**Lab Experiment 10**

**B. Tech CSF-CSE Semester III Course: Physical and IT Security**

**By: Dr. Gopal Singh Rawat**

**Lab Objective: Learn Nessus Essentials for Vulnerability Scanning**

**Note:** only use this version for personal projects at home on your Metasploitable2 OS only, using it in a business environment is a breach of Tenable’s licensing.

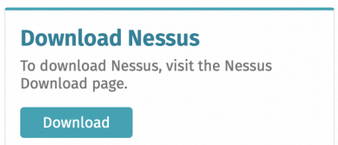
**Tools:**

* **Kali Linux**
* **Nessus Essentials**

**Steps to perform:**

**Step 1: Use Kali Linux Virtual Machine to install Nessus Essentials**

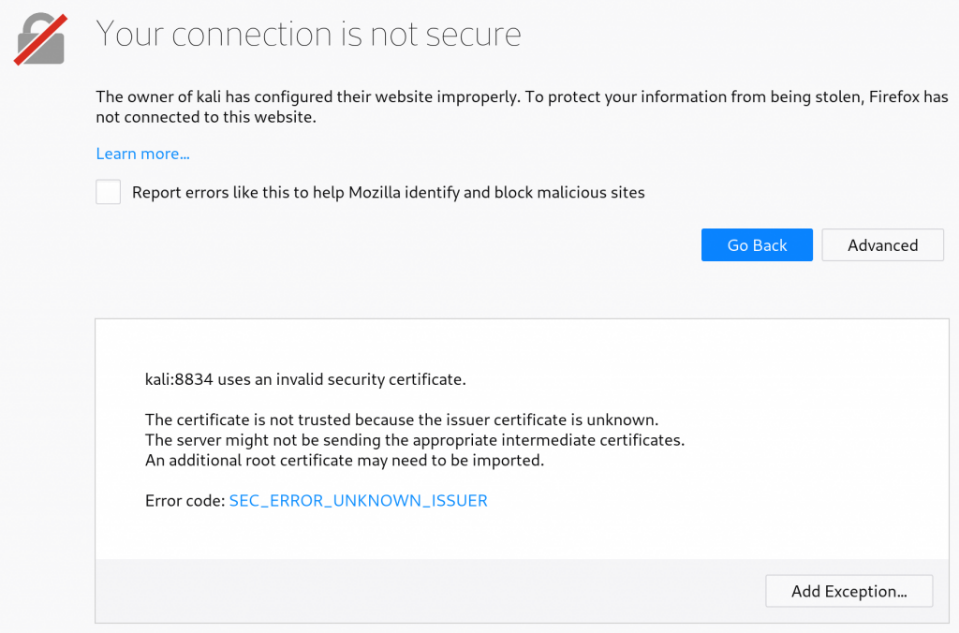
* Visit [**https://www.tenable.com/products/nessus/nessus-essentials**](https://www.tenable.com/products/nessus/nessus-essentials) and register for an activation code and on the next page 🡪 right-hand side 🡪 download Nessus.



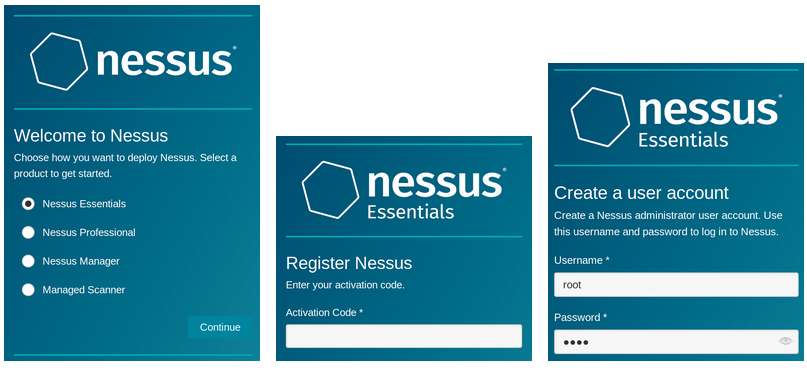
* If you can’t find that box, then visit [**https://www.tenable.com/downloads/nessus**](https://www.tenable.com/downloads/nessus) and download the right version for your host system – it should be **Nessus-X.X.X-debian6\_amd64.deb**
* In a terminal, navigate to the directory you downloaded the file to (Most likely **/root/Downloads** or **/(UsernameHere)/Downloads**) and install Nessus from the .deb file as

**# sudo dpkg -i <Nessus .deb file name> (run as super user)**

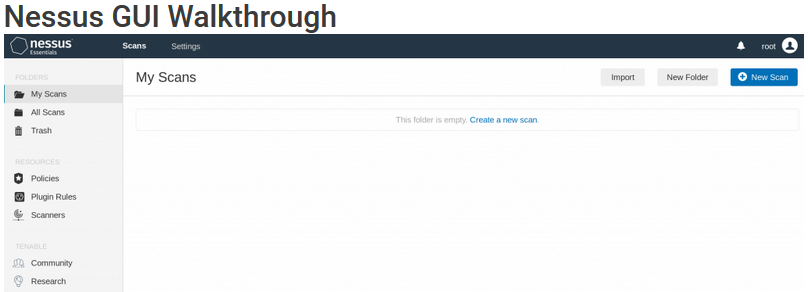
* Next you’ll need to run the Nessus service (also referred to as the Nessus Daemon), which starts everything up.
* Use the command **service nessusd start**, and after a few seconds, go to your browser, and visit [**https://kali\_Machine:8834**](https://kali_Machine:8834)  🡪 this is the local web GUI for your Nessus.



* If you’ve previously used Nessus, you may get an error stating you have a corrupt database. To fix this, you need to kill the service using **service nessud stop**, remove all Nessus files, download the latest version, and install it again.
* To delete all files for a clean re-install, use the following command in the terminal: **rm -rf /opt/nessus** (recursively remove everything in the Nessus directory, **opt/nessus**).
* Once you get to the site, you should now be asked what product you want to use, select Nessus Essentials. You can either register here, or if you did it earlier.

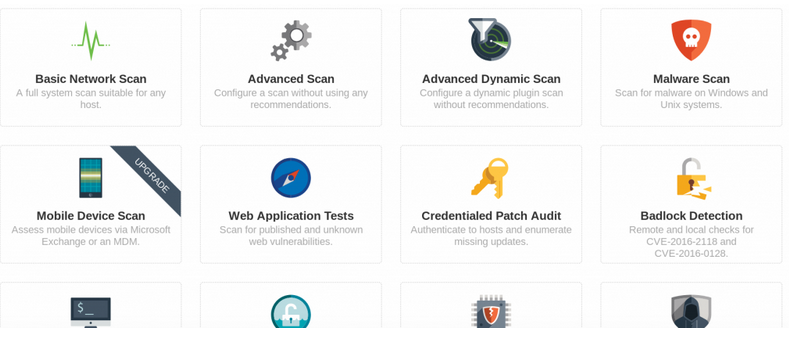


* Now Nessus will download plugins and other crucial files that it needs to function properly, so let it complete (this may take a LONG time, so get up, move around, and come back to it).
* Once that’s done, you’ll be presented with the Nessus dashboard. From here we can launch scans, create policies, review plugins, and more.

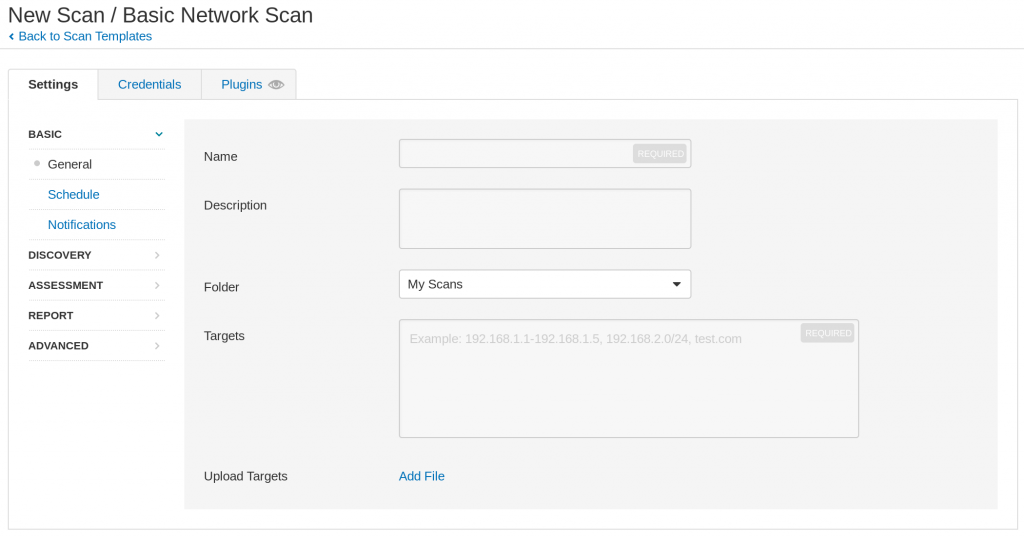


**Step 2: Scanning With Nessus**

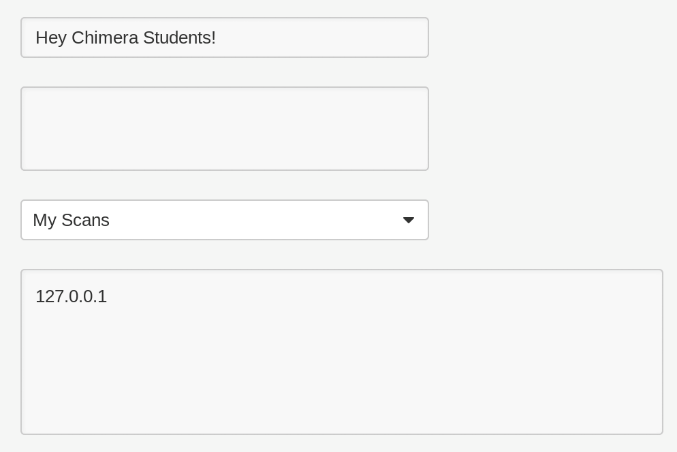
* Next we’re going to perform a simple scan of own our machine to demonstrate how scans work, and what the results look like.
* Head over to the Policies tab on the left, and click “Scan Templates” in the description text.
* Choose from a list of pre-defined templates that can be used for specific actions, such as vulnerability scanning, and host discovery. We will be using the Basic Network Scan in this session.



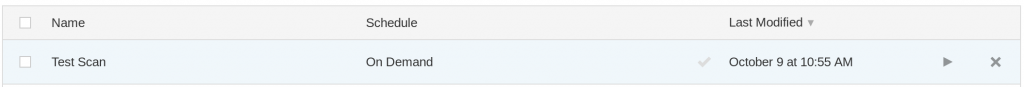
* After clicking on our scan template, we’ll be able to customize the settings for this specific scan.
* Take a look at all of the settings you can change, as well as the Credentials tab, and Plugins tab.
* They allow the scanner to log into the system, and collect much more valuable information, as opposed to being locked out and only being able to collect surface information.
* Companies will usually run credentialed scans internally to get the most valuable information, whereas non-credentialed scans give an unauthenticated attacker’s view of the network.



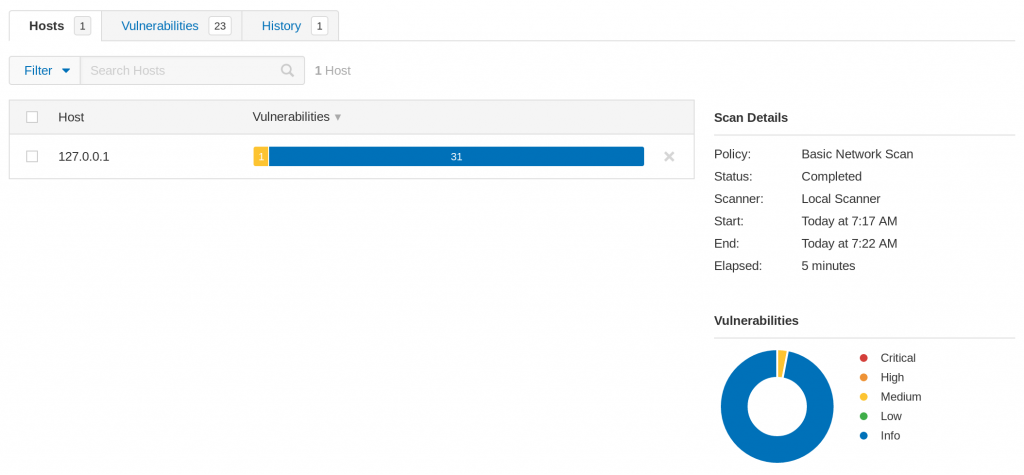
* For this session, you can name the scan anything you like, and we want to enter the localhost (127.0.0.1) as the target.



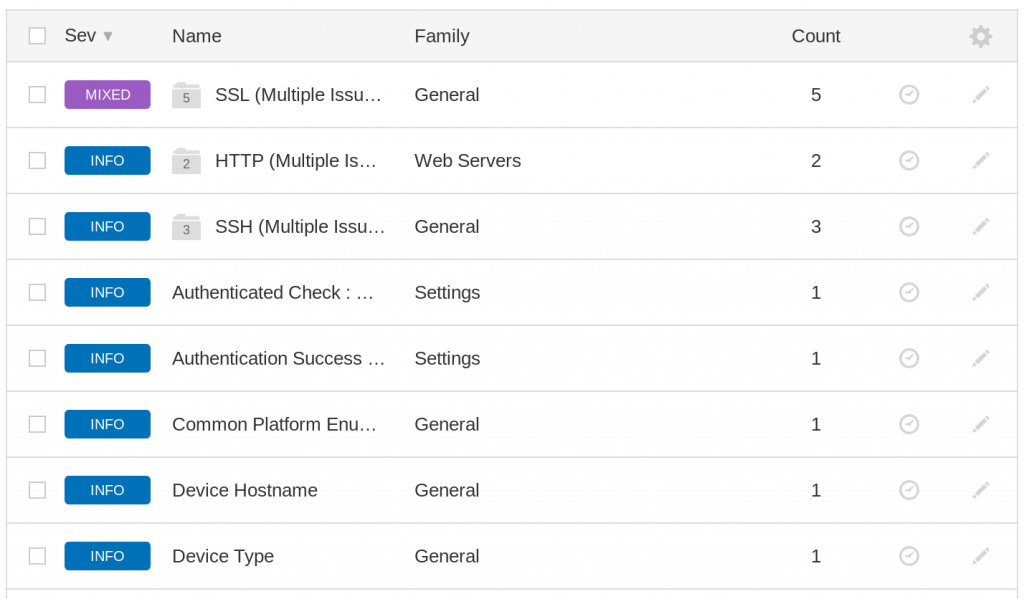
* Click Save in the bottom left-hand corner, and you’ll be taken to the “My Scans” page.
* From here, we can launch our scan by pressing on the play icon to the right.
* Once clicked, the scan will get to work. Once it’s finished, a tick will appear, and we’ll be able to take a look at the results.



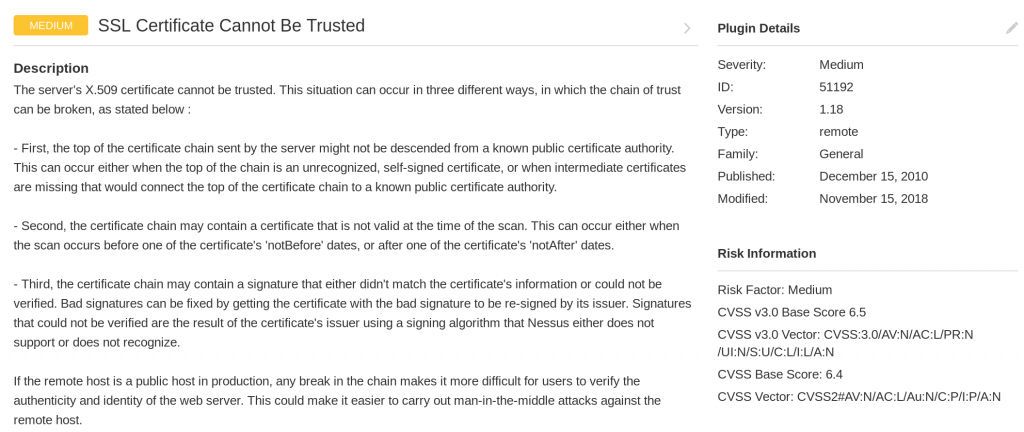
* This is the results pane and provides us with all of the information the scan collected. On the left we have a list of hosts scanned, along with a summary of any vulnerabilities discovered.
* This would be full of different hosts if we were scanning an entire network and is arranged by criticality of vulnerability by default.
* On the right we have the Scan Details, and below it we have a donut chart for the security issues identified.
* Click on the Vulnerabilities tab to see exactly what the scanner identified.



* In this example, we don’t expect to see any major vulnerabilities. Here can see there some issues with SSL (this is because of the security exception we had to allow earlier when trying to open Nessus), and some informational issues.
* You can click on any of these rows in order to get more information and see how to fix them.
* Click on the SSL issues, then click on the medium-rated SSL issue.



* This page shows us a description of the issue, the plugin that was used to detect it, and risk information. Below the description is a solution for how to address the issue.
* When contacting system owners after a scan, it is good practice to attach an export of the scan as a PDF, but also provide a concise summary in an email with a quick overview of the hosts, any issues, and how to fix them.



**Lab Practice:**

* **Perform Nessus Essential Scan for your local machine, take out the Vulnerability Report and patch the OS. Then again run the Nessus Scan, take out the second report. Compare the reports.**
* **Answer the following:**
  + **Which company created Nessus?**
  + **Under Scan Templates in Nessus, there is a scan for what type of Ransomware?**
  + **When creating a new Plugin Rule, what 4 fields do you need to enter?**
  + **Is there a scan template specifically designed for mobile devices?**

**OR**

* OpenVAS 8.0 (<https://www.kali.org/penetration-testing/openvas-vulnerability-scanning/>)

**OpenVAS**, an endpoint scanning application and web application used to identify and detect vulnerabilities. It is widely used by companies as part of their risk mitigation solutions to quickly identify gaps in their production and even development servers or applications. This is not a complete solution, but it can help you fix common security vulnerabilities that may not be discovered.

1. OpenVAS on Kali Linux to perform Vulnerability Scanning

**Objective is to**

* Vulnerability Scanning
* Learn DIRB Tools and Techniques

**Lab Environment**

To carry out the task you need:

* Windows/Linux OS & Wireshark Tools
* Detailed set of activities has been uploaded to Blackboard

**Theory:**

A path traversal attack is also known as “directory traversal” aims to access files and directories that are stored outside the web root folder. By manipulating variables with reference files with “dot-dot-slash (…/)” sequences and its variations or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system including application source code, configuration, and critical system files.

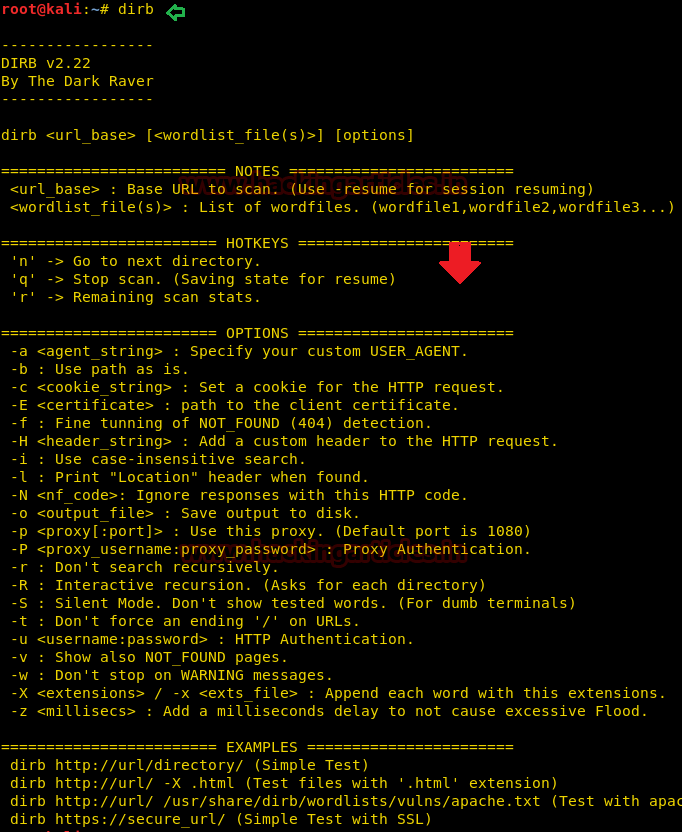
DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary-based attack against a web server and analyzing the response.

It comes with a set of preconfigured attack wordlists for easy usage, but you can use your custom wordlists. Also, DIRB sometimes can be used as a classic CGI scanner, but remember is a content scanner, not a vulnerability scanner.

The main purpose is to help in professional web application auditing. Especially in security-related testing. It covers some holes not covered by classic web vulnerability scanners. DIRB looks for specific web objects that other generic CGI scanners can’t look for. It doesn’t search vulnerabilities, nor does it look for web contents that can be vulnerable.

Tool DIRB is built-in Kali Linux. Open the terminal and type the following command to get an overview of the tools included in the package:

|  |
| --- |
| dirb  -a <agent\_string> : Specify your custom USER\_AGENT.  -c <cookie\_string> : Set a cookie for the HTTP request.  -f : Fine tunning of NOT\_FOUND (404) detection.  -H <header\_string> : Add a custom header to the HTTP request.  -i: Use case-insensitive search.  -l : Print “Location” header when found.  -N <nf\_code>: Ignore responses with this HTTP code.  -o <output\_file> : Save output to disk.  -p <proxy[:port]> : Use this proxy. (Default port is 1080)  -P <proxy\_username:proxy\_password> : Proxy Authentication.  -r: Don’t search recursively.  -R: Interactive recursion. (Asks for each directory)  -S: Silent Mode. Don’t show tested words. (For dumb terminals)  -t: Don’t force an ending ‘/’ on URLs.  -u <username:password> : HTTP Authentication.  -v: Show also NOT\_FOUND pages.  -w: Don’t stop on WARNING messages.  -X <extensions> / -x <exts\_file> : Append each word with this extensions.  -z: Add a milliseconds delay to not cause excessive Flood. |



**Utilizing Multiple Wordlist for Directory Traversing**

The above attack works by using the default wordlist files common.txt, but we can change this word list and could select another wordlist for directory traversal. You must follow the following path to view all available wordlists.

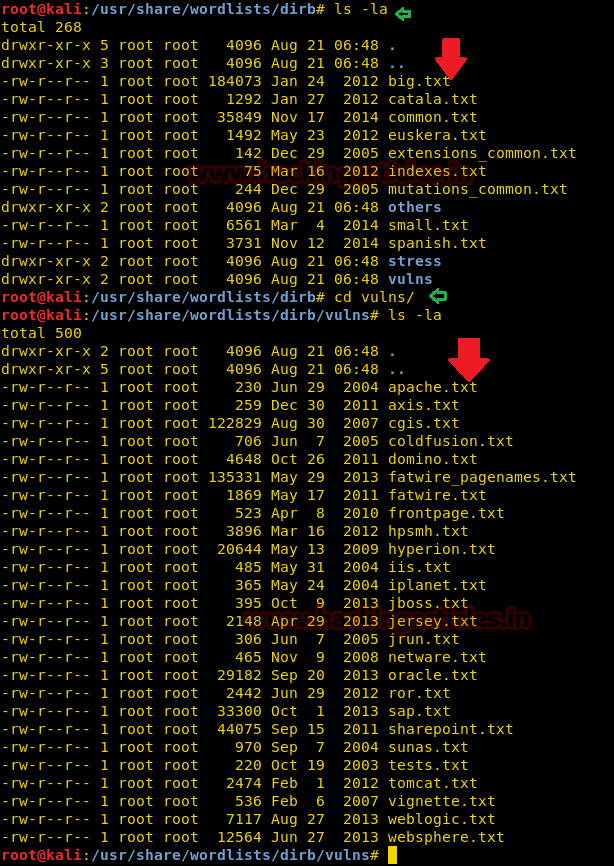
cd /usr/share/wordlists/dirb

ls –la

cd /usr/share/wordlists/vulns

ls -la

You can see from the image below that there are so many text files as wordlist; we can use them as required.

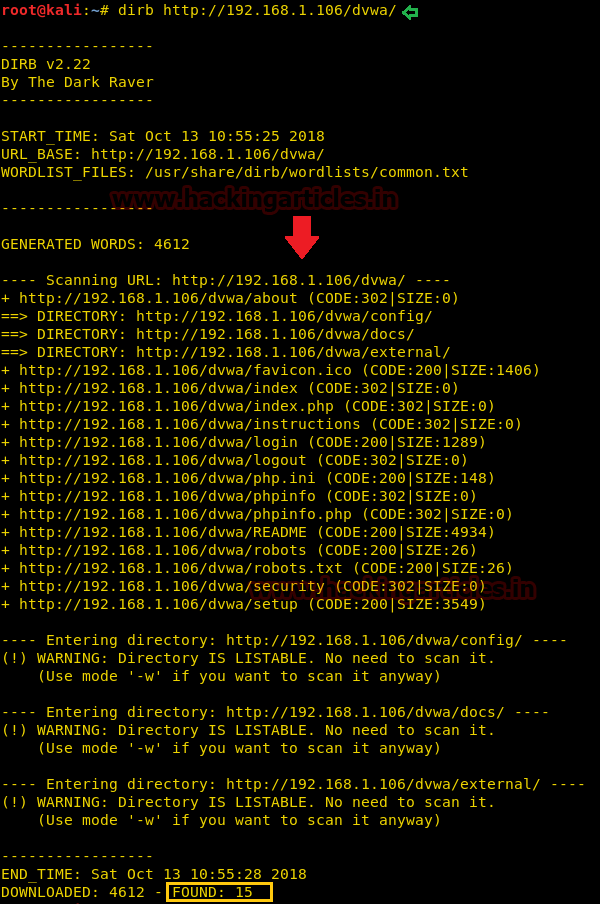


**Default working of Dirb**

In this attack the common.txt is set as a default word list for directory traversal, the protester can use the following command. Open the terminal and type the following command to start the Brussels Directory attack.

dirb http://192.168.1.106/dvwa/

Using the common.txt file, the DIRB returns the enumerated directories found within the target URL as shown in the below image.

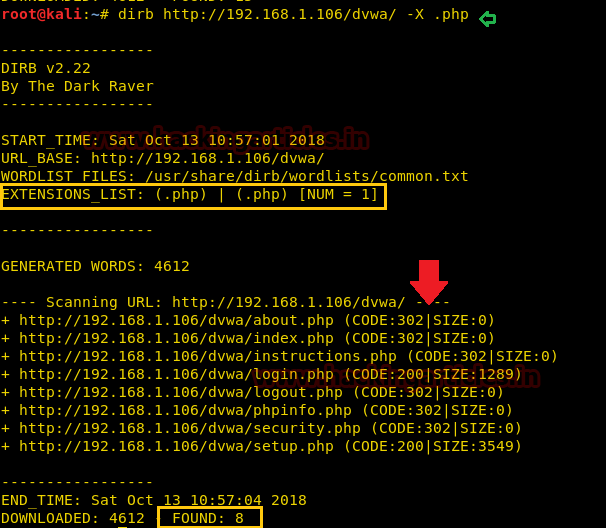


**Enumerating Directory with Specific Extension List**

There are a lot of situations where we need to extract the directories of a specific extension over the target server, and then we can use the -X parameter of the dirb scan. This parameter accepts the file extension name and then searches the given extension files over the target server or machine.

dirb http://192.168.1.106/dvwa/ -X .php

The above command will extract all directory path related to php extension as shown the following image.



**Save Output to Disk**

To the record maintenance, better readability, and future references, we save the output of the dirb scan onto a file. To this, we will use the parameter -o of the dirb scan we can save the output of the dirb scan in a text file.

dirb http://192.168.1.106/dvwa/ -o output.txt

The above command will generate an output.txt file at the desktop of the enumerated directories.

Text

Description automatically generated

**Lab #10 Activities: Perform the above steps for your Kali Linux Machine.**