Laboratory works no. 4

Classic pentest tools

In this laboratory we will analyze a number of tools and operating systems dedicated to penetration testing (pentest).

You are not allowed to apply these tools on any site of anyone (for testing you will use only personal computers depending on the red or blue team of which you are a part.

It is obvious that a professional will not waste time with this but will install the necessary tools in his favorite distribution and will modify them according to his experience and needs.

For newbies, however, this is the preferred approach. One of the most used operating systems in the field is Kali (based on Debian) who comes preconfigured with a collection of tools used in pentest. It can be installed as such or executed as a virtual machine.

For real installation will be used: https://www.kali.org/downloads/

For installation in the virtual machine is recommended for those experienced in Linux to use all the original image.

However for those with no Linux experience it is recommended to use the existing prepared machine at https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/ that has root with the password toor. After starting the machine the first thing to do at the console level is to bring UP to date OS and such applications

apt update

apt upgrade

This is necessary for both security reasons and for reasons of good functioning. In the type of upgrade operation, read carefully what you are being asked and do not make selections by ear. (Go on the Internet and read until you know it). Resume the process as many times as necessary. It also wouldn't hurt to see what users still exist and change your root password to a custom one. If launched into the virtual machine, adapt the initial configuration of the virtual machine according to the resources of the system.

Some kitchen stuff...

apt install mc

apt install doublecmd-qt

apt install xfce4-goodies

For quick screenshots: xfce4-screenshooter and create a link on the desktop In Kali does not go directly so we go to the console.

ln -s /usr/bin/xfce4-screenshooter /root/Desktop

Manually install the VMWare tools for that machine over those open source already preinstalled in the virtual machine. This will give you interoperability like shared directories with host and copy/paste. As discussed in the course the first phase is gathering information about the target entity. For starters we will present a simple example of footprinting for websites. For this we will use nikto from the equipment. A man nikto doesn't hurt at a man's house. Read carefully!

Now there will be a series of tests on a blog with joomla. Since the students have quite thin knowledges in Linux installations we present the entire installation and testing cycle. For tests we used Parrot up-to-date with the news.
For testing applications in this lab you have two options
1. Apply the tools to the ngix already made available on the docker 3 machine. In Kali from docker or for those who prefer windows in the virtual machine version will install a blog as follows:

joomla deploy

```
sudo apt-get update -y
sudo apt-get upgrade -y
```

Apache 2 is already available in Parrot

```
sudo systemctl start apache2
sudo systemctl enable apache2
```

```
dpkg --list | grep php | awk '/^ii/{ print $2}'
```

You can see that Php 7 is already installed in the Parrot but I can check this with

```
php -m | grep -i mysql
```

And you can see that mysql and mysqli extensions for php are not so:

```
apt-get install php-mysql
```

```
sudo nano /etc/php/7.3/apache2/php.ini
```

Check if ...

```
memory_limit = 256M

upload_max_filesize = 32M

post_max_size = 32M

date.timezone = whatever you want
```

At Parrot is ok
Now we will install mariodb

```
sudo apt-get install mariadb-server -y
```

And will assure boot time start

```
sudo systemctl start mysql
sudo systemctl enable mysql
```

Some security stuff for Mario

```
sudo mysql_secure_installation
```

If there are password problem we can reset it with
Parrot mysql / mariodb passwd reset

```
service mysql stop
```

mysqld_safe --skip-grant-tables &

Press Enter twice then
mysql -u root mysql

Then

UPDATE user SET password=PASSWORD('my new p4ssw0rd') WHERE user='root';

FLUSH PRIVILEGES;

exit
service mysql restart

joomla database creation - logging
mysql -u root -p

Now the new password will work and we can finally see the MarioDB prompter

joomla database creation - real deal
CREATE DATABASE joomla_db;

User creation
MariaDB [(none)]> CREATE USER joomla@localhost;
MariaDB [(none)]> SET PASSWORD FOR 'joomla'@'localhost' =
PASSWORD("password");

Priviledge grants for newly created user
GRANT ALL PRIVILEGES ON joomla_db.* TO 'joomla'@'localhost' IDENTIFIED BY
'password' WITH GRANT OPTION;

FLUSH PRIVILEGES;

Exit

Finally wil begin joomla deploy
We download it
wget
https://github.com/joomla/joomla-cms/releases/download/3.7.3-rc1/Joomla_3.7.3-rc1-Release_Candidate-Full_Package.tar.gz

It will be a good idea to check if a new version of joomla was deployed and modify the commands acordingly. Nomather what we continue...
sudo mkdir /var/www/html/joomla
sudo tar -xvzf Joomla_3.7.3-rc1-Release_Candidate-Full_Package.tar.gz -C
/var/www/html/joomla

Choose correct rights

```
sudo chown -R www-data:www-data /var/www/html/joomla
sudo chmod -R 750 /var/www/html/joomla
```

Then we create a Jooml virtual host in by deployng/creating a configuration file as follows

```
sudo nano /etc/apache2/sites-available/joomla.conf
```

The file will contain the following (if you are an experieced user feel free to make any supplementary modification as you will need/consider.

```
<VirtualHost *:80>

ServerAdmin [email protected]

DirectoryIndex index.php

DocumentRoot /var/www/html/joomla

ServerName 192.168.0.2

ServerAlias www.yourdomain.com

<Directory /var/www/html/joomla>

Options FollowSymLinks

AllowOverride All

Order allow,deny

allow from all

</Directory>

ErrorLog /var/log/apache2/joomla-error_log

CustomLog /var/log/apache2/joomla-access_log common

</VirtualHost>
```

Now disable the default host and activate the one for joomla

```
sudo a2dissite 000-default
sudo a2ensite joomla
```

Apache reload

```
sudo systemctl restart apache2
```

If it is not installed it will be a good idea to install and set the gufw to gain a minimal protection apt install gufw

Sudo ufw enable

Because we have the ufw this must be configured to let go for joomla

sudo ufw allow http

Configurations can be edited directly and with

Nano /var/www/html/joomla/installation/configuration.php-dist

For problems you can look in the log files /var/log/apache2/ And we should find to domain name or ip station joomla With nume_site/administrator enter the administration backend of joomla

Configuration.php-dist here must be put at least user name, database name and password Apt install php-xml for joomla in parrot It is done the installation of the default frontend and ready And now let's see the results of the analysis with nikto before the installation process completion from the web frontend

```
nikto -h 192.168.0.2

- Nikto v2.1.6

---------------------------------------------------------------------------

+ Target IP:            192.168.0.2

+ Target Hostname:      192.168.0.2

+ Target Port:          80

+ Start Time:           2019-11-15 08:39:03 (GMT2)

---------------------------------------------------------------------------

+ Server: Apache/2.4.41 (Debian)

+ 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

+ Root page / redirects to: installation/index.php
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or restrict access to allowed sources.

+ OSVDB-3092: /LICENSE.txt: License file found may identify site software.

+ /htaccess.txt: Default Joomla! htaccess.txt file found. This should be removed or renamed.

+ 8725 requests: 0 error(s) and 6 item(s) reported on remote host

+ End Time:           2019-11-15 08:39:47 (GMT2) (44 seconds)

---------------------------------------------------------------------------

+ 1 host(s) tested
```

Retest with nikto after the installation process is complete

```
nikto -h 192.168.0.2

- Nikto v2.1.6

---------------------------------------------------------------------------

+ Target IP:          192.168.0.2

+ Target Hostname:    192.168.0.2

+ Target Port:        80

+ Start Time:         2019-11-15 09:20:38 (GMT2)

---------------------------------------------------------------------------

+ Server: Apache/2.4.41 (Debian)

+ 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

+ Entry '/administrator/' in robots.txt returned a non-forbidden or redirect HTTP code (200)

+ "robots.txt" contains 14 entries which should be manually viewed.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.

+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache configuration file or restrict access to allowed sources.

+ OSVDB-3092: /administrator/: This might be interesting...

+ OSVDB-3092: /LICENSE.txt: License file found may identify site software.

+ /htaccess.txt: Default Joomla! htaccess.txt file found. This should be removed or renamed.

+ /administrator/index.php: Admin login page/section found.

+ 8740 requests: 0 error(s) and 11 item(s) reported on remote host

+ End Time:           2019-11-15 09:21:32 (GMT2) (54 seconds)

---------------------------------------------------------------------------

+ 1 host(s) tested
```

Already noted major "improvements" in blog security right? So!!!! stop installing/modifying, and that must occur in the maintenance cycle of an enabled web topic and retest with dedicated tools!!!! An example of nikto on a parrot with apache 2

```
nikto -h 127.0.0.1

- Nikto v2.1.6

---------------------------------------------------------------------------

+ Target IP:          127.0.0.1

+ Target Hostname:    127.0.0.1

+ Target Port:        80

+ Start Time:         2019-11-14 14:59:02 (GMT2)

---------------------------------------------------------------------------

+ Server: Apache/2.4.41 (Debian)

+ 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

+ No CGI Directories found (use '-C all' to force check all possible dirs)

+ Server may leak inodes via ETags, header found with file /, inode: 29cd, size: 5969754783fe9, mtime: gzip

+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD

+ ///etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.

+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache configuration file or restrict access to allowed sources.

+ /wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.

+ /wordpresswp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.

+ /wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.

+ /wordpresswp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.

+ /wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.

+ /wordpresswp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.

+ /assets/mobirise/css/meta.php?filesrc=: A PHP backdoor file manager was found.

+ /login.cgi?cli=aa%20aa%27cat%20/etc/hosts: Some D-Link router remote command execution.

+ /shell?cat+/etc/hosts: A backdoor was identified.

+ 7889 requests: 0 error(s) and 16 item(s) reported on remote host
+ End Time:           2019-11-14 14:59:46 (GMT2) (44 seconds)

---------------------------------------------------------------------------

+ 1 host(s) tested
```

We are now using a tool for database software analysis

```
sqlmap -u 192.168.0.2 --forms --crawl=2
```

Then patiently perform all the tests recommended by the tool - obviously you need to read extra to understand some of the problems reported or the analyses made. That is, if you don't already know from those who taught you to design systems of this type. Below is a short excerpt from the app log:

```
  you want to check for the existence of site's sitemap(.xml) [y/N] y

[09:37:52] [WARNING] 'sitemap.xml' not found

[09:37:52] [INFO] starting crawler

[09:37:52] [INFO] searching for links with depth 1

[09:37:53] [INFO] searching for links with depth 2

please enter number of threads? [Enter for 1 (current)] 1

[09:37:57] [WARNING] running in a single-thread mode. This could take a while

do you want to store crawling results to a temporary file for eventual further
processing with other tools [y/N] n

[09:38:03] [INFO] sqlmap got a total of 23 targets

[#1] form:

POST http://192.168.0.2/index.php

POST data: searchword=&task=search&option=com_search&Itemid=101

do you want to test this form? [Y/n/q]

> y

Edit POST data [default:
searchword=&task=search&option=com_search&Itemid=101] (do you want to fill
blank fields with random values? [Y/n] y
```

```
[09:38:14] [INFO] using '/root/.sqlmap/output/results-11152019_0938am.csv' as the CSV results file in multiple targets mode

sqlmap got a 303 redirect to 'http://192.168.0.2:80/index.php/component/search/?searchword=swQL&searchphrase=all&Itemid=101'. Do you want to follow? [Y/n] y

redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] y

[09:38:19] [ERROR] unable to retrieve page content, skipping to the next form

[#2] form:

POST http://192.168.0.2/index.php/author-login?task=user.login

POST data: username=&password=&remember=yes&return=&d12da3a2daaae07b9a39b1a9c6f01d8d=1

do you want to test this form? [Y/n/q]

> y

Edit POST data [default: username=&password=&remember=yes&return=&d12da3a2daaae0do you want to fill blank fields with random values? [Y/n] y

sqlmap got a 303 redirect to 'http://192.168.0.2:80/index.php/author-login'. Do you want to follow? [Y/n] y

redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] y

[09:38:26] [WARNING] the web server responded with an HTTP error code (404) which could interfere with the results of the tests

[09:38:26] [INFO] checking if the target is protected by some kind of WAF/IPS

[09:38:27] [INFO] testing if the target URL content is stable

[09:38:27] [WARNING] POST parameter 'username' does not appear to be dynamic

[09:38:27] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
```

Next Install burp if applicable
…

In Parrot we start directly Burp suite and then give the order
nikto -h test_server_name -useproxy http://localhost:8080/

In the parrot I didn't have Sparta installed so
git clone https://github.com/SECFORCE/sparta

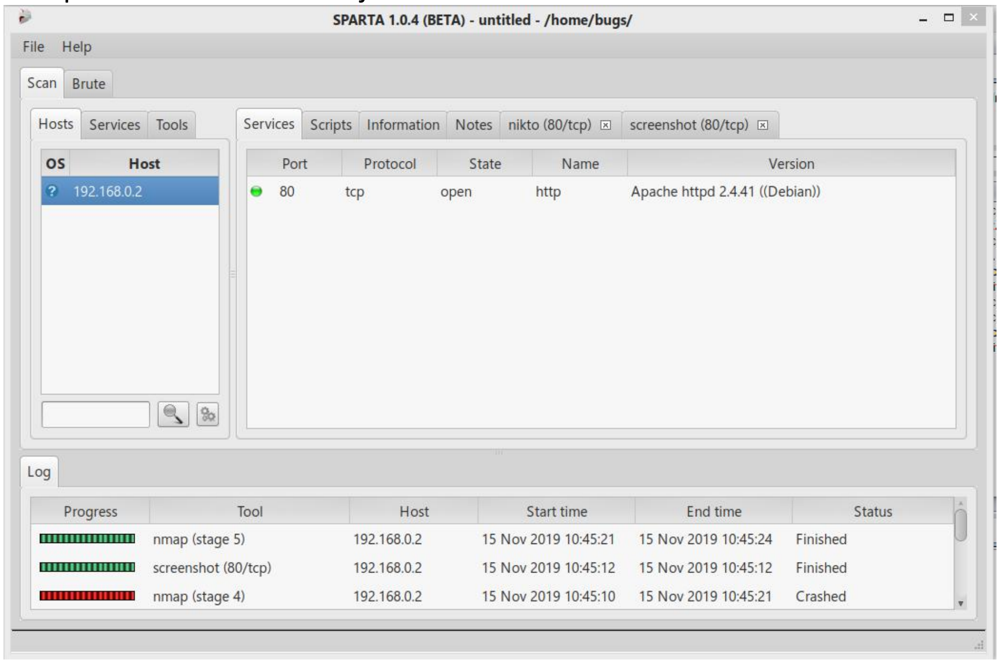
cd sparta

apt-get install python-elixir python-qt4 xsltproc

apt-get install ldap-utils rwho rsh-client x11-apps finger

Then
python sparta.py

Add the ip of the station where joomla is located

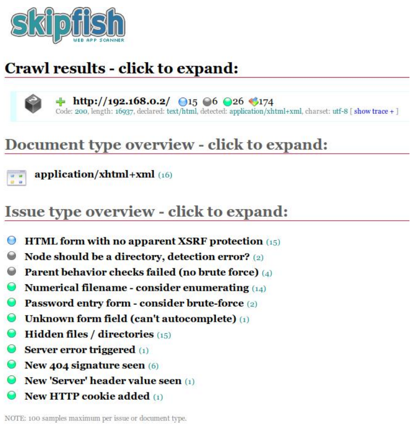


OS	Host
?	192.168.0.2

Port	Protocol	State	Name	Version
80	tcp	open	http	Apache httpd 2.4.41 ((Debian))

Progress	Tool	Host	Start time	End time	Status
🟩	nmap (stage 5)	192.168.0.2	15 Nov 2019 10:45:21	15 Nov 2019 10:45:24	Finished
🟩	screenshot (80/tcp)	192.168.0.2	15 Nov 2019 10:45:12	15 Nov 2019 10:45:12	Finished
🟥	nmap (stage 4)	192.168.0.2	15 Nov 2019 10:45:10	15 Nov 2019 10:45:21	Crashed

Looks like the parrot is “quieter” than the one cent(OS).
skipfish -o 202 http://192.168.0.2
And the report



Crawl results - click to expand:

http://192.168.0.2/ 15 6 26 174
Code: 200, length: 16937, declared: text/html, detected: application/xhtml+xml, charset: utf-8 [show trace +]

Document type overview - click to expand:

application/xhtml+xml (16)

Issue type overview - click to expand:

- HTML form with no apparent XSRF protection (15)
- Node should be a directory, detection error? (2)
- Parent behavior checks failed (no brute force) (4)
- Numerical filename - consider enumerating (14)
- Password entry form - consider brute-force (2)
- Unknown form field (can't autocomplete) (1)
- Hidden files / directories (15)
- Server error triggered (1)
- New 404 signature seen (6)
- New 'Server' header value seen (1)
- New HTTP cookie added (1)

NOTE: 100 samples maximum per issue or document type.

Now use netdiscover to pick up the network structure from the lab.
See if there is something interesting and on the
https://www.blackhillsinfosec.com/projects/ Recon-ng
- no longer has modules on the internet in the new github so you waste time
Maltego community is free but does not go capcha in the indicated machine (so you better install everything). After incomplete analysis with skipfish
skipfish -o 202

Resulting the following partial report .... see you

Metasploit
If it installs manually where you want
1. Start postgresql: service postgresql start
2. Automatically activate postgresql : update-rc.d postgresql enable
3. MSF database initialization: msfdb init
4. Msfconsole Execution: msfconsole
5. 5. Check the initialization of the database in the msfconsole console: > db_status

If you're in Kali at most you activate the service to boot

Beef
for debian
sudo apt-get install ruby ruby-dev

gem install bundler

git clone https://github.com/beefproject/beef

cd beef
./install.sh
apt install npm sqlite3-doc
./update-geoipdb
./generate-certificate
sudo ./update-beef
sudo ./update-geoipdb
nano config.yaml
Change the default password from beef to student
Then from the internet navigator you go to the http://localhost:3000/ui/authentication
Where you enter with beef/student
Now you can use BeEF hook which is a java script used for drilling and operating servers.
Beef is a powerful tool that can find a lot of information about a web server it has penetrated. Allows even a further phase of its operation by the execution of additional modules Testing the hook can be done locally with
http://(address of the station located with ip a):3000/hook.js.
To play again you can also use
http://localhost:3000/demos/butcher/index.html
I think now the need for some block-era script becomes clearer

Armitage
Installation ... as for dummies..
apt install armitage
Starting will require that metasploit must be active so ....
New console in which we launch service postgresql start
I check that everything is okay
service postgresql status
Then launch metasploit
Msfconsole
Then in another console or shortcut created manually by looser launches Armitage
You may look a little through what can do (dominant is a graphical interface for metasploit)
In campus or home after disconnecting the cable at isp you can do a deep scan with armitage (actually also with the old nmap)
HOSTS > NMAP Scan > Intense
E.g. 192.168.1.0/24 – for the home network
Then armitage will display what he discovered and you can move on to the next steps of kill chain

Interesting is the automation of attack of a site called Hail Mary play a little on an innocent site like students in the campus. As usual any automation that is not under your control (i.e. to write it yourself and you have experience) is detectable because of the network noise (flood ddos etc.) that it does.

Homework
Raise a web server under what you want xamaris, iis, apache etc. and analyze it attack it etc possibly take control using metasploit or armitage. You are not limited to the instruments in this laboratory. Any pentest distribution has a lot more included so test as many as you can and report what goes and where. It will use you later to quickly choose the best audit tool based on the situation on the battlefield.