

Laborator 8

Instrumente pentru auditarea securității aplicațiilor web

În cadrul acestui laborator vom analiza o serie de instrumente și sisteme de operare dedicate în realizarea pentest-ului

Nu vă este permis să aplicați aceste instrumente pe nici un site al nimănui (pentru testări veți utiliza numai computerele personale în funcție de echipa roșie sau albastră din care faceți parte.

Este evident că un profesionist nu va pierde vremea cu așa ceva ci își va instala instrumentele necesare în distribuția favorită și oricum le va modifica conform experienței și necesităților lui.

Pentru începători însă aceasta este abordarea preferată.

Unul dintre cele mai utilizate sisteme de operare în domeniu este Kali (bazat pe Debian) care vine preconfigurat cu o colecție de instrumente utilizate în pentest. Acesta poate fi instalat ca atare sau executat ca mașină virtuală.

Pentru instalare reală se va folosi: <https://www.kali.org/downloads/>

Pentru instalare în mașina virtuală este recomandat pentru cei cu experiență în Linux să se utilizeze tot imaginea originală. După pornirea mașinii primul lucru care se face la nivel de consolă este aducerea la zi a OS și aplicații astfel

```
apt update
```

```
apt upgrade
```

Acest lucru este necesar atât din motive de securitate cât și din motive de bună funcționare. În tipul operației de upgrade citiți cu atenție ce sunteți întrebați și nu dați selecții după ureche. (Vă duceți pe internet și citiți până pricepeți). Reluați procesul de câte ori este necesar.

De asemenea nu ar strica să vedeți ce utilizatori mai există și să schimbați parola de root cu una personalizată. Dacă este lansat în mașină virtuală adaptați configurația inițială a mașinii virtuale în conformitate cu resursele sistemului din dotare.

Ceva bucatărie...

```
apt install mc
```

```
apt install doublecmd-qt
```

```
apt install xfce4-goodies
```

Pentru capturi de ecran rapide: xfce4-screenshooter și creați pe desktop un link

În Kali nu merge direct deci mergem la consolă

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

Instalați manual instrumentele VMWare pentru respectiva mașină peste cele open source deja preinstalate în mașina virtuală. Acest lucru vă va da interoperabilitate gen shared directories cu gazda și copy/paste.

După cum am discutat la curs prima fază este adunare de informații despre entitatea țintă. Pentru început vom prezenta un exemplu simplu de footprinting pentru site-uri web. Pentru aceasta vom folosi nikto din dotare. Un *man nikto* nu strică la casa omului. Citiți cu atenție!

Acum vor urma o serie de teste asupra unui blog cu Joomla. Dat fiind că studenții sunt destul de subțiri în instalări Linux vom prezenta tot ciclul de instalare și testare. Pentru teste am utilizat Parrot la zi cu noutățile

Pentru testarea aplicațiilor din acest laborator aveți două opțiuni:

1. Aplicați tool-urile pe nginx-ul deja pus la dispoziție pe mașină de docker
2. În kali din docker sau pentru cei care preferă geamurile în versiunea de mașină virtuală se va instala un blog după cum urmează:

Instalam Joomla

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

Apache 2 este deja in parrot

```
sudo systemctl start apache2
sudo systemctl enable apache2
dpkg --get-selections | grep php | awk '/^ii/{ print $2}'
```

Se vede ca Php 7 este deja în parrot dar verific cu

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

Si se vede ca extensiile de mysql si mysql pt php nu sunt deci

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

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

Si vad daca comply cu ...

```
memory_limit = 256M
upload_max_filesize = 32M
post_max_size = 32M
date.timezone = ce vreti voi
```

La parrot este ok

Apoi instalăm mariodb

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

Și îl pornim inclusiv la boot

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

Un pic de configurare a securității pt mariodb

```
sudo mysql_secure_installation
```

Dacă avem probleme cu parola o resetăm

Parrot mysql / mariodb passwd reset

```
service mysql stop
mysqld_safe --skip-grant-tables &
```

Și apăsați de două ori Enter

```
mysql -u root mysql
```

Apoi

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

exit

```
service mysql restart
```

Creăm database pentru joomla

```
mysql -u root -p
```

Acum noua parola va merge și suntem în prompter de mariodb

Creăm noua bază date pt joomla

```
CREATE DATABASE joomla_db;
```

Acum creăm un user

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

Acum dăm drepturi

```
GRANT ALL PRIVILEGES ON joomla_db.* TO 'joomla'@'localhost' IDENTIFIED BY 'password'
WITH GRANT OPTION;
FLUSH PRIVILEGES;
Exit
```

Acum să instalăm Joomla

O aducem

```
wget
```

```
https://github.com/joomla/joomla-cms/releases/download/3.7.3-rc1/Joomla\_3.7.3-rc1-Release\_Candidate-Full\_Package.tar.gz
```

```
sau puteți încerca cu https://github.com/joomla/joomla-cms
```

Puteți verifica dacă nu a apărut o versiune mai nouă și modificați corespunzător

```
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
```

Dăm drepturi corecte

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

```
sudo chmod -R 750 /var/www/html/joomla
```

Apoi creăm o gazdă virtuală în Apache pentru Joomla prin crearea unui fișier de configurare corespunzător

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

Care va conține următoarele (puteți să modificați dacă înțelegeți ce faceți)

```
<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>
```

Acum se dezactivează gazda implicită și se activează cea pentru Joomla

```
sudo a2dissite 000-default
```

```
sudo a2ensite joomla
```

Acum se reîncarcă Apache

```
sudo systemctl restart apache2
```

Instalam ufw - o minimă protecție

```
apt install ufw
```

```
sudo ufw enable
```

Deoarece avem ufw-ul acesta trebuie configurat să-i dea drumul pt Joomla

```
sudo ufw allow http
```

Configurările pot fi editate direct și cu

```
nano /var/www/html/joomla/installation/configuration.php-dist
```

Pentru probleme vă puteți uita în fișierele de log /var/log/apache2/

Și ar trebui să găsim la nume de domeniu sau ip stație joomla
Cu nume_site/administrator se intră în backend-ul de administrare al joomla

Configuration.php-dist aici trebuie pus măcar nume utilizator, nume baza de date și parola
apt install php-xml pentru joomla în parrot

Se face instalarea din frontend-ul implicit și gata

Și acum să vedem rezultatele analizei cu nikto înainte de completarea procesului de instalare din
frontend-ul web

```
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
```

Retestăm cu nikto după terminarea procesului de instalare

```
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 conf 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
```

**Deja se observă majore "îmbunătățiri" în securitatea blog-ului nu?
Deci !!!! nu mai instalați/modificați și atât trebuie să apară în ciclul de mentenanță a unui subiect
web enabled și retestarea cu instrumentele dedicate!!!!**

Un exemplu de nikto pe o mașină parrot cu 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 conf 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
```

Acum utilizam un tool pentru analiza softului de baze de date

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

Apoi cu răbdare efectuați toate testele recomandate de tool - evident că trebuie să citiți în plus ca să înțelegeți unele din problemele raportate sau analizele făcute. Asta dacă nu știți deja de la cei care v-au învățat să proiectați sisteme de acest tip. Mai jos aveți un scurt extras din log-ul aplicației:

```
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

Instalare burp dacă este cazul

...

În Parrot pornim direct Burp suite și apoi dăm comanda
nikto -h nume_server_test -useproxy http://localhost:8080/

În parrot nu am sparta așa că

Git clone <https://github.com/SECFORCE/sparta>

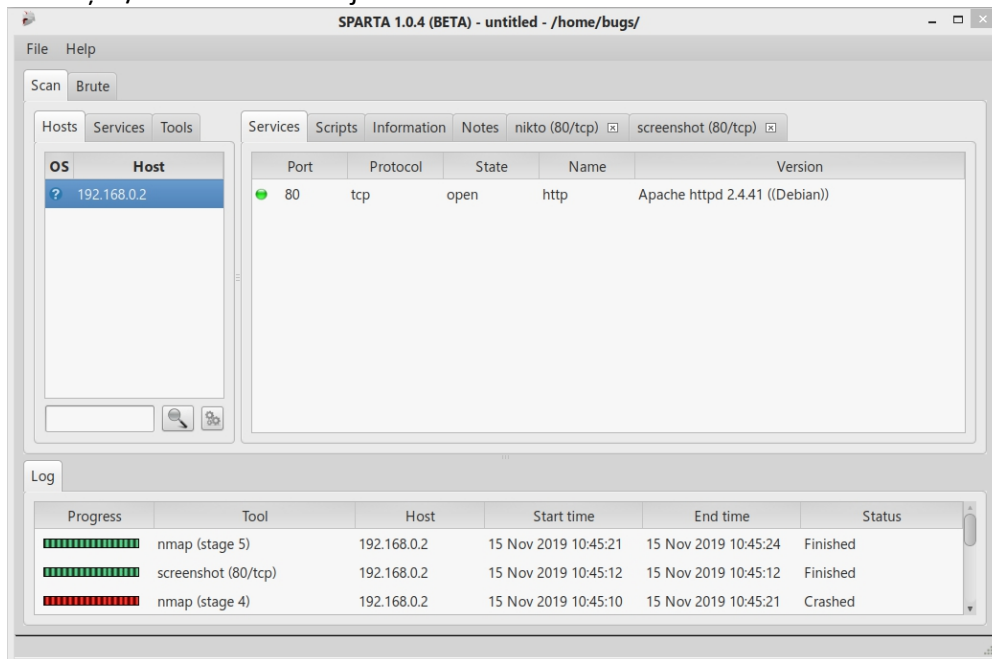
cd sparta

```
apt-get install python-elixir python-qt4 xsltproc  
apt-get install ldap-utils rwho rsh-client x11-apps finger
```

Apoi

python sparta.py

Adaugați ip-ul stației/mv unde se află joomla



Se pare ca papagalul este mai tăcut decât centișorul

skipfish -o 202 <http://192.168.0.2>

Și raportul



Crawl results - click to expand:

<http://192.168.0.2/> 15 6 26 174
Code: 200, length: 1997, declared: text/html, detected: application/xhtml+xml, charset: utf-8 [show trace +]

Document type overview - click to expand:

application/xhtml+xml (10)

Issue type overview - click to expand:

- HTML form with no apparent CSRF 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.

Utilizați netdiscover pentru a ridica structura de rețea din laborator

Vedeți dacă nu este ceva interesant și pe la <https://www.blackhillsinfosec.com/projects/>

Recon-ng - nu mai are module pe internet în noul github deci pierdeți vremea

Maltego community este gratuit dar nu merge captcha în mașina indicată (deci tot mai bine instalați la mânăuță tot).

În urma analizei incomplete cu skipfish

```
skipfish -o 202
```

A rezultat următorul raport parțial

Metasploit

Dacă se instalează manual unde vrei

1. Pornire **postgresql**: service postgresql start
2. Activare automată **postgresql** : update-rc.d postgresql enable
3. Inițializare baza de date MSF: msfdb init
4. Execuție msfconsole: msfconsole
5. Verificarea inițializării bazei de date din consola msfconsole: > db_status

Dacă ești în Kali cel mult activezi serviciul la boot

Beef

Pentru debian normal

```
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
```

Schimbați parola implicită din beef în student

Apoi din navigator internet va duceți la <http://localhost:3000/ui/authentication>

Unde intrați cu beef/student

Acum se poate utiliza BeEF hook care este un script java utilizat pentru găurirea și exploatarea serverelor. Beef este un instrument puternic care poate găsi multe informații despre un server web pe care l-a penetrat. Permite chiar și o fază de exploatare ulterioară a acestuia prin execuția unor module suplimentare

Testarea hook-ului se poate face local cu [http://\(adresa statiei aflata cu ip a\):3000/hook.js](http://(adresa statiei aflata cu ip a):3000/hook.js).

Pentru a va mai juca puteți utiliza și

<http://localhost:3000/demos/butcher/index.html>

Cred că acum necesitatea unor script block-ere devine mai clară

Armitage

Instalarea ... ca la năcăjiți...

apt install armitage

Pornirea necesită metasploit-ul activ deci

O consolă în care lansăm

service postgresql start

Verific că totul este în regulă

service postgresql status

Lansez metasploit

Msfconsole

Apoi în alta consolă sau din shortcut creat manual de looser se lansează

Armitage

Va uitați puțin prin ceea ce poate face (dominant este o interfață grafică pentru metasploit)

În cămin sau acasă după ce deconectați cablul de la isp puteți face o scanare deep cu armitage (de fapt tot cu bătrânul nmap)

HOSTS > NMAP Scan > Intense

De ex 192.168.1.0/24 – pentru rețeaua de acasă

Apoi armitage va afișa ce a descoperit și puteți trece la pașii următori din kill chain

Interesantă este automatizarea de atac a unui site numita Hail Mary jucați-vă puțin pe un site nevinovat gen al studenților din cămin. Ca de obicei orice automatizare care nu este sub controlul tău (adică să o scrii tu și să ai experiență) este detectabilă datorita zarvei (flood ddos etc) pe care o face.

Tema pe acasa

Ridicați un server web sub ce vreți voi xamaris, iis, apache etc și analizați-l atacați-l etc eventual preluați controlul utilizând metasploit sau armitage. Nu sunteți limitați numai la instrumentele din acest laborator. Orice distribuție de pentest are mult mai multe incluse deci testați cât de multe puteți și faceți raport ce merge și unde. Vă va folosi mai încolo pentru a alege rapid cel mai bun tool de auditare în funcție de situația de pe câmpul de luptă.