Metasploitable2 box as it is crippled with so many critical security flaws that it can be accessed very easily and privilege can be escalated with ease.

## Countermeasure Effectiveness

No resistance was met from either machine during the entire exploitation process. The Windows 7 machine has its user access control setting and firewall disabled, so they posed no resistance. No countermeasures were met during the attack against the Metasploitable2 machine.

# Conclusion

Though the number of issues in both machines are numerous, many of them can be corrected easily via updates. The operating systems run out of date versions, and the software loaded into the machines are equally, if not more, out of date. Many critical security updates are missing for everything. Spending time to update all of these would increase the security posture of both machines considerably. Also, no protection is enabled, meaning no firewalls or security policies whatsoever, so the ones natively on the machines, like Windows Defender, should be enabled on the systems, even until a better solution is found. Ports that are not in use should be closed to prevent attacks against them, as seen on the Metasploitable2 box with telnet. Scanners can easily detect open ports, and malicious scanners often attempt to scan for commonly exploitable and notoriously vulnerable ports. Spending the time to make these changes will prevent all of the attacks that took place, in addition to many more that were not tested here.