УДК 004.056

МЕТОДИ ВІДНОВЛЕННЯ ФАЙЛОВОЇ СИСТЕМИ У АТАКАХ ШИФРУВАЛЬНИКІВ НА ОСНОВІ ДИНАМІЧНОГО АНАЛІЗУ ШАБЛОНІВ ШИФРУВАННЯ

А. В. Войцеховський[1,a]

[1] Навчально-науковий Фізико-технічний інститут

Анотація

В роботі досліджено методи динамічного аналізу програм-вимагачів в хмарних та резервних системах, з метою вдосконалення систем антивірусного захисту та виявлення шкідливого програмного забезпечення на ранньому етапі, попереджаючи про можливу втрату важливих даних. Завдяки завчасно створеному знімку файлової системи, відновлюються файли, що були втрачені до моменту виявлення аномалій.

Ключові слова: шкідливе програмне забезпечення, програма-вимагач, резервні системи

Вступ

З кожним роком програми-вимагачі посилюють свою активність. [1] Все більше користувачів та організацій зберігають свої дані в хмарних сервісах та резервних системах, що можуть бути скомпрометовані, вражають велику кількість інформації, з метою отримання викупу. Хоч і в більшості випадках вдається відновити дані з резервного сховища,проте багато атакованих користувачів втрачають свої дані або змушені надсилати відшкодування в обмін на розблокування даних. Критичним є інфікування систем віртуалізації, що призводить до відмови в роботі та втрати багатьох сервісів. Зловмисники, що раніше атакували Windows системи, створюють аналогічне ШПЗ під Linux, інфікучи системи віртуалізації на базі esxi. В 2022-2023 році були помічені такі сімейства ШПЗ як *Royal*, *Blackbasta*, *Esxiargs*, *Redalert* націлені на VMware esxi гіпервізор. Атаки орієнтовані на широкий спектр галузей, включаючи охорону, засоби масової інформації , уряд, фінансові послуги, освіту та здоров'я. Методи доставки можуть бути різними: використавши вразливість сервісу, за допомогою *Cobalt Strike* чи подібних інструментів, за допомогою надсилання працівникам організації фішингових листів. Проникнувши в директорію `/vmfs/volume`, програми застосовують алгоритми шифрування до таки файлів: *.vmdk, .vmx, .iso .nvram .log.*, що повністю та надовго припиняє роботу віртуальних сервісів. Також були помічені зразки як *icefire* [2],*Blackcat*, *Cl0p*, розроблені для шифрування всіх файлів (*.png .txt .pdf* і т.д.) користувача Linux. Окрім функції шифрування зразки використовують різні методи закріплення в системі, уникнення від детектування антивірусними системами та системами виявлення вторгнення. Тому виникає необхідність вдоскрналювати моделі там методи класифікації даних зразків. В даній роботі пропонується використовувати динамічний аналіз шаблонів роботи шифрувальника в системах Linux та Windows. У разі виявлення підозрілої активності, система припиняє роботу відповідної програми, та відновлює всі інфіковані дані, використовуючи завчасно сформований знімок файлової системи.

Загальні методи класифікації

Одним з базових методів класифікації ШПЗ є статичний аналіз, беручи за атрибути структуру файлу, граф потоку виконання, та різні атрибути. Однак, він почав втрачати свою ефективність з розвитком методів мутації та обфускації зловмисних зразків.[3][4]

Тому більшість досліджень сконцентровані на динамічному та гібридному аналізі, запускаючи ШПЗ в певному ізольованому середовищі, спостерігаючи за його активністю.[5] В ході своєї роботи зразок може виконувати ті функції, що були раніше зашифровані, та не доступні для статичного аналізу. Також можливе поетапне динамічне завантаження додаткового корисного навантаження. Гібридні методи поєднують в собі статичний та динамічний аналіз, використовуючи сильні сторони обох методів. Дозволяють підвищити ефективність при класифікації та виявленню шкідливого програмного забезпечення.

Аналіз ентропії інфікованих файлів

Для виявлення шифрування на ранньому етапі, ефективно аналізувати ентропію змінених файлів, та у випадку виявлення інфікованих, припинити роботу зловмисної програми ,що перезаписала їх. [6] [7]

Відповідно до типу файлу можна застосовувати різні алгоритми обчислення ентропії, такі як ентропія Реньї, Шеннона, за допомогою ланцюга Маркова

[a]a.voitsekhovskyi@kpi.ua

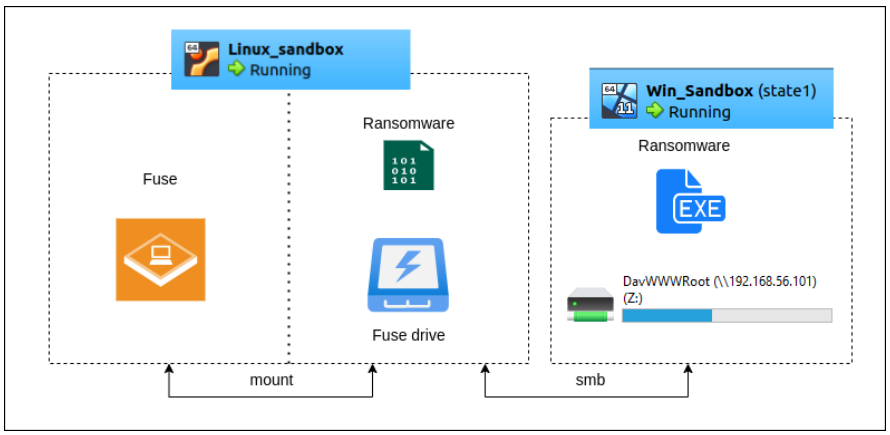

Рис. 1. Модель аналізу операцій в файловій системі

і т. д.. У результаті ентропія зашифрованого файлу, буде відрізнятись від ентропії, звичайного, не інфікованого файлу. Однак можуть існувати винятки у випадку з зображеннями або архівами, коли їх ентропія досягає високих значень. Вирішити цю проблему можна обчисливши різні значення ентропії притаманні для конкретного типу файлу та застосувавши ці значення при класифікації за допомогою машинного навчання.

Моніторинг операцій файлової системи

Для визначення активності зразків ШПЗ в файловій системі можна використовувати програмний інтерфейс *FUSE* [8]. Це програмний інтерфейс для *Unix* подібних операційних систем, що дозволяє не привілейованим користувачам створювати власну файлову систему без необхідності втручатись в код ядра. При створенні файлової системи на базі *FUSE* необхідно написати функції для всіх необхідних операції, що будуть виконуватись в файловій системі під час виконання досліджувальних зразків. Створені операції необхідно додати в структуру *fuse_ operations*.

Запустити інтерфейс можна за допомогою команди:

```
./fusefs /home/user/files \
/home/sandbox/sandbox
```

В даному випадку fusefs монтує директорію `/home/sandbox/sandbox` тестові файли розміщенні в директорії `/home/user/files`. Далі користувач `sandbox` запускає шкідливий зразок, що шифрує, перейменовує, створює та виконує інші зміни в файловій системі. Всі операції виконані шкідливим програмним забезпеченням, записуються в `/home/user/access.log`. Для дослідження поведінки програми-вимагача в Windows, необхідно також запустити `Webdav` сервер, за допомогою команди:

```
sudo wsgidav --host=0.0.0.0 --port=80
--root=/home/sandbox/sandbox
--auth=anonymous
```

або запустити `impacket-smbserver` за допомогою команди:

```
sudo impacket-smbserver share -smb2support
/tmp/smbshare
```

та створити мережевий диск в віртуальній машині Windows. Після час запуску шкідливого програмного зразка, він знаходить та шифрує файли у всіх доступних в системі дисках включаючи новостворений мережевий диск на базі директорії, що змонтувала програма fusefs. Відповідно дана активність також буде детально записана в `.log` файл. Загальна модель середовища для аналізу активності в файловій системі зображена на рис. 1. Приклад результатів отриманих з *.log* файлу представлений на табл. 1.

Отримані дані, можна обробляти та використовувати для класифікації ШПЗ методами машинного навчання. Більшість досліджених зразків виконують окремі записи по 4096 байти, в деяких випадках по 64 байти, та в кінці різними способами ту частину файлу що залишилась. Відповідно вдається сформувати певний шаблон активності злоякісної програми в створеній файловій системі. Всі зразки перейменовують зашифровані файли , залишаючи мітку у вигляді додавання назви ШПЗ до файлу. У випадку роботи програм архіваторів, також присутні послідовні записи однакової величини, але в той же час досить багато відмінних ознак, таких як значне зменшення розміру файлу та різні інформаційні записи у вигляді заголовків.

При виявлені шифрувальника необхідно відновити файли, які він вже встиг переписати. Для цього робиться знімок логічного тому `LVM`, компактна моментальна копія, що створюється і займає місце лише тоді коли в базовому логічному томі вносяться зміни. щоб створити викликається команда:

```
lvcreate --size 1G --snapshot --name
ubuntu_snap /dev/mapper/vgubuntu-root
```

Висновки

В даній роботі розглянуто використання динамічного аналізу для створення шаблонів роботи шифрувальника в файловій системі. Для визначе-

Таблиця 1. Аналіз шифрування тестового файлу за допомогою ШПЗ icefire

Operation name	Filename	Write bytes	Read bytes	Offset	New filename
Created file	/iFire-readme.txt	-	-	-	-
Writen	/iFire-readme.txt	1323	-	0	-
Read	/X6C9U8H2SW.jpg	-	30063	0	-
Writen	/X6C9U8H2SW.jpg	4096	-	0	-
Writen	/X6C9U8H2SW.jpg	4096	-	4096	-
Writen	/X6C9U8H2SW.jpg	4096	-	8192	-
Writen	/X6C9U8H2SW.jpg	4096	-	12288	-
Writen	/X6C9U8H2SW.jpg	4096	-	16384	-
Writen	/X6C9U8H2SW.jpg	4096	-	20480	-
Writen	/X6C9U8H2SW.jpg	1391	-	28672	-
Writen	/X6C9U8H2SW.jpg	520	-	30063	-
Renamed file	/X6C9U8H2SW.jpg	-	-	-	/X6C9U8H2SW.jpg.iFire

ння активності зразків ШПЗ в файловій системі запропоновано використовувати програмний інтерфейс FUSE, створивши віртуальну файлову систему. Отримані шаблони можна використовувати для класифікації ШПЗ методами штучного інтелекту. При виявлені інфікування файлів, дані відновлюються за допомогою завчасно сформованого знімку файлової системи. Проведені дослідження довели можливість використання запропонованих методів для виявлення підозрілої активності ШПЗ та відновлення даних у разі інфікування.

Перелік використаних джерел

1. *Hornetsecurity*. 2022 Ransomware Attacks survey. — 2022. — URL: https://www.hornetsecurity.com/en/knowledge-base/ransomware/ransomware-attacks-survey-2022/.
2. *Delamotte A.* IceFire Ransomware Returns | Now Targeting Linux Enterprise Networks. — 2023. — URL: https://www.sentinelone.com/labs/icefire-ransomware-returns-now-targeting-linux-enterprise-networks/.
3. *Alqahtani A.*, *Sheldon F. T.* A Survey of Crypto Ransomware Attack Detection Methodologies: An Evolving Outlook. — 2022.
4. Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions / U. Urooja, B. Ali, S. Al-rimy, A. Zainal, F. A. Ghaleb, M. A. Rassam. — 2021.
5. Avoiding Future Digital Extortion Through Robust Protection Against Ransomware Threats Using Deep Learning Based Adaptive Approaches / S. Sharmeen, Y. Abukar, A. Shamsul, M. Hdua, H. Mehedi. — 2019.
6. *Kyungroul L.*, *Sun-Young L.*, *Kangbin Y.* Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems. — 2019.
7. Enhancing File Entropy Analysis to Improve Machine Learning Detection Rate of Ransomware / H. Chia-Ming, Y. Chia-Cheng, C. Han-Hsuan, P. E. Setiasabda, L. Jenq-Shiou. — 2021.
8. FUSE (Filesystem in Userspace). — URL: https://github.com/libfuse/libfuse#readme.