УДК 004.42+004.6

ЗАСОБИ МОВИ ПРОГРАМУВАННЯ PYTHON ДЛЯ ВИРІШЕННЯ ЗАДАЧ КІБЕРБЕЗПЕКИ ТА ПРИКЛАДИ ЇХ ВИКОРИСТАННЯ

Глинчук Л.Я.(lydmilaglin@ukr.net)
Волинський національний університет імені Лесі Українки, м. Луцьк

В тезах обговорено важливість мови програмування Python у сфері кібербезпеки. Python відзначається своєю простотою та гнучкістю, що дозволяє швидко розробляти інструменти для автоматизації безпекових завдань, таких як тестування безпеки, аналіз шкідливого програмного забезпечення та забезпечення мережевої безпеки. Досліджено, які бібліотеки та для яких завдань кібербезпеки можуть бути використані. Зазначено, що Python активно використовується для автоматизації рутинних задач, включаючи виявлення вразливостей і аналіз логів. Крім того, розглядаються специфічні бібліотеки та інструменти, що допомагають у виконанні різних завдань кібербезпеки, таких як моніторинг інцидентів і захист веб-додатків. Також подано приклади українських та закордонних проєктів у цій галузі, які використовують згадані бібліотеки Python для розробки інструментів безпеки.

Python є однією з найпопулярніших мов програмування в галузі кібербезпеки завдяки своїй простоті, широкій системі бібліотек та інструментів, а також гнучкості, яка дозволяє швидко розробляти та автоматизувати завдання безпеки. Використання Python для вирішення задач кібербезпеки охоплює різноманітні аспекти: від автоматизації процесів тестування безпеки до написання експлойтів, аналізу шкідливого ПЗ та забезпечення мережевої безпеки.

Одним із головних переваг Python є його багата бібліотека для роботи з мережами, криптографією та аналізом даних, що робить його ідеальним інструментом для фахівців з безпеки. Наприклад, бібліотеки Scapy дозволяють проводити мережевий аналіз і моделювання трафіку, а Cryptography надає засоби для шифрування даних. Бібліотека Socket дозволяє легко створювати сервери та клієнтів для тестування мережевих протоколів та аналізу уразливостей.

Крім того, Python активно використовується для автоматизації рутинних задач, таких як сканування мереж, виявлення вразливостей, та аналіз логів. З допомогою інструментів, як-от Nmap та OpenVAS, Python може автоматизувати та розширювати можливості стандартних інструментів для тестування безпеки.

Важливим аспектом використання Python у кібербезпеці є здатність швидко створювати власні інструменти для вирішення специфічних задач. Завдяки гнучкості Python фахівці з безпеки можуть швидко адаптуватися до нових загроз, написавши власні скрипти для аналізу шкідливого ПЗ, тестування вразливостей або управління інцидентами безпеки. [1]

Основні задачі кібербезпеки, які можна програмувати за допомогою Python: автоматизація тестування безпеки, мережевий аналіз та тестування, шифрування та криптографія, аналіз шкідливого ПЗ, експлойтинг та тестування на проникнення, моніторинг і реагування на інциденти, аналіз та відновлення, розробка власних засобів захисту, захист веб-додатків. Розглянемо детальніше кожну задачу та бібліотеки, за допомогою яких, можна реалізувати розв’язок. [2]

До автоматизації тестування безпеки відносять можливості: сканування вразливостей, автоматизовані атаки типу «брутфорс», фаззинг (тестування на основі випадкових даних).

Python дозволяє автоматизувати процес пошуку вразливостей у веб-додатках, мережах та системах за допомогою таких інструментів, як: Nmap (бібліотека python-nmap, зовнішня), OpenVAS (зовнішня).

Скрипти на Python дозволяють реалізувати атаки перебору паролів (брютфорс) через: Paramiko (зовнішня, для SSH), Requests (зовнішня, для HTTP).

Інструменти для автоматичного генерування та відправки некоректних або випадкових даних для тестування на вразливості. Бібліотеки: Boofuzz (зовнішня), Atheris (зовнішня).

Python дозволяє перехоплювати та аналізувати мережевий трафік за допомогою таких бібліотек: Scapy (зовнішня), dpkt (зовнішня).

Можна писати власні інструменти для сканування мереж використовуючи вбудовану бібліотеку socket. Для автоматизації аналізу журналів мережевих пристроїв і серверів використовуються бібліотеки: Loguru (зовнішня), Pandas (зовнішня).

У задачах шифрування та криптографії Python дозволяє створювати власні рішення для захисту даних за допомогою таких бібліотек: PyCryptodome (зовнішня), Cryptography (зовнішня). Для генерації та управління ключами зовнішню бібліотеку Cryptography.

При аналізі шкідливого ПЗ може бути використаний статичний аналіз та динамічний аналіз. Статичний може допомогти аналізувати шкідливе ПЗ без його виконання за допомогою бібліотек: pefile (зовнішня), yara-python (зовнішня). Для моніторингу поведінки шкідливого ПЗ у реальному часі, тобто при динамічному аналізі, можна скористуватися такими інструментами як: frida (зовнішня), volatility (зовнішня).

Python дозволяє писати власні експлойти для вразливих систем і додатків за допомогою інструментів: Pwntools (зовнішня), Impacket (зовнішня). Для застереження від соціальної інженерії та фішингу допоможуть бібліотеки: smtplib (вбудована), Flask (зовнішня). За допомогою Python можна створювати інструменти для перехоплення та модифікації трафіку використовучи: Scapy (зовнішня), Mitmproxy (зовнішня).

Моніторинг і реагування на інциденти включає в себе завдання: виконання моніторингу систем та автоматизації реагування на інциденти. Моніторинг систем можна організувати за допомогою зовнішньої бібліотеки Psutil. Для автоматизації реагування на інциденти можна використати зовнішню бібліотеку Requests.

Для аналізу файлів і систем розроблені зовнішні бібліотеки: PyTSK3 та ExifRead. Відновлення даних можливе з використанням зовнішньої бібліотеки Scalpel. І нарешті, для аналізу пам’яті можна використати фреймворк Volatility.

При розробці власних засобів захисту можуть виникнути завдання інтеграції з SIEM-системами, тут допоможуть зовнішні бібліотеки: ElasticSearch API, Splunk SDK. Для розробки антивірусних рішень можуть підійти зовнішня бібліотека YARA-python та вбудована – hashlib.

Для захисту веб-додатків, зокрема, для тестування веб-безпеки використовуються зовнішні бібліотеки BeautifulSoup та Selenium. При реалізації моніторингу веб-трафіку – Requests (зовнішня) та Mitmproxy (зовнішня). [3]

Бачимо, що Python забезпечує широкий вибір інструментів для автоматизації та вирішення задач кібербезпеки на всіх рівнях: від тестування безпеки до розробки власних рішень захисту. Детальніше про згадані тут бібліотеки та їх функції можна дізнатися в документації кожної бібліотеки.

Наведемо кілька останніх закордонних розробок програмного забезпечення в сфері кібербезпеки, які були створені за допомого вибраних вище бібліотек Python.

Mitmproxy – потужний інструмент для перехоплення та модифікації HTTP-трафіку, який використовується для тестування безпеки веб-додатків. (розробник – Mitmproxy Team, 2023 рік, країна – Німеччина). Використана бібліотека Scapy для моделювання мережевих сценаріїв. [4]

Volatility 3 – оновлена версія популярного інструменту для аналізу пам’яті, що дозволяє досліджувати зразки пам’яті на предмет шкідливих програм та інших загроз (розробник – Volatility Foundation, 2021 рік, країна – США). Використана бібліотека PyTSK3 для аналізу файлових систем і метаданих. [5]

BloodHound – інструмент для аналізу активних директорій, що допомагає виявляти вразливості в інфраструктурі Windows (розробник – SpecterOps, 2022 рік, країна – США). Використана бібліотека Pandas для аналізу даних та їх візуалізації. [6]

Cortex – ɡлатформа для автоматизації та управління кіберінформацією, яка допомагає в обробці інцидентів, управлінні загрозами та аналітиці (розробник – TheHive Project, 2022 рік, країна – Франція). Використана бібліотека Cortex, що інтегрує бібліотеки для сканування вразливостей, такі як OpenVAS. [7]

Наведемо ще приклади українських проєктів у сфері кібербезпеки, які використовують описані вище бібліотеки Python.

Отже, CyberX – це система моніторингу безпеки, яка аналізує мережевий трафік і виявляє потенційні загрози в режимі реального часу. Для автоматизації аналізу трафіку команда використовувала бібліотеку Scapy, яка дозволяє створювати та відправляти пакети, а також здійснювати їхній аналіз (розробник – команда CyberX, 2022 рік).

Deep Security – це система, що інтегрує аналітику загроз з алгоритмами машинного навчання для виявлення аномалій у поведінці користувачів. Для шифрування даних та забезпечення криптографічної безпеки розробники використовували бібліотеку Cryptography, яка пропонує широкий спектр інструментів для роботи з шифруванням (розробник – компанія DTEK, 2023 рік).

Hacker Hunter – це інструмент для моніторингу та аналізу шкідливих дій в інтернеті. Він використовує бібліотеки requests для збору інформації про IP-адреси та BeautifulSoup для веб-скрапінгу. Інструмент також автоматично генерує звіти про виявлені загрози (розробник – Кіберполіція України, 2021 рік).

CyberShield – це рішення для аналізу вразливостей веб-додатків. Воно використовує бібліотеки python-nmap для сканування мереж і OpenVAS для виявлення вразливостей у системах (розробник – компанія SoftServe, 2023 рік).

MalwareAnalyzer – це інструмент для аналізу шкідливого ПЗ, який підтримує статичний та динамічний аналіз. Розробники використовували бібліотеки pefile для статичного аналізу виконуваних файлів і frida для динамічного моніторингу поведінки шкідливого ПЗ (розробник – команда Dnipro IT, 2022 рік).

Детальніше про кожне з ПЗ українського походження можна прочитати на офіційних сайтах розробників.

Python залишається потужним інструментом у сфері кібербезпеки завдяки своїй універсальності та широкій екосистемі бібліотек. Його здатність до автоматизації рутинних завдань, гнучкість у створенні спеціалізованих рішень і наявність бібліотек для аналізу, шифрування та тестування робить його незамінним для фахівців з безпеки. Різноманітні українські проекти, що використовують Python, підтверджують його ефективність і значення у розробці сучасних рішень у сфері кібербезпеки. У світлі постійно зростаючих загроз, знання і використання Python може стати ключовим фактором у забезпеченні надійного захисту інформаційних систем. Тому фахівцям у цій галузі важливо залишатися в курсі нових можливостей мови та її бібліотек, щоб оперативно реагувати на нові виклики.

Список використаної літератури

[1]. Haniel J. Для чого використовується Python? 11 найпоширеніших випадків використання Python. *JavaRush*. URL: https://javarush.com/ua/groups/posts/dlja-chogo-vikoristovutjhsja-python.

[2]. Найкраща мова програмування для вивчення кібербезпеки. *Online Coding Bootcamps | Code Labs Academy*. URL: https://codelabsacademy.com/uk/blog/the-best-programming-language-for-learning-cybersecurity.

[3]. PyPI · The Python Package Index. *PyPI*. URL: https://pypi.org/.

[4]. mitmproxy – an interactive HTTPS proxy. *mitmproxy - an interactive HTTPS proxy*. URL: https://mitmproxy.org/.

[5]. GitHub – volatilityfoundation/volatility3: Volatility 3.0 development. *GitHub*. URL: https://github.com/volatilityfoundation/volatility3.

[6]. GitHub – BloodHoundAD/BloodHound: Six Degrees of Domain Admin. *GitHub*. URL: https://github.com/BloodHoundAD/BloodHound.

[7]. GitHub – TheHive-Project/Cortex: Cortex: a Powerful Observable Analysis and Active Response Engine. *GitHub*. URL: https://github.com/TheHive-Project/Cortex.