Unit-3

W3af (Web Application Attack and Audit Framework)

w3af is divided into two main parts, the core and the plug-ins.[3] The core coordinates the process and provides features that are consumed by the plug-ins, which find the vulnerabilities and exploit them. The plug-ins are connected and share information with each other using a knowledge base.

W3af is a popular web application attack and audit framework. This framework aims to provide a better web application penetration testing platform. It was developed using Python. By using this tool, you will be able to identify more than 200 kinds of web application vulnerabilities including SQL injection, cross-site scripting and many others.

It comes with a graphical and console interface. You can use it easily, thanks to its simple interface.
If you are using it with a graphical interface, I do not think that you are going to face any problems with the tool. You only need to select the options and then start the scanner. If a website needs authentication, you can also use authentication modules to scan the session-protected pages.

We have already covered this tool in detail in our previous W3af walkthrough series. You can read those articles to know more about this tool.

Nikto Overview

Nikto was originally written and maintained by Sullo, CIRT, Inc. It is currently maintained by David Lodge, though other contributors have been involved in the project as well.
It is built to run on any platform which has a Perl environment and has been incorporated within the Kali Linux Penetration Testing distribution. It is an open source tool, supporting SSL, proxies, host authentication, IDS evasion, and more. It can be updated automatically from the command-line, and supports the optional submission of updated version data back to the maintainers.
Nikto allows pentesters, hackers and developers to examine a web server to find potential problems and security vulnerabilities, including:

- Server and software misconfigurations
- Default files and programs
- Insecure files and programs
- Outdated servers and programs

Nikto Features

During web app scanning, different scenarios might be encountered. Nikto supports a wide variety of options that can be implemented during such situations. The following is an overview of the included options in Nikto:

-Cgidirs: This option is used to scan specified CGI directories. Users can filter "none" or "all" to scan all CGI directories or none. A literal value for a CGI directory such as "/cgi-test/" may also be specified (note that a trailing slash is required). If this is option is not specified, all CGI directories listed in *config.txt* will be tested.

-config: This option allows the pentester, hacker, or developer to specify an alternative config file to use instead of the *config.txt* located in the install directory.

-Display: One can control the output that Nikto shows. Reference numbers are used for specification. Multiple numbers may be used as well. The allowed reference numbers can be seen below:
1 – Show redirects

2 – Show cookies received

3 – Show all 200/OK responses

4 – Show URLs which require authentication

D – Debug Output

V – Verbose Output

-evasion: pentesters, hackers and developers are also allowed to specify the Intrusion Detection System evasion technique to use. This option also allows the use of reference numbers to specify the type of technique. Multiple number references may be used:

1 – Random URI encoding (non-UTF8)

2 – Directory self-reference (/./)

3 – Premature URL ending

4 – Prepend long random string

5 – Fake parameter

6 – TAB as request spacer

7 – Change the case of the URL

8 – Use Windows directory separator (\)

-Format: One might require output/results to be saved to a file after a scan. This option does exactly that. The -o (-output) option is used; however, if not specified, the default will be taken from the file extension specified in the -output option. Valid formats are:
csv – for a comma-separated lists

htm – for an HTML report

txt – for a text report

xml – for an XML report

-host: This option is used to specify host(s) to target for a scan. It can be an IP address, hostname, or text file of hosts.

-id: **F**or websites that require authentication, this option is used to specify the ID and password to use. The usage format is "id:password".
-list-plugins: This option will list all plugins that Nikto can run against targets and then will exit without performing a scan. These can be tuned for a session using the -plugins option.
The output format is:

Plugin name

full name – description

-no404: This option is used to disable 404 (file not found) checking. This reduces the total number of requests made to the web server and may be preferable when checking a server over a slow internet connection or an embedded device. However, this will generally lead to more false positives being discovered.

-plugins: This option allows one to select the plugins that will be run on the specified targets. A comma-separated list should be provided which lists the names of the plugins. The names can be found by using *-list-plugins*.
There are two special entries: *ALL*, which specifies all plugins shall be run and *NONE*, which specifies no plugins shall be run. The default is ALL.

-port: This option specifies the TCP port(s) to target. To test more than one port on the same host, one can specify the list of ports in the -p (-port) option. Ports can be specified as a range (i.e., 80-90), or as a comma-delimited list, (i.e., 80,88,90). If not specified, port 80 is used.

-Pause: This option can be used to prevent tests from being blocked by a WAF for seeming too suspicious. It defines the seconds to delay between each test.

-timeout: It is sometimes helpful to wait before timing out a request. This option specifies the number of seconds to wait. The default timeout is 10 seconds.
-useproxy: This option is used in the event that the networks connected to require a proxy. This option asks Nikto to use the HTTP proxy defined in the configuration file.

-update: This option updates the plugins and databases directly from *cirt.net*.

Brute Force Attack

What is a Brute Force Attack

A brute force attack is a popular cracking method: by some accounts, brute force attacks accounted for five percent of confirmed security breaches. A brute force attack involves 'guessing' username and passwords to gain unauthorized access to a system. Brute force is a simple attack method and has a high success rate.

Some attackers use applications and scripts as brute force tools. These tools try out numerous password combinations to bypass authentication processes. In other cases, attackers try to access web applications by searching for the right session ID. Attacker motivation may include stealing information, infecting sites with malware, or disrupting service.

While some attackers still perform brute force attacks manually, today almost all brute force attacks today are performed by bots. Attackers have lists of commonly used credentials, or real user credentials, obtained via security breaches or the dark web. Bots systematically attack websites and try these lists of credentials, and notify the attacker when they gain access.

Types of Brute Force Attacks

- **Simple brute force attack:** uses a systematic approach to 'guess' that doesn't rely on outside logic.
- **Hybrid brute force attacks:** starts from external logic to determine which password variation may be most likely to succeed, and then continues with the simple approach to try many possible variations.
- **Dictionary attacks :**guesses usernames or passwords using a dictionary of possible strings or phrases.
- **Rainbow table attacks:** a rainbow table is a precomputed table for reversing cryptographic hash functions. It can be used to guess a function up to a certain length consisting of a limited set of characters.
- **Reverse brute force attack:** uses a common password or collection of passwords against many possible usernames. Targets a network of users for which the attackers have previously obtained data.

- **Credential stuffing:** uses previously-known password-username pairs, trying them against multiple websites. Exploits the fact that many users have the same username and password across different systems.

Hydra and Other Popular Brute Force Attack Tools

Security analysts use the THC-Hydra tool to identify vulnerabilities in client systems. Hydra quickly runs through a large number of password combinations, either simple brute force or dictionary-based. It can attack more than 50 protocols and multiple operating systems. Hydra is an open platform; the security community and attackers constantly develop new modules.

Other top brute force tools are:

- **Aircrack-ng** : can be used on Windows, Linux, iOS, and Android. It uses a dictionary of widely used passwords to breach wireless networks.
- **John the Ripper** : runs on 15 different platforms including Unix, Windows, and OpenVMS. Tries all possible combinations using a dictionary of possible passwords.
- **L0phtCrack** : tool for cracking Windows passwords. It uses rainbow tables, dictionaries, and multiprocessor algorithms.
- **Hashcat** : works on Windows, Linux, and Mac OS. Can perform simple brute force, rule-based, and hybrid attacks.
- **DaveGrohl** : an open-source tool for cracking Mac OS. Can be distributed across multiple computers.
- **Ncrack**: : a tool for cracking network authentication. It can be used on Windows, Linux, and BSD.

John the Ripper

- John the Ripper is another awesome tool that does not need any introduction. It has-been a favorite choice for performing brute-force attack for long time. This free password-cracking software was initially developed for Unix systems.
- Later, developers released it for various other platforms. Now, it supports fifteen different platforms including Unix, Windows, DOS, BeOS, and OpenVMS. You can use this either to identify weak passwords or to crack passwords for breaking authentication.
- This tool is very popular and combines various password-cracking features. It can automatically detect the type of hashing used in a password. Therefore, you can also run it against encrypted password storage.
- Basically, it can perform brute-force attack with all possible passwords by combining text and numbers. However, you can also use it with a dictionary of passwords to perform dictionary attacks.

L0phtCrack

- L0phtCrack is known for its ability to crack Windows passwords. It uses dictionary, brute-force, hybrid attacks, and rainbow tables.
- The most notable features ofl0phtcrack are scheduling, hash extraction from 64 bit Windows versions, multiprocessor algorithms, and networks monitoring and decoding.
- If you want to crack the password of Windows system, you can try this tool.

HTC Hydra

- Hydra is a parallelized network logon cracker.
- Hydra works by using different approaches of generating possible passwords, such as wordlist attacks, brute-force attacks and others.
- Hydra is commonly used by penetration testers together with a program named crunch, which is used to generate wordlists. Hydra is then used to test the attacks using the wordlists that crunch created.

THC Hydra

- THC Hydra is known for its ability to crack passwords of network authentications by performing brute-force attacks. It performs dictionary attacks against more than 30protocols including telnet, ftp, http, https and more.
- It is available for various platforms including Linux, Windows/Cygwin, Solaris 11, FreeBSD 8.1, OpenBSD, OSX and QNX/Blackberry
- Download THC Hydra from this link: https://sectools.org/tool/hydra/
- These are a few popular brute-forcing tools for password cracking. There are various other tools are also available which perform brute-force on different kinds of authentication. If I just give example of few small tools, you will see most of the PDF cracking, ZIP cracking tools use the same brute-force method to perform attacks and cracks passwords. There are many such tools available for free or paid.

Pwdump

- Pwdump2, http://www.openwall.com/passwords/nt.shtml, by Todd Sabin, can be used to extract the hashed passwords from a Windows system.
- It is a command-line tool that must be run locally on the target system; however, we'll take a look at pwdump3, which can operate remotely, later in this section.
  Implementation
- The program must be run locally on the system. This is version 2 of a tool first developed
- by Jeremy Allison of the Samba project.
- Unlike the first version, pwdump2 is not inhibited by Sys Key encryption of the SAM database. Sys Key was introduced in Windows NT in an attempt to add additional security to the SAM database, but its effectiveness is questionable, as we will see with pwdump2.
- The usage for pwdump2 is shown here:
  C:\>pwdump2.exe /?

OpenSSL

- OpenSSL is the cryptographic library behind all three of these tools. As such, you have the full power of OpenSSL at your disposal. Finally, there are a few caveats with OpenSSL and OpenSSH that must be addressed before going any further. Though these items are relatively easy to address, the specifics on how to do this fall outside the scope of this article and are left as an exercise for the reader. There is copious documentation readily available on the internet.

OpenSSL

1. Disallow SSL version 2 , commonly seen as SSLv2. SSLv2 is vulnerable to the cipher suite rollback attack. While this doesn't directly expose your data, it can cause the data to be encrypted with very low encryption, potentially allowing a Black Hat to decrypt your data at a later date.

2. OpenSSL allows you to specify which encryption methods you want to use and which methods you don't want to use. Take advantage of this and expressly disallow any low level encryption methods.

OpenSSH

1. Disallow SSH version 1, commonly seen as SSHv1. Tools like Ettercap can decrypt SSHv1 traffic in real time. I've done this in a lab environment, so I can personally vouch for the ease with which this can be done.

2. Increase the key size to as large as is feasible for your environment. Similarly, decrease the time between regenerating the key to as little as is feasible for your environment. Don't over do it, though. If you are generating huge keys too frequently, performance will take a tremendous hit.

Stunnel

Of the three tools we'll be discussing, Stunnel is the simplest; it is lightweight, does one thing, and does it very well. In most cases, you'll want to use Stunnel when the communication requirements are simple, both in terms of complexity and quantity. For example, we might want to add SSL support to a web server that doesn't have SSL support. Let's say we want to run Apache on a Windows box. Apache is readily available for Windows, but Apache with SSL for Windows is a bit more problematic. We could get a third party distribution like XAMPP, but in some circumstances, that might be overkill. We could try to compile Apache with mod_ssl for ourselves. This may sound relatively

simple, but trust me, it isn't. Stunnel to the rescue. With Stunnel, we can add SSL support in, literally, a matter of seconds.

Another case where we might want to use Stunnel is when we can't or don't want to establish and maintain a tunnel between the client and server. Let's say we have users who are telecommuting and they need to access a server on the inside of the enterprise firewall. Good firewall practices says we should set the connection timeout very low. If we establish a tunnel and due to lack of activity the firewall drops the connection, users are inconvenienced by having to login again. Stunnel is stateless, so we can conveniently work around this issue.

We may also decide to use Stunnel when we don't mind opening up additional ports on the server for each service we want to secure. It is important to note that this is a one-to-one ratio. For each port we want to secure, we must open another port for Stunnel.

Adding SSL to a Web Server Let's continue with the example mentioned previously: adding SSL support to a web server. In this particular case, since web browsers already know how to speak SSL, there is absolutely no client-side setup. On the server side, we setup Stunnel to listen for remote TCP requests on port 443 (this is the standard port for HTTPS, encrypted HTTP traffic) and to redirect locally to port 80, the standard HTTP port.

Example 1. Server Side Config cert = /some/directory/server.crt
key = /some/directory/server.key
ciphers = TLSv1:SSLv3:!SSLv2:!LOW:@STRENGTH
[https]
accept = 443
connect = 80

Pros

- Stunnel is easy to set up. Most setups, quite literally, only take a minute or two.
- Stunnel is secure. You can use encryption as high as OpenSSL supports.
- Multiple tunnels can be setup in a single config file, thus requiring only a single instance of stunnel on the server.
- As long as we're going to use an non-privileged port, Stunnel can be run by a non-privileged user.

Cons

- A separate tunnel is required for each service to be secured.
- Stunnel requires a modest understanding of setting up and using Certificate Authorities. As mentioned previously, though, tools like XCA can handle most of this for you.

Curl

What cURL stands for?

Client URL Request Library

cURL is an abbreviation for Client URL Request Library. Basically cURL is name of the project. cURL is used to transfer data from one place to another place. It is a command line tool for receiving and sending files using URL syntax.

What is a curl script?

curl is a command line tool to transfer data to or from a server, using any of the supported protocols (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP or FILE). curl is powered by Libcurl. This tool is preferred for automation, since it is designed to work without user interaction

Why is curl used?

curl is a widely used because of its ability to be flexible and complete complex tasks. For example, you can use curl for things like user authentication, HTTP post, SSL connections, proxy support, FTP uploads, and more! You can also do simple things with curl, such as download web pages and web images.

HTTP Curl

Curl also supports user and password in HTTP URLs, thus you can pick a file like:

```
curl http://name:passwd@machine.domain/full/path/to/file
```

or specify user and password separately like in

```
curl -u name:passwd http://machine.domain/full/path/to/file
```

HTTP offers many different methods of authentication and curl supports several: Basic, Digest, NTLM and Negotiate (SPNEGO). Without telling which method to use, curl defaults to Basic. You can also ask curl to pick the most secure ones out of the ones that the server accepts for the given URL, by using --anyauth.

Note! According to the URL specification, HTTP URLs cannot contain a user and password, so that style will not work when using curl via a proxy, even though curl allows it at other times. When using a proxy, you must use the -u style for user and password.