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, nikto, 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

Продовження табл. 1

-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

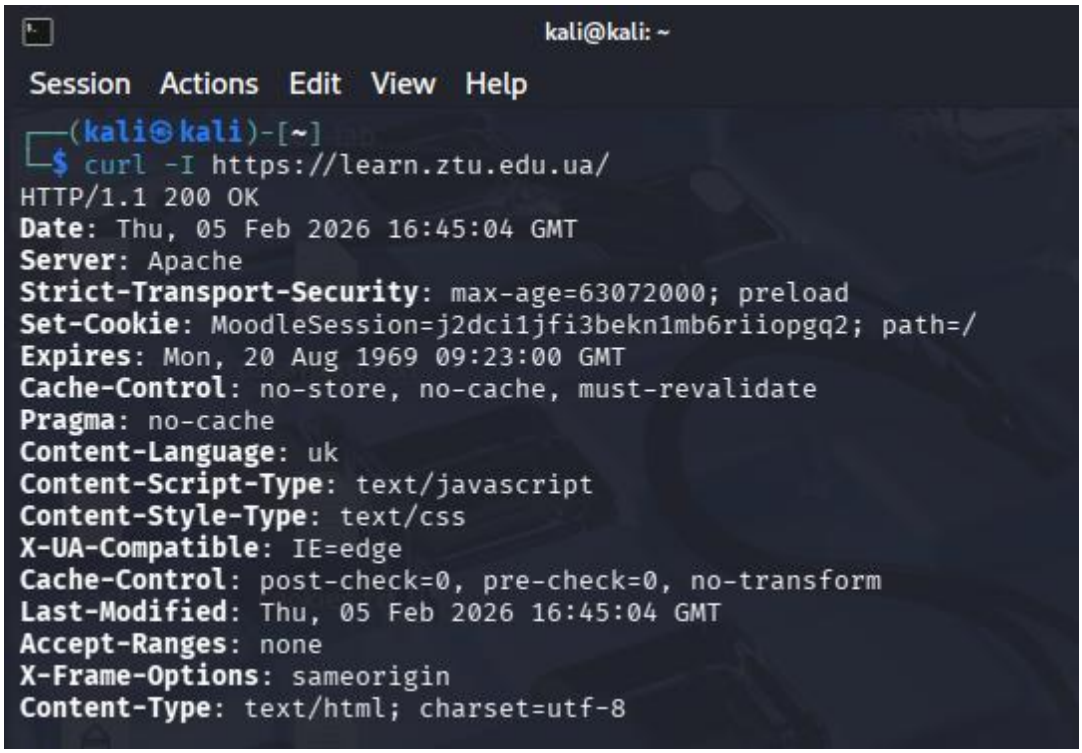

Рисунок 1 – Приклад використання утиліти cURL

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

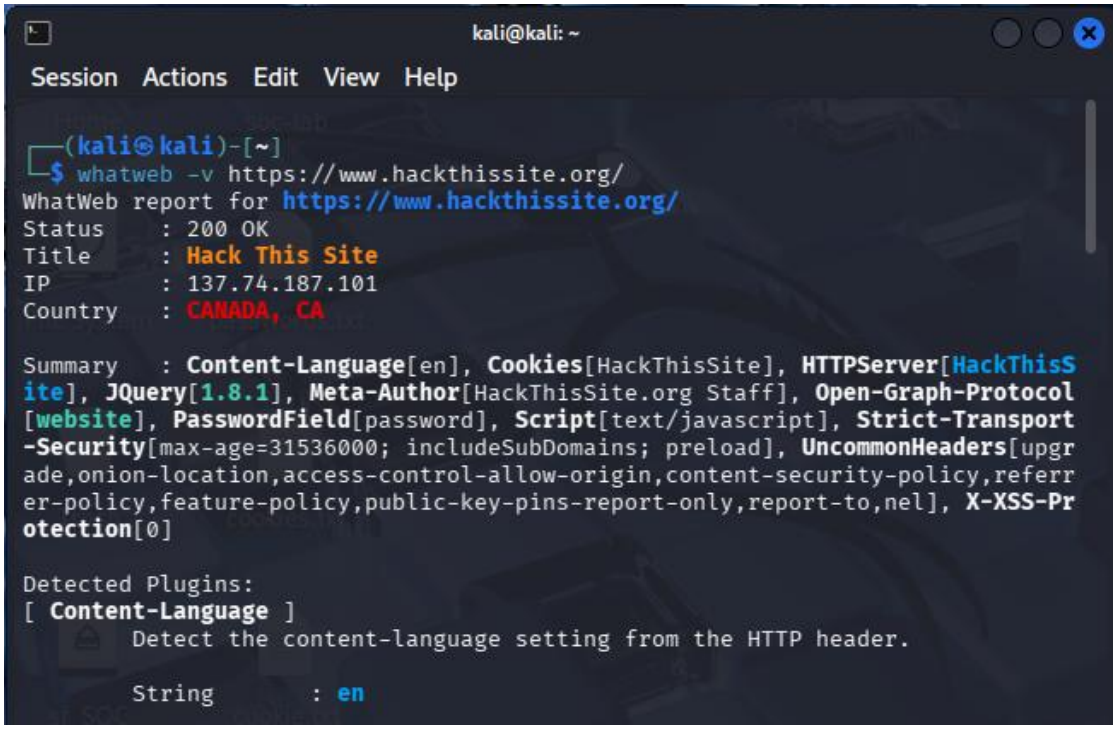

Рисунок 2 - Приклад використання утиліти WhatWeb

Nikto

Nikto - це сканер безпеки веб-серверів із відкритим кодом, який проводить швидку перевірку на наявність вразливих CGI-скриптів, застарілого програмного забезпечення та помилок у конфігурації. Завдяки підтримці SSL, проксі-серверів та автоматичному оновленню баз даних, він дозволяє ефективно виявляти потенційні загрози та генерувати детальні звіти у текстовому або HTML-форматі.

Примітка: Під час роботи з Nikto слід враховувати, що інструмент часто генерує «хибнопозитивні» результати (False Positives). Це означає, що деякі знайдені «вразливості» можуть бути лише особливостями конфігурації, які не становлять реальної загрози. Кожне попередження потребує подальшої ручної перевірки.

Синтаксис:

nikto -h [URL-адреса]

Перелік доступних параметрів можна отримати, виконавши команду:

nikto -h

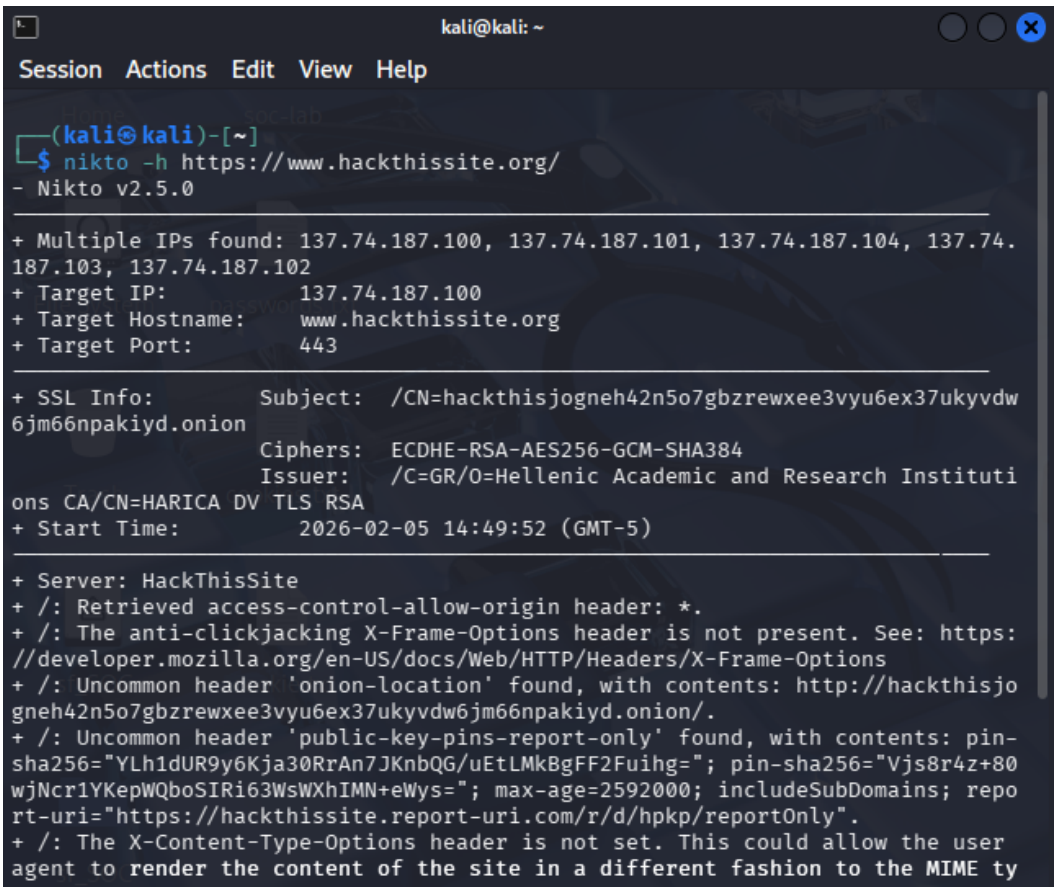

Рисунок 3 - Приклад використання утиліти Nikto

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

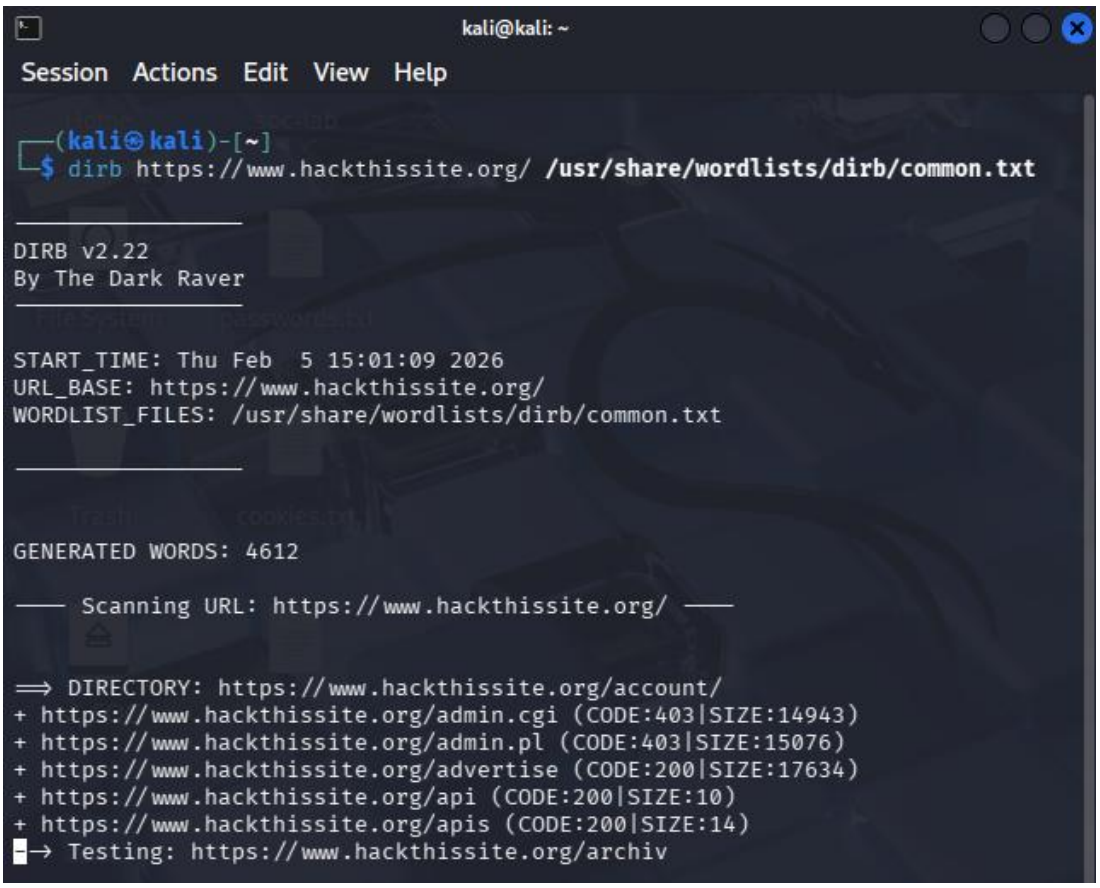

Рисунок 4 - Приклад використання утиліти dirb

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/big.txt

Приклад перебору файлів з різними розширеннями:

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

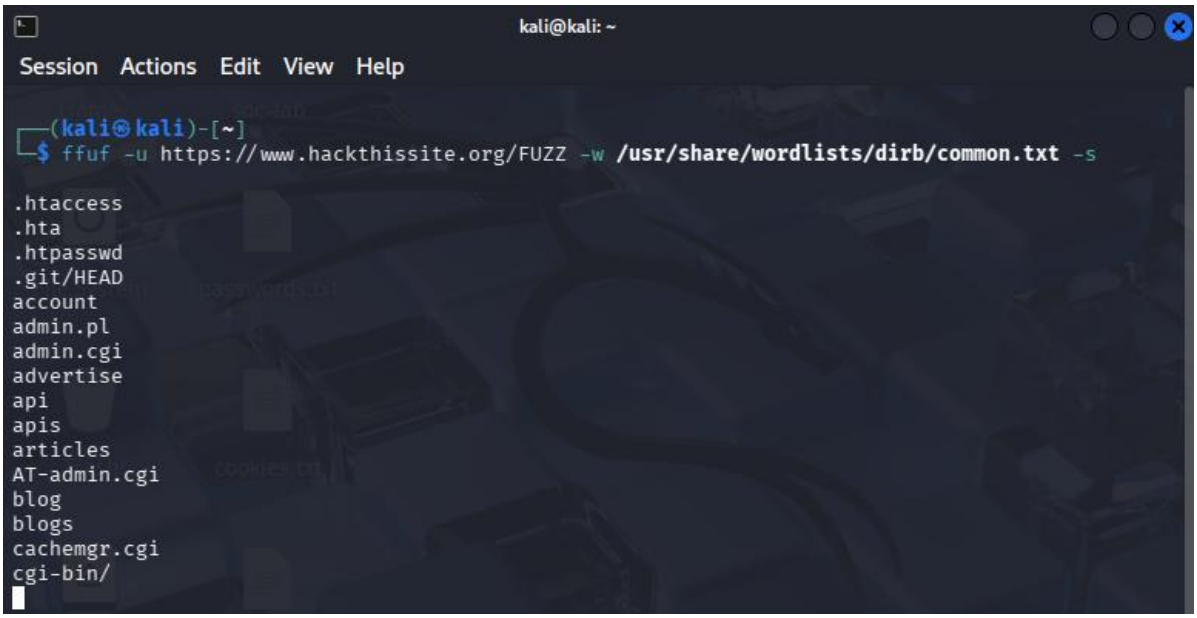

Рисунок 5 - Приклад використання утиліти ffuf

Завдання на лабораторну роботу

Примітка: якщо хост **testphp.vulnweb.com** недоступний, використовуйте **testaspnet.vulnweb.com** для виконання лабораторної роботи. В завданнях 4-5 знайдіть будь-яку директорію/файл шляхом перебору.

Завдання 1. Мережеве сканування цільового хоста (Nmap)

1. Виконати мережеве сканування цільового хоста **testphp.vulnweb.com** за допомогою інструмента **Nmap** (лаб. №1)**.**

2. Визначити:

– Доступні мережеві порти (порти зі статусом open).

– Сервіси, що працюють на виявлених портах.

– Версію веб-сервера.

3. За допомогою веб-браузера перевірити доступність хоста із знайденим відкритим портом.

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

1. Використовуючи інструмент cURL, надіслати HTTP-запит до головної сторінки веб-застосунку (**testphp.vulnweb.com**).

2. Використовуючи інструмент WhatWeb, виконати визначення технологічного стеку веб-застосунку.

3. Встановити:

– Тип і версію веб-сервера.

– Серверну операційну систему.

– Мову програмування (версію), на якій реалізовано веб-застосунок.

4. На основі отриманої інформації про тип і версію веб-сервера, використовуючи будь-яку пошукову систему, знайти 2-3 відомі вразливості (короткий опис вразливості).

Завдання 3. Ознайомлення з автоматизованим інструментом веб-сканування Nikto.

1. Запустити інструмент Nikto для первинного автоматизованого аналізу цільового веб-сервера (**http://testphp.vulnweb.com**).

2. Ознайомитися з результатами сканування.

Завдання 4. Перебір директорій за допомогою інструмента DIRB.

1. Виконати перебір директорій веб-застосунку (**http://testphp.vulnweb.com**) з використанням інструменту DIRB.

2. Виявити приховану директорію.

3. Вручну (у веб-браузері) перейти до виявленої директорії.

4. Виявити службовий або технічний файл, пов’язаний з роботою веб-застосунку.

Завдання 5. Автоматизований пошук файлів за допомогою ffuf.

1. Використовуючи інструмент ffuf, виконати перебір файлів у межах директорії, виявленої на попередньому етапі (завдання 4).

2. Налаштувати інструмент для пошуку файлів із розширенням .sql. Приклад:

ffuf -u http://example.com/hidden_dir/FUZZ -w wordlist.txt -e .sql , де в якості словника для перебору буде **/usr/share/wordlists/dirb/common.txt**.

3. Виявити файл, знайдений раніше під час ручного аналізу директорії.

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. nikto | Kali Linux Tools. *Kali Linux*. URL: https://www.kali.org/tools/nikto/.
4. ffuf | Kali Linux Tools. *Kali Linux*. URL: https://www.kali.org/tools/ffuf/.
5. dirb | Kali Linux Tools. *Kali Linux*. URL: https://www.kali.org/tools/dirb/.