МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

М. І. Ільїн, Д. І. Якобчук

ВСТУП ДО АНАЛІЗУ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лабораторний практикум

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського як навчальний посібник для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою “Системи, технології та математичні методи кібербезпеки” спеціальності 125 “Кібербезпека” та освітніми програмами “Математичні методи моделювання, розпізнавання образів та комп’ютерного зору”, “Математичні методи криптографічного захисту інформації” спеціальності 113 “Прикладна математика”

Київ
КПІ ім. Ігоря Сікорського
2021

Рецензенти	*Шелест М. Є.*, доктор техн. наук, проф. кафедри кібербезпеки та математичного моделювання Національного університету "Чернігівська політехніка"
	Писарчук О. О., доктор техн. наук, проф. кафедри обчислювальної техніки ФІОТ КПІ ім. Ігоря Сікорського
	Коломицев М. В., канд. техн. наук, доц. кафедри інформаційної безпеки НН ФТІ КПІ ім. Ігоря Сікорського
Відповідальний редактор	*Стьопочкіна І. В.*, канд. техн. наук, доц.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол №2 від 09.12.2021 р.) за поданням Вченої ради Навчально-наукового фізико-технічного інституту (протокол №20 від 29.11.2021 р.)

Електронне мережне навчальне видання

Ільїн Микола Іванович, канд. техн. наук
Якобчук Дмитро Ігорович

ВСТУП ДО АНАЛІЗУ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лабораторний практикум

Вступ до аналізу шкідливого програмного забезпечення: Лабораторний практикум [Електронний ресурс]: навч. посіб. для студ. спеціальностей 125 "Кібербезпека", 113 "Прикладна математика" / М. І. Ільїн, Д. І. Якобчук; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1 Мбайт). - Київ: КПІ ім. Ігоря Сікорського, 2021. - 31 с.

Навчальна дисципліна присвячена основам аналізу сучасного шкідливого програмного забезпечення (ШПЗ), оберненого проектування виконуваних файлів Windows та документів Microsoft Office, динамічного аналізу ШПЗ із застосуванням налагоджувача та поведінкового аналізу у пісочниці, застосуванню приманок, аналізу інцидентів із застосуванням ШПЗ, аналізу загроз і ознак компрометації цільової системи, технічній розвідці загроз.

Зміст

Вступ

Дякуємо, що відкрили методичні вказівки до лабораторних робіт з курсу "Вступ до аналізу шкідливого програмного забезпечення".

Навчальна дисципліна присвячена основам аналізу сучасного ШПЗ, оберненого проектування (reverse engineering) виконуваних файлів Windows та документів Microsoft Office, динамічного аналізу ШПЗ із застосуванням налагоджувача та поведінкового аналізу у пісочниці (malware sandbox), застосуванню приманок (honeypot), аналізу інцидентів із застосуванням ШПЗ (malware memory and disk forensics), аналізу загроз (endpoint detection) і ознак компрометації цільової системи (indicators of compromise), технічній розвідці загроз (technical threat intelligence).

Особливістю курсу є посилена активна складова захисту. В тому числі, досліджуються компоненти, що потенційно можуть бути використані для розробки ШПЗ та незаконного втручання в роботу комп'ютерів, систем та мереж. В Україні створення з метою використання, розповсюдження або збуту шкідливих програмних чи технічних засобів, а також їх розповсюдження або збут є кримінальним злочином (ст. 361-1 Кримінального кодексу), так само як і незаконне втручання в роботу електронно-обчислювальних машин (комп'ютерів), систем та комп'ютерних мереж (ст. 361).

Додаткова література з курсу:

- Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software [1];
- The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory [2];
- Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code [3];
- Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats [4];
- Practical Binary Analysis [5];
- The Ghidra Book: The Definitive Guide [6];
- The IDA Pro Book, 2nd Edition: The Unofficial Guide to the World's Most Popular Disassembler [7];
- Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware [8];

- Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware [9].

Додаткові матеріали до лабораторних робіт, матеріали для завантаження публікуються на сайті Лабораторії технічної інформаційної безпеки (https://infosec.kpi.ua) та Telegram групі курсу (https://t.me/crdf_re). Консультації можна отримати у групі та лабораторії 311-11 (розклад консультацій уточнюйте).

В посібнику варіант завдання – Ваш номер в списку групи за модулем кількість завдань. Звіт має містити вихідні коди, виконані команди та вивід (для консольних застосувань) або скріншоти (для графічних), коментарі до виконаних дій. Результати можна подавати в електронному вигляді. У випадку, коли обсяг перевищує 1 Мб, використовуйте зовнішні сховища (наприклад, https://mega.nz). Якщо передається скомпільований код, створюйте архів з випадковим паролем і шифруванням імен файлів. Згенерувати пароль можна, наприклад, за допомогою OpenSSL:

```
$ openssl rand -base64 12
YYIKI2bf6ahZAXeL
```

Приклади шифрування архіву з 7-Zip, OpenSSL, GPG:

```
$ 7z a -mhe -pYYIKI2bf6ahZAXeL lab_report.7z *
$ openssl enc -aes256 -pbkdf2 -in lab_report.tar -out out.enc
$ gpg -c lab_report.zip
```

Приклад роботи з хмарними сховищами за допомогою Rclone [10]:

```
$ rclone config
$ rclone copy lab_report mega:lab_report -q --ignore-existing --auto-confirm
    --multi-thread-streams 12 --transfers 12
```

Контактна інформація:

- Лекції – Микола Іванович Ільїн,
  Email m.ilin@kpi.ua, Telegram @mykola_ilin, Threema 2SS7EYDB;
- Лабораторний практикум – Дмитро Ігорович Якобчук,
  Email d.yakobchuk@kpi.ua, Threema TADKETKX;
- Асистенти – А.Войцеховський, Д.Мороз, О.Костюковець (всі, хто має статус адміністратора у @crdf_re).

Сподіваємось на співробітництво та ефективну роботу.

Лабораторна робота 1

Розгортання середовища аналізу ШПЗ

1.1 Мета роботи

Налаштування середовища аналізу ШПЗ.

Malware analysis lab setup. Learn how to prepare malware analysis environment, isolate network, prepare common analysis tools.

1.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- REMnux: A Linux Toolkit for Malware Analysis [11];
- FLARE VM [12];
- Tor TransparentProxy [13];
- kalitorify: Transparent proxy through Tor for Kali Linux OS [14].

Інсталюйте одну з систем віртуалізації, на вибір:

- VMware Workstation/Player [15];
- Oracle VM VirtualBox [16].

Завантажте та додайте в систему віртуалізації образи віртуальних машин:

- Windows 10 development environment [17];
- Windows 10 free VM [18];
- Kali Linux Virtual Machine [19];
- REMnux virtual appliance [20].

Оновіть віртуальні машини, за допомогою Windows Update та

```
$ sudo apt update
$ sudo apt dist-upgrade -y
```

у Kali (на основі Debian testing) та REMnux (на основі Ubuntu) або

```
$ remnux upgrade
```

у випадку REMnux.

Інсталюйте FLARE VM у Windows 10 development environment.

Налаштуйте ізольовану мережу, використовуючи Kali VM як шлюз (з 2 інтерфейсами – зовнішнім NAT, та внутрішнім в окремому сегменті LAN), і Windows VM в якості середовища аналізу (в створеному сегменті LAN). На шлюзі налаштуйте прозоре перенаправлення в Tor (конфігурація Anonymizing Middlebox) та VPN (OpenVPN клієнт, з репозиторію Kali або [21]).

1.3 Завдання

1. Після налаштування лабораторії ввімкніть Tor, перевірте витоки інформації у браузері Windows VM за допомогою:
   - `https://whoer.net`
   - `http://f.vision`
2. Перевірте те ж саме для OpenVPN, використовуючи безкоштовний або тестовий доступ провайдера VPN, наприклад FreeOpenVPN [22].
3. Збережість стан віртуальних машин (snapshot “clean system”) після закінчення тестування. Відновлюйтесь до нього по завершенню динамічного аналізу зразків ШПЗ.
4. (підвищеної складності) Інсталюйте пісочницю CAPEv2 [23].

Лабораторна робота 2

Застосування мови Python для аналізу ШПЗ

2.1 Мета роботи

Отримати навички використання Python 3 для дослідження ШПЗ.

Python basics for malware analysis. Learn how to programmatically analyse Windows executable files, Microsoft Office documents, malware C&C communications.

2.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- Black Hat Python, 2nd Edition: Python Programming for Hackers and Pentesters [24];
- Identifying almost identical files using context triggered piecewise hashing [25];
- PE Format [26];
- LIEF: Library to Instrument Executable Formats [27];
- Keystone: lightweight multi-platform, multi-architecture assembler framework [28];
- Requests: HTTP for Humans [29], HTTPX [30].

2.2.1 Аналіз виконуваних файлів Windows PE

Статичний аналіз PE. Ознайомтесь з матеріалами:

- pefile [31, 32];
- python-magic [33, 34, 35];
- python-ssdeep [36, 37].

Реалізуйте скрипт Python 3, що аналізує виконувані файли PE/PE+ за списком, та визначає:

- дату та час створення виконуваного файлу за даними заголовку (TimeDateStamp, ...);
- наявність і тип ресурсів (PE resources);
- вміст ресурсів за сигнатурою (libmagic);
- розраховує та порівнює нечіткі хеші секцій PE у всіх файлах.

Сигнатури PEiD. Ознайомтесь з матеріалами:

- LIEF Python API, PE Parser [38];
- Сигнатури PEiD userdb.txt [39];
- keystone-engine [40].

Реалізуйте скрипт Python 3, що модифікує виконуваний файл для обходу сигнатур PEiD, прив'язаних до точки входу (`ep_only=true`). Один з варіантів – у сегменті коду знайти невикористовувану область (code cave: вирівнювання 0 в кінці сегменту, послідовності 0xCC/int3 достатнього розміру, тощо), встановити точку входу (PE EntryPoint) на неї, записати код передачі керування на оригінальну точку входу (перехід відносно поточної інструкції, враховуйте ASLR).

Статичний аналіз документів Microsoft Office. Ознайомтесь з матеріалами:

- python-oletools [41];
- exif [42, 43, 44];
- hashlib [45].

Реалізуйте скрипт мовою Python 3, що аналізує документи Microsoft Office у форматі OLE2 за списком, розраховує SHA-256 для потоків OLE, у випадку, якщо у потоці зображення – виводить метаданні EXIF.

2.2.2 Аналіз механізмів віддаленого керування ШПЗ

Ознайомтесь з матеріалами:

- subprocess.Popen [46];
- PyNaCl [47, 48, 49, 50];
- py2exe [51], PyInstaller [52];
- socat [53].

Реалізуйте систему віддаленого керування, що виконує команди операційної системи і повертає результати (subprocess.Popen). Команди отримуються, результати надсилаються у зашифрованому вигляді (nacl.public.Box). Створіть виконуваний зразок за допомогою py2exe або PyInstaller. Замість реалізації мережевих комунікацій достатньо консольного вводу/виводу та перенаправлень socat TCP, TCP-LISTEN, EXEC.

2.2.3 Автоматизація аналізу даних відкритих джерел

Ознайомтесь з матеріалами:

- WiGLE [54, 55];
- ipyleaflet [56];
- Google Earth [57, 58];
- simplekml [59].

Реалізуйте скрипт мовою Python 3, що за даними оточення WiFi (результат команди `iwlist scanning` у Kali Linux з підключеним бездротовим адаптером) знаходить географічне положення системи (за даними wigle.net). Відобразіть положення на мапі (у ноутбуці Jupyter [60] та ipyleaflet). Експортуйте BSSID точок доступу з геопривязкою у KML, відобразіть у Google Earth.

2.3 Завдання

1. В 2.2.1 проаналізуйте зразки Windows EXE, DLL, Microsoft Office DOC за варіантом (набір ШПЗ надається на занятті, множини не перетинаються).
2. В 2.2.1 застосуйте модифікацію сигнатури точки входу для виконуваних файлів за варіантом, перевірте працездатність модифікованих зразків, порівняйте результати сигнатурного аналізу TrID [61], DIE [62].
3. В 2.2.2 додайте комунікаційний канал за варіантом:
   3.1. Telegram (private group, приклади [63]);
   3.2. Twitter (приклади [64]);
   3.3. Pastebin (приклад [65]);
   3.4. Google Docs/GMail (приклади [66, 67]).
4. В 2.2.3 оцініть з якою точністю визначається Ваше положення у випадку витоку інформації про WiFi оточення.

Лабораторна робота 3

Основи статичного аналізу ШПЗ

3.1 Мета роботи

Отримати навички статичного аналізу ШПЗ у зараженій системі та засобів доставки на основі документів Microsoft Office.

Basic static analysis. Learn basics how to analyse code without execution of malware sample.

3.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- The Sleuth Kit (TSK) [68];
- OSFMount [69];
- ClamAV [70], ClamWin [71];
- RegRipper3.0 [72].

Проаналізуйте образ диску зараженої системи, визначіть структуру розділів (Kali Linux `fdisk`, `parted`, TSK), видалені файли (TSK) та змонтуйте файлову систему (`mount -o loop,ro`).

Проаналізуйте зараження системи за допомогою ClamAV clamscan, Windows Security (змонтуйте образ у режимі тільки читання за допомогою OSFMount) або іншого антивіруса (наприклад, Bitdefender Antivirus Free Edition [73]). Знайдіть виконувані файли ШПЗ.

Проаналізуйте механізм закріплення ШПЗ у реєстрі за допомогою RegRipper.

Ознайомтесь з матеріалами:

- bulk_extractor [74, 75];
- flare-floss [76].

Проаналізуйте текстові рядки знайденого зразка ШПЗ за допомогою `strings`, `bulk_extractor` та `flare-floss`. Зробіть припущення про IP адресу центру керування.

Ознайомтесь з матеріалами:

- pestudio [77];
- PE-bear [78];
- Exeinfo PE [79].

Проаналізуйте артефакти виконуваного файлу ШПЗ, зробіть припущення про використаний компілятор, дату та час створення файлу, реалізований функціонал (зверніть увагу на імпортовані функції [80]).

Ознайомтесь з матеріалами:

- IDA Free [81], уточніть наявність у Лабораторії Educational ліцензії [82] на поточний рік;
- Ghidra [83].

За допомогою дизасемблера та декомпілятора проаналізуйте алгоритми закріплення в системі та взаємодії з центром керування ШПЗ.

Ознайомтесь з матеріалами:

- olevba [84];
- mraptor [85].

Проаналізуйте документи за списком, визначіть механізм запуску шкідливого навантаження.

3.3 Завдання

1. Проаналізуйте образ диску зараженої системи, що надається на занятті.
2. Проаналізуйте зразки документів Microsoft Office за варіантом (набір ШПЗ надається на занятті, множини не перетинаються).

Лабораторна робота 4

Основи динамічного аналізу ШПЗ

4.1 Мета роботи

Отримати навички динамічного аналізу виконуваних файлів та документів Microsoft Office.

Basic dynamic analysis. Learn basics how to analyse malware sample in runtime.

4.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- Sysinternals [86]: Process Explorer [87], Process Monitor [88], TCPView [89], WinObj [90], AutoRuns [91];
- Noriben Malware Analysis Sandbox [92];
- MITRE ATT&CK [93].

Проаналізуйте зразки ШПЗ за списком, визначіть породжувані процеси (Process Explorer, Process Monitor), мережеві з'єднання з центром керування (TCPView), об'єкти синхронізації (що використовуються для забезпечення одиничного запуску процесу ШПЗ, WinObj), методи закріплення у системі (Autoruns). Дослідіть поведінку за допомогою Noriben. Класифікуйте результати за MITRE ATT&CK.

Ознайомтесь з матеріалами:

- hollows_hunter [94];
- Wireshark [95];
- NetworkMiner [96];
- FakeNet-NG [97] у складі FLARE VM.

Проаналізуйте зразки ШПЗ за списком, визначіть методи інжектування у системні процеси (hollows_hunter), які дані передаються до центру керування (Wireshark, NetworkMiner). Дослідіть мережеву активність ШПЗ за допомогою FakeNet-NG (у складі FLARE VM) або INetSim.

Ознайомтесь з матеріалами:

- x64dbg [98];
- ScyllaHide [99], TitanHide [100];
- WinDbg [101, 102];
- Джерела інформації про процеси у psxview [103].

Проаналізуйте зразки ШПЗ у налагоджувачі (x64dbg, WinDbg). Для завантаження DLL зверніть увагу на `rundll32` та Remote DLL [104]. Дослідіть методи протидії налагодженню зразків (ScyllaHide, TitanHide). Знайдіть налаштування з'єднання з центром керування у пам'яті процесу ШПЗ, порівняйте з результатами динамічного аналізу вище.

Дослідіть засоби маскування процесів ШПЗ шляхом порівняння списків процесів зі структур, що описані у [103], за допомогою WinDbg на рівні ядра у реальному часі (live kernel mode у локальній системі).

Ознайомтесь з матеріалами:

- Advanced VBA Macros Attack & Defence [105];
- ViperMonkey [106];
- vhook [107].

Проаналізуйте зразки шкідливих документів Microsoft Office. Визначіть механізм доставки і закріплення у цільовій системі.

4.3 Завдання

1. Проаналізуйте зразки Windows EXE, DLL, Microsoft Office DOC за варіантом (набір ШПЗ надається на занятті, множини не перетинаються).

Лабораторна робота 5

Аналіз образу пам'яті систем з активним ШПЗ

5.1 Мета роботи

Отримати навички аналізу активного ШПЗ у зараженій системі.

Live malware memory analysis. Learn memory forensics for live malware analysis.

5.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- Volatility 3 [108], 2 [109];
- Volatility Workbench [110];
- WinPmem [111].

Запустіть зразок ШПЗ (ransomware) у ізольованому середовищі (ЛР 1). Зробіть зліпок оперативної пам'яті системи (WinPmem). Праналізуйте зліпок за допомогою Volatility, знайдіть процес ШПЗ, отримайте виконуваний файл та дамп пам'яті процесу. Дослідіть виконуваний файл за допомогою статичного аналізу (ЛР 3), знайдіть алгоритм шифрування. Знайдіть ключі шифрування у дампі пам'яті (для деяких криптоалгоритмів існують засоби автоматизації, наприклад, AES [112], RSA [112, 113]).

Реалізуйте скрипт Python 3, що отримує ключі шифрування зі зліпку пам'яті зараженої системи (volatility3 завантаження образу, yarascan, читання пам'яті процесу) та розшифровує файли у заданому каталозі та підкаталогах (рекурсивно).

5.3 Завдання

1. Проаналізуйте зразки за варіантом (ШПЗ надається на занятті).

Лабораторна робота 6

Ознаки компрометації

6.1 Мета роботи

Отримати навички роботи з індикаторами компрометації.

Indicators of Compromise. Learn how to extract indicators for unknown malware sample, analyse threat intelligence feeds for IoCs, check IoC on endpoint.

6.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- YARA [114];
- Приклади сигнатур YARA [115];
- Apache Tikka [116, 117];
- Tesseract-OCR [118, 119, 120].

Проаналізуйте зразки ШПЗ за списком (PE EXE). Побудуйте YARA сигнатури, що враховують виконуваний код у точці входу, рядки у Unicode (wide), наявність специфічних ресурсів (PE.resources), ентропію секції коду (math.entropy), хеш (hash.sha256).

Проаналізуйте зразки ШПЗ за списком (документи Microsoft Office). Реалізуйте скрипт Python 3, який детектує обфусковані зразки шляхом роспізнавання у документі фішингового зображення з проханням активувати макроси. Для аналізу документів застосуйте tikka-python, розпізнавання тексту pytesseract. Порівняйте результати з сигнатурами Halogen [121].

Ознайомтесь з матеріалами:

- VirusTotal [122]: API [123], vt-py [124];
- MISP [125]: default feeds [126], PyMISP [127];
- Munin [128].

Реалізуйте скрипт Python 3, який шукає отримані вище хеші у VirusTotal (vt-py) та каналах MISP (PyMISP, default feeds). Порівняйте результати з Munin.

Ознайомтесь з матеріалами:

- Інтерфейс командного рядка YARA [129];
- Плагін Volatility yarascan [130, 131];
- Loki [132], THOR Lite [133].

Проаналізуйте системи з ЛР 1, оцініть кількість хибних спрацювань розроблених сигнатур у чистій системі.

6.3 Завдання

1. Проаналізуйте зразки за варіантом (ШПЗ надається на занятті).
2. (підвищеної складності) Реалізуйте засіб протидії Halogen [121] шляхом перекодування та додавання шуму до вбудованих зображень у документах Microsoft Office.

Лабораторна робота 7

Аналіз загроз у цільовій системі

7.1 Мета роботи

Отримати навички використання ETW для детектування ШПЗ.

Endpoint threat detection. Learn Event Tracing for Windows (ETW) capabilities for threat detection.

7.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- Sysmon [134, 135], MSTIC Sysmon Resources [136];
- ThreatHunter-Playbook [137], sysmon-modular [138];
- Windows Defender Exploit Guard events [139];
- Atomic Red Team [140, 141, 142].

Розгорніть Sysmon у тестовій системі з ЛР 1, налаштуйте обробку подій WDEG, отримання журналів подій у іншій системі. Перевірте роботу на тестах Invoke-AtomicRedTeam.

Ознайомтесь з матеріалами:

- pywin32 win32evtlog [143];
- python-evtx [144].

Розробіть скрипт Python 3, що оброблює події у віддаленій системі і виявляє інжектування в процеси (у AtomicRedTeam тести T1055 Process Injection, T1055.001 Dynamic-link Library Injection, T1055.004 Asynchronous Procedure Call, T1055.012 Process Hollowing).

Ознайомтесь з матеріалами:

- Metasploit [145], windows/meterpreter/reverse_https [146];

- Veil [147];
- Windows Process Injection [148, 149];
- The C2 Matrix [150].

Створіть зразок системи віддаленого керування у Metasploit, навантаження Meterpreter, шаблон виконуваного файлу exe-small. Запустіть в системі з налаштованим раніше Sysmon. Проаналізуйте події при детектуванні зразка антивірусом (Windows Security).

Додайте до зразку обфускацію (Veil), замініть корисне навантаження на завантаження і запуск системи керування на Ваш вибір (приклади C2 Matrix). Проаналізуйте події при запуску зразка.

Розгляньте ознаки інжектування в процеси методами, що не тестувалися AtomicRedTeam вище (приклади Windows Process Injection).

7.3 Завдання

1. При дослідженні зразку системи керування отримайте виконуваний файл, що не детектується Windows Security. Оцініть реакцію інших антивірусів у VirusTotal.
2. (підвищеної складності) Додайте у скрипт аналізу подій підтримку правил Sigma [151, 152].
3. (підвищеної складності) Змоделюйте активну MITM атаку на систему аналіза (scapy [153] з реалізацією ARP спуфінгу, mitmproxy [154] TCP proxy, Impacket [155]), підмініть інформацію про події, що ведуть до детектування інжектування в процеси.

Лабораторна робота 8

Застосування приманок для аналізу ШПЗ

8.1 Мета роботи

Отримати навички виявлення атак та аналізу ШПЗ за допомогою приманок.

Honeypots. Learn malware honeypot deployment, analysis and evasion basics.

8.2 Порядок виконання роботи

Ознайомтесь з матеріалами:

- Awesome Deception [156];
- Awesome Honeypots [157].

Налаштуйте сервер з прямою IP адресою, ОС Debian або Ubuntu x86_64 останньої версії. Достатньо безкоштовних пропозицій хмарних провайдерів, наприклад:

- Digital Ocean, Azure у GitHub Student Developer Pack [158];
- GCP Free Tier [159];
- Amazon Free Tier [160];
- Azure free account [161].

Інсталюйте приманки, по одній з класу [157]:

1. Вразливий сервіс SSH;
2. Вразлива система керування базою даних (DBMS);
3. Вразливий пристрій інтернету речей (IoT) або компоненти критичної інфраструктури АСК ТП (ICS/SCADA).

Отримайте та проаналізуйте 3-4 зразка ШПЗ з детектованих атак.

Розробіть скрипт Python 3, що виявляє розгорнуті приманки за результатами поведінкового аналізу (приклади у [162]), і відрізняє від сервісів ОС зі списку.

8.3 Завдання

1. Список ОС для порівняння з приманками за варіантом, остання стабільна версія на час виконання ЛР:
   - 1.1. Debian stable;
   - 1.2. Ubuntu Server;
   - 1.3. Oracle Linux;
   - 1.4. FreeBSD;
   - 1.5. OpenBSD;
   - 1.6. Windows 10 з вбудованим OpenSSH сервером.

Список джерел

[1] Sikorski Michael, Honig Andrew. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. — 1st вид. — USA : No Starch Press, 2012. — ISBN: 1593272901.

[2] The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory / Michael Hale Ligh, Andrew Case, Jamie Levy, Aaron Walters. — 1st вид. — Wiley Publishing, 2014. — ISBN: 1118825098.

[3] Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code / Michael Ligh, Steven Adair, Blake Hartstein, Matthew Richard. — Wiley Publishing, 2010. — ISBN: 0470613033.

[4] Matrosov A., Rodionov E., Bratus S. Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats. — No Starch Press, 2019. — ISBN: 9781593278830. — Режим доступу: https://books.google.com.ua/books?id=xzGLDwAAQBAJ.

[5] Andriesse D. Practical Binary Analysis: Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly. — No Starch Press, 2018. — ISBN: 9781593279127. — Режим доступу: https://books.google.com.ua/books?id=laWgswEACAAJ.

[6] Eagle C., Nance K. The Ghidra Book: The Definitive Guide. — No Starch Press, 2020. — ISBN: 9781718501034. — Режим доступу: https://books.google.com.ua/books?id=Bm_cDwAAQBAJ.

[7] Eagle Chris. The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler. — USA : No Starch Press, 2011. — ISBN: 1593272898.

[8] Mohanta A., Saldanha A. Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware. — Apress, 2020. — ISBN: 9781484261927. — Режим доступу: https://books.google.com.ua/books?id=TDGWzQEACAAJ.

[9] K A Monnappa. Learning Malware Analysis: Explore the Concepts, Tools, and Techniques to Analyze and Investigate Windows Malware. — Birmingham : Packt Publishing, Limited, 2018. — ISBN: 1788392507.

[10] Rclone. — Режим доступу: https://rclone.org.

[11] REMnux: A Linux Toolkit for Malware Analysis. — Режим доступу: https://remnux.org.

[12] FLARE VM. — Режим доступу: https://github.com/mandiant/flare-vm.

[13] Tor: TransparentProxy. — Режим доступу: https://gitlab.torproject.org/legacy/trac/-/wikis/doc/TransparentProxy.

[14] kalitorify. — Режим доступу: https://github.com/brainfucksec/kalitorify.

[15] VMware Workstation Pro. — Режим доступу: https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html.

[16] VirtualBox. — Режим доступу: https://www.virtualbox.org.

[17] Windows 10 development environment. — Режим доступу: https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/.

[18] Virtual Machines - Microsoft Edge Developer. — Режим доступу: https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/.

[19] Kali Linux Virtual Machines. — Режим доступу: https://www.kali.org/get-kali/#kali-virtual-machines.

[20] REMnux virtual appliance. — Режим доступу: https://docs.remnux.org/install-distro/get-virtual-appliance.

[21] OpenVPN. — Режим доступу: https://community.openvpn.net/ openvpn/ wiki/ OpenvpnSoftwareRepos.

[22] FreeOpenVPN. — Режим доступу: https://www.freeopenvpn.org.

[23] CAPEv2: Malware Configuration And Payload Extraction. — Режим доступу: https://github.com/kevoreilly/CAPEv2.

[24] Seitz J., Arnold T. Black Hat Python, 2nd Edition: Python Programming for Hackers and Pentesters. — No Starch Press, 2021. — ISBN: 9781718501126. — Режим доступу: https://books.google.com.ua/books?id=HaqdzQEACAAJ.

[25] Kornblum Jesse. Identifying almost identical files using context triggered piecewise hashing // Digital Investigation. — 2006. — Т. 3. — С. 91–97. — The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06). Режим доступу: https://www.sciencedirect.com/science/article/pii/S1742287606000764.

[26] PE Format. — Режим доступу: https://docs.microsoft.com/en-us/windows/win32/debug/pe-format.

[27] LIEF: Library to Instrument Executable Formats. — Режим доступу: https://lief-project.github.io/.

[28] Keystone: lightweight multi-platform, multi-architecture assembler framework. — Режим доступу: https://www.keystone-engine.org/.

[29] Requests: HTTP for Humans. — Режим доступу: https://docs.python-requests.org/en/latest/.

[30] HTTPX. — Режим доступу: https://www.python-httpx.org.

[31] Python PE parsing module. — Режим доступу: https://pypi.org/project/pefile/.

[32] pefile. — Режим доступу: https://github.com/erocarrera/pefile.

[33] File type identification using libmagic. — Режим доступу: https://pypi.org/project/python-magic/.

[34] python-magic. — Режим доступу: https://github.com/ahupp/python-magic.

[35] file(1). — Режим доступу: http://www.darwinsys.com/file/.

[36] Python wrapper for the ssdeep library. — Режим доступу: https://pypi.org/project/ssdeep/.

[37] python-ssdeep. — Режим доступу: https://github.com/DinoTools/python-ssdeep.

[38] LIEF Python API: PE Parser. — Режим доступу: https://lief-project.github.io/doc/latest/api/python/pe.html.

[39] PE iDentifier. — Режим доступу: https://github.com/wolfram77web/app-peid.

[40] keystone-engine. — Режим доступу: https://pypi.org/project/keystone-engine/.

[41] oletools: python tools to analyze OLE and MS Office files. — Режим доступу: http://www.decalage.info/python/oletools.

[42] Read and modify image EXIF metadata using Python. — Режим доступу: https://pypi.org/project/exif/.

[43] exif. — Режим доступу: https://gitlab.com/TNThieding/exif.

[44] ExifTool. — Режим доступу: https://github.com/exiftool/exiftool.

[45] hashlib. — Режим доступу: https://docs.python.org/3/library/hashlib.html.

[46] subprocess: Subprocess management. — Режим доступу: https://docs.python.org/3/library/subprocess.html#popen-constructor.

[47] Python binding to the Networking and Cryptography (NaCl) library. — Режим доступу: https://pypi.org/project/PyNaCl/.

[48] libsodium: A modern, portable, easy to use crypto library. — Режим доступу: https://github.com/jedisct1/libsodium.

[49] TweetNaCl: auditable high-security cryptographic library. — Режим доступу: https://tweetnacl.cr.yp.to/.

[50] TweetNaCl.js: a port of TweetNaCl / NaCl to JavaScript. — Режим доступу: https://tweetnacl.js.org.

[51] py2exe. — Режим доступу: http://www.py2exe.org/.

[52] PyInstaller. — Режим доступу: http://www.pyinstaller.org/.

[53] socat: Multipurpose relay. — Режим доступу: http://www.dest-unreach.org/socat/.

[54] WiGLE: Wireless Network Mapping. — Режим доступу: https://wigle.net.

[55] WiGLE API. — Режим доступу: https://api.wigle.net.

[56] ipyleaflet: A Jupyter / Leaflet bridge enabling interactive maps in the Jupyter notebook. — Режим доступу: https://github.com/jupyter-widgets/ipyleaflet.

[57] Google Earth. — Режим доступу: https://earth.google.com.

[58] Import KML map data into Google Earth. — Режим доступу: https://support.google.com/earth/answer/7365595.

[59] simplekml. — Режим доступу: https://pypi.org/project/simplekml/.

[60] Jupiter. — Режим доступу: https://jupyter.org.

[61] TrID: File Identifier. — Режим доступу: https://mark0.net/soft-trid-e.html.

[62] Detect It Easy. — Режим доступу: https://github.com/horsicq/Detect-It-Easy.

[63] Telegram Bot Code Examples: Python. — Режим доступу: https://core.telegram.org/bots/samples#python.

[64] Twitter Official v2 tools and libraries. — Режим доступу: https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries/v2.

[65] Python Pastebin API interaction object. — Режим доступу: https://pypi.org/project/Pastebin/.

[66] Gcat. — Режим доступу: https://github.com/byt3bl33d3r/gcat.

[67] Gdog. — Режим доступу: https://github.com/maldevel/gdog.

[68] The Sleuth Kit (TSK). — Режим доступу: https://www.sleuthkit.org/sleuthkit/.

[69] OSFMount. — Режим доступу: https://www.osforensics.com/tools/mount-disk-images.html.

[70] ClamAV: an open-source antivirus engine. — Режим доступу: https://www.clamav.net/.

[71] ClamWin Free Antivirus. — Режим доступу: https://clamwin.com/.

[72] RegRipper3.0. — Режим доступу: https://github.com/keydet89/RegRipper3.0.

[73] Bitdefender Antivirus Free Edition. — Режим доступу: https://www.bitdefender.com/solutions/free.html.

[74] bulk_extractor. — Режим доступу: https://github.com/simsong/bulk_extractor.

[75] Garfinkel Simson L. Digital Media Triage with Bulk Data Analysis and Bulk_extractor // Comput. Secur. — 2013. — Feb. — Т. 32, № С. — С. 56–72.

[76] FLARE Obfuscated String Solver. — Режим доступу: https://github.com/mandiant/flare-floss.

[77] pestudio. — Режим доступу: https://www.winitor.com/.

[78] PE-bear. — Режим доступу: https://hshrzd.wordpress.com/pe-bear/.

[79] Exeinfo PE for Windows. — Режим доступу: http://www.exeinfo.xn.pl/.

[80] MalAPI.io. — Режим доступу: https://malapi.io/.

[81] IDA Freeware. — Режим доступу: https://hex-rays.com/ida-free/.

[82] IDA Educational. — Режим доступу: https://hex-rays.com/educational/.

[83] Ghidra software reverse engineering (SRE) suite. — Режим доступу: https://ghidra-sre.org/.

[84] olevba. — Режим доступу: https://github.com/decalage2/oletools/ wiki/ olevba.

[85] mraptor (MacroRaptor). — Режим доступу: https://github.com/decalage2/oletools/wiki/mraptor.

[86] Sysinternals. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/.

[87] Process Explorer. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer.

[88] Process Monitor. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/downloads/procmon.

[89] TCPView. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/downloads/tcpview.

[90] WinObj. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/downloads/winobj.

[91] Autoruns for Windows. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns.

[92] Noriben Malware Analysis Sandbox. — Режим доступу: https://github.com/Rurik/Noriben.

[93] MITRE ATT&CK. — Режим доступу: https://attack.mitre.org/.

[94] hollows_hunter. — Режим доступу: https://github.com/ hasherezade/ hollows_hunter.

[95] Wireshark. — Режим доступу: https://www.wireshark.org/.

[96] NetworkMiner. — Режим доступу: https://www.netresec.com/?page=networkminer.

[97] FakeNet Genie: Improving Dynamic Malware Analysis with Cheat Codes for FakeNet-NG. — Режим доступу: https://www.mandiant.com/resources/improving-dynamic-malware-analysis-with-cheat-codes-for-fakenet-ng.

[98] x64dbg. — Режим доступу: https://x64dbg.com/.

[99] ScyllaHide. — Режим доступу: https://github.com/x64dbg/ScyllaHide.

[100] TitanHide. — Режим доступу: https://github.com/mrexodia/TitanHide.

[101] Debugging Tools for Windows. — Режим доступу: https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/debugger-download-tools.

[102] Debugging Malware with WinDbg. — Режим доступу: https://blogs.keysight.com/ blogs/ tech/ nwvs.entry.html/ 2020/ 07/ 27/ debugging_malwarewi-hk5u.html.

[103] psxview. — Режим доступу: https://github.com/ volatilityfoundation/ volatility/ wiki/ Command-Reference-Mal#psxview.

[104] Remote DLL. — Режим доступу: https://securityxploded.com/remotedll.php.

[105] Advanced VBA Macros Attack & Defence - Black Hat Europe 2019. — Режим доступу: http://www.decalage.info/en/bheu2019.

[106] ViperMonkey. — Режим доступу: https://github.com/decalage2/ViperMonkey.

[107] VBA Dynamic Hook. — Режим доступу: https://github.com/eset/vba-dynamic-hook.

[108] Volatility 3. — Режим доступу: https://volatility3.readthedocs.io.

[109] Volatility 2. — Режим доступу: https://github.com/ volatilityfoundation/ volatility/ wiki.

[110] Volatility Workbench. — Режим доступу: https://www.osforensics.com/tools/volatility-workbench.html.

[111] The Pmem Suite. — Режим доступу: https://winpmem.velocidex.com.

[112] Automatic key-finding. — Режим доступу: https://citp.princeton.edu/our-work/memory/code/.

[113] RSA key-restore tools. — Режим доступу: https://github.com/einaros/heartbleed-tools.

[114] YARA's documentation. — Режим доступу: https://yara.readthedocs.io.

[115] Signature-Base. — Режим доступу: https://github.com/Neo23x0/signature-base.

[116] Apache Tika - a content analysis toolkit. — Режим доступу: https://tika.apache.org.

[117] Tika-Python. — Режим доступу: https://github.com/chrismattmann/tika-python.

[118] Tesseract. — Режим доступу: https://tesseract-ocr.github.io/tessdoc/.

[119] Tess4J. — Режим доступу: http://tess4j.sourceforge.net.

[120] Python-tesseract. — Режим доступу: https://pypi.org/project/pytesseract/.

[121] Halogen. — Режим доступу: https://github.com/target/halogen.

[122] VirusTotal. — Режим доступу: https://www.virustotal.com.

[123] VirusTotal API. — Режим доступу: https://developers.virustotal.com/reference/ overview.

[124] vt-py: official Python client library for VirusTotal. — Режим доступу: https://github.com/VirusTotal/vt-py.

[125] MISP: Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing. — Режим доступу: https://www.misp-project.org/documentation/.

[126] Default feeds available in MISP. — Режим доступу: https://www.misp-project.org/feeds/.

[127] PyMISP: Python Library to access MISP. — Режим доступу: https://github.com/MISP/PyMISP.

[128] Munin. — Режим доступу: https://github.com/Neo23x0/munin.

[129] Running YARA from the command-line. — Режим доступу: https://yara.readthedocs.io/en/latest/commandline.html.

[130] Volatility 2 yarascan. — Режим доступу: https://github.com/volatilityfoundation/volatility/wiki/Command-Reference-Mal#yarascan.

[131] Volatility 3 yarascan. — Режим доступу: https://volatility3.readthedocs.io/en/stable/volatility3.plugins.yarascan.html.

[132] Loki: Simple IOC and YARA Scanner. — Режим доступу: https://github.com/Neo23x0/LOKI.

[133] THOR Lite. — Режим доступу: https://www.nextron-systems.com/thor-lite/.

[134] Sysmon. — Режим доступу: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon.

[135] MITRE ATT&CK technique coverage with Sysmon for Linux. — Режим доступу: https://techcommunity.microsoft.com/t5/microsoft-sentinel-blog/mitre-att-amp-ck-technique-coverage-with-sysmon-for-linux/ba-p/2858219.

[136] MSTIC Sysmon Resources. — Режим доступу: https://github.com/microsoft/MSTIC-Sysmon.

[137] The ThreatHunter-Playbook. — Режим доступу: https://github.com/OTRF/ThreatHunter-Playbook.

[138] sysmon-modular. — Режим доступу: https://github.com/olafhartong/sysmon-modular.

[139] Exploit Protection Event Documentation. — Режим доступу: https://github.com/palantir/exploitguard.

[140] Atomic Family. — Режим доступу: https://atomicredteam.io/.

[141] Atomic Red Team. — Режим доступу: https://github.com/redcanaryco/atomic-red-team.

[142] Invoke-AtomicRedTeam. — Режим доступу: https://github.com/redcanaryco/invoke-atomicredteam.

[143] Python for Windows (pywin32) Extensions. — Режим доступу: https://github.com/mhammond/pywin32.

[144] python-evtx. — Режим доступу: https://github.com/williballenthin/python-evtx.

[145] Metasploit. — Режим доступу: https://www.metasploit.com/.

[146] windows/meterpreter/reverse_https. — Режим доступу: https://github.com/ rapid7/ metasploit-framework/ blob/ master/ documentation/ modules/ payload/ windows/ meterpreter/ reverse_https.md.

[147] Veil. — Режим доступу: https://github.com/Veil-Framework/Veil.

[148] Windows Process Injection. — Режим доступу: https://github.com/odzhan/injection.

[149] Pinjectra. — Режим доступу: https://github.com/SafeBreach-Labs/pinjectra.

[150] The C2 Matrix. — Режим доступу: https://www.thec2matrix.com.

[151] Sigma: Generic Signature Format for SIEM Systems. — Режим доступу: https://github.com/SigmaHQ/sigma.

[152] pySigma. — Режим доступу: https://github.com/SigmaHQ/pySigma.

[153] Scapy. — Режим доступу: https://scapy.net.

[154] mitmproxy. — Режим доступу: https://mitmproxy.org.

[155] Impacket. — Режим доступу: https://www.secureauth.com/labs/open-source-tools/impacket/.

[156] Awesome Deception. — Режим доступу: https://github.com/tolgadevsec/Awesome-Deception.

[157] Awesome Honeypots. — Режим доступу: https://github.com/paralax/awesome-honeypots.

[158] GitHub Student Developer Pack. — Режим доступу: https://education.github.com/pack.

[159] Google Cloud Free Tier products. — Режим доступу: https://cloud.google.com/free.

[160] AWS Free Tier. — Режим доступу: https://aws.amazon.com/free/.

[161] Azure free account. — Режим доступу: https://azure.microsoft.com/en-us/free/.

[162] Remote detection of low and medium interaction honeypots. — Режим доступу: https://youtu.be/9qw-DichVGg.