Justin Strelka

Exploit\_metasploitable

6/29/20

**Ethically Exploiting Metasploitable2**

**Table of Contents:**

Abstract…………………………………1

Motivation…………………………….1

Introduction………………………….2

Method/Measurement………….2

Results…………………………………..4

Conclusion……………………………10

Bibliography………………………….11

Reflection……………………………..11

**Abstract:**

This paper begins with a brief explanation of why it is important to practice hacking your own machines. Next there is an overview of the contained environment we will be using to conduct exploitations against Metasploitable2. Following is an introduction to what Metasploit is along with a step-by-step guide on how to install Metasploit on Kali Linux. After the Metasploit installation guide there is a demonstration of how to deploy Metasploit and use the tool to ethically exploit a specified host. This paper contains a demonstration on four different exploits present on the Metasploitable2 host. Lastly there is a recap of the exploits that were exploited and what we can learn from these exploits when hardening our production systems.

**Motivation:**

This paper is intended for students with knowledge of a command line interface that want to learn more about how systems are exploited, and the tools used to do so. Upon reading this paper you will be able to ethically exploit a system within a contained environment and understand how to identify exploits within a specified host. You will be able to describe and understand what Metasploit is and how it can be used against our production systems. You will gain an understanding of how a malicious individual may try to exploit a production system. Most importantly you will have an appreciation for practicing your skills with permission against known hosts to continue developing an awareness for how to defend against pre-documented and future system exploits.

**Introduction:**

Understanding the vulnerabilities of a production system will give us an edge against future exploits and allow us to mitigate or eliminate the potential for exploits. We can learn valuable skills by attacking and exploiting replicas of our production systems in a contained environment. When we successfully attack replicas of our production systems, we better understand how to mitigate the potential for a malicious individual to also successfully conduct an attack. Understanding the tools that a malicious individual will most likely deploy to discover, plan, and later attack a production system is essential knowledge for making educated cybersecurity decisions.

**Method/Measurement:**

Before we begin attacking known hosts, we must first ensure we have permission to conduct such operations. For the purposes of this paper, permission will be a non-issue since we will be conducting our penetration testing within a contained environment that we stood up previously. The contained environment we will be testing within consists of a Type II hypervisor (Virtual Box) that will host our virtual machines to be used for penetration testing.

The first virtual machine (VM) we stood up is a Kali Linux distribution obtained from the following link: <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>. The environment we established previously used the Virtual Box compatible 64-bit 2020.2a disc image. The Kali Linux VM will be our machine used to conduct our penetration testing and attacking. We previously installed OpenVAS on the Kali Linux VM which is a vulnerability scanning tool used for discovering vulnerabilities that we can later attempt to exploit on a given host.

The second VM we stood up previously runs on the opensource vulnerable framework Metasploitable2. The framework and disc image for Metasploitable2 can be downloaded from the following link: <https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>. The Metasploitable2 framework has been designed to have known vulnerabilities built into the OS and will be used to target our penetration testing and attacking against. Because Metasploitable2 does have known vulnerabilities built into the OS we need to ensure that the VM will never be visible to the global network. We can do this by changing the network adapter on the VM from the default NAT adapter to the Host-only adapter setting.

Now that we understand the purpose of each element within our contained environment, we can implement the final preparations for penetration testing and attacking. Begin by booting up our hypervisor of choice VirtualBox on the host machine. Once VirtualBox has finished booting we will then boot our Kali Linux VM which we will refer to as “kali\_attack” moving forward. Ensure that your kali\_attack VM has the default NAT network adapter selected so we can access the internet. We will install Metasploit on kali\_attack which is an exploitation and vulnerability validation tool that helps organize and divide penetration testing into manageable sections. [1]

Open a browser on kali\_attack and navigate to the following link to download the Metasploit framework: <https://www.metasploit.com/download>. Click the download link for the Metasploit Framework Open source version. The link will route you to a GitHub repository where you can select the installer type. We are working on our kali\_attack VM which is a Linux distribution so we will select the Linux/Mac OS X installer. Open a terminal on kali\_attack and copy/paste the provided “curl” command into the terminal as shown in **Figure 1.** You will be prompted to enter your sudo password and the install will automatically begin and complete.

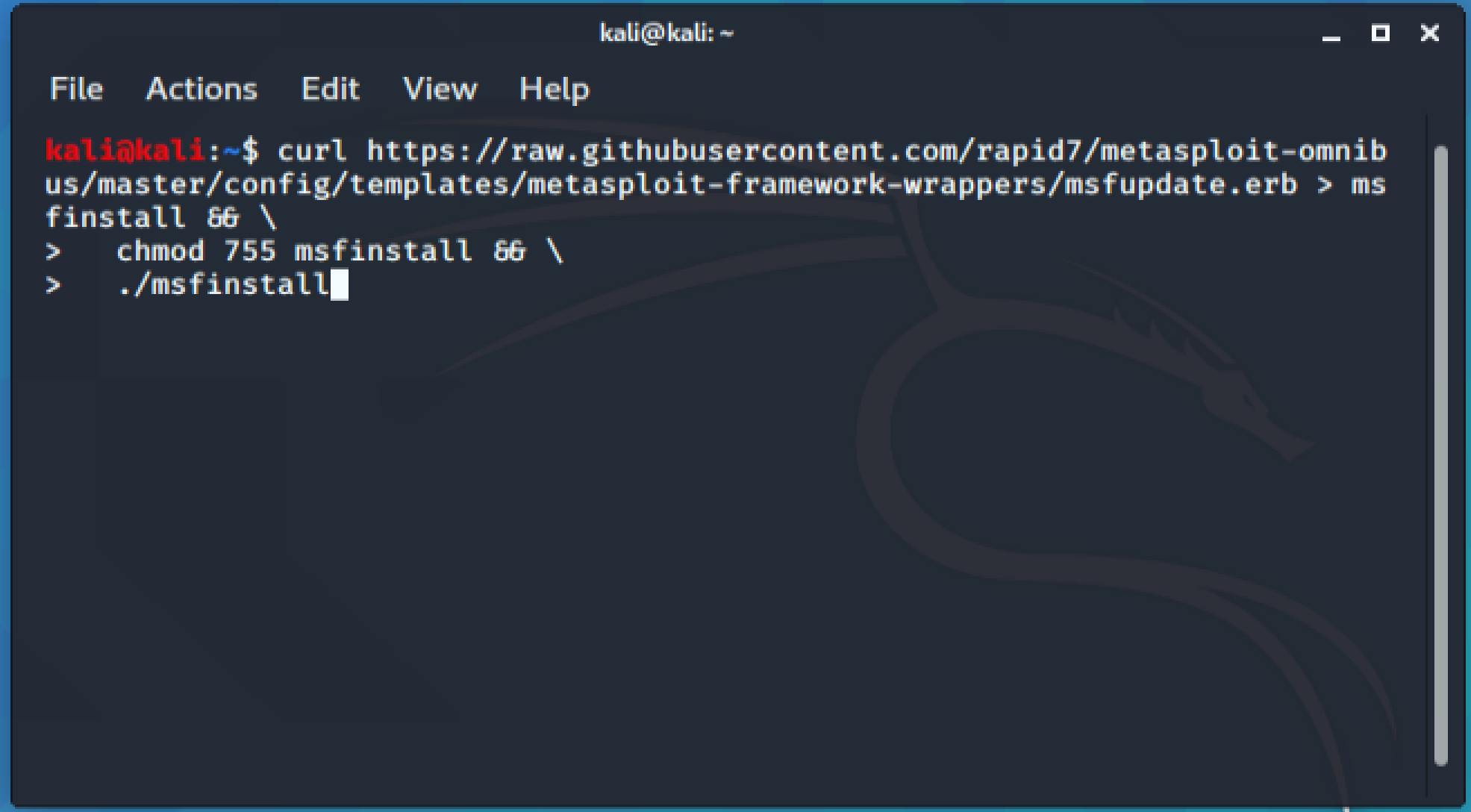


Figure : Obtaining Metasploit on kali\_attack VM.

Once successfully installed we will need to complete a few more configuration steps to get Metasploit up and running with database support. The following link outlines the steps necessary to connect Metasploit to PostgreSQL: <https://www.kali.org/docs/tools/starting-metasploit-framework-in-kali/>. Complete the steps outlined in the Kali Metasploit document shown in **Figure 2,** and Metasploit will be fully functional. Next we will want to ensure we install all the support gem files by running the command “bundle install” within the msfconsole. [2] Finally we will run the command msfupdate within the msfconsole to ensure we have all the latest exploits available to run penetration tests with.

A screen shot of a social media post

Description automatically generated

Figure : Starting Metasploit framework with PostgreSQL support.

Now that we have Metasploit fully configured on our kali\_attack VM we will want to remove the VM from the global network by changing the network adapter setting back to “Host-only”. To do so power down the VM and navigate to the network settings for kali\_attack in VirtualBox. Once the network adapter setting has been changed accordingly power up the kali\_attack VM again. We will now also boot our second VM metasploitable2 which we will be using to conduct our penetration testing against. As previously mentioned double-check that the network adapter for the metasploitable2 VM is set to “Host-only” to hide the vulnerable framework from the global network and protect your host machine.

Once metasploitable2 has booted login using the provided msfadmin credentials. After authenticating as the msfadmin user we can now obtain the IP address of the metasploitable2 VM using the ifconfig command. The IPv4 address for the metasploitable2 machine in my contained environment is 192.168.56.103. Be sure to log your metasploitable2 IP address correctly as this is the host IP address we will be exploiting in the coming steps. You do not want to accidentally run an exploit against an unauthorized target due to a mistake in the targets IP address.

We will now boot the Metasploit framework on kali\_attack by running the command msfconsole. Once the framework has booted, we can begin our target discovery phase of penetration testing. We already know our targets IP address so we will start by discovering the open and available ports on our target metasploitable2 machine. To do so we use the nmap utility inside of the msfconsole by running the following command “sudo nmap -sV -O 192.168.56.103”. As shown in **Figure 3** the -sV flag will detect the services and their version running on the target IP and the -O flag will tell us information about the operating system of the target device. We can see the OS of the target does indeed have an Oracle Virtual Box MAC address and the service info confirms that the host is indeed our metasploitable2 VM.

A close up of text on a black surface

Description automatically generated

Figure : Output of nmap ran against our metasploitable2 target VM.

A circuit board

Description automatically generatedMetasploit has built in scripts called modules that can be used to automatically test various exploits. [2] To find pre-made modules in Metasploit we use the search command within the msfconsole. The search command allows you to specify a variety of filters to find modules for known exploits. As we can see in the nmap output the target is running MySQL on port 3306 which has available pre-built modules in the Metasploit framework. We can search for these modules using the following command inside the msfconsole “search <option>:<data>”. As you can see in **Figure 4,** we have 17 pre-defined MySQL exploits available within the Metasploit module database.

Figure : Results of search name:mysql command within msfconsole.

**Results:**

A screen shot of a computer

Description automatically generatedWe can then use the “info <module\_path>” command to get information regarding a specific module and the requirements to deploy the module. As we can see in **Figure 5** the “auxiliary/admin/mysql/mysql\_sql” module info is displayed to the console and the various options available to use are reported. To use this exploit, we must be able to authenticate into MySQL on the target device. There is a module to attempt to login with standard credentials against a MySQL instance. In **Figure 4** we can see a module by the name of “auxiliary/scanner/mysql/mysql\_login” which is a MySQL login utility. We can deploy this module to obtain the authentication credentials needed to conduct the MySQL command exploit viewed previously.

Figure : Info display of module with options available to configure exploit.

A close up of a keyboard

Description automatically generatedTo run the MySQL login exploit we must first configure the module as follows. Inside the msfconsole run the command “use auxiliary/scanner/mysql/mysql\_login”. Next we can view the configuration options by running the command “show options”. We can configure the login credentials used for a brute force attempt. We will be using the default authentication attempts of standard credentials and empty password as well. We know from our nmap scan in **Figure 3** that MySQL is listening on port 3306 which is the default port and is already configured in our module options. Next we must set our RHOSTS variable to the target we wish to deploy against. In this case it is our metasploitable2 VM and we can set the option using the following command “set rhosts 192.168.56.103”. After we can confirm that the option was indeed set correctly by viewing the show options output again. Once we have configured the module, we can now deploy it against the target host. To deploy a module with the specified configurations we must remain in the use section of the console and enter the “run” command. As we can see in **Figure 6** the module was successful logging in as root with an empty password; meaning the root user does not have a configured credential. This will be our exploit entry to obtain control of the database and run commands against the database through the “auxiliary/admin/mysql/mysql\_sql” module.

Figure : Output of the deployed MySQL login module.

A screen shot of a computer

Description automatically generatedExit out of the current module being used by entering the “back” command in the console. Next we will want to reload the “auxiliary/admin/mysql/mysql\_sql” module and configure it accordingly. Use the “show options” command to display the available configuration options for the module. We will set the password to blank, rhosts to 192.168.56.103, rport to 3306, username to root, and lastly set sql to the sql command we wish to run. We will set the sql field to “show databases” for discovering what data may be available for us to extract or corrupt. **Figure 7** shows the options set and the output of deploying the module along with the return of the SQL command ran against the metasploitable2 VM. As you can see, we have successfully exploited the MySQL database and have access to extracting/manipulating data within it.

Figure : Successful exploit of MySQL database on Metasploitable2 VM.

If we look back at **Figure 3** we can see that vsftp version 2.3.4 is listening on port 21. VSFTP stands for very secure file transfer protocol and if we can gain access to it we will be able to extract or ingest files into the target that can be used to further our control of the machine. We will begin by searching the Metasploit database for a module that can exploit the vsftp service. Enter the command “search name:vsftp” to get a return of all modules that have vsftp in the file path. We can see there is a single module that is listed named “exploit/unix/ftp/vsftpd\_234\_backdoor” which can provide us with command execution capabilities. First, we will ensure that the version numbers of the protocol and modules align by typing the “info exploit/unix/ftp/vsftpd\_234\_backdoor” command. In the name of the module and the description section shown in **Figure 8** we can confirm that the version number of the exploit does indeed align with the target service in **Figure 3**.

A screenshot of a computer screen

Description automatically generated

Figure : Output of the info command for vsftpd234 module.

Next we will begin configuring the module to conduct the exploit against the specified target. To do so we must first load the module with the “use exploit/unix/ftp/vsftpd\_234\_backdoor” command. Once the module is loaded we can explore the options available for configuration using the “show options” command. There will be two options available to us for configuring the module. The first will be the RHOSTS field which will be set to the IP of our target using the “set rhosts 192.168.56.103” command. The second field is the RPORT field which will be set to the port vsftp is listening on using the “set rport 21” command. Once we have the module configured we will deploy the module and attempt to exploit the metasploitable2 VM. To deploy the module enter the “run” command in the console.

The exploit will successfully open an interactive shell terminal where I can run any standard shell command. As shown in **Figure 9** the exploit worked and I can access a standard bash shell as root. From here I can remove, inject or even change permissions to any file on the metasploitable2 target. This exploit is extremely effective as I have root access on the target machine. Having root access will allow me to access any part of the system without bounds. I could also utilize this machine as a pivot to attack another machine of my choice on the network or if I accessed the machine from outside the network it means that the machine can attack other machines outside of the local network.

A screenshot of text

Description automatically generated

Figure : Successful exploit against vsftpd234 on metasploitable2 VM with root access.

Next we will return to **Figure 3** and pick the next service to attempt an exploit against using the Metasploit framework. We can see ssh is open and listening on port 22 on the metasploitable2 target VM. We will use the command “search name:ssh” which returns 38 different modules in the Metasploit database. We will deploy the module at index 13 “ssh\_login” against the metasploitable2 VM.

We must first load the “ssh\_login” module by entering the “use ssh\_login” command. We can then begin configuring the module by listing the available options with the “show options” command. As we can see in **Figure 10** the module allows us to specify a file of our choosing for usernames to attempt to authenticate with. Kali has a built-in common word lists located at “/usr/share/wordlists/dirb/common.txt”. [3]

This will work perfectly for trying to brute force a common login and password. Use the “set user\_file /usr/share/wordlists/dirb/common.txt” command to point Metasploit at the file for brute force attempts. Next we will assume a user that has a common login name may also have an identical password. To check against this we can use the command “set user\_as\_pass true” to check identical username and password combinations. To prevent checking against every credential combination in a file we can set the STOP\_ON\_SUCCESS field to true once a successful login has been made. Finally we must set the rhosts field to 192.168.56.103 with the “set rhots 192.168.56.103” command. **Figure 10** displays the configurations for the ssh\_login module.

In **Figure 11** we can see that the module was successful at finding an SSH username and password combination of “user” for both credentials. I then tested the combination and was able to successfully ssh into the metasploitable2 VM with “user” as both username and password. From here I would be able to access anything the user “user” has privileges to and possible gain access to other user accounts if there are other exploits available. I could also pivot from this machine to conduct an attack against another machine.

A screen shot of a computer

Description automatically generated

Figure : Display of the configurations for the ssh\_login module.

A screenshot of a computer

Description automatically generated

Figure : Display of successful username and password exploit against ssh on metasploitable2 VM.

For our final exploit against a service running on metasploitable2 we will target the postgresql service listening on port 5432 on our metasploitable2 VM. We can use the postgres\_login module to try and brute force a successful login. To configure the “postgres\_login” module we need to enter the host and configure the common username/password file to be used for attempting logins. We will set the following settings: rhosts to 192.168.56.103, blank\_passwords to true, stop\_on\_success to true, verbose to false. If we wanted to conduct the full exploit we would be setting stop\_on\_success to false and verbose to true for keeping track of all successful and unsuccessful combinations. We can see the configured module in **Figure 12** and the successfully exploited credentials (postgres:postgres@template1) in **Figure 13.**

A close up of a screen

Description automatically generatedA screen shot of a computer

Description automatically generated

Figure : Postgres\_login module configured for metasploitable2 host.

Figure : Successful credential for postgresql user and password.

**Conclusion:**

We have seen how easily some basic exploits can be carried out using the opensource tools that are available to both ethical and unethical hackers. Tools such as NMAP, Metasploit, and OpenVAS make it very easy for unethical hackers to exploit known vulnerabilities. We have seen an example of an exploit that did not require any known data about a target other than the IP address, and knowledge of a running service’s version. This was shown in the vsftp v2.3.4 attack where we were able to access a fully functioning root shell just by deploying the Metasploit module. These types of attacks require us to keep our service versions up to date with new releases that have patched vulnerabilities. We have also seen brute force attacks where weak passwords allowed us to access a service quickly. These attacks are made possible by system users not being well educated on the dangers weak/standard passwords can have on a company asset such as a database. Moving forward in securing our systems we must be aware of the tools that unethical hackers can and will use against our systems. Practicing with these tools in a contained environment gives us an edge on security to stay one step ahead of our unethical adversaries.

**Bibliography**

[1] <https://metasploit.help.rapid7.com/docs>

[2] <https://metasploit.help.rapid7.com/docs/msf-overview>

[3] <https://security.stackexchange.com/questions/195663/username-and-password-files-in-kali>

**Reflection:**

Writing this paper, I realized how easy it is for unethical hackers to gain access to a system that has out of date services running on it. I intern at National Renewable Energy Laboratory (NREL) and have the privilege of interacting with the HPC system Eagle. I have a new appreciation for our login scanners that continuously scan services on the network for standard passwords. Users are encapsulated by privileges however if a malicious user were to gain access to an account, they may be able to escalate their foothold or pivot to attack other machines both within and outside of the network. I found the vsftp v2.3.4 exploit highly unsettling that with minimal information about a system a malicious user can obtain root access. When other backdoors such as the vsftp v2.3.4 exploit are found entire systems can be compromised in moments. When an entire system becomes compromised overnight in the private sector catastrophic events can occur due to the downtime, cleansing, and reputation costs. I find it ironic (less the amusing part) that companies are spending vast amounts of resources on cybersecurity and yet a single weak link from an outside application or service provider can end the life of a company.