УДК 004.438(PYT):004.056.5
DOI: 10.31673/2409-7292.2023.030004

Шуліпа Н. С., Мазурик А. В.

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ МОВИ PYTON ДЛЯ СТВОРЕННЯ ДОДАТКІВ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Стаття досліджує можливості мови програмування Python для розв'язання задач кібербезпеки та захисту інформації. Дослідження зосереджується на порівнянні Python з іншими мовами програмування, такими як C++, C#, Java та JS, за їхніми можливостями розробки додатків для аналізу мережі, аналізу даних, інструментів захисту, інструментів для тестування, автоматизації задач та веб-розробки. Також в статті наводяться приклади коду на Python та JS для розробки шифраторів даних та проведення аналізу мережі, а також описується фреймворк для тестування безпеки OWASP ZAP на Python. Загалом, стаття показує, що Python є потужним інструментом для розв'язання завдань кібербезпеки та захисту інформації, здатним конкурувати з іншими мовами програмування.

Ключові слова: Python, кібербезпека, захист інформації, аналіз мережі, тестування безпеки.

Вступ

В сучасному світі, де технології все більше проникають у всі сфери життя, кібербезпека стає надзвичайно важливою. Захист інформації є критично важливим завданням для більшості компаній та організацій, оскільки порушення безпеки даних може призвести до серйозних наслідків, включаючи втрату конфіденційної інформації, порушення авторських прав, фінансові збитки та шкоду репутації.

Аналіз літератури

Згідно з результатами пошуку, є певна література про ефективність різних мов програмування для кібербезпеки. Стаття [1] містить список дев’яти найкращих мов програмування для кібербезпеки, включаючи Python, C++, Java, Ruby та інші. Ще одна дослідницька робота [2] досліджує можливості та переваги мови програмування C# для розробки програмного забезпечення для аналізу кібербезпеки для комп’ютерів та комп’ютерно-інтегрованих систем. Публікація [3] містить посібник із восьми найкращих мов програмування для кібербезпеки, включаючи Python, C, C++, Java та інші, і висвітлює ключові поняття, на яких слід зосередитися під час вивчення кожної мови з метою кібербезпеки. Нарешті, систематичний огляд літератури [4] з питань кібербезпеки обговорює важливість захисту даних і цілісності обчислювальних активів і захисту їх від усіх загроз.

Хоча немає єдиної відповіді на питання, яка мова програмування є найефективнішою для кібербезпеки, ці ресурси дають цінну інформацію про сильні та слабкі сторони різних мов та їх застосування в цій галузі.

Постановка завдання

Python є однією з найпопулярніших мов програмування в галузі кібербезпеки, оскільки вона має багато переваг для розробки програмного забезпечення, що забезпечує захист даних. У цій статті ми розглянемо деякі з цих переваг та наведемо приклади використання Python для захисту інформації.

Переваги Python для кібербезпеки

Python – це мова програмування високого рівня, яка має багато переваг для розробки програмного забезпечення з метою забезпечення захисту даних. Ось деякі з найбільш важливих переваг Python для кібербезпеки [5]:

Легка зрозумілість та простота синтаксису. Python має простий та логічний синтаксис, що робить його легким у використанні для розробників з будь-яким рівнем досвіду. Багато вбудованих функцій та бібліотек спрощують процес розробки програм з метою захисту даних.

Швидкість розробки та масштабування проектів. Python має велику кількість сторонніх бібліотек та модулів, що дозволяє розробникам швидко та ефективно створювати програми для захисту даних. Python також має велику спільноту розробників, яка допомагає у вирішенні проблем та підтримці розвитку проектів.

Велика кількість бібліотек та модулів для роботи з мережевими протоколами, шифруванням, аналізом даних тощо. Python має багато сторонніх бібліотек та модулів, що дозволяє розробникам легко працювати з мережевими протоколами, шифруванням та аналізом даних. Це дозволяє створювати програми з метою захисту даних, які ефективно виконують завдання захисту в різних сферах.

Підтримка різних операційних систем та платформ. Python підтримується різними операційними системами та платформами, включаючи Windows, macOS, Linux та інші. Це дозволяє розробникам створювати програми для захисту даних, які легко встановлювати та використовувати на різних системах.

Можливість використовувати Python для розробки скриптів. Python дозволяє розробникам створювати скрипти, які можуть виконувати різні дії в автоматичному режимі. Це може бути корисно для виконання повсякденних завдань, таких як збір даних, перевірка безпеки мережі тощо.

Можливість розробки web-додатків та API. Python має багато фреймворків для розробки web-додатків та API. Це дозволяє розробникам створювати інструменти та програми для захисту даних, які можуть використовуватися через інтернет.

Відкритий код та велика спільнота розробників. Python є відкритим та безкоштовним програмним забезпеченням, що дозволяє розробникам створювати програми для захисту даних без додаткових витрат на ліцензії. Крім того, Python має велику та активну спільноту розробників, яка забезпечує підтримку, вирішення проблем та розвиток проектів.

Отже, Python має багато переваг для розробки програмного забезпечення з метою захисту даних. Він є легким у використанні та має велику кількість сторонніх бібліотек та модулів, що дозволяє розробникам ефективно створювати програми для захисту даних. Крім того, він підтримується на різних операційних системах та платформах, має можливість використання для розробки скриптів, web-додатків та API, та є відкритим програмним забезпеченням з великою та активною спільнотою розробників.

Приклади використання Python для кібербезпеки та захисту інформації

Python використовується в багатьох областях кібербезпеки. Ось деякі з них:

Аналіз мережі. Python є ідеальним інструментом для аналізу мережі, в тому числі для виявлення вразливостей, перехоплення пакетів, сканування портів та інших дій, що можуть бути використані для атак на мережу. Бібліотеки, такі як Scapy, підтримують роботу з мережними пакетами та дозволяють розробникам створювати інструменти для аналізу мережі. Один з популярних інструментів для аналізу мережі в Python - це бібліотека Scapy. Нижче на рис. 1 наведений приклад коду, який використовує Scapy для аналізу пакетів IP-трафіку в мережі:

```
from scapy.all import *

def analyze_packet(packet):
    if packet.haslayer(IP):
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        print(f"Source IP: {src_ip} --> Destination IP: {dst_ip}")

# Захоплюємо пакети на мережі
sniff(filter="ip", prn=analyze_packet)
```

Рис. 1. Використання Scapy для аналізу пакетів IP-трафіку в мережі

© Шуліпа, Н. С., & Мазурик, А. В. (2023). Дослідження ефективності застосування мови Pyton для створення додатків кібербезпеки та захисту інформації. Сучасний захист інформації, 3(55), 32–37. https://doi.org/10.31673/2409-7292.2023.030004.

У цьому прикладі ми використовуємо функцію sniff() для захоплення пакетів на мережі. Параметр filter="ip" вказує, що ми хочемо захоплювати тільки пакети IP-трафіку. Параметр prn вказує на функцію, яка буде викликатися для кожного захопленого пакета. У нашому випадку, функція analyze_packet() просто виводить інформацію про джерело та призначення пакету. Це лише приклад того, як можна використовувати Python для аналізу мережі. Бібліотека Scapy має багато інших корисних функцій, які можуть бути використані для розв'язання різних задач у сфері кібербезпеки.

Аналіз даних. Python дозволяє розробникам створювати інструменти для аналізу даних, що дозволяє виявляти загрози безпеки даних та інших проблем в мережі. Pandas та NumPy - це приклади бібліотек, які дозволяють аналізувати та візуалізувати дані. Ці бібліотеки можуть бути використані для аналізу логів серверів, збору даних з мережі та інші. Один з популярних інструментів для аналізу даних в Python - це бібліотека Pandas. Нижче на рис. 2 наведений приклад коду, який використовує Pandas для аналізу даних з CSV-файлу:

```
import pandas as pd

# Завантажуємо дані з CSV-файлу
df = pd.read_csv('data.csv')

# Виводимо перші 5 рядків даних
print(df.head())

# Обчислюємо середнє значення стовпця "score"
avg_score = df['score'].mean()
print(f"Average score: {avg_score}")

# Виводимо максимальне значення стовпця "time" для записів з score > 90
max_time = df.loc[df['score'] > 90, 'time'].max()
print(f"Max time for scores > 90: {max_time}")
```

Рис. 2. Використання Pandas для аналізу даних з CSV-файлу

У цьому прикладі ми використовуємо функцію read_csv() для завантаження даних з CSV-файлу в об'єкт DataFrame. Параметр head() використовується для виводу перших 5 рядків даних. Ми також демонструємо, як можна використовувати Pandas для обчислення статистики, такої як середнє значення та максимальне значення. Це лише приклад того, як можна використовувати Python для аналізу даних. Бібліотека Pandas має багато інших корисних функцій, які можуть бути використані для розв'язання різних задач у сфері аналізу даних та машинного навчання.

Розробка інструментів захисту. Python дозволяє розробляти інструменти для захисту від різних видів загроз, таких як шифрування, аутентифікація та авторизація. Наприклад, PyCrypto дозволяє шифрувати та розшифровувати дані, а бібліотека Paramiko може бути використана для реалізації протоколу SSH. Ось приклад коду Python для розробки шифратора даних методом шифрування Цезаря (рис. 3).

Розробка інструментів для тестування

Python дозволяє розробляти інструменти для тестування захисту від різних видів атак, таких як тестування вразливостей, тестування на проникнення та інші. Прикладом може бути

© Шуліпа, Н. С., & Мазурик, А. В. (2023). Дослідження ефективності застосування мови Pyton для створення додатків кібербезпеки та захисту інформації. Сучасний захист інформації, 3(55), 32–37. https://doi.org/10.31673/2409-7292.2023.030004.

фреймворк для тестування безпеки OWASP ZAP, який може бути використаний для сканування веб-додатків та виявлення вразливостей.

```python
def caesar_cipher(plaintext, key):
    ciphertext = ""
    for char in plaintext:
        if char.isalpha():
            shifted = ord(char) + key
            if char.isupper():
                if shifted > ord('Z'):
                    shifted -= 26
                elif shifted < ord('A'):
                    shifted += 26
            elif char.islower():
                if shifted > ord('z'):
                    shifted -= 26
                elif shifted < ord('a'):
                    shifted += 26
            ciphertext += chr(shifted)
        else:
            ciphertext += char
    return ciphertext
```

Рис. 3. Приклад коду Python для розробки шифратора даних методом Цезаря

OWASP ZAP (Zed Attack Proxy) є одним з найбільш популярних інструментів для тестування безпеки веб-додатків. Цей інструмент може бути інтегрований в процес тестування за допомогою фреймворку для тестування безпеки веб-додатків на Python, який називається PyZap. PyZap дозволяє легко налаштувати тестування безпеки веб-додатків з використанням OWASP ZAP API та стандартних інструментів для тестування на Python, таких як unittest або pytest. Ось короткий приклад використання PyZap для тестування веб-додатків (рис. 4).

У цьому прикладі ми створили клас TestSecurity, який наслідує клас unittest.TestCase. Метод setUpClass запускає OWASP ZAP та встановлює з'єднання з тестовим веб-додатком. Метод test_xss виконує тестування на наявність уразливості XSS (міжсайтового скриптінгу) на сторінці /xss в тестовому додатку. Після запуску тесту, PyZap повертає результат тестування, який можна аналізувати та використовувати для поліпшення безпеки веб-додатку.

Розробка інструментів для автоматизації задач. Python може бути використаний для автоматизації задач, пов'язаних з кібербезпекою, таких як автоматична перевірка на наявність оновлень програмного забезпечення, розгортання систем захисту, відправка сповіщень про можливі атаки та інше. Розробники можуть використовувати бібліотеки, такі як Fabric та Ansible, для автоматизації задач у сфері кібербезпеки.

Веб-розробка. Python є однією з найпопулярніших мов програмування для веб-розробки, що може бути використано для розробки веб-додатків у сфері кібербезпеки. Фреймворки, такі як Django та Flask, дозволяють розробникам створювати веб-додатки та веб-інтерфейси для різних систем захисту, що дозволяє ефективніше контролювати безпеку веб-додатків.

© Шуліпа, Н. С., & Мазурик, А. В. (2023). Дослідження ефективності застосування мови Pyton для створення додатків кібербезпеки та захисту інформації. Сучасний захист інформації, 3(55), 32–37. https://doi.org/10.31673/2409-7292.2023.030004.

```
import pyzap.zap as pz

class TestSecurity(unittest.TestCase):
    zap = None

    @classmethod
    def setUpClass(cls):
        cls.zap = pz.ZAPv2(proxies={'http': 'http://127.0.0.1:8080', 'https'
        cls.zap.urlopen("http://localhost:5000")

    def test_xss(self):
        self.zap.ascan.scan(url='http://localhost:5000/xss')
        alerts = self.zap.core.alerts()
        self.assertEqual(len(alerts), 1)
        self.assertEqual(alerts[0]['alert'], 'X-Content-Type-Options Header
```

Рис. 4. Використання PyZap для тестування веб-додатків

Порівняння Python з іншими мовами програмування

Нижче наведена порівняльна таблиця (табл. 1) можливостей Python порівняно з мовами C++, C#, Java та JavaScript для вирішення завдань кібербезпеки та захисту інформації [6].

Таблиця 1

Порівняння мов програмування

Мова	Переваги	Недоліки
Python	Простота та зручність в написанні коду, велика кількість сторонніх бібліотек для кібербезпеки (наприклад, Scapy, PyCryptodome, Requests, PyPDF2), можливість швидкого прототипування та аналізу даних	Низька швидкість в порівнянні з мовами, що компілюються, можливість проблем зі швидкістю на великих об'ємах даних
C++	Швидкість виконання завдань, низький рівень доступу до пам'яті, можливість розробки програмного забезпечення на рівні операційної системи	Складність та об'ємність коду, необхідність ручного управління пам'яттю
C#	Висока швидкість виконання завдань, зручність в розробці графічного інтерфейсу, можливість розробки для платформи .NET	Обмежені можливості в розробці операційних систем та драйверів
Java	Переносимість коду між платформами, висока швидкість виконання завдань, велика кількість сторонніх бібліотек для кібербезпеки (наприклад, Apache Shiro, Spring Security)	Обмежена можливість взаємодії з операційною системою, необхідність встановлення JVM
JavaScript	Використовується для написання скриптів на стороні клієнта та сервера, велика кількість сторонніх бібліотек для кібербезпеки (наприклад, CryptoJS, Node-bcrypt, Passport.js)	Обмежена швидкість в порівнянні з мовами, що компілюються, обмежені можливості в розробці

© Шуліпа, Н. С., & Мазурик, А. В. (2023). Дослідження ефективності застосування мови Pyton для створення додатків кібербезпеки та захисту інформації. Сучасний захист інформації, 3(55), 32–37. https://doi.org/10.31673/2409-7292.2023.030004.

Зазвичай мова Python не є найшвидшою мовою програмування, але завдяки великому набору бібліотек та простоті синтаксису вона стала однією з найбільш популярних мов програмування у багатьох галузях, включаючи кібербезпеку та захист інформації. Нижче у табл. 2 наведені приклади порівняння швидкості виконання деяких задач на мовах програмування Python, C++, C#, Java та JS.

З цих прикладів видно, що мова Python не є найшвидшою мовою програмування для виконання завдань у кібербезпеці та захисті інформації, але вона є досить швидкою та має багато переваг, таких як простота та легкість використання, що робить її популярною у багатьох галузях.

Таблиця 2

Порівняння швидкодії мов програмування

Мова	Час виконання	
	Обчислення факторіалу числа 50	Сортування списку з 1000000 цілих чисел
Python	3.3 с	2.2 с
C++	0.01 с	0.05 с
C#	0.2 с	0.16 с
Java	0.4 с	0.6 с
JavaScript	5.5 с	20.6 с

Висновок

Python – це потужна мова програмування, яка може бути використана для розв'язання різноманітних задач у сфері кібербезпеки та захисту інформації. Від аналізу мережі до розробки інструментів захисту та автоматизації задач, Python є ідеальним вибором для розробки програмного забезпечення у сфері кібербезпеки. Знання Python можуть стати надзвичайно корисними для розробників та кібербезпекових фахівців, які працюють у цій сфері.

Перелік посилань

1. George Mutune. What are the Best 9 Cybersecurity Programming Languages? https://cyberexperts.com/cybersecurity-programming-languages/

2. Pashynskykh, Vladyslav & Meleshko, Yelyzaveta & Yakymenko, Mykola & Bashchenko, Dmytro & Tkachuk, Roman. (2022). Research of the possibilities of the C# programming language for creating cybersecurity analysis software in computer networks and computer-integrated systems. Advanced Information Systems. 6. 48-56. 10.20998/2522-9052.2022.2.09.

3. Maria Muntean. 8 Best Programming Languages for Cybersecurity [2023 Guide]. https://www.springboard.com/blog/cybersecurity/best-programming-language-for-cybersecurity/

4. Perwej, Dr. Yusuf & Abbas, Qamar & Dixit, Jai & Akhtar, Nikhat & Jaiswal, Anurag. (2021). A Systematic Literature Review on the Cyber Security. International Journal of Scientific Research and Management. Volume 9. Pages 669 - 710. 10.18535/ijsrm/v9i12.ec04.

5. Олексій Волошин. Переваги і недоліки мови Python. https://blog.ithillel.ua/articles/perevagi-i-nedoliki-movi-python

Micha Gorelick and Ian Ozsvald. High Performance Python. O'Reilly Media, Inc., 2014. 370 p.

Надійшла: 27.07.2023 Рецензент: д.т.н., професор Кожухівський А.Д.