

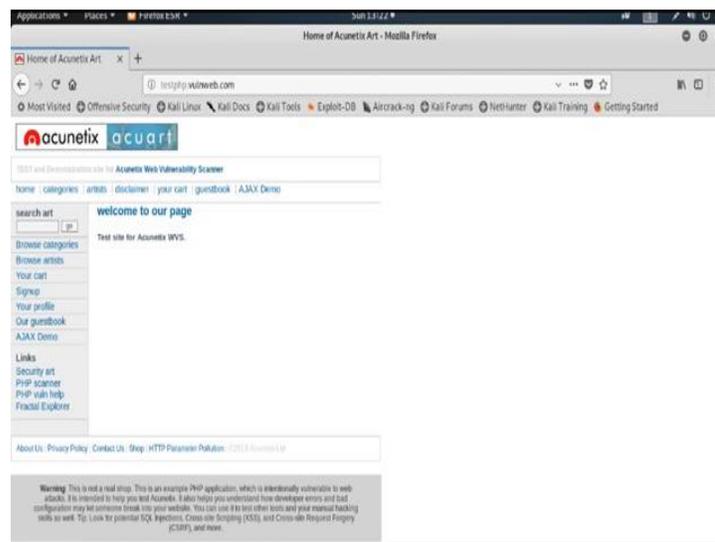
**Ethical Hacking**  
**Prof. Indranil Sengupta**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 51**  
**Web Application Vulnerability Scanning**

In this week, we will discuss about how to hack Web Application. Web application penetration testing is the most commonly used security testing technique for web applications. Web application penetration testing is done by unauthorized attack internally or maybe externally to get access to the sensitive data or may sometimes change the data stored inside the web application. A web penetration testing helps end user find out the possible vulnerabilities for a hacker to access the data from the internet; find all the security of their servers and also get to know how secure the web hosting site and server are.

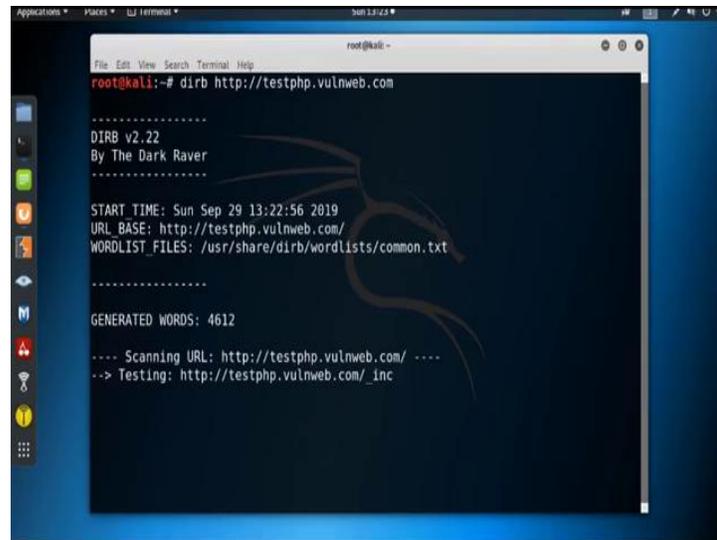
Like network for web application, our first step is also information gathering and scanning. We already discuss about the information gathering part previously; now we will discuss about the scanning. For web application for network scanning, we use the tool like nmap or nessus. For web application scanning we use the tool like dirb, uniscan, nikto, vega, acunetix, etc. Now, I will show you few tools.

(Refer Slide Time: 01:54)



Now, suppose consider a web site **testphp.vulnweb.com** as our target web site or web application, **testphp.vulnweb.com**. So, now our first step is to find out the vulnerabilities by scanning this particular web application. So now, first we will use the tool dirb to find out all possible directories of these particular web application. So, open terminal.

(Refer Slide Time: 03:05)



```
root@kali:~# dirb http://testphp.vulnweb.com

-----
DIRB v2.22
By The Dark Raver
-----

START TIME: Sun Sep 29 13:22:56 2019
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

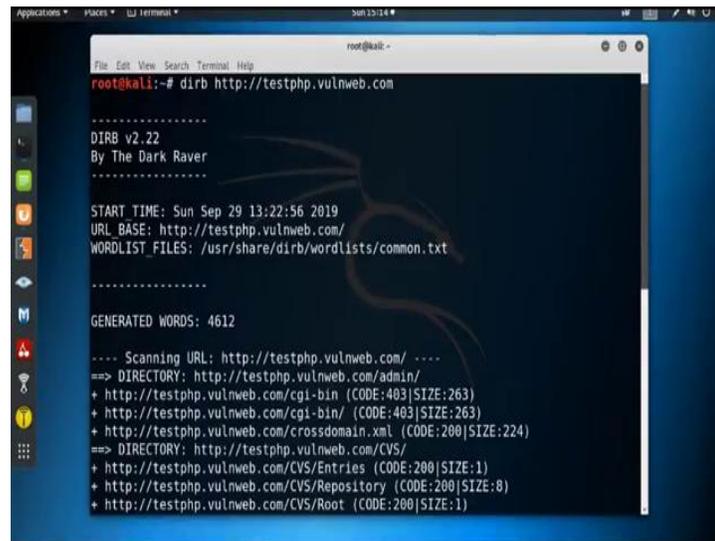
-----

GENERATED WORDS: 4612

---- Scanning URL: http://testphp.vulnweb.com/ ----
--> Testing: http://testphp.vulnweb.com/_inc
```

And use the command dirb followed by the url **http://testphp.vulnweb.com**, hit enter and it will search for the possible directories for this particular web application; we got the result.

(Refer Slide Time: 03:43)



```
root@kali:~# dirb http://testphp.vulnweb.com

-----
DIRB v2.22
By The Dark Raver
-----

START TIME: Sun Sep 29 13:22:56 2019
URL BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

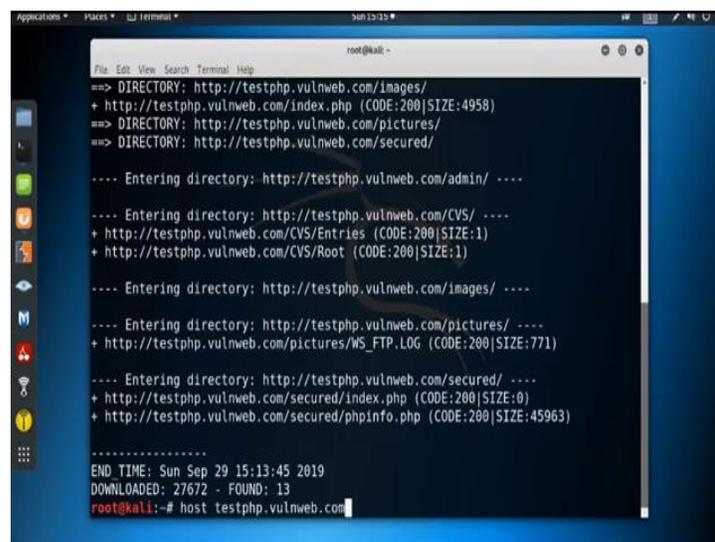
-----

GENERATED WORDS: 4612

---- Scanning URL: http://testphp.vulnweb.com/ ----
==> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:263)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:263)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
==> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
```

And see, we got all the directories inside this particular web application. We got common dot txt file and admin directory, CVS directory and images, picture, secured. We got all the directories which is inside this particular web application.

(Refer Slide Time: 04:09)



```
==> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:4958)
==> DIRECTORY: http://testphp.vulnweb.com/pictures/
==> DIRECTORY: http://testphp.vulnweb.com/secured/

---- Entering directory: http://testphp.vulnweb.com/admin/ ----

---- Entering directory: http://testphp.vulnweb.com/CVS/ ----
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)

---- Entering directory: http://testphp.vulnweb.com/images/ ----

---- Entering directory: http://testphp.vulnweb.com/pictures/ ----
+ http://testphp.vulnweb.com/pictures/MS_FTP.LOG (CODE:200|SIZE:771)

---- Entering directory: http://testphp.vulnweb.com/secured/ ----
+ http://testphp.vulnweb.com/secured/index.php (CODE:200|SIZE:0)
+ http://testphp.vulnweb.com/secured/phpinfo.php (CODE:200|SIZE:45963)

-----
END TIME: Sun Sep 29 15:13:45 2019
DOWNLOADED: 27672 - FOUND: 13
root@kali:~# host testphp.vulnweb.com
```

Next we will use the tool nikto to find out the vulnerabilities in this web application. So, to use the tool nikto, first we need the IP address of this particular web application. So, by using the command host, we can find out the IP address, host testphp.vulnweb.com.

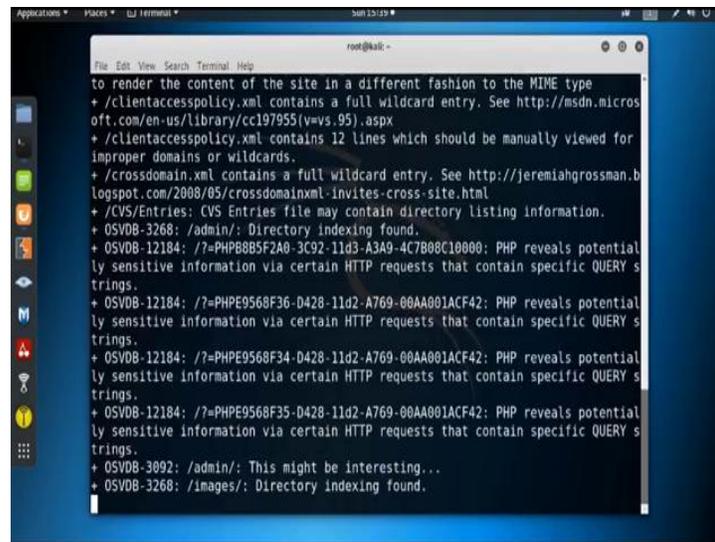
(Refer Slide Time: 04:55)



```
root@kali:~# http://testphp.vulnweb.com/secured/index.php (CODE:200|SIZE:0)
root@kali:~# http://testphp.vulnweb.com/secured/phpinfo.php (CODE:200|SIZE:45963)
-----
END TIME: Sun Sep 29 15:13:45 2019
DOWNLOADED: 27672 - FOUND: 13
root@kali:~# host testphp.vulnweb.com
testphp.vulnweb.com has address 176.28.50.165
root@kali:~# niko -host 176.28.50.165
- Nikto v2.1.6
-----
+ Target IP:      176.28.50.165
+ Target Hostname: 176.28.50.165
+ Target Port:    80
+ Start Time:    2019-09-29 15:15:59 (GMT-4)
-----
+ Server: nginx/1.4.1
+ Retrieved x-powered-by header: PHP/5.3.10-1-lucid+2uwsgi2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
```

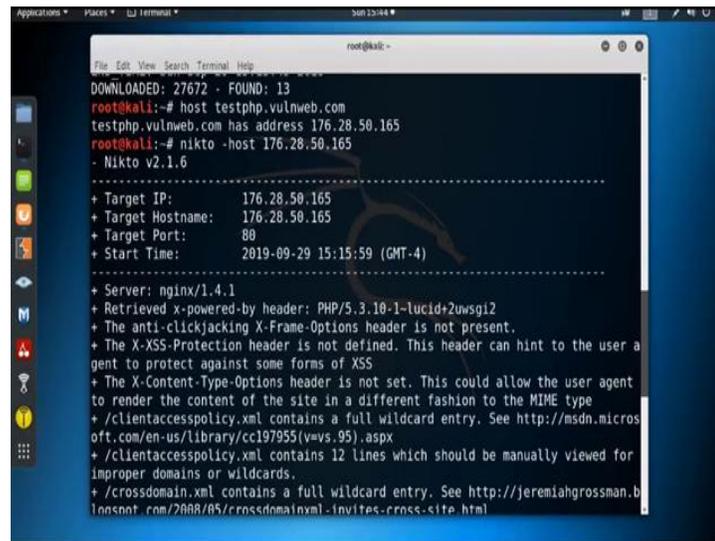
So, you got the IP address. So, use this IP address in the tool nikto; nikto, then this host specify the IP address, then the IP address of this particular web application.

(Refer Slide Time: 5:28)



```
to render the content of the site in a different fashion to the MIME type
+ /clientaccesspolicy.xml contains a full wildcard entry. See http://msdn.microsoft.com/en-us/library/cc197955(v=vs.95).aspx
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards.
+ /crossdomain.xml contains a full wildcard entry. See http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
+ /CVS/Entries: CVS Entries file may contain directory listing information.
+ OSVDB-3268: /admin/: Directory indexing found.
+ OSVDB-12184: /?=PHPE9568F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
```

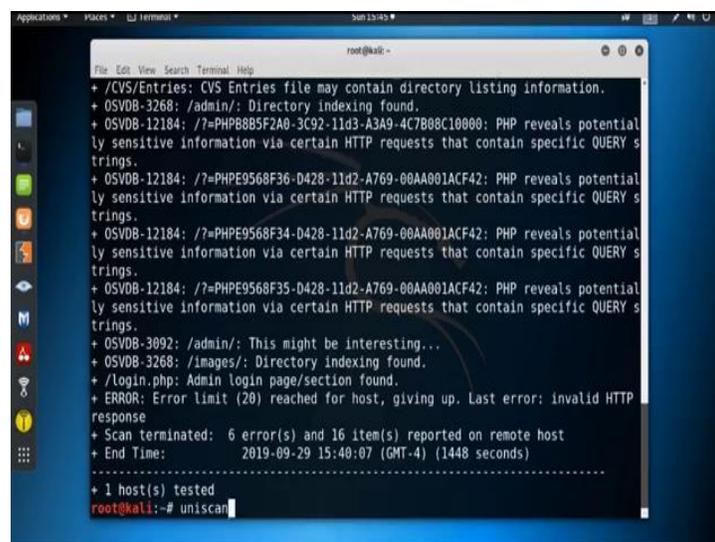
(Refer Slide Time: 05:39)



```
root@kali:~# host testphp.vulnweb.com
testphp.vulnweb.com has address 176.28.50.165
root@kali:~# niko -host 176.28.50.165
- Nikto v2.1.6
-----
+ Target IP:      176.28.50.165
+ Target Hostname: 176.28.50.165
+ Target Port:    80
+ Start Time:    2019-09-29 15:15:59 (GMT-4)
-----
+ Server: nginx/1.4.1
+ Retrieved x-powered-by header: PHP/5.3.10-1-lucid+2uwsgi2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ /clientaccesspolicy.xml contains a full wildcard entry. See http://msdn.microsoft.com/en-us/library/cc197955(v=vs.95).aspx
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards.
+ /crossdomain.xml contains a full wildcard entry. See http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
```

We got the result and see some vulnerabilities listed, target IP, this target hostname, target port is 80 and server is nginx retrieved x powered by header this; the x-xss protection header is not defined; this header can hint to the user agent to protect against some form of xss. Shows all the vulnerabilities which are available in that particular web application are listed here.

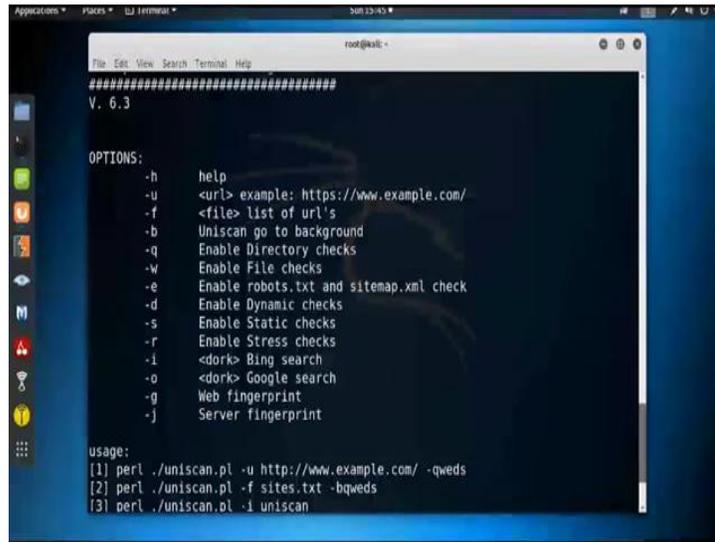
(Refer Slide Time: 06:25)



```
root@kali:~# uniscan
-----
+ /CVS/Entries: CVS Entries file may contain directory listing information.
+ OSVDB-3268: /admin/: Directory indexing found.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potential ly sensitive information via certain HTTP requests that contain specific QUERY s trings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potential ly sensitive information via certain HTTP requests that contain specific QUERY s trings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potential ly sensitive information via certain HTTP requests that contain specific QUERY s trings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potential ly sensitive information via certain HTTP requests that contain specific QUERY s trings.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
+ /Login.php: Admin login page/section found.
+ ERROR: Error limit (20) reached for host, giving up. Last error: invalid HTTP response
+ Scan terminated: 6 error(s) and 16 item(s) reported on remote host
+ End Time:      2019-09-29 15:40:07 (GMT-4) (1448 seconds)
-----
+ 1 host(s) tested
root@kali:~#
```

Now, I can also use the tool uniscan.

(Refer Slide Time: 06:36)



```
root@kali: ~
#####
V. 6.3

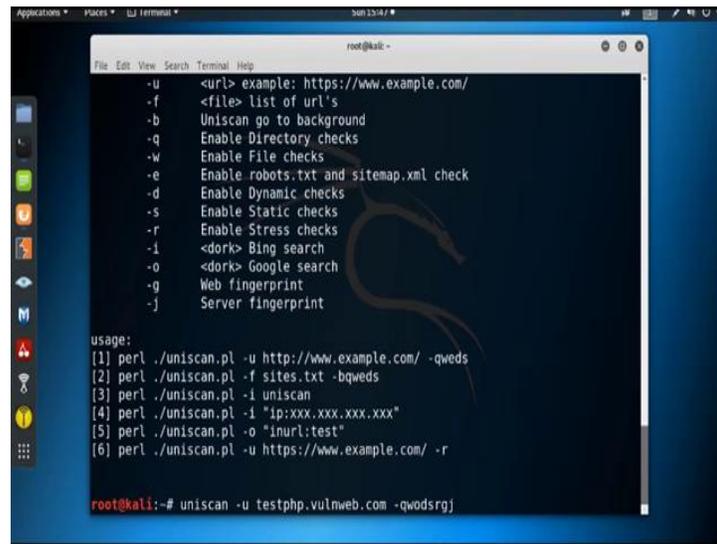
OPTIONS:
-h      help
-u      <url> example: https://www.example.com/
-f      <file> list of url's
-b      Uniscan go to background
-q      Enable Directory checks
-w      Enable File checks
-e      Enable robots.txt and sitemap.xml check
-d      Enable Dynamic checks
-s      Enable Static checks
-r      Enable Stress checks
-i      <dork> Bing search
-o      <dork> Google search
-g      Web fingerprint
-j      Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bqweds
[3] perl ./uniscan.pl -i uniscan
```

Now, just by typing uniscan, we can get the help of uniscan and see all these options are available; h for help, then -u specify the url and -f list of urls, -b uniscan to go to background, -q enable directory checks, -w check file, -e enable robots.txt file and sitemap.xml file check, -t enable dynamic check, -s enable static check, -r enable stress check, -i it search in Bing, -o search in Google, -g find out the web fingerprint, -j server fingerprint.

So, now suppose I am going to scan the same web application testphp.vulnweb.com using the tool uniscan with some specified options.

(Refer Slide Time: 07:50)



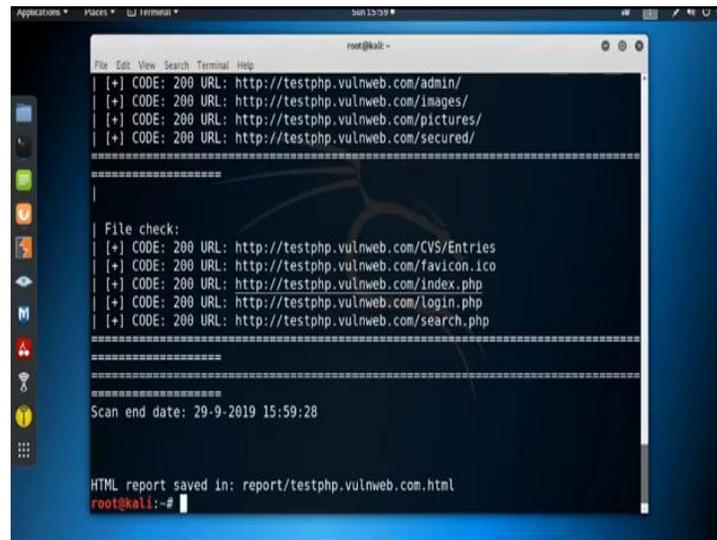
```
root@kali:~# uniscan -h
-u <url> example: https://www.example.com/
-f <file> list of url's
-b Uniscan go to background
-q Enable Directory checks
-w Enable File checks
-e Enable robots.txt and sitemap.xml check
-d Enable Dynamic checks
-s Enable Static checks
-r Enable Stress checks
-i <dork> Bing search
-o <dork> Google search
-g Web fingerprint
-j Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bqweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r

root@kali:~# uniscan -u testphp.vulnweb.com -qwdsrgj
```

uniscan then -u specify the url testphp.vulnweb.com, then specify the option -qwdsrgj.

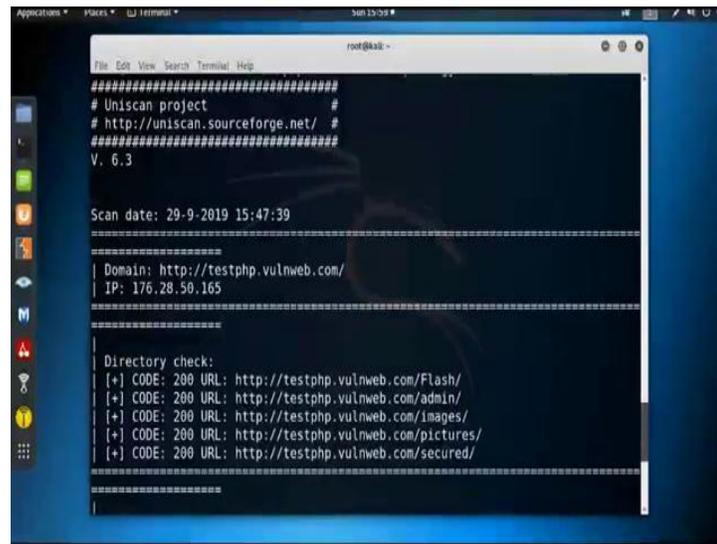
(Refer Slide Time: 08:32)



```
root@kali:~# uniscan -u testphp.vulnweb.com -qwdsrgj
[+] CODE: 200 URL: http://testphp.vulnweb.com/admin/
[+] CODE: 200 URL: http://testphp.vulnweb.com/images/
[+] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
[+] CODE: 200 URL: http://testphp.vulnweb.com/secured/
=====
|
| File check:
| [+] CODE: 200 URL: http://testphp.vulnweb.com/CVS/Entries
| [+] CODE: 200 URL: http://testphp.vulnweb.com/favicon.ico
| [+] CODE: 200 URL: http://testphp.vulnweb.com/index.php
| [+] CODE: 200 URL: http://testphp.vulnweb.com/login.php
| [+] CODE: 200 URL: http://testphp.vulnweb.com/search.php
|
=====
=====
Scan end date: 29-9-2019 15:59:28

HTML report saved in: report/testphp.vulnweb.com.html
root@kali:~#
```

(Refer Slide Time: 08:39)



```
root@kali:~# uniscan -u http://testphp.vulnweb.com/
#####
# Uniscan project #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3

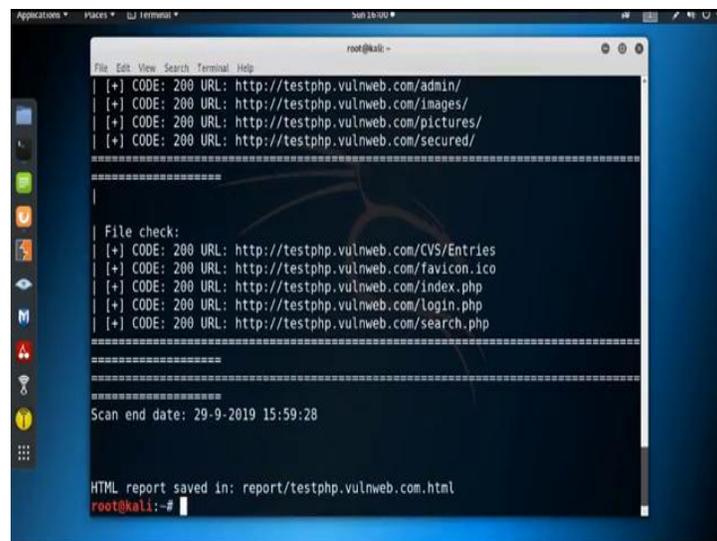
Scan date: 29-9-2019 15:47:39

=====
| Domain: http://testphp.vulnweb.com/
| IP: 176.28.50.165
=====

Directory check:
[+] CODE: 200 URL: http://testphp.vulnweb.com/Flash/
[+] CODE: 200 URL: http://testphp.vulnweb.com/admin/
[+] CODE: 200 URL: http://testphp.vulnweb.com/images/
[+] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
[+] CODE: 200 URL: http://testphp.vulnweb.com/secured/
=====
```

We got the result. It find out all the directory, flash, admin, images, pictures, secured and it also check all the file.

(Refer Slide Time: 08:49)



```
root@kali:~# uniscan -u http://testphp.vulnweb.com/
#####
# Uniscan project #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3

Scan date: 29-9-2019 15:47:39

=====
| Domain: http://testphp.vulnweb.com/
| IP: 176.28.50.165
=====

Directory check:
[+] CODE: 200 URL: http://testphp.vulnweb.com/Flash/
[+] CODE: 200 URL: http://testphp.vulnweb.com/admin/
[+] CODE: 200 URL: http://testphp.vulnweb.com/images/
[+] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
[+] CODE: 200 URL: http://testphp.vulnweb.com/secured/
=====

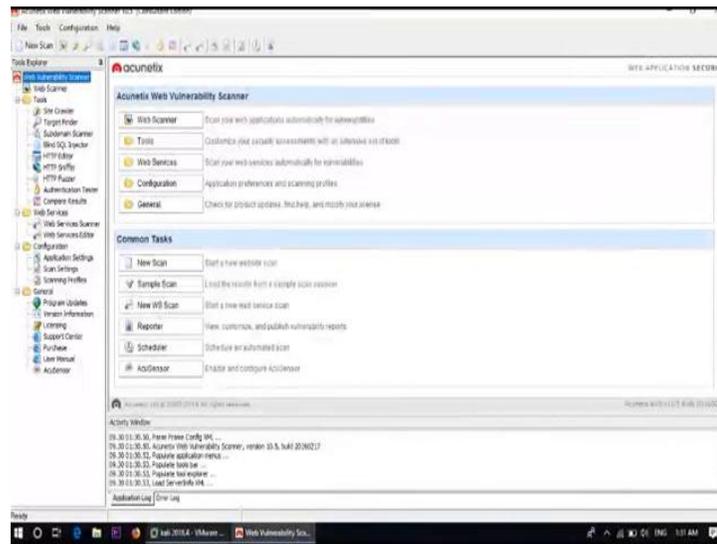
File check:
[+] CODE: 200 URL: http://testphp.vulnweb.com/CVS/Entries
[+] CODE: 200 URL: http://testphp.vulnweb.com/favicon.ico
[+] CODE: 200 URL: http://testphp.vulnweb.com/index.php
[+] CODE: 200 URL: http://testphp.vulnweb.com/login.php
[+] CODE: 200 URL: http://testphp.vulnweb.com/search.php
=====

Scan end date: 29-9-2019 15:59:28

HTML report saved in: report/testphp.vulnweb.com.html
root@kali:~#
```

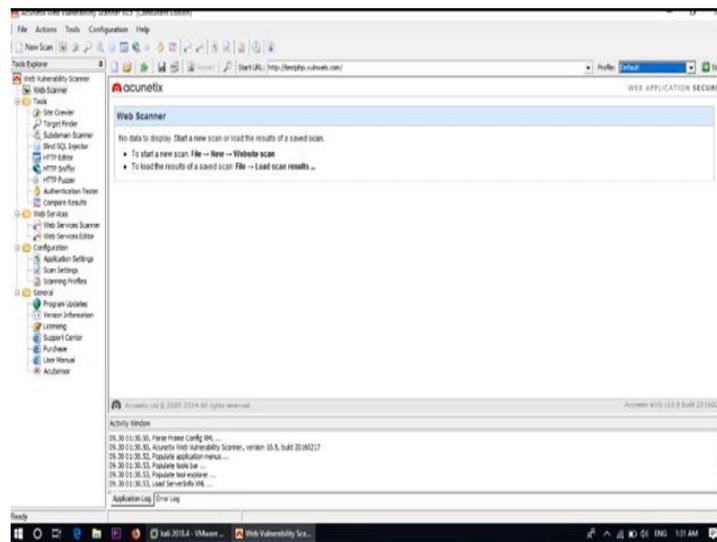
And also set the html report into the folder report testphp.vulnweb.com.html. Now finally, we will use the best tool acunetix to find out the vulnerabilities to that particular web application; open the tool acunetix.

(Refer Slide Time: 09:21)



Here is my tool acunetix; go to web scanner.

(Refer Slide Time: 09:28)



And put the web application url, <http://testphp.vulnweb.com> which is our target web application. Now, we can select your profile; for blind SQL injection, you can use this option csrf, dictionary and file check, file upload, all these option are available. For the time being, I am using default option; then click on start.

