

ЛАБОРАТОРНА РОБОТА №2

ДОСЛІДЖЕННЯ ВЕБ-ЗАСТОСУНКУ НА ЕТАПІ АКТИВНОЇ ВЕБ-РОЗВІДКИ (WEB RECONNAISSANCE)

Мета роботи:

1. Ознайомлення з поняттям веб-розвідки (Web Reconnaissance) у межах етичного хакінгу та тестування на проникнення.
2. Дослідження веб-застосунку з метою визначення його технологічного стеку та доступних ресурсів.
3. Набуття практичних навичок дослідження веб-сервера та веб-застосунку за допомогою спеціалізованих інструментів розвідки.

Інструменти та ПЗ: Kali Linux, Nmap, cURL, whatweb, dirb, ffuf.

Теоретичні відомості

Розвідка веб-застосунків (Web Reconnaissance) – це процес активного збору інформації про цільовий веб-застосунок, що включає аналіз його інфраструктури, використовуваних технологій та потенційних вразливостей з метою підготовки до подальших етапів тестування на проникнення.

Методи розвідки поділяються на дві основні категорії:

1. Пасивна розвідка.
2. Активна розвідка.

Пасивна розвідка передбачає прихований збір інформації з загальнодоступних джерел без надсилання запитів до інфраструктури цільової системи (OSINT, веб-сайти компаній, пошукові системи, соціальні мережі тощо).

Активна розвідка передбачає безпосередню взаємодію з цільовою системою для отримання детальної технічної інформації. Під час активної розвідки здійснюється сканування мереж, відкритих портів і сервісів, а також взаємодія зі службами для виявлення потенційних слабких місць безпеки (збір технічних даних, визначення версій програмного забезпечення та неправильних конфігурацій тощо).

Розвідка веб-застосунків має низку особливостей, пов'язаних зі специфікою HTTP-протоколу та архітектурою веб-систем. На відміну від мережевої розвідки, веб-розвідка зосереджується не лише на доступності сервісів, а й на логіці роботи веб-застосунку, структурі ресурсів та способах взаємодії клієнта з сервером.

Під час веб-розвідки досліджуються:

1. Технологічний стек (визначення типу та версії веб-сервера, мови програмування, використовуваних CMS та фреймворків).
2. Аналіз HTTP-заголовків (аналіз відповідей сервера, наявність прапорів безпеки тощо).
3. Перебір директорій та файлів (пошук прихованих шляхів на сервері, які не мають прямих посилань на сайті)
4. Аналіз параметрів та вхідних точок (виявлення форм, полів вводу та параметрів URL, які можуть бути використані для подальших атак).

Процес веб-розвідки реалізується із використанням спеціалізованих інструментів, які дозволяють дослідити різні рівні взаємодії клієнта з веб-застосунком та сервером.

cURL

cURL (Client URL) – це утиліта командного рядка для передавання даних з використанням протоколів, зокрема HTTP та HTTPS. У контексті веб-розвідки `curl` використовується для ручного аналізу HTTP-взаємодії між клієнтом і веб-сервером, отримання заголовків відповіді, перевірки доступності ресурсів та дослідження конфігурації сервера.

Синтаксис команди `curl`:

`curl [параметри] [URL-адреса]`

Ключові параметри **cURL**:

Таблиця 1. Ключові параметри **cURL**

Параметр	Опис
-I	Отримання лише HTTP-заголовків без тіла відповіді

-v	Детальний режим, відображає повний обмін між клієнтом і сервером
-X	Явне задання HTTP-методу (GET, POST, PUT тощо)
-H	Додавання або модифікація HTTP-заголовків запиту
-A	Зміна User-Agent для аналізу поведінки сервера
-L	Автоматичне слідування за HTTP-редіректами

```

kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ curl -I https://learn.ztu.edu.ua/
HTTP/1.1 200 OK
Date: Thu, 05 Feb 2026 16:45:04 GMT
Server: Apache
Strict-Transport-Security: max-age=63072000; preload
Set-Cookie: MoodleSession=j2dciljfi3bekn1mb6riiopgq2; path=/
Expires: Mon, 20 Aug 1969 09:23:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Language: uk
Content-Script-Type: text/javascript
Content-Style-Type: text/css
X-UA-Compatible: IE=edge
Cache-Control: post-check=0, pre-check=0, no-transform
Last-Modified: Thu, 05 Feb 2026 16:45:04 GMT
Accept-Ranges: none
X-Frame-Options: sameorigin
Content-Type: text/html; charset=utf-8

```

Рисунок 1 – Приклад використання утиліти cURL

WhatWeb

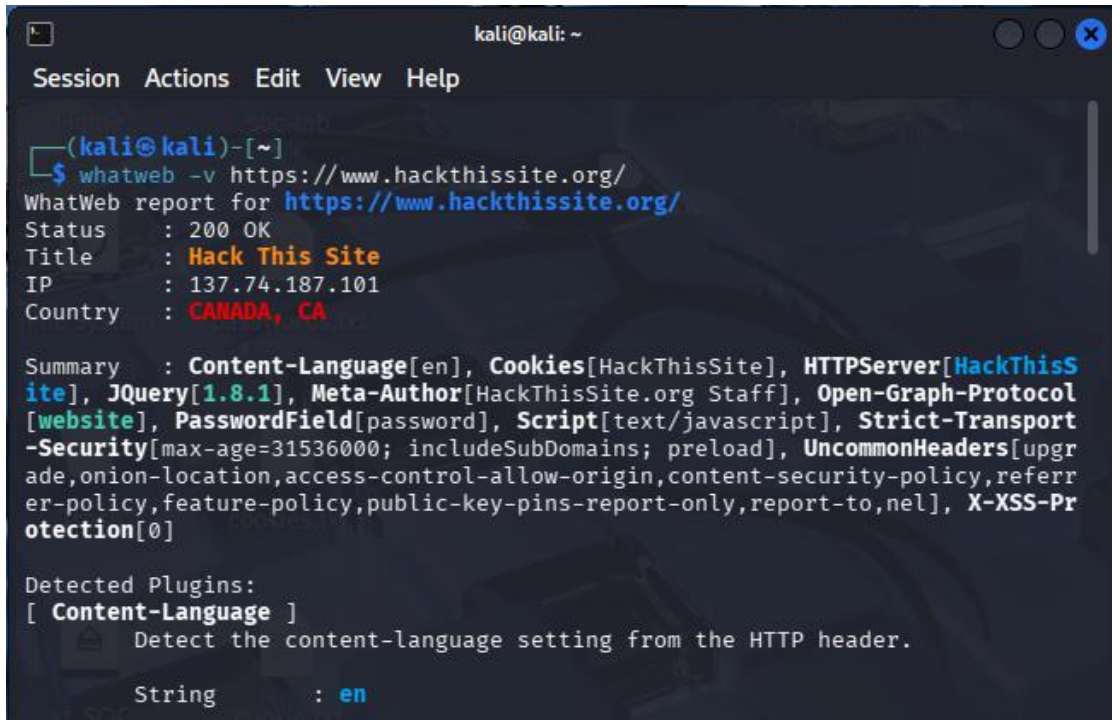
WhatWeb – це інструмент для ідентифікації веб-сайтів, призначений для визначення технологічного стеку веб-застосунків, зокрема CMS, JavaScript-бібліотеки, веб-сервери та аналітичні сервіси. Популярним графічним аналогом WhatWeb є браузерне розширення **Wappalyzer**, яке дозволяє миттєво візуалізувати технологічний стек сайту (CMS, фреймворки, аналітику) безпосередньо під час перегляду сторінки.

Синтаксис WhatWeb:

whatweb [параметри] [URL-адреса]

Перелік доступних параметрів можна отримати, виконавши команду:

whatweb -h



```
kali@kali: ~  
Session Actions Edit View Help  
  
(kali@kali)~  
$ whatweb -v https://www.hackthissite.org/  
WhatWeb report for https://www.hackthissite.org/  
Status      : 200 OK  
Title       : Hack This Site  
IP          : 137.74.187.101  
Country     : CANADA, CA  
  
Summary      : Content-Language[en], Cookies[HackThisSite], HTTPServer[HackThisSite], JQuery[1.8.1], Meta-Author[HackThisSite.org Staff], Open-Graph-Protocol[website], PasswordField[password], Script[text/javascript], Strict-Transport-Security[max-age=31536000; includeSubDomains; preload], UncommonHeaders[upgrade, onion-location, access-control-allow-origin, content-security-policy, referer-policy, feature-policy, public-key-pins-report-only, report-to, nel], X-XSS-Protection[0]  
  
Detected Plugins:  
[ Content-Language ]  
  Detect the content-language setting from the HTTP header.  
  
String       : en
```

Рисунок 2 - Приклад використання утиліти WhatWeb

dirb

dirb – це спеціалізований сканер веб-контенту, який використовує словникові атаки для виявлення існуючих або прихованих об'єктів на сервері шляхом аналізу отриманих відповідей. Інструмент містить вбудовані списки слів (wordlists), що дає змогу автоматизувати процес пошуку директорій і файлів.

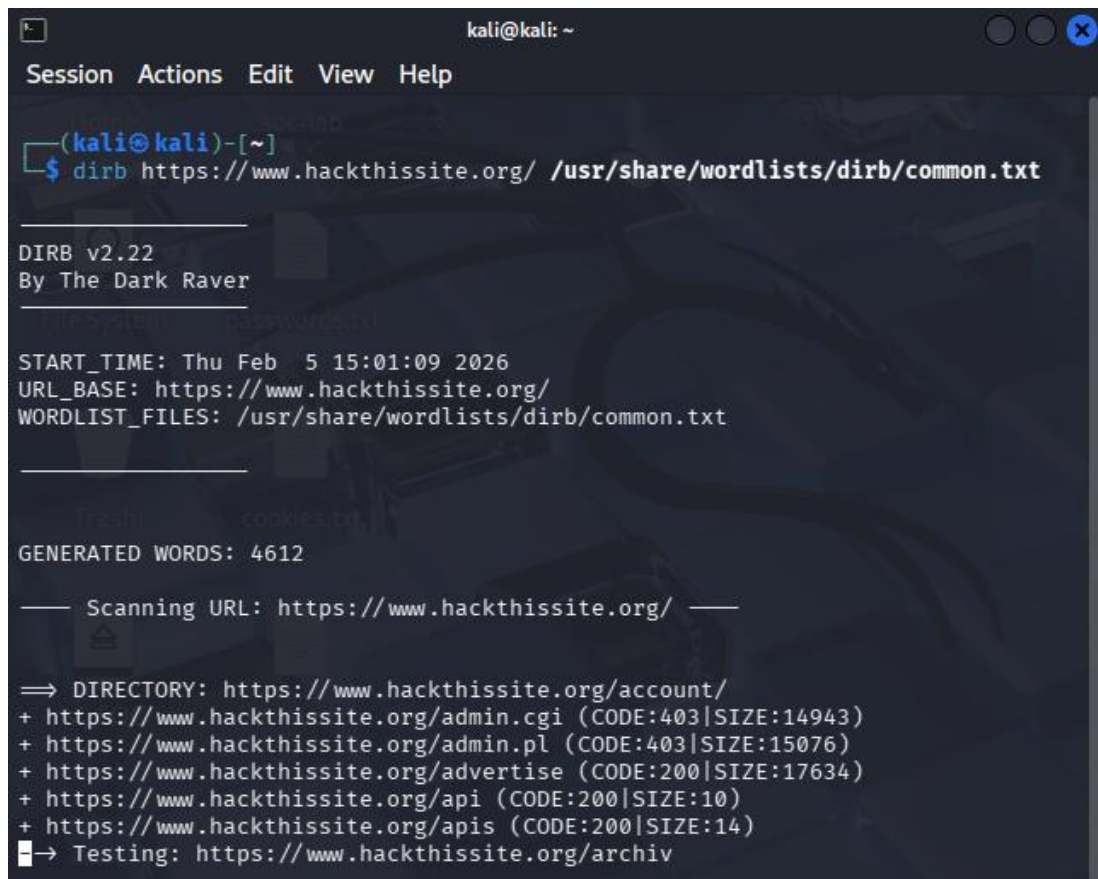
У навчальних цілях DIRB може використовуватися для ознайомлення з базовими принципами перебору директорій, однак на практиці доцільніше застосовувати сучасні інструменти, зокрема ffuf.

Синтаксис:

dirb [URL-адреса] [wordlist.txt]

Приклад використання:

dirb http://example.com /usr/share/wordlists/dirb/common.txt



```
kali@kali: ~  
Session Actions Edit View Help  
  
(kali@kali)-[~]  
$ dirb https://www.hackthissite.org/ /usr/share/wordlists/dirb/common.txt  
  
DIRB v2.22  
By The Dark Raver  
  
START_TIME: Thu Feb 5 15:01:09 2026  
URL_BASE: https://www.hackthissite.org/  
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt  
  
GENERATED WORDS: 4612  
  
— Scanning URL: https://www.hackthissite.org/ —  
  
=> DIRECTORY: https://www.hackthissite.org/account/  
+ https://www.hackthissite.org/admin.cgi (CODE:403|SIZE:14943)  
+ https://www.hackthissite.org/admin.pl (CODE:403|SIZE:15076)  
+ https://www.hackthissite.org/advertise (CODE:200|SIZE:17634)  
+ https://www.hackthissite.org/api (CODE:200|SIZE:10)  
+ https://www.hackthissite.org/apis (CODE:200|SIZE:14)  
=> Testing: https://www.hackthissite.org/archiv
```

Рисунок 3 - Приклад використання утиліти dirb

ffuf

ffuf – це швидкий інструмент для веб-фазингу, який дозволяє виконувати перебір директорій, файлів із різними розширеннями, віртуальних хостів (без використання DNS-записів), а також параметрів HTTP-запитів (GET і POST). Інструмент застосовується під час тестування безпеки веб-застосунків з метою виявлення прихованих ресурсів та потенційних вхідних точок.

Синтаксис:

ffuf -u [URL-адреса]/FUZZ -w [wordlist.txt] [-s] [-e <extensions>]

де:

- **-s** - тихий режим, у якому відображаються лише знайдені результати;
- **-e** - параметр для задання розширень файлів, які додаються до кожного значення зі словника.

Маркер **FUZZ** в інструменті ffuf використовується для позначення позиції в URL або HTTP-запиті, значення якої буде автоматично замінюватися словами зі словника під час перебору.

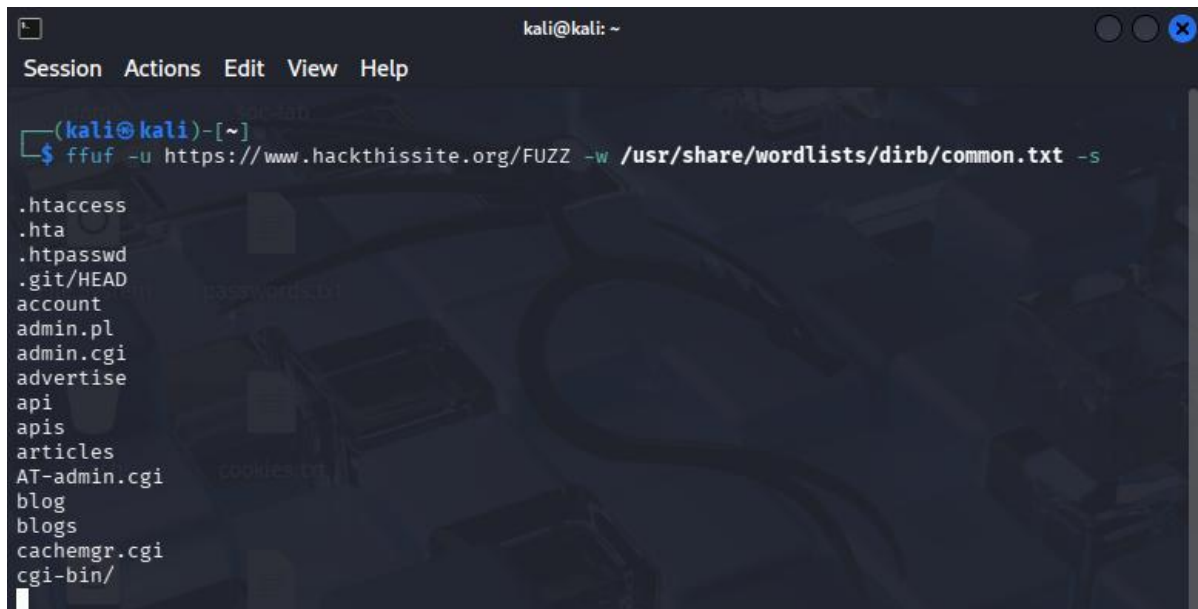
Таким чином, FUZZ визначає точку фазингу, у якій ffuf підставляє кожне значення зі словника для пошуку прихованих директорій, файлів або інших ресурсів веб-застосунку.

Приклад використання для пошуку прихованих директорій:

ffuf -u http://example.com/FUZZ -w /usr/share/wordlists/dirb/common.txt

Приклад перебору файлів з різними розширеннями:

ffuf -u http://example.com/FUZZ -w wordlist.txt -e .php,.txt,.sql,.html



```
kali@kali: ~  
Session Actions Edit View Help  
(kali@kali)-[~]  
$ ffuf -u https://www.hackthissite.org/FUZZ -w /usr/share/wordlists/dirb/common.txt -s  
.htaccess  
.hta  
.htpasswd  
.git/HEAD  
account  
admin.pl  
admin.cgi  
advertise  
api  
apis  
articles  
AT-admin.cgi  
blog  
blogs  
cachemgr.cgi  
cgi-bin/
```

Рисунок 4 - Приклад використання утиліти ffuf

Завдання на лабораторну роботу

Примітка: Усі завдання виконуються стосовно цільового хоста *zero.webappsecurity.com*

Завдання 1. Мережеве сканування цільового хоста (Nmap)

1. Виконати активне мережеве сканування цільового хоста *zero.webappsecurity.com* за допомогою утиліти *Nmap* з визначенням версій сервісів.

2. Визначити:

- Відкриті мережні порти.
- Мережні сервіси, що працюють на виявлених портах.
- Версії веб-сервера та супутніх компонентів.

3. За допомогою веб-браузера перевірити доступність виявленого веб-ресурсу.

Завдання 2. Визначення технологічного стеку веб-застосунку (cURL, WhatWeb).

1. Використовуючи інструмент cURL, надіслати HTTP-запит до цільового хоста (<http://zero.webappsecurity.com>) та отримати лише HTTP-заголовки відповіді сервера (curl -I).

2. Використовуючи утиліти WhatWeb (або розширення Wappalyzer), виконати ідентифікацію технологічного стеку веб-застосунку.

3. За результатами аналізу встановити:

- Тип і версію веб-сервера та ОС.
- Використовувані JS-бібліотеки, модулі сервера та їхні версії.

4. На основі отриманої інформації про версії ПЗ веб-сервера, використовуючи публічні бази даних (CVE, Exploit-DB) або пошукові системи, знайти 2-3 відомі вразливості. Навести їхні CVE-ідентифікатори та короткий опис.

Завдання 3. Фазинг та пошук прихованих ресурсів (ffuf).

1. Використовуючи інструмент ffuf, виконати перебір директорій та службових шляхів веб-застосунку <http://zero.webappsecurity.com>.

- Використати словник /usr/share/wordlists/dirb/common.txt
- Приклад команди: ffuf -u <http://zero.webappsecurity.com/FUZZ> -w /usr/share/wordlists/dirb/common.txt -s

2. Проаналізувати виявлені шляхи та звернути увагу на:

- Приховані адміністративні директорії (наприклад, /admin).
- Службові сторінки моніторингу стану сервера (наприклад, /server-status).

Завдання 4. Аналіз критичних витоків конфігурації (Information Disclosure).

1. Перейти за допомогою браузера до виявленої сторінки /server-status.
2. Проаналізувати вміст сторінки та занотувати у звіт:

- Повну версію Apache, операційну систему, а також встановлені модулі (mod_ssl, mod_jk, версію OpenSSL).

- Чому виявлення сторінки /server-status вважається вразливістю (помилкою конфігурації) і яку небезпеку це становить для сервера у реальних умовах?

Завдання 5. Пошук прихованих ресурсів та аналіз витоків даних (Sensitive Data Exposure).

1. Використовуючи інструмент ffuf, виконати перебір ресурсів у межах виявленої директорії /admin/ з пошуком веб-сторінок та службових файлів розширення .html, .txt, .log:

```
ffuf -u http://zero.webappsecurity.com/admin/FUZZ -w /usr/share/wordlists/dirb/common.txt -e .html,.txt,.log -s
```

2. За допомогою браузера або curl переглянути вміст виявленого лог-файлу WS_FTP.LOG (шлях: /admin/WS_FTP.LOG).

3. Проаналізувати вихідні дані та внести до звіту:

- Внутрішні (приватні) IP-адреси розробників/адміністраторів.
- Внутрішню структуру локальних каталогів та шляхи до проектів на сервері.
- Імена системних облікових записів або хостів, згаданих у логах.

4. Вручну (у веб-браузері) відкрити виявлену веб-сторінку users.html (шлях: /admin/users.html).

5. Проаналізувати вміст сторінки та внести до звіту:

- Облікові дані одного з користувачів системи.
- Додаткові персональні дані (наприклад, номер соціального страхування), якщо вони присутні на сторінці.

Контрольні запитання

1. Що таке активна розвідка?
2. Що таке пасивна розвідка?
3. У чому полягає різниця між пасивною та активною розвідкою веб-застосунку?
4. Який метод розвідки передбачає збір даних із соціальних мереж та пошукових систем без прямої взаємодії з сервером цілі?
5. Яке основне призначення утиліти cURL?
6. Який параметр утиліти cURL дозволяє отримати лише HTTP-заголовки відповіді без завантаження тіла сторінки?
7. Яке основне призначення інструменту WhatWeb?
8. Яка основна мета використання інструмента Nikto?
9. Для чого використовується ключове слово FUZZ у команді ffuf?
10. Який параметр у ffuf дозволяє вказати розширення файлів (наприклад, .php або .txt), які будуть додані до кожного слова зі словника?
11. У чому полягає основна перевага ffuf над dirb?

Список джерел

1. Cymulate. Cybersecurity Reconnaissance. *Cymulate*. URL: <https://cymulate.com/cybersecurity-glossary/cyber-reconnaissance/>.
2. WhatWeb – Next generation web scanner. URL: <https://whatweb.net/>
3. ffuf | Kali Linux Tools. *Kali Linux*. URL: <https://www.kali.org/tools/ffuf/>.
4. dirb | Kali Linux Tools. *Kali Linux*. URL: <https://www.kali.org/tools/dirb/>.