Assignment 2.1

Name: Web Server Vulnerabilities

Developers:

- Anders Carlsson
- Oleksii Baranovskyi

Performer:

- FB-01 Sakhnii Nazar

Table of Contents

Task 1. Web Server Vulnerabilities Detection

Purpose: understand how to use browsing in web-server vulnerabilities detection, how to use network scanning tools for detect web-server vulnerabilities.

After the work the student must

- know: what is web surfing, how it could be performed;
- be able to: analyze results of HTTP surfing to detect web-server vulnerabilities, analyze the results of network scanning for detection of web-server vulnerabilities

Tasks:

- analyze provided web-server on virtual machine 192.168.56.3, check its' parameters, analyze headers, perform surfing, perform network scanning

Technical equipping of the workplace:

- nmap
- dirb
- dirbuster
- Vega
- OWASP ZAP
- OWASP Burp Suite
- Browser Developer Tools.

Solution:
Run web vulnerability scanners in different modes in order to search and detect vulnerabilities, establish their reasons and learn about possible countermeasures.

TASK 1

Which vulnerabilities provided web server has? Prove it with screenshot.

Answer:

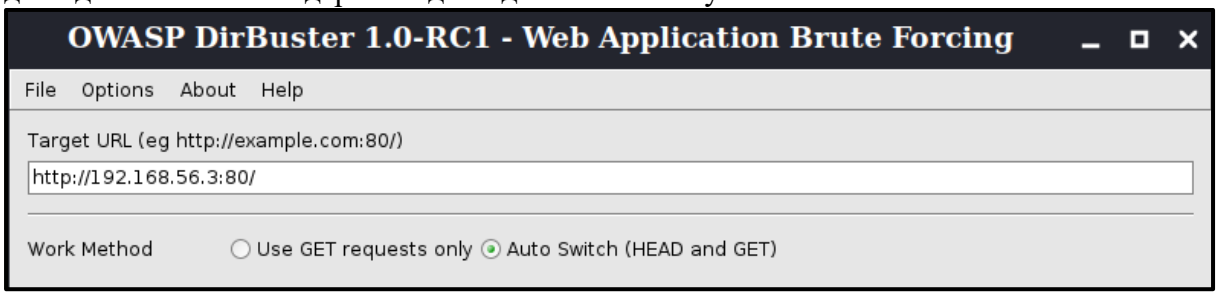
Для початку проведемо просте загальне мережеве сканування, щоби визначити можливі порти, на яких було розгорнуто веб-сервер ↓

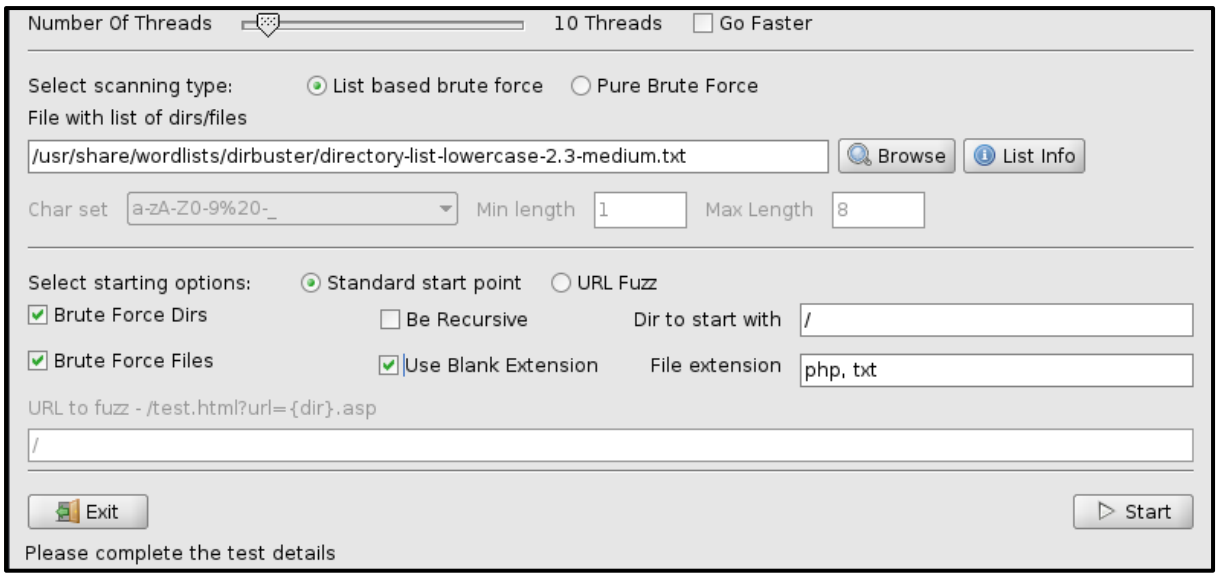
```
┌──(nazar㉿snz24)-[~]
└─$ nmap 192.168.56.3
Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-12 17:43 EET
Nmap scan report for 192.168.56.3
Host is up (0.0043s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.84 seconds
```

Звідси, знову скористаємося утилітою **nmap**, для того щоб провести, так зване, “aggressive” (повноцінне) сканування обраних портів відповідної цільової адреси ↓

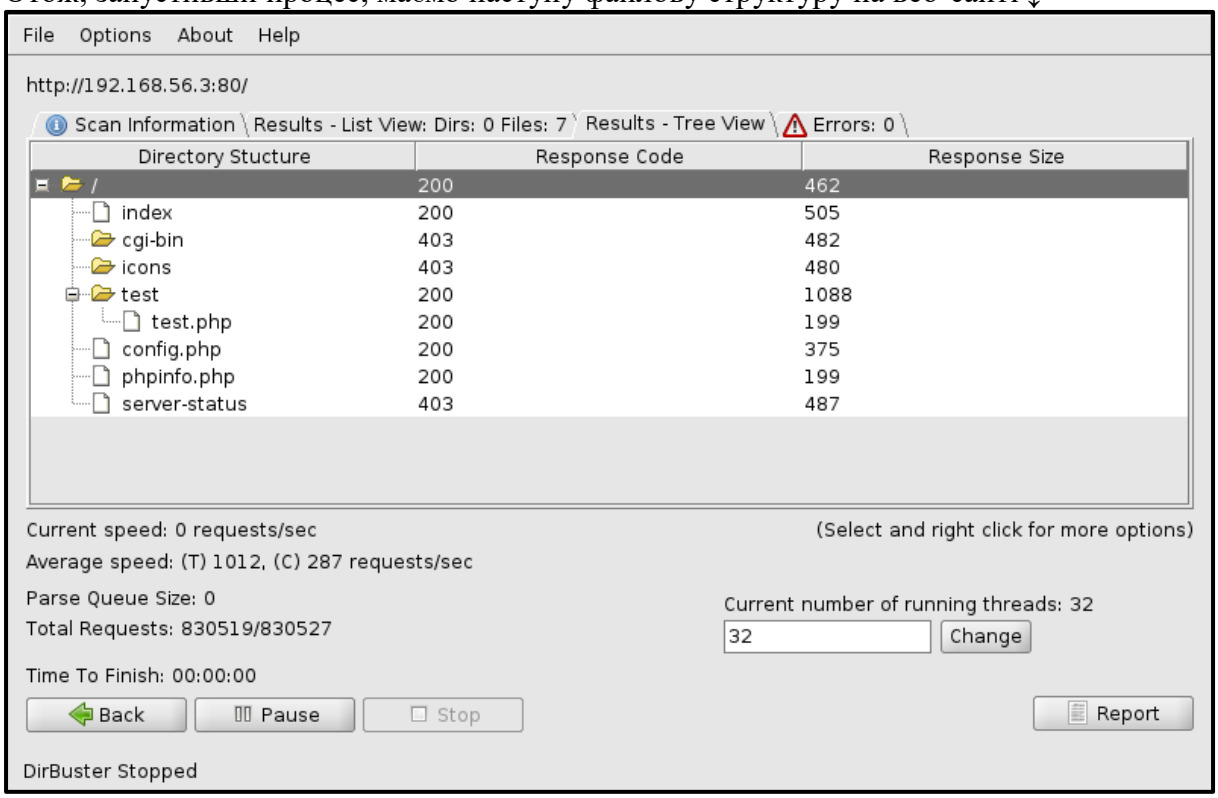
```
┌──(nazar㉿snz24)-[~]
└─$ nmap -A -T 4 -p 22,80,443 192.168.56.3
Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-12 17:57 EET
Nmap scan report for 192.168.56.3
Host is up (0.0025s latency).

PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 6.0p1 Debian 4+deb7u4 (protocol 2.0)
| ssh-hostkey:
|   1024 71:83:d0:c2:7e:a4:62:02:d6:55:ea:2b:80:16:3a:bb (DSA)
|   2048 5c:ac:53:9c:ec:fd:da:cc:b7:63:b8:ac:c4:ab:41:3f (RSA)
|_  256 ca:54:b4:e9:cf:a1:12:f7:b5:79:6a:2c:13:84:5f:01 (ECDSA)
80/tcp  open  http     Apache httpd 2.2.22 ((Debian))
|_http-server-header: Apache/2.2.22 (Debian)
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/http Apache httpd 2.2.22 ((Debian))
|_http-server-header: Apache/2.2.22 (Debian)
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=DIDA/organizationName=BTH/stateOrProvinceName=Blekinge/countryName=SE
| Not valid before: 2020-08-26T07:30:05
|_Not valid after:  2021-08-26T07:30:05
|_ssl-date: 2023-03-02T12:06:22+00:00; -10d03h51m16s from scanner time.
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: -10d03h51m16s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.77 seconds
```

На даний момент, уже відомо про відкриті порти та відповідні сервіси, такі як `ssh:22`, `http:80`, `https:443`, а також знаємо про ОС та сам веб-сервер, на якому був розгорнутий цільовий онлайн-ресурс. Тому далі попрацюємо з **dirbuster**, щоб трохи дослідити каталогове дерево відповідного веб-сайту.

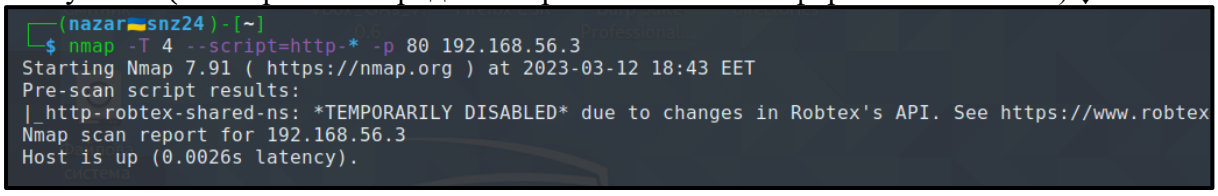




Отож, запустивши процес, маємо наступу файлову структуру на веб-сайті ↓



Хм, різні файли та директорії… До речі, ось що цікавого знайшло `--script=http-*` сканування (на зображенні продемонстровані лише найінформативніші звіти) ↓



```
PORT   STATE SERVICE
80/tcp open  http
|_http-apache-negotiation: mod_negotiation enabled.
| http-brute:
|_  Path "/" does not require authentication
|_http-chrono: Request times for /; avg: 28.05ms; min: 13.33ms; max: 49.63ms
|_http-comments-displayer: Couldn't find any comments.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-date: Thu, 02 Mar 2023 12:52:36 GMT; -10d02h51m18s from local time.
|_http-devframework: Couldn't determine the underlying framework or CMS. Try increasing 'httpspider.max
.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /test/: Test page
|_  /phpinfo.php: Possible information file
|_http-errors: Couldn't find any error pages.
```

.....

```
| http-vhosts:
|_128 names had status 200
|_http-xssed: No previously reported XSS vuln.

Nmap done: 1 IP address (1 host up) scanned in 1811.08 seconds
```

Але окрім файлу налаштувань `/php-config.php` та `/test/test.php`, справді цікава інформація знаходиться у файлі `/config.php`, а саме облікові дані для входу на віддаленому веб-сервері, на якому було розгорнуто цей веб-сайт.

Відкрити

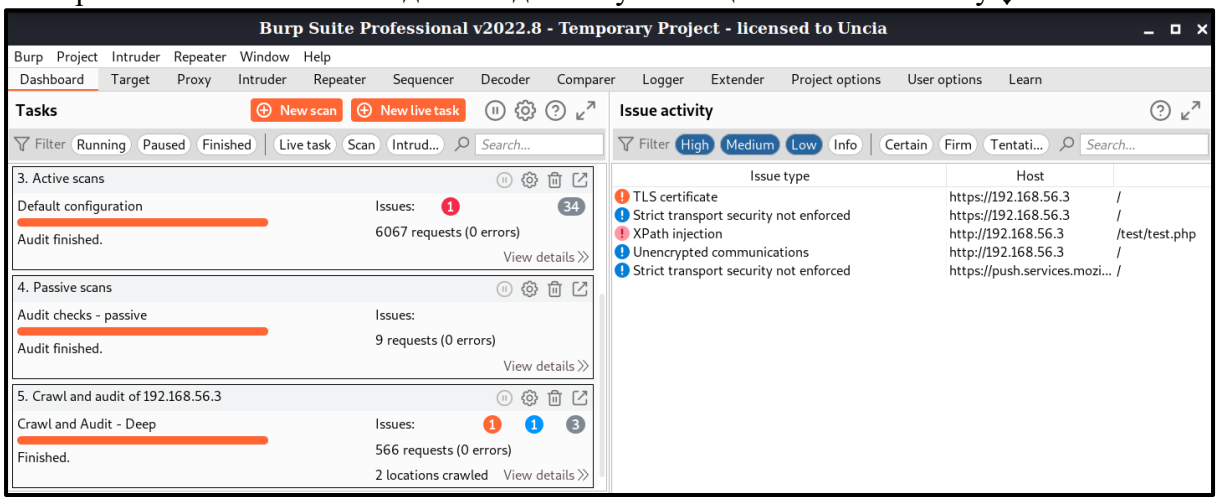
config.php
~/Завантажене

```
1 <?
2         set("user","lab");
3         set("password","lab123@");
```

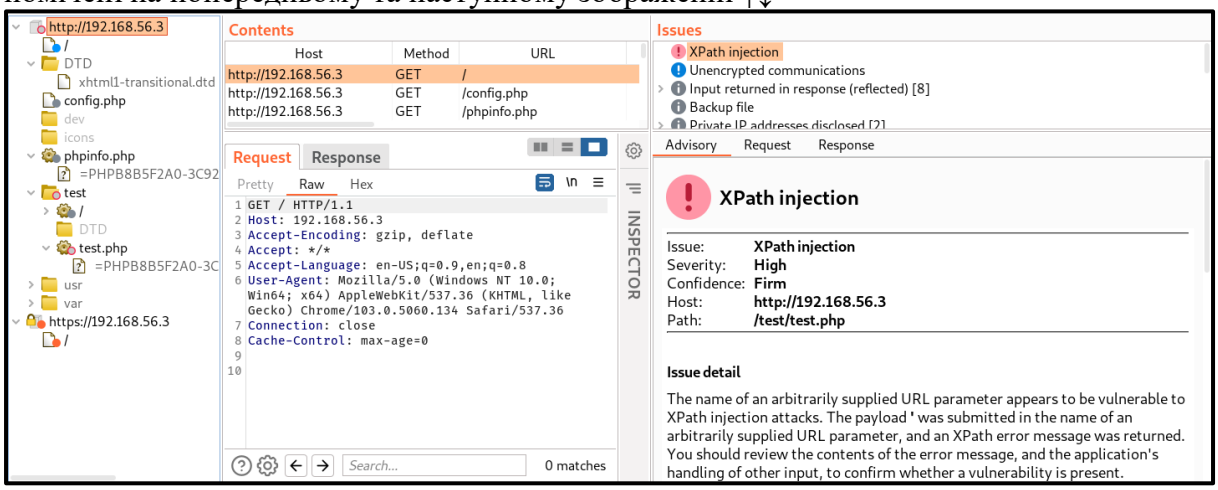
А далі уже, без проблем, заходимо на сервер і робимо що завгодно ツ

```
Debian GNU/Linux 7 lab tty1

Hint: Num Lock on

lab login: lab
Password:
Last login: Thu Mar  2 05:28:25 EST 2023 on tty1
Linux lab 3.2.0-4-amd64 #1 SMP Debian 3.2.78-1 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
lab@lab:~$ whoami
lab
lab@lab:~$ pwd
/home/lab
lab@lab:~$ cd /var/www/
lab@lab:/var/www$ ls -la
total 28
drwxr-xr-x  4 root root 4096 Aug 26  2020 .
drwxr-xr-x 12 root root 4096 Aug 26  2020 ..
-rw-r--r--  1 root root   51 Aug 26  2020 config.php.old
drwxrwxrwx  3 root root 4096 Aug 26  2020 icons
-rw-r--r--  1 root root  177 Aug 26  2020 index.html
-rw-r--r--  1 root root   18 Aug 26  2020 phpinfo.php
d---rwxrwx  2 root root 4096 Aug 26  2020 test
lab@lab:/var/www$ cat ./config.php.old
<?
        set("user","lab");
        set("password","lab123@");
lab@lab:/var/www$
```

Коротше, маємо як би вже доступ до сторони сервера, тому тут ще, хіба що можна було б спробувати провести ескалацію `root`-привілеїв, проте вже якось іншим разом. А зараз ще повивчаємо базові можливості **`Burp Suite Professional`**, тобто для пошуку веб-вразливостей виконаємо деякі види сканування цільового веб-сайту ↓



Ну ось і маємо, що на перший погляд простенький сайт, на основній веб-сторінці якого стоїть "`default web-page`", має декілька веб-вразливостей, які були знайдені та помічені на попередньому та наступному зображенні ↑↓



TASK 2

What are the reasons of detected vulnerabilities?

Answer:

Загалом, вразливості на веб-сайті можуть виникати з різних причин, таких як помилки при проектуванні та розробці, необережність при налаштуванні серверів, сторонніх бібліотек або фреймворків, а також використання застарілих або не оновлених компонентів. Хоча взагалі-то можна переглянути главу "**`Issue background`**" у вкладці опису вразливості, проте додатково перепишемо їх сюди.

Отже, ось можливі причини деяких вразливостей, які були знайдені при скануванні:

- Strict transport security not enforced – ця вразливість може бути наслідком невірно налаштованого сервера або відсутності встановленого HTTPS-з'єднання на веб-сайті. У такому випадку, даний веб-сайт може бути піддається MITM-атакам (Man-In-The-Middle), де зловмисник може перехоплювати трафік та отримувати доступ до конфіденційної інформації.

- XPath Injection – ця вразливість зазвичай виникає, коли вхідні дані не валідуються на стороні сервера, тому зловмисники можуть використовувати спеціальні символи та запити, щоб отримати доступ до конфіденційної інформації або виконати шкідливі дії на веб-сайті.

- Unencrypted communications – ця вразливість може виникнути, коли дані не шифруються під час передачі між веб-сервером та клієнтом, тому зловмисники можуть перехоплювати трафік та отримувати доступ до конфіденційної інформації.

- Private IP address disclosed – ця вразливість може виникнути, коли веб-сайт відображає конфіденційну інформацію, таку як приватний IP-адреса, на веб-сторінках або в повідомленнях про помилку. Це може допомогти зловмисникам відшукати вразливості у мережевій інфраструктурі.

- Backup file – ця вразливість може виникну ти, коли на веб-сайті залишаються резервні копії файлів, які можуть містити конфіденційну інформацію, таку як паролі або особисті дані. Якщо ці файли не захищені від доступу, зловмисники можуть отримати доступ до цієї інформації та використати її для атак на веб-сайт або інших послуг.

TASK 3
Could you provide countermeasures for detected vulnerabilities?

Answer:
В основному можливі контрзаходи для захисту веб-сторінки, що працює за протоколом HTTP, включають встановлення оновлень програмного забезпечення, використання мережевого екрану, застосування безпеки на рівні коду, захист від SQL-ін'єкцій та XSS-атак, а також відмову від застарілих технологій та сервісів, що можуть бути вразливими до атак. Аналогічно, щоб детальніше переглянути інформацію про запобіжні заходи, можна прочитати главу "**Issue remediation**" у вкладці опису до кожної вразливості.

Отож, все одно додатково випишемо можливі контрзаходи для деяких попередньо уже згаданих вразливостей:

- "Strict transport security not enforced": Цю вразливість можна виправити, додавши заголовок "HTTP Strict Transport Security (HSTS)" до веб-сайту, що забезпечить, що всі з'єднання між клієнтом і сервером будуть зашифровані SSL/TLS. Це можна зробити, встановивши сертифікат SSL/TLS на веб-сервері та налаштувавши сервер так, щоб він вимагав захищеного з'єднання для всіх запитів.

- "XPath Injection": Щоб убезпечитися від цієї вразливості, потрібно виконувати валідацію та фільтрування вхідних даних перед використанням їх у запитах XPath. Краще використовувати параметризовані запити, які дозволяють ізольовано передавати вхідні дані в запит.

- "Unencrypted communications": Для захисту від цієї вразливості можна застосувати шифрування за допомогою протоколів SSL/TLS. Всі комунікації між клієнтом і сервером повинні бути захищені SSL/TLS, зокрема використовувати HTTPS для передачі конфіденційної інформації.

- "Private IP address disclosed": Цю вразливість можна виправити, використовуючи публічні IP-адреси замість приватних IP-адрес, відображаючи власний IP-адрес сервера, що використовується для надання послуг. Також можна використовувати firewalls, які відфільтрують доступ до приватних IP-адресів з зовнішнього світу.

- "Backup file": Для убезпечення від цієї вразливості необхідно видаляти всі непотрібні резервні копії та не зберігати їх на веб-сервері або у відкритому доступі, а також необхідно шифрувати резервні копії та перевіряти їх регулярно на наявність вразливостей.