Rameses Security LLC

113 W 60th St, New York, NY 10023
867-5309
RamesesSecurity.com

Network and Binary Exploitation: XYZ Corporation

March 18th, 2021

Network and Binary Exploitation: XYZ Corporation

Network and Binary Exploitation: XYZ Corporation

Executive Summary: *Overview of issues at C-Level and Business Impact*

Rameses Security LLC was contracted by XYZ Corporation to conduct a penetration test that evaluates the network security of a server and a desktop computer. The first test was run against a server called Metasploitable 2. The next test was run against a Windows 7 desktop computer. In addition to these tests, Rameses Security LLC also demonstrated how to conduct a buffer overflow attack. Buffer overflows are a common attack method used by threat actors. Rameses Security LLC deemed it necessary to show XYZ Corporation how these attacks are carried out and how to protect against them. In this report on the penetration tests, performed by Rameses Security LLC, we will also present proposed solutions to the vulnerabilities along with our methodology.

All penetration testing activities conducted by Rameses Security LLC were conducted using Kali Linux, Nmap, and Metasploit. Kali Linux is the same system that Rameses Security LLC has used in the past two penetration tests. Nmap, as a reminder, scans a network for open ports that attackers can exploit in a cyber attack. Metasploit is a tool that contains a large number of known vulnerabilities that can be used to simulate a cyberattack. Any threat actor or threat actors, could easily use the same tools to exploit the vulnerabilities of XYZ Corporation's network and cause serious harm to the business and reputation of XYZ Corporation. This means that anyone with a computer, the right tools, and the motivation to cause harm to XYZ Corporation could do so easily.

It's the hope of Rameses Security LLC that the leadership of XYZ Corporation will recognize the issues in their systems, consult their engineering team, implement new policies and procedures, and see to it that steps are taken to rectify the weaknesses. Rameses Security LLC firmly believes that the C-Suite Level members at any company must buy into security policies and procedures so that the whole company will have a stake in maintaining a good security posture that does not disrupt business continuity.

During the course of the penetration testing Rameses Security LLC gained remote access into XYZ Corporation's network and successfully carried out cyberattacks.

Network and Binary Exploitation: XYZ Corporation

Rameses Security LLC only conducted a handful of attacks against XYZ Corporation, but as the Metasploit Tool demonstrates, there are many more attack options for threat actors to choose from. The harm that a successful attack would cause to XYZ Corporation and the fallout that would ensue cannot be overstated. The leadership team of XYZ Corporation needs to recognize this and work to remedy the issues as soon as possible. If the services provided by XYZ Corporation were disrupted by a cyberattack the company would sustain a financial loss and weakening of their, currently, stellar reputation.

Rameses Security LLC is happy to provide suggested necessary actions to the wonderful people at XYZ Corporation. These suggested actions will ensure that XYZ Corporation has the knowledge and ability to address the vulnerabilities within its systems and achieve a high level of confidence in its security posture. When the actions outlined in this report are acted on XYZ Corporation can be confident that they will have done their best to protect their assets.

Rameses Security LLC thanks XYZ Corporation for the opportunity to conduct penetration tests on their network. We look forward to a continued partnership moving forward.

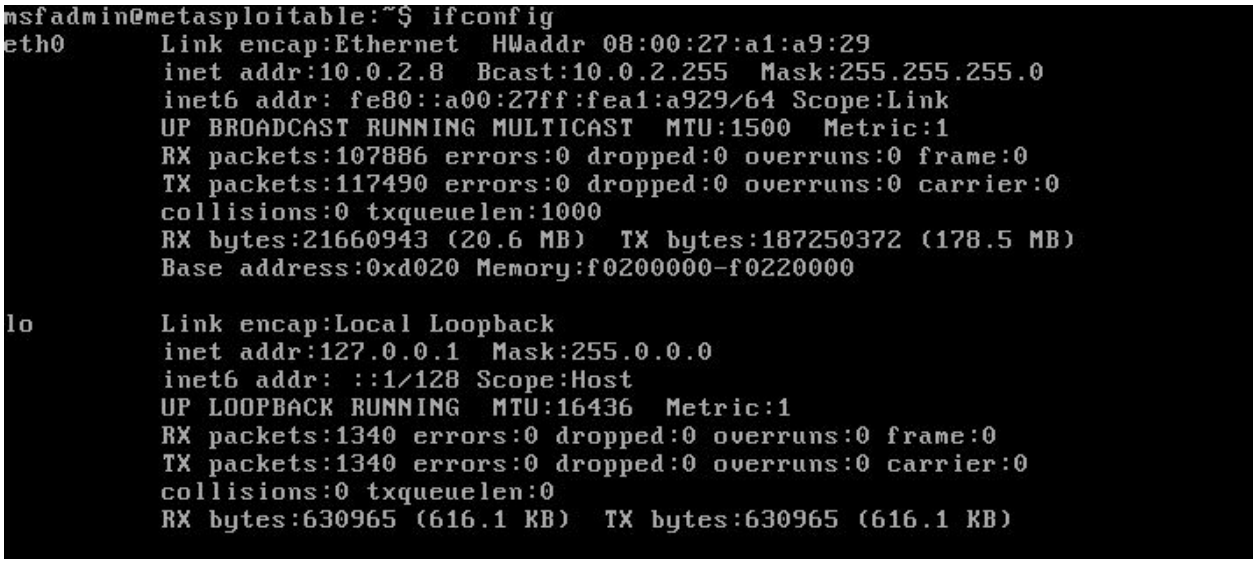
Metasploitable 2 Server Findings

Rameses Security LLC began by performing a penetration test against the Metasploitable 2 server, owned by XYZ Corporation . We executed two payloads against the server.

Before we dive into the exploits themselves, we need to understand the setup that allows an attacker to execute any exploits. The exploits could not be performed without the IP address. An IP address can be obtained a few different ways, all of them simple. An attacker could go directly to the website or use a 3rd party lookup service. The IP

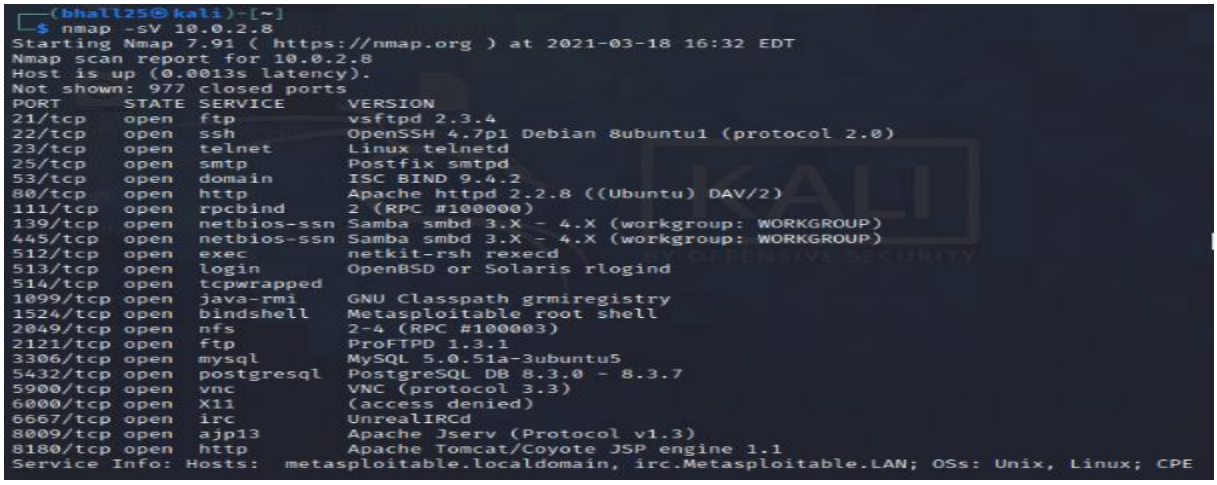
address of the Metasploitable 2 machine is 10.0.2.8, as shown in Figure 1. For the sake of this report we simply wanted to verify that we have the correct IP address.

Figure 1



Now that we know the IP address we need to run an Nmap scan to see what ports are open on the Metasploitable 2 server. Open ports are glaring vulnerabilities for any system. We opted to do a service and version scan, **-sV**, because this allows us to use exploits that work against the specific versions running on the Metasploitable 2 server. Without knowing the versions we may have a harder time gaining access to the target and conducting a successful attack. Below are the results of the scan in Figure 2.

Figure 2

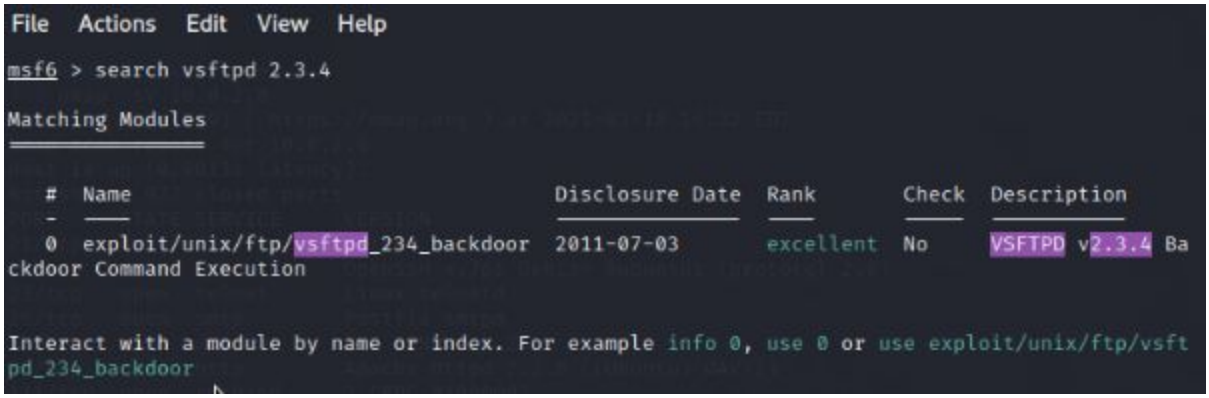


Now that we have identified the open ports the next step is to take these results and use Metasploit framework to exploit any vulnerabilities against Metasploitable 2.

1st Exploit of Metasploitable 2 server - ftp vsftpd 2.3.4

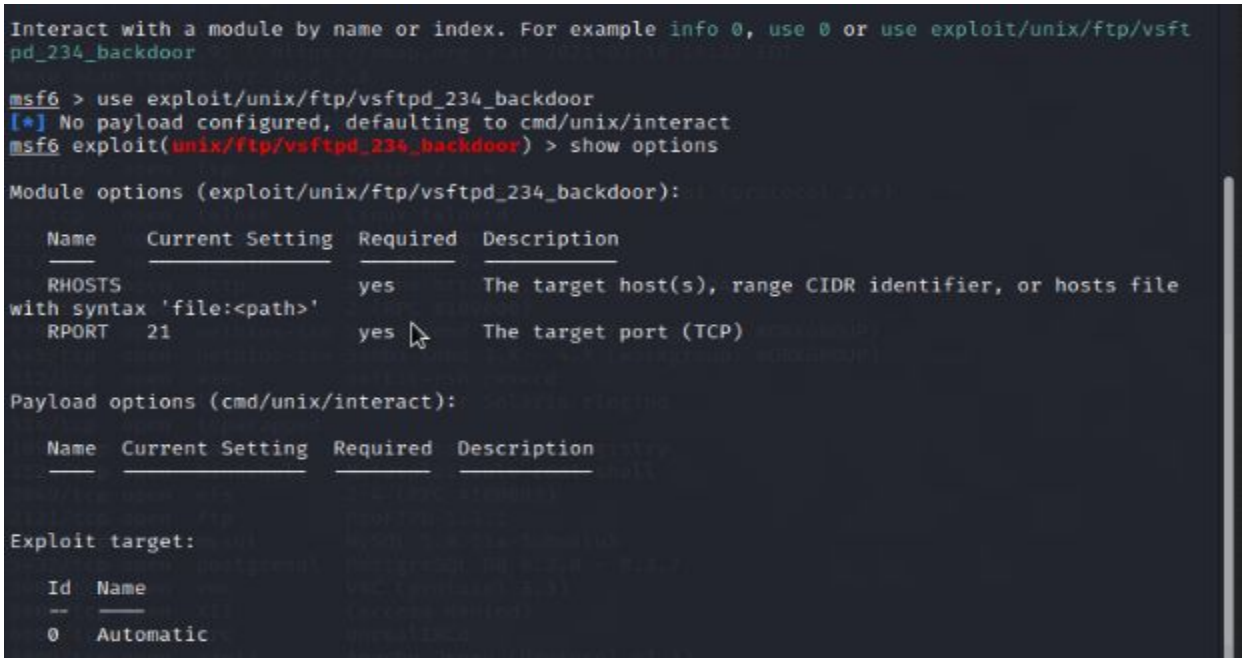
The very first open port found in port scan was *port 21, ftp*. Refer back to Figure 2 to see this. The service, *ftp*, stands for file transfer protocol and it does exactly what the name implies it does. It's a method for transferring files. The version running on the Metasploitable 2 server is *vsftpd 2.3.4.* Now we can go into Metasploit and type **search** followed by **vsftpd 2.3.4**. Because Metasploit has a large database of exploits it's best to be specific and narrow down our search, hence why we searched for the specific version of *ftp*. We then combed through the results and selected our exploit of choice for the attack. The Metasploit search returned with a vulnerability that uses a backdoor to gain access into a system via this version of *ftp*. Figure 3 shows us these results.

Figure 3



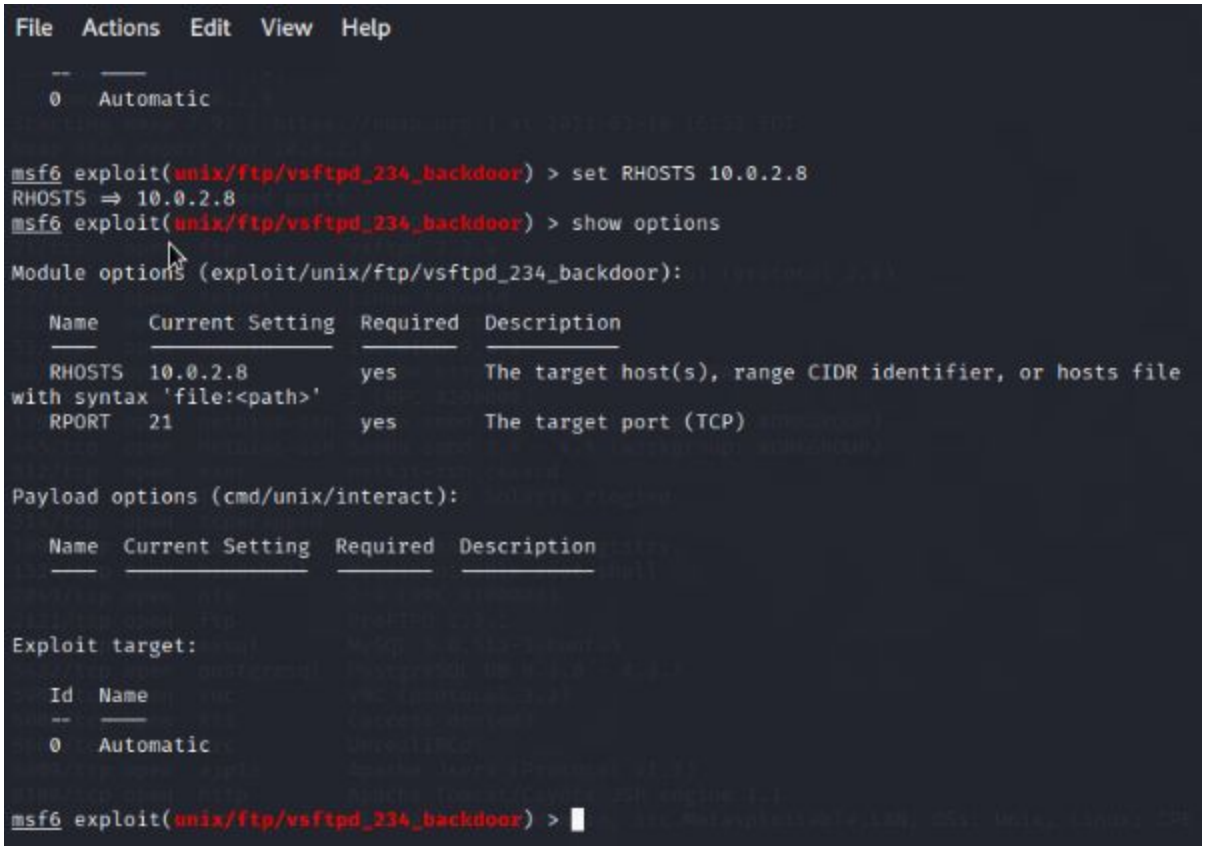
The next step is to copy the name of the exploit and then type **use** followed by the exploit name that we have copied into the system: **exploit/unix/ftp/vsftpd_234_backdoor**. This drops us into the module prompt as evidenced by the text color changing to red. See the top of Figure 4, below. Once in the module we need to configure that module. We type in **show options**. Refer to the top of Figure 4 to see this and the results it yields. We can see all of the settings for the module. The results show us that we need to make some changes.

Figure 4



Looking at Figure 4 we can see that there is a blank space next to **RHOSTS** and this is where we need to input the victim's IP address. In this case the victim is Metasploitable 2 and the IP address, which we found earlier, is 10.0.2.8. The command that we will use is **set RHOSTS 10.0.2.8**. We can then do the **show options** command again to verify that our changes were stuck. Figure 5 shows us that we have successfully made the update that we sought to make and that the Metasploitable 2 IP address is now listed as our target.

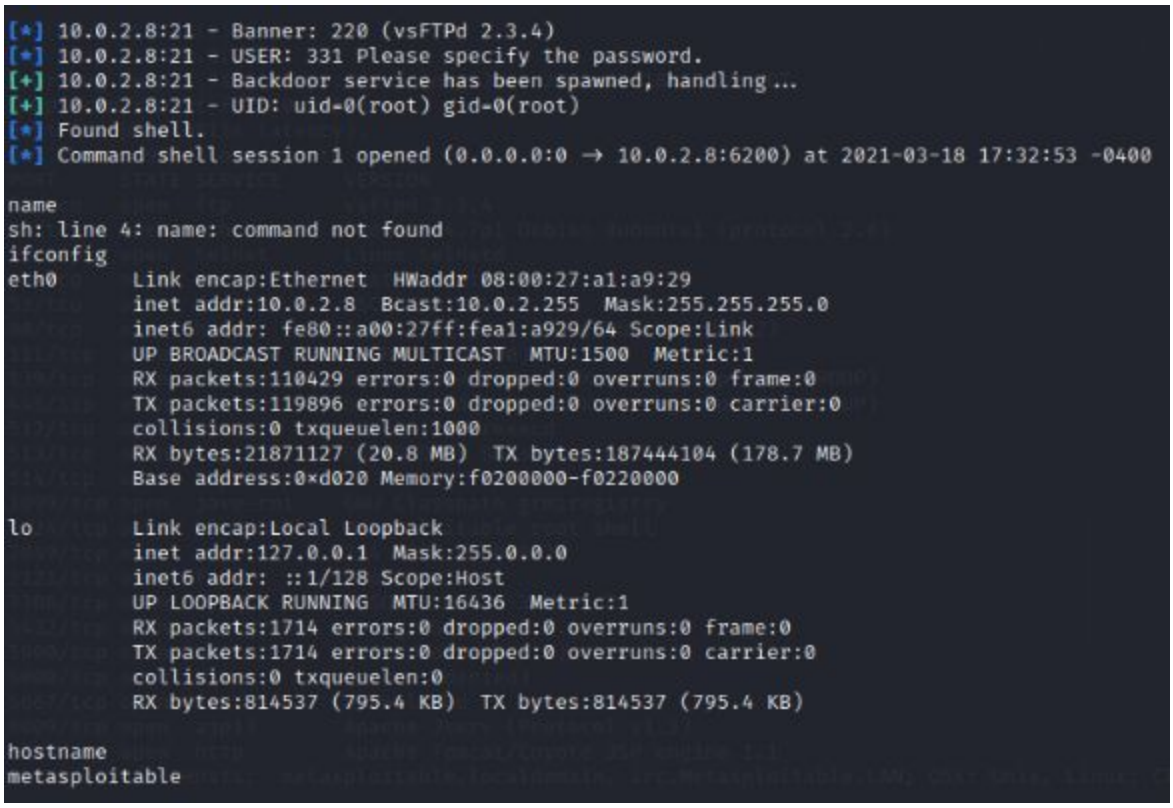
Figure 5



Now all that we need to do it type in **exploit**. We have now gained access to the Metasploitable 2 server as we can see from the newly opened *command shell* in Figure 6. We can execute commands like **ifconfig**. We can compare these results, seen in Figure 6, to the results from Figure 1. In Figure 1 we were able to view the IP address of Metasploitable 2 and a lot of other information. In figure 6 we can see that the **ifconfig**

command yields the same results. If we type in **hostname** we can see that the system verifies that we are inside Metasploitable. Please see Figure 6 for these results.

Figure 6



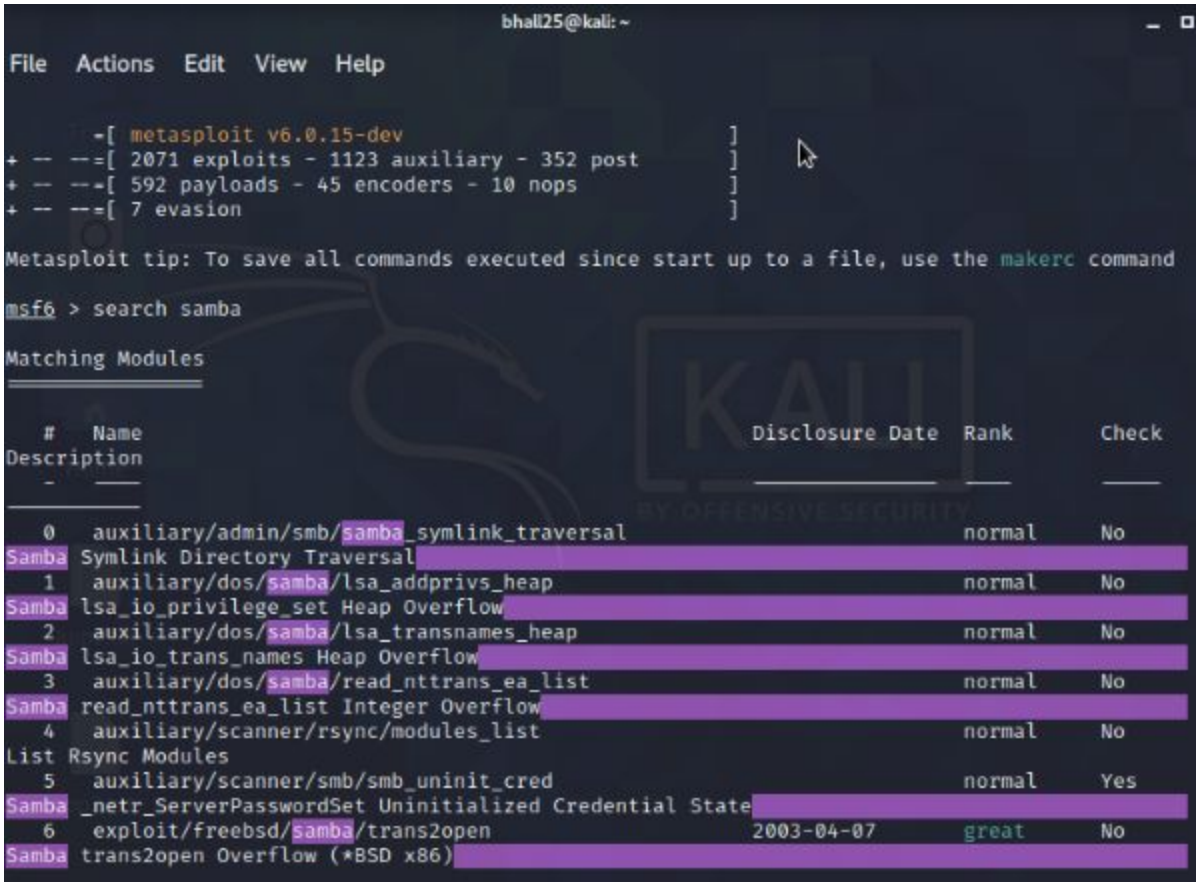
FTP is simply not a secure system. It relies on cleartext names and passwords and does not use encryption. FTP is vulnerable to sniffing, spoofing, brute force attacks, among many others.

2nd Exploit of Metasploitable 2 Server - netbios samba smbd 3.X - 4.X

If we refer back to Figure 2 we know that there are no shortage of vulnerabilities that can be found against the Metasploitable 2 server owned by XYZ Corporation. The next exploit that we performed was done against *port 139, netbios-ssn,* which was running version *Samba smbd 3.X - 4.X*. *Samba* is a free software service that gives end users access to resources including printers, file sharing, among others. *Samba* is a useful

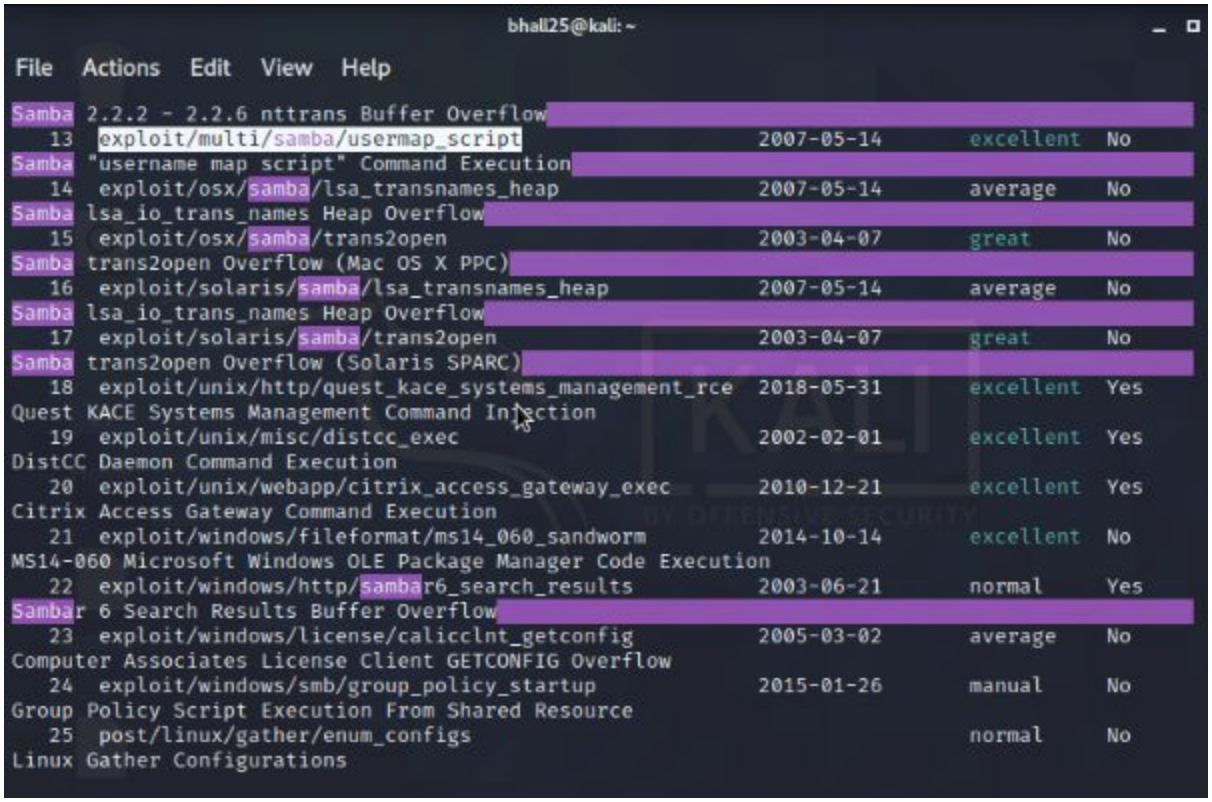
program, but it's not without its vulnerabilities. In this exploit Rameses Security LLC was able to gain remote access into Metasploitable 2 through *Samba*. We used the same steps as we did for the previous exploit. Once we had the target and our route for getting there we opened Metasploit and did a **search** for **samba** that yielded a lot of options to choose from as seen in Figure 7.

Figure 7



Then we selected our exploit. For this exercise we selected an exploit called **username map script**. This exploit allows an attacker to bypass authentication and execute arbitrary commands. Figure 8 shows the exploit, it's the 13th option on the list.

Figure 8



Once our exploit was selected we then proceeded to set up the payload to give us access to the Metasploitable 2 server. We copied the name of the exploit, **exploit/multi/samba/usermap_script,** typed **use** into the command line and then pasted the name of the exploit next to it. We then used the **info** command to see if our attack needed more details added to it in order to execute. Once again we simply needed to add the target IP address into the payload with **set RHOSTS 10.0.2.8.** Once the payload was ready we used the **exploit** command and successfully gained access into the Metasploitable 2 server. Once inside we used **ifconfig** to demonstrate that we were in fact in the system and able to execute commands. We also did a **whoami** to demonstrate that we had gained root access. Figure 8 has the first couple of steps and Figure 9 shows us adding in the IP address to specify the target, gaining access to the Metasploitable 2 server with root privileges, and performing various commands.

Figure 8

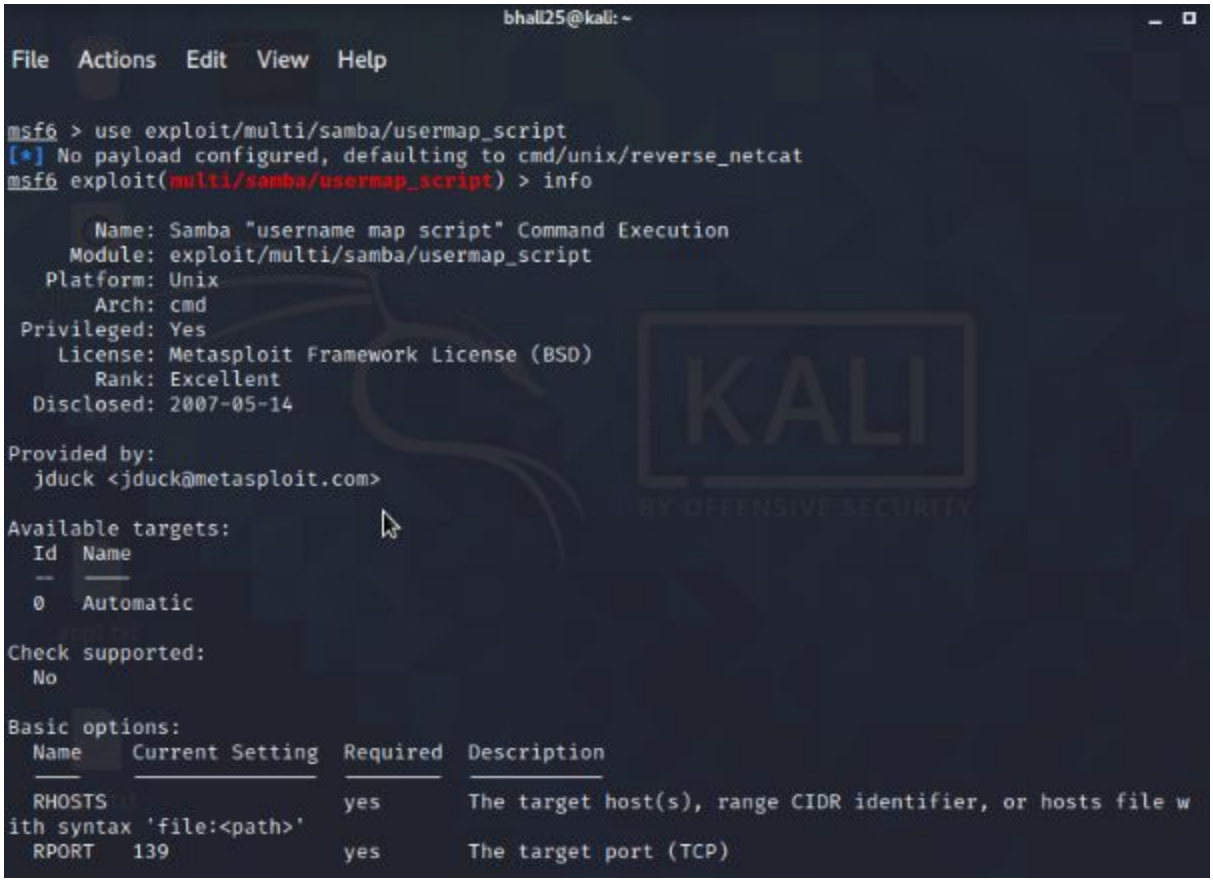
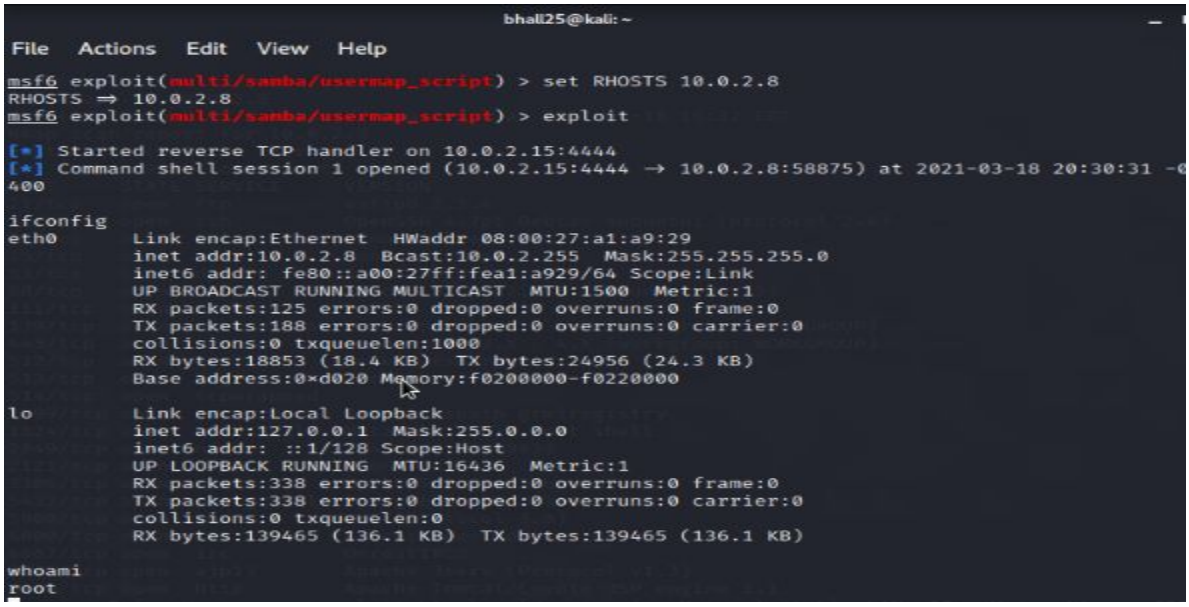


Figure 9



Network and Binary Exploitation: XYZ Corporation

Windows 7 Desktop Findings

The second portion of the project was a penetration test of a Windows 7 desktop computer that XYZ Corporation wanted us to attempt to hack. We executed two payloads against the computer.

As was the case with the Metasploitable 2 server the exploits could not be performed without the IP address. The IP address is 10.0.2.6. We then ran an Nmap scan, against the computer. Figure 10 verifies the IP address of the target computer and Figure 11 shows the results of the Nmap scan. As we can see from this scan the Windows 7 machine has fewer open ports to attack than the Metasploitable 2 server, but we still have plenty of options.

Figure 10

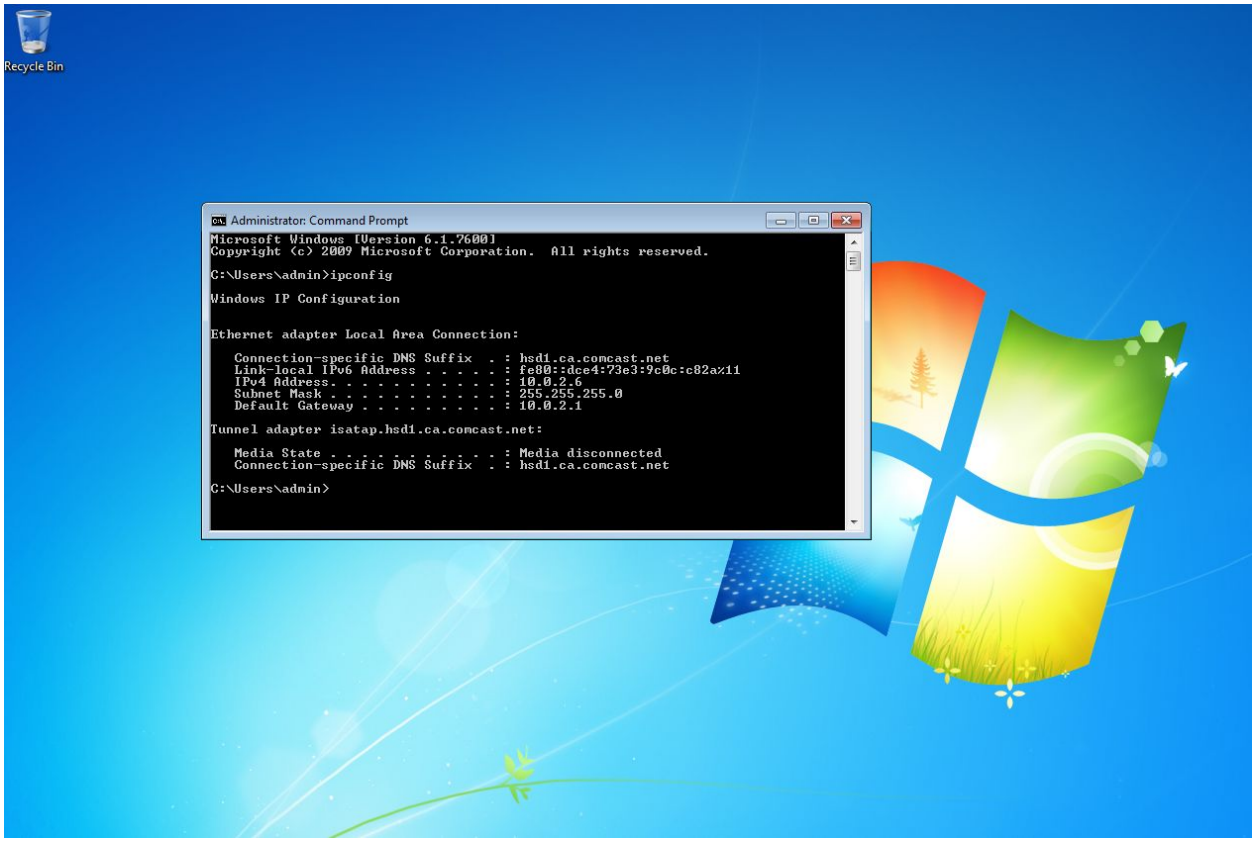
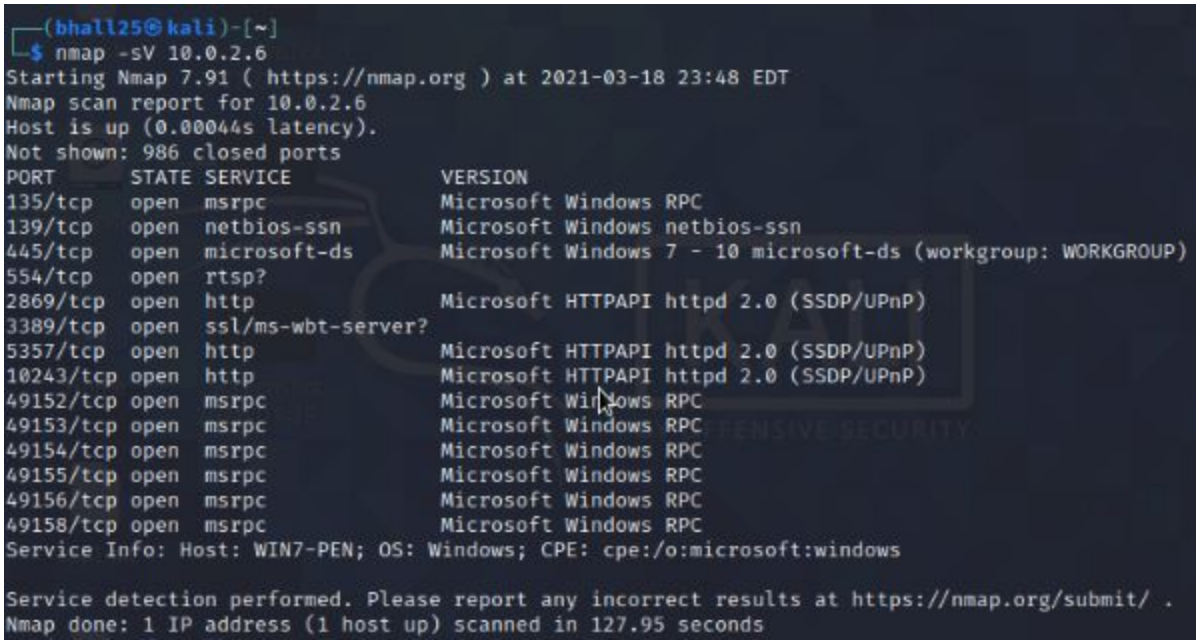


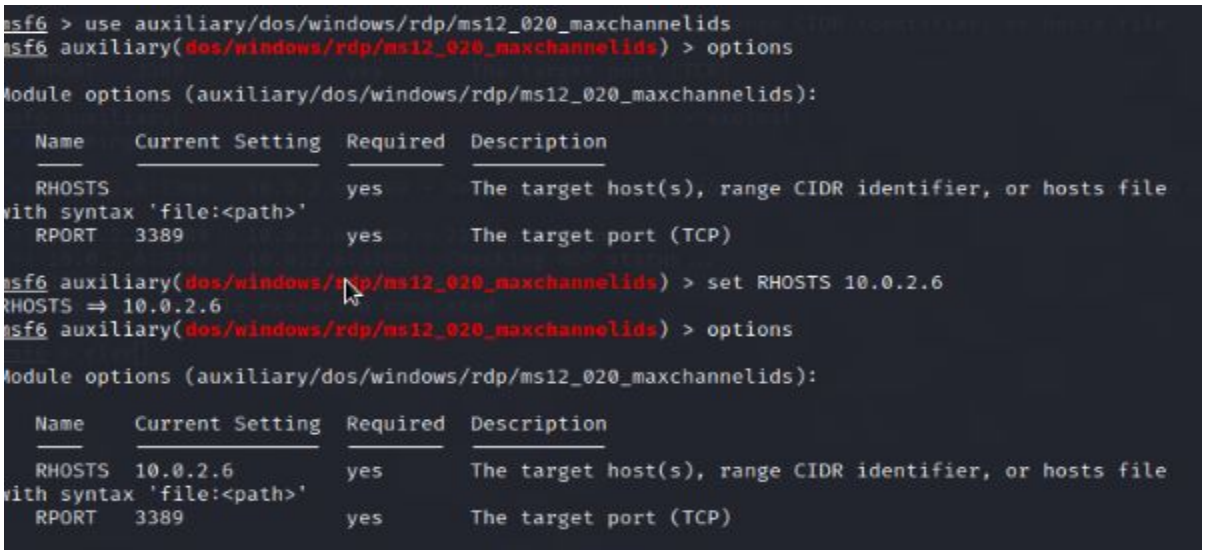
Figure 11



1st Exploit of Windows 7 Desktop - DoS Attack

The first exploit that we conducted against the desktop computer was a denial of service attack. The DoS attack exploits the *MS12-20* vulnerability. Just like we did against the Metasploitable 2 server, we selected our exploit and typed in the command **use** followed by the exploit into Metasploit: **use auxiliary/docs/windows/rdp/ms12_020_maxchannelids**. We then set the target to be the Windows 7 desktop with IP address 10.0.2.6 and verified that it stuck with the **options** command. Figure 12 shows these steps that we followed.

Figure 12



Once we confirmed that it did in fact stick, we executed the attack against the Windows 7 desktop computer that was our target. We used the **exploit** command to do so. If you look at Figure 13 you will see that Metasploit delivered the payload and forced the Windows 7 Desktop computer to crash. Figure 14 shows the results of the crash as seen from the perspective of the Windows 7 desktop computer. In looking at the screenshot you can see that the computer simply had to shut down in order to “prevent damage” to the computer. This result is sometimes referred to as the “blue screen of death.”

Figure 13

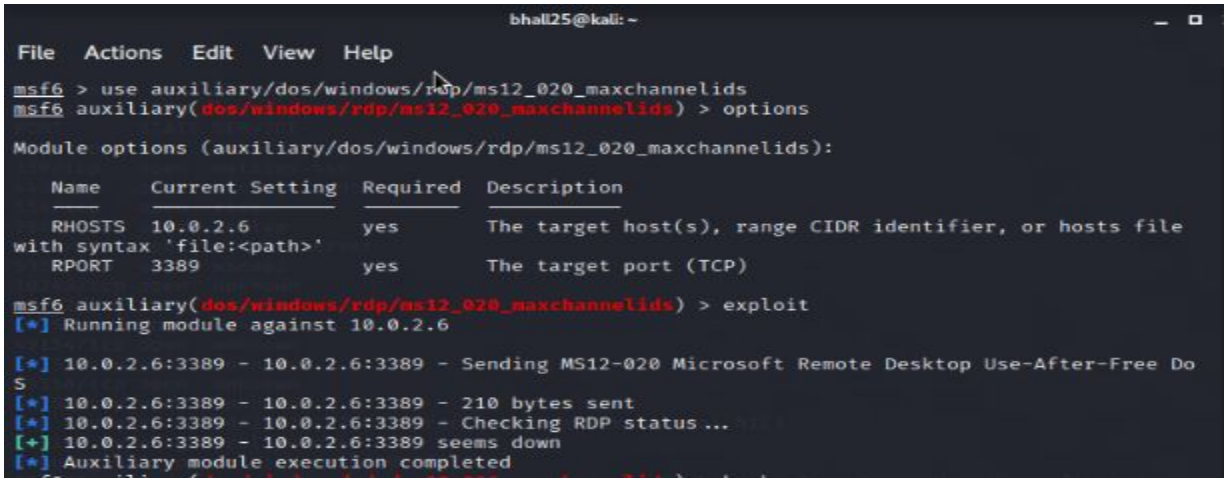
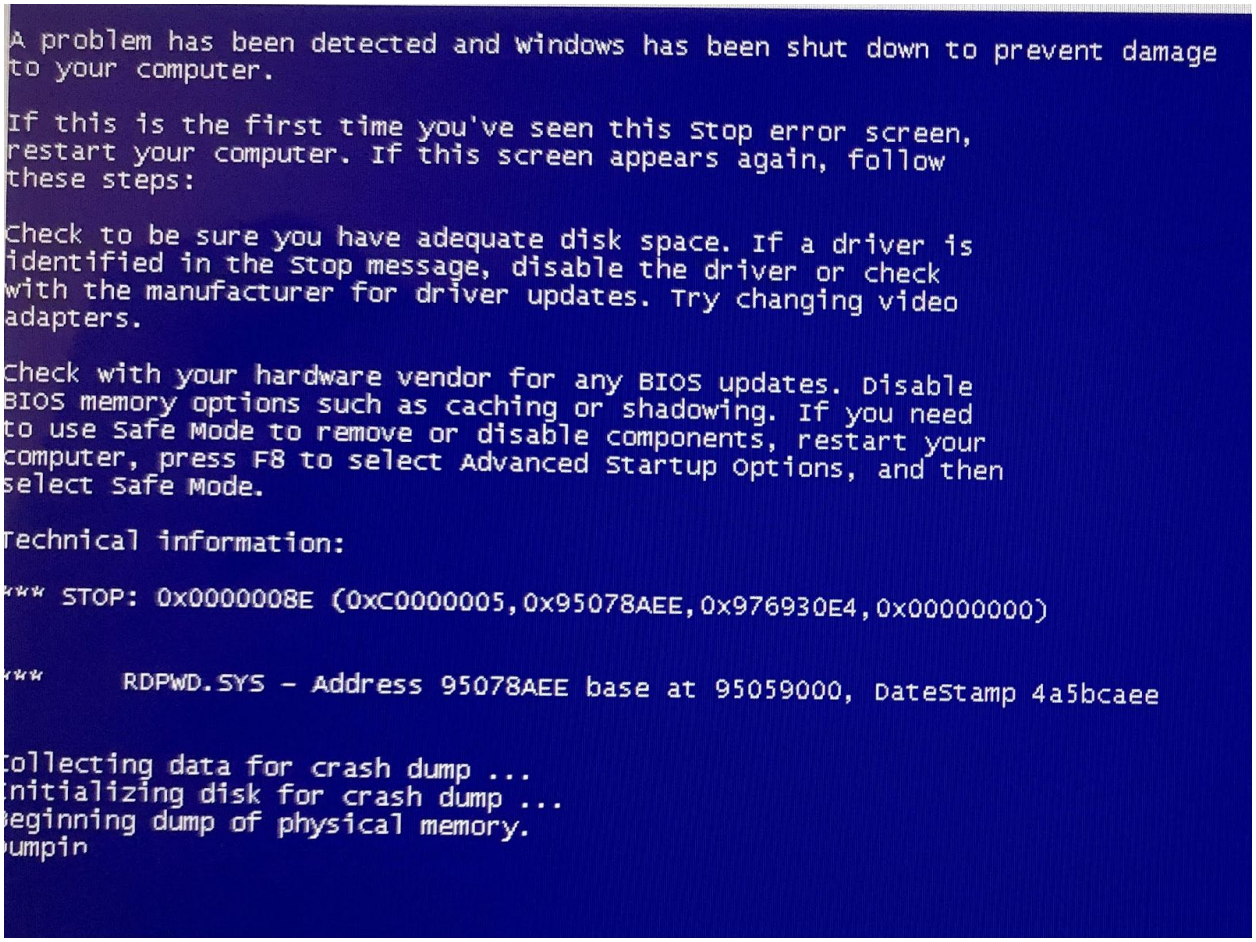


Figure 14



2nd Exploit of Windows 7 Desktop - Eternal Blue

The next exploit that we ran against the Windows 7 desktop is called eternal blue. Eternal Blue, when executed successfully, forces a machine to crash and return with the infamous “blue screen of death.” To use this exploit we searched for **smb** in Metasploit. We then located the eternal blue attack from the returned search results. Figures 15 and 16 contain these steps.

Figure 15

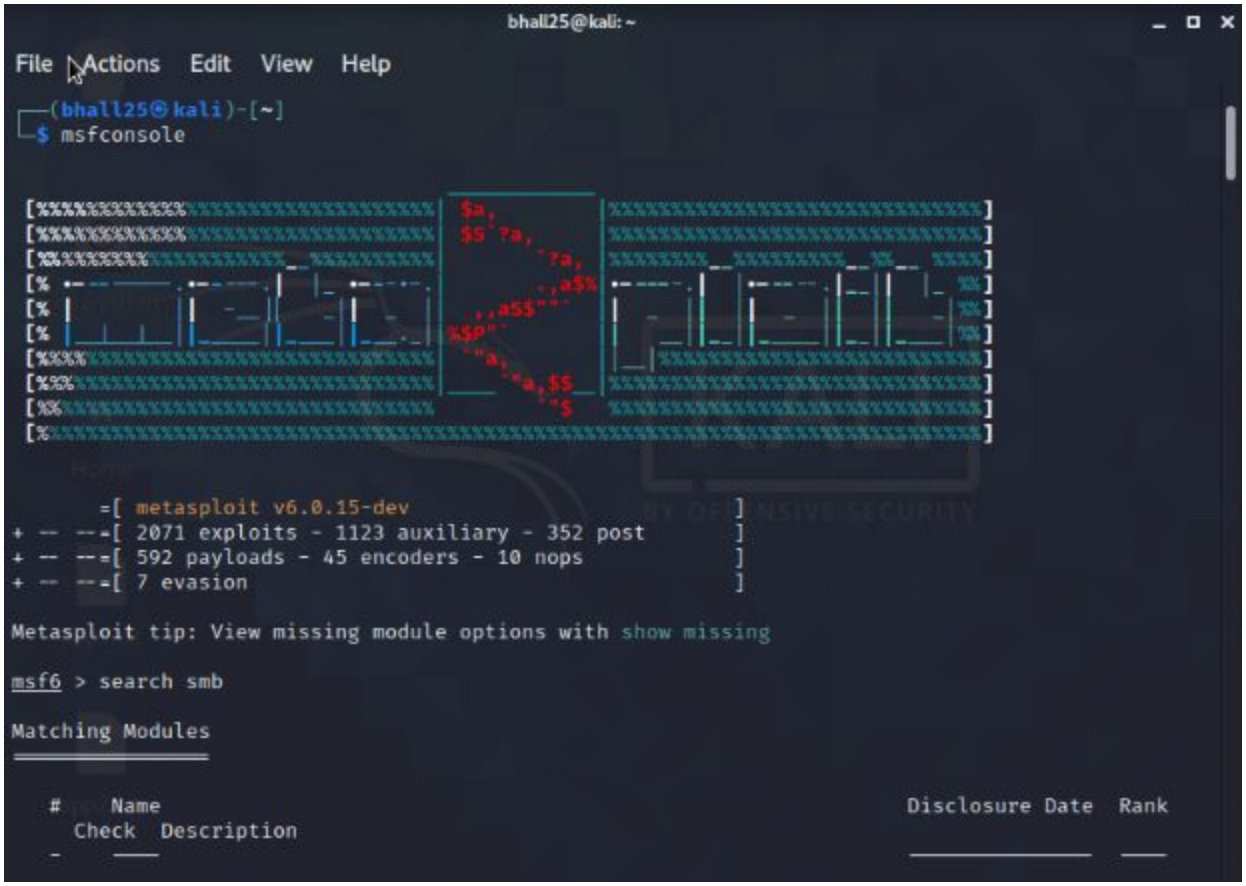
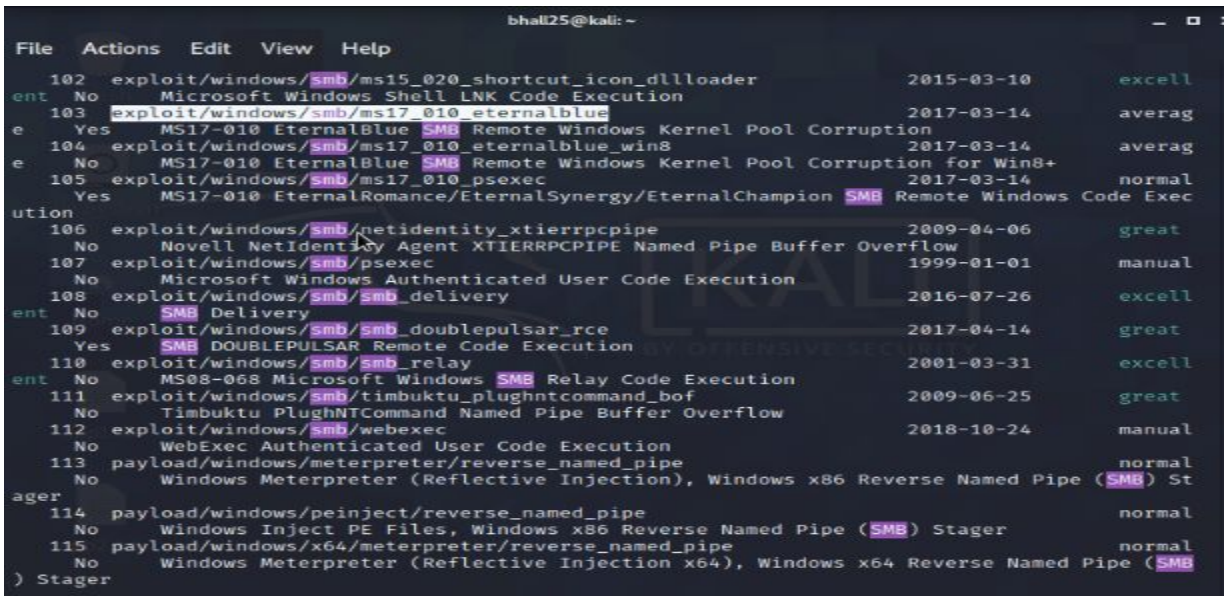


Figure 16



Network and Binary Exploitation: XYZ Corporation

The next step was to follow the same instructions that we did each of the previous exploits which was to use the exploit, input the IP address of the Windows 7 desktop that we're targeting, confirm that the IP stuck, and then deliver the payload to the target. Figures 17 and 18 show the results of these actions.

Figure 17

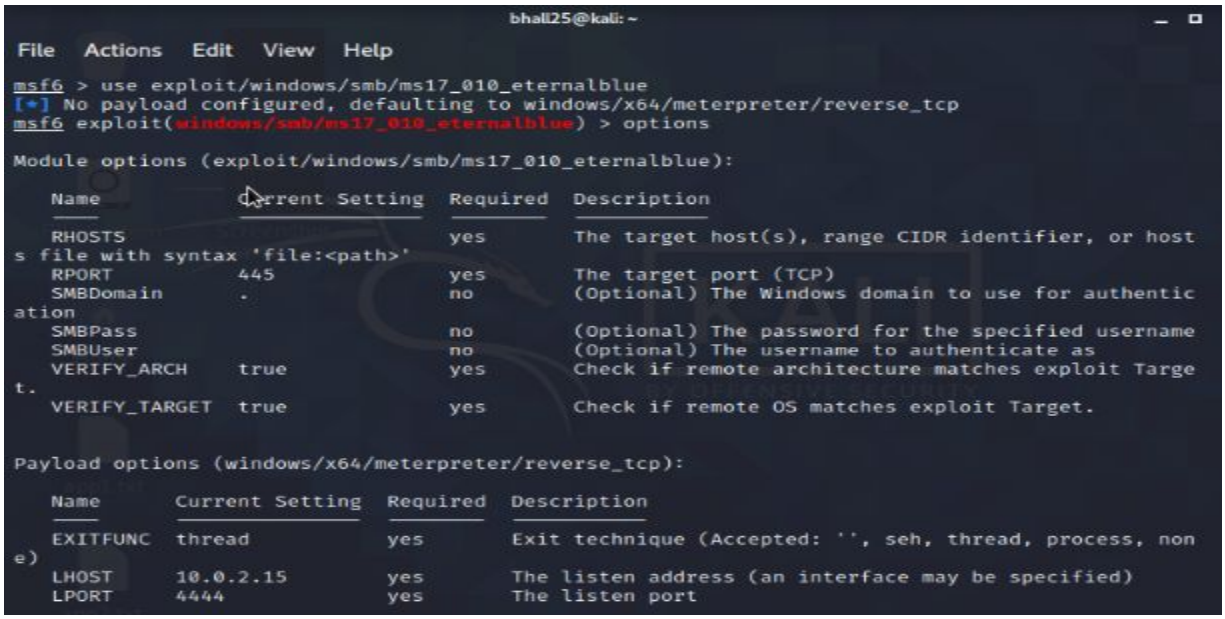
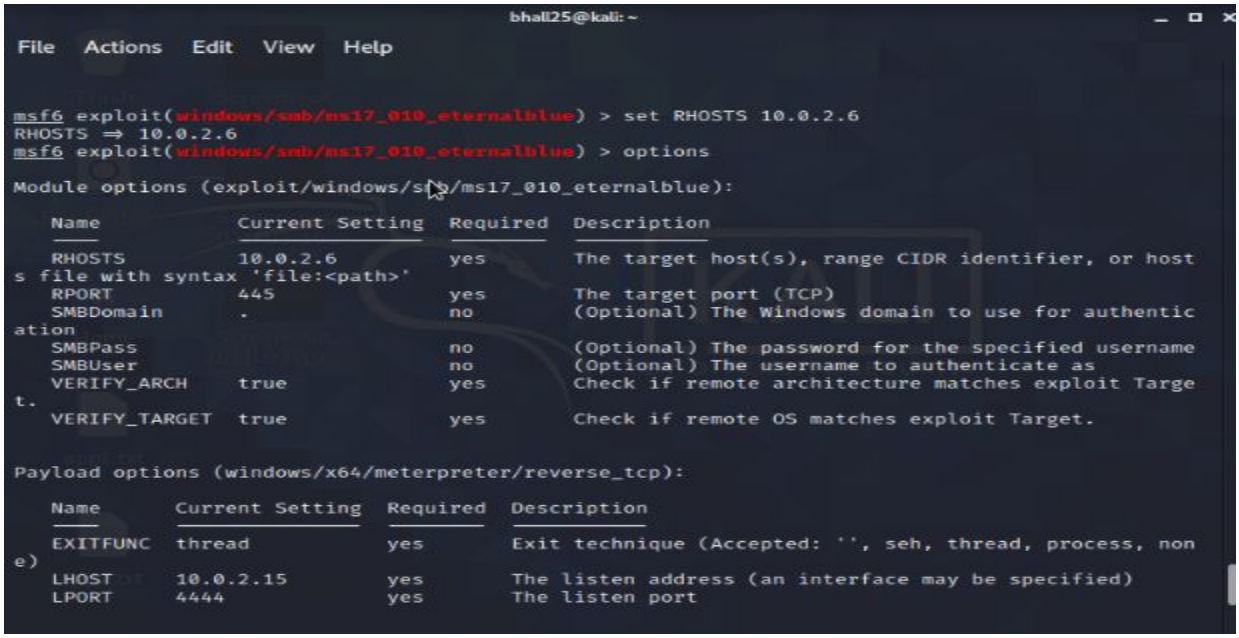


Figure 18



Network and Binary Exploitation: XYZ Corporation

The next step was to execute the attack against the Windows 7 desktop machine. We did this by using the exploit command. This resulted in the “blue screen of death” which knocked the target computer out of commision. Figure 19 has the execution in Metasploit and Figure 20 shows the resulting “blue screen of death” as seen from the Windows 7 Desktop computer.

Figure 19

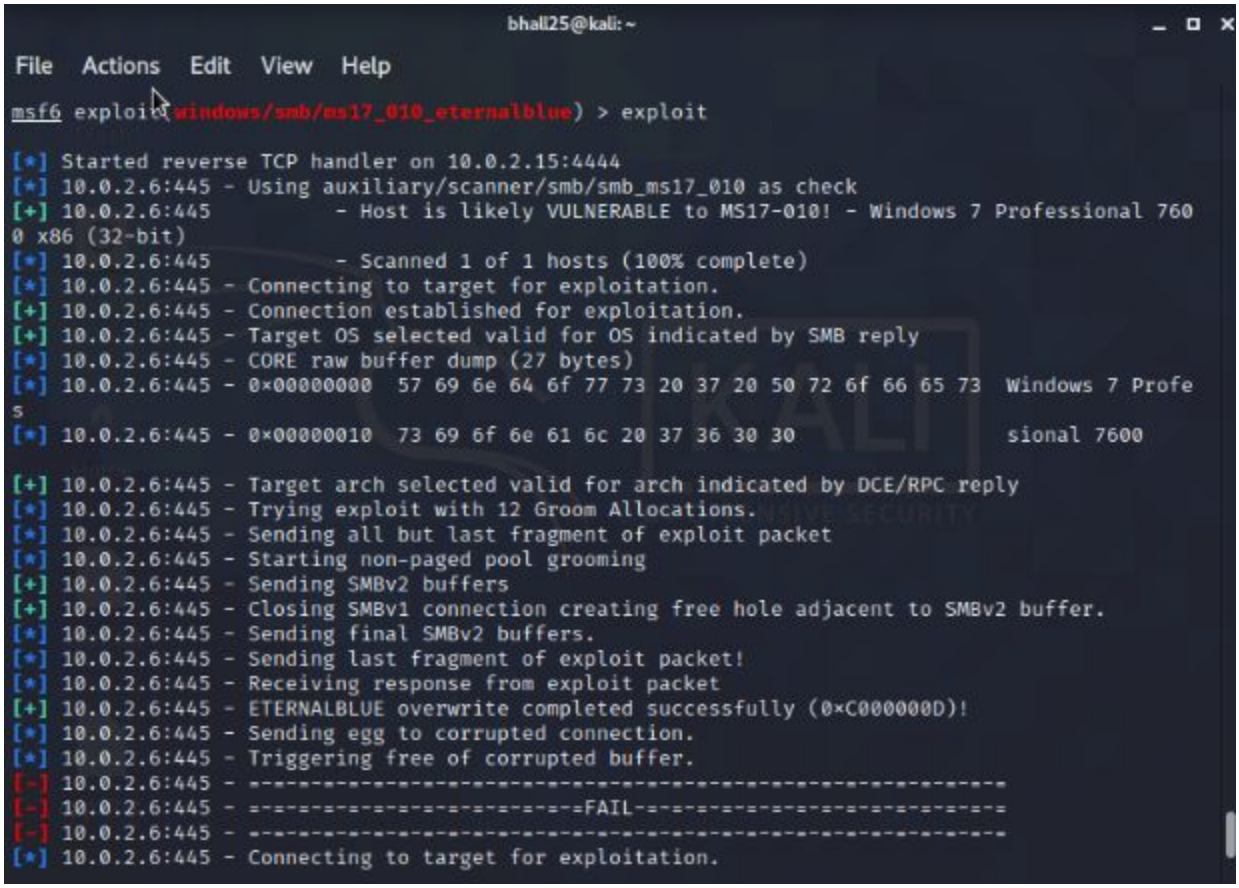
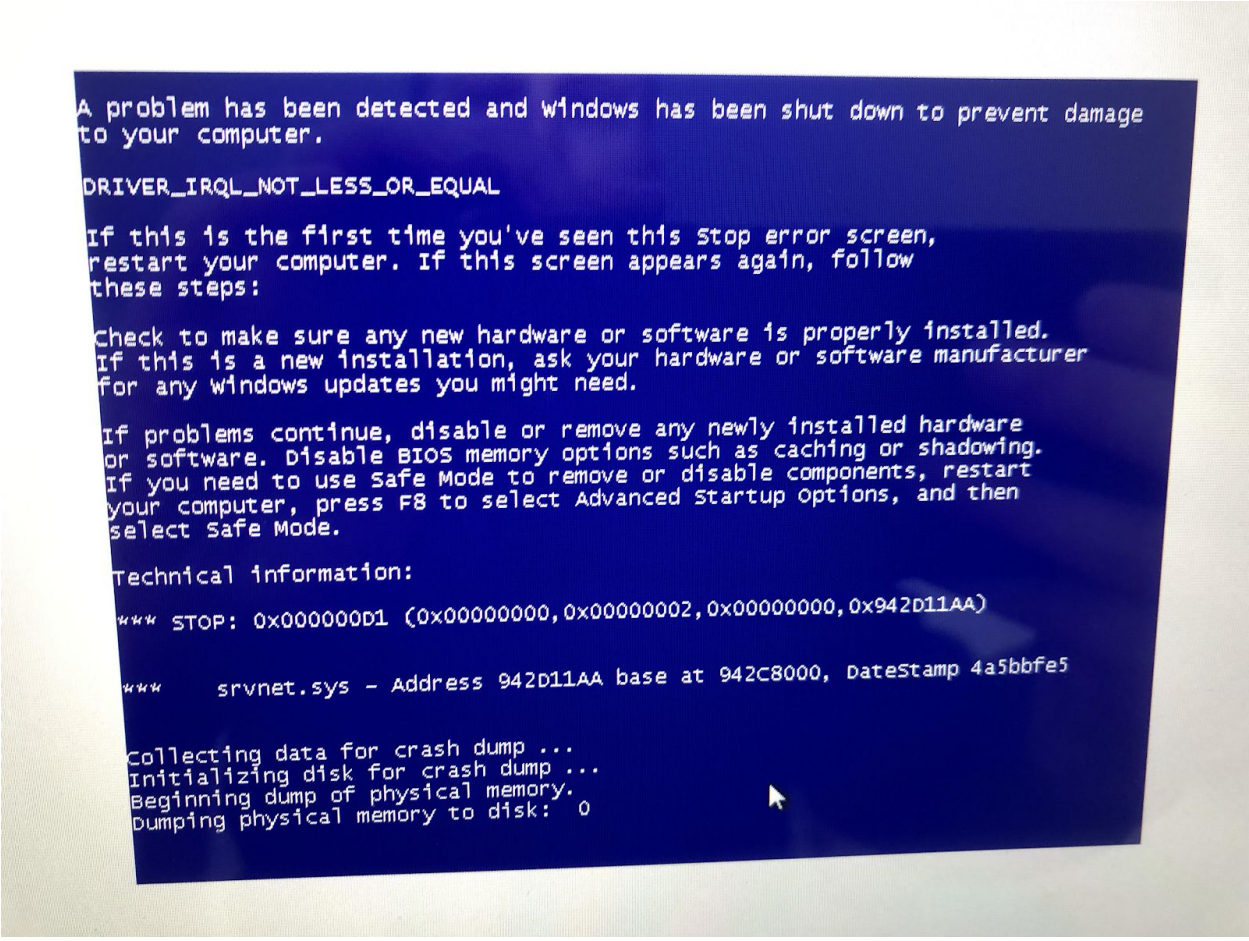


Figure 20



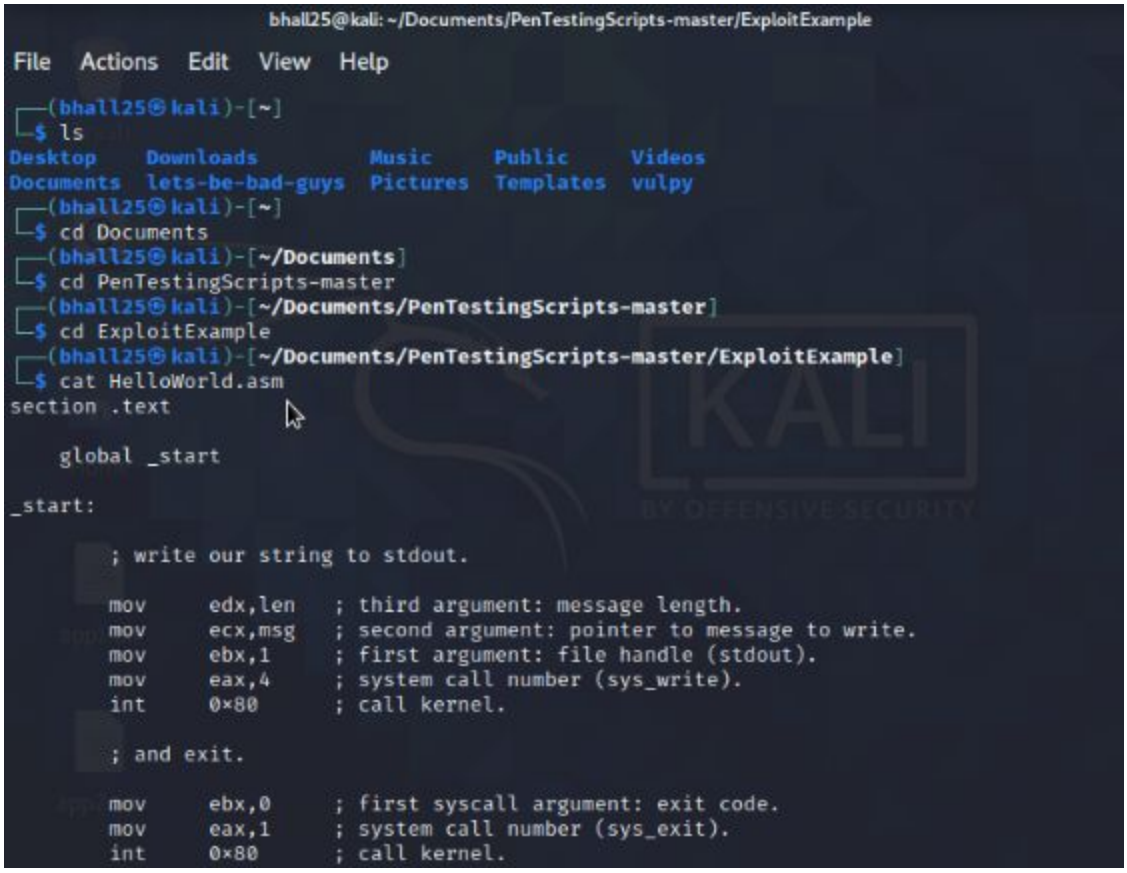
Binary Exploitation - Buffer Overflow

In addition to exploiting the two machines, Rameses Security LLC also demonstrated a buffer overflow exploitation. A buffer overflow is a common attack method that occurs when an attacker puts too much into the stack and the amount of data exceeds the capacity of the stack. A buffer overflow will cause a system to crash. While these attacks are being phased out by better programming practices they still occur far too often.

To simulate the buffer overflow we began by accessing a custom script. Normally the program will execute a function called *main()* which in turn calls a function named

echo() and requests input from the user. In our script there is a function called *secret function()*. This is where we will execute the buffer overflow and force the program to execute. We used the command **objdump -d vuln.** We then scroll through and we can see the *secret function()*. The *echo function()* will take the user input and outputs data. Figure 21, figure 22, and figure 23 demonstrate these steps.

Figure 21



Network and Binary Exploitation: XYZ Corporation

Figure 22

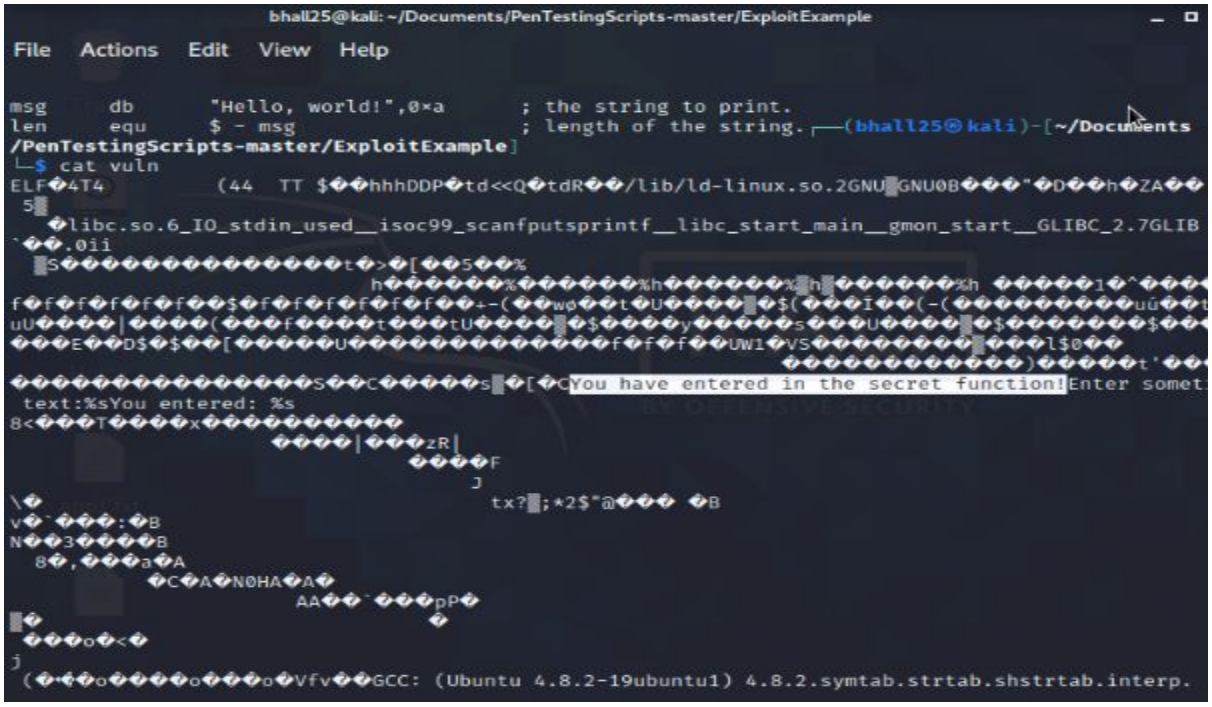
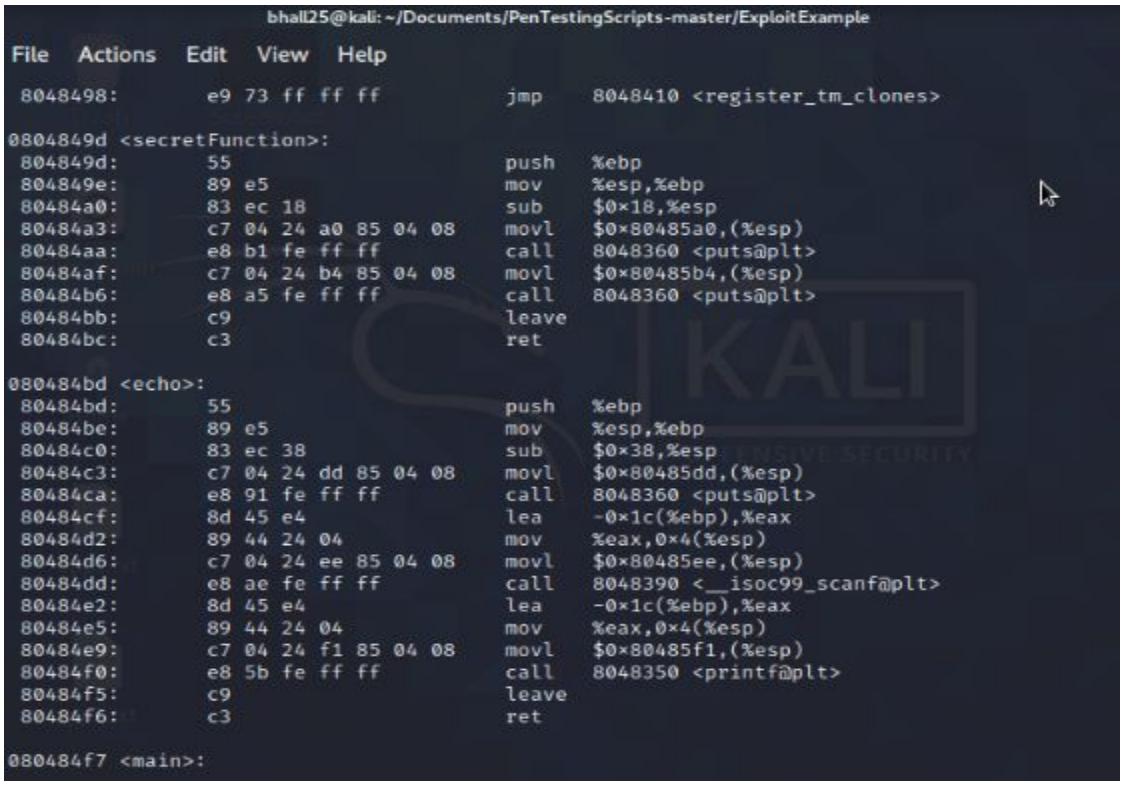


Figure 23



Network and Binary Exploitation: XYZ Corporation

Referring back to Figure 23 we see that the esp has 28 bytes allocated for the buffer. Now we know that the 4 bytes after it are arbitrary and the following 4 bytes are the address of the *secret function().* So our input string is as follows: 28 * 1 byte + 4 bytes = 32 bytes. We now run the command: **python -c 'print "a"*32 + "\x9d\x84\x04\x08"' | ./vuln.** This compiles and executes the command instructions. As you can see from Figure 24 we have successfully run the command, please take note of the success messages confirming that we ran the *secret function()*.

Figure 24

Recommendations

Tactical Recommendations

Rameses Security LLC recommends that XYZ Corporation take the following actions immediately to secure the Metasploitable 2 server and the Windows 7 desktop computer:

1) Close all open ports.
2) Upgrade the Windows 7 desktop to a current version of Windows that is still being updated and patched regularly by Microsoft.
3) Repeat the process of installing updated versions of Windows on every single computer within the XYZ Corporation infrastructure.
4) Conduct an immediate bounds checking of all code that is susceptible to buffer overflows.

Network and Binary Exploitation: XYZ Corporation

Strategic Recommendations

Rameses Security LLC recommends that XYZ Corporation take the following actions to ensure the long term security of their company:

1) Establish a policy that bans open ports unless it's absolutely necessary to have them open for the sake of business continuity.
2) Establish a schedule of regular port scans of the XYZ Corporation network to enforce the policy of no open ports.
3) Establish a policy banning the use of any end of life products including outdated versions of any Windows OS.
4) Require that all software updates and patches be installed by all users as soon as they are deployed.
5) Implement secure programming practices that do not leave openings for buffer overflow.
6) Require regular, either weekly or bu-weekly, bounds checking of all static code that could fall victim to a buffer overflow

Methodology

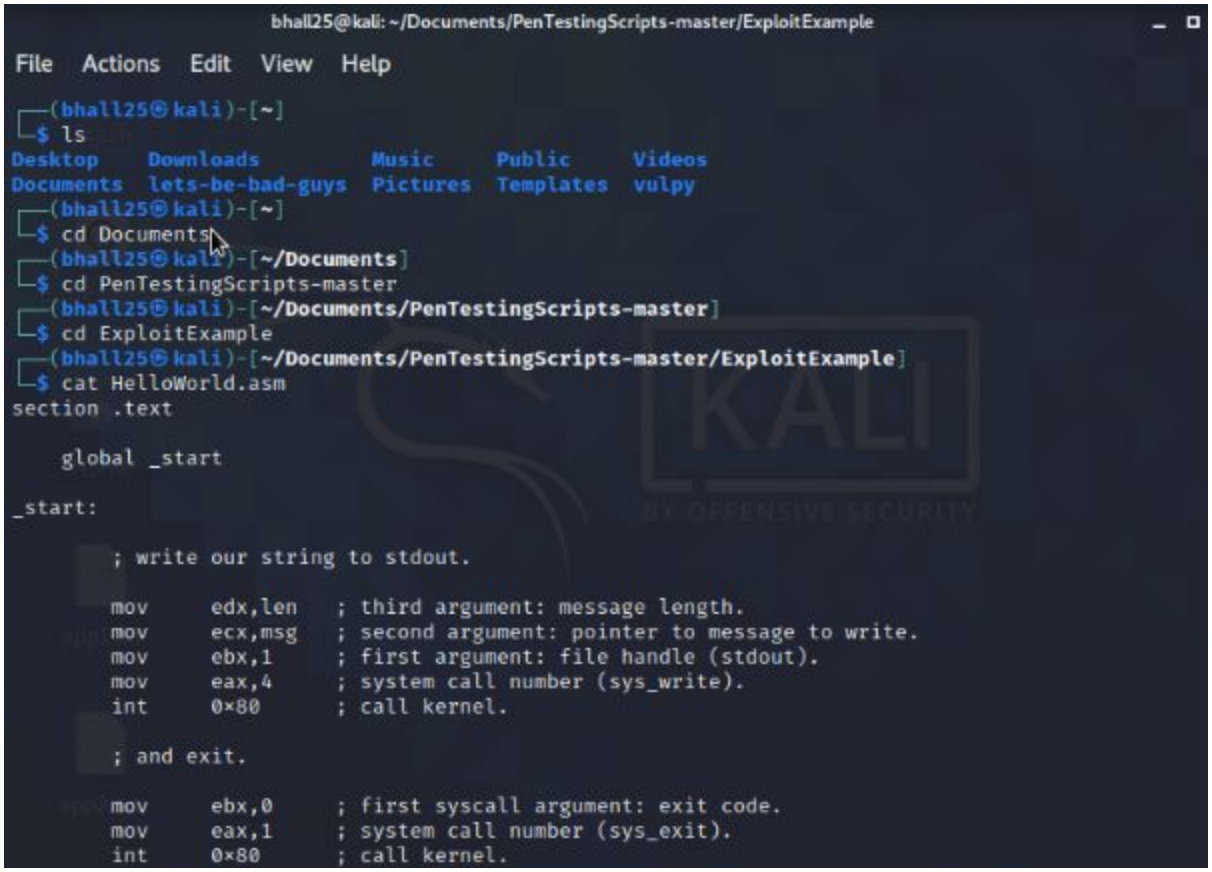
Rameses Security LLC is a leader in penetration testing and general cybersecurity services. We conducted four successful exploits against two devices owned by XYZ Corporation. The devices were the Metasploitable 2 server and the Windows 7 desktop computer.

To execute against these machines we used Kali Linux and its built in tools Nmap and Metasploit. Nmap allowed us to view open ports and Metasploit allowed us to set up and deliver the payloads. To help select which exploits to use we conducted open source intelligence gathering from a few different online sources that are cited in the works cited page at the end of this document.

Network and Binary Exploitation: XYZ Corporation

We also demonstrated how to execute a buffer overflow with custom scripts. Figure 25 documents the custom scripts that we used within Kali Linux to execute the buffer overflow.

Figure 25



Rameses Security LLC then presented XYZ Corporation with the documented results of the exploits. We then presented short term solutions to fix the problems and long term policies that should be adopted to reduce the likelihood of future, successful exploits of the XYZ Corporation systems and network.

Rameses Security LLC appreciates the cooperation and partnership with XYZ Corporation and looks forward to a continued partnership in the future.

Network and Binary Exploitation: XYZ Corporation

Works Cited

1. Lord, Nate. “What is FTP Security? Securing FTP Usage.” What is FTP Security? Securing FTP Usage. September 7th, 2018 *Digital Guardian.* Accessed March 18th, 2021.
2. Samba. “About Samba.” Samba - opening windows to a wider world. Accessed March 18th, 2021.
3. Rapid7. “Samba ‘username map script’ command execution.” Samba "username map script" Command Execution Disclosed: May 14, 2007 module Explore. Accessed March 18th, 2021.
4. OWASP. “Use of Hard-Coded Password.” OWASP Foundation | Open Source Foundation for Application Security. Accessed March 18th, 2021.
5. MITRE. “CVE.” CVE - Common Vulnerabilities and Exposures (CVE). Accessed March 18th, 2021.

.