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

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Рекомендації щодо виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor»

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

_____ Дмитро АРХИПЕНКО
(підпис)

Виконав: здобувач вищої освіти групи БСД-43
АРХИПЕНКО Дмитро
(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

ВСТУП

Актуальність дослідження. Виявлення шкідливого програмного забезпечення на кінцевих точках організації є актуальною проблемою в наш час. По-перше, сучасне шкідливе програмне забезпечення стає дедалі витонченішим і здатне маскувати свою присутність, використовуючи безфайлові атаки, шифрування коду та інші методи уникнення виявлення традиційними антивірусними системами. По-друге, кінцеві точки мають великий розподіл і різноманітність пристроїв, що ускладнює уніфіковане впровадження ефективних засобів моніторингу і аналізу. По-третє, велика кількість легітимних процесів і активності на кінцевих точках призводить до високого рівня шуму в даних, що ускладнює ідентифікацію справжніх загроз та збільшує ризик хибнопозитивних спрацьовувань. Все це створює значні виклики для побудови системи своєчасного та точного виявлення шкідливого програмного забезпечення, яка б ефективно протидіяла зростаючим кіберзагрозам і захищала інформаційну інфраструктуру організації.

Сучасні системи виявлення шкідливого програмного забезпечення на кінцевих точках, зокрема рішення на базі платформи Velociraptor, забезпечують постійний моніторинг та збір детальної інформації про поведінку системи в реальному часі. Velociraptor дозволяє виявляти аномалії та потенційні загрози, що дає змогу своєчасно реагувати на інциденти кібербезпеки. Завдяки гнучкій мові запитів VQL та можливості кастомізації сценаріїв, ця платформа забезпечує глибокий аналіз та цифрову криміналістику, що значно підвищує ефективність виявлення навіть складних та прихованих атак.

Використання Velociraptor допомагає організаціям покращити видимість активності кінцевих точок, автоматизувати збір і аналіз артефактів, а також впровадити адаптивні політики безпеки, що відповідають актуальним викликам кіберзагроз. Це дозволяє забезпечити комплексний захист кінцевих точок, запобігти

поширенню шкідливого програмного забезпечення та мінімізувати потенційні збитки для інформаційної інфраструктури організації.

Вищевикладене обґрунтовує актуальність теми даної кваліфікаційної роботи, яка зосереджена на дослідженні методів та засобів виявлення шкідливого програмного забезпечення на кінцевих точках організації з використанням платформи Velociraptor.

Об'єкт дослідження – виявлення шкідливого програмного забезпечення на кінцевих точках організації.

Предмет дослідження – методи та засоби виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor.

Мета роботи запропонувати порядок виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor та розробити рекомендації щодо його застосування.

Наукові завдання:

дослідити сутність проблеми виявлення шкідливого програмного забезпечення на кінцевих точках організації;

проаналізувати підходи до виявлення шкідливого програмного забезпечення на кінцевих точках організації;

проаналізувати існуючі рішення із виявлення шкідливого програмного забезпечення на кінцевих точках організації;

проаналізувати методи та засоби виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor;

запропонувати порядок виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor;

розробити рекомендацій щодо виявлення шкідливого програмного забезпечення на кінцевих точках організації.

Методи дослідження – опрацювання літератури за темою дослідження, аналіз технічної документації та міжнародних стандартів, практичне тестування програмного забезпечення, порівняльний аналіз отриманих результатів.

Практичне значення одержаних результатів: розроблено порядок щодо виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor та розроблено рекомендації щодо його застосування.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА КІНЦЕВИХ ТОЧКАХ ОРГАНІЗАЦІЇ

1.1 Актуальність проблеми виявлення шкідливого програмного забезпечення на кінцевих точках організації

У сучасному цифровому середовищі ШПЗ є однією з найпоширеніших та найнебезпечніших кіберзагроз для організацій. Кінцеві точки – персональні комп'ютери, ноутбуки, мобільні пристрої, IoT-компоненти – відіграють роль основних вхідних каналів в корпоративну мережу, і саме через них найчастіше реалізується проникнення ШПЗ. Згідно з International Data Corporation, понад 70% успішних атак на організації розпочинаються саме з компрометації кінцевих точок [1].

Під ШПЗ мається на увазі програмне забезпечення, яке за умови запуску може завдати шкоди пристрою різними способами, зокрема – призвести до [6]:

блокування пристрою та його непридатності для використання;

крадіжки, видалення або шифрування даних;

використання пристрою для атак на інші пристрої;

отримання кіберзловмисниками інформації щодо облікових даних, які дозволяють дістати доступ до систем або служб, якими ви користуєтесь;

використання для незаконного майнингу криптовалюти на зараженому пристрої;

використання платних послуг на основі скомпрометованих даних, тощо.

Щодня системи кібербезпеки виявляють приблизно 560 000 нових загроз з боку ШПЗ, що підкреслює невпинний, автоматизований характер сучасної кіберзлочинності та масштаби глобальної вразливості. У 2025 році очікується швидке зростання кількості заражень ~6,5 мільярда через адаптивне ШПЗ, що використовує штучний інтелект та технології діпфейків [3].

Зростання кількості та складності ШПЗ ускладнює процес його виявлення та нейтралізації. Традиційні методи захисту, такі як антивірусні програми, стають менш ефективними перед новими видами загроз, що використовують складні техніки уникнення виявлення.

Хоча існує багато різних варіацій ШПЗ, найімовірніше, зіткнутися з такими типами шкідливих програм [2]:

Програми-вимагачі (Ransomware) – це програмне забезпечення, яке використовує шифрування, щоб блокувати доступ жертви до своїх даних до сплати викупу. Організація-жертва частково або повністю не може працювати, доки не сплатить викуп, але немає гарантії, що оплата призведе до отримання необхідного ключа розшифрування або що наданий ключ розшифрування працюватиме належним чином. У 2023 році місто Балтимор постраждало від програми-вимагача під назвою RobbinHood , яке на тижні зупинило всю діяльність міста, включаючи збір податків, передачу майна та урядову електронну пошту. Ця атака коштувала місту понад 18 мільйонів доларів. Такий самий тип ШПЗ був використаний проти міста Атланта у 2018 році, що призвело до збитків у розмірі 17 мільйонів доларів;

Безфайлове ШПЗ (Fileless Malware) спочатку нічого не встановлює, натомість воно вносить зміни до файлів, які є власними для операційної системи, таких як PowerShell або WMI. Оскільки операційна система розпізнає відредаговані файли як легітимні, безфайлова атака не виявляється антивірусним програмним забезпеченням, а оскільки ці атаки є прихованими, вони у десять разів успішніші за традиційні атаки ШПЗ. Прикладом безфайлового ШПЗ є Astaroth. Astaroth – це кампанія безфайлового ШПЗ, яка розсилала користувачам посилання на файл ярлика .LNK. Коли користувачі завантажували файл, запускався інструмент WMIC разом з низкою інших легітимних інструментів Windows. Ці інструменти завантажували додатковий код, який виконувався лише в пам'яті, не залишаючи жодних доказів, які могли б виявити сканери вразливостей. Потім зловмисник завантажував і запускав трояна, який крав облікові дані та завантажував їх на віддалений сервер;

Шпигунське ПЗ (Spyware) збирає інформацію про діяльність користувачів без їхнього відома чи згоди. Це може включати паролі, PIN-коди, платіжну інформацію та неструктуровані повідомлення. Використання шпигунського ПЗ не обмежується браузером на комп'ютері: воно також може працювати в критично важливому додатку або на мобільному телефоні. Навіть якщо викрадені дані не є критично важливими, наслідки шпигунського ПЗ часто поширюються по всій організації, оскільки знижується продуктивність та підривається продуктивність;

Рекламне ПЗ (Adware) відстежує активність користувача в Інтернеті, щоб визначити, яку рекламу йому показувати. Хоча рекламне ПЗ схоже на шпигунське ПЗ, воно не встановлює жодного програмного забезпечення на комп'ютер користувача та не фіксує натискання клавіш. Небезпека рекламного ПЗ полягає в порушенні конфіденційності користувача – дані, отримані рекламним ПЗ, зіставляються з даними, отриманими явно чи приховано, про активність користувача в інших місцях Інтернету, та використовуються для створення профілю цієї особи, який включає інформацію про її друзів, що вона купила, куди подорожувала тощо. Ця інформація може бути передана або продана рекламодавцям без згоди користувача;

Троян (Trojan) маскується під бажаний код або ПЗ. Після завантаження нічого не підозрюючими користувачами троян може взяти під контроль системи жертв у зловмисних цілях. Трояни можуть ховатися в іграх, додатках або навіть патчах ПЗ, або ж вони можуть бути вбудовані у вкладення фішингових електронних листів. Emotet – це гарний приклад складного банківського трояну, який існує з 2014 року. З Emotet важко боротися, оскільки він уникає виявлення на основі сигнатур, є стійким та містить модулі розповсюдження, які допомагають йому поширюватися. Троян настільки поширений, що Міністерство внутрішньої безпеки США опублікувало про нього попередження , в якому зазначається, що усунення наслідків Emotet коштувало урядам штатів, місцевим, племінним та територіальним органам влади до 1 мільйона доларів за кожен інцидент;

Хробак (Worm) встановлюється в мережу через вразливості операційної системи. Він може отримати доступ кількома способами: через вбудовані в

програмне забезпечення бекдори, через ненавмисні вразливості ПЗ або через флеш-накопичувачі. Після встановлення хробака зловмисники можуть використовувати його для запуску DDoS-атак , крадіжки конфіденційних даних або проведення атак програм-вимагачів. Наприклад Stuxnet, ймовірно, був розроблений американськими та ізраїльськими спецслужбами з наміром зупинити ядерну програму Ірану. Він був впроваджений в іранське середовище через флеш-накопичувач. Оскільки середовище було ізольоване, його творці ніколи не думали, що Stuxnet покине мережу своєї цілі - але це сталося. Опинившись на волі, Stuxnet агресивно поширювався, але не завдав великої шкоди, оскільки його єдиною функцією було втручання в роботу промислових контролерів, які керували процесом збагачення урану;

Вірус (Virus) – це фрагмент коду, який вставляється в програму та виконується під час її запуску. Потрапивши в мережу, вірус може бути використаний для крадіжки конфіденційних даних, запуску DDoS-атак або атак програм-вимагачів. Вірус не може виконуватися або розмножуватися, якщо не запущено заражену ним програму. Ця залежність від основної програми відрізняє віруси від троянів, які вимагають від користувачів завантаження, та хробаків, які не використовують програми для запуску. Багато екземплярів ШПЗ можна розділити на кілька категорій: наприклад, Stuxnet – це хробак, вірус та руткіт;

Руткіт (Rootkit) – це програмне забезпечення, яке надає зловмисникам віддалений контроль над комп’ютером жертви з повними адміністративними правами. Руткіти можуть бути впроваджені в програми, ядра, гіпервізори або прошивки. Вони поширюються через фішинг, шкідливі вкладення, шкідливі завантаження та скомпрометовані спільні диски. Руткіти також можуть використовуватися для приховування інших шкідливих програм, таких як кейлогери;

Кейлогер (Keylogger) – це тип шпигунського програмного забезпечення, яке відстежує активність користувачів. Кейлогери мають законне використання; компанії можуть використовувати їх для моніторингу активності співробітників, а сім'ї можуть використовувати їх для відстеження онлайн-поведінки дітей. Однак,

якщо їх встановити зі зловмисною метою, кейлогери можуть бути використані для крадіжки паролів, банківської інформації та іншої конфіденційної інформації. Кейлогери можуть бути введені в систему за допомогою фішингу, соціальної інженерії або шкідливих завантажень. Кейлогер під назвою Olympic Vision використовувався для атак на бізнесменів зі США, Близького Сходу та Азії з метою компрометації ділової електронної пошти. Olympic Vision використовує методи фішингу та соціальної інженерії для зараження систем своїх цілей з метою крадіжки конфіденційних даних та шпигунства за діловими операціями. Кейлогер не є складним, але його можна придбати на чорному ринку за 25 доларів, тому він легкодоступний для зловмисників;

Боти/Ботнети (Bots/Botnets) – це програмний застосунок, який виконує автоматизовані завдання за командою. Він використовується для законних цілей, таких як індексація пошукових систем, але коли використовується для зловмисних цілей, він набуває форми саморозповсюджувального ШПЗ, яке може підключатися до центрального сервера. Зазвичай боти використовуються у великій кількості для створення ботнету, який являє собою мережу ботів, що використовуються для запуску широких потоків атак дистанційно керованих систем, таких як DDoS-атаки. Ботнети можуть стати досить розлогими. Наприклад, ботнет Mirai IoT налічував від 800 000 до 2,5 мільйона комп'ютерів;

ШПЗ для стирання даних (Wiper) – це тип ШПЗ з єдиною метою: стерти дані користувача та забезпечити їх неможливість відновлення. ШПЗ для стирання даних використовуються для виведення з ладу комп’ютерних мереж у державних або приватних компаніях у різних секторах. Зловмисники також використовують ШПЗ для стирання даних, щоб приховати сліди, що залишаються після вторгнення, послаблюючи здатність жертви реагувати. 15 січня 2022 року повідомлялося про використання набору ШПЗ під назвою WhisperGate проти українських цілей. За загальними повідомленнями, інцидент містив три окремі компоненти, розгорнуті одним і тим самим зловмисником, включаючи шкідливий завантажувач, який пошкоджує виявлені локальні диски, завантажувач на основі Discord та програму

для видалення файлів. Активність сталася приблизно в той самий час, коли було пошкоджено кілька веб-сайтів, що належать уряду України.

У підсумку, складність виявлення ШПЗ на кінцевих точках зумовлена рядом технологічних та методологічних викликів. Сучасне ШПЗ дедалі частіше використовує техніки уникнення виявлення, зокрема поліморфізм, метаморфізм, шифрування коду та запуск без створення файлів. Це ускладнює виявлення за допомогою традиційних сигнатурних антивірусів, які орієнтовані на фіксовані шаблони шкідливого коду. Особливо критичною є тенденція до зростання безфайлових атак, за даними CrowdStrike, 79% атак у 2024 році були безфайловими, це значне зростання у порівнянні з 40% у 2019 році (рисунок 1.1) [4].

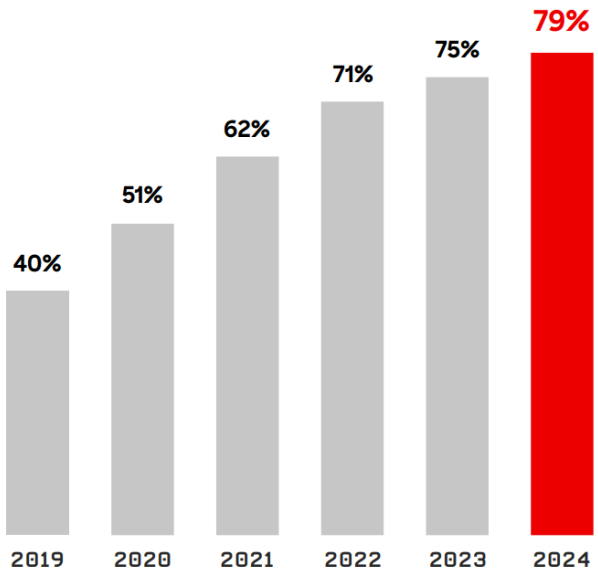

Рис. 1.1. Відсоток виявлень, які не містили ШПЗ, 2019-2024 рр. [4]

Таке ШПЗ не залишає артефактів на диску, діє через вбудовані інструменти Windows, а отже – обходить більшість систем сканування. Подібні кампанії, як-от Astaroth, демонструють здатність таких програм до прихованої та тривалої присутності у системі без виявлення. Крім того, високий ступінь модульності й спеціалізації сучасного ШПЗ дозволяє йому гнучко адаптуватися до умов конкретного середовища. Такі зразки як Emotet чи WhisperGate ілюструють, як ШПЗ може одночасно діяти як троян, кейлогер, бот або деструктивний агент. Ще одним ускладнюючим чинником є поява ШПЗ, орієнтованого на крадіжку

облікових даних та телеметрії, що діє приховано, не викликаючи явних аномалій у роботі системи. За звітом IBM X-Force, лише за 2024 рік кількість атак, пов'язаних з інфостілерами, зросла на 84% [5].

Загалом, високий рівень технологічної складності, здатність до маскування, швидка еволюція кодової бази та прихованість дій роблять виявлення ШПЗ на кінцевих точках однією з найскладніших задач сучасної кібербезпеки.

1.2 Аналіз підходів до виявлення шкідливого програмного забезпечення на кінцевих точках організації

Зіткнувшись зі все більш складними кіберзагрозами, організації повинні використовувати багатогранний підхід для ефективного виявлення та зменшення впливу ШПЗ. Традиційні методи, хоча й є фундаментальними, часто не здатні протистояти складним стійким загрозам та експлойтам нульового дня. Тому інтеграція різних методів виявлення має вирішальне значення для підвищення рівня безпеки та стійкості. Ці методи відрізняються методологією, обсягом та ефективністю, але разом утворюють багаторівневий захисний механізм. Нижче представлено основні підходи, що використовуються для виявлення та зменшення впливу ШПЗ на рівні кінцевих точок (рисунок 1.2) [7]:

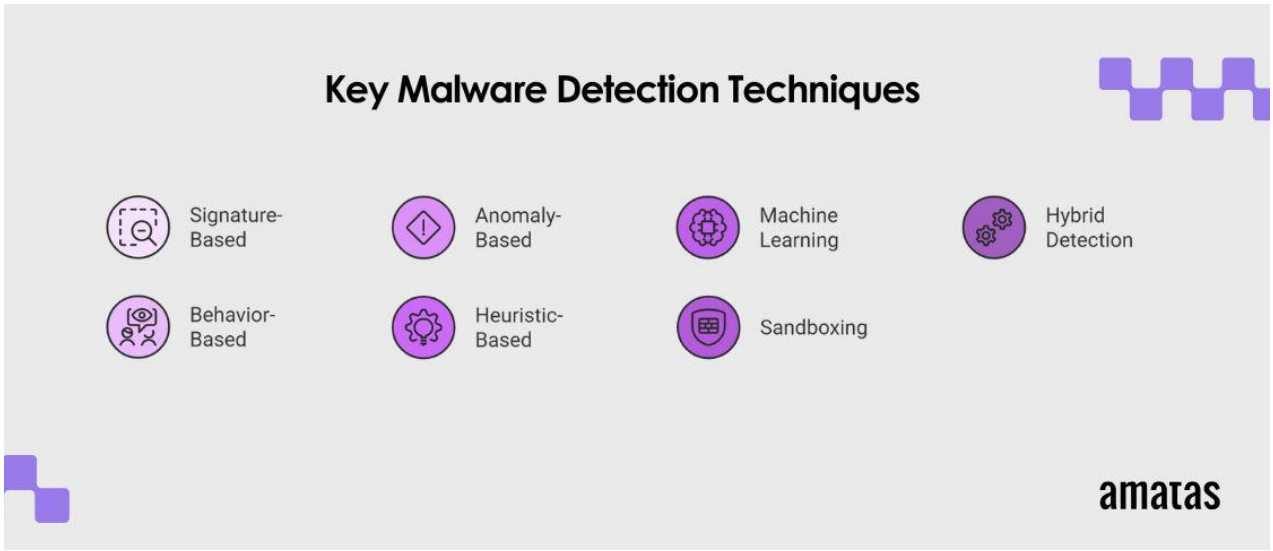

Рис. 1.2. Основні методи виявлення ШПЗ [7]

Виявлення на основі сигнатур. Це як використання «розшукуваного плаката» для виявлення загроз. Інструменти безпеки ведуть базу даних відомих сигнатур ШПЗ (унікальні шаблони або характеристики шкідливих програм). Коли файл відповідає одній із цих сигнатур, він позначається як шкідливий. Список дозволених файлів може доповнити цей підхід, гарантуючи, що запускатимуться лише попередньо схвалені, перевірені програми, додаючи ще один рівень захисту;

Виявлення на основі поведінки. Замість пошуку конкретних сигнатур, цей метод зосереджується на виявленні шкідливої поведінки. Якщо програма раптово отримує доступ до конфіденційних файлів або змінює системні налаштування без дозволу, вона позначається як потенційно шкідлива;

Виявлення на основі аномалій. Цей метод визначає відхилення від нормальної поведінки системи для виявлення потенційних загроз. Наприклад, якщо потік даних або використання ресурсів відхиляється від встановлених базових значень, це може свідчити про зловмисну діяльність;

Евристичне виявлення. Цей метод застосовує правила або алгоритми для виявлення підозрілої поведінки або шаблонів файлів, навіть якщо ШПЗ нове. Наприклад, він може позначити файл, який намагається виконати код незвичним чином, як потенційно шкідливий;

Машинне навчання та виявлення на основі ШІ. Ця передова методика використовує ШІ для вивчення величезних обсягів даних про ШПЗ та нормальну поведінку системи. З часом вона навчається виявляти закономірності або аномалії, що вказують на шкідливу активність, навіть для нових типів ШПЗ;

Методи пісочниці. Підозрілий файл запускається в безпечному, ізольованому середовищі «пісочниці», де він не може завдати шкоди самій комп’ютерній системі. Пісочниця відстежує поведінку файлу, щоб побачити, чи не робить він чогось шкідливого, наприклад, спроби видалити файли або викрасти дані. Пісочниця наступного покоління покращує це, динамічно аналізуючи файли на рівні процесора, а не на рівні програми, що робить її стійкою до ШПЗ, яке намагається уникнути виявлення. Цей глибший аналіз виявляє складні ризики, такі як

низькорівневі експлойти та атаки нульового дня, які традиційна пісочниця може пропустити;

Гібридні методи виявлення. Гібридні технології виявлення поєднують два або більше методів виявлення, таких як виявлення на основі сигнатур та підходи, засновані на поведінці, для забезпечення більш надійного та комплексного рішення безпеки. Наприклад, гібридна система може спочатку сканувати на наявність відомих сигнатур ШПЗ та, якщо збігів не знайдено, аналізувати поведінку файлу на наявність підозрілої активності. Використовуючи сильні сторони кількох методів, гібридне виявлення може виявляти ширший спектр загроз, включаючи як добре відомі загрози, так і нові шкідливі програми, мінімізуючи водночас слабкі сторони окремих методів. Такий багаторівневий підхід підвищує точність виявлення та зміцнює загальний захист у сфері кібербезпеки.

Зважаючи на різноманіття методів виявлення ШПЗ, важливо не лише розуміти принципи їхньої роботи, але й оцінити їхню ефективність у практичному застосуванні. Кожен підхід має свої переваги і недоліки, які визначають доцільність його використання в конкретному середовищі. Деякі методи забезпечують високу точність виявлення, але потребують значних обчислювальних ресурсів, тоді як інші є менш ресурсоємними, однак мають вищу ймовірність помилкових спрацювань. У таблиці 1.1 наведено порівняльний аналіз основних методів виявлення ШПЗ, де охарактеризовано їх сильні та слабкі сторони, а також вказано типові інструменти й платформи, в яких вони реалізуються.

Таблиця 1.1.

Порівняння різних методів виявлення ШПЗ

Метод виявлення ШПЗ	Сильні сторони	Слабкі сторони	Інструменти, що використовують метод
Виявлення на основі сигнатур	Добре працює з відомими варіантами ШПЗ, такими як старіші віруси або черв'яки.	Не може виявити нове або модифіковане ШПЗ, включаючи поліморфне.	Антивірусне програмне забезпечення, таке як Norton або McAfee.

Продовження таблиці 1.1.

Порівняння різних методів виявлення ШПЗ

Метод виявлення ШПЗ	Сильні сторони	Слабкі сторони	Інструменти, що використовують метод
Виявлення на основі поведінки	Ефективний для виявлення шкідливих програм нульового дня та нових і невідомих шкідливих програм, оскільки не залежить від попередніх знань про ШПЗ.	Може призвести до хибнопозитивних результатів, якщо позначено законну поведінку.	Рішення, такі як InsightIDR від Rapid7.
Виявлення на основі аномалій	Ефективний проти складного ШПЗ, яке імітує легітимну діяльність.	Потрібні базові дані та можуть призвести до хибнопозитивних результатів.	DarkTrace та Splunk.
Евристичне виявлення	Виявляє нові варіанти ШПЗ, які традиційні методи виявлення на основі сигнатур можуть пропустити.	Може не виявляти дуже складне ШПЗ та іноді може помилково ідентифікувати легітимні файли як загрози.	Bitdefender Total Security, ESET NOD32 Antivirus, Trend Micro Internet Security.
Машинне навчання та виявлення на основі ШІ	Виявляє невідомі та постійно мінливі загрози. З часом стає розумнішим, навчаючись на нових даних.	Для ефективної роботи потрібен великий обсяг даних та обчислювальна потужність.	Такі інструменти, як Darktrace та SentinelOne, використовують штучний інтелект та машинне навчання для адаптації до нових загроз.

Продовження таблиці 1.1.

Порівняння різних методів виявлення ШПЗ

Метод виявлення ШПЗ	Сильні сторони	Слабкі сторони	Інструменти, що використовують метод
Методи пісочниці	Надає детальний огляд дій шкідливого програмного забезпечення без ризику для хост-системи.	Ресурсомісткі та деякі складні шкідливі програми можуть виявити, що вони перебувають у «пісочниці», і поводитися нормально, щоб уникнути виявлення.	Аналіз шкідливого програмного забезпечення FireEye, Cuckoo Sandbox, Palo Alto Networks WildFire.
Гібридні методи виявлення	Поєднуючи кілька методів виявлення, підхід забезпечує високу точність, дозволяє ідентифікувати як відомі, так і нові загрози та знижує ризик хибнопозитивних і хибнонегативних спрацювань.	Висока складність впровадження, значне споживання ресурсів і потреба в кваліфікованому персоналі обумовлені інтеграцією різних методів і великим обсягом даних для аналізу.	Cisco Secure Endpoint, Velociraptor.

1.3 Аналіз існуючих рішень із виявлення шкідливого програмного забезпечення на кінцевих точках організації

У контексті зростаючої складності та різноманіття ШПЗ, а також враховуючи обмеження традиційних методів його виявлення, особливу увагу заслуговують сучасні технічні рішення, що реалізують принципи багаторівневого та адаптивного захисту кінцевих точок. На практиці ефективна протидія ШПЗ вимагає використання інструментів, які поєднують кілька підходів – від сигнатурного аналізу до поведінкової аналітики та цифрової криміналістики. Нижче буде розглянуто найбільш відомі та широко застосовувані системи виявлення ШПЗ, які використовуються в корпоративному середовищі. Аналіз охоплює як комерційні

рішення класу EDR/XDR, так і інструменти з відкритим кодом, що дозволяють здійснювати глибоке розслідування інцидентів. Зокрема, увага приділена платформам CrowdStrike Falcon, Microsoft Defender for Endpoint, SentinelOne Singularity та Velociraptor – кожне з яких реалізує власну стратегію виявлення, фіксації та реагування на загрози.

CrowdStrike Falcon – це одна з провідних платформ корпоративного класу для виявлення, запобігання та реагування на загрози безпеці, зокрема шкідливого програмного забезпечення на кінцевих точках. Продукт розроблено з урахуванням найвищих вимог сучасного кіберзахисту та орієнтовано на протидію як масовим, так і цілеспрямованим атакам, включаючи безфайлове ШПЗ, APT-кампанії та атаки, які використовують легітимні інструменти системи. Завдяки своїй хмарній архітектурі Falcon забезпечує централізоване управління захистом кінцевих точок у масштабах великої організації, незалежно від її географічної структури чи кількості підключених пристроїв (рисунок 1.3) [8].

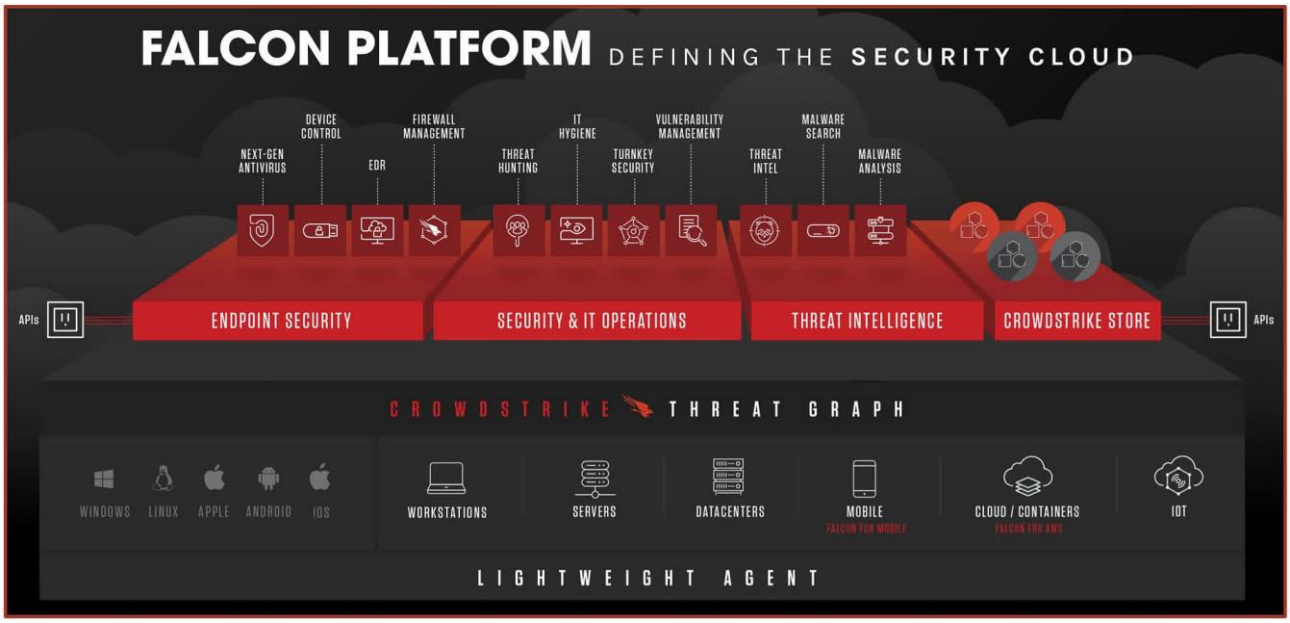

Рис. 1.3. Огляд CrowdStrike Falcon [22]

Ключовою особливістю CrowdStrike Falcon є поєднання кількох механізмів виявлення: сигнатурного, поведінкового, AI-орієнтованого та заснованого на глобальній базі розвідки загроз. Система постійно відстежує активність процесів на кінцевих точках, аналізує її у реальному часі та корелює з відомими шаблонами

загроз, що зберігаються у хмарній базі. Falcon активно використовує машинне навчання для побудови поведінкових профілів систем і виявлення аномалій, які можуть свідчити про наявність ШПЗ, навіть якщо воно не має жодної відомої сигнатури. Особливо ефективним є виявлення fileless malware – ШПЗ, яке не залишає слідів на диску, а діє виключно в оперативній пам'яті [8].

Окрім виявлення, платформа забезпечує потужний функціонал реагування: можливість ізолювати скомпрометовану кінцеву точку, зупинити виконання підозрілих процесів або автоматично запустити сценарії аналізу. У разі серйозного інциденту адміністратори мають змогу скористатися детальною хронологією подій, журналами активності, телеметрією процесів, що дозволяє провести поглиблене розслідування. Рішення також підтримує інтеграцію з іншими платформами (SIEM, SOAR), що дозволяє вбудовувати CrowdStrike у широку систему корпоративного захисту [8].

Завдяки своїм можливостям Falcon є одним із найефективніших інструментів для організацій, які стикаються з загрозами високого рівня складності. Він дозволяє не лише швидко виявляти вже відомі зразки ШПЗ, але й виявляти нові загрози на основі поведінки, що особливо важливо в умовах швидко змінюваного ландшафту кібербезпеки. За оцінкою експертів у CrowdStrike Global Threat Report 2025, Falcon забезпечив високий рівень детектування навіть у випадках безсигнатурних атак, які склали 79% усіх виявлень у 2024 році [4].

Microsoft Defender for Endpoint – це багатофункціональна платформа корпоративного захисту кінцевих точок, що є складовою екосистеми безпеки Microsoft 365. Її головна мета – виявлення, дослідження та нейтралізація загроз, зокрема ШПЗ, на основі глибокої інтеграції з операційною системою Windows. Як стандартний компонент сучасних версій Windows, Defender for Endpoint забезпечує базовий рівень захисту «з коробки», який може бути масштабований до повноцінного EDR-рішення в рамках Microsoft Security Stack (рисунок 1.4) [9].

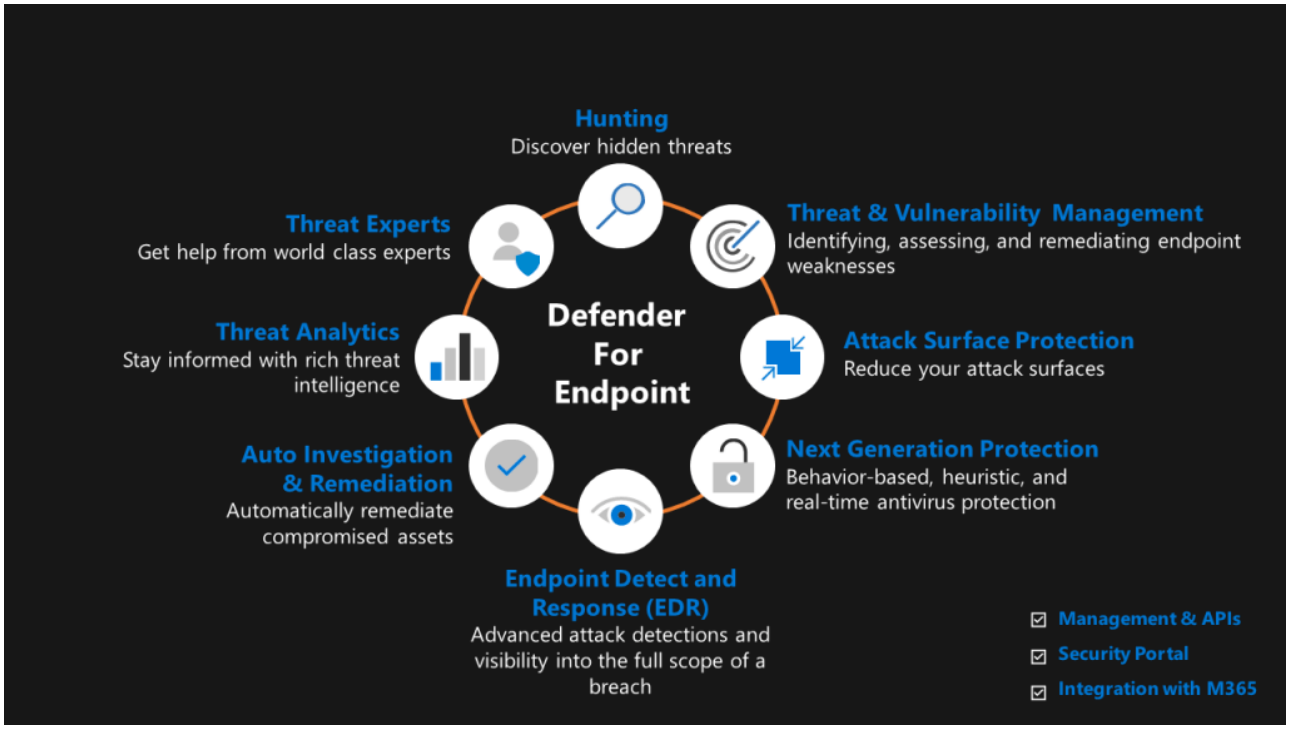

Рис. 1.4. Огляд Microsoft Defender for Endpoint [23]

Однією з головних функціональних переваг системи є використання поведінкових сигнатур, хмарної аналітики та ізольованого аналізу (sandbox) для виявлення як відомих, так і нових загроз. Defender автоматично реєструє та корелює події з кінцевих точок, формуючи цілісну картину активності у системі. Інструмент також підтримує автоматизовані сценарії реагування: наприклад, при виявленні підозрілої активності може бути автоматично ізольовано відповідний процес або заблоковано доступ до мережевих ресурсів. Інтеграція з Microsoft Threat Experts відкриває доступ до глобальної аналітики загроз та кваліфікованої експертної підтримки. Крім того, система включає модуль виявлення вразливостей (Threat & Vulnerability Management), що дозволяє не тільки виявляти активні атаки, але й запобігати їм, закриваючи відомі слабкі місця у програмному забезпеченні [9].

У тестах незалежної лабораторії AV-TEST, Defender for Endpoint регулярно отримує найвищі оцінки за рівень захисту (6.0/6.0), що підтверджує його ефективність проти широкого спектра шкідливих програм [10]. Завдяки поєднанню автоматизації, широкої інтеграції, ефективного виявлення та глибокої аналітики, Microsoft Defender for Endpoint залишається одним із найбільш збалансованих

рішень для виявлення й боротьби з ШПЗ на кінцевих точках, особливо у середовищах, що базуються на технологіях Microsoft [9].

SentinelOne Singularity – це сучасна платформа захисту кінцевих точок, яка поєднує традиційні антивірусні функції (EPP) з можливостями розширеного виявлення та реагування (EDR). Унікальність рішення полягає в орієнтації на повну автономність – платформа здатна виявляти, класифікувати та нейтралізувати ШПЗ без участі аналітика, що суттєво підвищує швидкість реагування на загрози та зменшує навантаження на команди безпеки. Система використовує локальні моделі машинного навчання для виявлення аномалій на рівні файлів, мережі, реєстру та пам’яті, що дозволяє виявляти не лише відомі загрози, а й модифіковані або zero-day-атаки (рисунок 1.5) [11].

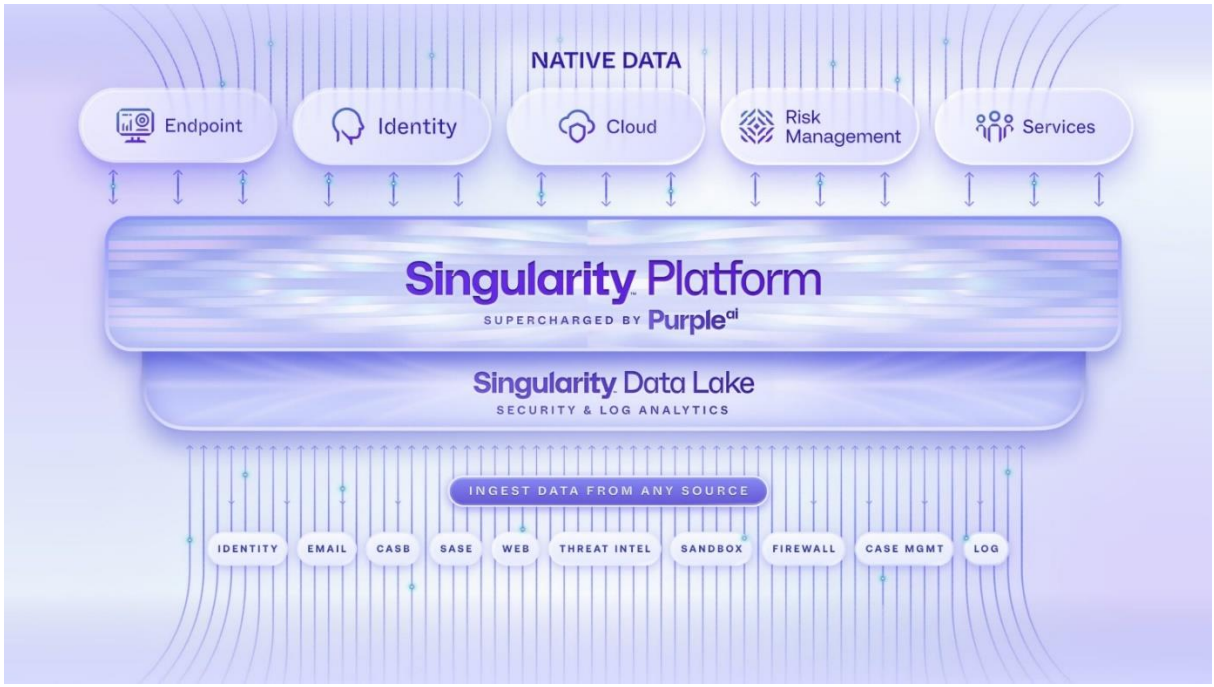

Рис. 1.5. Огляд SentinelOne Singularity [24]

Особливістю SentinelOne є функція Rollback – автоматичне відновлення системи до стану, що передував зараженню, навіть у разі шифрування даних. Це дозволяє організаціям знизити збитки від атак програм-вимагачів та зменшити час на відновлення інфраструктури. Ще однією ключовою перевагою платформи є потужна видимість, яка включає повний запис подій, побудову графів взаємодії між

процесами, інтеграцію з MITRE ATT&CK та можливість виконання сценаріїв hunt-аналізу [11].

SentinelOne підтримує захист не лише класичних ОС, а й контейнерів, хмарних інстансів і навіть OT/IoT-пристроїв, що робить її універсальним рішенням для різних типів організацій. Інтерфейс рішення інтуїтивно зрозумілий і орієнтований на швидке реагування, дозволяючи навіть невеликим командам ІБ ефективно протидіяти загрозам [11]. За результатами незалежного оцінювання NSS Labs та MITRE, SentinelOne стабільно демонструє один із найнижчих рівнів false positives і високий рівень детектування. Це свідчить про високу якість виявлення ШПЗ навіть у складних сценаріях [12].

Velociraptor – це інструмент з відкритим кодом, орієнтований на професійне виявлення ШПЗ та цифрову криміналістику. На відміну від автоматизованих EDR-рішень, Velociraptor надає повний контроль над процесом збору, аналізу та кореляції артефактів з кінцевих точок. Програму активно використовують фахівці з кібербезпеки для глибокого ручного threat hunting, відстеження складних атак і проведення повноцінних розслідувань інцидентів (рисунок 1.6) [13].

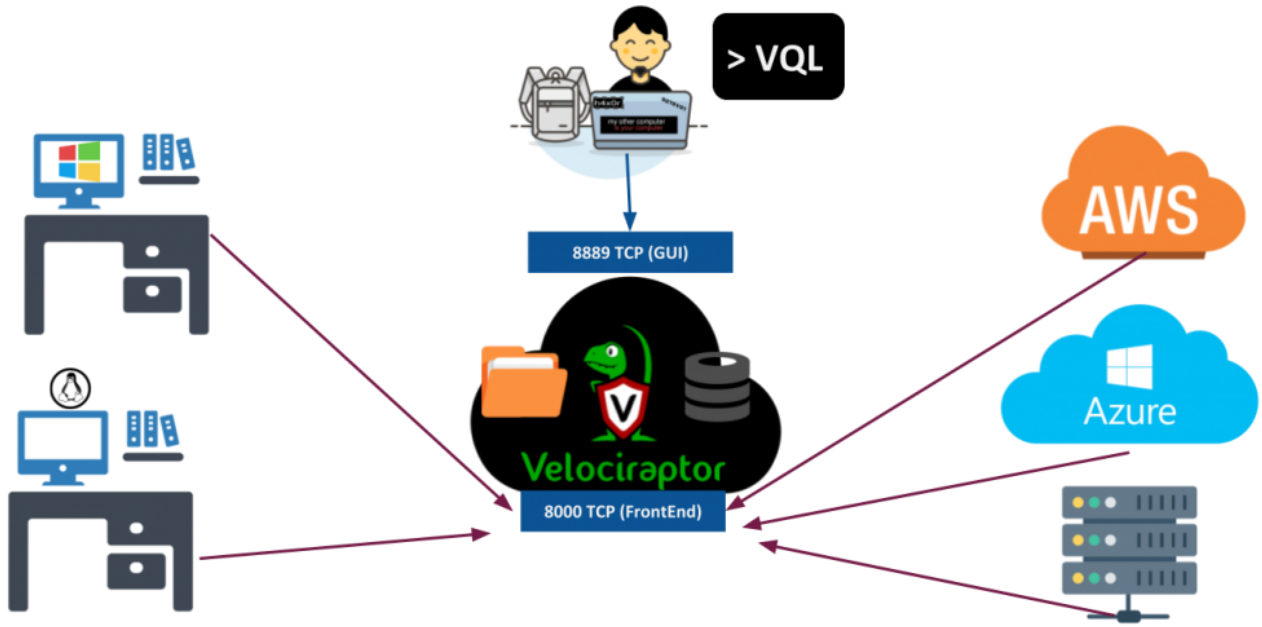

Рис. 1.6. Інтеграція Velociraptor у мережу [25]

Серцем системи є власна мова запитів –VQL (Velociraptor Query Language), яка дозволяє створювати кастомні сценарії збору інформації з файлової системи, журналів, реєстру, мережевої активності та пам’яті. Інструмент також підтримує розгортання масштабованої інфраструктури агентів, які непомітно працюють на кінцевих точках, збираючи дані в реальному часі або за запитом. Таким чином, Velociraptor підходить не лише для реактивного аналізу, а й для постійного моніторингу систем на предмет наявності ШПЗ чи ознак його присутності [13].

Завдяки своїй гнучкості, прозорості та відкритому коду, Velociraptor може бути адаптований до потреб конкретної організації, включаючи написання власних артефактів, інтеграцію з іншими інструментами та автоматизацію процесів на основі кастомних правил. При цьому система не потребує значних апаратних ресурсів і може використовуватись навіть у невеликих командах безпеки з обмеженим бюджетом.

Документація та приклади використання Velociraptor доступні на офіційному сайті, а також у відкритому репозиторії GitHub, що підтверджує активну підтримку з боку спільноти. У контексті захисту кінцевих точок від ШПЗ, Velociraptor не є класичним EDR-рішенням, але дає унікальні можливості для виявлення складних загроз на рівні артефактів та глибокого аналізу, які недоступні для більшості комерційних продуктів.

З огляду на широкий спектр доступних рішень для виявлення ШПЗ, важливо не лише ознайомитися з їхніми функціональними можливостями, але й здійснити критичну оцінку їхньої ефективності, гнучкості та придатності до різних умов експлуатації. Кожне з розглянутих рішень реалізує власний підхід до виявлення ШПЗ – від автономного реагування з використанням штучного інтелекту до глибокої криміналістики та ручного аналізу. Разом із перевагами, такими як висока точність детектування або підтримка автоматичного відновлення системи, кожна платформа має свої обмеження – потребу в потужних ресурсах, високу вартість чи складність налаштування. У таблиці 1.2 наведено порівняльну характеристику чотирьох розглянутих вище рішень, у якій узагальнено ключові функції, сильні

сторони та недоліки. Такий аналіз дозволяє обґрунтовано оцінити доцільність впровадження кожного з рішень у конкретному організаційному контексті.

Таблиця 1.2.

Порівняння різних рішень для виявлення ШПЗ

Рішення	Ключові функції	Переваги	Недоліки
CrowdStrike Falcon	Поведінковий аналіз, threat intelligence, ML-аналіз, ізоляція пристроїв, телеметрія, виявлення безфайлових атак.	Висока точність виявлення, масштабованість, реальний час, швидка ізоляція, захист від fileless malware.	Комерційна модель (висока вартість), залежність від хмари.
Microsoft Defender for Endpoint	Захист нового покоління, sandbox, автоматизоване реагування, управління вразливостями, інтеграція з Microsoft Threat Experts.	Вбудована в Windows, автоматизація, відповідність AV-тестам, глибока інтеграція.	Обмежена кастомізація, орієнтованість переважно на Microsoft-інфраструктуру.
SentinelOne Singularity	Автоматизоване виявлення, rollback, ізоляція процесів, аналітика поведінки, інтеграція з MITRE ATT&CK, захист хмари та IoT.	Висока автономність, швидке реагування, rollback файлів після шифрування, аналітична візуалізація.	Ціна, складність початкового налаштування.
Velociraptor	VQL-запити, цифрова криміналістика, моніторинг активності, кастомні артефакти, масштабованість, збір артефактів на запит або в реальному часі.	Безкоштовне ПЗ, повний контроль, глибокий аналіз, можливість кастомізації, активна спільнота.	Високий поріг входу, не автоматизоване реагування, потреба в фахівцях.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА КІНЦЕВИХ ТОЧКАХ ОРГАНІЗАЦІЇ НА БАЗІ VELOCIRAPTOR

2.1 Архітектура та компоненти рішення Velociraptor

У процесі дослідження методів виявлення ШПЗ на кінцевих точках організації виникає потреба у виборі ефективного, надійного та масштабованого інструменту, який би відповідав сучасним викликам кібербезпеки. Ураховуючи складність і різноманітність сучасних загроз, а також обмеження класичних антивірусних рішень, було прийнято рішення зосередитись на використанні платформи Velociraptor – інструменту з відкритим кодом, що призначений для цифрової криміналістики, активного пошуку загроз (threat hunting) та глибокого аналізу кінцевих точок.

Velociraptor вирізняється серед аналогів своєю унікальною комбінацією відкритості, гнучкості і масштабованості, що дозволяє адаптувати його під найрізноманітніші сценарії використання – від аудиту окремої машини до централізованого моніторингу тисяч кінцевих точок у великих підприємствах. На відміну від багатьох комерційних систем, які орієнтовані на автоматичне виявлення загроз, Velociraptor дає змогу фахівцям безпеки глибоко досліджувати зібрані артефакти, розробляти власні сценарії пошуку підозрілої активності та проводити ретельні розслідування інцидентів.

Серед головних переваг Velociraptor слід виділити можливість індивідуального налаштування логіки збору даних за допомогою власної мови запитів VQL (Velociraptor Query Language), що відкриває безмежні можливості для детального аналізу операційної системи, мережевої активності, пам’яті та інших системних ресурсів. Крім того, Velociraptor підтримує як режим постійного моніторингу, так і одноразовий збір артефактів, що робить його гнучким інструментом для різних етапів аналізу та реагування.

Важливою особливістю архітектури Velociraptor є те, що система не використовує зовнішні бази даних для зберігання інформації. Натомість усі зібрані дані, у тому числі артефакти, логи та інша інформація, зберігаються у файловій системі сервера у вигляді звичайних файлів і каталогів. Такий файло-орієнтований підхід значно спрощує резервне копіювання, управління життєвим циклом даних і усуває потребу в додатковій інфраструктурі, такій як спеціалізовані бази даних або хмарні сервіси зберігання.

Крім того, завдяки цій конструкції Velociraptor є сумісним із розподіленими файловими системами, такими як Amazon EFS, Google Filestore або універсальна NFS. Це дозволяє масштабувати систему та інтегрувати її в існуючу інфраструктуру організації без значних змін, забезпечуючи надійне і гнучке зберігання великих обсягів даних, необхідних для глибокого аналізу та розслідування кіберінцидентів

З технічної точки зору, архітектура Velociraptor побудована за класичною моделлю «клієнт-сервер» (рисунок 2.1) та складається з кількох ключових компонентів [13]:

Client – це екземпляр Velociraptor, що працює на кінцевій точці, тобто це агент кінцевої точки;

Frontend – це серверний компонент , який взаємодіє з клієнтом;

GUI – це сервер веб-застосунків, який забезпечує адміністративний інтерфейс.

API — це сервер API на основі gRPC.

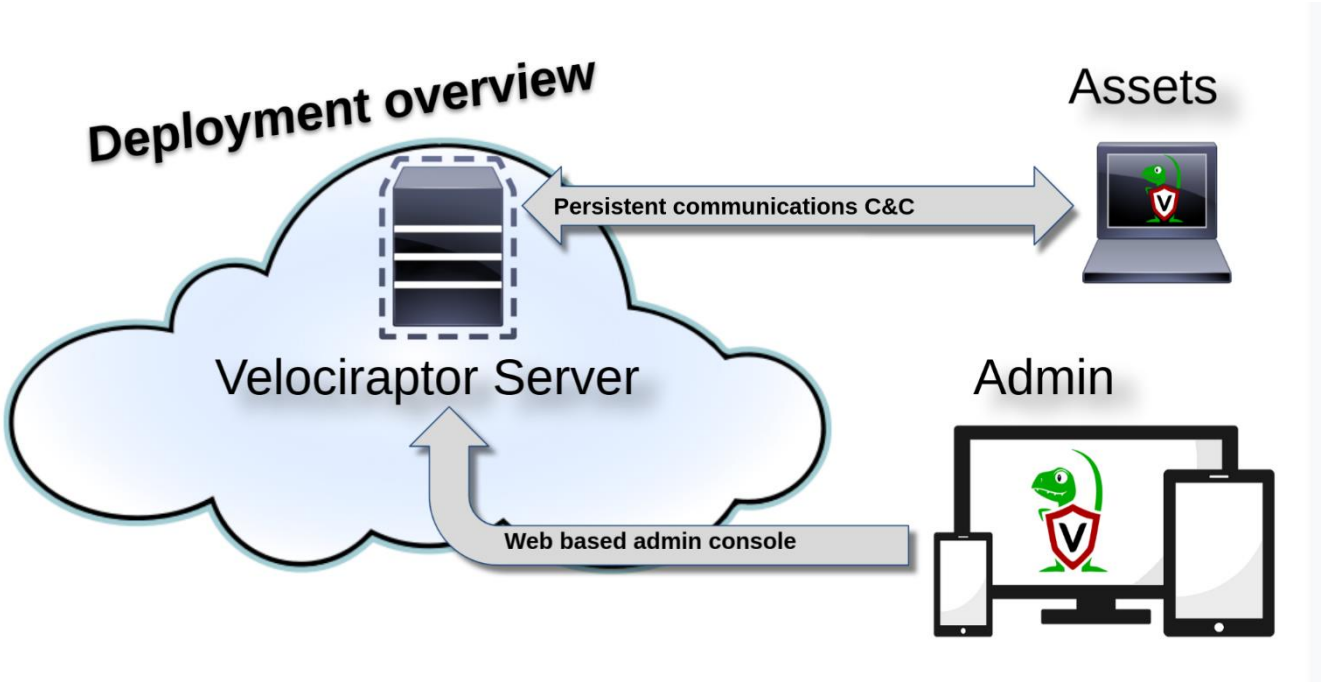

Рис. 2.1. Архітектура Velociraptor [13]

Враховуючи вищезазначене, Velociraptor є оптимальним вибором для дослідження та впровадження в межах цієї роботи, оскільки поєднує в собі глибину аналізу, гнучкість налаштувань і відкритість, що сприяє підвищенню якості виявлення і розслідування шкідливих програм на кінцевих точках.

2.2 Призначення та основні функції компонентів Velociraptor

Після попереднього огляду архітектури системи Velociraptor доцільним є детальний аналіз функціонального призначення та основних завдань її складових компонентів. Такий підхід дозволяє більш глибоко розкрити механізми взаємодії між серверною інфраструктурою, агентами, адміністративним інтерфейсом та системою зберігання даних. Розуміння специфіки функціонування кожного з елементів є необхідною умовою для комплексного оцінювання ефективності системи в контексті виявлення та розслідування ШПЗ на кінцевих точках організації. Нижче проведено систематичний розгляд основних функцій і ролей компонентів Velociraptor, що формують основу її працездатності та здатність до масштабування [13]:

Перший і, безумовно, найважливіший компонент – це *Client*. Це програмний агент, що розгортається безпосередньо на кінцевих точках, таких як персональні комп'ютери, сервери чи інші пристрої корпоративної мережі. Агент відповідає за збір великого спектру інформації, включаючи системні логи, список активних процесів, мережеву активність, зміни у файловій системі і вміст реєстру. Особливістю цього компонента є те, що він підтримує постійний зв'язок із сервером, що дозволяє миттєво отримувати нові завдання та оперативно передавати результати. Водночас агент розроблений таким чином, щоб бути максимально непомітним для користувача та не створювати надмірного навантаження на ресурси системи, що є критично важливим для збереження продуктивності підприємства.

Другим ключовим елементом є *Frontend* – серверна частина, яка виступає посередником між адміністративним інтерфейсом і агентами. Вона забезпечує прийом запитів від користувачів, формує інструкції для агентів і розподіляє завдання за допомогою системи черг, що дозволяє оптимально управляти навантаженням та забезпечує масштабованість системи. Frontend також відповідає за отримання та первинну обробку даних, що надходять від агентів, створюючи основу для подальшого детального аналізу. Завдяки такому поділу ролей система здатна ефективно працювати навіть у великих корпоративних мережах із тисячами кінцевих точок.

Третій компонент – це *GUI*, який надає адміністраторам зручний та інтуїтивно зрозумілий веб-інтерфейс для роботи із системою. Через цей інтерфейс користувачі можуть створювати та запускати запити на збір даних за допомогою потужної та гнучкої мови запитів VQL, контролювати стан агентів, аналізувати отримані результати та формувати детальні звіти. Така централізація управління дозволяє суттєво підвищити продуктивність роботи команд безпеки, спрощує налаштування і моніторинг системи, а також забезпечує високу гнучкість у побудові кастомних сценаріїв для пошуку і реагування на загрози.

Останнім із базових компонентів є *API*, який реалізовано як сервер на основі протоколу gRPC і який забезпечує програмний доступ до функцій Velociraptor. API дає змогу інтегрувати платформу з іншими інформаційними системами,

автоматизувати рутинні процеси, розширювати функціональність шляхом додавання власних сценаріїв, а також масштабувати рішення відповідно до специфічних потреб організації. Цей компонент суттєво підвищує гнучкість та адаптивність системи, дозволяючи побудувати єдину екосистему кіберзахисту, де Velociraptor є одним із ключових елементів.

Загалом, комбінація цих чотирьох компонентів – Client, Frontend, GUI та API – забезпечує цілісний, масштабований і надійний інструментарій для сучасних організацій, які прагнуть не лише виявляти шкідливе програмне забезпечення на кінцевих точках, а й проводити глибокий аналіз інцидентів, швидко реагувати на загрози та ефективно управляти безпекою у динамічному кіберпросторі.

2.3 Процеси функціонування рішення Velociraptor

Функціональне ядро Velociraptor базується на трьох концептуально відмінних, але взаємодоповнюючих процесах: *Collect*, *Monitor* та *Hunt*. Кожен з них виконує окрему роль у межах більш широкої стратегії цифрової криміналістики та виявлення загроз, формуючи унікальний підхід до безпеки, що базується на достовірних даних, контекстуальному аналізі та миттєвому реагуванні [13].

Процес *Collect* у Velociraptor є центральним механізмом для швидкого і цілеспрямованого збору цифрових артефактів із кінцевих точок, що є надзвичайно важливим у рамках цифрової криміналістики (DFIR) та реагування на інциденти. Збір даних може включати як великі файлові об’єкти, так і більш динамічні дані – інформацію про запущені процеси, мережеві підключення та інші аспекти системного стану. Усі ці дані збираються за допомогою артефактів – спеціально налаштованих VQL-запитів, що гнучко підлаштовуються під конкретні потреби розслідування.

Одним із найпопулярніших артефактів для масового збору файлів є Windows.KapeFiles.Targets, який базується на відкритому репозиторії KapeFiles. Цей артефакт дозволяє збирати широкий спектр важливих для тріажу файлів, згрупованих у цілі (Targets). Таке групування полегшує налаштування та

масштабування збору. Однак важливо враховувати, що обсяг зібраних даних може бути дуже великим – наприклад, збір $MFT з тисяч кінцевих точок може легко призвести до сотень гігабайтів або навіть терабайтів даних. Тому Velociraptor передбачає механізми контролю ресурсів, включно з можливістю обмежувати максимальний розмір завантажуваних файлів та автоматично припиняти збір після досягнення встановлених лімітів [15].

У випадках, коли неможливо або небажано встановлювати агент Velociraptor безпосередньо на кінцеві точки, або коли пристрої знаходяться в ізольованих мережах без постійного підключення, застосовується offline triage – офлайн-збір даних за допомогою спеціально підготовлених виконуваних файлів (offline collectors). Ці збори можуть бути заздалегідь сконфігуровані у вигляді виконуваного файлу, що не вимагає від користувача введення параметрів і автоматично збирає налаштовані артефакти, формуючи захищений архів із результатами. Таким чином, навіть непрофесіонали можуть виконати збір важливих даних з мінімальними зусиллями [15].

Ключовим аспектом є безпека і конфіденційність зібраної інформації, адже вона часто містить чутливі відомості. Для цього Velociraptor підтримує різні методи шифрування створених ZIP-архівів. Стандартне шифрування ZIP файлів має недолік – не шифрує метадані, такі як імена файлів і їх розміри. Щоб захистити цю інформацію, архіви створюються у вигляді зашифрованого контейнера, всередині якого знаходиться вкладений ZIP-файл (data.zip). Для підвищення безпеки рекомендується використовувати X509 сертифікатне шифрування: у цьому випадку у бінарний offline collector вбудовується публічний сертифікат сервера, який використовується для шифрування випадкового пароля, яким зашифровано архів. Розшифрувати такі дані може лише сервер із відповідним приватним ключем, що значно підвищує безпеку передачі та зберігання даних. Після успішного збору офлайн-архів можна завантажити на сервер кількома способами, залежно від умов мережі і політики безпеки. Velociraptor підтримує автоматичне завантаження архівів у різні віддалені сховища, зокрема [16]:

SMB Share – протокол файлового обміну, популярний у Windows-середовищах. Для налаштування потрібно створити спеціального користувача з правом запису в певну папку, забезпечити коректні ACL та протестувати можливість завантаження. У Velociraptor це можна легко зробити за допомогою VQL-запитів, що перевіряють доступність і права;

Хмарні сховища, як-от Azure Blob Storage – сучасне і зручне рішення для безпечного збереження архівів у хмарі. Velociraptor використовує механізм SAS (Shared Access Signature), який дозволяє створювати тимчасові токени доступу з обмеженими правами (наприклад, лише запис), що робить завантаження безпечним і контрольованим;

Також підтримуються завантаження у Google Cloud Storage, AWS S3, SFTP та інші сервіси [16].

Ці можливості дозволяють уникнути ризиків, пов’язаних із ручною передачею файлів, зменшують людський фактор і підвищують швидкість доставлення даних до аналітиків. Для великих організацій та розгалужених мереж, де офлайн-збір може бути використаний як один із способів тріажу, можливість інтеграції з різними типами віддалених сховищ робить процес максимально зручним і безпечним. Після того, як колекція зібрана та завантажена на сервер Velociraptor, її можна імпортувати у графічний інтерфейс платформи для подальшого аналізу, інтерактивного розслідування та створення звітів. Це робить процес *Collect* не просто збором даних, а комплексним рішенням, яке охоплює всі етапи від ініціації тріажу до обробки зібраної інформації.

Таким чином, механізм *Collect* у Velociraptor – це масштабований, гнучкий і безпечний спосіб отримання цифрових доказів з кінцевих точок, який дозволяє ефективно працювати як у мережевому режимі з постійним зв’язком, так і в автономних умовах з віддаленим завантаженням результатів.

Процес *Monitor* у Velociraptor – це потужний механізм безперервного відстеження подій на кінцевих точках мережі, що значно підвищує ефективність реагування на інциденти безпеки. Головна мета цього процесу – фіксувати та зберігати всі важливі події, які відбуваються на пристроях користувачів, а також

забезпечувати доступ до цих даних для подальшого аналізу. Завдяки цьому система безпеки може не лише реагувати на загрози у момент їхньої появи, а й звертатись до історичних записів, що є надзвичайно важливим для глибинного розслідування інцидентів.

Технічно, Velociraptor спирається на потужну мову запитів VQL, яка дозволяє будувати як одноразові запити на збір даних, так і довготривалі подієві запити (event queries). Ці подієві запити працюють у безперервному режимі, не завершуючись, і повертають дані щоразу, коли у системі виникає подія, що відповідає заданим критеріям. Наприклад, запит можна налаштувати так, щоб він відстежував запуск нових процесів, встановлення сервісів, зміни у файловій системі чи будь-які інші критичні для безпеки операції. Це дозволяє отримувати інформацію в реальному часі, що значно підвищує якість моніторингу. Прикладом такого підходу є використання плагіна watch_evtx(), який слугує аналогом плагіна parse_evtx(), але замість одноразового зчитування журналу подій безпосередньо «слухає» цей журнал у реальному часі, фіксуючи всі нові події. Завдяки цьому, якщо в системі встановлюється новий сервіс або відбувається будь-яка інша подія, Velociraptor миттєво її фіксує і відправляє на сервер. При цьому результати моніторингу надсилаються у пакетах – розмір цих пакетів можна контролювати за допомогою спеціального параметра, що дозволяє балансувати між оперативністю надсилання даних і навантаженням на мережу [17].

Для ефективної роботи подієвих запитів у Velociraptor клієнтська частина має спеціальну структуру – таблицю подій, в якій зберігаються всі активні запити, що виконуються у фоновому режимі. Коли система отримує нові запити або оновлення існуючих, вона динамічно оновлює цю таблицю, скасовуючи старі запити та запускаючи нові, без необхідності перезавантаження агентів. Це робить моніторинг гнучким і адаптивним. Кожен зібраний подієвий запис автоматично зберігається в спеціальному віртуальному файловому сховищі (VFS) клієнта, а потім надсилається на центральний сервер. Це дає змогу централізовано зберігати та обробляти історичні дані про події, а також готувати звіти чи сповіщення на їх основі [17].

Хоча на даному етапі Velociraptor просто фіксує і зберігає події, у майбутньому планується додати механізми автоматичного опрацювання подій та генерації тривог, що зробить моніторинг ще більш потужним. Окремо варто виділити можливості плагіна wmi_events(), який використовує механізми Windows Management Instrumentation (WMI) для відстеження створення нових процесів у системі. Цей плагін інсталює спеціальний слухач подій, який у режимі реального часу повідомляє про запуск процесів. Однак такі події зазвичай містять лише базову інформацію, наприклад, ідентифікатор процесу (PID), і щоб отримати детальніші дані, наприклад повну командну строку запуску, Velociraptor виконує додаткові запити до WMI. Це забезпечує повноцінний і деталізований моніторинг процесів, що є критично важливим для виявлення шкідливої активності [17].

Налаштування того, які саме події повинні моніторитись, у Velociraptor спрощене завдяки використанню артефактів – заздалегідь визначених іменованих наборів VQL-запитів. Користувачам достатньо вказати назву артефакту (наприклад, Windows.Events.ProcessCreation або Windows.Events.ServiceCreation), і система автоматично розгорне відповідні подієві запити на кінцевих точках. Ці налаштування зберігаються у конфігурації сервера, і кожен раз, коли список моніторингових подій оновлюється, клієнти отримують інструкції щодо оновлення своїх таблиць подій. Такий підхід робить процес масштабування і адміністрування моніторингу максимально простим і гнучким [17].

Інтерфейс користувача Velociraptor відображає зібрані події у вигляді файлів у віртуальній файловій системі клієнтів, що дозволяє зручно завантажувати, переглядати і обробляти їх за межами платформи. Хоча на даний момент моніторинг у Velociraptor зосереджений на зборі та збереженні подій, ця функціональність вже приносить значну користь, наприклад, ведення журналу запуску процесів і можливість віддаленого збору логів. Підсумовуючи, збір історичних і поточних подій дозволяє не лише оперативно виявляти загрози, але й проводити глибокий аналіз минулих подій, що є основою для якісної цифрової криміналістики та безперервного контролю безпеки [17].

Процес *Hunt* у Velociraptor являє собою організований і масштабований механізм одночасного збору артефактів із великої кількості кінцевих точок у мережі. Основна ідея *Hunt* полягає у тому, щоб забезпечити можливість централізовано та автоматизовано запускати однакові запити для збору інформації на багатьох пристроях, що значно спрощує проведення аудиту, розслідувань і виявлення загроз.

Hunt керується спеціальним компонентом – Hunt Manager. Цей менеджер відповідає за планування запусків збору на клієнтах, які відповідають певним критеріям (наприклад, наявність конкретної мітки або певні характеристики системи), а також відслідковує статус кожного з цих зборів у межах конкретного полювання. Важливо розуміти, що механізм збору артефактів на кінцевих точках при цьому не змінюється – просто він автоматизується і масштабується завдяки Hunt Manager [18].

Планування нового полювання починається з вибору цільових машин, що можуть бути відфільтровані за допомогою ярликів (labels), що дуже зручно для вибіркового ітеративного сканування. Наприклад, можна створити *Hunt*, який буде виконуватись лише на системах із міткою “Compromised” або “Preserve”, що дозволяє гнучко адаптуватися під поточні потреби розслідування або захисту. Hunt має дату закінчення – так званий термін придатності, після якої нові клієнти не долучаються до полювання, однак *Hunt* може тривати досить довго, адже нові пристрої можуть з’являтися у мережі у будь-який момент [18].

Кожен клієнт може брати участь у конкретному *Hunt* лише один раз, що дозволяє уникнути дублювання зборів. Після налаштування цілей *Hunt*, користувач обирає і конфігурує артефакти, які потрібно зібрати, – це можуть бути будь-які доступні у Velociraptor VQL-запити чи комплексні набори даних. Після запуску *Hunt* сотні і тисячі клієнтів одночасно почнуть збирати інформацію, що дозволяє оперативно отримати повну картину стану мережі чи конкретних систем. Однак через значні обсяги даних рекомендується тестувати артефакти на кількох машинах перед широкомасштабним запуском, щоб уникнути перевантаження мережі чи серверів [18].

У процесі виконання *Hunt* адміністратори можуть відслідковувати статус збору на кожному клієнті у реальному часі. Як тільки дані збираються і надходять на сервер, вони стають доступними для подальшої обробки. Velociraptor автоматично створює спеціальні нотатки (notebooks) для кожного *Hunt*, де можна застосовувати VQL-запити до зібраних результатів, щоб виділити найважливіші записи, фільтрувати дані, агрегувати та аналізувати їх. Це спрощує пошук підозрілих індикаторів, таких як запуск специфічних команд, аналіз незвичних файлів чи інших поведінкових патернів [18].

Важливо враховувати, що при полюванні на великій кількості кінцевих точок обсяг даних може дуже швидко зростати. Наприклад, збір 100 Мб файлу з 10 тисяч пристроїв може вимагати понад 1 терабайт сховища. Тому при плануванні *Hunt* рекомендується віддавати перевагу більш цільовим артефактам, які повертають обмежену кількість рядків, а не великим сирим файлам, що потребують подальшої офлайн-обробки [18].

Крім автоматичного додавання, Velociraptor також дозволяє вручну додавати або перепланувати збори в межах *Hunt*. Це корисно у випадках, коли збір не вдався або потрібно оновити дані. Адміністратор може скопіювати попередній збір, налаштувати параметри і знову додати його до полювання, що забезпечує максимальну гнучкість у керуванні процесом. Ще одна важлива можливість – це інтеграція VQL-функції hunt_add(), що дозволяє повністю автоматизувати додавання потоків збору до Hunt та їх повторний запуск, що особливо корисно для великих мереж і автоматизованих процедур. Завдяки нотаткам (notebooks) для кожного *Hunt* адміністратори отримують потужний інструмент для управління і аналізу прогресу полювання. Вони можуть застосовувати підказки (Cell Suggestions) та корисні VQL-запити, які спрощують роботу з великими обсягами даних і дозволяють швидко виділяти релевантну інформацію [18].

Таким чином, процес *Hunt* у Velociraptor – це інноваційний і масштабований спосіб централізованого збору і аналізу даних з кінцевих точок, який автоматизує рутинні операції, оптимізує роботу команд безпеки і дозволяє оперативно виявляти та реагувати на загрози у масштабах всієї організації.

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ЗАСТОСУВАННЯ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА КІНЦЕВИХ ТОЧКАХ ОРГАНІЗАЦІЇ

3.1 Порядок розгортання системи виявлення шкідливого програмного забезпечення на кінцевих точках організації на базі Velociraptor

Розгортання Velociraptor – це комплексний процес, який включає три основні етапи: налаштування сервера, забезпечення доступу користувачів до веб-інтерфейсу, а також розгортання клієнтських компонентів на кінцевих точках [20].

Перший і найважливіший крок – це налаштування серверного компонента Velociraptor. На цьому етапі потрібно встановити серверний компонент Velociraptor та налаштувати його під вимоги вашої організації. Сервер є центром збору і обробки даних, тому важливо забезпечити його стабільну роботу, правильну конфігурацію мережі та безпеки. Також варто переконатися, що сервер запускається автоматично після перезавантаження.

Наступний крок – надати користувачам доступ до веб-інтерфейсу сервера. Важливо створити відповідні облікові записи, налаштувати права доступу та ролі, щоб кожен користувач міг працювати в межах своїх повноважень. Безпечна авторизація гарантує, що інформація і керування системою будуть доступні лише уповноваженим особам.

Останній, але не менш важливий етап – це розгортання клієнтських агентів Velociraptor на кінцевих точках. Існує кілька способів розгортання:

Інтерактивний запуск клієнта для тестування або невеликих систем;

Встановлення клієнта як системної служби через спеціальні інсталятори для стабільної роботи;

Безагентне розгортання для тимчасового або швидкого збору даних без встановлення постійного агента;

Використання офлайн-колекторів — автономних клієнтів для ізольованих систем або ситуацій з обмеженим мережевим доступом.

Вибір методу залежить від конкретних потреб і масштабів інфраструктури.

Нижче буде описано спосіб встановлення клієнта як системної служби через спеціальний інсталятор. Цей варіант розгортання передбачає, що серверна частина Velociraptor уже налаштована й функціонує у вашій внутрішній мережі.

Після налаштування серверу наступна мета – створити інсталяційний пакет для кінцевих точок Windows.

Найважливіше, що потрібно знати, це те, що всім клієнтам Velociraptor потрібна конфігурація клієнта, яка є специфічною для розгортання. Ця конфігурація є підмножиною повної конфігурації на основі YAML (рисунок 3.1). Оскільки сервер має доступ до повної конфігурації, він може надати нам конфігурацію клієнта, коли це необхідно у вигляді YAML-файлу. Сервер також може використовувати його внутрішньо, наприклад, під час створення інсталяційного пакета клієнта (для клієнтів Windows), як буде показано нижче.

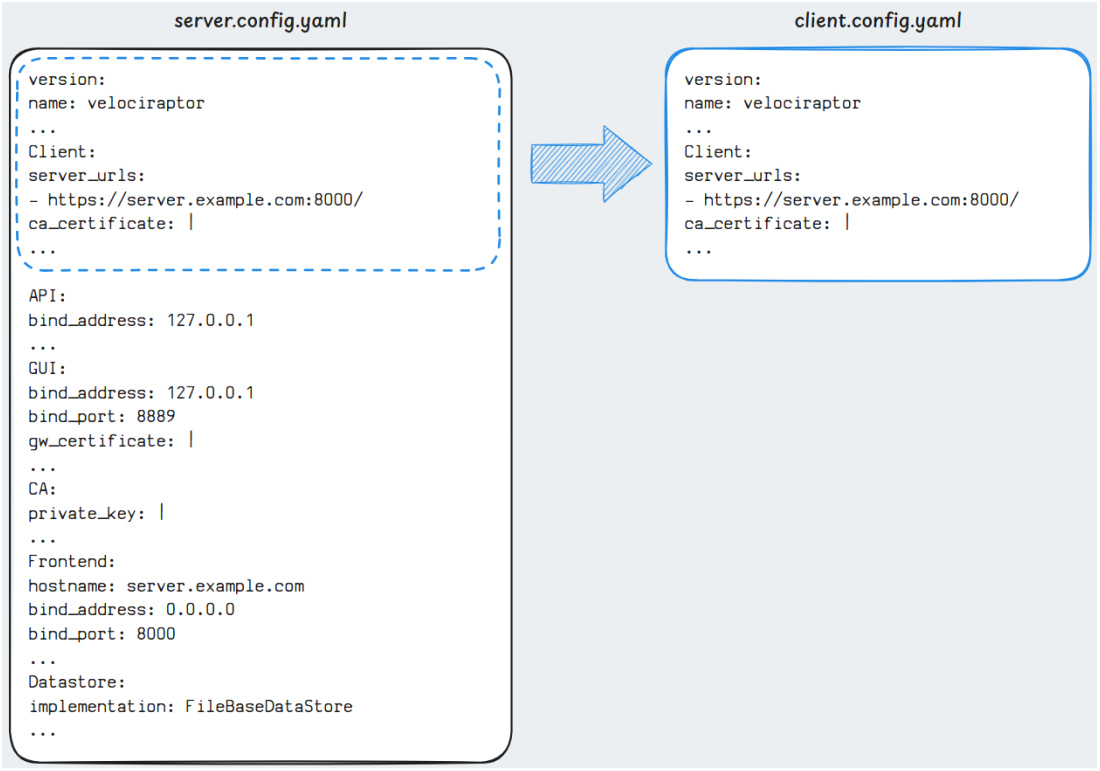

Рис. 3.1. Конфігурація клієнта є підмножиною повної конфігурації [19]

Найпоширенішим клієнтом на сьогоднішній день є 64-розрядна Windows. Стандартним форматом пакета інсталятора для Windows у корпоративних середовищах є формат MSI. Найшвидший і найпростіший спосіб створення власного MSI-пакета, повністю виконується в графічному інтерфейсі адміністратора [19]:

У веб-інтерфейсі Velociraptor обираємо «Server Artifacts» на бічній панелі з лівого боку сторінки та натискаємо «New Collection» (рисунок 3.2);

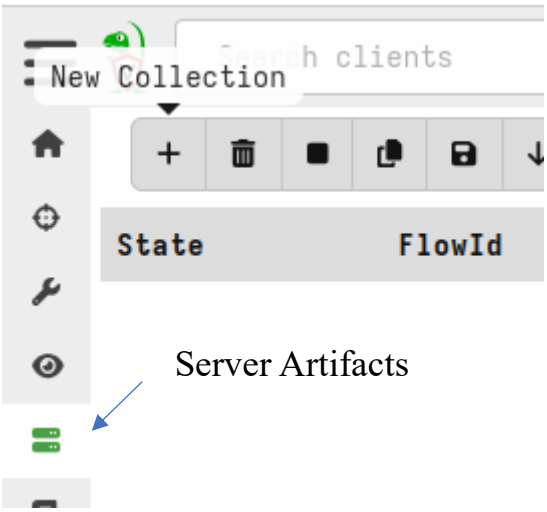

Рис. 3.2. Створення нової колекції

Додаємо нову колекцію: знаходимо Server.Utils.CreateMSI, вибераємо її та натискаємо «Launch» (рисунок 3.3);

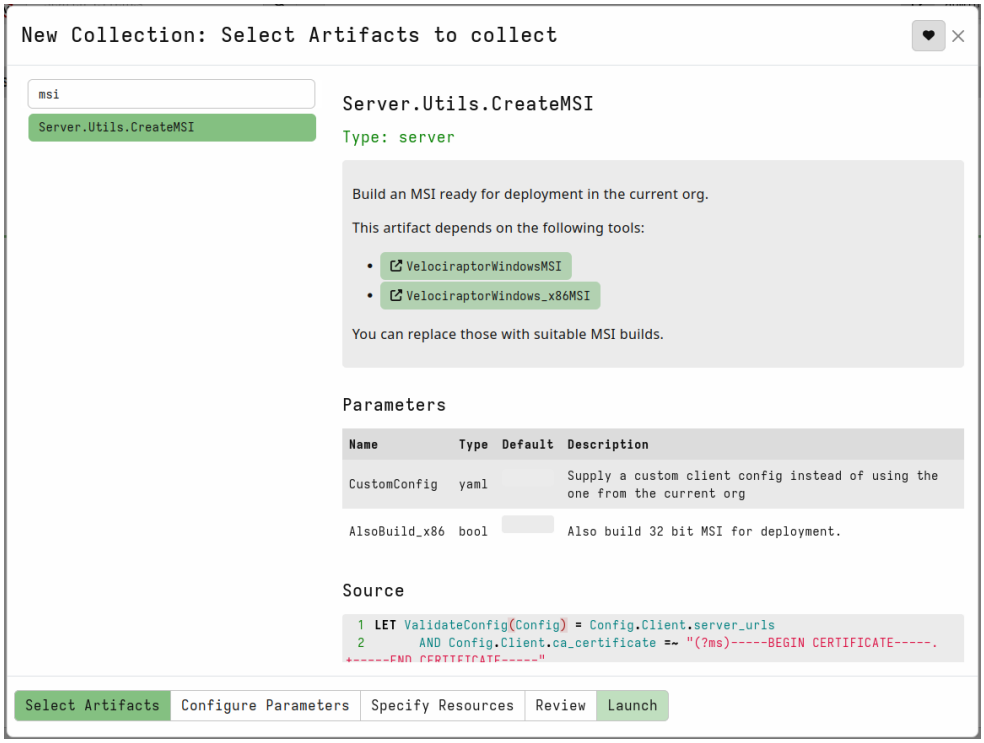

Рис. 3.3. Створення MSI-файлу

Перепакований MSI-файл буде доступний на вкладці «Uploaded Files» в колекції артефактів (рисунок 3.4);

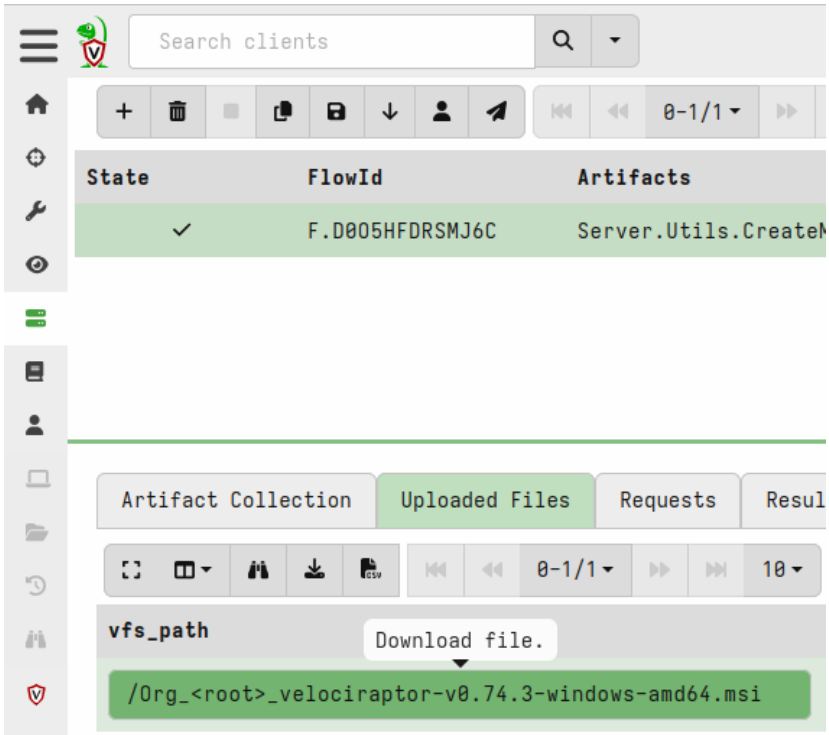

Рис. 3.4. Завантаження MSI-файлу

Завантажуємо MSI-файл, для цього натискаємо на нього. Потім потрібно буде скопіювати його на мережевий ресурс, USB-носій або інший носій, який дозволить отримати до нього доступ з кінцевої точки Windows.

На кінцевій точці Windows інсталяція виконується шляхом запуску MSI-файлу, створеного раніше. MSI встановлює клієнт як службу Windows, тому для цього потрібні підвищені права, і може бути запропоновано знайомий запит UAC, щоб дозволити продовження встановлення (рисунок 3.5);

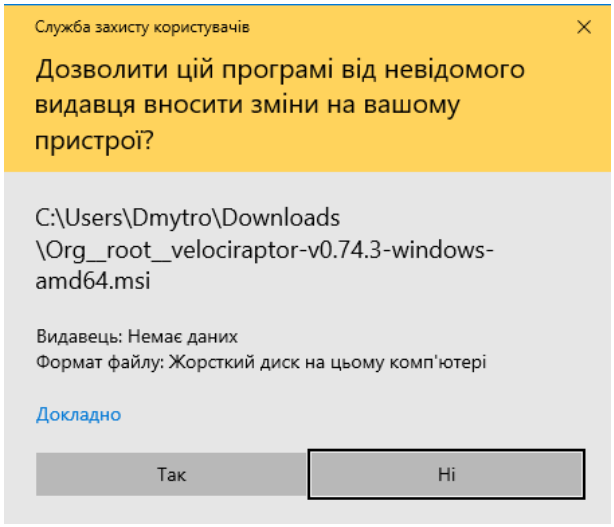

Рис. 3.5. Повідомлення від служби захисту користувачів

Після встановлення клієнт працюватиме як «Velociraptor Service», яку можна побачити в додатку «Services» (рисунок 3.6);

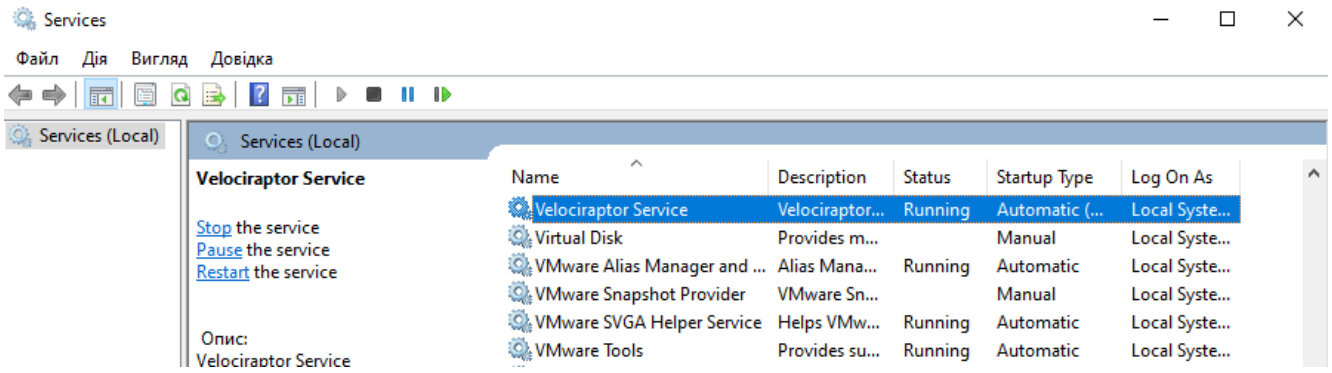

Рис. 3.6. Перевірка встановлення клієнту на машині Windows

Повернувшись на сервер Velociraptor, ми можемо побачити щойно зареєстрованого клієнта, натиснувши кнопку на панелі «Search clients» у верхній частині сторінки (рисунок 3.7).

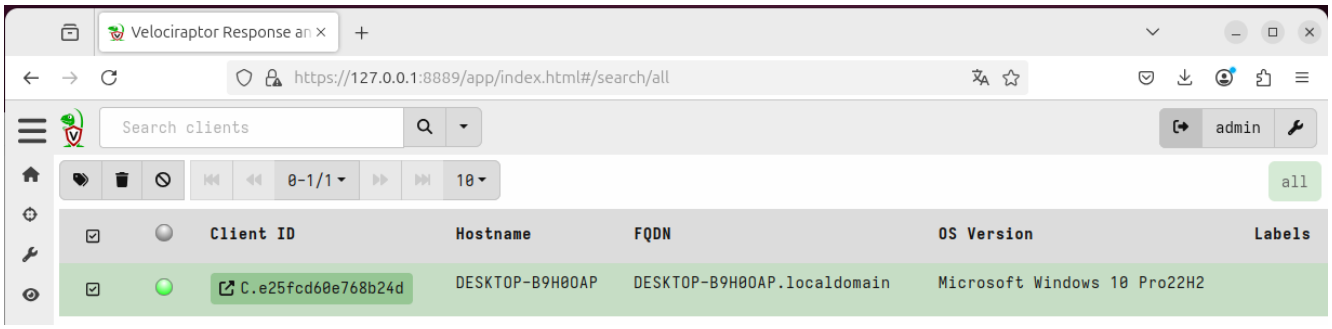

Рис. 3.7. Огляд зареєстрованого клієнта

3.2 Порядок виявлення шкідливого програмного забезпечення на кінцевих токах організації з використанням Velociraptor

Після встановлення клієнту Velociraptor, на віртуальну машину Windows було завантажено і запущено ШПЗ AgentTesla.

Agent Tesla – ШПЗ, яке виконує функції кейлогера (keylogger), викрадача інформації (stealer) та є вдосконаленим трояном віддаленого доступу (RAT), що написане мовами, які використовуються в Microsoft .Net (C #, Visual Basic .NET, C++/CLI, тощо). Наразі ШПЗ здатне відслідковувати та збирати дані вводу з клавіатури, робити скріншоти та отримувати облікові дані, що використовуються в різних програмах системи (наприклад Google Chrome, Mozilla Firefox, Microsoft Outlook, IceDragon, FILEZILLA, тощо) [21].

Почнемо процес виявлення ШПЗ за допомогою Velociraptor, для цього на бічній панелі з лівого боку сторінки та натискаємо «Hunt Manager» та обираємо «New Hunt» (рисунок 3.8);

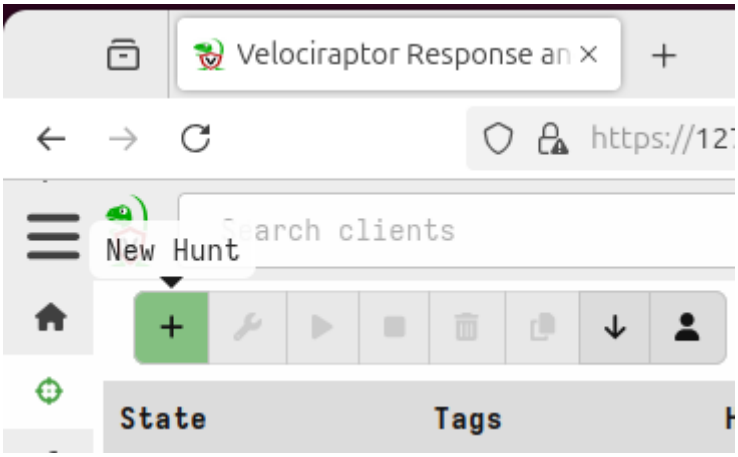

Рис. 3.8. Створення нового полювання

Обираємо та налаштовуємо артефакти для пошуку ШПЗ (рисунок 3.9).

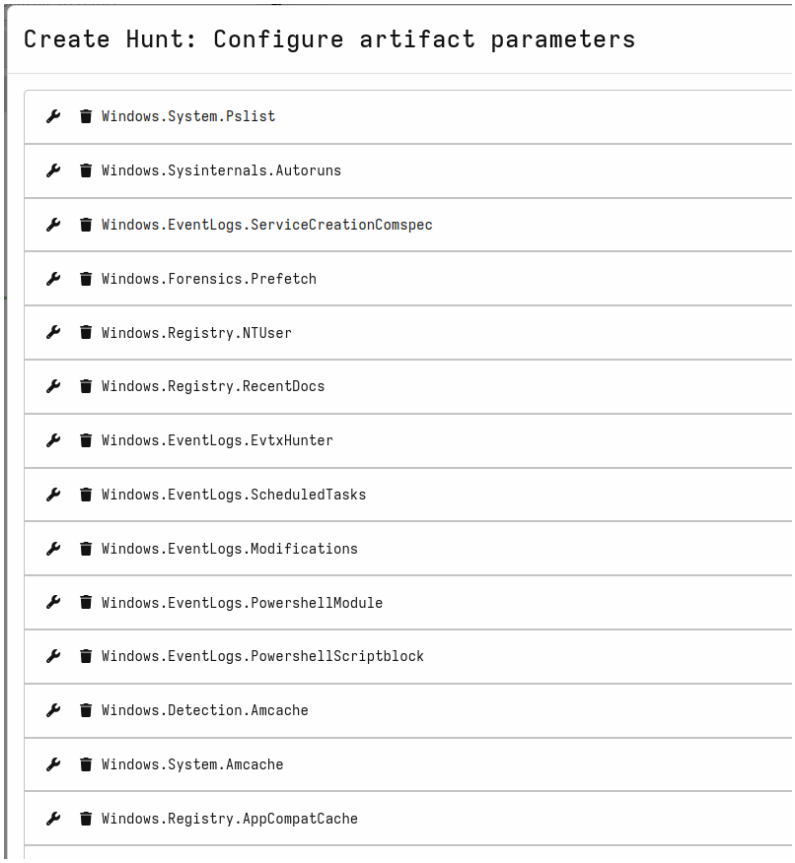

Рис. 3.9. Вибір артефактів

В нашому випадку це:

Windows.System.Pslist – збирає список запущених процесів у системі;

Windows.Sysinternals.Autoruns – аналізує всі можливі механізми автозапуску програм і служб;

Windows.EventLogs.ServiceCreationComspec – перевіряє журнали подій на створення служб з викликом командних інтерпретаторів;

Windows.Forensics.Prefetch – аналізує префетч-файли для визначення запуску програм;

Windows.Registry.NTUser – збирає дані з реєстру користувача для аналізу користувацьких налаштувань;

Windows.Registry.RecentDocs – вивчає список останніх відкритих документів у реєстрі;

Windows.EventLogs.EvtxHunter – шукає підозрілі події у системних журналах Windows;

Windows.EventLogs.ScheduledTasks – аналізує події, пов'язані з планувальником завдань;

Windows.EventLogs.Modifications – відстежує зміни у файлах і системі через журнали подій;

Windows.EventLogs.PowershellModule – збирає інформацію про завантажені модулі PowerShell з журналів;

Windows.EventLogs.PowershellScriptblock – аналізує виконання скриптів PowerShell за допомогою логів ScriptBlock;

Windows.Detection.Amcache – отримує інформацію з AMCache для аналізу запусків програм;

Windows.System.Amcache – аналізує системний AMCache для отримання деталей про виконувані файли;

Windows.Registry.AppCompatCache – вивчає кеш сумісності додатків для визначення запусків програм;

Windows.System.TaskScheduler – збирає інформацію про заплановані завдання Windows Task Scheduler.

Після створення полювання натискаємо «Run Hunt» (рисунок 3.10)

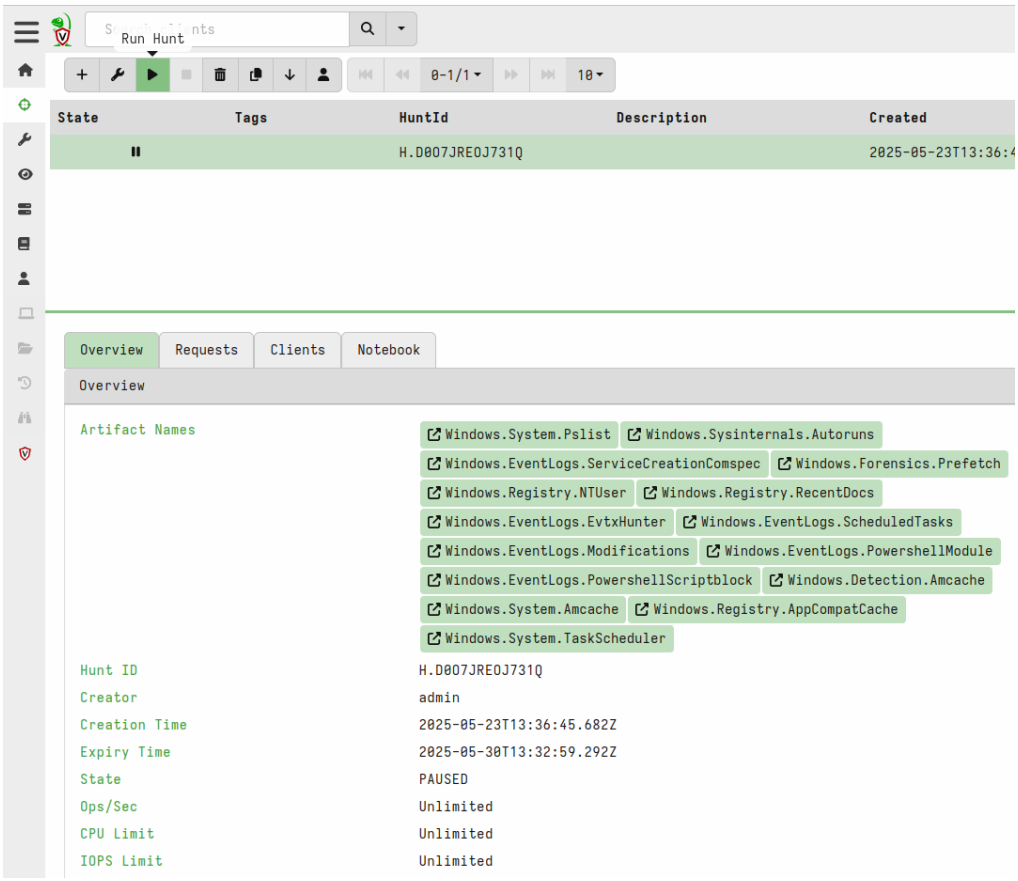

Рис. 3.10. Початок полювання

При швидкому аналізі результатів було помічено очевидну роботу ШПЗ (рисунок 3.11).

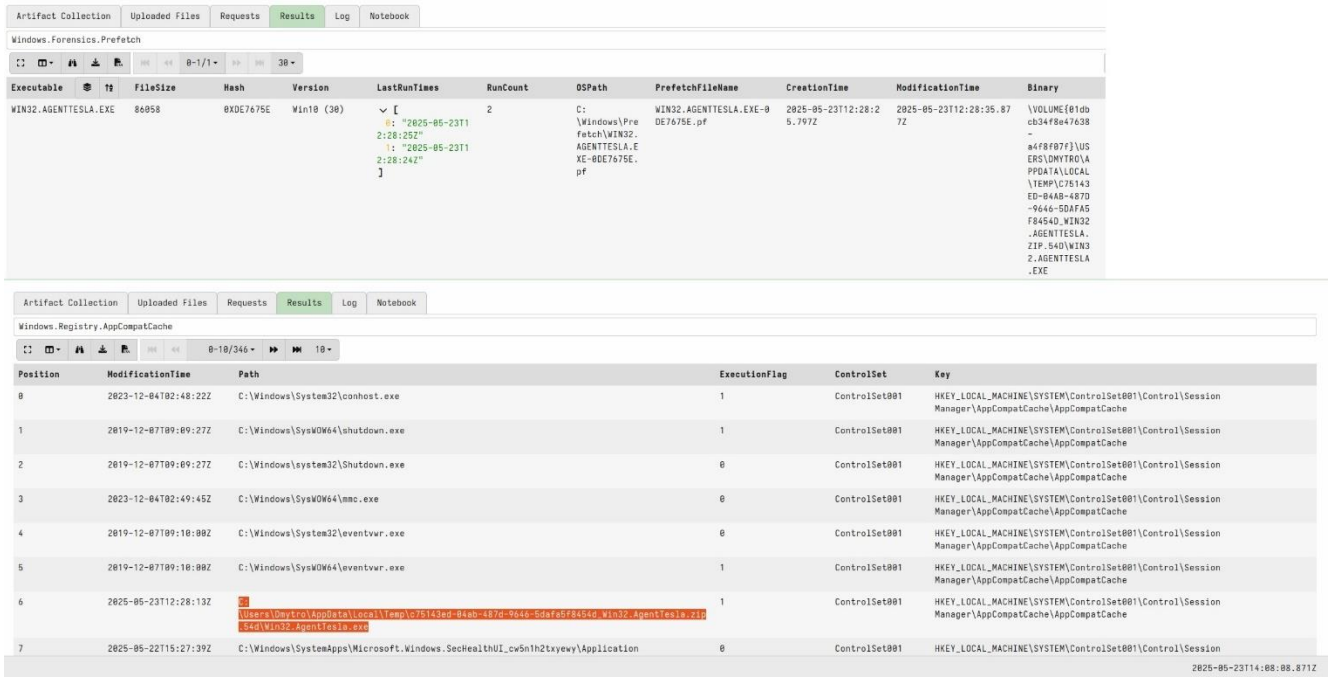

Рис. 3.11. Дані про ШПЗ

Також з неявних ознак було помічено незвичні записи автозапуску в реєстрі Windows, які імітують системні додатки (рисунок 3.12);

Artifact Collection | Uploaded Files | Requests | Results | Log | Notebook

Windows.Sysinternals.Autoruns

0-2/2 | 100 | Category (hijacks) | Mtime

Time	Entry Location	Entry	Enabled	Category	Profile	Description	Signer	Company	Image Path	Version	Launch String	MD5	SHA-1	PESHA-1
20250522-162000	HKLM\SOFTWARE\Classes\Htmlfile\Shell\Open\Command\(Default)			Hijacks	System-wide									
19950208-123550	HKLM\SOFTWARE\Classes\Htmlfile\Shell\Open\Command\(Default)	C:\Program Files\Internet Explorer\iexplore.exe	enabled	Hijacks	System-wide	Internet Explorer	(Verified) Microsoft Corporation	Microsoft Corporation	c:\program files\internet explorer\iexplore.exe	11.0.19041.5794		FC4CE1E98125E7E30AF6BC2D4A10250D	CC7E3F18771A3944CB08EC A03AE7D5062B911FFD	6C675333EA49F38BC569ABE9EA51DE78018BF3F1

Рис. 3.12. Записи автозапуску в реєстрі Windows

При більш детальному аналізі результатів можливо виявити додаткові ознаки присутності ШПЗ, такі як:

запуск шкідливих скриптів PowerShell;

незвичайні зміни в реєстрі та файлах;

мережеві з'єднання з C2-серверами.

3.3 Рекомендації щодо виявлення шкідливого програмного забезпечення на кінцевих точках організації

Потрібно впровадити системи поведінкового аналізу активності кінцевих точок. Такі системи мають виявляти відхилення від звичайного функціонування пристроїв, що дозволяє виявити нові або модифіковані зразки шкідливого програмного забезпечення. Поведінкове виявлення передбачає фіксацію нетипових дій, наприклад, змін у файловій системі, незвичних запитів до мережі або неавторизованих спроб запуску процесів. На відміну від сигнатурного аналізу, цей підхід здатен виявити загрози навіть тоді, коли вони ще не мають відповідних записів у базах антивірусних систем.

Має бути організована інтеграція з системами обміну інформацією про кіберзагрози (threat intelligence). Ці системи надають актуальні індикатори компрометації та техніки атак, які активно використовуються кіберзлочинцями. Інформація, що надходить з аналітичних центрів або галузевих платформ обміну даними про загрози, повинна автоматично оновлювати бази сигнатур, правила

фільтрації та політики безпеки. Це забезпечує більш оперативне виявлення актуальних загроз і підвищує ефективність захисту кінцевих точок.

Треба використовувати ізольовані середовища (sandbox) для аналізу підозрілих об'єктів. Підозрілі файли та процеси мають бути запущені в безпечному віртуальному середовищі з метою дослідження їх поведінки без ризику для основної інфраструктури. Це дає змогу виявляти шкідливу активність, яка може бути непомітною для традиційних механізмів виявлення, особливо якщо мова йде про цільові атаки або нові варіанти відомих загроз. Аналіз у пісочниці дозволяє точно визначити наміри коду до його реального запуску у виробничому середовищі.

Необхідно регулярно проводити сканування кінцевих точок на предмет вразливостей та оновлювати програмне забезпечення. Автоматизовані засоби керування вразливостями повинні виявляти застарілі версії операційних систем, бібліотек, драйверів і прикладних програм, які можуть бути використані шкідливим програмним забезпеченням. Після виявлення таких вразливостей треба оперативно застосовувати патчі або вживати інших заходів щодо мінімізації ризику експлуатації. Своєчасне оновлення забезпечує підвищення рівня кіберстійкості організації.

Слід забезпечити безперервний моніторинг та збір логів з кінцевих точок. Всі дії користувачів, процесів та служб повинні бути зафіксовані у системі централізованого логування для подальшого аналізу. Використання SIEM-систем дозволяє здійснювати кореляцію подій і виявляти потенційно небезпечні шаблони поведінки, що можуть вказувати на спроби вторгнення або розповсюдження шкідливого коду. Такий підхід також дозволяє забезпечити доказову базу для подальшого реагування на інциденти.

Варто впровадити принцип найменших привілеїв для користувачів кінцевих точок. Кожен співробітник повинен мати доступ лише до тих ресурсів і функцій, які необхідні йому для виконання посадових обов’язків. Обмеження прав доступу значно ускладнює роботу шкідливого програмного забезпечення в разі компрометації пристрою. Окрім цього, має бути організовано контроль запуску

додатків, щоб заборонити встановлення неперевіреного або несанкціонованого ПЗ, яке може містити вбудовані загрози.

Треба систематично проводити навчання співробітників з основ кібербезпеки. Людський фактор залишається однією з головних причин інцидентів, пов’язаних з поширенням шкідливого програмного забезпечення. Співробітники мають бути поінформовані щодо найпоширеніших методів атак, зокрема фішингу, соціальної інженерії та маскування шкідливого ПЗ під легітимне програмне забезпечення. Необхідно регулярно проводити тренінги, симуляції атак і навчальні кампанії, що допомагають виявляти слабкі місця в поведінці користувачів. Кожен працівник повинен чітко розуміти, які дії можуть призвести до інфікування кінцевої точки, і як правильно реагувати у випадку підозрілої активності. Освічені користувачі — це перша лінія захисту будь-якої організації.

ВИСНОВКИ

У роботі досліджено сутність проблеми виявлення шкідливого програмного забезпечення на кінцевих точках організації. Показано, що сучасні загрози стають дедалі складнішими, а традиційні антивірусні рішення не забезпечують належного рівня захисту. Основні виклики полягають у широкому використанні безфайлових атак, кейлогерів, руткітів та іншого складного ШПЗ, яке здатне маскувати свою присутність у системі.

Було проаналізовано підходи до виявлення шкідливого програмного забезпечення на кінцевих точках організації. Зокрема, розглянуто сигнатурний, поведінковий, евристичний та гібридний методи, а також інструменти на базі штучного інтелекту. Визначено, що найбільш ефективними є багаторівневі підходи, які дозволяють компенсувати недоліки кожного окремого методу й забезпечити вищу стійкість до нових загроз.

Проведено детальний аналіз існуючих рішень із виявлення шкідливого програмного забезпечення на кінцевих точках. Було охарактеризовано можливості таких платформ, як CrowdStrike Falcon, Microsoft Defender for Endpoint, SentinelOne Singularity та Velociraptor. Порівняння показало, що Velociraptor вирізняється гнучкістю, відкритим кодом та можливістю глибокого аналізу без значного навантаження на інфраструктуру.

Особливу увагу було приділено аналізу методів і засобів виявлення ШПЗ на базі Velociraptor. Розглянуто архітектуру платформи, принципи її роботи, мову запитів VQL та функціональність компонентів. Засвідчено, що Velociraptor забезпечує високий рівень кастомізації та є ефективним інструментом цифрової криміналістики й полювання на загрози у великих інформаційних системах.

Також було запропоновано порядок виявлення шкідливого програмного забезпечення на базі Velociraptor. Описано етапи розгортання системи, вибір артефактів для аналізу та механізм виявлення активності реального зразка ШПЗ (AgentTesla).

Було розроблено низку рекомендацій для підвищення ефективності виявлення ШПЗ на кінцевих точках організації. До них належать впровадження поведінкового аналізу, інтеграція з системами кіберрозвідки, використання sandbox-середовищ, регулярне сканування на вразливості, централізоване логування, обмеження прав доступу та підвищення обізнаності персоналу. Застосування цих заходів дає змогу суттєво підвищити рівень інформаційної безпеки організації.

Отже, виявлення шкідливого програмного забезпечення на кінцевих точках стало критично важливим елементом сучасної стратегії кіберзахисту, з огляду на постійне зростання складності та витонченості загроз. Velociraptor дає змогу організаціям ефективно реалізовувати сучасні підходи до виявлення загроз, використовуючи кастомні запити, механізми полювання на загрози, збір артефактів у реальному часі та централізоване управління кінцевими точками. Це сприяє впровадженню найкращих практик інформаційної безпеки та підвищенню рівня кіберстійкості.

Таким чином, використання Velociraptor у системі захисту кінцевих точок організації дозволяє ефективно виявляти шкідливе програмне забезпечення, швидко реагувати на інциденти та проводити детальний аналіз подій. Це рішення підвищує загальний рівень кіберзахисту, сприяє впровадженню сучасних підходів до безпеки та забезпечує стабільну роботу інформаційної інфраструктури організації в умовах зростаючих кіберзагроз.

ПЕРЕЛІК ПОСИЛАНЬ

1. Kirsch C. IDC: 70% of Successful Breaches Originate on the Endpoint | Rapid7 Blog. *Rapid7*. URL: https://old.rapid7.com/blog/post/2016/03/31/idc-says-70-of-successful-breaches-originate-on-the-endpoint/ (дата звернення: 30.04.2025).
2. Baker K. 12 Types of Malware + Examples That You Should Know | CrowdStrike. *CrowdStrike: We Stop Breaches with AI-native Cybersecurity*. URL: https://www.crowdstrike.com/en-us/cybersecurity-101/malware/types-of-malware/ (дата звернення: 30.04.2025).
3. DeepStrike. 50+ Malware Statistics 2025: Attacks, Trends and Infections. *DeepStrike*. URL: https://deepstrike.io/blog/Malware-Attacks-and-Infections-2025 (дата звернення: 30.04.2025).
4. CrowdStrike. 2025 Global Threat Report. *CrowdStrike: We Stop Breaches with AI-native Cybersecurity*. URL: https://www.crowdstrike.com/en-us/global-threat-report/ (дата звернення: 30.04.2025).
5. IBM X-Force 2025 Threat Intelligence Index. *IBM*. URL: https://www.ibm.com/thought-leadership/institute-business-value/en-us/report/2025-threat-intelligence-index (дата звернення: 30.04.2025).
6. Що таке шкідливе програмне забезпечення? Якої шкоди воно може завдати комп'ютеру чи смартфону?. *Державна служба спеціального зв'язку та захисту інформації України*. URL: https://cip.gov.ua/ua/faqs/sho-take-shkidlive-programne-zabezpechennya-yakoyi-shkodi-vono-mozhe-zavdati-komp-yuteru-chi-smartfonu (дата звернення: 01.05.2025).
7. Goncharov B. Top Malware Detection Techniques - Key Methods Explained. *AMATAS*. URL: https://amatas.com/blog/top-malware-detection-techniques-key-methods-explained/ (дата звернення: 01.05.2025).
8. Secure the Endpoint, Stop the Breach | CrowdStrike Endpoint Security. *CrowdStrike: We Stop Breaches with AI-native Cybersecurity*.

URL: https://www.crowdstrike.com/en-us/platform/endpoint-security/ (дата звернення: 02.05.2025).

9. Microsoft Defender for Endpoint - Microsoft Defender for Endpoint. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/defender-endpoint/microsoft-defender-endpoint (дата звернення: 03.05.2025).
10. Test Microsoft Defender Antivirus (Enterprise) 4.18 for Windows 11 (242214). *AV-TEST | Unabhängige Tests von Antiviren- & Security-Software*. URL: https://www.av-test.org/en/antivirus/business-windows-client/windows-11/april-2024/microsoft-defender-antivirus-enterprise-4.18-242214/ (дата звернення: 03.05.2025).
11. Singularity™ Endpoint Security | SentinelOne DE. *SentinelOne DE*. URL: https://www.sentinelone.com/platform/endpoint-security/ (дата звернення: 03.05.2025).
12. SentinelOne Sets the Standard with 100% Detection and 88% Fewer Alerts than Median Across All Vendors Evaluated in the 2024 MITRE ATT&CK® Evaluations: Enterprise. *SentinelOne*. URL: https://www.sentinelone.com/press/sentinelone-sets-the-standard-with-100-detection-and-88-fewer-alerts-than-median-across-all-vendors-evaluated-in-the-2024-mitre-attck-evaluations-enterprise/ (дата звернення: 03.05.2025).
13. Velociraptor Overview :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/docs/overview/ (дата звернення: 04.05.2025).
14. Welcome :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/ (дата звернення: 08.05.2025).
15. Triage and acquisition :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/docs/offline_triage/ (дата звернення: 08.05.2025).
16. Remote Uploads :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*.

URL: https://docs.velociraptor.app/docs/offline_triage/remote_uploads/ (дата звернення: 08.05.2025).

17. Event Queries and Endpoint Monitoring :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/blog/html/2018/11/09/event_queries_and_endpoint_monitoring/ (дата звернення: 08.05.2025).
18. Hunting :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/docs/hunting/ (дата звернення: 08.05.2025).
19. Quickstart Guide :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/docs/deployment/quickstart/ (дата звернення: 20.05.2025).
20. Deployment Overview :: Velociraptor - Digging deeper!. *Welcome :: Velociraptor - Digging deeper!*. URL: https://docs.velociraptor.app/docs/deployment/ (дата звернення: 22.05.2025).
21. Agent Tesla (шкідливе програмне забезпечення). *cert.gov.ua*. URL: https://cert.gov.ua/article/3028 (дата звернення: 23.05.2025).
22. A Brief Look at Crowdstrike and the Benefits it offers to Cloud Centric Customers. *RedSky*. URL: https://redskydigital.com/a-brief-look-at-crowdstrike-and-the-benefits-it-offers-to-cloud-centric-customers/ (дата звернення: 23.05.2025).
23. Hasayen A. P1: Microsoft Defender for Endpoint - Architecture | Ammar Hasayen. *AMMAR HASAYEN BLOG*. URL: https://blog.ahasayen.com/p1-microsoft-defender-for-endpoint-architecture/ (дата звернення: 23.05.2025).
24. Singularity™ XDR AI-Plattform – SentinelOne. *SentinelOne DE*. URL: https://www.sentinelone.com/platform/ (дата звернення: 23.05.2025).
25. Introducción a Velociraptor – D-Virus | DFIR. *D-Virus* \URL: https://dvirus.training/2020/06/09/introduccion-a-velociraptor/ (дата звернення: 23.05.2025).