Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

Випускна кваліфікаційна робота бакалавра

«ПОШУК КЛЮЧОВИХ СЛІВ ДЛЯ ВИЗНАЧЕННЯ ПРИХОВАНИХ ДИРЕКТОРІЙ ПІД ЧАС ТЕСТУВАННЯ БЕЗПЕКИ ВЕБ-РЕСУРСУ ДЛЯ ПОШУКУ ПРИХОВАНОГО ФУНКЦІОНАЛУ АБО ВИТОКІВ ІНФОРМАЦІЇ»

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проєкту

№	П.І.Б.	Група
1	Волков Кирил Владиславович	КН-А-191

Автор звіту

Волков Кирил Владиславович

Одеса – 2023

АНОТАЦІЯ

УДК 004.9
Д-30

Волков К. В. Пошук ключових слів для визначення прихованих директорій під час тестування безпеки веб-ресурсу для пошуку прихованого функціонала або витоків інформації: Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2023 - 13 с.

Робота націлена на дослідження та розробку програмного забезпечення, яке оптимізує та вивчає методи пошуку додаткових точок входу в систему при тестуванню безпеки в організації. Пропоноване ПЗ має широкий спектор функціональних можливостей, включаючи вилучення піддоменів, скрапінг веб-сторінок для виявлення унікальних слів та налаштування заголовків HTTP для ефективної взаємодії з цільовими серверами

Головним аспектом роботи є створення унікальних словників для виявлення прихованої "attack surface" (Атакуючої поверхні) шляхом брутфорсу (грубого перебору) піддоменів, пошуку файлів та API ендпоінтів. Це дозволяє виявити приховані або незахищені точки входу, які можуть стати об'єктами для потенційних атак. Створення таких спеціалізованих словників забезпечує більш ретельне дослідження системи з метою виявлення потенційних вразливостей та покращення загальної безпеки в організації. Актуальність даної роботи визначається унікальністю словників, які використовуються для пошуку додаткових ділянок attack surface при тестуванні безпеки мережевих та веб ресурсів. Необхідність створення ручних словників обумовлюється такими недоліками існуючих готових словників, як
- Надмірний розмір, що уповільнює тестування безпеки
- Універсальність, яка пропонує нерелевантні рядки
- популярність через яку всі решта тестувальників безпеки вже знаходять ділянки attack surface.

ВСТУП

У ході тестування безпеки веб-ресурсів часто виникає необхідність визначення потенційно вразливих місць у веб-додатку. Ці місця утворюють так звану площу атаки. Прикладом можуть бути піддомени, віртуальні хости, API-ендпоінти, порти та мережеві сервіси. Для виявлення таких місць використовуються активні та пасивні методи пошуку.

Прикладом пасивного методу пошуку активів компанії є збір піддоменів сайту компанії за допомогою пошукових систем, таких як Google або Bing. Однак цей метод має деякі обмеження. Наприклад, тестувальник безпеки може пропустити піддомени, які не були проіндексовані пошуковим двіжком. Виникає потреба в активному переборі піддоменів.

Активний пошук піддоменів здійснюється шляхом надсилання багатьох DNS-запитів на сервери DNS. При отриманні відповіді, яка містить IP-адресу, ми успішно виявляємо новий піддомен. Цей метод дозволяє ефективно досліджувати мережу та розширювати область видимості піддоменів.

Для активного пошуку піддоменів використовуються словники. Словник є список слів, розділених символом нового рядка, які потенційно можуть бути піддоменами домену. Наприклад, "www", "mail", "blog", "dev", "qa" і таке інше. Однак використання готових словників має деякі недоліки. Оскільки такі словники прагнуть охопити найбільшу кількість випадків, вони включають лише ті слова, які зустрічаються найчастіше. Однак компанії часто створюють унікальні піддомени, наприклад, "mystat.itstep.org" або "homeworks.itstep.org". Слова "mystat" та "homeworks" швидше за все не будуть присутні в універсальних словниках, що може призвести до того, що тестувальник безпеки не виявить таких сайтів, де можуть існувати вразливості.

Для розуміння того, як виглядають сучасні словники, нижче наведено приклад з найпопулярніших словників SecLists:

Файли	**Юзернейми**	**Піддомени**	**Паролі**
index.php	root	www	123456
search.php	admin	mail	password
cron.php	test	ftp	12345678
login.aspx	guest	localhost	qwerty
xmlrpc.php	info	webmail	123456789
LICENSE.txt	adm	smtp	12345
install.php	mysql	webdisk	1234

Приклад, що наведений вище, показує, що популярні словники часто прагнуть бути універсальними, охоплюючи широкий спектр слів та термінів. Однак це рішення не завжди є оптимальним. У деяких випадках, такі словники можуть обмежувати можливості пошуку, особливо, коли йдеться про специфічні терміни або домени.

Далі наведено технічне завдання проекту, в рамках якого розробляється CLI-додаток мовою програмування Python. Вибір CLI замість GUI обумовлений необхідністю забезпечити зручність використання під час тестування безпеки операційних систем, заснованих на ядрі Linux. CLI формат є найбільш зручним з точки зору користувача в даному контексті. Вибір мови програмування обґрунтовується її здатністю забезпечити швидку розробку, що дозволяє швидко тестувати гіпотези та впроваджувати новий функціонал. Такий підхід забезпечує ефективність та гнучкість процесу розробки.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЄКТУ

Загальний опис.

Програмне забезпечення призначене для створення унікальних словників ключових слів, які не трапляються в інших словниках. Основна функція ПЗ - найбільш ефективно зібрати та проаналізувати ключові слова на веб ресурсів.

Призначення:

ПЗ орієнтується на професіоналів у галузі інформаційної безпеки з різних регіонів, у тому числі з України. Користувачі будуть запускати програмне забезпечення переважно на операційних системах на базі ядра Linux. Проте, можуть запускати і на операційній системі Windows завдяки кроссплатформенності мови програмування Python.

Вимоги до функціоналу:

Автоматичний збір веб-сторінок

ПЗ повинно мати можливість автоматичної роботи, за умови, коли користувач надає лише домен. Для оптимізації збору веб-сторінок припустимо використання сторонніх архівів URL-адрес, таких як Wayback Machine, Common Crawl, URLScan і Open Threat Exchange.

Збір ключових слів зі знайдених сторінок

ПЗ має розшукати сайт розташований на 80 (HTTP) або 443 (HTTPS) порту, зібрати усі веб-сторінки та усі слова з веб-сторінок, у тому числі, слова з коментарів HTML сторінок. Далі, воно має дедуплікувати слова (видаляти дублікати) зі списку зібраних слів, видаляти ключові слова з інших словників, видаляти найпоширеніші слова і видаляти рядки, які швидше за все не мають сенсу, такі як: UUID рядки, MD5-подібні хеші, рядки, в яких одні числа, рядки, з символами, що повторюються, рядки коротше 3х символів. Збір ключових слів не повинен відбуватися зі сторінок з медіа-ресурсами, такими як: зображення, відео, шрифти, бінарні файли та інше.

Підтримка роботи зі списку доменів або URL-адрес

ПЗ має робити запити до URL-адреси у файлі наданому, який надав користувач, і збирати всі ключові слова, як зазначено у попередній вимозі до функціоналу. Якщо користувач вказав файл зі списком доменів, програмне забезпечення має шукати ці піддомени і збирати ключові слова.

Локалізація англійською та українською мовами

ПЗ повинно автоматично визначати змінну оточення LANG, та, залежно від її значення, виводити інформацію відповідною мовою. Українською це uk_UA, англійською - en_US.

Підтримка HTTP проксі

ПЗ має підтримувати роботу через HTTP проксі. Для цього необхідно 4 складові:, адреса проксі, проксі порт, ім'я користувача на проксі сервері, пароль на проксі сервері.

Налаштування мінімальної та максимальної довжини слова

ПЗ повинен мати підтримку двох аргументів, які контролюють мінімальну та максимальну довжину слів, з яких буде складено словник.

Вибір опції для складання словника

ПЗ має підтримувати кілька опцій, які змінюють поведінку програми:

1. опція збору коментарів з HTML сторінок, до них не застосовується фільтрація
2. опція збору виключно імен файлів із HTML сторінок. Наприклад, з https://itstep.org/homeworks.php ПЗ поверне “homeworks.php”
3. опція збору виключно шляхів із HTML сторінок. Наприклад, із посилання на https://itstep.org/homework/programming/done ПЗ поверне “homework”, “programming”, “done”
4. опція складання виключно піддоменів вказаного домену з сторінок HTML

Налаштування кількості потоків

ПЗ має підтримувати налаштування кількості потоків для краулера. Стандартна кількість потоків дорівнює 50.

Підтримка користувацьких HTTP заголовків

ПО повинно мати підтримку додавання заголовків HTTP в запити при пошуку. Наприклад, аргумент -H Authorization: Basic YWRtOnBhc3M= додасть у всі HTTP заголовки відповідний заголовок.

Підтримка оновлення існуючих словників

ПЗ повинно мати можливість автоматичного збереження словників у файли та оновлення існуючих словників при повторному запуску програми у разі виявлення нових слів.

Технології

Представлена програма є консольною програмою, розробленою мовою програмування Python. Для передачі додаткових параметрів програмі , використовуються аргументи командного рядка. Для обробки аргументів застосовується бібліотека asyncio, а для виконання HTTP-запитів - urllib, asyncio, bs4 та aiohttp. Для реєстрації журнальних повідомлень використовуються такі бібліотеки: loguru, tqdm та модуль warnings. Для збору посилань з відкритих джерел застосовується інструмент getallurls (gau).

Для локалізації програми використовується система gettext у поєднанні з локальною бібліотекою. Для цього, всі рядки, що логуються, помістилися в функцію gettext. Після цього, утилітою командою рядка xgettext було складено список рядків, які необхідно перекласти. Він був збережений у директорії місцевих організацій та був переведений у програму poedit. Поточний локаль у системі виходить за допомогою функції getdefaultlocale, після чого в залежності від встановленої локалі вибирається відповідний переклад.

Для полегшення використання написано Dockerfile, в якому прописані команди для встановлення всіх необхідних залежностей. Він забезпечує надійну роботу програми на всіх сумісних операційних системах.

Для забезпечення дедуплікації слів застосовується структура даних set. Основна характеристика цієї структури полягає в тому, що вона гарантує унікальність усіх елементів, що містяться в ній. До того ж, операції перевірки наявності елемента в множині відбувається набагато швидше, ніж у списку (list) або словнику (dict).

Для оптимізації словників програма видаляє всі слова з текстових файлів з директорії denylists. За замовчуванням до неї поміщено такі файли:

- google-10000-english.txt - 10 тисяч найчастіших англійських слів
- dutch-10000.txt - 10 тисяч найчастіших німецьких слів
- ukrainian-10000.txt - 10 тисяч найчастіших українських слів
- rfc1866.txt - усі слова, що зустрічаються в RFC1866. Цей файл потрібен, щоб виключити зі словників терміни з HTML, такі як div, css, br, href і подібні.

Дипломну роботу виконано відповідно до методології Agile, що характеризується гнучким та ітеративним підходом до розробки. Основний акцент було зроблено на швидкому тестуванні гіпотез з метою прийняття обґрунтованих рішень. Для управління завданнями використовувався інструмент Trello. Під час розробки було протестовано гіпотезу, пов'язану з додаванням функції пошуку конфіденційної інформації, такої як API-токени, паролі та приватні ключі. Однак з'ясувалося, що наявні регулярні вирази призводять до великої кількості хибнопозитивних результатів, унаслідок чого ідея була відкинута.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

Як формат для зберігання словників було обрано текстовий файл, де слова розділені символом перенесення рядка. Такий вибір обумовлюється тим, що більшість інструментів для перебору піддоменів, файлів, віртуальних хостів приймають словники в цьому форматі. Таким чином, зберігати словники в реляційних базах даних сенсу немає.

РОЗДІЛ 2

РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Тестування ПЗ проводилося часто, після кожної великої зміни в логіку програми тестувалася її ефективність, швидкість роботи, якість зібраних слів, їхня релевантність до тестованої цілі. Виявилася проблема в тому, що деякий "шум", він же слова, які не мають жодного сенсу, майже неможливо прибрати без втрати релевантних слів. Приклади таких слів:

- 063rem
- __NEXT_DATA__
- googletagmanager
- __Cancel
- DMCA
- cloudflare
- HTMLImageElement
- getBrowserFingerprintId

Найімовірніше, це пов'язано це з тим, що функція find_all(text=True) бібліотеки BeautifulSoup повертає весь текстовий читабельний контент з HTML-сторінки, але може повертати й інший текст, що міститься всередині тегів. Проте, програма показала хороші, релевантні результати під час тестування. Нижче наведено приклад зібраних слів, у дужках вказано загальну кількість зібраних слів:

Сервіс для допомоги в медитації (367)	Блог про машини (361)	Сайт американського університету (708)	Сайт зоопарку (122)
comfortably	airbag	Archaeology	Mammal
empathetic	Crossover	Cardiology	Keepers
Mindfulness	Deliveries	syllabus	behaviors
workouts	Driveway	Deadlines	ecosystem

burnout	Liftback	Diplomatic	elephants
holistic	Motorsports	Entrepreneurship	habitats
pandemic	Supercharging	Psychologist	panda
stresses	autonomous	Webinar	playful
unconscionability	automakers	economist	grasses

Далі, за наявності готового словника, тестувальник безпеки має використовувати його для розширення attack surface. У цьому прикладі я зібрав піддомени сайту університету за допомогою створеного словника:

- brattle.harvard.edu
- berkman.harvard.edu
- arboretum.harvard.edu
- archaeology.harvard.edu
- digital.harvard.edu
- radcliffe.harvard.edu
- syllabus.harvard.edu
- harvardonline.harvard.edu
- bill.harvard.edu
- commencement.harvard.edu

Оскільки слова на кшталт "berkman", "radcliffe" і "archaeology" зазвичай не трапляються в словниках, пошук таких піддоменів був би набагато складнішим. Аналогічним чином, можливе використання словників для пошуку додаткових API ендпоїнтів, файлів, віртуальних хостів та інших ресурсів, пошук яких був би важким без використання розробленого інструменту.

ВИСНОВКИ

Розроблено багатофункціональний додаток мовою програмування Python, здатний ефективно створювати унікальні словники для брутфорсу піддоменів, API-ендпоінтів, параметрів у HTTP-запитах, віртуальних хостів, файлів та інших елементів. Цей застосунок дає змогу розширити атакуючу поверхню і провести тестування безпеки важливих компонентів, які можуть залишитися непоміченими за допомогою звичайних методів.

Додаток було розроблено з урахуванням ефективності та продуктивності, забезпечуючи потужні інструменти для збору слів і створення словників. Такий підхід дає змогу проводити тестування безпеки з високою точністю та ефективністю. Особливу увагу приділено деталям, забезпечуючи зручні та інтуїтивно зрозумілі аргументи і простоту використання, що зробило додаток простим для використання.

Під час проведених досліджень було виявлено, що збільшення кількості зібраних сторінок не призводить до значного збільшення розміру словника. Експериментальним шляхом було встановлено, що оптимальна кількість сторінок становить від 500 до 1000. Це спостереження вказує на те, що під час збирання більшої кількості сторінок у словнику не відбувається значущого приросту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://github.com/lc/gau
2. https://www.youtube.com/watch?v=W4_QCSIujQ4
3. https://github.com/BonJarber/SecUtils/tree/master/clean_wordlist
4. https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/raft-large-directories-lowercase.txt
5. https://github.com/oprogramador/most-common-words-by-language/blob/master/src/resources/dutch.txt
6. https://github.com/first20hours/google-10000-english/blob/master/google-10000-english.txt
7. https://tools.ietf.org/html/rfc1866
8. https://github.com/projectdiscovery/dnsx
9. https://poedit.net/
10. https://www.gnu.org/software/gettext/manual/html_node/xgettext-Invocation.html