SIG Security

Basic Network Security

Week of March 07, 2017

Goals:

- Gain familiarity with basic network security.
- Learn about natural address translation (NAT) and intrusion detection/prevention systems (IDS/IPS).
- Exploit network vulnerabilities on both non-protected and hardened networks.

Required:

- Kali Linux – installed or as a VM. See me if you require a recent image.

  *Note: Kali is not *strictly* required, although you will need to install several tools. I would not recommend it!

Approach:

1. We will be exploiting various vulnerabilities on Metasploitable 2, first on a completely open network and again after mitigations have been enabled.

Disclaimer

The information provided herein is intended for educational purposes only. Misuse of the materials presented could lead towards criminal charges. Please refer to any applicable local, state and federal laws that may apply. Any actions taken or resulting from following this guide are the responsibility of the reader and the reader alone. I nor the Association for Computing Machinery at the University of Illinois at Chicago will be held responsible in the event that any of this information is misused.

Please take care and exercise caution: ignorance is not an excuse.

1. Every group is assigned an IP address of an instance of Metasploitable. Your group's IP address is:

2. There are also two Metasploitable instances behind pfSense, an open source firewall/router, one with only NAT enabled and another with NAT and Snort IDS enabled. The IP's for these machines are:

3. Let's start by sniffing out what ports Metasploitable 2 is listening on. If you were not here last semester for our discussion of Nmap, check out the help and try these commands (where $IP is the ip assigned to your group):

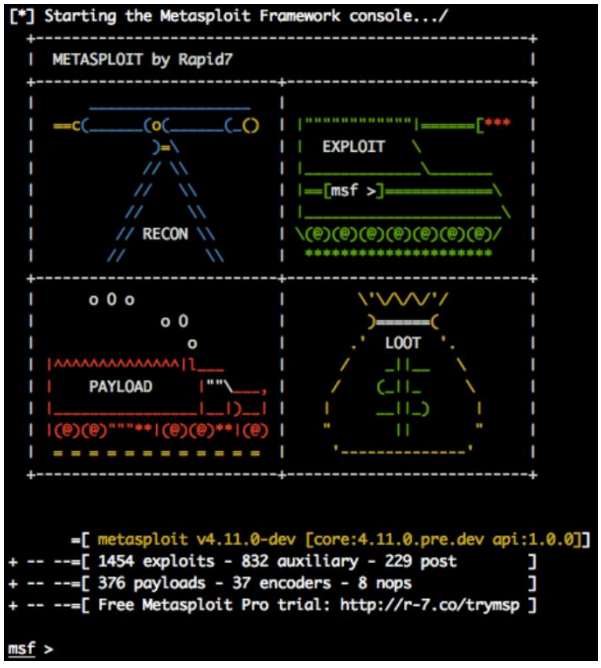
`nmap -h`	Display the help menu.
`nmap -v $IP`	Run the standard scan on the host with verbose output.
`nmap -v -T4 -p1-65535 $IP`	Run a scan on all available ports, but faster!
`nmap -v -T4 -sV -p21-10000 $IP`	Do a service/version scan on ports 21-10000.

   Nmap can also log output in a number of different formats, some of which can be exported to other applications. Use the -oA option to output into the 3 major formats.

4. If you ran the version scan, you'll notice that Nmap can sometimes fingerprint a service to determine its version. Take note of vsftpd version that is running and try doing a search for it online along with the word "vulnerability."

5. If you followed the above advice, one of the first links displayed should be a link to Rapid7's website. Once you've read the description, notice that there is module available in Metasploit. Go ahead and fire it up (this vulnerability is easily exploited without the help of Metasploit, but we'll be using it anyways for demonstration).

6. Once Metasploit has started, you'll be greeted with the following interface:



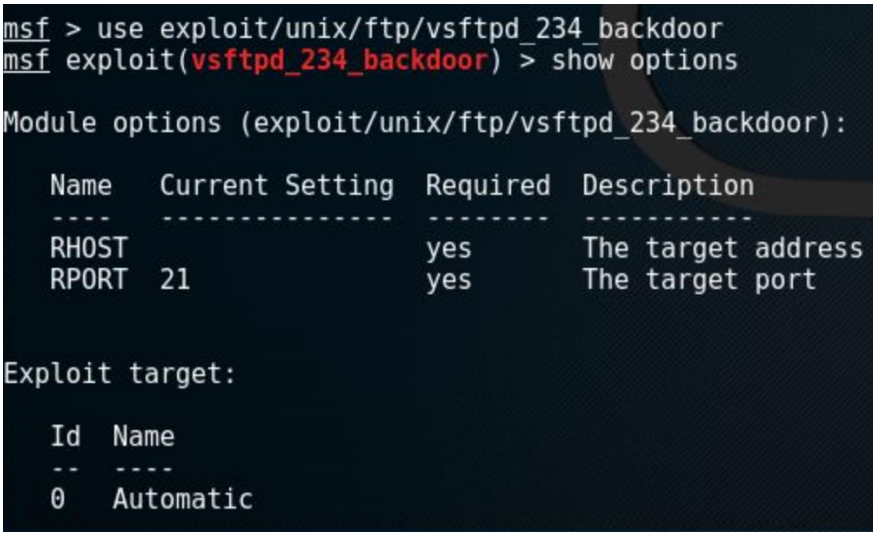
7. Rapid7's website listed the module as "exploit/unix/ftp/vsftpd_234_backdoor". Use the **info** command to get more information about the module:

   ```
   info exploit/unix/ftp/vsftpd_234_backdoor
   ```

8. Load the module using the **use** command:

   ```
   use exploit/unix/ftp/vsftpd_234_backdoor
   ```

9. This module requires some configuration before we can launch it. Use the command **show options** to display the available variables:



10. RPORT (remote port) has a default value of 21 that does not need to be changed. However, we need to set RHOST so Metasploit knows where to send the exploit and payload. Use the **set** command to do this:

    ```
    set RHOST <YOUR TARGET IP HERE>
    ```

11. You can also specify which payload you would like to use. Use **show payloads** to get a listing of compatible payloads. This particular exploit only has one compatible payload which creates a remote shell.

12. Use the **exploit** (or its alias **run**) to run the exploit. Since this exploit takes advantage of a built-in backdoor, it should run without a hitch.

13. You'll notice that after the shell spawns, you're not given the prompt that you're accustomed to. Regardless, if you run the command **whoami**, you'll see that you have root!

14. We won't be going too far in-depth on the capabilities of Metasploit post-exploitation. We will however explore one more vulnerability: weak credentials.

15. We will be using Metasploit again for this. This time we will be using the ssh_login module. Try using the **search** command to find the module and load it up.

16. Read through the options and set the appropriate values. Note that RHOST is now RHOSTS, allowing you to specify multiple targets. However, for this demonstration we will only specify one. We need to specify a list of passwords and a list of usernames to try. We will save some time and specify a user-pass file, which contains a list of usernames with their associated passwords. Let's try out the user-pass combinations that Mirai was using:

    ```
    set USERPASS_FILE /usr/share/wordlists/metasploit/mirai_user_pass.txt
    ```

17. Launch the attack when you are ready. Notice how slowly Metasploit goes through the list: SSH is rate-limited to mitigate such an attack. Luckily (or unfortunately, depending on which side you sit on), Metasploitable has plenty of weak usernames and passwords.

18. Metasploit will automatically create sessions for any combination that worked. Use the **sessions** command to list all available sessions. To interact with one, you can use the **-i** option along with the session id. If you want to return to Metasploit, background the current session with **CTRL-z**.

19. Metasploitable 2 has plenty of other vulnerabilities. If you are interested, try probing around on your own or searching for a guide online. There are plenty of resources available. The next section will cover basic mitigations that can slow down and/or prevent these attacks completely.

Mitigations

1. Metasploitable 2 has a LOT of listening services that probably aren't necessary or required… If you have SSH enabled do you really need ftp or telnet? The more services you have running on any host, the larger your attack surface is. Any services that are not being used should be closed or uninstalled. This would have completely eliminated the vsftpd attack vector for instance.

2. Weak credentials are another issue. Metasploitable 2 had easily guessed usernames and passwords. Since your username and password are supposed to uniquely identify you, it can be extremely hard to protect your system if those credentials can be easily guessed. Do not use overly simple or easy to guess passwords and usernames and implement some rate-limiting solution. For our SSH brute-force example, fail2ban would have temporarily locked us out after a small number of attempts. You can also use two-factor authentication or other means of authentication, such as a public-private key pair.

3. Outdated software is a huge security risk. Metasploitable 2 is vulnerable to numerous well-known and highly publicized vulnerabilities, such as Dirty CoW and ShellShock. Many of the services that Metasploitable 2 is running would be (mostly) safe if they were simply upgraded to the most recent version.

4. Metasploitable 2 does not have any sort of firewall. We generally take this for granted on Windows, but Linux does not have any firewall rules set by default in the most basic installations. For example, a firewall could have prevented all traffic from entering except for ports 22 and 80.

Demonstration

1. There are two-instances of Metasploitable 2 that have some mitigations in place: one is behind a NAT enabled router, and another is protected by Snort IDS. We will begin with the instance behind NAT.

2. NAT stands for natural address translation (in the Linux world, this is often known as masquerade). It allows a host to "share" its IP address on one of its interfaces with another host on the same network. NAT acts as a "natural" firewall, because any traffic that is destined for the internal network must first be port forwarded. In this example, pfSense is connected to the ACM network and an internal network shared only with Metasploitable; the internal network is unreachable except via pfSense. pfSense is configured to forward all traffic on ports 22 and 80 to Metasploitable and block all other traffic. Try to repeat the steps we followed earlier starting with the Nmap scan. What happens? Can we gain root through vsftp?

3. The other machine is protected with Snort. I have intentionally forwarded all traffic from pfSense to Metasploitable in order to demonstrate Snort's capabilities, meaning that there is no firewall in place. Try repeating the steps as above. What happens? There are some vectors that Snort does not have rules in place for. If you SSH using one of the credentials you gained above, you should be able to connect. What happens if you try to upgrade this shell to a meterpreter session?