ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Рекомендації щодо виявлення активності шкідливого програмного забезпечення Raccoon Stealer на кінцевих точках організації»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Владислав ДУДАРЕНКО
(підпис)

Виконав: здобувач вищої освіти групи БСД-44
ДУДАРЕНКО Владислав
(прізвище, ім'я)

Керівник к.військ.н., доцент ГАХОВ Сергій
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ЗМІСТ

ВСТУП

Актуальність дослідження. У сучасних умовах цифровізації діяльності організацій кінцеві точки, зокрема персональні комп'ютери, ноутбуки та інші пристрої, стали основними об'єктами атак з боку кіберзлочинців. Одним із поширених класів шкідливого програмного забезпечення є інфостілери, які спеціалізуються на викраденні облікових даних, фінансової інформації, даних з браузерів та криптогаманців. Особливу загрозу становить Raccoon Stealer – ШПЗ, що активно використовується в атаках та швидко еволюціонує завдяки моделі поширення Malware-as-a-Service.

Враховуючи високу частоту використання цього інфостілера в реальних атаках, виникає потреба у розробці ефективних підходів до виявлення його активності на кінцевих точках організацій. Традиційні антивірусні рішення не завжди здатні вчасно виявити загрозу, тому на перший план виходить використання інструментів цифрової криміналістики, таких як Velociraptor, Autopsy та Wireshark. Це зумовлює актуальність обраної теми дослідження та її практичну значущість у сфері забезпечення кібербезпеки.

Об'єкт дослідження. Процеси функціонування шкідливого програмного забезпечення у середовищі кінцевих точок організації.

Предмет дослідження. Активність шкідливого програмного забезпечення Raccoon Stealer, яку можна виявити за допомогою засобів цифрової криміналістики.

Мета роботи. Розробити підхід до виявлення активності Raccoon Stealer на кінцевій точці організації з використанням цифрових криміналістичних інструментів.

Наукові завдання:

дослідити принципи роботи інфостілера Raccoon Stealer.

проаналізувати сучасні підходи до виявлення активності ШПЗ.

ознайомитися з можливостями Velociraptor, Autopsy та Wireshark для аналізу кінцевих точок.

змоделювати інцидент зараження системи Raccoon Stealer.

виконати аналіз цифрових артефактів та мережевих слідів, які залишає інфостілер.

розробити практичні рекомендації з виявлення подібної активності.

Методи дослідження. Аналіз літературних джерел, вивчення технічної документації, експериментальне моделювання інцидентів, збір та аналіз цифрових артефактів, використання криміналістичних інструментів у лабораторних умовах.

Практичне значення одержаних результатів. Отримані результати можуть бути використані фахівцями з кібербезпеки для покращення процесів виявлення інфостілерів у корпоративних середовищах, формування внутрішніх процедур реагування на інциденти та навчання персоналу засобам цифрової криміналістики.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ВИЯВЛЕННЯ АКТИВНОСТІ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ RACCOON STEALER НА КІНЦЕВИХ ТОЧКАХ

1.1 Аналіз проблеми виявлення шкідливого програмного забезпечення Raccoon Stealer

Визначення інфостілера

Шкідливе програмне забезпечення типу «інфостілер» було створено для викрадення конфіденційних даних жертви[1]. Ціль викрадення зазвичай включає в себе конфіденційну інформацію, облікові записи жертви, фінансову інформацію, криптовалютні гаманці та інша персональна інформація.

Концепція Malware-as-a-Service (MaaS) передбачає використання шкідливого програмного забезпечення як сервіс. MaaS працює за аналогією з легальним програмним забезпеченням: розробники створюють загрози, продають або здають їх в оренду, а замовники використовують для атак. Така модель дозволяє особам без спеціальної технічної підготовки, які мають намір використовувати ШПЗ в протиправних цілях, отримувати доступ до потужних шкідливих інструментів без необхідності глибоких технічних знань.

Перші інфостілери з'явилися ще в 2000-х роках, однак їхнє масове використання значною мірою активізувалося після 2018 року. За даними звіту, згідно з IBM X-Force Threat Intelligence Index 2023, активність шкідливого програмного забезпечення для крадіжки інформації зросла на 266% у 2023 році порівняно з попереднім роком. Це свідчить про те, що зловмисники все більше використовують такі програми для отримання несанкціонованого доступу до облікових даних користувачів.

Основні компоненти інфостілерів

Інфостілери зазвичай складаються з наступних компонентів:

Фреймворк бота, який дозволяє зловмиснику налаштовувати поведінку ШПЗ відповідно до своїх потреб;

Панель управління, що виконує роль сервера для отриманнях викрадених даних[2].

Методи розповсюдження

Зазвичай зловмисники використовують наступі рішення для поширення інфостілерів:

Фішингові атаки (шкідливі листи, підроблені сайти) – зловмисники розсилають електронні листи, які імітують офіційні повідомлення від банків або популярних сервісів. Вкладені файли або посилання містять шкідливий код, що активує інфостілер;

Зараження веб-сайтів – кіберзлочинці інтегрують експлойти у звичайні веб-ресурси, що дозволяє непомітно завантажувати ШПЗ на пристрої відвідувачів;

Розповсюдження через піратське програмне забезпечення та моди для відеоігор.

Принцип дії інфостілер

Шкідливе програмне забезпечення інфостілер використовує різні методи для націлювання та вилучення певних типів даних із заражених систем. Кожен штам шкідливого програмного забезпечення, що класифікується як інфостілер, має різні можливості, починаючи від простих скриптів і закінчуючи складними модульними шкідливими програмами. Функціональність інфостілерів може змінюватися залежно від версії та розробника. Деякі з них орієнтовані на збір даних жертви, тоді як інші можуть надавати віддалений доступ до пристрою, що дозволяє в майбутньому запускати інше шкідливе програмне забезпечення.

Після успішного проникнення на пристрій жертви інфостілер активується та починає збір конфіденційних даних. Після збору цих даних ШПЗ передає їх на сервер зловмисника, де вони можуть бути використані для подальших атак, продажу на чорному ринку або шантажу жертви.

Основні методи крадіжки даних

Способи дії шкідливого програмного забезпечення інфостілер:

кейлоггер: одна з найпоширеніших тактик, кейлоггер включає запис натискань клавіш користувачем. Захоплюючи все, що друкується, зловмисники можуть пізніше відфільтрувати паролі, дані кредитних карток та іншу конфіденційну особисту інформацію;

захоплення форм: ця техніка використовується для перехоплення даних, надісланих у формах на веб-сторінках, до того як вони будуть зашифровані браузером. Вона особливо ефективна для крадіжки облікових даних для входу, платіжної інформації та інших даних, введених на веб-сайтах;

перехоплення буфера обміну: інфостілер може відстежувати та змінювати вміст буфера обміну на зараженому пристрої. Коли користувач копіює дані, наприклад номери рахунків або паролі, шкідлива програма замінює або краде цю інформацію. Ця техніка атаки може бути використана для несанкціонованого вилучення інформації незалежно від дій користувача, наприклад, імена користувачів та паролі;

захоплення екрана: роблячи знімки екрана користувача в критичні моменти, наприклад, при введенні облікових даних або перегляді особистої інформації, цей метод дозволяє обійти обмеження на отримання текстових даних, захоплюючи дані, що відображаються на екрані в будь-якій формі;

перехоплення сеансу браузера: цей метод передбачає крадіжку файлів cookie та токенів сеансу з кешованої пам'яті браузера, що дозволяє кіберзлочинцям видавати себе за онлайн-сеанс жертви та отримувати несанкціонований доступ до онлайн-акаунтів без введення імені користувача та пароля;

витяг облікових даних: цей метод витягує дані з облікових записів користувачів, що зберігаються в системі, наприклад, облікові дані для входу, збережені у веб-браузерах або в іншому клієнтському програмному забезпеченні. Якщо вони зберігаються у зашифрованому форматі, зловмисники спробують зламати їх в автономному режимі, використовуючи спеціалізовані апаратні та програмні інструменти;

атаки «Man-in-the-Browser»: це складніші атаки, при яких шкідливе програмне забезпечення впроваджує шкідливий код у сам веб-браузер. Це дозволяє

зловмиснику перехоплювати та маніпулювати інформацією у режимі реального часу, коли вона вводиться на захищених веб-сайтах;

збір адрес електронної пошти: шкідлива програма переглядає файли та електронні листи, що зберігаються на комп'ютері, для збирання адрес електронної пошти та іншої контактної інформації, яка може бути використана для розсилки спаму або подальших атак фішингу;

збір криптогаманців: деякі шкідливі програми інфостілери можуть шукати відомі шляхи встановлення для поширеного програмного забезпечення криптогаманців та намагатися вкрасти закриті ключі. Потрапивши у володіння зловмисника, ці ключі можуть бути використані для переведення криптовалюти жертви на рахунки, що контролюються зловмисником.

Наслідки роботи інфостілерів

Викрадення конфіденційних даних, облікових записів жертви, фінансової інформації, криптовалютних гаманців та іншої персональної інформації за допомогою інфостілерів може призвести до серйозних наслідків. Наприклад:

фінансові втрати – якщо зловмисники отримують доступ до банківських акаунтів або криптовалютних гаманців, вони можуть здійснювати несанкціоновані транзакції;

компрометація особистих акаунтів – викрадені паролі можуть бути використані для входу в соціальні мережі, електронну пошту або робочі системи;

розповсюдження загроз – зловмисники можуть використовувати скомпрометовані облікові записи для подальшого поширення ШПЗ серед контактів жертви.

Інфостілер Raccoon Stealer

Raccoon Stealer – ШПЗ, призначене для крадіжки інформації, вперше опубліковане у квітні 2019 року на російськомовних форумах. Raccoon Stealer здобув широке поширення серед кіберзлочинців завдяки своїй гнучкості та моделі розповсюдження MaaS.

Його можна отримати на форумах кіберзлочинців: підписка коштує 200 доларів на місяць. Raccoon Stealer уже заразив понад 100 000 пристроїв, серед

організацій і окремих осіб, і став однією з найбільш згадуваних атак на підпільних форумах[3].

Еволюція Raccoon Stealer

Випущені версії та основні зміни:

початкова версія (2019 рік): Написана на мові C++. Ця версія пропонувала базові функції крадіжки даних з браузерів та криптовалютних гаманців;

Рис.1.1 Перша згадка Raccoon Stealer на російськомовному форумі «Hack Forums» 13 квітня 2019 р.

Raccoon Stealer v2 (червень 2022 рік): Після тимчасового припинення діяльності у березні 2022 року, зловмисники випустили другу версію, переписану на мові C. Ця версія мала покращену функціональність та оновлену інфраструктуру;

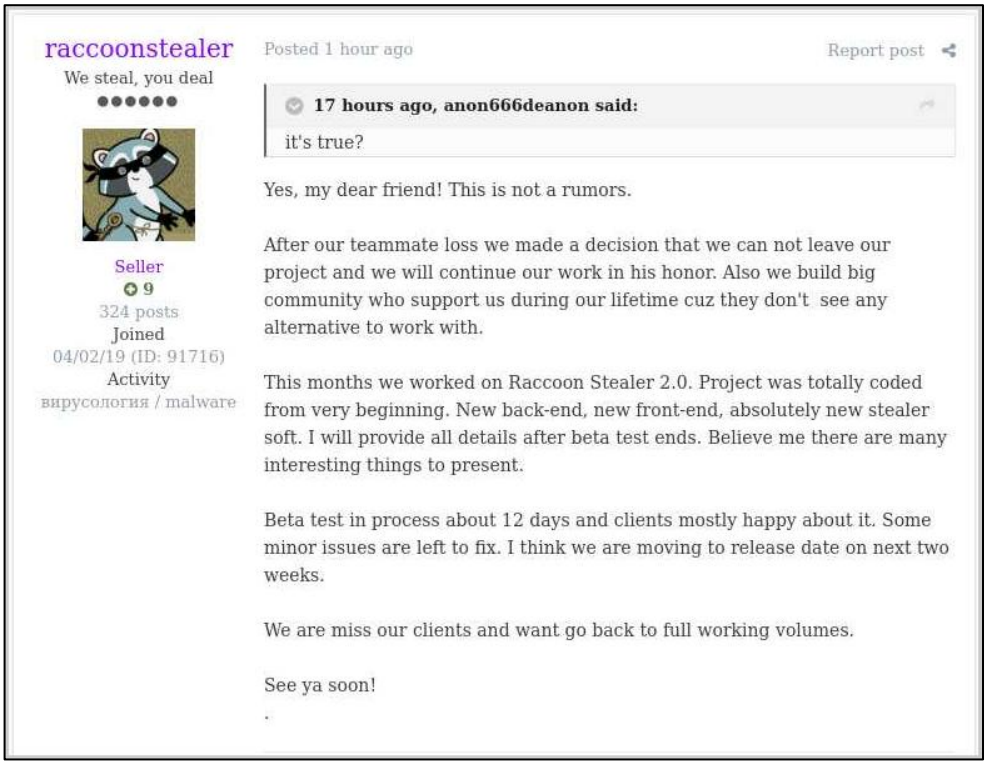


Рис.1.2 Анонс нової версії Raccoon Stealer v2

Raccoon Stealer v2.3.0 (серпень 2023): Після шестимісячної перерви розробники представили оновлену версію 2.3.0 з новими функціями, включаючи ефективний механізм пошуку файлів cookie та паролів, а також покращену панель адміністратора.

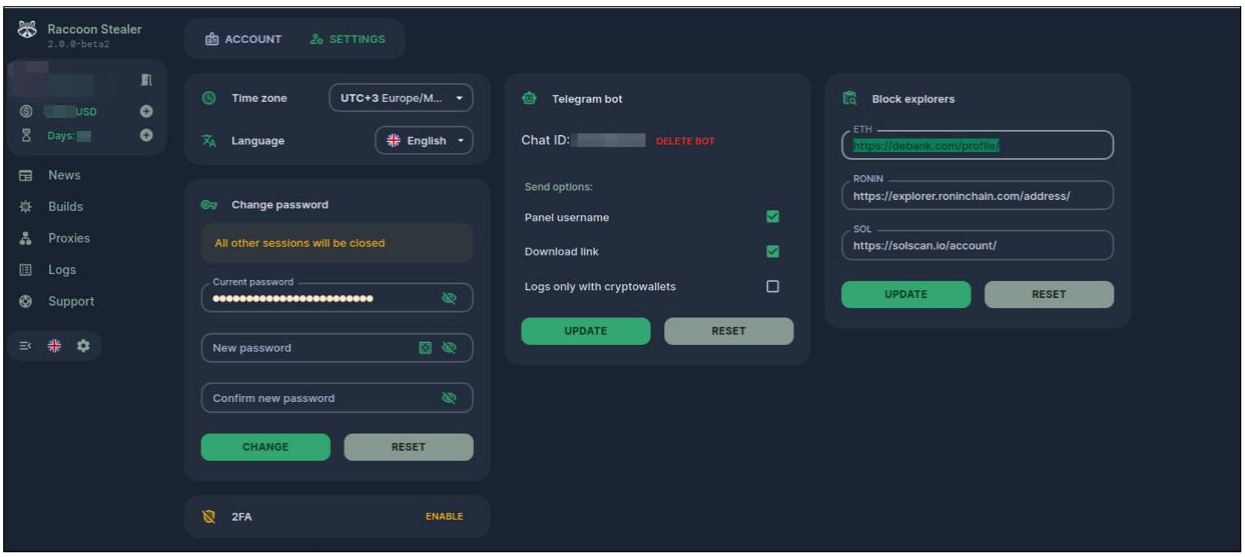


Рис.1.3 Панель адміністратора Raccoon Stealer v2.3.0

1.2 Аналіз підходів до виявлення активностей шкідливого програмного забезпечення

Виявлення активностей ШПЗ є однією з основних задач цифрової криміналістики та інформаційної безпеки. Ефективне виявлення шкідливої активності дозволяє мінімізувати наслідки інцидентів, зберегти цілісність систем і попередити витік конфіденційних даних.

Класифікація підходів до виявлення активностей ШПЗ

У практиці інформаційної безпеки існують кілька основних методів виявлення активностей шкідливого програмного забезпечення. Найбільш поширеними є сигнатурний, евристичний, поведінковий підходи, а також методи на основі машинного навчання.

Сигнатурний підхід

Сигнатурний підхід базується на порівнянні файлів або процесів із відомими шаблонами – сигнатурами шкідливого коду. Ці сигнатури зазвичай містять унікальні фрагменти коду, характерні для конкретних зразків ШПЗ. Перевагою цього методу є його висока точність при роботі з відомими загрозами [4]. Проте основним недоліком сигнатурного підходу є нездатність виявляти нові або модифіковані зразки ШПЗ, які не мають відповідних сигнатур у базі даних [5].

Евристичний підхід

Евристичний аналіз використовує набори правил для виявлення потенційно шкідливих програм шляхом аналізу їхньої структури або поведінки без необхідності точного збігу з відомою сигнатурою. Він дозволяє виявляти модифіковані або невідомі загрози. Однак через загальність правил часто виникають хибно позитивні спрацювання [6].

Поведінковий підхід

Поведінковий аналіз базується на моніторингу активності програм у режимі реального часу. Система відстежує такі дії, як створення файлів, зміни реєстру, встановлення мережевих з'єднань, та аналізує їх на предмет аномальної активності.

Цей підхід є особливо ефективним для виявлення складних атак, однак потребує значних обчислювальних ресурсів та якісної бази поведінкових шаблонів.

Методи на основі машинного навчання

У сучасних рішеннях дедалі частіше використовуються моделі машинного навчання для виявлення аномальної активності. Такі моделі аналізують великі обсяги даних для виявлення зразків, притаманних шкідливій поведінці, що дозволяє ефективно виявляти як відомі, так і невідомі загрози. Проте ефективність цих моделей залежить від якості даних для навчання та правильності вибору ознак.

Інструменти для виявлення активностей ШПЗ

Для реалізації згаданих підходів використовуються різноманітні інструменти цифрової криміналістики та інформаційної безпеки. Серед них можна виділити системи EDR (Endpoint Detection and Response), SIEM (Security Information and Event Management) рішення, а також спеціалізовані інструменти аналізу артефактів.

Системи EDR

Системи EDR забезпечують постійний моніторинг та збір даних з кінцевих точок, дозволяючи виявляти підозрілу активність у режимі реального часу. Вони поєднують поведінковий аналіз із можливістю проведення аналізу подій, що вже відбулися, з метою розслідування інцидентів безпеки.

SIEM рішення

Рішення класу SIEM збирають та аналізують журнали подій з різних систем і мережевих пристроїв. Вони застосовують кореляційні правила для виявлення індикаторів компрометації і використовуються переважно для виявлення загроз на рівні всієї організації.

Спеціалізовані інструменти аналізу

Інструменти, такі як Velociraptor та Autopsy, дозволяють проводити глибокий аналіз системних артефактів, визначаючи можливі ознаки присутності ШПЗ. Їх використання особливо актуальне у процесах криміналістичного розслідування інцидентів [7].

Порівняльний аналіз підходів

Таблиця 1.1.

Порівняння підходів виявлення загроз

Підхід	Переваги	Недоліки
Сигнатурний підхід	Швидке та точне виявлення відомих загроз	Неефективність проти нових або модифікованих загроз
Евристичній підхід	Можливість виявлення нових або невідомих загроз	Високий ризик хибнопозитивних спрацювань
Поведінковий підхід	Виявлення невідомих загроз через аналіз аномальної активності	Потреба у складних алгоритмах та високі вимоги до обчислювальних ресурсів
Методи машинного навчання	Висока гнучкість, здатність адаптуватися до нових загроз	Потреба у великих обсягах якісних даних та значних ресурсах для навчання
Гібридні підходи	Поєднання переваг різних методів для підвищення ефективності виявлення	Висока складність впровадження та підтримки системи

Кожен із розглянутих підходів має свої переваги і недоліки залежно від контексту використання.

Сигнатурний підхід забезпечує швидке виявлення відомих загроз, але неефективний проти нових атак.

Евристичний підхід підвищує вірогідність виявлення нових загроз, однак має ризик помилкових спрацювань.

Поведінковий підхід дозволяє фіксувати невідомі загрози, але потребує складних алгоритмів аналізу.

Методи машинного навчання забезпечують високу гнучкість і адаптивність, однак вимагають великих обсягів якісних даних і ресурсів для навчання моделей.

У практиці інформаційної безпеки часто використовуються гібридні підходи, які поєднують переваги кількох методів для досягнення максимальної ефективності.

Гібридні методи виявлення ШПЗ

Одним із сучасних напрямів розвитку технологій виявлення активностей ШПЗ є застосування гібридних підходів. Гібридний метод передбачає інтеграцію кількох способів аналізу – наприклад, сигнатурного, поведінкового та машинного навчання одночасно. Таке поєднання дозволяє компенсувати недоліки кожного окремого підходу та забезпечити комплексний захист.

Гібридні системи виявлення здатні спочатку швидко ідентифікувати відомі загрози за допомогою сигнатурного аналізу, а потім застосувати поведінкові механізми до невідомих процесів або застосувати алгоритми аномального виявлення для більш глибокої оцінки підозрілих активностей. Прикладом такої інтеграції є сучасні системи класу XDR (Extended Detection and Response), що об'єднують дані з різних джерел для комплексного аналізу.

Завдяки гібридним методам виявлення стало можливим досягати високої точності при одночасному зменшенні кількості хибних спрацьовувань, що суттєво полегшує роботу фахівців із безпеки.

Поточний стан розвитку методів виявлення ШПЗ

У зв'язку з постійною еволюцією шкідливого програмного забезпечення виникає необхідність у розвитку більш адаптивних і розумних механізмів виявлення. Серед основних тенденцій можна виділити кілька напрямів:

застосування глибокого навчання;

використання Threat Intelligence;

застосування технологій безперервного моніторингу;

автоматизація аналізу інцидентів.

Глибоке навчання

Традиційні методи машинного навчання все частіше доповнюються технологіями глибокого навчання. Нейронні мережі можуть автоматично виділяти релевантні ознаки з великих обсягів сирих даних і самостійно навчатися

розпізнавати складні шаблони шкідливої активності [8]. Це відкриває нові можливості для виявлення складних і раніше невідомих атак.

Threat Intelligence

Інтеграція даних про загрози (Threat Intelligence) у процес виявлення дозволяє оперативно отримувати інформацію про нові види ШПЗ, індикатори компрометації (IoC), тактики та техніки атак [9]. Це забезпечує більш активний підхід до виявлення загроз.

Технології безперервного моніторингу

У сучасних системах акцент робиться на безперервний моніторинг подій без прив'язки до певних часових вікон. Це дозволяє швидше виявляти підозрілу активність, знижуючи час виявлення інциденту (MTTD, Mean Time to Detect).

Автоматизація аналізу інцидентів

Ще одним важливим трендом є автоматизація реагування на інциденти за допомогою технологій SOAR (Security Orchestration, Automation, and Response).

1.3 Аналіз методів та засобів виявлення активностей шкідливого програмного забезпечення

У сучасних умовах зростання кількості та складності кіберзагроз виявлення активності ШПЗ на кінцевих точках і в мережах є критично важливим завданням для забезпечення інформаційної безпеки організацій. У цьому розділі розглянуто три популярні інструменти: Velociraptor, Autopsy та Wireshark, які ефективно застосовуються для аналізу й виявлення слідів діяльності шкідливих програм.

Загальні підходи до виявлення активностей шкідливого програмного забезпечення

Процес виявлення активностей ШПЗ базується на аналізі артефактів операційної системи, мережевого трафіку, змін у файловій системі та реєстрі, а також поведінкових характеристик програм. Традиційно використовуються дві основні стратегії: сигнатурний аналіз, що передбачає пошук відомих шаблонів

шкідливої активності, та поведінковий аналіз, який виявляє нетипову або підозрілу поведінку процесів [10].

Однак через швидку еволюцію загроз класичні антивірусні рішення часто є недостатніми, тому необхідне застосування спеціалізованих інструментів цифрової криміналістики та аналізу інцидентів, таких як Velociraptor, Autopsy і Wireshark.

Особливості роботи Velociraptor

Velociraptor – це інструмент для збору, моніторингу та аналізу артефактів на кінцевих точках, орієнтований на цифрову криміналістику та реагування на інциденти [11]. Його основна перевага полягає у здатності швидко масштабуватися для аналізу великої кількості пристроїв одночасно.

Velociraptor дозволяє створювати користувацькі запити за допомогою власної мови запитів VQL (Velociraptor Query Language), що забезпечує гнучкість у зборі даних про системну активність. Це включає аналіз журналів подій, реєстру Windows, процесів, з'єднань мережі та файлової системи.

Використання Velociraptor для виявлення ШПЗ

За допомогою Velociraptor можна здійснювати:

пошук слідів запуску підозрілих програм через аналіз автозавантаження та артефактів реєстру Windows [12];

виявлення аномальних мережевих підключень або нестандартних процесів;

ідентифікацію змін у критичних директоріях або системних файлах.

Особливістю Velociraptor є можливість оперативного реагування - оператор може не лише збирати дані, але й запускати скрипти або команди для ізоляції потенційно скомпрометованого пристрою.

Загальна характеристика Autopsy

Autopsy – це модульна платформа з відкритим кодом для цифрового аналізу носіїв інформації [13]. Вона надає зручний графічний інтерфейс для роботи з різними видами артефактів, що є важливою перевагою для криміналістичного аналізу даних.

Autopsy дозволяє автоматично аналізувати файлові системи, відновлювати видалені файли, знаходити артефакти діяльності користувача (наприклад, історію

браузера, електронну пошту, чати), а також шукати в системі потенційні індикатори компрометації.

Застосування Autopsy у виявленні шкідливої активності

Інструмент широко використовується для виявлення:

залишкових слідів діяльності ШПЗ у файловій системі;

модифікацій або видалень файлів, які можуть свідчити про діяльність шкідливого програмного забезпечення;

активності користувачів, що могла бути зумовлена впливом шкідливих програм [14].

Завдяки розширенням і плагінам Autopsy може проводити автоматичний пошук шкідливих елементів, що значно спрощує аналіз великих обсягів даних.

Огляд можливостей Wireshark

Wireshark є одним із найефективніших і найпопулярніших інструментів для аналізу мережевого трафіку. Він дозволяє захоплювати й детально аналізувати пакети даних, що проходять через мережу, ідентифікувати аномалії та шкідливу активність.

Wireshark підтримує глибоку інспекцію сотень мережевих протоколів і дає змогу виконувати фільтрацію даних за різними критеріями, що надзвичайно важливо для розслідування інцидентів безпеки.

Використання Wireshark у виявленні шкідливого трафіку

Wireshark дає можливість виявити:

ознаки командно-контрольного трафіку між зараженими машинами та серверами управління (C2) [15];

експлуатацію вразливостей через аналіз незашифрованих пакетів;

аномальні запити до зовнішніх ресурсів, характерні для активності ботнетів або крадіжки даних.

Інструмент особливо ефективний при виявленні загроз, що здійснюють активне мережеве спілкування, таких як Raccoon Stealer, оскільки характерний мережевий трафік ШПЗ може бути виявлений через нетипову активність DNS або HTTP-запитів [16].

Особливості застосування Velociraptor для розслідування інцидентів

Velociraptor дозволяє будувати детальний профіль активності системи за допомогою збору великого обсягу артефактів. У контексті розслідування діяльності ШПЗ це надає можливість виявити наступні ознаки компрометації:

наявність невідомих або підозрілих запущених процесів;

модифікації системних параметрів автозавантаження;

відхилення в стандартних записах журналів подій Windows.

Завдяки можливостям розподіленого збору даних Velociraptor стає незамінним під час розслідування атак на великі корпоративні мережі. Крім того, використання VQL дозволяє створювати складні запити, наприклад, для виявлення артефактів запуску відомих шкідливих програм, таких як Raccoon Stealer.

Серед основних переваг Velociraptor також слід відзначити високу швидкість обробки запитів, підтримку сценаріїв для автоматизації аналізу та можливість гнучкої конфігурації серверної частини рішення.

Поглиблений аналіз з Autopsy

Autopsy, завдяки своїй модульній архітектурі, надає широкі можливості для глибокого дослідження слідів діяльності шкідливого програмного забезпечення.

Основні можливості платформи включають:

автоматичне індексування вмісту дисків і створення карти активності файлів;

виявлення ознак компрометації через пошук аномалій у метаданих файлів;

відновлення файлів, які були навмисно видалені зловмисником для приховання слідів.

Особливо важливою функцією є можливість інтеграції Autopsy із зовнішніми базами даних шкідливих хешів, що дозволяє швидко виявляти вже відомі зразки ШПЗ. У випадках розслідування інцидентів, пов'язаних із діями ШПЗ, Autopsy допомагає не лише встановити факт зараження, а й простежити весь ланцюг подій, що спричиняли активації ШПЗ.

Аналіз мережевого трафіку за допомогою Wireshark

У випадку зараження системи шкідливим програмним забезпеченням важливу роль у розслідуванні відіграє аналіз мережевої активності. Wireshark надає

фахівцям можливість ідентифікувати специфічні сигнали діяльності ШПЗ у мережі, зокрема:

встановлення небажаних з'єднань із зовнішніми IP-адресами;

передачу конфіденційних даних через незахищені протоколи;

використання нестандартних портів або тунелювання трафіку [17].

За допомогою застосування фільтрів відображення у Wireshark можна оперативно виявляти підозрілу активність. Наприклад, при дослідженні активності Raccoon Stealer можливо ідентифікувати мережеві запити, які передають викрадені облікові дані на сервери керування. Завдяки можливості збереження та аналізу файлів захоплення трафіку (PCAP) Wireshark також використовується для ретроспективного аналізу інцидентів.

Значення комплексного підходу до виявлення активностей ШПЗ

Ефективне виявлення активності ШПЗ неможливе без поєднання декількох підходів і засобів аналізу. Використання Velociraptor, Autopsy та Wireshark дозволяє:

охопити як локальні зміни на кінцевій точці (Velociraptor і Autopsy), так і мережеву активність (Wireshark);

детально дослідити контекст подій, що призвели до інциденту;

швидко ідентифікувати джерело компрометації та масштаби ураження.

Інтеграція даних з різних інструментів дає змогу формувати повну картину інциденту й оперативно вживати заходів для ліквідації наслідків атаки.

Висновок

Розгляд Velociraptor, Autopsy та Wireshark як засобів виявлення активності шкідливого програмного забезпечення показав, що кожен із них має свої унікальні можливості та застосовується на різних етапах розслідування інцидентів. Velociraptor забезпечує оперативний збір артефактів із кінцевих точок, Autopsy – поглиблений криміналістичний аналіз даних, а Wireshark – детальне вивчення мережевого трафіку. Комплексне використання цих інструментів підвищує ефективність виявлення і нейтралізації загроз у сучасному кіберпросторі.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Архітектура та компоненти рішення Velociraptor

Velociraptor – це сучасне рішення з відкритим кодом для цифрової криміналістики та реагування на інциденти безпеки . Його головна мета – забезпечити швидкий та ефективний доступ до критичних даних з кінцевих точок організаційної інфраструктури. Завдяки своїй архітектурі, заснованій на моделі "клієнт-сервер", гнучкому механізму запитів на мові VQL (Velociraptor Query Language) та модульній структурі, інструмент набув широкого визнання серед фахівців з кібербезпеки.

Загальний огляд архітектури

Архітектура Velociraptor побудована з урахуванням масштабованості, надійності та мінімального навантаження на кінцеві системи. Вона складається з кількох ключових елементів: серверного компонента , клієнтів, бази даних, веб-інтерфейсу керування та VQL.

Сервер виступає центральним координатором системи: він керує клієнтами, розсилає команди, приймає зібрані артефакти та забезпечує безпечну комунікацію. Клієнти – це невеликі агенти, які встановлюються на кінцеві точки (Windows, Linux або macOS) і виконують запити, отримані з сервера. Передача даних від клієнтів до сервера відбувається асинхронно через протокол HTTP/S, що дозволяє працювати за умов обмеженого чи нестабільного мережевого з'єднання.

Окрему увагу приділено безпеці – взаємодія клієнта з сервером зашифрована, використовуються TLS-сертифікати, а також механізми автентифікації та авторизації. Це особливо важливо при роботі в корпоративних середовищах [18].

Ключові компоненти системи

Серверний компонент – це головний компонент, який відповідає за управління інфраструктурою Velociraptor. Він отримує запити від веб-інтерфейсу, надсилає інструкції клієнтам, збирає результати та зберігає їх у сховищі даних. Сервер може бути запущений як у режимі одиночного вузла, так і в кластерному режимі з використанням балансувальників навантаження для високої доступності.

У великих організаціях для масштабування можливо створення кількох екземплярів, які обробляють паралельні запити та запобігають перевантаженню.

Клієнт Velociraptor – це невеликий двійковий файл (менше 10 МБ), який працює на цільовій машині та виконує запити, надіслані з сервера. Клієнт підтримує два основних режими: інтерактивний та автономний. У першому режимі він активно обмінюється інформацією з сервером у режимі реального часу. У другому – виконує заздалегідь визначені сценарії без постійного з'єднання [19].

Це дозволяє використовувати Velociraptor у різноманітних середовищах, включно з віддаленими системами або пристроями з обмеженим підключенням.
Усі результати роботи клієнта та метадані зберігаються в централізованому сховищі. У базовій конфігурації використовується SQLite, однак для великих організацій можливе застосування Redis або власного формату Velociraptor DataStore. Структура зберігання підтримує журнали дій, історію змін, метаінформацію та кеш результатів [20].

Для взаємодії з Velociraptor використовується зручний веб-інтерфейс, який дозволяє проводити пошук, запускати VQL-запити, переглядати результати, завантажувати артефакти та створювати нові шаблони артефактів. Інтерфейс також забезпечує багаторівневий контроль доступу та логування всіх дій аналітиків, що підвищує прозорість та відповідність вимогам безпеки.

VQL – це декларативна мова запитів, створена спеціально для роботи з артефактами операційних систем. Вона дозволяє ефективно описувати умови збору інформації, фільтрувати результати та будувати складні аналітичні сценарії. Наприклад, за допомогою VQL можна швидко отримати всі запуски програм з

директорії Temp, виявити підозрілі ключі автозапуску або проаналізувати таймстомпи (від англ. timestomping).

Кожен VQL-запит реалізує один або кілька артефактів, які зберігаються у репозиторії Velociraptor і можуть повторно використовуватися або адаптуватися до конкретних кейсів.

Особливості масштабування та розгортання

Однією з ключових переваг Velociraptor є його гнучкість у розгортанні. У невеликих середовищах достатньо одного сервера та кількох клієнтів. У великих інфраструктурах система легко масштабується горизонтально – додаються нові серверні компоненти, застосовується розподілене зберігання та створюються незалежні середовища для різних відділів компанії.

Інструмент підтримує розгортання за допомогою Docker-контейнерів, скриптів автоматизації або вручну.

Переваги та обмеження

Velociraptor має низку переваг:

відкритий код та активна спільнота розробників;

підтримка всіх популярних ОС;

низьке споживання ресурсів;

висока гнучкість за рахунок VQL;

простота розгортання і масштабування [19].

Робота з артефактами

Однією з центральних концепцій у Velociraptor є артефакти – заздалегідь визначені шаблони збору даних, написані на мові VQL. Кожен артефакт представляє собою набір інструкцій, який визначає, які саме дані слід зібрати, за яких умов і як їх обробити. Наприклад, можна створити артефакт, який збирає інформацію про процеси, які запущені з нестандартних директорій, або про всі зміни у реєстрі Windows, що стосуються автозапуску програм [19].

Velociraptor постачається з великою кількістю вбудованих артефактів, однак користувачі можуть створювати власні – для цього використовується зрозумілий YAML-синтаксис. Усі артефакти організовані в бібліотеки, які можна імпортувати,

змінювати та ділитися ними з іншими аналітиками. Це значно підвищує ефективність повторного використання знань у великих командах безпеки.

Крім того, Velociraptor дозволяє збирати артефакти як у реальному часі (наприклад, коли користувач запускає підозрілу програму), так і в ретроспективі – наприклад, шляхом аналізу системних журналів або файлів Prefetch, які вказують на історію запусків програм.

Приклади практичного використання

Velociraptor активно застосовується у розслідування інцидентів ШПЗ. Наприклад, при виявленні зараження системи шкідливим програмним забезпеченням, таким як Raccoon Stealer, Velociraptor дозволяє швидко ідентифікувати відповідні артефакти: шляхи запуску, мережеву активність, ключі реєстру, сліди в AppData тощо [20]. Завдяки підтримці VQL можна створити спеціальні запити, які будуть шукати шаблони активності, притаманні цій родині ШПЗ.

Аудит безпеки кінцевих точок. За допомогою систематичного збору інформації про конфігурацію ОС, активні служби, відкриті порти, історію команд у PowerShell тощо, можна оцінити поточний стан захищеності систем і виявити слабкі місця.

Пошук IoC.

Після отримання індикаторів компрометації, наприклад, від сторонньої Threat Intelligence-платформи, аналітики можуть швидко перевірити наявність цих IoC в інфраструктурі шляхом масового запуску запитів на всіх підключених клієнтах.

Періодичний моніторинг.

За допомогою планувальника можна налаштувати періодичне виконання певних запитів. Наприклад, щоденний збір списку запущених процесів або підключених USB-пристроїв.

Інтеграція з іншими системами

Хоча Velociraptor є автономним інструментом, його можна інтегрувати з іншими системами для побудови повноцінного середовища реагування на інциденти. Серед можливих напрямків інтеграції:

SIEM-системи – для централізованого зберігання та аналізу зібраних логів.

Системи керування конфігурацією – для масового розгортання агентів.

Рішення SOAR – для автоматизації інцидент-реагування на основі подій, отриманих з Velociraptor.

Крім того, завдяки REST API, Velociraptor підтримує взаємодію з зовнішніми скриптами та інструментами, що дозволяє організувати повноцінний цикл "збір даних → аналіз → реакція" в автоматизованому режимі [18].

Порівняння з іншими DFIR-рішеннями

На ринку існує низка альтернатив Velociraptor, таких як GRR Rapid Response, Osquery, KAPE та інші. У порівнянні з ними Velociraptor має низку унікальних переваг:

більша гнучкість завдяки мові VQL;

зручний веб-інтерфейс із підтримкою шаблонів;

активна спільнота та регулярні оновлення;

краща інтеграція у сценарії DFIR, а не лише інвентаризації.

2.2 Структура та складові елементи рішення Autopsy

Autopsy – це сучасне рішення з відкритим кодом для цифрової криміналістики, яке дозволяє проводити глибокий аналіз цифрових носіїв, здійснювати витяг артефактів користувача та виявляти потенційні докази у справах, пов’язаних із кіберзлочинністю. Він розроблений як графічний інтерфейс до платформи The Sleuth Kit (TSK), яка забезпечує низькорівневу обробку файлових систем. Autopsy забезпечує ефективну взаємодію аналітика з даними, які зберігаються на файлових системах різних типів [21].

Інтерфейс користувача

Графічний інтерфейс користувача Autopsy є одним із найважливіших його елементів, оскільки він забезпечує зручну взаємодію з великою кількістю артефактів, які можуть бути виявлені під час розслідування. Інтерфейс базується на Java і реалізований із використанням NetBeans Platform, що забезпечує стабільність та інтеграцію з іншими модулями системи. Основні панелі інтерфейсу включають дерево структури справи, вікно попереднього перегляду, панель пошуку, журнал активності, а також вкладки для доступу до конкретних артефактів (наприклад, веб-браузерів, електронної пошти, історії чату тощо).

Завдяки інтеграції із системою TSK, користувачі мають змогу досліджувати структуру файлової системи, переглядати вміст видалених або прихованих файлів, а також ідентифікувати системні артефакти операційної системи Windows. Така взаємодія значно полегшує процес навігації у великому обсязі цифрових доказів і дозволяє швидко концентруватися на важливих для розслідування об'єктах.

Механізм збору та обробки даних

Autopsy підтримує аналіз як повних образів дисків, так і окремих логічних томів або директорій. Завдяки гнучкій архітектурі, він здатен працювати з різними типами джерел, включаючи фізичні накопичувачі, файли формату E01, AFF та RAW. Під час імпорту джерела даних, система автоматично ініціює базові процедури обробки:

обчислення хешів;

побудову файлової системи;

виділення метаданих;

ідентифікацію артефактів.

Окрему увагу варто приділити процесу обробки даних, який відбувається на основі заздалегідь визначених конфігурацій. Autopsy виконує серію автоматизованих кроків, таких як:

індексація вмісту;

виявлення типів файлів за сигнатурами;

аналіз тимчасових файлів;

аналіз реєстру Windows;

збір інформації про використання програмного забезпечення та підключення пристроїв.

Модулі аналізу

Одна з ключових переваг Autopsy – це його модульна структура, що забезпечує широку гнучкість у підходах до аналізу. Існує велика кількість вбудованих модулів, які виконують спеціалізовані завдання:

аналіз веб-історії;

виявлення ключових слів;

екстракція метаданих із мультимедійних файлів;

обробка електронної пошти;

виявлення ознак ШПЗ.

Крім того, Autopsy підтримує інтеграцію з зовнішніми інструментами. Наприклад, модуль PhotoDNA дозволяє проводити ідентифікацію зображень за хешами, а інтеграція з Python або Java-скриптами дає змогу створювати власні модулі аналізу відповідно до специфіки справи. Написання користувацьких модулів значно спрощується завдяки документації та SDK, що надається розробниками Autopsy.

Також важливим компонентом є модулі розвідки відкритих джерел (OSINT), які дозволяють, наприклад, автоматично перевіряти виявлені домени, IP-адреси або електронні адреси через відомі бази даних, що особливо корисно в розслідуваннях пов’язаних з фішинговими кампаніями або шкідливими ботнетами.

База даних кейсу

Autopsy використовує реляційну базу даних (SQLite або PostgreSQL) для збереження всієї інформації про кейс, включаючи хронологію дій, артефакти, результати обробки, хеші та інші метадані. Така структура дозволяє ефективно організувати дані, забезпечити швидкий доступ до потрібних елементів та підтримувати багатокористувацький доступ до кейсу при розслідуванні в команді [22].

Кожна справа в Autopsy – це окремий набір структур, у межах яких здійснюється організація джерел даних, створення міток, зв’язування артефактів та

фіксація всіх етапів аналізу. Такий підхід дозволяє зберігати цілісність розслідування, а також надає можливість повернутися до будь-якого з етапів на пізнішій стадії роботи.

Система звітності та журналювання

Автоматизована генерація звітів – ще один із важливих елементів платформи Autopsy. Система дозволяє створювати звіти у форматах HTML, CSV, Excel та PDF. У процесі створення звіту користувач може вибрати конкретні модулі або типи артефактів, які мають бути включені у фінальний документ. Це дає змогу формувати як стислий огляд, так і розгорнутий технічний аналіз.

Журналювання змін та дій аналітика виконується автоматично. Autopsy зберігає всі кроки обробки, що важливо для забезпечення принципу повторюваності розслідування, прийнятого в цифровій криміналістиці. Також підтримується збереження «hash set» та «known bad hash set» списків, що дозволяє прискорити процес повторного аналізу або ідентифікації відомих шкідливих об’єктів у майбутніх кейсах.

Інтеграція з іншими інструментами та платформами

Однією з ключових переваг Autopsy є його здатність до інтеграції з іншими інструментами цифрової криміналістики. Це дозволяє спеціалістам з інформаційної безпеки будувати гнучкі та масштабовані аналітичні середовища. Наприклад, завдяки вбудованій підтримці The Sleuth Kit, Autopsy має можливість глибокого аналізу файлових систем, включаючи FAT, NTFS, exFAT, HFS+, UFS та EXT [1]. Ця взаємодія забезпечує доступ до низькорівневих структур, що може мати вирішальне значення при відновленні видалених або прихованих даних.

Крім того, Autopsy підтримує використання зовнішніх модулів і фреймворків, таких як Plaso (log2timeline), який дозволяє побудову хронології подій на основі журналів системи та артефактів. Інтеграція з Volatility забезпечує можливість аналізу дампів оперативної пам’яті, що розширює межі традиційного аналізу дисків і дозволяє отримати динамічні докази, пов’язані з активністю програм, мережевими з’єднаннями та завантаженими бібліотеками.

Підтримка роботи в команді та масштабованість

Autopsy підтримує спільну роботу кількох аналітиків над однією справою завдяки можливості централізованого зберігання інформації та віддаленого підключення. За допомогою функціональності «Multi-User Case» можна організувати роботу цілої команди спеціалістів, які мають доступ до єдиної бази даних кейсу через мережу. Це особливо важливо у великих розслідуваннях, де потрібно розподілити обсяг аналізу між кількома учасниками або залучити вузькопрофільних експертів на певних етапах.

Масштабованість Autopsy реалізується також через можливість використання потужних бекенд-серверів, розширення сховищ та адаптацію до великих обсягів даних. У разі роботи з десятками гігабайтів або терабайтами цифрової інформації, система здатна ефективно обробляти масиви даних без втрати продуктивності завдяки використанню оптимізованої індексації та асинхронного аналізу.

Безпека, аудит та збереження цілісності доказів

Одним з основоположних принципів цифрової криміналістики є забезпечення цілісності доказової бази. Autopsy дотримується цього принципу шляхом автоматичного обчислення хешів (MD5, SHA1, SHA256) під час імпорту кожного джерела даних. Результати хешування зберігаються у звіті кейсу, що дозволяє пізніше підтвердити, що жодна частина доказу не була змінена в процесі аналізу.

Окрім того, Autopsy веде докладний аудит усіх дій користувача. Журнали активності містять інформацію про час, модуль, тип дії та об’єкт, що був оброблений. Цей підхід забезпечує повну прозорість процесу розслідування та дозволяє відтворити всі кроки аналітика у судовому процесі або при зовнішній експертизі.

Безпека даних справи також забезпечується на рівні доступу: Autopsy може бути налаштований так, щоб обмежити права користувачів у багатокористувацькому режимі, тим самим запобігаючи несанкціонованій модифікації кейсу або його структури.

Приклади застосування Autopsy на практиці

На практиці Autopsy широко застосовується як у кримінальних розслідуваннях, так і в реагуваннях на інциденти в процесах корпоративного рівня. Наприклад, у випадках аналізу шкідливого програмного забезпечення типу Raccoon Stealer, який спеціалізується на викраденні облікових даних, Autopsy дозволяє ідентифікувати характерну активність у файловій системі, виявляти конфігураційні файли, лог-файли або залишки шкідливих процесів [22].

Зокрема, завдяки модулям, що аналізують браузерні дані, Autopsy може виявляти артефакти з Chrome, Firefox, Edge, включаючи cookie-файли, історію відвіданих сайтів, збережені паролі та autofill-дані. У розслідуваннях, пов’язаних із викраденням облікових даних, така інформація є критично важливою для розуміння вектору атаки та її масштабу.

У розслідування корпоративних витоків конфіденційної інформації, Autopsy застосовується для відновлення видалених файлів, аналізу зовнішніх підключень, перевірки активності USB-пристроїв, що дозволяє виявити спроби несанкціонованого копіювання або передачі важливих даних [23].

2.3 Основні компоненти рішення Wireshark

Графічний інтерфейс користувача (GUI)

Інтерфейс користувача є центральним компонентом взаємодії користувача з Wireshark. Побудований за принципом багатовіконного відображення, GUI надає інтуїтивно зрозумілий доступ до інформації, отриманої в результаті аналізу трафіку.

Основні елементи GUI включають:

Панель переліку пакетів (Packet List Pane) – відображає список усіх захоплених пакетів, упорядкованих за часом отримання, з основною інформацією (номер, час, джерело, призначення, протокол, довжина тощо);

Панель деталей пакету (Packet Details Pane) – дозволяє користувачеві вивчити структуру обраного пакету в ієрархічному вигляді;

Панель байтів (Packet Bytes Pane) – показує необроблений вміст пакету в шістнадцятковому та ASCII форматах [24].

Інтерфейс також підтримує функції пошуку, маркування, статистики, експорту та фільтрації, що істотно спрощує процес аналізу великих обсягів даних.

Драйвери захоплення пакетів

Wireshark не здійснює захоплення трафіку безпосередньо. Замість цього він використовує спеціалізовані бібліотеки, які надають доступ до мережевого інтерфейсу на низькому рівні. Основною такою бібліотекою є libpcap (у Linux/Unix системах) або WinPcap/Npcap (у Windows-середовищах).

Ці бібліотеки працюють як посередники між Wireshark і мережевим адаптером, забезпечуючи захоплення вхідного та вихідного трафіку. У контексті цифрової криміналістики важливо підкреслити, що захоплення здійснюється в пасивному режимі, тобто без впливу на об’єкти дослідження, що відповідає принципам недоторканності доказів.

Система фільтрації

Один з ефективних інструментів Wireshark – це система фільтрації, яка дозволяє користувачеві зосередитися лише на релевантному трафіку. Існує два основних типи фільтрів:

Фільтри захоплення (Capture Filters) – застосовуються до даних у процесі їх захоплення, дозволяючи записувати лише потрібні пакети. Синтаксис заснований на правилах libpcap, наприклад: tcp port 80;

Фільтри відображення (Display Filters) – дозволяють користувачеві аналізувати лише ті пакети, які відповідають певним умовам. Приклад: http.request.method == "GET" [25];

Декодування протоколів

Wireshark підтримує понад 2500 мережевих протоколів, які можуть бути автоматично розпізнані та декодовані. Для кожного протоколу реалізовано відповідний модуль, який розшифровує структуру пакету та надає її у зручному для користувача вигляді.

Ці модулі дозволяють не лише бачити заголовки та корисне навантаження, але й виконують додаткову обробку, наприклад, відображення TLS-сертифікатів, збирання HTTP-сесій чи реконструкцію зображень, переданих через протокол HTTP [26].

Користувач також може вручну вказати інтерпретацію певних портів або типів трафіку, що підвищує гнучкість аналізу.

Інструменти для аналізу та статистики

Wireshark надає набір інструментів для поглибленого аналізу мережевого трафіку. Вони дозволяють не лише переглядати окремі пакети, а й отримувати зведену інформацію про мережеві з'єднання, трафік за протоколами та часову динаміку.

Основні функції включають:

Статистика протоколів (Protocol Hierarchy) – дозволяє побачити розподіл трафіку за протоколами та їх обсяг;

Статистика кінцевих точок (Endpoints) – надає інформацію про всі мережеві адреси, задіяні у сесії, включно з кількістю пакетів і обсягом переданих даних;

Статистика потоків (Conversations) – відображає пари джерело-призначення для TCP, UDP та інших протоколів;

Графік потоку (IO Graphs) – дозволяє будувати графіки активності в залежності від часу, що є корисним для виявлення аномалій у мережевому трафіку.

Ці інструменти є особливо важливими при аналізі атак, наприклад, визначенні наявності DDoS-атак або витоків даних.

Підтримка скриптів та розширень

Wireshark підтримує розширення функціоналу за допомогою скриптів на мові Lua. Це дозволяє створювати власні протокольні декодери, фільтри або інструменти аналізу без необхідності змінювати вихідний код програми.

Основні можливості скриптів у Wireshark:

розробка власних дисекторів (dissector) для нестандартних або рідкісних протоколів;

автоматизація складних аналізів та обробка великих обсягів даних;

створення спеціалізованих звітів або експорту даних у користувацьких форматах.

Крім того, існує розвинене співтовариство користувачів та розробників, яке регулярно публікує нові скрипти та доповнення, що значно розширюють можливості базового функціоналу Wireshark.

Безпека при використанні Wireshark

Хоча Wireshark є ефективним інструментом, його використання передбачає певні ризики з точки зору безпеки, особливо при захопленні трафіку у відкритих мережах. Оскільки Wireshark працює на рівні мережевого інтерфейсу, йому необхідні права адміністратора, що створює потенційну вразливість.

Основні заходи безпеки при роботі з Wireshark:

використання окремого облікового запису або віртуального середовища для запуску Wireshark;

захоплення трафіку в безпечному середовищі та обмеження доступу до захоплених даних;

регулярне оновлення програми для усунення відомих вразливостей.

Wireshark також підтримує функцію "Capture Privileges Separation", яка дозволяє мінімізувати ризики, виконуючи процес захоплення з підвищеними привілеями, а інтерфейс користувача – з обмеженими.

Висновок

Wireshark – це комплексне рішення для захоплення та аналізу мережевого трафіку, яке складається з багатьох компонентів, що працюють у тісній взаємодії. Графічний інтерфейс користувача забезпечує зручну навігацію даними, система фільтрації допомагає швидко знаходити необхідні пакети, модулі декодування протоколів надають детальну інформацію про структуру трафіку, а аналітичні інструменти дозволяють виявляти закономірності та аномалії. Підтримка скриптів робить Wireshark гнучким інструментом, здатним адаптуватися під специфічні завдання. Завдяки цим можливостям Wireshark є незамінним інструментом як для фахівців з інформаційної безпеки, так і для дослідників мережевих технологій.

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ВИЯВЛЕННЯ АКТИВНСТІ ШКІДЛИВОГО ЗАБЕЗПЕЧЕННЯ RACCOON STEALER НА КІНЦЕВИХ ТОЧКАХ

3.1 Алгоритм застосування рішень для виявлення шкідливого програмного забезпечення Raccoon Stealer

Алгоритм застосування Wireshark для аналізу мережевої активності

На першому етапі було запущено Wireshark та вибрано активний мережевий інтерфейс (Ethernet), через який здійснюється трафік аналізованої системи. Було активовано запис усіх вхідних і вихідних пакетів у реальному часі.

Фільтрація трафіку в Wireshark

Для виділення лише релевантних до аналізу даних було застосовано фільтр відображення:

```
(ip.addr == 10.0.2.15) && (dns || tls || tcp.port == 80 ||
tcp.port == 443)
```

Рис.3.1 Фільтр відображення трафіку в Wireshark

Цей фільтр дозволив зосередитися на трафіку від або до IP-адреси підозрюваної машини (10.0.2.15), обмежившись лише DNS-запитами, TLS-трафіком, HTTP (порт 80) та HTTPS (порт 443), які найчастіше використовуються ШПЗ для передачі вкрадених даних або з'єднання з C2-сервером.

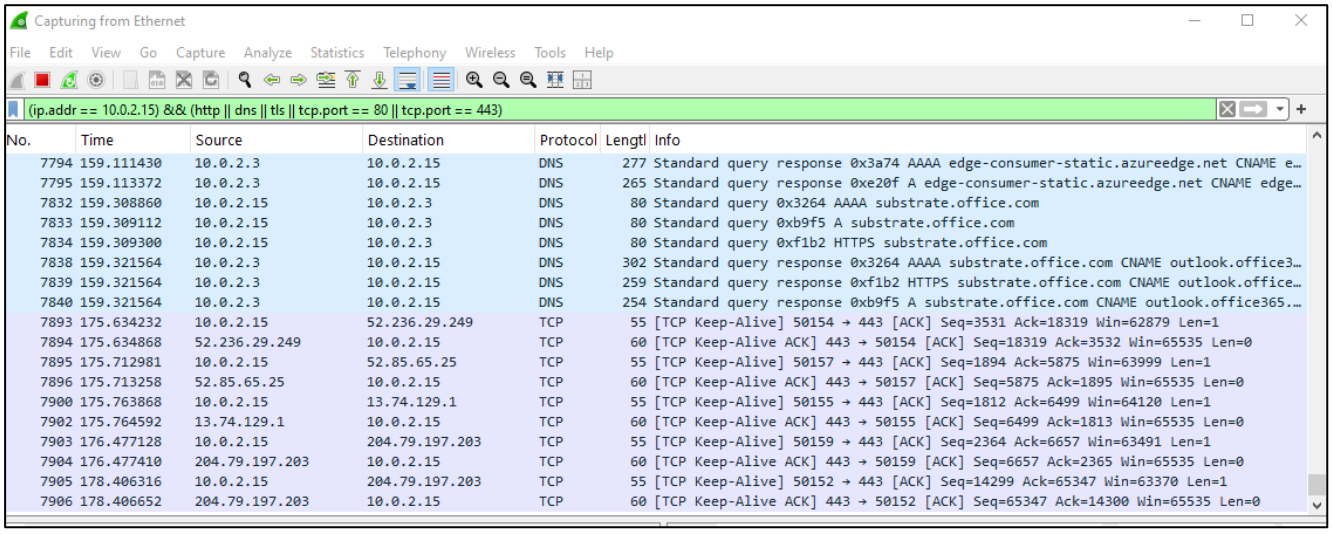


Рис.3.2 Фільтрація трафіку в Wireshark

Виявлення підозрілих з'єднань в Wireshark

У результаті захоплення трафіку було переглянуто перелік усіх встановлених з'єднань. Особливу увагу приділено:

TCP-сесіям із повторним використанням портів (ознака нестандартної поведінки);

з'єднанням, які завершувались RST-пакетами, що може свідчити про відхилення або блокування з'єднання з сервером керування;

великій кількості коротких запитів типу TCP Keep-Alive, які можуть бути періодичних спроб спілкування з віддаленим сервером.

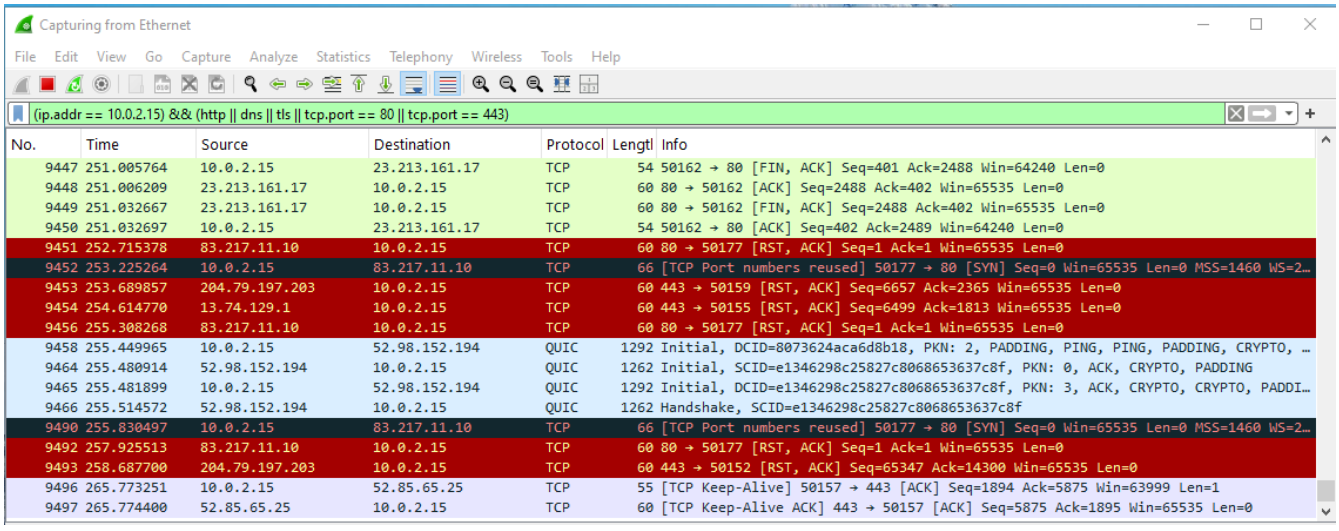


Рис.3.3 Виявлення з'єднань в Wireshark

Аналіз DNS-запитів в Wireshark

Переглянуто запити до DNS-серверів, сформовані з IP-адреси підозрюваного вузла.

Зокрема, перевірено:

які доменні імена запитувалися;

які IP-адреси було повернуто у відповідях;

чи містять ці імена ознаки маскування під легітимні сервіси (наприклад, azureedge.net, substrate.office.com тощо).

Аналіз вмісту пакетів в Wireshark

Для глибшого розуміння було обрано окремі пакети для перегляду на рівні:

Ethernet II - джерело й одержувач MAC-адрес;

IP - джерело й одержувач IP-адрес, TTL, checksum тощо;

TCP - порти, прапори (SYN, ACK, FIN, RST), порядкові номери, вікна, довжина;

Payload (Data) – зміст пакета у вигляді шістнадцяткового і ASCII-представлення.

Збереження результату в Wireshark для подальшого аналізу

Отримані пакети збережено у файл формату .pcapng під назвою «raccoon_stealer», який може бути повторно використаний для документування артефактів або створення звітів.

Алгоритм застосування Velociraptor

У веб-інтерфейсі Velociraptor створено нове полювання з тегом "Raccoon Stealer". Було обрано опцію запуску на всіх клієнтах (Run everywhere) і ввімкнено “Start Hunt Immediately”.

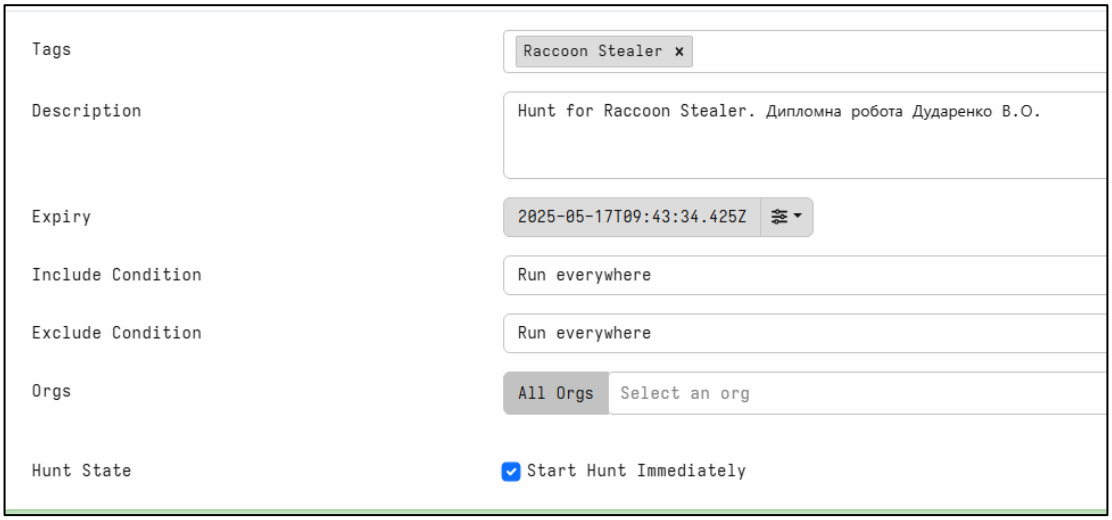


Рис.3.4 Створене нове полювання в Velociraptor

Вибір артефактів для збору

Зі стандартної бібліотеки Velociraptor обрано набір артефактів, що дозволяють виявити ключові ознаки шкідливої активності:

Windows.Forensics.Prefetch – інформація про нещодавно запущені файли .exe;

Windows.Registry.NTUser – дані користувацького реєстру, зокрема автозапуск;

Windows.Registry.MountPoints2 – контроль за підключенням зовнішніх носіїв;

Windows.Registry.RecentDocs – перелік нещодавно відкритих документів;

Windows.Sysinternals.Autoruns – точки автозапуску;

Windows.EventLogs.EvtxHunter – аналіз журналів подій Windows;

Windows.EventLogs.ScheduledTasks – перевірка запланованих завдань (Task Scheduler);

Windows.EventLogs.ServiceCreationComspec – відстеження створення нових служб;

Windows.EventLogs.Modifications – зміни в журналах подій;

Windows.EventLogs.PowershellScriptblock / Module – виконання PowerShell-скриптів;

Windows.Detection.Amcache – встановлені застосунки;

Windows.System.Amcache – кеш виконуваних програм;

Windows.Registry.AppCompatCache – історія запуску додатків;

Windows.System.TaskScheduler – завдання в планувальнику завдань.

Запуск і збір даних

Полювання розпочалося автоматично після створення. Підключений клієнт виконав усі запити. У вкладці Log фіксувалося успішне виконання кожного артефакта.

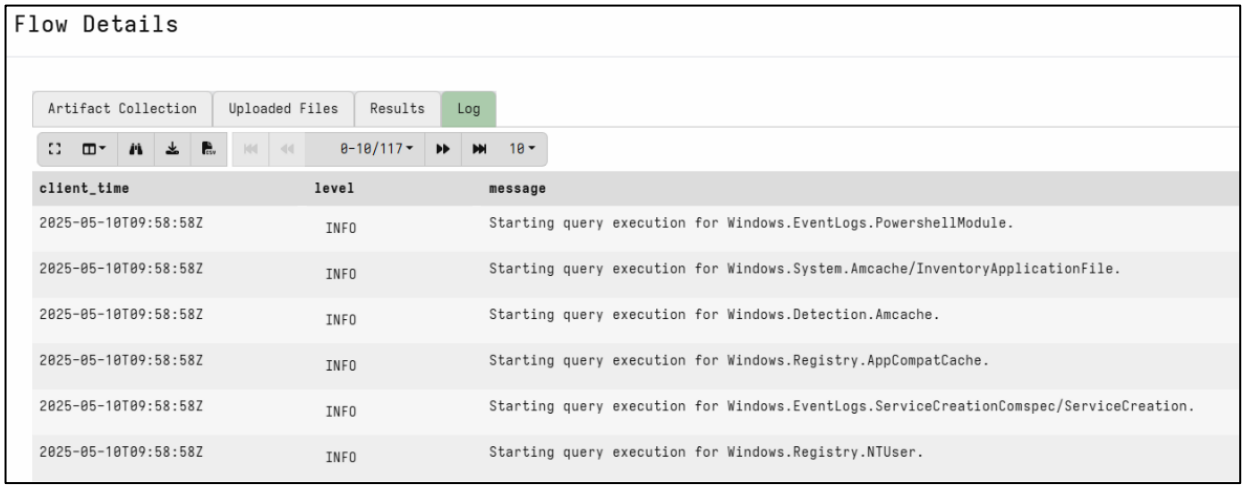


Рис.3.5 Перегляд артефактів у вкладці «Log»

Перегляд результатів

У вкладці Results відображено результати. Такі результати дозволяють виявити нестандартні завдання, які могли бути створені ШПЗ для забезпечення стійкості в системі.

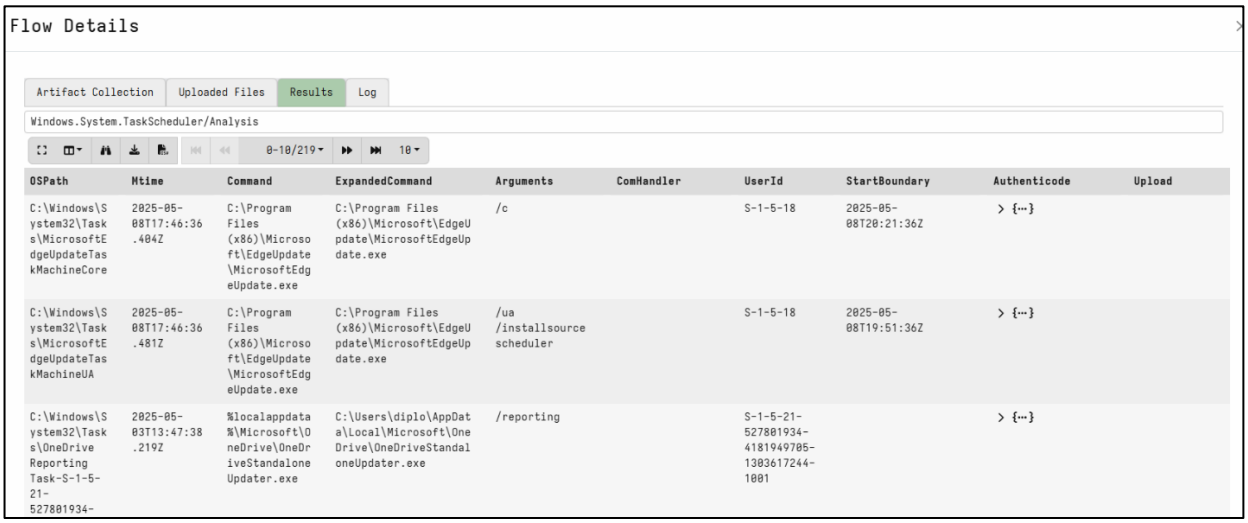


Рис.3.6 Перегляд результатів в Velociraptor

Завершення полювання та збереження даних

Після завершення результати було збережено у вигляді CSV-файлів, а також експортовано для подальшого включення в технічний звіт.

Алгоритм застосування Autopsy

У справу додано джерело типу disk image (E01). Всі параметри залишено за замовчуванням. Після додавання джерело автоматично почало індексуватись.

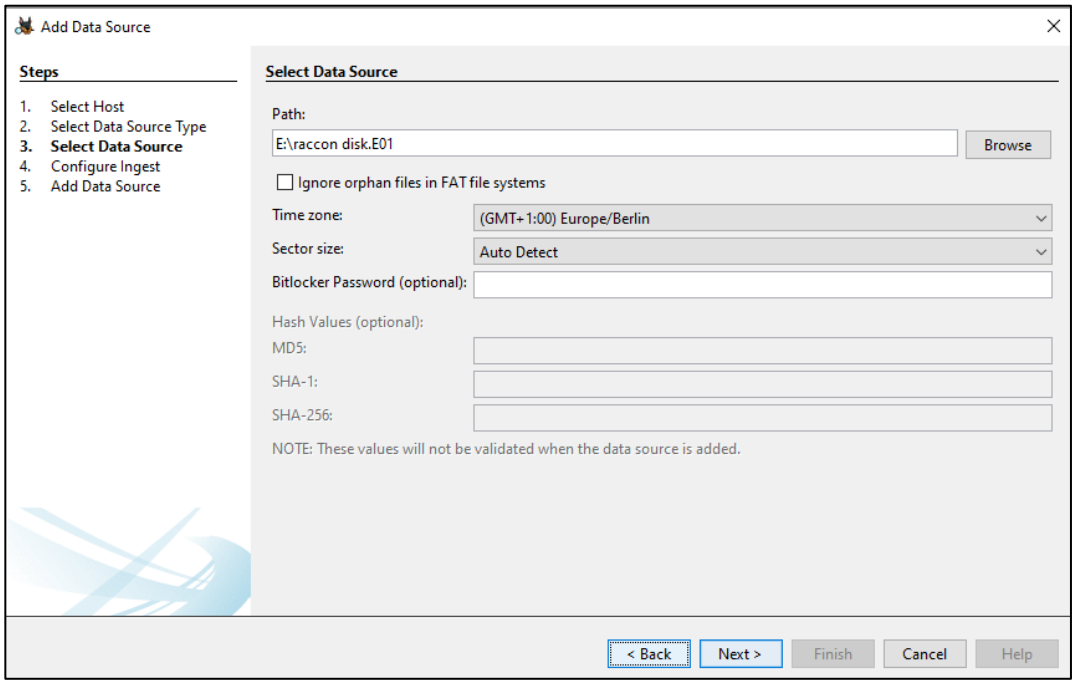


Рис.3.7 Додавання джерела для аналізу

Вибір модулів аналізу

Для автоматичного збору артефактів були активовані стандартні модулі Autopsy, які забезпечують виявлення основної користувацької активності, пов’язаної з потенційно шкідливим програмним забезпеченням.

Перегляд результатів

У результатах було знайдено:

файли в Recycle Bin;

підозрілі документи;

файли з ознаками шифрування;

профілі користувачів системи.

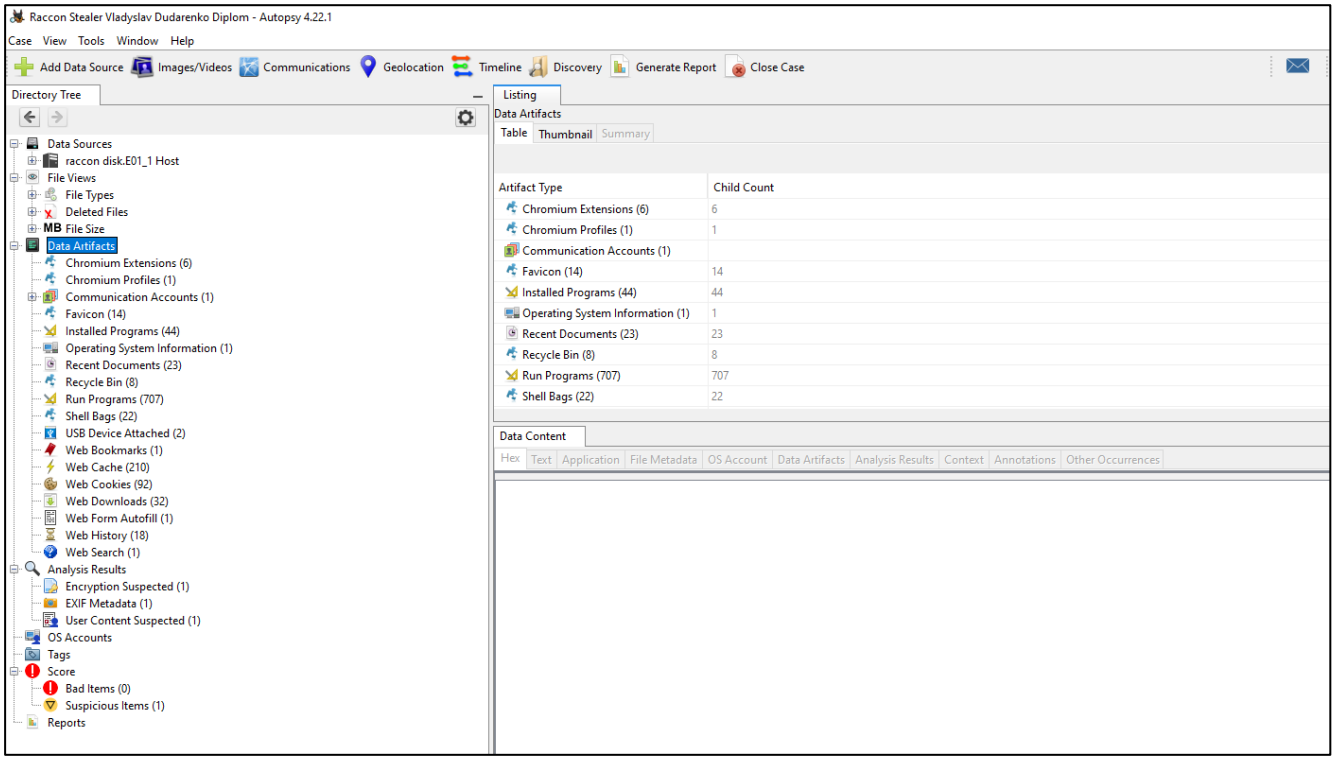


Рис.3.8 Перегляд результатів в Autopsy

Збереження звіту

Після аналізу згенеровано звіт у форматі .html, що містить виявлені артефакти та основні висновки.

3.2 Артефакти у реєстрі (ключі, пов’язані з Raccoon Stealer)

У цьому розділі здійснено дослідження трафіку, пов’язаного з діяльністю шкідливої програми Raccoon Stealer. Виявлені індикатори мережевої активності були класифіковані та задокументовані для підтвердження фактів передачі конфіденційної інформації, командного керування (C2), а також технік приховування взаємодії з серверами.

Алгоритм роботи Raccoon Stealer

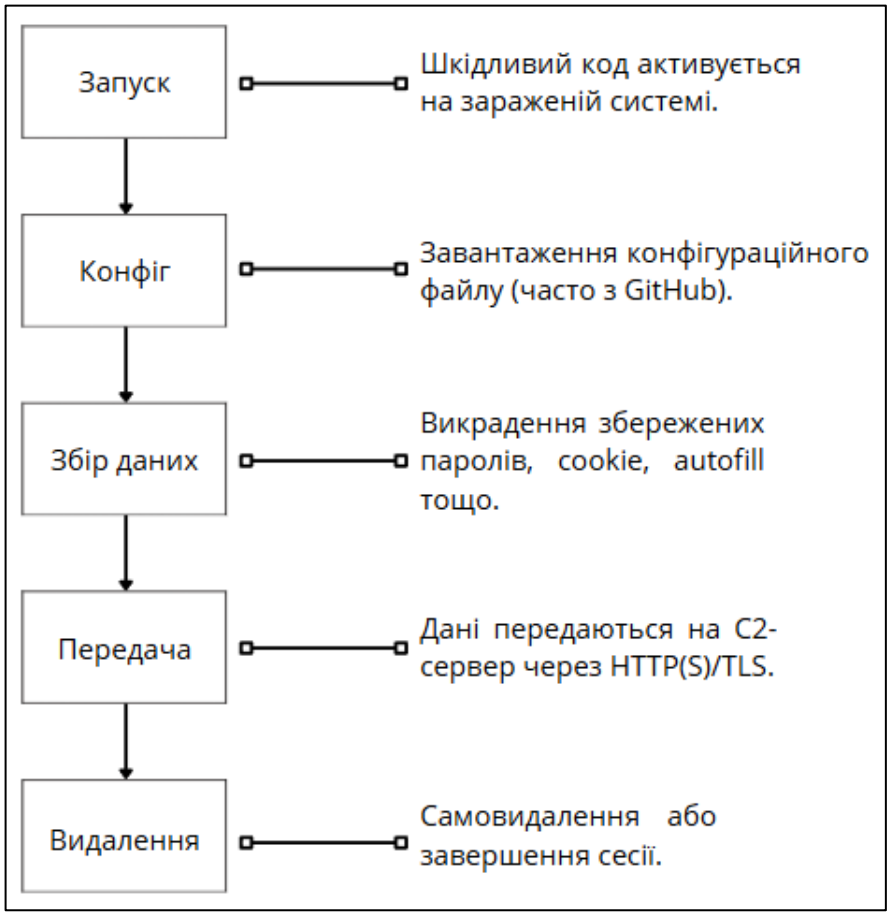
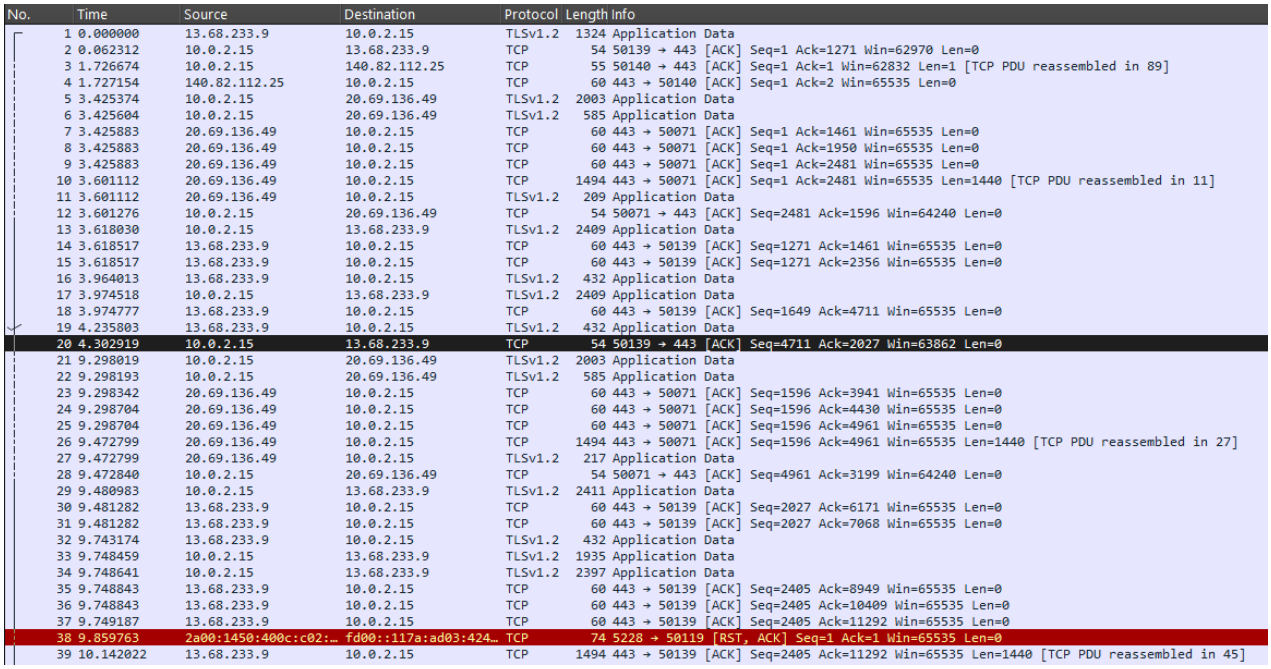


Рис.3.9 Блок-схема роботи Raccoon Stealer

TLS-з'єднання з хмарними сервісами та аномальне IPv6-з'єднання

Виявлено TLS-з'єднання з IP-адресами, пов'язаними з Azure (13.68.233.9, 20.69.136.49) та GitHub (140.82.112.25), яке розпочалося одразу після запуску системи. Це вказує на можливе завантаження конфігурацій та ініціацію передачі даних. Також зафіксовано IPv6-з'єднання з Google, що раптово завершується TCP Reset – характерна ознака прихованого каналу зв'язку або альтернативного C2.

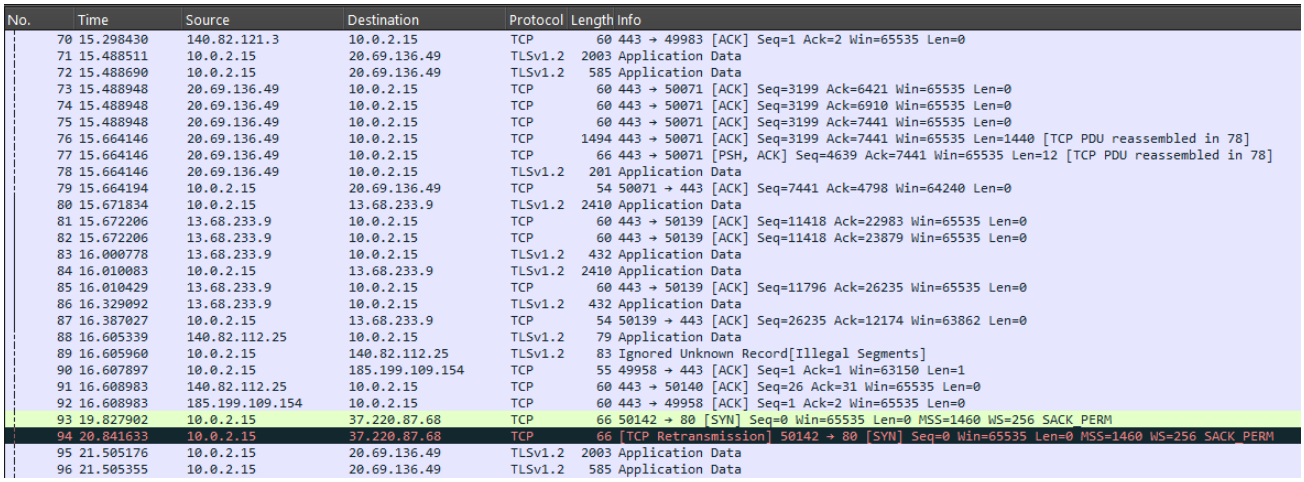


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	13.68.233.9	10.0.2.15	TLSv1.2	1324	Application Data
2	0.062312	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=1 Ack=1271 Win=62970 Len=0
3	1.726674	10.0.2.15	140.82.112.25	TCP	55	50140 → 443 [ACK] Seq=1 Ack=1 Win=62832 Len=1 [TCP PDU reassembled in 89]
4	1.727154	140.82.112.25	10.0.2.15	TCP	60	443 → 50140 [ACK] Seq=1 Ack=2 Win=65535 Len=0
5	3.425374	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
6	3.425604	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data
7	3.425883	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=1 Ack=1461 Win=65535 Len=0
8	3.425883	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=1 Ack=1950 Win=65535 Len=0
9	3.425883	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=1 Ack=2481 Win=65535 Len=0
10	3.601112	20.69.136.49	10.0.2.15	TCP	1494	443 → 50071 [ACK] Seq=1 Ack=2481 Win=65535 Len=1440 [TCP PDU reassembled in 11]
11	3.601112	20.69.136.49	10.0.2.15	TLSv1.2	209	Application Data
12	3.601276	10.0.2.15	20.69.136.49	TCP	54	50071 → 443 [ACK] Seq=2481 Ack=1596 Win=64240 Len=0
13	3.618030	10.0.2.15	13.68.233.9	TLSv1.2	2409	Application Data
14	3.618517	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=1271 Ack=1461 Win=65535 Len=0
15	3.618517	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=1271 Ack=2356 Win=65535 Len=0
16	3.964013	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
17	3.974518	10.0.2.15	13.68.233.9	TLSv1.2	2409	Application Data
18	3.974777	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=1649 Ack=4711 Win=65535 Len=0
19	4.235803	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
20	4.302919	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=4711 Ack=2027 Win=63862 Len=0
21	9.298019	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
22	9.298193	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data
23	9.298342	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=1596 Ack=3941 Win=65535 Len=0
24	9.298704	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=1596 Ack=4430 Win=65535 Len=0
25	9.298704	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=1596 Ack=4961 Win=65535 Len=0
26	9.472799	20.69.136.49	10.0.2.15	TCP	1494	443 → 50071 [ACK] Seq=1596 Ack=4961 Win=65535 Len=1440 [TCP PDU reassembled in 27]
27	9.472799	20.69.136.49	10.0.2.15	TLSv1.2	217	Application Data
28	9.472840	10.0.2.15	20.69.136.49	TCP	54	50071 → 443 [ACK] Seq=4961 Ack=3199 Win=64240 Len=0
29	9.480983	10.0.2.15	13.68.233.9	TLSv1.2	2411	Application Data
30	9.481282	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=2027 Ack=6171 Win=65535 Len=0
31	9.481282	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=2027 Ack=7068 Win=65535 Len=0
32	9.743174	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
33	9.748459	10.0.2.15	13.68.233.9	TLSv1.2	1935	Application Data
34	9.748641	10.0.2.15	13.68.233.9	TLSv1.2	2397	Application Data
35	9.748843	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=2405 Ack=8949 Win=65535 Len=0
36	9.748843	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=2405 Ack=10409 Win=65535 Len=0
37	9.749187	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=2405 Ack=11292 Win=65535 Len=0
38	9.859763	2a00:1450:400c:c02:…	fd00::117a:ad03:424…	TCP	74	5228 → 50119 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
39	10.142022	13.68.233.9	10.0.2.15	TCP	1494	443 → 50139 [ACK] Seq=2405 Ack=11292 Win=65535 Len=1440 [TCP PDU reassembled in 45]

Рис.3.10 Аномальне з’єднання

Beaconing-з'єднання з IP 37.220.87.68

На даному етапі видно, як хост 10.0.2.15 намагається встановити з'єднання з IP 37.220.87.68 через порт 80, надсилаючи SYN-пакети. Сервер не відповідає, що призводить до повторного надсилання SYN. Така активність характерна для ШПЗ, що працює з резервними C2-адресами або використовує механізм регулярного з'єднання.



No.	Time	Source	Destination	Protocol	Length	Info
70	15.298430	140.82.121.3	10.0.2.15	TCP	60	443 → 49983 [ACK] Seq=1 Ack=2 Win=65535 Len=0
71	15.488511	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
72	15.488690	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data
73	15.488948	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=3199 Ack=6421 Win=65535 Len=0
74	15.488948	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=3199 Ack=6910 Win=65535 Len=0
75	15.488948	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=3199 Ack=7441 Win=65535 Len=0
76	15.664146	20.69.136.49	10.0.2.15	TCP	1494	443 → 50071 [ACK] Seq=3199 Ack=7441 Win=65535 Len=1440 [TCP PDU reassembled in 78]
77	15.664146	20.69.136.49	10.0.2.15	TCP	66	443 → 50071 [PSH, ACK] Seq=4639 Ack=7441 Win=65535 Len=12 [TCP PDU reassembled in 78]
78	15.664146	20.69.136.49	10.0.2.15	TLSv1.2	201	Application Data
79	15.664194	10.0.2.15	20.69.136.49	TCP	54	50071 → 443 [ACK] Seq=7441 Ack=4798 Win=64240 Len=0
80	15.671834	10.0.2.15	13.68.233.9	TLSv1.2	2410	Application Data
81	15.672206	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=11418 Ack=22983 Win=65535 Len=0
82	15.672206	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=11418 Ack=23879 Win=65535 Len=0
83	16.000778	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
84	16.010083	10.0.2.15	13.68.233.9	TLSv1.2	2410	Application Data
85	16.010429	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=11796 Ack=26235 Win=65535 Len=0
86	16.329092	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
87	16.387027	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=26235 Ack=12174 Win=63862 Len=0
88	16.605339	140.82.112.25	10.0.2.15	TLSv1.2	79	Application Data
89	16.605960	10.0.2.15	140.82.112.25	TLSv1.2	83	Ignored Unknown Record[Illegal Segments]
90	16.607897	10.0.2.15	185.199.109.154	TCP	55	49958 → 443 [ACK] Seq=1 Ack=1 Win=63150 Len=1
91	16.608983	140.82.112.25	10.0.2.15	TCP	60	443 → 50140 [ACK] Seq=26 Ack=31 Win=65535 Len=0
92	16.608983	185.199.109.154	10.0.2.15	TCP	60	443 → 49958 [ACK] Seq=1 Ack=2 Win=65535 Len=0
93	19.827902	10.0.2.15	37.220.87.68	TCP	66	50142 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
94	20.841633	10.0.2.15	37.220.87.68	TCP	66	[TCP Retransmission] 50142 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
95	21.505176	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
96	21.505355	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data

Рис.3.11 Beaconing-з'єднання

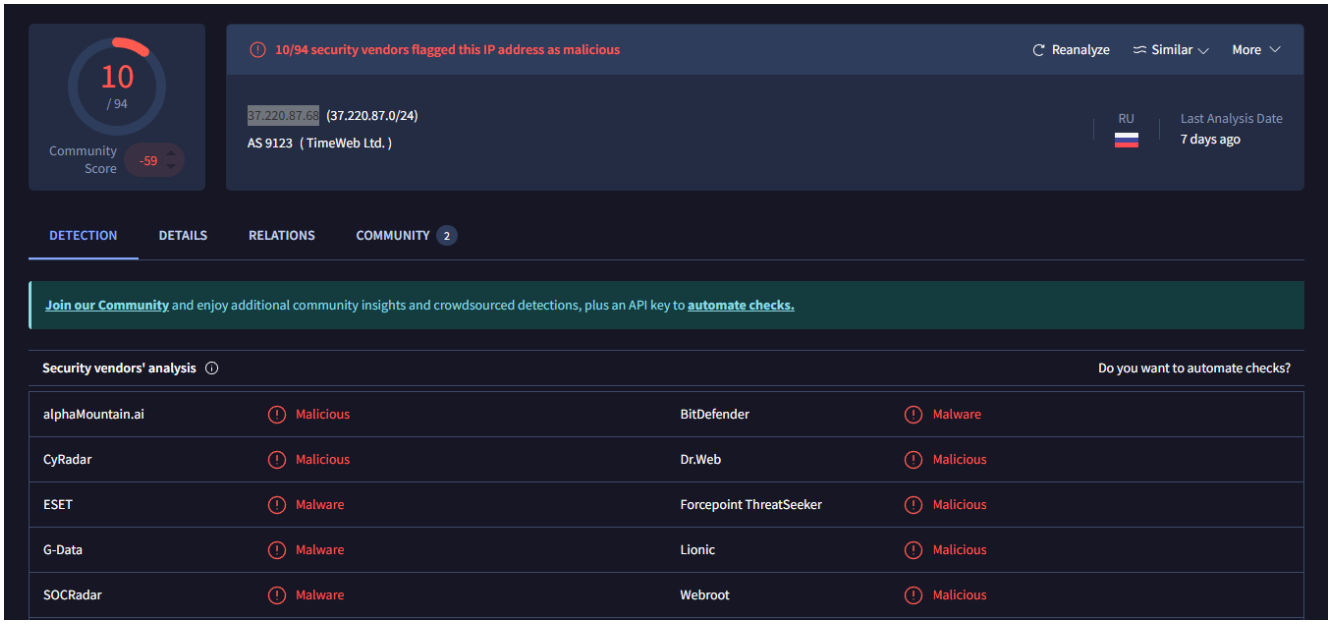
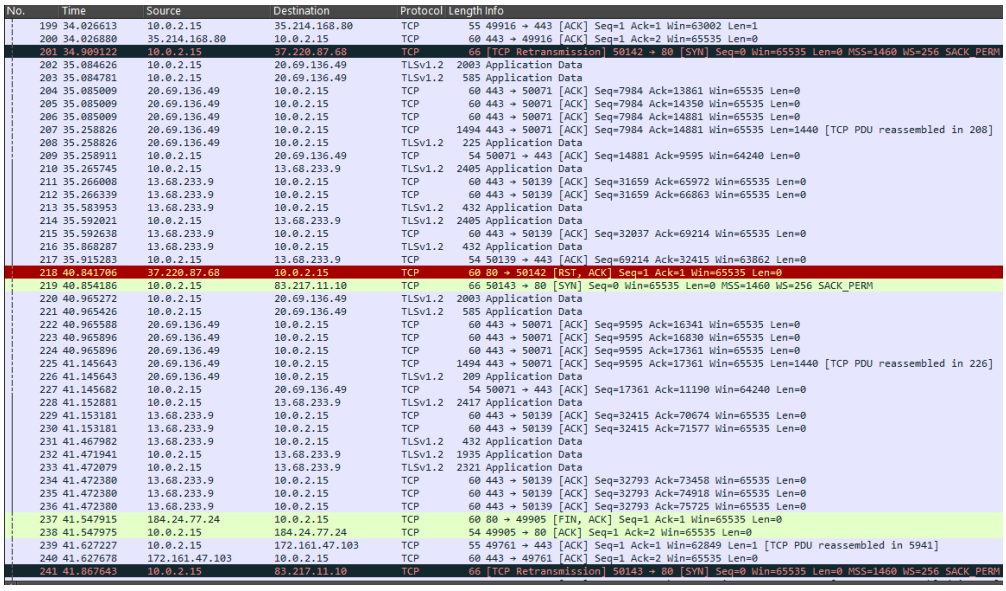


Рис.3.12 Аналіз IP-адреси 37.220.87.68 через VirusTotal

Поведінка Raccoon Stealer при недоступному C2

На цьому етапі видно серію спроб зв'язку з двома підозрілими серверами – 37.220.87.68 та 83.217.11.10. У першому випадку сервер повертає RST, ACK, відмовляючись від з'єднання. У другому клієнт не отримує відповіді й повторює SYN-запити. Така поведінка вказує на наявність списку C2-серверів, до яких Raccoon Stealer намагається підключитися в циклі. Це типовий механізм резервного зв'язку шкідливих програм, якщо основний C2 недоступний.



No.	Time	Source	Destination	Protocol	Length	Info
199	34.026613	10.0.2.15	35.214.168.80	TCP	55	49916 → 443 [ACK] Seq=1 Ack=1 Win=63002 Len=1
200	34.026880	35.214.168.80	10.0.2.15	TCP	60	443 → 49916 [ACK] Seq=1 Ack=2 Win=65535 Len=0
201	34.909122	10.0.2.15	37.220.87.68	TCP	66	[TCP Retransmission] 50142 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
202	35.084626	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
203	35.084781	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data
204	35.085009	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=7984 Ack=13861 Win=65535 Len=0
205	35.085009	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=7984 Ack=14350 Win=65535 Len=0
206	35.085009	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=7984 Ack=14881 Win=65535 Len=0
207	35.258826	20.69.136.49	10.0.2.15	TCP	1494	443 → 50071 [ACK] Seq=7984 Ack=14881 Win=65535 Len=1440 [TCP PDU reassembled in 208]
208	35.258826	20.69.136.49	10.0.2.15	TLSv1.2	225	Application Data
209	35.258911	10.0.2.15	20.69.136.49	TCP	54	50071 → 443 [ACK] Seq=14881 Ack=9595 Win=64240 Len=0
210	35.265745	10.0.2.15	13.68.233.9	TLSv1.2	2405	Application Data
211	35.266008	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=31659 Ack=65972 Win=65535 Len=0
212	35.266339	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=31659 Ack=66863 Win=65535 Len=0
213	35.583953	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
214	35.592021	10.0.2.15	13.68.233.9	TLSv1.2	2405	Application Data
215	35.592638	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=32037 Ack=69214 Win=65535 Len=0
216	35.868287	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
217	35.915283	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=69214 Ack=32415 Win=63862 Len=0
218	40.841706	37.220.87.68	10.0.2.15	TCP	60	80 → 50142 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
219	40.854186	10.0.2.15	83.217.11.10	TCP	66	50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
220	40.965272	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
221	40.965426	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data
222	40.965588	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=9595 Ack=16341 Win=65535 Len=0
223	40.965896	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=9595 Ack=16830 Win=65535 Len=0
224	40.965896	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=9595 Ack=17361 Win=65535 Len=0
225	41.145643	20.69.136.49	10.0.2.15	TCP	1494	443 → 50071 [ACK] Seq=9595 Ack=17361 Win=65535 Len=1440 [TCP PDU reassembled in 226]
226	41.145643	20.69.136.49	10.0.2.15	TLSv1.2	209	Application Data
227	41.145682	10.0.2.15	20.69.136.49	TCP	54	50071 → 443 [ACK] Seq=17361 Ack=11190 Win=64240 Len=0
228	41.152881	10.0.2.15	13.68.233.9	TLSv1.2	2417	Application Data
229	41.153181	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=32415 Ack=70674 Win=65535 Len=0
230	41.153181	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=32415 Ack=71577 Win=65535 Len=0
231	41.467982	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
232	41.471941	10.0.2.15	13.68.233.9	TLSv1.2	1935	Application Data
233	41.472079	10.0.2.15	13.68.233.9	TLSv1.2	2321	Application Data
234	41.472380	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=32793 Ack=73458 Win=65535 Len=0
235	41.472380	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=32793 Ack=74918 Win=65535 Len=0
236	41.472380	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=32793 Ack=75725 Win=65535 Len=0
237	41.547915	184.24.77.24	10.0.2.15	TCP	60	80 → 49905 [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
238	41.547975	10.0.2.15	184.24.77.24	TCP	54	49905 → 80 [ACK] Seq=1 Ack=2 Win=65535 Len=0
239	41.627227	10.0.2.15	172.161.47.103	TCP	55	49761 → 443 [ACK] Seq=1 Ack=1 Win=62849 Len=1 [TCP PDU reassembled in 5941]
240	41.627678	172.161.47.103	10.0.2.15	TCP	60	443 → 49761 [ACK] Seq=1 Ack=2 Win=65535 Len=0
241	41.867643	10.0.2.15	83.217.11.10	TCP	66	[TCP Retransmission] 50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

Рис.3.13 Підключення до резервного C2

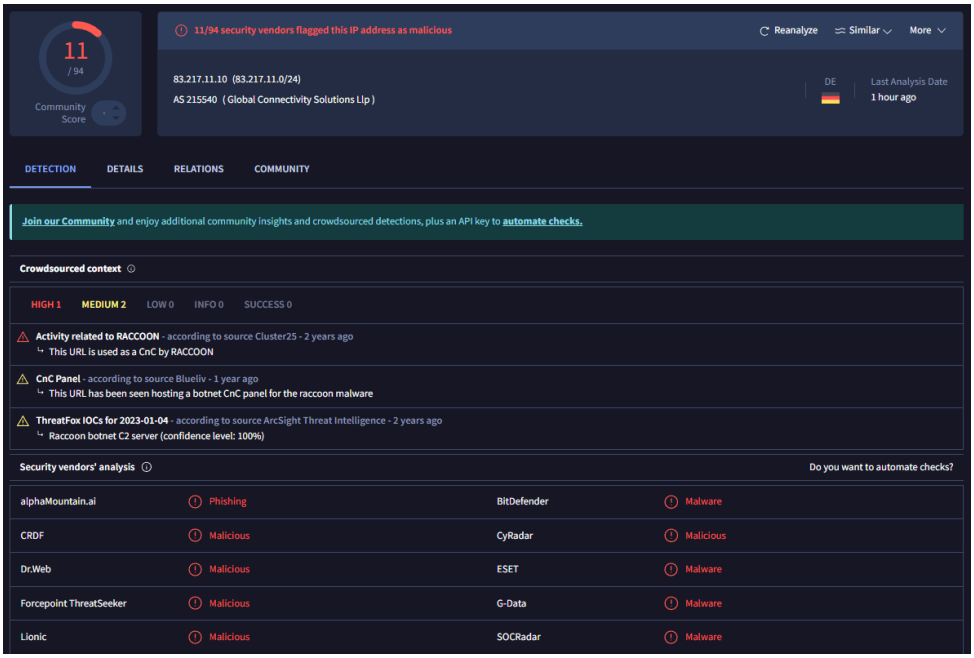
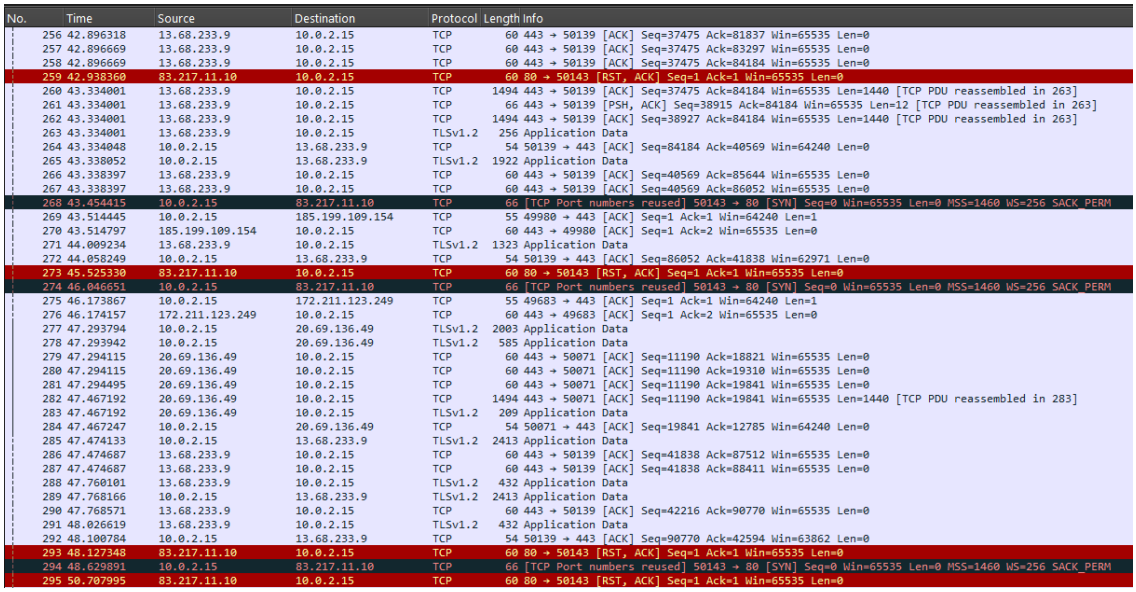


Рис.3.14 Аналіз IP-адреси 83.217.11.10 через VirusTotal

Виявлення активного C2-сервера та його поведінки

Зафіксовано численні спроби встановити з'єднання з IP-адресою 83.217.11.10 на порт 80. Кожна спроба або завершується RST, ACK з боку сервера, або вимагає повторного встановлення з'єднання з того самого порту, що є ознакою активного з’єднання. Такі ознаки характерні для InfoStealer-подібних програм, які мають вбудовану логіку повторних спроб у випадку невдалої комунікації з C2.

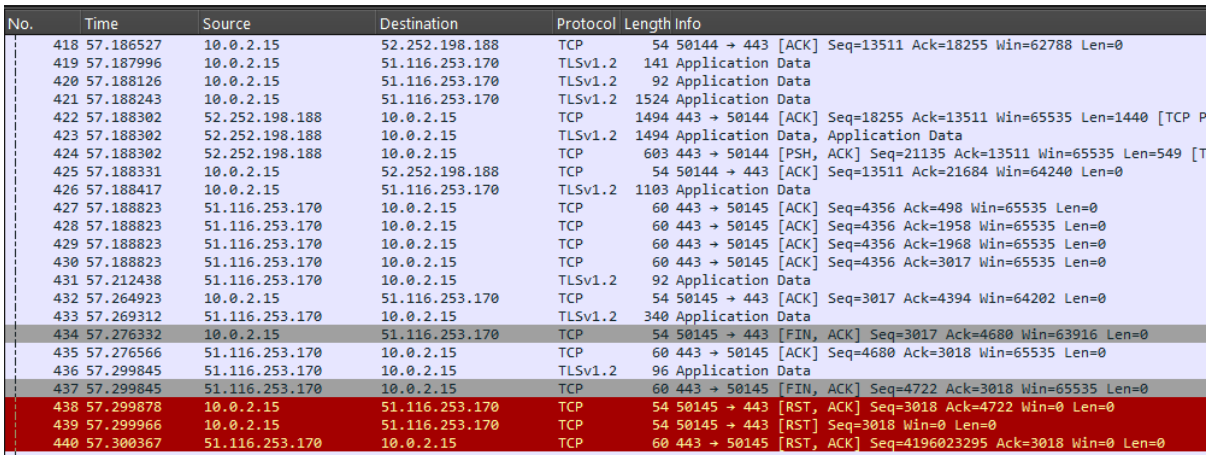


No.	Time	Source	Destination	Protocol	Length	Info
256	42.896318	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=37475 Ack=81837 Win=65535 Len=0
257	42.896669	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=37475 Ack=83297 Win=65535 Len=0
258	42.896669	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=37475 Ack=84184 Win=65535 Len=0
259	42.938360	83.217.11.10	10.0.2.15	TCP	60	80 → 50143 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
260	43.334001	13.68.233.9	10.0.2.15	TCP	1494	443 → 50139 [ACK] Seq=37475 Ack=84184 Win=65535 Len=1440 [TCP PDU reassembled in 263]
261	43.334001	13.68.233.9	10.0.2.15	TCP	66	443 → 50139 [PSH, ACK] Seq=38915 Ack=84184 Win=65535 Len=12 [TCP PDU reassembled in 263]
262	43.334001	13.68.233.9	10.0.2.15	TCP	1494	443 → 50139 [ACK] Seq=38927 Ack=84184 Win=65535 Len=1440 [TCP PDU reassembled in 263]
263	43.334001	13.68.233.9	10.0.2.15	TLSv1.2	256	Application Data
264	43.334048	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=84184 Ack=40569 Win=64240 Len=0
265	43.338052	10.0.2.15	13.68.233.9	TLSv1.2	1922	Application Data
266	43.338397	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=40569 Ack=85644 Win=65535 Len=0
267	43.338397	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=40569 Ack=86052 Win=65535 Len=0
268	43.454415	10.0.2.15	83.217.11.10	TCP	66	[TCP Port numbers reused] 50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
269	43.514445	10.0.2.15	185.199.109.154	TCP	55	49980 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=1
270	43.514797	185.199.109.154	10.0.2.15	TCP	60	443 → 49980 [ACK] Seq=1 Ack=2 Win=65535 Len=0
271	44.009234	13.68.233.9	10.0.2.15	TLSv1.2	1323	Application Data
272	44.058249	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=86052 Ack=41838 Win=62971 Len=0
273	45.525330	83.217.11.10	10.0.2.15	TCP	60	80 → 50143 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
274	46.046651	10.0.2.15	83.217.11.10	TCP	66	[TCP Port numbers reused] 50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
275	46.173867	10.0.2.15	172.211.123.249	TCP	55	49683 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=1
276	46.174157	172.211.123.249	10.0.2.15	TCP	60	443 → 49683 [ACK] Seq=1 Ack=2 Win=65535 Len=0
277	47.293794	10.0.2.15	20.69.136.49	TLSv1.2	2003	Application Data
278	47.293942	10.0.2.15	20.69.136.49	TLSv1.2	585	Application Data
279	47.294115	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=11190 Ack=18821 Win=65535 Len=0
280	47.294115	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=11190 Ack=19310 Win=65535 Len=0
281	47.294495	20.69.136.49	10.0.2.15	TCP	60	443 → 50071 [ACK] Seq=11190 Ack=19841 Win=65535 Len=0
282	47.467192	20.69.136.49	10.0.2.15	TCP	1494	443 → 50071 [ACK] Seq=11190 Ack=19841 Win=65535 Len=1440 [TCP PDU reassembled in 283]
283	47.467192	20.69.136.49	10.0.2.15	TLSv1.2	209	Application Data
284	47.467247	10.0.2.15	20.69.136.49	TCP	54	50071 → 443 [ACK] Seq=19841 Ack=12785 Win=64240 Len=0
285	47.474133	10.0.2.15	13.68.233.9	TLSv1.2	2413	Application Data
286	47.474687	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=41838 Ack=87512 Win=65535 Len=0
287	47.474687	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=41838 Ack=88411 Win=65535 Len=0
288	47.760101	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
289	47.768166	10.0.2.15	13.68.233.9	TLSv1.2	2413	Application Data
290	47.768571	13.68.233.9	10.0.2.15	TCP	60	443 → 50139 [ACK] Seq=42216 Ack=90770 Win=65535 Len=0
291	48.026619	13.68.233.9	10.0.2.15	TLSv1.2	432	Application Data
292	48.100784	10.0.2.15	13.68.233.9	TCP	54	50139 → 443 [ACK] Seq=90770 Ack=42594 Win=63862 Len=0
293	48.127348	83.217.11.10	10.0.2.15	TCP	60	80 → 50143 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
294	48.629891	10.0.2.15	83.217.11.10	TCP	66	[TCP Port numbers reused] 50143 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
295	50.707995	83.217.11.10	10.0.2.15	TCP	60	80 → 50143 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0

Рис.3.15 Повторні спроби з’єднання

Штучне завершення TLS-сесії

Після встановлення TLS-з'єднання між 10.0.2.15 та IP 51.116.253.170 сесія переривається через серію RST, ACK з обох сторін. Клієнт спочатку ініціює FIN, після чого сервер різко відмовляє через RST. Така поведінка характерна для Raccoon Stealer, що використовує швидке примусове завершення TLS-з'єднання, щоб приховати слід комунікації та унеможливити дешифрування.



No.	Time	Source	Destination	Protocol	Length	Info
418	57.186527	10.0.2.15	52.252.198.188	TCP	54	50144 → 443 [ACK] Seq=13511 Ack=18255 Win=62788 Len=0
419	57.187996	10.0.2.15	51.116.253.170	TLSv1.2	141	Application Data
420	57.188126	10.0.2.15	51.116.253.170	TLSv1.2	92	Application Data
421	57.188243	10.0.2.15	51.116.253.170	TLSv1.2	1524	Application Data
422	57.188302	52.252.198.188	10.0.2.15	TCP	1494	443 → 50144 [ACK] Seq=18255 Ack=13511 Win=65535 Len=1440 [TCP P
423	57.188302	52.252.198.188	10.0.2.15	TLSv1.2	1494	Application Data, Application Data
424	57.188302	52.252.198.188	10.0.2.15	TCP	603	443 → 50144 [PSH, ACK] Seq=21135 Ack=13511 Win=65535 Len=549 [T
425	57.188331	10.0.2.15	52.252.198.188	TCP	54	50144 → 443 [ACK] Seq=13511 Ack=21684 Win=64240 Len=0
426	57.188417	10.0.2.15	51.116.253.170	TLSv1.2	1103	Application Data
427	57.188823	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [ACK] Seq=4356 Ack=498 Win=65535 Len=0
428	57.188823	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [ACK] Seq=4356 Ack=1958 Win=65535 Len=0
429	57.188823	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [ACK] Seq=4356 Ack=1968 Win=65535 Len=0
430	57.188823	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [ACK] Seq=4356 Ack=3017 Win=65535 Len=0
431	57.212438	51.116.253.170	10.0.2.15	TLSv1.2	92	Application Data
432	57.264923	10.0.2.15	51.116.253.170	TCP	54	50145 → 443 [ACK] Seq=3017 Ack=4394 Win=64202 Len=0
433	57.269312	51.116.253.170	10.0.2.15	TLSv1.2	340	Application Data
434	57.276332	10.0.2.15	51.116.253.170	TCP	54	50145 → 443 [FIN, ACK] Seq=3017 Ack=4680 Win=63916 Len=0
435	57.276566	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [ACK] Seq=4680 Ack=3018 Win=65535 Len=0
436	57.299845	51.116.253.170	10.0.2.15	TLSv1.2	96	Application Data
437	57.299845	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [FIN, ACK] Seq=4722 Ack=3018 Win=65535 Len=0
438	57.299878	10.0.2.15	51.116.253.170	TCP	54	50145 → 443 [RST, ACK] Seq=3018 Ack=4722 Win=0 Len=0
439	57.299966	10.0.2.15	51.116.253.170	TCP	54	50145 → 443 [RST] Seq=3018 Win=0 Len=0
440	57.300367	51.116.253.170	10.0.2.15	TCP	60	443 → 50145 [RST, ACK] Seq=4196023295 Ack=3018 Win=0 Len=0

Рис.3.16 Завершення TLS-сесії

Висновок мережевого аналізу

Аналіз мережевого трафіку показав ознаки, типові для роботи Raccoon Stealer: періодичні спроби підключення до C2-серверів, використання хмарної інфраструктури, нестандартних портів та ретрансляції SYN-пакетів. Виявлено також штучне завершення TLS-сесій через RST, що ускладнює виявлення та аналіз. Ці ознаки свідчать про автоматизовану та приховану активність ШПЗ.

Аналіз логів від Velociraptor

На основі аналізу зібраних артефактів за допомогою інструменту Velociraptor було виявлено ознаки Raccoon Stealer. Аналіз охоплював системні журнали та файли Prefetch, що дозволило ідентифікувати нетипову поведінку окремих виконуваних файлів та встановити часові рамки їхньої активності.

Ідентифікація виконуваних файлів

У файловій системі було виявлено сліди запуску «FILECOAUTH.EXE» – файлу, який не належить до стандартних компонентів операційної системи Windows. Файл фіксувався в Prefetch-кеші, що вказує на його щонайменше п’ятикратне використання.

Маскування шкідливої активності

Увагу привернули також файли «VC_REDIST.X64.EXE», які зазвичай використовуються для встановлення бібліотек Visual C++ Redistributable. У даному випадку зафіксовано підозрілу активність двох версій цього файлу, що з'явилися одночасно та були виконані майже синхронно. Така поведінка може свідчити про спробу маскування шкідливих дій під виглядом системного оновлення або встановлення програмного забезпечення.

Можливий канал ексфільтрації

Серед активних процесів виявлено «ONEDRIVE.EXE», який, хоча і є легітимним додатком, міг бути використаний для несанкціонованої передачі даних. Така гіпотеза ґрунтується на типових для Raccoon Stealer техніках ексфільтрації через хмарні сервіси.

Хронологія активності

Згідно з часовими мітками у Prefetch файлах, підозріла активність розпочалася щонайменше 29 квітня 2025 року та продовжувалась до 10 травня 2025 року. Це узгоджується з поведінкою *Raccoon Stealer*, який зазвичай діє приховано, викрадаючи збережені паролі, cookie браузера, криптогаманці та інші конфіденційні дані користувача.

Висновок аналізу за допомогою Velociraptor

Таким чином, результати аналізу підтверджують наявність ознак компрометації системи за допомогою шкідливого програмного забезпечення, характерного для діяльності *Raccoon Stealer*. Отримані дані можуть бути використані як доказ у подальшому цифровому розслідуванні та для розробки заходів реагування.

Виявлення Raccoon Stealer у середовищі Autopsy

У процесі аналізу цифрового образу диску було здійснено вивчення низки артефактів. Було проаналізовано журнали запуску програм, shellbags, cookies, завантаження, кеш браузера, історію переглядів, підозрілі файли, дампи, останні документи та потенційно зашифровані файли.

Виявлення запуску підозрілих програм (SRUDB.dat)

У файлі SRUDB.dat виявлено рядки, що вказують на запуск виконуваних файлів із підозрілих директорій. Наприклад, запуск процесів з каталогу \Program files (x86)\Microsoft\Edgeupdate\ із підозріло великою мережевою активністю, яка не характерна для системного оновлення, дозволяє припустити завантаження або ексфільтрацію даних.

Shellbags

Це артефакти, що зберігаються у файлі реєстру UsrClass.dat та відображають історію перегляду директорій у провіднику Windows. У межах цього аналізу shellbags не містять прямих вказівок на відкриття директорій на кшталт AppData, Temp, Roaming або запуску .exe з користувацьких шляхів, що не виключає, але й не підтверджує ручну взаємодію з інфікованими файлами.

Cookies з акаунтів Google

У збережених cookies браузера знайдено записи з домену accounts.google.com, включаючи ключі «LSID», «__Host-GAPS», що потенційно дають змогу відновити сесію доступу. Це є типовою ціллю Raccoon Stealer, який краде cookies для обходу автентифікації.

Веб-історія та Google Drive

Історія веб-перегляду свідчить про відвідування Google Drive та акаунтів Google 8 травня 2025 року – саме в період, коли активність ШПЗ вже була зафіксована. Це створює вікно можливості для крадіжки файлів або доступу до хмарного сховища.

Дампи пам'яті та підозрілі файли

У директорії ProgramData\Microsoft\Windows\WER було виявлено .dmp файли, створені після запуску підозрілих виконуваних файлів. Це може свідчити

про нестабільність ШПЗ або спроби приховування діяльності через виклик системних помилок.

Підозра на зашифровані файли

Файл mpenginedb.db був позначений як файл із високою ентропією (7.98), що вказує на можливе шифрування або упаковку даних. Можливо, він був використаний як контейнер для краденої інформації або конфігурації Raccoon Stealer.

Висновки аналізу за допомогою Autopsy

На основі виявлених артефактів – зокрема, завантаженого шкідливого архіву, cookies з акаунтів Google, дампів пам’яті, історії браузера та інших слідів – можна зробити обґрунтований висновок про активність Raccoon Stealer на досліджуваній системі.

3.3 Загальні рекомендації щодо виявлення активності шкідливого забезпечення Raccoon Stealer на кінцевих точках

На основі проведеного дослідження, присвяченого виявленню активності шкідливого програмного забезпечення Raccoon Stealer на кінцевих точках, сформульовано низку загальних рекомендацій щодо підвищення рівня безпеки інформаційної інфраструктури організації. Надані рекомендації охоплюють аспекти технічного моніторингу, організаційного управління безпекою, а також підвищення обізнаності працівників. Особливу увагу приділено використанню інструментів цифрової криміналістики та впровадженню процедур реагування на інциденти безпеки.

Рекомендації щодо технічного моніторингу та виявлення ШПЗ

Для ефективного виявлення активності подібної до Raccoon Stealer, організаціям необхідно впровадити постійний моніторинг кінцевих точок за допомогою спеціалізованих інструментів цифрової криміналістики та агентів збору артефактів системи.

Одним із таких рішень є Velociraptor, який дає змогу в режимі реального часу збирати артефакти системи Windows та виконувати запити, що дають змогу виявити підозрілу активність, пов’язану із запуском несанкціонованих процесів, модифікацією реєстру або нестандартними мережевими підключеннями [26]. Крім того, доцільно впроваджувати такі заходи:

розгортання EDR-рішень, які забезпечують поглиблений аналіз поведінки системи та здатні виявляти шкідливу активність на основі поведінкових ознак;

аналіз мережевого трафіку за допомогою Wireshark або інших NIDS-рішень, що дозволяє фіксувати витоки даних, типові для Raccoon Stealer, наприклад, HTTP POST-запити до зовнішніх C2-серверів;

регулярне збереження образів систем для подальшого криміналістичного аналізу за допомогою таких засобів, як Autopsy, з метою відтворення послідовності подій та виявлення джерела інфекції.

Важливо, щоб усі зібрані лог-файли, артефакти браузера, ключі реєстру та інші цифрові сліди зберігалися відповідно до стандартів доказової бази (наприклад, ISO/IEC 27037:2012) для забезпечення юридичної валідності отриманих доказів [27].

Організаційні заходи та політика безпеки

Наявність чітко визначених політик інформаційної безпеки є передумовою ефективного реагування на інциденти. У контексті загроз типу Raccoon Stealer, організація має впровадити наступні підходи[28]:

розробка політик доступу та контролю привілеїв. Багато зразків ШПЗ запускаються від імені користувача з підвищеними правами. Мінімізація прав доступу значно ускладнює роботу таких загроз;

заборона використання застарілого програмного забезпечення. Raccoon Stealer активно експлуатує вразливості браузерів, поштових клієнтів та інших клієнтських додатків. Ретельне оновлення ПЗ і операційної системи значно знижує ризики зараження;

реалізація процесу регулярного аудиту безпеки. Періодичні перевірки систем на наявність підозрілих змін у файлах, налаштуваннях або журналах подій допомагають виявляти шкідливу активність на ранніх етапах;

резервне копіювання критично важливих даних. Враховуючи, що деякі модифікації Raccoon Stealer можуть призводити до пошкодження або втрати даних, регулярне резервне копіювання є необхідним елементом стратегії відновлення.

Підвищення обізнаності персоналу

Незважаючи на технічні заходи захисту, людський фактор залишається одним із найуразливіших елементів інформаційної безпеки. Шкідливе програмне забезпечення часто проникає до системи через фішингові електронні листи, шкідливі вкладення або атаки соціального характеру. У зв’язку з цим рекомендується:

проводити регулярні тренінги з кібергігієни, зокрема на тему фішингу, підозрілих вкладень, правил користування корпоративною електронною поштою тощо;

імітаційні атаки (phishing simulations) для виявлення слабких місць серед працівників та подальшого їх усунення шляхом цільового навчання;

розробка простих і зрозумілих інструкцій дій у разі підозри на зараження системи, які допоможуть уникнути паніки, а також збережуть цінні цифрові сліди для подальшого аналізу.

Реагування на інциденти та цифрова криміналістика

Важливим аспектом забезпечення безпеки є не лише виявлення та попередження загроз, а й здатність ефективно реагувати на інциденти. Як показало дослідження, інструменти цифрової криміналістики (Velociraptor, Autopsy) дозволяють не лише фіксувати факт зараження, а й відновлювати повну хронологію подій [29]. Рекомендації:

визначити чітку процедуру реагування на інциденти, включаючи етапи ідентифікації, ізоляції, ліквідації, відновлення та документування;

створити команду реагування на інциденти безпеки , яка включатиме як технічних фахівців, так і представників юридичної та адміністративної сфер;

використання чек-листів при аналізі заражених систем, що охоплюють ключові точки збору артефактів: автозапуск, браузери, системні логи, мережеві підключення, зміни в реєстрі тощо;

забезпечення доказовості цифрових слідів згідно з принципами ланцюга зберігання доказів, що є критичним у разі подальших юридичних дій.

Висновки рекомендацій

Реалізація наведених рекомендацій дозволяє створити багаторівневу систему захисту, що враховує як технічні, так і організаційні аспекти. У випадку із загрозами, подібними до Raccoon Stealer, важливо забезпечити раннє виявлення, оперативне реагування та можливість проведення ретроспективного аналізу інцидентів. Комплексний підхід, що охоплює інструменти цифрової криміналістики, автоматизацію процесів моніторингу, навчання персоналу та підтримку нормативних політик, є запорукою надійного захисту інформаційних активів організації.

ВИСНОВКИ

У межах виконання бакалаврської кваліфікаційної роботи було проведено дослідження процесів виявлення активності шкідливого програмного забезпечення типу інфостілер на прикладі загрози Raccoon Stealer. З огляду на стрімке зростання кількості атак з використанням такого типу ШПЗ, дослідження набуло особливої актуальності, а результати можуть бути корисними як для аналітиків з кібербезпеки, так і для розробників систем захисту.

На основі аналізу літературних джерел та звітів провідних аналітичних центрів було встановлено, що інфостілери становлять одну з найбільш динамічно зростаючих категорій ШПЗ. Вони застосовуються для викрадення облікових даних, фінансової інформації, вмісту криптогаманців, історії браузера, сеансів веб-застосунків та інших персональних даних. Особливу небезпеку становить використання моделі Malware-as-a-Service, яка зробила такі загрози доступними для некваліфікованих зловмисників.

У роботі було детально розглянуто архітектуру та особливості дії Raccoon Stealer, який активно розвивався упродовж останніх років і став одним із найпоширеніших прикладів інфостілера у відкритих джерелах. Проведено порівняльний аналіз версій Raccoon Stealer, включаючи специфіку функціональності, канали комунікації з C2-сервером, методи крадіжки інформації та способи уникнення виявлення.

На основі вивчення теоретичних аспектів виявлення активності ШПЗ було класифіковано методи виявлення: сигнатурні, евристичні, поведінкові, а також засновані на машинному та глибокому навчанні. Встановлено, що кожен із методів має власні переваги і недоліки, однак найбільш ефективним є гібридний підхід, який поєднує декілька підходів одночасно. Зазначено важливість використання Threat Intelligence, систем SIEM, XDR, SOAR, а також застосування постійного моніторингу та інструментів цифрової криміналістики.

Проведено поглиблений аналіз трьох ключових інструментів, які використовуються в практиці цифрової криміналістики:

Velociraptor – рішення для збору та аналізу артефактів із кінцевих точок, з можливістю формування VQL-запитів, масштабованого моніторингу та інтеграції з іншими платформами;

Autopsy – універсальний графічний інструмент для криміналістичного аналізу цифрових носіїв, що забезпечує автоматичний збір, індексацію, аналіз і збереження артефактів;

Wireshark – засіб для аналізу мережевого трафіку, який дозволяє виявляти характерні шаблони взаємодії між інфікованими машинами та серверами управління.

У рамках практичного дослідження було змодельовано сценарій атаки за участю Raccoon Stealer. Проведено аналіз мережевої активності за допомогою Wireshark, який дозволив виявити підозрілий трафік, DNS-запити до аномальних доменів, спроби передачі даних на C2-сервер, а також аналіз з'єднань, що завершувались RST-пакетами. За допомогою Velociraptor були зібрані критичні артефакти: Prefetch-записи, ключі реєстру, Task Scheduler, NTUser.dat, AppCompatCache та інші, які дали змогу простежити активність зараженого файлу. Autopsy забезпечив додаткову перевірку образу диска, де були виявлені файли із залишковими слідами шкідливої активності, включаючи ознаки шифрування, файли cookie, активність браузерів та історію користувача.

В результаті роботи:

проведено аналіз Raccoon Stealer, його архітектури, еволюції та функціональності;

сформовано системну класифікацію методів виявлення активностей ШПЗ та визначено їх релевантність до задач цифрової криміналістики;

виконано поглиблений аналіз інструментів Velociraptor, Autopsy та Wireshark, а також протестовано їх на прикладі моделювання реального інциденту;

розроблено та задокументовано алгоритм дій для практичного виявлення інфостілера в інформаційній системі.

Комплексне використання наведених засобів дозволило виявити взаємодоповнюючі результати, що підтвердило доцільність інтегрованого підходу в практиці реагування на інциденти та цифрового аналізу.

Отримані результати можуть бути використані в навчальному процесі для підготовки фахівців з інформаційної безпеки, зокрема при вивченні цифрової криміналістики, аналізу інцидентів та дослідження шкідливого програмного забезпечення. Представлені інструменти, методи та алгоритми можна застосовувати як у лабораторних роботах, так і в тренінгах з практичного аналізу загроз.

ПЕРЕЛІК ПОСИЛАНЬ

1. Hao, S., Thomas, K., Bursztein, E., Invernizzi, L., Pearce, P., Yadav, S., McCoy, D., Paxson, V. Malware Finances and Operations: a Data-Driven Study of the Value Chain for Infections and Compromised Access // ACM Transactions on Privacy and Security (TOPS). 2023. Vol. 26, No. 4. P. 1–30. URL: https://dl.acm.org/doi/10.1145/3600160.3605047 (дата звернення: 14.05.2025).
2. Infostealer – Wikipedia. URL: https://en.wikipedia.org/wiki/Infostealer (дата звернення: 14.05.2025).
3. Acronis. Raccoon Stealer – A Popular and Dangerous Threat. URL: https://www.acronis.com/en-us/cyber-protection-center/posts/raccoon-stealer-a-popular-and-dangerous-threat/?utm_source=chatgpt.com (дата звернення: 14.05.2025).
4. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. Gaithersburg, MD: National Institute of Standards and Technology, 2007. 127 p.
5. Sikorski M., Honig A. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. San Francisco: No Starch Press, 2012. 800 p.
6. Szor P. The Art of Computer Virus Research and Defense. Boston: Addison-Wesley, 2005. 744 p
7. Casey E. Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet. 3rd ed. Amsterdam: Academic Press, 2011. 807 p.
8. Hardy W., Chen L., Hou D., Ye Y. DL4MD: A Deep Learning Framework for Smart Malware Detection. In: Proceedings of the International Conference on Data Mining Workshops (ICDMW). 2016. P. 61–68.
9. MITRE ATT&CK Framework. URL: https://attack.mitre.org/ (дата звернення: 14.05.2025).
10. Zetter K. Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon. New York: Crown Publishing, 2014. 432 p.

11. Velociraptor. Official Documentation. 2024. URL: https://docs.velociraptor.app (дата звернення: 14.05.2025).
12. Carvey H. Windows Forensic Analysis Toolkit: Advanced Analysis Techniques for Windows 10. 4th ed. Burlington: Syngress, 2018. 464 p.
13. Carrier B. Autopsy Forensic Browser User Guide. Version 4.20.0. 2023. URL: https://www.sleuthkit.org/autopsy/docs/user_guide.pdf (дата звернення: 14.05.2025).
14. Cohen M. Practical Forensic Imaging: Securing Digital Evidence with Linux Tools. Waltham: Syngress, 2020. 402 p.
15. Chappell L. Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide. 3rd ed. Clearwater: Protocol Analysis Institute, 2022. 828 p.
16. Malwarebytes Labs. Raccoon Stealer Analysis Report. 2022. URL: https://www.malwarebytes.com/blog/news/2022/10/raccoon-stealer-analysis (дата звернення: 14.05.2025).
17. Bejtlich R. The Practice of Network Security Monitoring: Understanding Incident Detection and Response. San Francisco: No Starch Press, 2013. 376 p.
18. Velociraptor GitHub repository. URL: https://github.com/Velocidex/velociraptor (дата звернення: 14.05.2025).
19. Velociraptor VQL Guide. URL: https://docs.velociraptor.app/docs/vql/ (дата звернення: 14.05.2025).
20. DFIR Report. URL: https://thedfirreport.com (дата звернення: 14.05.2025).
21. Carrier, B. “The Sleuth Kit & Autopsy.” https://www.sleuthkit.org/autopsy/
22. Marrington, A., et al. “A Model for Structuring Digital Forensic Investigations.”
23. Ligh, M. H., et al. “Malware Analyst's Cookbook.”
24. Wireshark Documentation Team. "Wireshark GUI Overview", 2023.
25. Wireshark Documentation Team. "Display Filters Reference", 2023.
26. Wireshark Development Team. "Supported Protocols List", 2023.
27. Velociraptor. Digital Forensics Platform. 2024. URL: https://docs.velociraptor.app/ (дата звернення: 14.05.2025).

28. ISO/IEC 27037:2012. Information technology – Security techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence. Geneva: ISO, 2012. 46 p.
29. Mandia K., Prosise C., Pepe M. Incident Response and Computer Forensics. 3rd ed. New York: McGraw-Hill, 2014. 544 p.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)