Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
завідувач кафедри кібербезпеки
та захисту інформації
_____ Н.В. Лукова-Чуйко
“_____”_____ 2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
Дипломної роботи

магістра
(назва освітньо-кваліфікаційного рівня)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній рівень ***магістр***
(назва освітнього рівня)

кваліфікація *магістр кібербезпеки*
(код і назва кваліфікації)

на тему: Метод проактивного пошуку як захист від кіберзагроз

Виконавець: студент 2 **курсу, групи** КБм-21

	Папірна Ганна Костянтинівна
(підпис)	(прізвище ім’я по-батькові)

	Прізвище, ініціали	**Оцінка**	**Підпис**
Науковий керівник	Лукова-Чуйко Н. В.		
Рецензент			
Нормоконтроль	Фесенко А.О.		

Київ
2021

Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри
кібербезпеки та захисту інформації

_____ Лукова-Чуйко Н.В.

«_____»_____ 20___року

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

студенту КБм-21 Папірній Ганні Костянтинівні
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи Метод проактивного пошуку як захист від кіберзагроз

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол №2 від 08.10.2020 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень Процес проактивного пошуку кіберзагроз.

Предмет досліджень Керівний звід правил із проведення проактивного пошуку визначених кіберзагроз, створений засобами хмарної аналітики та спеціалізованими інструментами запитів.

Мета Розробка керівного зводу правил із проведення проактивного пошуку загроз: від ідентифікації кіберзагроз до написання запитів для пошуку, їх застосування на базі технічної платформи та оцінки ефективності результатів.

Вихідні дані для проведення роботи Інструменти та існуючі реалізації проведення процесу проактивного пошуку загроз.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна Набування подальшого розвитку методу проактивного пошуку загроз шляхом доведення його ефективності, порівняно із виявленням аналогічних загроз традиційними засобами захисту.

Практична цінність Створення чіткого керівного зводу правил із проведення проактивного пошуку загроз та підвищення якості результатів їх виявлення.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота повинна бути виконана у повному обсязі відповідно до теми та згідно із діючою законодавчою та нормативною базою у сфері технічного захисту інформації.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
1. Уточнення поставленої задачі.	12.10.2020 – 16.10.2020
2. Аналіз літератури.	19.10.2020 – 10.12.2020
3. Збір даних. Обґрунтування вибору рішення.	25.01.2021 – 12.02.2021
4. Дослідження процесу проведення проактивного пошуку загроз.	15.02.2021 – 26.02.2021
5. Пошук шляхів вдосконалення проактивного пошуку загроз.	05.04.2021 – 16.04.2021
6. Аналіз результатів проведеного дослідження.	19.04.2021 – 07.05.2021
7. Оформлення пояснювальної записки. Підготовка до захисту роботи	11.05.2021 – 17.05.2021

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект

Соціальний ефект

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Лукова-Чуйко Н.В.
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ Папірна Г.К.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 12.10.2020

Термін подання дипломної роботи до ЕК 17.05.2021

РЕФЕРАТ

Пояснювальна записка: 126 с., 36 рис., 5 табл., 2 додатки, 112 джерел.

Об’єкт дослідження: процес проактивного пошуку кіберзагроз.

Мета роботи: розробка керівного зводу правил із проведення проактивного пошуку загроз: від ідентифікації кіберзагроз до написання запитів для пошуку, їх застосування на базі технічної платформи та оцінки ефективності результатів.

Методи дослідження: структурний аналіз, порівняння, системний та історичний підходи, дедукція, абстрагування, експеримент, моделювання, формалізація.

У спеціальній частині дана характеристика розроблених запитів для проведення проактивного пошуку загроз.

У роботі досліджено процес виявлення кіберзагроз методом проактивного пошуку. Проведено аналіз наявних методів проведення пошуку, їх класифікацій. Запропоновано застосування мови KQL для створення запитів пошуку. Побудовано керівний звід правил із виявлення загроз. Розроблено запити проактивного пошуку.

Практичне значення роботи полягає у створенні чіткого керівного зводу правил із проведення проактивного пошуку загроз та підвищення якості результатів їх виявлення. Результати здійснених у дипломній роботі досліджень можуть бути використані у якості відправної точки в аналітиці кіберзагроз для практиків галузі.

Наукова новизна дослідження полягає у набуванні подальшого розвитку методу проактивного пошуку загроз шляхом доведення його ефективності, порівняно із виявленням аналогічних загроз традиційними засобами захисту.

Напрямки подальших досліджень можуть включати вдосконалення методу проактивного пошуку загроз, розширення типів кіберзагроз, які можуть бути виявлені даним методом, а також зменшення кількості помилкових спрацьовувань.

Ключові слова: ПРОАКТИВНИЙ ПОШУК ЗАГРОЗ, ІНДИКАТОРИ КОМПРОМЕТАЦІЇ, MITRE ATT&CK, ЦІЛЬОВІ КІБЕРАТАКИ, ТАКТИКИ І

ТЕХНІКИ АТАК, ВРАЗЛИВОСТІ НУЛЬОВОГО ДНЯ, THREAT HUNTING, SECURITY OPERATIONS CENTER.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ППЗ – проактивний пошук загроз

ТТП – тактики, техніки і процедури

EDR – Endpoint Detection and Response (технологія захисту кінцевих робочих станцій)

SOC – Security Operations Center (центр забезпечення безпеки)

KQL – мова запитів Kusto Query Language

ВСТУП

За даними провідної аналітичної компанії у сфері інформаційних технологій – SANS Institute, дослідження засобів захисту від кібератак в інформаційних системах та мережах показало, що 95% загроз інформаційної безпеки є відомими, і захиститися від них можна традиційними засобами захисту на кшталт антивірусів, міжмережевих екранів, тощо. До таких загроз відносяться такі види, як спам, атаки відмови в обслуговуванні, віруси, руткіти та інше класичне зловмисне програмне забезпечення. Решта 5% загроз – невідомі і найнебезпечніші. Вони складають 70% ризику для організацій через те, що їх дуже непросто виявити і тим більше від них захиститися. Прикладами таких загроз є віруси-шифрувальники, криптомайнери, тощо [1, с. 2].

У компанії, де організовані процеси управління інформаційною безпекою, 30% ризику реалізації відомих загроз можливо так чи інакше урегулювати за допомогою традиційного підходу управління ризиками, тобто: уникнути, прийняти (змиритися із можливими фінансовими або іміджевими втратами), зменшити (впровадити необхідні засоби захисту) або перекласти (наприклад, на постачальника послуг). Натомість захиститися від вразливостей нульового дня, цільових атак, фішингу, атак через ланцюжок поставок, а також від великої кількості інших атак вже набагато складніше. Наслідки від цих 5% загроз будуть набагато серйозніші, ніж наслідки від спаму або вірусів, від яких цілком здатне захистити сучасне антивірусне програмне забезпечення [2, с. 6; 3, с. 5].

Така статистика зумовила поштовх у розвитку засобів захисту від кібератак у бік розробки нової технології, що була б здатна протидіяти цим 5% найбільш серйозних і складних загроз.

Проактивний пошук загроз або Threat hunting – це новітній спосіб протидії кібератакам, який завдяки проактивному та ітеративному пошуку, дозволяє виявити складні загрози, які традиційні засоби захисту не здатні навіть помітити. Необхідно відзначити, що Threat hunting – це не якийсь конкретний програмний або апаратний

продукт і не пасивна активність. Проактивний пошук загроз – це, перш за все, переважно ручний процес з елементами автоматизації, в рамках якого аналітик, спираючись на свої знання і кваліфікацію, перевіряє великі обсяги інформації в пошуках ознак компрометації, відповідно до попередньо визначеної гіпотези про присутність певної загрози [4].

Така статистика обумовлює актуальність даної дипломної роботи через необхідність впровадження проактивного пошуку загроз до процесу управління ризиками інформаційної безпеки в організаціях будь-яких сфер діяльності.

Незважаючи на велику кількість наявних на ринку систем виявлення та захисту від кіберзагроз, більшість з них не включає інструменти для здійснення дослідницької проактивної діяльності із пошуку та протидії найбільш складним загрозам.

Дані останнього щорічного дослідження провідної дослідницької компанії, що спеціалізується на інформаційних технологіях, – Gartner, вказують на те що, у перспективі наступних двох – п’яти років, обов’язковими до впровадження є платформи, що містять функціонал проактивного пошуку загроз і реагування. Адже, стандартні технології, які реалізовують лише розпізнавання та генерацію сповіщень (інцидентів) про загрози більше не демонструють достатньої ефективності [5].

У згаданому дослідженні також підіймається проблема недостатньої кваліфікації персоналу організацій для впровадження механізму проактивного пошуку загроз до повсякденного процесу відділів інформаційної безпеки. У такому випадку, Gartner активно рекомендує співробітництво із організаціями, які надають кероване виявлення загроз як послугу (тобто, винести даний процес на аутсорс) [5].

Для більш великих підприємств, Gartner виводить необхідність у впровадженні повномасштабних платформ у період найближчих десятьох років, що охоплюватимуть функціонал керування безпекою, автоматизацію та реагування, шляхом інтеграції з іншими технологіями, зокрема проактивного пошуку загроз, які будуть автоматизувати весь процес керування кібербезпекою максимально спрощуючи роботу професійним командам центрів реагування на загрози [5].

Із вище наведеного випливає, що проактивний пошук загроз є технологією, що підходить для організацій різних розмірів та сфер діяльності та прогнозовано є націленою на довгу перспективу та стале закріплення в загальному процесі протидії кіберзагрозам.

Однак, спираючись на дослідження Gartner, головними стоп-факторами впровадження проактивного пошуку загроз в організаціях є: розрізненість підходів у формуванні запитів на пошук загроз; відсутність шаблонів механізмів створення таких процедур; неповні відомості щодо типів загроз, яким здатен протидіяти проактивний пошук; а також неточні відомості щодо вхідних даних, які мають бути базисом при створенні процесу проактивного пошуку загроз [5].

Проблемою розробки правил та шаблонів здійснення ефективного процесу проактивного пошуку загроз займалися у своїх дослідженнях [6; 7; 8; 9] такі вітчизняні науковці: А. В. Жилін, М. М. Худинцев, М. Ю. Літвінов, С. О. Гахов, І. В. Кифоренко та інші. Серед закордонних науковців значний вклад у розвиток проактивного пошуку загроз внесли: R. M. Lee, D. J. Bianco, S. Caltagirone, C. Crowley, та інші [10; 11; 12; 13; 14].

Однак, жодна із наявних на сьогодні публікацій не охоплює створення стандартного процесу проведення проактивного пошуку загроз. Сучасні дослідження стосуються створення загальних моделей проведення пошуку, мов написання запитів для виявлення загроз, але не охоплюють повного циклу створення пошуку для протидії конкретним загрозам із демонстрацією спрацьовувань та оцінкою ефективності.

Метою дипломної роботи «Метод проактивного пошуку як захист від кіберзагроз» є розробка керівного зводу правил із проведення проактивного пошуку загроз: від ідентифікації кіберзагроз до написання запитів для пошуку, їх застосування на базі технічної платформи та оцінки ефективності результатів.

Для досягнення даної мети необхідно вирішити наступні задачі:

1. провести аналітичний огляд наявних методів проведення проактивного пошуку загроз;

2. створити порівняльну характеристику наявних методів проведення проактивного пошуку загроз із виділенням їхніх особливостей та недоліків;

3. обґрунтувати необхідність створення керівного зводу правил із проведення проактивного пошуку загроз;

4. організувати лабораторне середовище для проведення наукового експерименту із проактивного пошуку загроз;

5. обґрунтовано підібрати набір кіберзагроз із розробкою критеріїв оцінки актуальності, які мають бути виявлені в процесі проактивного пошуку та провести їх емуляцію;

6. розробити та протестувати запити проактивного пошуку загроз для виявлення технік атак із зазначенням обмежень;

7. надати підтвердження визначення емульованих загроз пропонованими запитами проактивного пошуку загроз;

8. провести порівняння результатів виявлення загроз за допомогою проактивного пошуку та традиційних засобів захисту з огляду детектування технік різних груп критичності;

9. формалізувати керівний звід правил із проведення проактивного пошуку загроз із демонстрацією ефективності методу.

Об’єктом дослідження є процес проактивного пошуку кіберзагроз.

Предметом дослідження є керівний звід правил із проведення проактивного пошуку визначених кіберзагроз, створений засобами хмарної аналітики та спеціалізованими інструментами запитів.

Для досягнення мети дипломної роботи були використані наступні методи дослідження:

1. у розділі 1 було проведено аналіз наявних методів проведення проактивного пошуку загроз шляхом застосування структурного аналізу, методу порівняння та системного підходу;

2. у розділі 2 було здійснено розробку критеріїв оцінки актуальності, вибір та емуляцію кіберзагроз, а також створено та протестовано запити проактивного

пошуку загроз для протидії ним шляхом застосування історичного методу, дедукції, абстрагування та наукового експерименту;

3. у розділі 3 для здійснення порівняння результатів виявлення загроз за допомогою проактивного пошуку та традиційних засобів захисту та доведення ефективності керівного зводу правил із проведення проактивного пошуку загроз було застосовано методи математичного моделювання, а також формалізацію.

Наукова новизна дипломної роботи полягає в:

- удосконаленні процесу проактивного пошуку загроз шляхом застосування хмарної аналітики кіберзагроз із введенням критеріїв оцінки їх актуальності;
- адаптації мови запитів для обробки структурованих та неструктурованих даних для вирішення проблеми створення запитів проактивного пошуку загроз;
- набуванні подальшого розвитку методу проактивного пошуку загроз шляхом доведення його ефективності, порівняно із виявленням аналогічних загроз традиційними засобами захисту.

Практичне значення отриманих результатів забезпечується створенням чіткого керівного зводу правил із проведення проактивного пошуку загроз та підвищенням якості результатів їх виявлення, порівняно із аналогічними дослідженнями з даної тематики, для формування відправної точки аналітики кіберзагроз для практиків галузі.

Основні результати дипломної роботи доповідалися та обговорювалися на IV Міжнародній науково-практичній конференції «Інформаційні технології та взаємодії» (Київ, 2016); V Міжнародній науково-практичній конференції «Інформаційні технології та взаємодії» (Київ, 2018); IV Всеукраїнській науково-практичній конференції «Перспективні напрямки захисту інформації» (Одеса, 2018) та XVII Міжнародній науково-практичній конференції «Современные средства связи» (Мінськ, 2018); VII Міжнародній науково-практичній конференції «Інформаційні технології та взаємодії» (Київ, 2020). Основні положення дипломної роботи викладені у 9 наукових працях, серед яких: 1 стаття у науковому фаховому

виданні України, 2 статті у міжнародних наукових фахових виданнях, 6 – у матеріалах наукових конференцій, 4 з яких – міжнародні (ДОДАТОК А).

РОЗДІЛ 1
АНАЛІЗ НАЯВНИХ МЕТОДІВ ПРОВЕДЕННЯ ППЗ

1.1 Сутність та призначення ППЗ

Проактивний пошук загроз (далі – ППЗ) або Threat Hunting (дослівний переклад терміну з англійської – «полювання на загрози») – це комплекс попереджувальних заходів із пошуку ознак зловмисної активності в інфраструктурі інформаційних технологій (далі – ІТ), як поточних, так і ретроспективних (історичних), які дозволили обійти існуючі засоби захисту [15, с. 6].

Таке ухилення від засобів захисту може бути пов’язане з використанням нових, поліпшених або невідомих методів, уразливостями нульового дня або відсутністю адекватної технології виявлення загроз в організації. Хоча неповна або помилкова конфігурація технології виявлення або неправильна інтерпретація подій безпеки аналітиками також можуть бути причинами для ухилення, пошук загроз передбачає правильно запущений процес моніторингу безпеки в організації.

Оскільки дане поняття є відносно новим у сфері кібербезпеки, є доцільним пояснити його від зворотного, тобто описати, чим даний процес не є для уникнення змішування понять та технологій, спираючись на провідні дослідження галузі.

Отже, ППЗ – це не [15, с. 6-7; 16, с. 5-7]:

- форма тестування на проникнення (penetration testing), хоча ППЗ може привести до розуміння того, в якій області ІТ-інфраструктури проводити тестування на проникнення;
- постійний пошук індикаторів компрометації, хоча вони і використовуються в процесі;
- моніторинг безпеки, хоча результати ППЗ можуть використовуватися для забезпечення нового механізму виявлення, за яким слідує моніторинг безпеки;
- реагування на інциденти, хоча ППЗ може привести до розкриття інцидентів, тим самим ініціюючи процес реагування на інциденти;

- просто виконання запиту в інструменті безпеки, хоча автоматизація і запит даних є важливою частиною ППЗ. Іншими словами, якщо інструмент інформаційної безпеки (такий як антивірус, система попередження вторгнень та інші) може виконувати процес автономно, це не є проактивним пошуком загроз. Фахівці зі ППЗ повинні використовувати інструменти для підтримки своїх гіпотез та розслідувань, однак просте використання інструменту не здатне забезпечити цей процес повністю;
- процес, який дає гарантований результат. Не кожний проактивний пошук здатний розкрити зловмисника або призводить до створення нових механізмів виявлення. Це не обов'язково означає, що зловмисний вплив відсутній. Наприклад, дані, необхідні для проведення розслідування, можуть бути відсутніми, або під час пошуку досліджуваний індикатор компрометації не був присутній. Однак пошук завжди буде давати якийсь вторинний результат, наприклад, більш глибоке розуміння інфраструктури або ідентифікацію відсутніх даних;
- простий процес. ППЗ вимагає глибоких знань про інформаційне середовище і відмінного розуміння можливостей зловмисника. Якщо порівнювати із традиційним моніторингом безпеки, ППЗ є значно складнішим завданням.

Основна мета ППЗ – скоротити час, необхідний для пошуку слідів зловмисників, які вже скомпрометували IT-середовище. Якомога швидше виявивши ці сліди, можна звести до мінімуму вплив порушень на організацію. Прогалини у виявленні порушень – важливе поняття в контексті цієї мети [17, с. 2].

Як було зазначено вище, мета ППЗ – зменшити розрив між первинним зломом системи зловмисником і виявленням цього зловмисника в середовищі: розрив у виявленні порушення, також відомий як час очікування.

Розрив у виявленні порушень виникає зі здатності зловмисників ухилятися від механізмів виявлення. У міру того, як можливості виявлення продовжують розвиватися і розширюватися, кіберзлочинці будуть знаходити нові способи ухилення від цих заходів. Таким чином, з часом тактики, техніки і процедури (далі – ТТП) зловмисників будуть розвиватися, щоб гарантувати, що вони можуть ухилятися від виявлення і діяти непомітно в IT-середовищі [18, с. 165].

Таким чином, завжди буде існувати розрив між тим, що організація може виявити, і здатністю кваліфікованого зловмисника уникнути виявлення. Можливості зловмисника будуть відрізнятися для кожного зловмисника, а можливості виявлення будуть відрізнятися для кожної організації. Хоча зловмисники зазвичай мають можливості уникнути виявлення, в якийсь момент вони можуть запустити механізми виявлення, або тому, що ці механізми виявлення еволюціонували, або через людський фактор [19, с. 117].

Стабільна та ефективна програма ППЗ спрямована на відстеження ТТП і поведінки зловмисників і безперервне скорочення прогалин у виявленні порушень. Зокрема, пошук загроз фокусується на діях, які можуть залишитися непоміченими. Постійне розуміння стану механізмів виявлення необхідне для уникнення здійснення ППЗ на шкідливу активність, яка вже охоплена традиційними механізмами виявлення. ППЗ зосереджений на подіях, що виходять за рамки традиційних можливостей виявлення, і може виявити пропущені або невірно витлумачені події, що може бути використано для поліпшення виявлення і подальшого навчання аналітиків інформаційної безпеки [20, с. 4].

Виходячи із вище сказаного можливо вивести основні характеристики ППЗ [21, с. 5-8; 22, с. 3-4]:

- ініціативний характер. Фахівці із ППЗ проактивно шукають індикатори шкідливої активності в мережі, замість того, щоб чекати сигналів від традиційних механізмів виявлення для початку розслідування;
- допущення можливості випадку порушення. Проактивність не має сенсу, якщо фахівець з інформаційної безпеки вважає, що механізми запобігання та виявлення достатні для запобігання порушень. ППЗ передбачає, що порушення вже було здійснено, але ще не було ідентифіковано;
- розуміння атакуючого. Важливо розуміти мотивацію і хід думок зловмисника. Це важливі характеристики, що визначають, наскільки наполегливим і професійним є зловмисник;
- знаходження невідомого. ППЗ в основному зосереджений на відомих елементах для виявлення нових (пошук слідів невідомих зловмисників через відомі

ТТП), однак і може виявляти невідомі елементи в процесі (виявлення слідів невідомих зловмисників і раніше невідомих ТТП);

- творчість і повторюваність процесу. Процес ППЗ – це творчий процес. Якість процесу багато в чому залежить від творчих здібностей, досвіду і знань фахівця, який проводить пошук. Це ітеративний процес: пошук загроз може призвести до нових відкриттів і нових розслідувань; збір інформації також може призвести до нових припущень про поточний процес ППЗ;
- процес, керований даними. Для пошуку загроз потрібно багато даних. Кращими джерелами інформації для ППЗ прийнято вважати: інформацію про кінцеві точки, журнали брандмауера, журнали служби доменних імен (DNS) і т. п. Чим вище якість даних, тим вище ймовірність успіху розслідування. Ці дані повинні допомогти фахівцю проводити дослідження гіпотез ППЗ, а не ускладнювати ситуацію, додаючи інформаційний шум;
- базування на гіпотезах. Вивід гіпотез та їх доказ відіграють ключову роль в процесі ППЗ за будь-якою відомою наразі методологією;
- потребує командної роботи. Команда фахівців із ППЗ використовує загальний підхід і визначає, які загрози шукати. Команда також буде визначати пріоритети гіпотез на основі рівнів ризику, пов'язаних із загрозою. Зазвичай набір навичок для роботи в таких командах зводиться до загальних знань в області інформаційної безпеки, знання IT-середовища, знання методів аналізу, знання методів зловмисника і хороших комунікативних навичок.

1.2 Класифікація наявних методів із проведення ППЗ

Згідно із провідною американською компанією у сфері кібербезпеки та аналітики великих даних – Sqrrl, та провідних наукових досліджень у галузі кібербезпеки [23; 24, с. 635-637] існує загальний метод здійснення ППЗ, який має п’ять типів.

Перший тип – ППЗ на основі даних. Природною відправною точкою для стимулювання ППЗ є створення гіпотез на основі даних спостережень, тобто

перегляд даних, які вже є у наявності. Наприклад, журнали проксі, статистика трафіку, дані служби доменних імен тощо. Аналітики можуть використовувати будь-яке з джерел даних в якості основи для генерації гіпотез, створюючи запити або звіти, які ідентифікують аномальну поведінку [23; 25, с. 7].

Другий тип – ППЗ на основі інтелектуальної оцінки даних. Дані про загрози і аналітика можуть надати організаціям широкі можливості для ППЗ. На жаль, такий підхід є одним із найскладніших через те, що організації повинні усвідомлювати як різний рівень достовірності інформації, так і корисність і рідкісну природу збору внутрішньої інформації, заснованої на таких речах, як дії з реагування на інциденти. Допомогти аналітикам, які дотримуються даного підходу можуть графіки поведінки безпеки для отримання контексту загрози. Графік поведінки безпеки забезпечує критичні точки інтеграції, можливості обробки та аналізу, дозволяючи аналітику ефективно використовувати інформацію про загрози для управління і збагачення діяльності із ППЗ [23; 26, с. 4543].

Третій тип – ППЗ на основі сутності. Такий тип характеризується зосередженням навколо сутностей з високим ризиком та цінністю, таких як найважливіша інтелектуальна власність і мережеві ресурси. Зловмисники зазвичай націлені на певні активи або користувачів з високим ризиком в організації (наприклад, сервер, на якому проводяться дослідження і розробки, контролер домену або обліковий запис системного адміністратора). Все більше і більше організацій проактивно визначають, що це за активи, перш ніж противник зробить це за них [23; 27, с. 18].

Четвертий тип – ППЗ на основі ТТП. Даний підхід зосереджується на тому, що набагато важливіше за просто статичні індикатори (домени, IP-адреси, хеші тощо) є методи, тактики і процедури зловмисників. Ці спостереження є відмінним матеріалом для ППЗ, оскільки вони надають контекстуальні відправні точки, які більше підходять для людського аналізу, ніж для автоматичного розв’язання [23; 28, с. 30].

П’ятий тип – гібридний ППЗ. Насправді, будь-який успішний ППЗ буде поєднувати комбінації вищезазначених типів. Наприклад, ППЗ може бути

сформований за допомогою інформації про загрози навколо певного противника, яка інформує аналітика про типи ТТП, які може використовувати противник, і про критичні активи, на які він може націлюватися [23].

Існує також інший більш спрощений підхід, який поділяє ППЗ на два типи [29 с. 28; 30].

Структурований ППЗ – це ППЗ, заснований на гіпотезах: створюється гіпотеза, оцінюється об'єм та рамки ППЗ і згодом ППЗ виконується. Такий підхід може містити в своїй основі як інтелектуальну оцінку даних та аналіз сутностей, так і ТТП.

Неструктурований ППЗ – це ППЗ, керований даними. Потенційно шкідливу активність може виявити аналітик, який просто проглядає наявні дані у пошуках аномалій. Цей тип пошуку загроз не починається з гіпотези, не слідує заздалегідь визначеному шляху і, таким чином, вважається неструктурованим. Слід зазначити, що неструктурований ППЗ вимагає великих зусиль і з меншою ймовірністю дає цінні результати.

1.2.1 Метод застосування піраміди індикаторів

Виходячи із викладеного вище, ППЗ – це упереджувальна, ітеративна і орієнтована на людину ідентифікація кіберзагроз, які є внутрішніми по відношенню до ІТ-мережі і обходять існуючі заходи безпеки.

Аналітичний огляд літератури у галузі кібербезпеки демонструє, що тісно пов'язаною із процесом ППЗ є аналітика кіберзагроз – це процес збору, обробки та аналізу інформації про зловмисників в кіберпросторі з метою поширення дієвої інформації про загрози шляхом розуміння мотивів, можливостей і методів дій зловмисників для інформування про заходи щодо зниження кібербезпеки [31].

Для такої аналітики використовуються індикатори компрометації, які формують методи проведення ППЗ. На сьогоднішній день найпопулярнішою та найдієвішою, на думку провідних фахівців галузі, моделлю, що пояснює метод індикаторів компрометації в процесі ППЗ, є піраміда Девіда Біянко [32, с. 17-23].

Піраміда побудована із шести рівнів, які демонструють взаємозв'язок між типами індикаторів, які можливо використовувати для виявлення дій зловмисника, і ступенем «болю» (тобто, ймовірністю невдачі), яку вони заподіють йому, якщо фахівці зуміють попередити та заборонити використання цих індикаторів зловмисником.

Даний метод передбачає пошук індикаторів компрометації за їх ієрархією.

Хеш-значення знаходяться на найнижчому рівні піраміди. Більшість алгоритмів хешування обчислюють дайджест повідомлення за все введення і виводять хеш фіксованої довжини, унікальний для даного введення. Іншими словами, якщо вміст двох файлів відрізняється навіть на один біт, результуючі хеш-значення двох файлів будуть абсолютно різними. SHA1 і MD5 – два найбільш поширених приклади хешування цього типу.

З одного боку, хеш-індикатори – це найточніший тип індикатора, на який можна орієнтуватися. Шанси на те, що два різних файли мають однакові хеш-значення, настільки низькі, що можливо майже повністю виключити цю можливість. З іншого боку, будь-яка зміна файлу, навіть несуттєва, наприклад, перекидання біта в невживаному ресурсі або додавання нуля в кінець, призводить до зовсім іншого і непов'язаного хеш-значення. Таким чином, значення хешів так легко змінити, і їх так багато, що в багатьох випадках навіть не варто їх відстежувати [32; 33, с. 38].

На один рівень вище знаходяться IP-адреси – найфундаментальніший індикатор мережі. Атакуючим в значній мірі необхідно мати якесь мережеве з'єднання, щоб провести атаку, а з'єднання означає IP-адреси. Однак, будь-який достатньо професійний противник може змінити IP-адреси в будь-який зручний для нього час з дуже невеликими зусиллями. У деяких випадках, якщо зловмисники використовують анонімну проксі-службу, таку як Tor, вони можуть досить часто змінювати IP-адреси і навіть не звертати на це уваги. Ось чому блокування атак на базі IP-адрес є низькоефективним – якщо заблокувати зловмисника за одною IP-адресою, він зазвичай може відновитися, навіть не перериваючись [32; 34, с. 19].

Далі вище у піраміді знаходяться доменні імена. Їх трохи складніше змінити у порівнянні із IP-адресами, тому що для того, щоб вони працювали, вони повинні

бути зареєстровані, оплачені (навіть якщо вони вкрадені) і десь розміщені. Проте, існує велика кількість постачальників DNS із не суворими стандартами реєстрації (багато з яких безкоштовні), тому на практиці змінити домен не так вже й складно [32].

Посередині піраміди у жовтій зоні знаходяться артефакти мережі і хоста. Це перший рівень, на якому фахівець із ППЗ може почати здійснювати негативний вплив на супротивника. Якщо буде існувати можливість виявити індикатори на цьому рівні і реагувати на них, зловмисник буде змушений змінити налаштування і компіляцію своїх інструментів.

З технічної точки зору, мережевим артефактом може бути кожен байт, який проходить мережею в результаті взаємодії зловмисника. Однак на практиці, це означає ті частини активності, які можуть відрізняти шкідливу активність від активності законних користувачів. Типовими прикладами можуть бути шаблони URI, інформація, вбудована в мережеві протоколи, особливі значення HTTP User-Agent або SMTP Mailer і т. п. [32; 35, с. 27].

Артефактами хоста можуть бути спостереження, викликані діями зловмисників на одному або декількох хостах. Це можуть бути ключі або значення реєстру, які створені певними шкідливими програмами, файлами або каталогами, розміщеними в певних місцях або використовують певні імена, описи або шкідливі служби.

Блокування за артефактами мережі або хоста змушує атакуючих повернутися на декілька кроків назад і витратити час на з’ясування того, як було виявлено їх інструмент розвідки і виправлення його [32; 36, с. 14].

На передостанньому рівні знаходяться інструменти. Інструменти у даному контексті – це програмне забезпечення, що використовується противником для виконання своєї місії. В основному це програми, які вони встановлюють самостійно, а не програмне забезпечення або команди, які вже можуть бути встановлені на комп'ютері звичайного користувача. До таких програм відносяться: утиліти, призначені для створення шкідливих документів для цільового фішингу, бекдори, використовувані для встановлення зломщиків паролів, тощо [32; 37, с. 68-69].

Виявити індикатори на такому високу рівні можуть допомогти сигнатури антивірусів нового покоління або інші системи, які здатні знаходити варіанти одних і тих же файлів навіть із помірними змінами (протоколи зв'язку, хеш-значення тощо).

На цьому рівні можливо позбавити противника можливості використовувати один або декілька конкретних інструментів. Таким чином, зловмисникам буде необхідно витратити час на дослідження (знайти існуючий інструмент з такими самими можливостями), розробку (створити новий інструмент, якщо вони володіють відповідними знаннями та вміннями) і навчання (з'ясувати, як використовувати інструмент і оволодіти ним). Саме тому, блокування на рівні інструментів є одним із найбільш ефективних при протидії цільовим атакам [32; 38, с. 27-28].

Нарешті, на вершині знаходяться тактики, техніки і процедури (ТТП) зловмисників. Коли виявлення і реагування відбувається на цьому рівні, дія спрямована безпосередньо на поведінку зловмисників, а не проти їх інструментів. З точки зору ефективності, цей рівень є найідеальнішим. Якщо фахівець із ППЗ здатен досить швидко реагувати на підозрілі ТТП, то він змушує зловмисників здійснювати найбільш трудомістку із можливих дій: перенавчатися і опановувати нову поведінку [32; 39, с. 125-127].

Оскільки, винаходити будь-що заново є дуже складною задачею, то переважна більшість атакуючих відступає, якщо стикається із таким глибоким розумінням індикаторів компрометації.

Із вище наведеного можна зробити висновок, що процес успішного застосування методу піраміди індикаторів при здійсненні ППЗ полягає на зосередженні на найвищому рівні піраміди, тобто ТТП, задля зниження ймовірності досягнення мети атакуючим із найбільшою ефективністю. У разі застосування зловмисником нетривіальних дій або відсутності даних ТТП у відомих фреймворках, є сенс здійснювати блокування на нижчих рівнях піраміди до моменту виявлення ТТП.

1.2.2 Метод застосування «петлі» ППЗ

Відповідно до підходу провідної американської компанії у сфері кібербезпеки та аналітики великих даних – Sqrrl, та сучасних наукових досліджень у галузі кібербезпеки [40; 41, с. 1825-1827], весь процес ППЗ можна звести до чотирьох основних етапів, що циклічно повторюються.

Такий метод носить назву «петлі ППЗ» (Hunting loop) і призначений для уникнення потенційно неефективних процесів ППЗ і створення формалізованого процесу. Згідно із підходом Sqrrl, метою ППЗ повинно бути якомога більш швидке і ефективне подолання петлі. Чим ефективніше буде виконано ітерації, тим якісніше буде можливість автоматизувати нові процеси і перейти до пошуку нових загроз.

Процес ППЗ починається із формування питань щодо того, як зловмисник може отримати доступ до мережі організації. Потім ці питання необхідно розбити на конкретні і вимірні гіпотези, які визначають, які загрози можуть бути присутніми в мережі і як їх можна ідентифікувати.

Гіпотези не можуть бути згенеровані за допомогою інструментів, замість цього вони повинні бути отримані зі спостережень фахівця, заснованих на аналітиці кіберзагроз, ситуаційної обізнаності або знанні предметної області [42, с. 5].

Гіпотези також повинні бути перевірені, тобто фахівці із ППЗ повинні мати в своєму розпорядженні необхідну видимість даних і інструменти для пошуку передбачуваних свідчень зловмисної діяльності. Велика різноманітність типів даних дозволяє досліджувати більше методів, а більша кількість джерел даних розширює арену для проведення ППЗ. Гіпотези, як правило, зосереджені на виявленні конкретного джерела загрози, інструменту або техніки [40; 43, с. 94].

Після того, як спостереження привели до створення гіпотез, їх необхідно перевірити з використанням всіх відповідних інструментів і методів, що є в розпорядженні фахівців із ППЗ. Видимість даних повинна бути максимізована за рахунок збільшення охоплення збору даних в централізованому сховищі, а типи даних повинні бути інформативними та різноманітними. Існуючі інструменти керування інцидентами можуть використовуватися для запиту даних, від базового

пошуку до більш просунутих методів, а візуалізація може допомогти у виявленні аномалій і незвичних шаблонів поведінки [40].

Досить ефективним методом на даному етапі є Linked data (Пов’язані дані) – це метод публікації структурованих даних, що дозволяє пов’язувати їх і шукати підтвердження гіпотез за допомогою семантичних запитів. Аналіз пов’язаних даних особливо ефективний при викладі даних, необхідних для вирішення гіпотез, в зрозумілій формі, і тому є важливим компонентом ППЗ. Пов’язані дані можуть навіть допомогти розставити пріоритети і спрямувати візуалізацію, полегшуючи пошук у великих наборах даних і використання більш потужної аналітики. Методи аналізу як вихідних, так і пов'язаних даних повинні використовуватися для виявлення закономірностей в розрізнених наборах даних, задля виявлення дій зловмисників [44, с. 70-73].

Загалом можна виділити чотири типи технік, які можуть використовувати фахівці із ППЗ на даному етапі [45, с. 67-68]:

- пошук – це найпростіший метод запиту зібраних даних. Критерії пошуку повинні бути досить конкретними, щоб результати не були некерованими, але в той же час досить загальними, щоб не пропустити жодних дій зловмисників. При необхідності в запитах можна використовувати символи узагальнення (також відомі як wildcards);
- кластеризація – це форма статистичного аналізу, який відокремлює групи (кластери) схожих точок даних від більшого набору на основі конкретних характеристик, тоді як угруповання визначає, коли кілька унікальних точок даних з'являються разом на основі певних критеріїв, наприклад, кілька подій, що відбуваються в конкретному часовому вікні. Основна відмінність полягає в тому, що для угруповання необхідний явний набір точок даних в якості вхідних даних. Однак, обидва способи (кластеризація і угруповання) корисні для виявлення аномалій;
- підрахунок стека або накопичення – це своєрідний додаток частотного аналізу до великих наборів даних для виявлення аномалій;
- машинне навчання – використовує алгоритми і статистичні моделі для поступового підвищення продуктивності конкретного завдання; для ППЗ – це

виявлення аномальних даних, які можуть вказувати на дії зловмисників. При керованому машинному навчанні набір навчальних даних вводиться в алгоритм, причому кожна точка даних позначена бажаним результатом, тобто як нормальні, так і аномальні дані чітко визначені. У разі використання неконтрольованого машинного навчання (або навчання «без вчителя») на вхід подаються немарковані дані, тому алгоритм замість цього використовує такі методи, як кластеризація і угруповання, для категоризації вихідних даних.

Проходження другого етапу за допомогою інструментів дозволяє розкривати нові шкідливі шаблони поведінки і ТТП. Даний етап є одним із найкритичніших в рамках всього циклу.

Прикладом цього процесу може бути те, що попереднє розслідування показало, що обліковий запис користувача поводився аномально, причому він спочатку був зламаний за допомогою експлойта, націленого на стороннього постачальника послуг організації. Цей ТТП (початковий злом через сторонню систему за допомогою певного типу шкідливого програмного забезпечення) повинен реєструватися, передаватися (як всередині, так і ззовні організації) і відслідковуватися в контексті більшої кампанії атаки. Зв'язки пов'язаних даних також контекстуально покажуть, які інші облікові записи були пов'язані зі зламаною сторонньою службою [46, с. 231].

Саме завдяки цьому етапу попередніх процедур ППЗ і наповнюються і вдосконалюються фреймворки ТТП.

Четверта фаза циклу формує основу для інформування та збагачення автоматизованої аналітики. Ні в якому разі не можна допустити упущення загроз, важливо автоматизувати їх за допомогою аналітики, щоб команда із ППЗ могла і далі зосередитися на наступних процедурах. Це можна здійснити різними способами, в тому числі розробити пошук «за замовчуванням» для регулярного виконання, створити нову аналітику з використанням таких інструментів, як Sqrrl, Apache Spark, R або Python, або навіть надати зворотний зв'язок контрольованому алгоритму машинного навчання, що підтверджує, що ідентифікований шаблон поведінки аномальний та шкідливий [40; 47, с. 46-47].

Розширення аналітики може приймати і більш просту форму – надати новий індикатор компрометації для зіставлення або написати нове правило для системи керування інцидентами для реактивного виявлення. Чим швидше можна автоматизувати ППЗ, тим менше повторень буде вимагатися від фахівців і тим швидше їх навички можуть бути використані для перевірки нових гіпотез.

1.2.3 Метод застосування тріади

Відповідно до дослідження провідного аналітика та фахівця із кібербезпеки Деніела Аккакі та інших науковців галузі [32, с. 9-17; 48, с. 157], успішна процедура ППЗ містить три компоненти, тобто базується на методі тріади: люди, процеси і технології.

Перший компонент – це люди, тобто фахівці у сфері кібербезпеки, що мають відповідні навички. У зазначеному вище дослідженні, пропонується наступний набір навичок, які повинен мати фахівець для успішного проведення ППЗ [49, с. 8-11]:

- аналітичне мислення – це, без сумніву, найважливіша якість, якою може володіти аналітик. Фахівець повинен вміти робити аргументовані припущення і намічати новий курс;
- вміння аналізувати журналу аудиту системи (логи). Журнали служб і пристроїв – це лише пара найбільш важливих і недостатньо використовуваних джерел інформації для будь-якого відділу безпеки. Можливість аналізувати журнали на предмет аномалій і перемикатися між джерелами даних, щоб побачити загальну картину, є ключовою компетенцією;
- володіння навичками мережевої криміналістики – здатність читати і розуміти дані захоплених мережевих пакетів і визначати шкідливий характер мережевого трафіку;
- знання мережевої архітектури – розуміння різних мережевих пристроїв і того, як вони працюють в IT-середовищі;

- розуміння життєвого циклу атакуючого. Розуміння різних подій, які відбуваються на будь-якому етапі життєвого циклу атаки, підготовлює аналітиків до поділу і розстановки пріоритетів їх результатів і дій;
- володіння роботою із інструментами інформаційної безпеки – це велика область, але на базовому рівні розуміння того, як агрегатори журналів отримують дані, а також функції інструментів аналізу захоплення пакетів є достатніми для аналітика;
- знання архітектури операційних систем. Різні операційні системи представляють різні вектори атак. Дуже важливо добре розбиратися в операційних системах на базі Windows і Linux;
- розуміння базових методів атак. Комплекти експлойтів, шкідливе програмне забезпечення, фішинг і неправильна конфігурація – розуміння того, як зловмисник намагається проникнути до мережі, є ключем до пошуку індикаторів компрометації.

Другий компонент процедури ППЗ – це процеси, адже метою зрілого процесу безпеки повинна бути автоматизація значної частини виявлення загроз за допомогою надійних правил і своєчасного сповіщення, щоб дати аналітикам більше часу для безпосереднього проведення ППЗ.

Процеси повинні бути спроектовані з урахуванням бажання розуміти не тільки, які дані вже є у наявності, але також і джерела даних, які відсутні або неправильно налаштовані, адже неможливо захистити те, про існування чого невідомо [50, с. 5-6].

Отже, даний компонент полягає у зборі необхідної інформації і її подальшій аналітиці. Як відомо, дані самі по собі не рівні інтелекту, тому простий збір журналів аудиту (логів) буде створювати шум.

Виходячи із досвіду фахівців у даній сфері, нижче наведені типи журналів, які є важливими для збору та здатні надати репрезентативні дані, які можна використати під час ППЗ: база даних управління конфігурацією (CMDB); журнали додатків або служб; служба отримання мережевої конфігурації (DHCP); проксі; веб-сервер і сервер додатків; каталоги Active Directory; служба доменних імен (DNS);

брандмауер додатків; міжмережевий екран; антивірус; операційна система (наприклад, Windows Event і UNIX Syslog); гіпервізор віртуальної машини, тощо [51, с. 859].

Однак, простого збору логів недостатньо. Для того, аби дані були здобуті для проведення ефективного ППЗ, необхідно застосувати контекст, щоб зробити ці дані дієвими. Чим більше контексту застосовано, тим вище точність подальших висновків. Інформація може надходити з багатьох джерел, як внутрішніх, так і зовнішніх по відношенню до організації.

Прикладами внутрішніх джерел контексту можуть бути: минулі інциденти; спроби розвідки інфраструктури організації (наприклад, сканування портів); конкретні загрози характерні для напряму бізнесу і галузевих вертикалей; загрози інтелектуальної власності клієнтів організації, тощо. Прикладами зовнішніх джерел контексту можуть бути: оплачувані інформаційні канали; дані розвідки із відкритих джерел – Open Source Intelligence (OSINT); співпраця із правоохоронними органами, тощо [52, с.147-150].

Як вже було зазначено вище, крім надійного запобігання, захист організації від поточних загроз вимагає розвитку можливостей виявлення та реагування. Існують різні моделі методологій зловмисників, з яких можливо почати закладати основу для стратегії ППЗ. Найвідомішими збірниками таких методологій є: The Lockheed Martin Cyber Kill Chain; The Mandiant Attack Lifecycle; The MITRE ATT&CK Framework [53, с.17-21].

Третім і завершальним компонентом даного методу із проведення ППЗ є технології.

На сьогоднішній день існує безліч технологій інформаційної безпеки, які здатні забезпечити допомогу в процесі ППЗ. Однак, провідні фахівці галузі схильні звужувати набір технологій до наступних найнеобхідніших [54, с. 43-45]:

- системи класу SIEM (Security Information and Event Management) – дозволяють здійснювати моніторинг інформаційних систем, аналізувати події безпеки в режимі реального часу, наприклад, що відбуваються на робочих станціях, мережевих пристроях, засобах захисту інформації та інших елементах

інфраструктури. Зібрані та проаналізовані ними дані допомагають виявити інциденти або аномалії, що залишилися непомітними для спеціалізованих засобів захисту;

- системи класу EDR (Endpoint Detection and Response) – є альтернативою традиційним антивірусним рішенням і забезпечують сучасний захист кінцевих точок із адаптацією до сучасного ландшафту складних загроз. Такі системи включають як функціонал із виявлення комплексних атак, спрямованих на кінцеві точки, так і здатні оперативно реагувати на знайдені інциденти;
- системи класу NTA (Network Traffic Analysis) – це нова категорія систем мережевої безпеки, призначена для перехоплення потоків трафіку і виявлення ознак складних, найчастіше цільових атак.

Особливо показовим є те, що всі ці три типи систем (за умови підбору актуальних рішень) підтримують можливість безшовної інтеграції та постійного обміну даними. Тобто, у такій схемі NTA відповідає за видимість інформації, переданої по мережі, EDR поставляє відповідні відомості від кінцевих точок, а SIEM агрегує журнали подій.

Фахівці галузі кібербезпеки відмічають, що зазначені вище системи є технічним базисом при побудові сучасного SOC (Security Operations Center). SOC – це спеціалізований центр моніторингу та оперативного реагування на інциденти інформаційної безпеки. Такий центр являє собою групу експертів із захисту інформації, що відповідає за постійний контроль і аналіз стану безпеки організації, використовуючи комбінацію технологічних рішень і діючи в рамках чітко вибудованих процесів. SOC зазвичай укомплектовані аналітиками і інженерами в галузі безпеки, а також сервіс-менеджерами, які забезпечують оперативну взаємодію з клієнтом. SOC покликаний відслідковувати активність в мережах, на серверах і робочих станціях, в базах даних, додатках, веб-сайтах та інших системах, виявляючи аномальні і зловмисні дії, які можуть вказувати на інцидент безпеки або компрометацію даних [55, с. 134].

Важливо зазначити, що найчастіше процеси ППЗ прагнуть впровадити організації, що вже мають власний SOC або користуються такими послугами за

допомогою аутсорсингу. Таким чином, можна зробити висновок про те, що планування та впровадження процедури ППЗ за методом тріади до повсякденного процесу забезпечення інформаційної безпеки можуть собі дозволити організації із високим рівнем зрілості процесів безпеки, що вже мають налагоджені процедури та технології попередження загроз і готові перейти на більш високий рівень, а саме на рівень проактивної протидії загрозам.

1.3 Порівняльна характеристика існуючих методів із проведення ППЗ

Аналітичний огляд літератури (див. 1.2) показав, чотири основних методи проведення ППЗ. З метою обґрунтування мети даної дипломної роботи, було створено порівняльну характеристику існуючих методів, задля виявлення їхніх недоліків та перспективних напрямків розвитку (Таблиця 1.1).

Таблиця 1.1

Порівняльна характеристика існуючих методів ППЗ

Метод	Ключові особливості та переваги	Недоліки

Загальний метод Sqrrl	• Враховує різноманіття вхідних даних для проведення ППЗ: дані, які вже є у наявності, наприклад, журнали аудиту (логи); дані про загрози та минулі інциденти; сутності з високим ризиком або цінністю (активи); дані ТТП; гібридні комбінації даних; • враховує достатньо широкий спектр можливостей формування гіпотез ППЗ: створення запитів або звітів, які ідентифікують аномальну поведінку; графіки поведінки безпеки для отримання контексту загрози; проактивне визначення цільових активів; спостереження за контекстуальними відправними точками; комбінований підхід; • передбачає використання структурованого та неструктурованого підходів; • включає відомості щодо ймовірностей успішного результату для кожного із класів ППЗ.	• Не містить чіткого порядку дій (керівництва) для формування відправної точки всього процесу ППЗ; • не містить конкретних показників ефективності кожного із класів ППЗ; • не містить правил формування гіпотез (запитів) ППЗ; • не містить чіткого переліку вхідних даних для проведення ППЗ (конкретні типи логів тощо); • не містить переліку загроз (принципу їх формування), для яких є застосовним даний метод; • не містить рекомендацій щодо інструментарію (програмного забезпечення, фреймворків атак, мов створення запитів ППЗ тощо) для успішного отримання результатів процесу ППЗ; • не містить роз’яснень щодо критерію доцільності проведення ППЗ для певних типів загроз.

продовження табл. 1.1

Метод застосування піраміди індикаторів	• Розширює стандартне поняття процесу ППЗ за допомогою досвіду тісно пов'язаного процесу аналітики кіберзагроз; • містить чіткі роз'яснення щодо місця застосування індикаторів компрометації в рамках всього процесу ППЗ; • за кожним із індикаторів компрометації закріплений чіткий перелік даних, що можуть становити потенційний інтерес під час здійснення процесу ППЗ; • містить відомості щодо фундаментальності кожного із індикаторів компрометації та їх ефективності при різних типах реалізації атак.	• Не містить чіткого порядку дій (керівництва) для формування відправної точки всього процесу ППЗ; • зациклений лише на статичних індикаторах компрометації та не враховує динамічних показників, що стосуються атак; • деякі типи індикаторів компрометації легко підробити, що робить даний метод слабшим у порівнянні із іншими представленими; • не містить правил формування пошукових запитів ППЗ із зазначенням гіпотез, щодо яких проводиться перевірка; • не містить конкретних показників ефективності кожного із рівнів індикаторів компрометації; • не містить рекомендацій щодо інструментарію (програмного забезпечення, фреймворків атак, мов створення запитів ППЗ тощо) для успішного отримання результатів процесу ППЗ.
Метод застосування «петлі» ППЗ	• Містить чітку етапність (порядок дій) для формування відправної точки всього процесу ППЗ; • передбачає чіткі правила формування гіпотез (запитів) ППЗ; • включає механізм верифікації гіпотез ППЗ, задля гарантування успішності процесу на певному етапі; • містить методики структурування даних, що дозволяє пов'язувати їх і шукати підтвердження гіпотезам; • містить типи технік, придатних для аналізу та оцінки даних ППЗ; • включає процедури контекстуального пов'язування отриманих даних; • містить рекомендації щодо конкретного програмного забезпечення, інструментів для розширення аналітики.	• Не включає роз'яснень щодо критерію доцільності проведення ППЗ для певних типів загроз; • не формує уявлень щодо конкретних типів даних, журналів аудиту (логів), що є необхідними для збору в процесі ППЗ; • не включає опису (шаблонів) формування питань щодо того, як зловмисник може отримати доступ до мережі організації; • не містить конкретних показників ефективності результатів ППЗ; • не містить градації та типів загроз, для яких буде придатним даний метод ППЗ; • не містить чітких вимог щодо компетенцій фахівців із ППЗ.

продовження табл. 1.1

Метод застосування тріади	• Базується на синергії компетентності, вмінь та навичок фахівців, сучасних процесів і технології автоматизації та спрощення ППЗ; • містить чіткі вимоги щодо компетенцій, кваліфікацій та вмінь фахівців із ППЗ; • містить конкретний перелік репрезентативних даних, які можна використати під час ППЗ; • містить досить повний список внутрішніх джерел контексту, задля полегшення та збагачення всього процесу ППЗ; • містить перелік класів продуктів (технологій), що спрощують процес ППЗ, підвищують його ефективність та здійснюють уточнюючу роль.	• Не містить чіткого порядку дій (керівництва) для формування відправної точки всього процесу ППЗ; • не містить конкретних показників ефективності результатів ППЗ; • характеризується надмірним узагальненням та спрощенням процесу ППЗ (всього три компоненти); • не містить чітких роз'яснень місця відомих методологій (фреймворків) світових практиків у даному підході; • порядок впровадження процесів, в контексті даного методу, не деталізовано; • не містить типів загроз, для яких пропонуються джерела контексту, технології тощо.

Результати порівняння існуючих методів із проведення ППЗ демонструють два головні недоліки, що є спільними для всіх методів: відсутність чіткого порядку дій (керівництва) для проведення ППЗ та відсутність конкретних показників ефективності. Додатково, в існуючих методах незрозуміла роль світових фреймворків практиків у сфері кібербезпеки, що вказують на конкретні типи загроз, які можна виявити за допомогою ППЗ, адже у методах не зазначено чітких практичних дій для пошуку того чи іншого типу загроз.

З огляду на це, для реалізації мети даної дипломної роботи, вирішено створити керівний звід правил із проведення ППЗ.

Під «керівним зводом правил» мається на увазі практичний посібник, який:

- чітко формулює класи та типи загроз, які можна виявити за допомогою ППЗ;
- містить конкретний синтаксис написання запитів (гіпотез) ППЗ, які підходять до обраних типів загроз;
- містить приклади виявлення загроз за допомогою запитів ППЗ;
- має обґрунтовану оцінку ефективності виявлення загроз із ППЗ.

1.4 Висновки за розділом 1

У даному розділі було проведено аналіз існуючих підходів та методів проведення ППЗ та отримано наступні результати:

1. проведено аналітичний огляд літератури та наукових досліджень у галузі ППЗ, відповідно до мети дипломної роботи. Виділено загальний метод та класифікацію проведення ППЗ за структурованим та неструктурованим підходами;

2. виділено огляд трьох специфічних методів проведення ППЗ, які зарекомендували себе у науковій та практичних площинах. Виокремлено ключові елементи кожного із методів, їх особливості та рекомендовані етапи процесу ППЗ;

3. створено порівняльну характеристику існуючих методів проведення ППЗ із виділенням їхніх недоліків та неточностей напрямків розвитку.

4. обґрунтовано створення власного керівного зводу правил із проведення ППЗ для реалізації більш ефективного підходу до даної задачі.

РОЗДІЛ 2
РОЗРОБКА КЕРІВНОГО ЗВОДУ ПРАВИЛ ІЗ ПРОВЕДЕННЯ ППЗ

2.1 Організація лабораторного середовища для проведення наукового дослідження

Лабораторне середовище для проведення наукового дослідження зі створення керівного зводу правил із проведення ППЗ організовано на базі хмарної платформи Microsoft Azure.

Хмарні служби Azure є прикладом концепції платформи як послуги (Platform as a Service – PaaS).

PaaS – модель надання хмарних обчислень, при якій підписник отримує доступ до використання інформаційно-технологічних платформ: операційних систем, систем управління базами даних, сполучного програмного забезпечення, засобів розробки і тестування, розміщеними у хмарного провайдера [56].

Технологія Microsoft Azure призначена для підтримки масштабованих та надійних віртуальних середовищ. На віртуальних машинах, які використовують хмарні служби Azure, можна встановити власне програмне забезпечення, а потім отримати віддалений доступ до нього.

Перевірка і написання запитів гіпотез із ППЗ організовано на базі хмарної служби Microsoft Defender for Endpoint.

Microsoft Defender for Endpoint – це корпоративна платформа для захисту кінцевих точок, розроблена для запобігання, виявлення, дослідження та реагування на складні загрози в корпоративних мережах. Рішення є представником серед технологій класу Endpoint Detection and Response. Рішення використовує технологію, вбудовану до Windows 10 та Windows Server 2019 [57].

Дана хмарна служба включає [57]:

- датчики поведінки кінцевої точки (встановлюються безпосередньо на клієнтських або серверних машинах), які збирають і обробляють сигнали поведінки

операційної системи і відправляють ці дані датчиків на приватний, ізольований, хмарний екземпляр Microsoft Defender;

- хмарну аналітику безпеки: використання великих даних, машинного навчання та унікальних характеристик Microsoft для екосистеми Windows та Інтернет-ресурсів;
- інструменти для проведення ППЗ на базі запитів Kusto Query Language (KQL).

KQL – це мова запитів, призначених тільки для читання (read-only), яка обробляє дані і забезпечує повернення результатів. Запит викладено у вигляді звичайного тексту з використанням моделі потоку даних, розробленої для полегшення читання, створення та автоматизації синтаксису. У запиті використовуються об'єкти схеми, які організовані в ієрархію, аналогічну SQL: бази даних, таблиці та стовпці [58].

Запит складається з послідовності операторів запиту, причому принаймні один оператор є оператором табличного вираження, який представляє собою оператор, який виробляє дані, впорядковані у вигляді таблиці, що складається з стовпців і рядків. Оператори табличного вираження запиту видають результати запиту [58].

Синтаксис оператора табличного вираження має потік табличних даних від одного оператора табличного запиту до іншого, починаючи з джерела даних (наприклад, таблиці в базі даних або оператора, який виробляє дані), а потім проходить через набір операторів перетворення даних, які пов'язані один з одним [58].

Для тестування роботи запитів гіпотез із ППЗ в лабораторному середовищі розгорнуті віртуальні машини, які при кожному наступному перезавантаженні налаштовувалися як нові клієнти для хмарної служби Microsoft Defender for Endpoint. Така установка дозволяє ізолювати дані тестування від вже протестованих технік і спростити оцінку можливостей виявлення.

Віртуальні машини у лабораторному середовищі, на яких тестуються атаки та запити із ППЗ, складаються із домену Windows з одним контролером домену (сервером) і одним клієнтом. Всі віртуальні машини є екземплярами Azure «Standard

B4MS» (базова продуктивність процесора віртуальної машини – 90%; максимальна продуктивність процесора віртуальної машини – 400%) з чотирма віртуальними процесорами, 16 Гб оперативної пам'яті і розміром жорсткого диску 32 Гб. На сервері встановлений Windows Server 2019 Datacenter, а на клієнті – Windows 10 1903.

Схема створеного лабораторного середовища наведена на Рисунку 2.1.

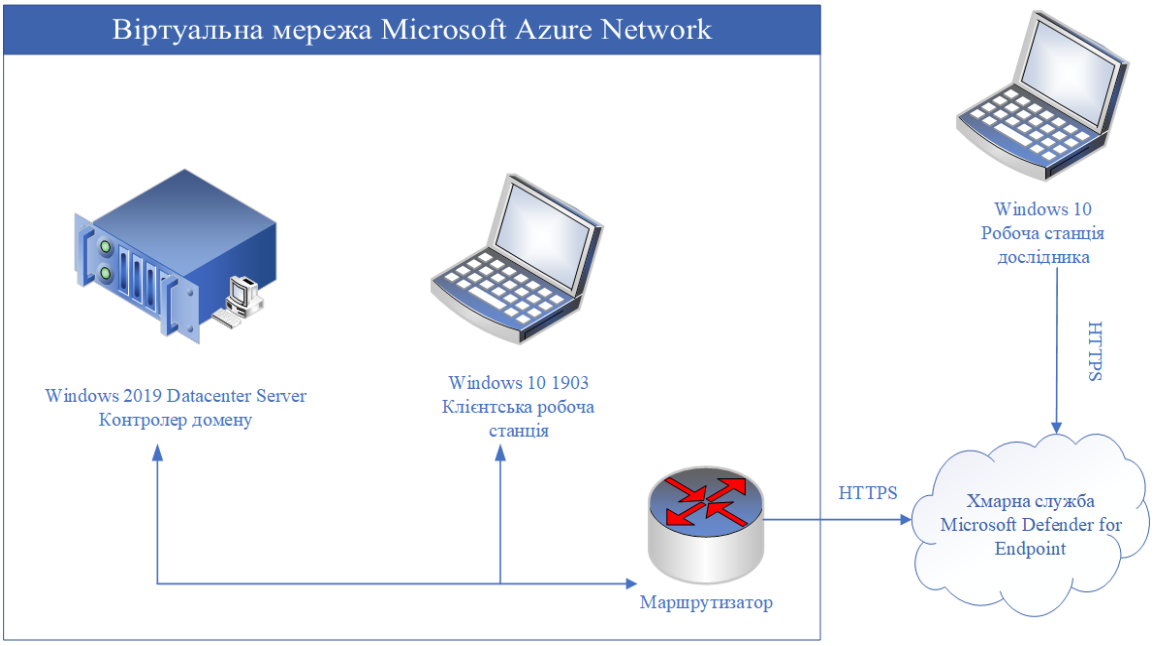


Рисунок 2.1 – Схема лабораторного середовища

Всі машини знаходяться в ізольованому домені у власній групі ресурсів у власній віртуальній мережі. Хости доступні тільки в рамках однієї підмережі, у них немає публічних IP-адрес, призначених ним через Azure, але є доступ до Інтернету. Після створення віртуальних машин був запущений додатковий сценарій для підключення віртуальних машин до домену.

2.2 Здійснення оптимальних налаштувань та виділення обмежень лабораторного середовища

З метою успішного здійснення наукового дослідження були внесені наступні зміни в стандартні екземпляри віртуальних машин в Azure:

- віддалене керування Windows (WinRM) увімкнено для всіх хостів;

- для політики виконання PowerShell (об’єктно-орієнтований програмний компонент і мова написання скриптів з інтерфейсом командного рядка, що надає широкі можливості для конфігурації операційних систем сімейства Windows) задано значення «Bypass (обхід)»;
- налаштований брандмауер хостів для дозволу використання віддаленого доступу до файлів за допомогою протоколу SMB;
- відключені сповіщення при налаштуванні контролю облікових записів (UAC);
- доступ за протоколом віддаленого доступу робочих столів (RDP) увімкнено для всіх хостів.

Для прозорого проведення наукового дослідження всі превентивні дії хмарної служби Microsoft Defender for Endpoint були відключені і налаштовані тільки на моніторинг і генерацію попереджень. Така конфігурація застосовується для уникнення блокування перших етапів методів атаки і запобігання втраті можливостей аналізу зібраних даних і попереджень на більш пізніх етапах.

В той же час функціонал Microsoft Defender на тестових віртуальних машинах було увімкнено, але переведено у режим виключно моніторингу за допомогою налаштувань групової політики домену. Увімкнення даного функціоналу забезпечує зв’язок між тестовими машинами і хмарною службою Microsoft Defender for Endpoint, на базі якої виконується тестування гіпотез ППЗ.

2.3 Аналіз актуальних ТТП

Для побудови ефективного керівного зводу правил із проведення ППЗ, необхідно визначити актуальні ТТП, на пошук яких має бути спрямоване дослідження. Для виконання даної мети, задля вибору найбільш актуальних та оптимальних ТТП, було здійснено аналіз провідних світових звітів (публікацій).

Під час аналізу провідних світових звітів (публікацій), що стосуються ТТП, було досліджено наступні періодичні видання міжнародних виробників рішень у галузі кібербезпеки:

- CrowdStrike Global Threat Report [59; 60; 61];
- Red Canary Threat Detection Report [62; 63; 64];
- Recorded Future Top MITRE ATT&CK Techniques [65; 66];
- Rapid7 Threat Report [67; 68; 69].

Дані, зазначені у публікаціях, були проаналізовані за період із 2019 по 2021 роки.

Зазначені вище звіти представляють собою щорічні провідні дослідження із ТТП, які використовуються кіберзлочинцями найчастіше та найбільш успішно.

При описі ТТП дані публікації спираються на фреймворк MITRE ATT&CK.

MITRE ATT&CK є структурою, яка описує методології, що використовуються зловмисниками під час кібератак. Вона представлена у вигляді матриці, що складається із чотирнадцяти тактик, кожна з яких містить список пов'язаних технік [70]:

- тактика розвідки являє собою методи збору інформації про систему зловмисником, яку він може використовувати для планування майбутніх операцій;
- тактика розвитку ресурсів являє собою методи пошуку ресурсів зловмисником, які він може використовувати для підтримки операцій;
- тактика початкового доступу являє собою вектори, які зловмисники використовують для закріплення в мережі;
- тактика виконання являє собою методи, які призводять до виконання коду, керованого зловмисником, в локальній або віддаленій системі. Ця тактика часто використовується в поєднанні з початковим доступом як засіб виконання коду після отримання доступу і бічним переміщенням для розширення доступу до віддалених систем в мережі;
- тактика постійності – це будь-яка зміна доступу, дії або конфігурації системи, яка забезпечує постійну присутність зловмисника в цій системі;
- тактика підвищення привілеїв – це результат дій, які дозволяють зловмиснику отримати більш високий рівень дозволів в системі або мережі;

- тактика ухилення від захисту складається з прийомів, які противник може використовувати для ухилення від виявлення або ухилення від інших захистів (брандмауерів, антивірусів тощо);
- тактика доступу з обліковими даними включає методи, що забезпечують доступ або контроль над обліковими даними системи, домену або служби, які використовуються в корпоративному середовищі. Це дозволяє зловмисникові прийняти особистість облікового запису з усіма дозволами цього облікового запису в системі. Маючи достатній доступ в мережі, зловмисник може створювати облікові записи для подальшого використання в середовищі;
- тактика виявлення складається із методів, які дозволяють зловмиснику отримати інформацію про систему і внутрішню мережу. Коли зловмисники отримують доступ до нової системи, вони повинні зорієнтуватися в тому, що вони тепер контролюють і які переваги дає використання цієї системи для їх поточної мети. Операційна система надає безліч вбудованих інструментів, які допомагають на етапі збору інформації після компрометації;
- тактика бічного (горизонтального) переміщення складається з методів, які дозволяють зловмиснику отримати доступ до віддалених систем в мережі і керувати ними, а також запускати виконання інструментів у віддалених системах. Методи бічного переміщення можуть дозволити зловмиснику збирати інформацію з системи без необхідності використання додаткових інструментів, таких як інструмент віддаленого доступу;
- тактика збору складається із методів, використовуваних для ідентифікації та збору інформації, такої як конфіденційні файли, з цільової мережі до ексфільтрації. До цієї категорії також входять місця в системі або мережі, де зловмисник може шукати інформацію для проникнення;
- тактика ексфільтрації включає методи і атрибути, які призводять або допомагають зловмиснику видаляти файли і інформацію з цільової мережі;
- тактика командування і управління показує, як зловмисники взаємодіють з системами, що знаходяться під їх контролем в цільовій мережі. Існує багато способів, якими зловмисник може встановити командування і контроль з

різними рівнями секретності, в залежності від конфігурації системи і топології мережі;

- тактика впливу являє собою методи, які зловмисник використовує для маніпулювання, переривання або знищення систем і даних.

Метою даного аналізу є виділення найактуальніших ТТП, які використовуються найбільш часто та успішно, для подальшого формування запитів ППЗ у лабораторному середовищі для протидії ним.

2.3.1 Методологія CrowdStrike

CrowdStrike – провідний світовий виробник рішень із кіберзахисту кінцевих точок, мережі, мобільних пристроїв, хмарний середовищ тощо. Входить до лідерів рішень із захисту кінцевих точок від кіберзагроз за версією світової дослідницько-консалтингової компанії Gartner [59].

Методологія CrowdStrike в частині аналізу ТТП базується на наступних ресурсах [59; 60; 61]:

- технологія CrowdStrike Intelligence – забезпечує розуміння зловмисників, їх кампаній та мотивів, аналізує ТТП;
- технологія CrowdStrike Falcon OverWatch – забезпечує проактивний пошук загроз, що проводиться командою досвідчених фахівців, які працюють над виявленням прихованих загроз в середовищі клієнтів, сортуванням, розслідуванням і усуненням інцидентів в режимі реального часу;
- технологія CrowdStrike Threat Graph – це масштабована хмарна графова модель бази даних. Вона обробляє, зіставляє і аналізує дані, що збираються із більш ніж одного трильйона подій на тиждень в 176 країнах. Архітектура Threat Graph поєднує в собі запатентовані методи зіставлення поведінкових шаблонів з машинним навчанням і штучним інтелектом для відстеження поведінки кожного виконуваного файлу в глобальному співтоваристві клієнтів CrowdStrike;
- сервіси CrowdStrike – надають попереджувальні послуги, такі як оцінка зрілості кібербезпеки, розробка політик і правил, оцінка компрометації. Послуги з

реагування і виправлення надаються досвідченими експертами, які розслідують порушення.

Загалом, звіти CrowdStrike із ТТП представляють огляд та більш глибоке розуміння кібербезпеки, яку досвідчені експерти CrowdStrike ведуть проти найвитонченіших на сьогоднішній день супротивників, а також містять пропозиції та рекомендації щодо підвищення готовності організацій всіх основних галузей діяльності до кібербезпеки.

2.3.2 Методологія Red Canary

Red Canary – це компанія-лідер у галузі надання послуг керованого виявлення загроз та реагування (Managed Detection & Response – MDR). Дана організація є хмарним провайдером послуг із кіберзахисту. Потреба у центрах MDR виникла через те, що часто власних ресурсів будь-якої організації недостатньо для усунення інцидентів інформаційної безпеки, до того ж ефективне управління загрозами передбачає постійне підвищення професійної компетенції, що часто є неможливим в рамках звичайного робочого процесу, адже стрімкість розвитку загроз є дуже високою [62].

Red Canary бере на себе роль центру забезпечення безпеки (SOC) на аутсорсі, тобто займається повноцінним розслідуванням інцидентів кібербезпеки організацій-клієнтів (аналіз телеметрії, необроблених журналів моніторингу із кінцевих точок, мережевого обладнання тощо та надання рекомендацій для швидкого та успішного вирішення проблем).

Суттєвим показником є те, що Red Canary є провайдером сервісу MDR для таких світових рішень у галузі кіберзахисту як: Microsoft Defender for Endpoint, CrowdStrike Falcon Endpoint Detection and Response та VMware Carbon Black. На додачу до цього, авторитетне аналітичне видання The Forrester Wave визнало Red Canary беззаперечним лідером у галузі MDR [63].

Методологія Red Canary в частині аналізу ТТП досліджує дані кінцевих точок у великому масштабі у пошуках зловмисників в середовищі своїх клієнтів. Звіти Red

Canary базуються на наборі даних з 10 000 підтверджених загроз. Кожна підтверджена загроза позначається відповідною технікою MITRE ATT&CK. Ці дані включають інформацію про загрози, виявлені компаніями будь-якого розміру і практично у всіх галузях. Платформа Red Canary щодня збирає і обробляє сотні терабайт телеметрії з датчиків кінцевих точок, а потім проводить всебічний аналіз цієї телеметрії для виявлення і повідомлення клієнтів про загрози, підозрілу поведінку і потенційні ризики [62; 63; 64].

2.3.3 Методологія Recorded Future

Recorded Future – це один із найбільших провайдерів аналітичних даних для корпоративної кібербезпеки.

Платформа Recorded Future Security Intelligence Platform забезпечує аналітичну інформацію про безпеку, яка направлена на протидію кіберзлочинцям. Вона поєднує в собі автоматизовану аналітику та досвід експертів, а також аналіз відкритого вихідного коду, зловмисних мереж, технічних джерел і оригінальних досліджень. Шляхом динамічної категоризації, зв'язування і аналізу даних в режимі реального часу платформа надає прості у використанні аналітичні дані для попереднього зниження ризиків [65].

Методологія Recorded Future в частині аналізу ТТП базується на використанні трьох запитів в Intelligence Platform. Дані для кожного запиту беруться із компоненту платформи –Insikt Notes, що об'єднує дослідників кіберзагроз та аналітиків, а також зразків пісочниці Recorded Future та векторів атак, які автоматично класифікуються платформою [65; 66].

Insikt Notes охоплюють широкий спектр аналітичних даних про загрози і кібератаки, які представляються у першому запиті. Другий запит, який використовує в якості джерела аналіз тестового середовища детонації шкідливих програм, дає технічну перспективу, орієнтовану на виконання шкідливих програм. Третій запит шукає події кібератак, в яких в якості вектора атаки була вказана техніка MITRE ATT&CK, щоб спробувати захопити будь-яку додаткову інформацію. Запити

розділені, щоб виключити помилкові спрацьовування і надати більш цілісну картину використовуваних методів [65; 66].

2.3.4 Методологія Rapid7

Компанія Rapid7 – це один із лідерів в розробці рішень для управління вразливостями і тестування на проникнення, які допомагають отримати повне уявлення про безпеку інформаційної інфраструктури організації. Компанія широко відома своїми рішеннями, такими як Metasploit (платформа для створення і тестування експлойтів, яка також надає інформацію щодо вразливостей та сигнатур попередження вторгнень до системи) і Nexpose (система керування вразливостями, яка виконує проактивне сканування IT-інфраструктури на наявність неправильних конфігурацій, слабких місць, шкідливих компонентів і надає рекомендації щодо усунення існуючих ризиків) [67].

Методологія Rapid7 в частині аналізу ТТП базується на дослідженнях їх центру забезпечення безпеки (SOC), а також команди, що займається керуванням проактивним пошуком загроз (MDR). MDR SOC зазвичай виявляє загрози, які були пропущені технологією запобігання загрозам мережі і кінцевих точок (або технологія запобігання загрозам не мала видимості), адже до 55% атак не відповідають відомим індикаторами загроз [67].

У якості вихідних даних для проведення дослідження провідних кібератак Rapid7 використовує дані великої мережі власних сенсорів (встановлених у клієнтів компанії), включаючи хмару Rapid7 Insight, керовані сервіси, засоби реагування на інциденти, Project Sonar, Heisenberg Cloud і співтовариство Metasploit, задля детектування мінливої ситуації із кіберзагрозами в перспективі. Такий підхід надає чітке уявлення про загрози та вразливості, з якими організації стикаються у різних сферах діяльності, і про те, як ці загрози змінюються протягом року [67; 68; 69].

2.4 Визначення та обґрунтування критеріїв порівняння ТТП

Всього в чотирьох проаналізованих звітах було представлено 120 технік MITRE ATT&CK – дані за три роки (2019-2021) по 10 технік у кожному.

Перш за все були виключені повтори технік у різних звітах, таким чином до кінцевого переліку увійшло 52 унікальні техніки.

Визначення найактуальніших технік для майбутнього керівного зводу правил із ППЗ відбувалося за наступними критеріями:

- для кожної техніки оцінювалася її поширеність у 4 звітах: за згадку у звіті 2021 року – 1 бал, 2020 року – 0,7 бала, 2019 року – 0,5 бала;
- якщо одна й та сама техніка зустрічається у звіті різних років (2019-2021), то бали підсумовуються;
- далі розраховується загальна оцінка за 4 звітами для конкретної техніки;
- для кожної із техніки додатково вказується її присутність у кампаніях APT3 та APT29.

APT3 – це китайська група із організації кіберзагроз, яку дослідники приписують Міністерству державної безпеки Китаю. APT3 покладається на збір облікових даних, введення команд з клавіатури і використання програм, яким вже довіряє операційна система. Дана кампанія атак набула найбільшого розквіту у період з 2017 по 2018 роки, а в 2019 році MITRE ATT&CK випустила дослідження даної кампанії [71].

APT29 – це група із організації кіберзагроз, яку приписують російському уряду і яка діє принаймні з 2008 року. Ця група, як повідомляється, скомпрометувала Національний комітет Демократичної партії, починаючи з літа 2015 року. Дана кампанія атак набула найбільшого розквіту у період з 2018 по 2019 роки, а в 2020 році MITRE ATT&CK випустила дослідження даної кампанії [72].

Присутність у кампаніях APT3 та APT29 у даному дослідженні найактуальніших ТТП є суттєвим фактором, адже більшість сучасних кампаній із кіберзагроз копіюють дані схеми роботи [73, с. 186127-186130].

2.5 Побудова порівняльної характеристики ТТП

Спираючись на критерії викладені вище, було складено порівняльну таблицю найактуальніших ТТП та проранжовано техніки за їх поширеністю (Таблиця 2.1).

Таблиця 2.1

Порівняльна характеристика найактуальніших ТТП

Назва звіту (публікації)				CrowdStrike Global Threat Report	Red Canary Threat Detection Report	Recorded Future Top MITRE ATT&CK Techniques	Rapid7 Threat Report	APT3	APT29
Тактика	Техніка	Назва	Пошире-ність						
Defense Evasion	T1036	Masquerading	7,6	2,2	2,2	1	2,2	+	+
Execution	T1059	Command and Scripting Interpreter (Power Shell)	6,6	2,2	2,2	0	2,2	+	+
Defense Evasion, Privilege Escalation	T1055	Process Injection	5,4	2,2	2,2	1	0	+	+
Execution	T1064	Scripting	5,4	1,5	1,2	0	2,2	+	+
Persistence	T1060	Registry Run Keys / Start Folder	5,4	1,7	1,5	0	2,2	+	+
Initial Access	T1193	Spearphishing Attachment	4,9	0	1,2	1,5	2,2	-	-
Discovery	T1087	Account Discovery	4,9	2,2	0,5	0	2,2	+	-
Credential Access	T1003	OS Credential Dumping	4,4	2,2	2,2	0	0	+	+
Defense Evasion	T1027	Obfuscated Files or Information	4,4	0	1,5	2,2	0,7	-	+
Defense Evasion	T1117	Signed Binary Proxy Execution	4,4	0	2,2	0	2,2	-	+
Execution, Persistence, Privilege Escalation	T1053	Scheduled Task / Job	3,9	0	2,2	0	1,7	+	-
Defense Evasion	T1089	Disabling Security Tools	3,9	1,5	1,2	0	1,2	-	-

продовження табл. 2.1

Collection	T1005	Data from Local System	3,9	1,2	1	1,7	0	-	+
Defense Evasion, Persistence, Privilege Escalation	T1078	Valid Accounts	3,7	0	0	1,5	2,2	+	+
Discovery	T1082	System Information Discovery	3,7	0,7	1	1	1	+	+
Privilege Escalation, Persistence	T1546	Event Triggered Execution	3,4	0	2,2	0	1,2	-	+
Execution	T1204	User Execution	3,4	0	1,2	0	2,2	+	+
Command And Control	T1105	Ingress Tool Transfer	3,2	0	2,2	1	0	-	-
Command And Control	T1219	Remote Access Tools	3,2	0	0	1	2,2	-	-
Defense Evasion, Persistence	T1197	BITS Jobs	2,9	0	1,2	0	1,7	-	-
Resource Development	T1587	Develop Capabilities	2,2	2,2	0	0	0	-	-
Defense Evasion	T1564	Hide Artifacts	2,2	2,2	0	0	0	-	-
Discovery	T1057	Process Discovery	2,2	0	0	2,2	0	+	+
Discovery	T1018	Remote System Discovery	2	0	1	1	0	+	+
Lateral Movement	T1077	Windows Admin Shares	1,9	0	1,2	0	0,7	+	+
Execution	T1035	Service Execution	1,9	0	1,2	0	0,7	+	+
Discovery	T1482	Domain Trust Discovery	1,7	0	1,7	0	0	-	-
Impact	T1496	Resource Hijacking	1,7	0	0	0	1,7	-	-
Credential Access	T1557	Man-in-the-Middle	1,7	0	0	0	1,7	-	-
Collection	T1114	Email Collection	1,7	0	0	0	1,7	-	+
Persistence	T1098	Account Manipulation	1,5	0	0	1,5	0	-	-
Discovery	T1083	File and Directory Discovery	1,5	0	0,5	1	0	+	+
Discovery	T1063	Security Software Discovery	1,5	0	0	1,5	0	+	+
Persistence, Privilege Escalation	T1543	Create and Modify Process	1	0	1	0	0	-	-

продовження табл. 2.1

Execution	T1569	System Services	1	0	1	0	0	-	-
Discovery	T1033	System Owner/User Discovery	1	1	0	0	0	+	+
Impact	T1486	Data Encrypted for Impact	1	0	0	1	0	+	+
Credential Access	T1056	Input Capture	1	0	0	1	0	+	+
Defense Evasion	T1497	Virtualization /Sandbox Evasion	1	0	0	1	0	-	+
Command And Control	T1571	Non-Standard Port	1	0	0	1	0	-	-
Command And Control	T1071	Standard Application Layer Protocol	1	0	0	1	0	+	+
Reconnaissance	T1598	Phishing	1	0	0	1	0	-	-
Impact	T1498	Network Denial of Service	1	0	0	1	0	-	-
Command And Control	T1102	Web Service (Web Application Exploitation)	1	0	0	1	0	-	+
Initial Access	T1189	Drive-by Compromise	1	0	0	1	0	-	-
Exfiltration	T1041	Exfiltration Over C2 Channel	1	0	0	1	0	+	-
Persistence, Privilege Escalation, Defense Evasion	T1038	DLL Search Order Hijacking	0,7	0	0,7	0	0	-	-
Privilege Escalation, Defense Evasion	T1073	DLL Side-Loading	0,7	0	0	0,7	0	-	-
Collection	T1022	Data Encrypted	0,7	0	0	0,7	0	+	+
Execution	T1106	Execution through API	0,7	0	0	0,7	0	-	+
Command And Control	T1032	Standard Cryptographic Protocol	0,7	0	0	0,7	0	+	+
Command And Control	T1090	Connection Proxy	0,5	0	0,5	0	0	-	-

Для спрощення сприйняття категоріювання ТТП, було збережено іменування та маркування MITRE ATT&CK [74; 75, с. 51-60].

2.6 Визначення набору досліджуваних ТТП

До фінального набору технік, що складуть основу майбутнього керівного зводу правил із проведення ППЗ, увійшли техніки із показником поширеності принаймні вище 3 та такі, що були присутні хоча б в одній із кампаній APT3 або APT29 (Таблиця 2.2).

Таблиця 2.2

Визначення набору досліджуваних технік

№	Тактика	Техніка	Назва	Поширеність	APT3	APT29
1.	Defense Evasion	T1036	Masquerading	7,6	+	+
2.	Execution	T1059	Command and Scripting Interpreter (Power Shell)	6,6	+	+
3.	Defense Evasion, Privilege Escalation	T1055	Process Injection	5,4	+	+
4.	Execution	T1064	Scripting	5,4	+	+
5.	Persistence	T1060	Registry Run Keys / Start Folder	5,4	+	+
6.	Discovery	T1087	Account Discovery	4,9	+	-
7.	Credential Access	T1003	OS Credential Dumping	4,4	+	+
8.	Defense Evasion	T1027	Obfuscated Files or Information	4,4	-	+
9.	Defense Evasion	T1117	Signed Binary Proxy Execution (Rundll32)	4,4	-	+
10.	Execution, Persistence, Privilege Escalation	T1053	Scheduled Task / Job	3,9	+	-
11.	Collection	T1005	Data from Local System	3,9	-	+
12.	Defense Evasion, Persistence, Privilege Escalation, Initial Access	T1078	Valid Accounts	3,7	+	+
13.	Discovery	T1082	System Information Discovery	3,7	+	+
14.	Privilege Escalation, Persistence	T1546	Event Triggered Execution	3,4	-	+
15.	Execution	T1204	User Execution	3,4	+	+

Після виділення найактуальніших технік, було здійснено аналіз груп тактик, до складу яких вони входять. Результати представлені у вигляді діаграми (Рисунок 2.2).

	Ухилення від захисту	Виконання	Підвищення привілеїв	Постійність	Виявлення	Доступ з обліковими даними	Збір	Початковий доступ
Частота зустрічання	5	4	4	4	2	1	1	1

Рисунок 2.2 – Оцінка найзастосованіших тактик, використовуваних кіберзлочинцями

Результати свідчать про те, що більшість атак можна було б зупинити на етапах початкового доступу та середнього впливу на систему (тактики 3-11 за MITRE ATT&CK), якби були завчасно впроваджені механізми для такого виявлення.

Найбільш актуальними є етапи атак, пов’язані із ухиленням від захисту (Defense Evasion). Цей етап складається з прийомів, які противник може використовувати для ухилення від виявлення брандмауерами, антивірусами та іншими інструментами. Це свідчить про актуальність підходу ППЗ, адже більшість сучасних атак направляють значну частину ресурсів на ухилення від традиційних рішень інформаційної безпеки.

Другі за розповсюдженістю тактики виконання, підвищення привілеїв та постійності (Execution, Privilege Escalation, Persistence) пов’язані із закріпленням зловмисника в системі після отримання початкового доступу, що ще раз доводить необхідність ППЗ як способу зупинення атаки на ранніх етапах компрометації.

Найменш розповсюджені, проте присутні у всіх сучасних атаках, тактики виявлення, доступу з обліковими даними, збору та початкового доступу (Discovery, Credential Access, Collection, Initial Access) дуже рідко визначаються традиційними засобами антивірусного захисту через свою непомітність і застосовність без прямого доступу до системи.

2.7 Здійснення емуляції загроз

Для реалізації мети складання керівного зводу правил із проведення ППЗ та успішного тестування ППЗ, необхідно створити набір інцидентів (емуляцію загроз) в лабораторному середовищі, які б підпадали під обрані вище техніки. Джерелами для емуляції загроз було обрано MITRE ATT&CK Arsenal та Red Canary Atomic Red Team [76; 77; 78].

MITRE ATT&CK Arsenal – це колекція ресурсів емуляції загроз і супротивників, розроблена і випущена MITRE. Контент в ATT&CK Arsenal був отриманий в результаті багатьох зусиль, включаючи ATT&CK Evaluations та інших ініціатив в області захисту з урахуванням сучасних кіберзагроз [76].

Red Canary Atomic Red Team – це бібліотека специфічних тестів, які може виконати група безпеки для перевірки своїх засобів управління інформаційною безпекою. Тести сфокусовані, мають мало залежностей та визначені в структурованому форматі, який може використовуватися середовищами автоматизації. Atomic Red Team дозволяє тестувати засоби захисту, виконуючи прості «атомарні тести», в яких використовуються ті ж методи, які застосовують зловмисники (всі вони зіставлені з MITRE ATT&CK) [77].

Для обраних вище технік ППЗ, були здійснені відповідно зіставлені емуляції загроз шляхом запуску скриптів PowerShell на віртуальних машинах у лабораторному середовищі. Здійснення певних загроз викликало наповнення даними в хмарній службі Microsoft Defender for Endpoint, у якій в подальшому буде здійснено процес ППЗ.

Нижче наведено короткий опис тестів для емуляції загроз, пов'язаних із обраними техніками [76; 77; 78].

Техніка: Маскування (Masquerading). Використовуваний тест: шкідливий файл, скинутий на диск, являє собою перейменовану копію виконуваного файлу WinRAR. Тест копіює powershell.exe, перейменовує його і запускає, маскуючись під екземпляр recycler.exe. Після успішного виконання powershell.exe перейменовується в taskhostw.exe і запускається по нестандартному шляху.

Техніка: Інтерпретатор команд і скриптів (Command and Scripting Interpreter). Використовуваний тест: вбудований модуль runas, запускає шкідливий скрипт VBScript (autoupdate.vbs). Тест виконує корисне навантаження PowerShell з командного рядка Windows аналогічно тому, що спостерігається при зараженні безфайловими шкідливими програмами.

Техніка: Ін'єкція процесу (Process Injection). Використовуваний тест: вбудована можливість впровадження процесу Windows, виконувана для впровадження зворотного виклику в cmd.exe. Тест виконує шкідливий непідписаний процес (WerFault.exe) шляхом використання вбудованої можливості PowerShell, що в свою чергу впроваджує зворотній виклик через cmd.exe.

Техніка: Виконання сценаріїв (Scripting). Використовуваний тест: заздалегідь виконаний архів, що самостійно розпаковується (Resume Viewer.exe), запускає вбудований командний файл (pdfhelper.cmd). Тест створює і виконує простий командний файл. Після виконання cmd.exe на короткий час запускається для виконання пакетного сценарію autoupdate.bat, а потім знову закривається.

Техніка: Виконання автозапуску (Registry Run Keys / Start Folder). Використовуваний тест: msiexec.exe додає ключ tvncontrol до елементу реєстру

Windows. Тест запускає програмне забезпечення віддаленого доступу шляхом його додавання як ключа запуску реєстру.

Техніка: Виявлення облікових записів (Account Discovery). Використовуваний тест: послідовність виконання cmd.exe, що запускає net.exe з аргументами командного рядка. Тест запускає команди перерахування облікових записів домену.

Техніка: Дампінг облікових даних (OS Credential Dumping). Використовуваний тест: виконання вбудованої можливість дамперу облікових даних Mimikatz (дампер облікових даних, здатний отримувати логіни і паролі облікових записів Windows відкритим текстом). Тест забезпечується процесом svchost.exe, який в свою чергу відкриває lsass.exe з метою отримання доступу до облікового запису.

Техніка: Обфускація файлів або інформації (Obfuscated Files or Information). Використовуваний тест: запуск шкідливого файлу unprotected.vbe, що є закодованим та впроваджується легітимним процесом Windows wscript.exe.

Техніка: Проксі-виконання коду через підписані бінарні файли (Signed Binary Proxy Execution). Використовуваний тест: запуск корисного навантаження DLL (update.dat) за допомогою Rundll32. Тест включає DLL з низькою репутацією, що завантажується підписаним виконуваним файлом через виконання rundll32.exe файлу update.dat, спираючись на елементи панелі управління (Windows CPL).

Техніка: Заплановане завдання (Scheduled Task / Job). Використовуваний тест: планувальних задач Windows через командний рядок створює завдання, яке виконує корисне навантаження зловмисного файлу (updater.dll).

Техніка: Дані з локальної системи (Data from Local System). Використовуваний тест: здійснення скриптового пошуку у файловій системі для документів і медіа файлів з використанням PowerShell. Тест застосовує процес powershell.exe, що виконує команду пошуку дочірніх зв’язків – (Get-) ChildItem.

Техніка: Діючі облікові записи (Valid Accounts). Використовуваний тест: запуск системної утиліти Windows через PowerShell для успішної автентифікації новоствореного користувача, за допомогою облікових даних іншого користувача, які були раніше скомпрометовані. Тест використовує утиліту net.exe, в якій

реалізовані команди для керування мережевими компонентами: розділами, сесіями, службами та обліковими записами користувачів.

Техніка: Виявлення системної інформації (System Information Discovery). Використовуваний тест: шкідливе програмне забезпечення використовує командний рядок для збору основної інформації про систему: версію, архітектуру ядра тощо. Тест здійснює запуск процесу systeminfo.exe через командний рядок, що розкриває критичну інформацію про систему.

Техніка: Виконання за подією (Event Triggered Execution). Використовуваний тест: шкідливе програмне забезпечення перезаписується засобами PowerShell на легітимний системний файл. Тест здійснює захоплення бінарного файлу шляхом застосування закріплених ключів для забезпечення стійкого доступу зловмисника до системи.

Техніка: Виконання користувачем (User Execution). Використовуваний тест: розміщення в системі шкідливого архіву, що містить функцію саморозпакування та здійснює доставку двох безпосередньо небезпечних файлів, шляхом відкриття архіву звичайним користувачем системи.

2.8 Розробка запитів ППЗ

Розробка запитів ППЗ для виявлення визначених технік відбувалася у наступному порядку:

- збір відомостей щодо способів реалізації техніки;
- підбір параметрів KQL для написання запиту ППЗ;
- визначення обмежень, «сліпих зон» розроблюваного запиту ППЗ;
- розробка та тестування запиту ППЗ на емульованій загрозі;
- виявлення джерел можливих помилкових спрацьовувань;
- виправлення помилок.

Нижче представлені результати розробки запитів ППЗ для детектування загроз, що пов’язані із 15 техніками, визначеними у пункті 2.6.

2.8.1 Техніка: Маскування

Відповідно до відомостей про дану техніку із MITRE ATT&CK, маскування пов’язане із перейменуванням системних утиліт. В контексті операційної системи Windows, утиліти надають доступ до можливостей (параметрів, налаштувань, установок), недоступних без їх застосування, або роблять процес зміни деяких параметрів простішим (автоматизують його). Системні утиліти пов’язані, перш за все, із диспетчером задач (утиліта керування процесами), яка дозволяє спостерігати завантаженість процесора, оперативної пам’яті, мережевих підключень та інших ресурсів, а також може запропонувати можливість завершити один із процесів або зазначити йому інший пріоритет. Іншими критичними системними утилітами є: оптимізатор диска (оптимізує розміщення файлів на дисковому накопичувачі, наприклад, шляхом дефрагментації диска), утиліти відновлення після збоїв та діагностики апаратного та програмного забезпечення [79].

В ході застосування даної техніки, зловмисники використовують так звані living off the land binaries (далі – LOLBIN), – легітимне програмне забезпечення, яке використовується для зловмисних цілей, в даному випадку для завантаження вірусу за допомогою Microsoft Defender for Endpoint [79].

Термін «LOLBIN» був запропонований дослідниками у сфері шкідливих програм К. Кемпбеллом і М. Гребером для пояснення використання надійних встановлених системних інструментів для поширення шкідливих програм. Застосування LOLBIN передбачає використання виконавчих файлів Windows для приховування шкідливої активності [80].

LOLBIN використовують справжні системні утиліти та інструменти у зловмисних цілях, що дозволяє їм вписуватися до звичайної мережевої активності і залишатися прихованими. Деякі можливості LOLBIN включають: захоплення динамічних бібліотек (DLL), приховування корисного навантаження, дампінг процесів, завантаження файлів, обхід кейлогерів контролю облікових записів Windows, компіляцію коду, тощо [80].

Щоб вважатися LOLBIN, бінарний файл, бібліотека або сценарій повинні бути в системі за замовчуванням або додані в систему користувачем. Він також повинен володіти несподіваною функціональністю з можливістю перепрофілювання [80].

До недавнього часу методи LOLBIN використовувалися в контексті дій після злому, коли зловмисники використовували законні інструменти адміністрування, такі як PowerShell, Windows Management Instrumentation (WMI), командний рядок, Psxec.exe та інші, для виконання розвідки і бічного переміщення. Але за останні кілька років LOLBIN стали популярними серед авторів шкідливих програм як частина їх первісного корисного навантаження для компрометації [81, с. 9].

Найчастіше мішенню зловмисників, в контексті даної техніки, є хост-процес Windows – Rundll32.exe, що є відповідальним за запуск програм в бібліотеках, які динамічно підключаються. Перейменування даної утиліти дозволяє обійти механізми моніторингу та контролю безпеки. Альтернативним варіантом є випадок, коли допустима утиліта копіюється або переміщується в інший каталог і перейменовується, щоб уникнути виявлення на основі системних утиліт, що виконуються за нестандартними шляхами [81, с. 34-36].

Отже, з технічної точки зору в рамках даної техніки, крім копіювання бінарних файлів до нетипових розположень і їх виконання звідти, також досить поширеною практикою є копіювання і перейменування бінарних файлів до зовсім інших утиліт.

Таким чином, задля детектування техніки маскування в системі, доцільно взяти до уваги, якщо імена файлів не збігаються між ім’ям файлу на диску та ім’ям метаданих виконуваного файлу, це ймовірний індикатор того, що бінарний файл був перейменований після того, як він був скомпільований шкідливою активністю. Збір і порівняння імен файлів на диску і ресурсів для бінарних файлів з перевіркою відповідності внутрішнього файлового імені очікувано може дати корисні відомості, але не завжди може вказувати на зловмисну активність. Адже, доцільніше зосередитись на аргументах командного рядка, які активніше використовуються і відрізняються один від одного, і відповідно мають кращу швидкість виявлення.

Виходячи із зазначеного вище, запит ППЗ було розроблено за принципом правила, яке виявляє перейменовані LOLBIN за допомогою двох масивів. Перший містить всі імена файлів найбільш поширених LOLBIN Windows, другий містить всі відомі вихідні імена файлів, які можуть бути вбудовані в заголовок метаданих, які відрізняються від самого імені файлу. Наступним кроком є об'єднання цих двох масивів і використання їх в якості основи для зіставлення всіх виконуваних процесів (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки маскування, доцільно брати до уваги наступні міркування:

- деякі програмні пакети мають власну копію LOLBIN Windows, що становить ризик, оскільки Microsoft не виправляє і не підтримує їх, таким чином це створює додаткову поверхню для атак;
- деякі установники програмного забезпечення розпаковують і запускають виконавчі файли Windows, використовуючи інші імена. Відповідно, це може провокувати помилкове спрацьовування розроблено запиту ППЗ. У такому випадку слід додати виключення для довірених установників програмного забезпечення, що є прийнятими в конкретній організації;
- зловмисник може раніше змінити заголовок метаданих і вихідне ім'я файлу, щоб обійти виявлення розробленим запитом ППЗ. У такому випадку слід створити додаткові правила виявлення, які враховують конкретну поведінку бінарних файлів.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.3).

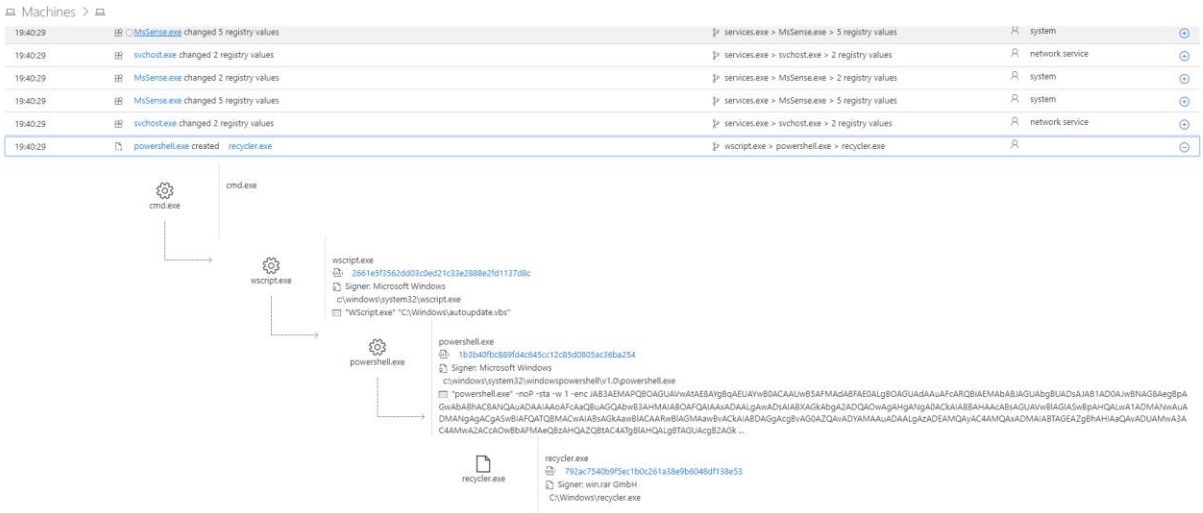


Рисунок 2.3 – Виявлення техніки «Маскування» запитом ППЗ

Застосування ППЗ у Microsoft Defender for Endpoint дозволило виявити шкідливий файл [76; 77], скинутий на диск, що являв собою перейменовану копію виконуваного файлу WinRAR. Даний процес також дозволив системі побудувати взаємозв'язок між запуском перейменованого powershell.exe і маскуванням під екземпляр recycler.exe.

2.8.2 Техніка: Інтерпретатор команд і скриптів

Техніка «Інтерпретатор команд і скриптів» пов'язана із тим, що зловмисники можуть використовувати інтерпретаторами команд і сценаріїв для виконання команд, скриптів або бінарних файлів. Ці інтерфейси і мови надають способи взаємодії з комп'ютерними системами і є спільною рисою багатьох різних платформ [79].

До таких інтерпретаторів відносяться як традиційні Windows Command Shell та PowerShell, так і кросплатформні – Python, JavaScript і Visual Basic.

Досить часто при використанні даної техніки зловмисники можуть застосовувати команди і скрипти PowerShell. Це потужний інтерактивний інтерфейс командного рядка і середовище скриптів, включений в операційну систему Windows. Зловмисники можуть використовувати PowerShell для виконання ряду дій, включаючи виявлення інформації та виконання коду. Приклади включають

командлет Start-Process, який можна використовувати для запуску виконуваного файлу, і командлет Invoke-Command, який запускає команду локально або на віддаленому комп'ютері (хоча для використання PowerShell для підключення до віддалених систем потрібні права адміністратора) [79; 82].

Додатково, команди і скрипти PowerShell можуть виконуватися без прямого виклику виконуваного файлу powershell.exe через інтерфейси до базової динамічної бібліотеки збірки PowerShell System Management Automation, що надається через платформу .NET і Windows Common Language Interface (CLI) [82].

PowerShell також може використовуватися для завантаження і запуску виконуваних файлів з Інтернету, які можуть виконуватися з диска або в пам'яті.

Для успішного детектування техніки «Інтерпретатор команд і скриптів» необхідно брати до уваги те, що зловмисники можуть створювати власну політику виконання PowerShell, якщо вони отримають доступ адміністратора або системи, або через реєстр, або з командного рядка. Ця зміна в політиці системи може бути способом виявлення зловмисного використання PowerShell. Також є корисним відстеження завантаження і виконання артефактів, пов'язаних зі специфічними збірками PowerShell, такими як System.Management.Automation.dll (особливо для незвичайних імен або розташувань процесів) [83].

Спираючись на викладене вище, запит ППЗ було створено як правило, що визначає всі процеси, які завантажують System.Management.Automation.dll, і виключає деякі відомі процеси, які це роблять легітимно. Окрім цього, правило також дозволяє виявити запуск командлетів PowerShell у зловмисних цілях (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Інтерпретатор команд і скриптів», доцільно брати до уваги наступні міркування:

- при першому запуску запит може надавати занадто багато помилкових спрацьовувань, у цьому випадку доцільно додати фільтрацію за всіма виконавчими файлами, підписаними Microsoft;

- у випадку, коли запит все ще надає занадто багато помилкових спрацьовувань, доцільно додати фільтрацію за всіма підписаними виконавчими файлами;
- існує ризик використання оператора !Contains у запиті для InitiatingProcessFolderPath. Зловмисник, який зможе дізнатися ці правила, може підробити ці імена, щоб залишитися непоміченим. У такому випадку доцільно включити повний шлях і його зіставлення до запиту. Повні шляхи складніше обійти, оскільки в більшості випадків потрібен локальний адміністратор, щоб здійснювати запис до розположень вказаних повним шляхом;
- у розробленому запиті присутнє обмеження у виявленні додатків, які завантажили DLL з блоку вбудованої пам’яті, вони не будуть відображатися. Оскільки DLL знаходиться в пам’яті як масив байтів і не має імені, вона є невидимою. Прикладом є UnmanagedPowershell.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.4).

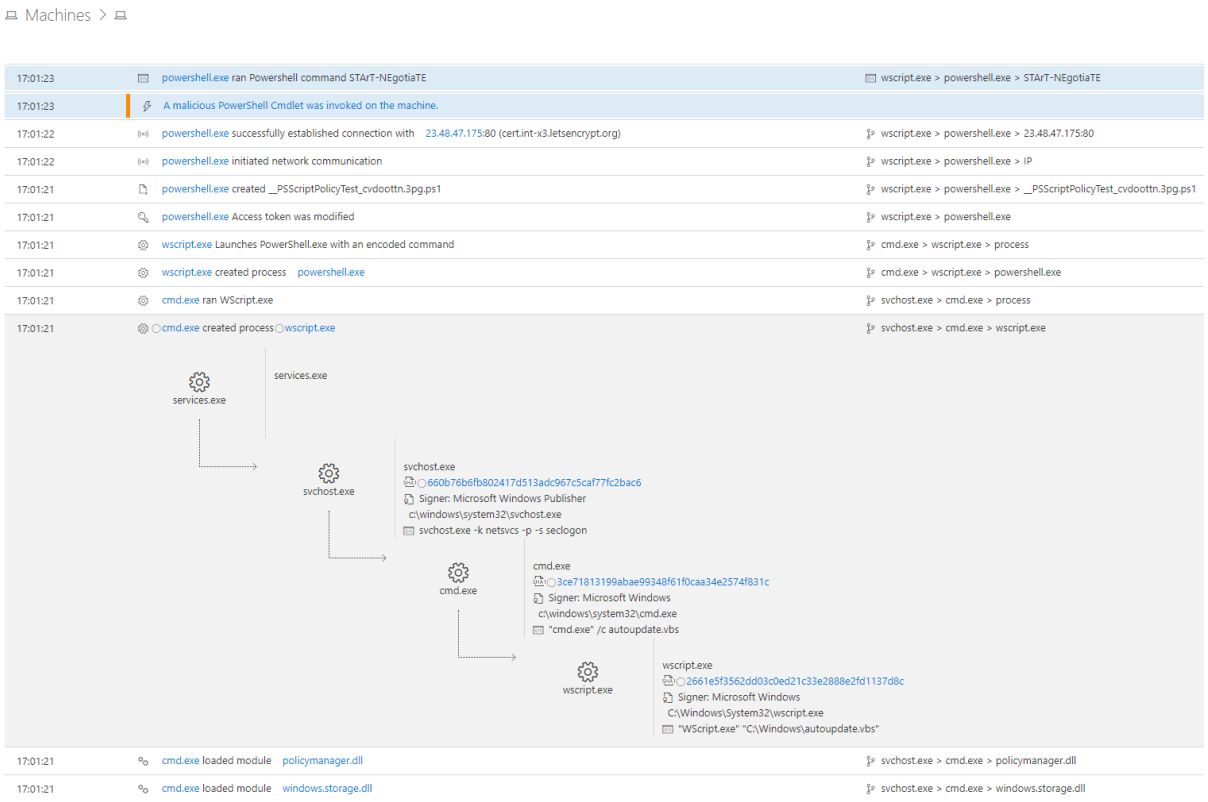


Рисунок 2.4 – Виявлення техніки «Інтерпретатор команд і скриптів» запитом ППЗ

Застосування ППЗ у Microsoft Defender for Endpoint дозволило виявити шкідливий скрипт VBScript (autoupdate.vbs) [76; 77]. Даний процес також дозволив системі побудувати взаємозв'язок між виконанням корисного навантаження PowerShell, використанням DLL та запуском шкідливого скрипта з командного рядка Windows.

2.8.3 Техніка: Ін'єкція процесу

Техніка ін'єкції процесу застосовується зловмисниками для впровадження коду до процесів, з метою обійти захист на основі процесів, а також, можливо, підвищити привілеї. Впровадження процесу – це метод виконання довільного коду в адресному просторі окремого активного процесу. Запуск коду в контексті іншого процесу може дозволити доступ до пам'яті процесу, системних або мережевих ресурсів і, можливо, до підвищених привілеїв. Виконання за допомогою

впровадження процесу може дозволити уникнути виявлення продуктами безпеки, оскільки виконання маскується під легітимним процесом [79].

Ін’єкція зазвичай виконується шляхом копіювання коду (можливо, без файлу на диску) до віртуального адресного простору цільового процесу перед його викликом через новий потік. Запис може виконуватися за допомогою власних викликів Windows API, таких як VirtualAllocEx і WriteProcessMemory, а потім запускатися за допомогою CreateRemoteThread або додаткового коду (наприклад, шелл-коду). Зсув впровадженого коду вводить додаткові вимоги до функціональності для перепризначення посилань на пам’ять [84].

Спираючись на сказане вище, виявити техніку ін’єкції процесу неможливо здійснивши лише моніторинг викликів Windows API, що вказують на різні типи впровадження коду, адже при цьому генерується значний обсяг даних, який не може бути безпосередньо корисний для ППЗ, якщо тільки він не зібраний за певних обставин для відомих зловмисних послідовностей викликів, оскільки безпечне використання функцій API може бути звичайним і слабко відрізнятися від зловмисної поведінки. Натомість, у випадку шкідливої активності, для цієї техніки можуть використовуватися виклики Windows API, такі як CreateRemoteThread, і такі, що можуть використовуватися для зміни пам’яті в іншому процесі, наприклад VirtualAllocEx або WriteProcessMemory [85].

Розроблений запит ППЗ для виявлення ін’єкції процесу направлений на пошук непідписаних процесів або процесів із низьким значенням Global Prevalence, які впроваджуються в інші процеси (Додаток Б).

Global Prevalence – це функція, що застосовується механізмом ППЗ Microsoft Defender for Endpoint. Вона постійно динамічно відслідковує присутність подібних процесів, виявлених на інших кінцевих точках по всьому світу, і відповідно розраховує значення ризику.

Розроблений запит ППЗ визначає використання виклику API NtAllocateVirtualMemoryRemote, зареєстрованого Microsoft Defender for Endpoint. Це легітимне використання виклику, однак при виконанні рідкісним бінарним файлом, який також є непідписаним, така поведінка є сумнівною.

При використанні даного розробленого запиту ППЗ для детектування техніки ін’єкції процесу, доцільно брати до уваги наступні міркування:

- процеси можуть бути впроваджені різними способами, даний запит ППЗ фокусується виключно на варіанті, що використовує виклик API NtAllocateVirtualMemoryRemote;
- можливим обмеженням виявлення при використанні пропонованого запиту ППЗ є ситуація, коли Global Prevalence використовуваного процесу є занадто високим, таким чином запит може його пропустити, те ж саме вірно для програмного забезпечення, підписаного за допомогою перевіреного сертифіката.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.5).

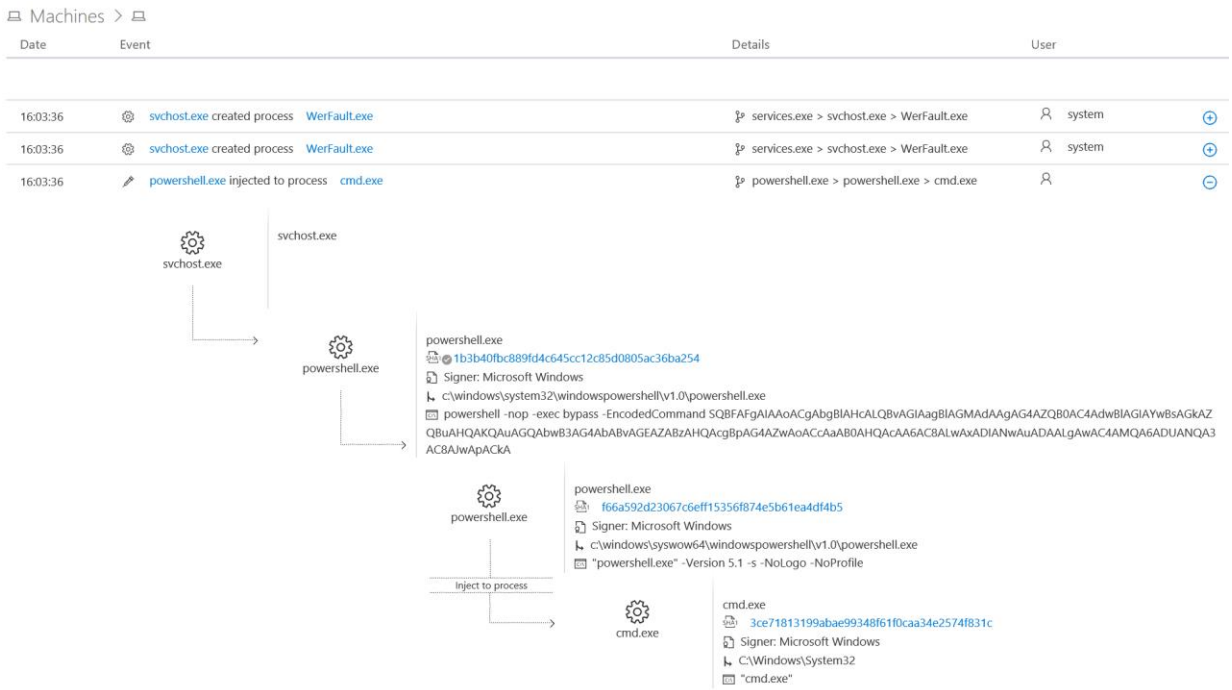


Рисунок 2.5 – Виявлення техніки «Ін'єкція процесу» запитом ППЗ

Застосування ППЗ дозволило виявити виконання шкідливого непідписаного процесу [76; 77], який використовував вбудовану можливість PowerShell Windows, що в свою чергу впроваджує зворотній виклик через cmd.exe.

2.8.4 Техніка: Виконання сценаріїв

Виконання сценаріїв – це відносно універсальний метод, який охоплює широкий спектр дій, багато з яких пов'язані із використанням інструментів і мов, які є функціонально важливими компонентами операційних систем. Сценарії є загальновживаними через свою простоту, що ускладнює визначення вихідних показників нормальної активності сценаріїв і побудову стратегій виявлення аномальної активності [79].

Зловмисники використовують сценарії для виконання безпосередніх дій на кінцевій точці. Це може включати використання скрипта для завантаження шкідливої програми із зовнішнього хоста, вбудовування скрипта у вкладення електронної пошти як частина фішингової кампанії, вбудовування бінарних файлів, які можуть бути заблоковані, або написання скрипта, який автоматизує і прискорює рутинну роботу, яку в іншому випадку доведеться виконувати вручну [86].

Оскільки постійні поліпшення і впровадження рішень для управління додатками ускладнюють установку шкідливого виконуваного файлу на хост-машину, сценарії дозволяють зловмисникам виконувати дії, не встановлюючи нічого.

При виявленні техніки виконання сценаріїв слід пам'ятати, що це всеосяжний метод, який використовується для виконання коду в інших процесах. Існує велика кількість задокументованих підходів до виконання сценаріїв. Більш класичні підходи до впровадження процесу використовують API CreateRemoteThread. Такий метод також часто використовується зловмисниками для переходу до інших процесів [87].

Таким чином, розроблений запит ППЗ для виявлення техніки виконання сценаріїв відфільтровує всі зареєстровані використання API CreateRemoteThread, в яких процес впровадження не дорівнює процесу, який впроваджується. Далі відбувається фільтрація унікальних впроваджень на базі хешу SHA1 і збагачення даних за допомогою функції FileProfile в Microsoft Defender for Endpoint. Потім, список всіх унікальних хешів SHA1 підсумовується, де Global Prevalence нижче

стандартно визначеного порогу. Насамкінець, відбувається фільтрація всіх подій журналу CreateRemoteThread, де впровадження (інжектор) був в раніше створеному списку підозрілих Global Prevalence (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки виконання сценаріїв, доцільно брати до уваги наступні міркування:

- граничні значення зазначені у пропонованому запиті ППЗ вимагають точного налаштування конкретного IT-середовища. Чим нижче є значення, тим краще, але необхідно врівноважити кількість помилкових спрацьовувань. У даному випадку слід встановити максимально можливі низькі значення, при яких імовірність помилкових спрацьовувань буде прийнятною;
- часовий інтервал наявних даних за якими запит ППЗ здійснює пошук повинен бути досить коротким, щоб кількість хешів була менше 1000, що обумовлено обмеженнями функції FileProfile;
- існують легітимні додатки, які використовують CreateRemoteThread для впровадження процесу, це може спровокувати деякі помилкові спрацьовування пропонованого запиту ППЗ.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.6).

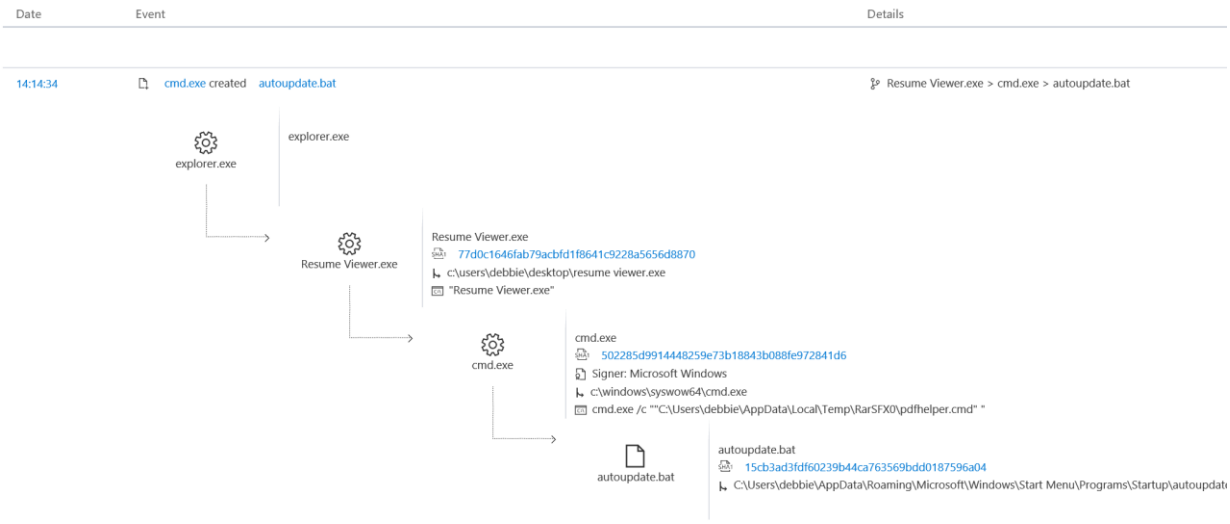


Рисунок 2.6 – Виявлення техніки «Виконання сценаріїв» запитом ППЗ

Застосування ППЗ дозволило виявити заздалегідь виконаний архів, що самостійно розпаковується (Resume Viewer.exe) та запускає вбудований командний файл (pdfhelper.cmd) [76; 77]. Додатково, було виявлено запуск cmd.exe на короткий час для виконання пакетного сценарію autoupdate.bat.

2.8.5 Техніка: Виконання автозапуску

Техніка «Виконання автозапуску» або запуску ключів реєстру пов'язана із тим, що зловмисники можуть налаштувати параметри системи для автоматичного виконання програми під час завантаження системи або входу в систему для збереження стійкості або отримання привілеїв вищого рівня в скомпрометованих системах. Операційні системи можуть мати механізми для автоматичного запуску програми при завантаженні або вході в систему. Ці механізми можуть включати в себе програми, які автоматично виконуються та поміщаються в спеціально призначені каталоги або на які посилаються репозиторії, що зберігають інформацію про конфігурацію, наприклад, реєстр Windows. Кіберзлочинець може досягти тієї ж мети, змінюючи або розширюючи можливості ядра. Оскільки деякі програми автозапуску завантаження або входу в систему працюють з більш високими правами, зловмисник може використовувати їх для підвищення привілеїв [79].

Технічно даний процес пов'язаний із формуванням стійкості в системі, шляхом додавання програми до папки автозавантаження або посилаючись на неї за допомогою ключа запуску реєстру. Додавання запису до ключів запуску в реєстрі або папки автозавантаження призведе до того, що зазначена програма буде виконуватися при вході користувача в систему. Ці програми будуть виконуватися в контексті користувача і матимуть пов'язані з обліковим записом дозволи [88].

Розміщення програми в папці автозавантаження також викликає виконання цієї програми при вході користувача в систему. Існує папка автозавантаження для окремих облікових записів користувачів, а також загальносистемна папка

автозавантаження, яка буде перевірятися незалежно від того, яка обліковий запис користувача входить в систему [88].

При детектуванні техніки «Виконання автозапуску» доцільно слідкувати за реєстром на предмет змін, щоб запускати ключі, які не корелюють з відомим програмним забезпеченням, циклами виправлень. Корисним також є відстеження початкової папки на предмет змін. Такі інструменти, як Sysinternals Autoruns, також можуть використовуватися для виявлення системних змін, які можуть бути спробами збереження, включаючи перерахування розташувань реєстру ключів запуску і папок автозавантаження. Підозріле виконання програми при запуску може проявлятися у вигляді незвичних процесів, які раніше не спостерігалися при порівнянні з історичними даними. Щоб підвищити впевненість у тому, що це дійсно зловмисна активність, дані і події слід розглядати не ізольовано, а як частину ланцюжка поведінки, яка може привести до інших дій, таких як мережеві з'єднання [89].

Пропонований запит ППЗ для виявлення техніки «Виконання автозапуску» здійснює пошук ініційованих процесів, що супроводжуються змінами в реєстрі, а додатково умовою є спрацьовування дій із боку Antimalware Scan Interface (AMSI) – універсального інтерфейсу Windows для будь-яких додатків, які можуть звертатися до будь-яких встановлених на комп'ютері програм захисту і використовувати їх для додаткової перевірки процесів на більш глибокому рівні (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки виконання автозапуску, доцільно брати до уваги наступні міркування:

- зміни деяких параметрів конфігурації автозапуску можуть відбуватися при звичайних умовах, коли встановлено легітимне програмне забезпечення, тому у цьому випадку можливі помилкові спрацьовування запиту ППЗ;
- для підвищення впевненості у детектуванні справжнього інциденту, рекомендується поєднувати запит ППЗ з іншими спрацьовуваннями антивірусу, наприклад із інцидентами, що вказують на ознаки бічного переміщення.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.7).

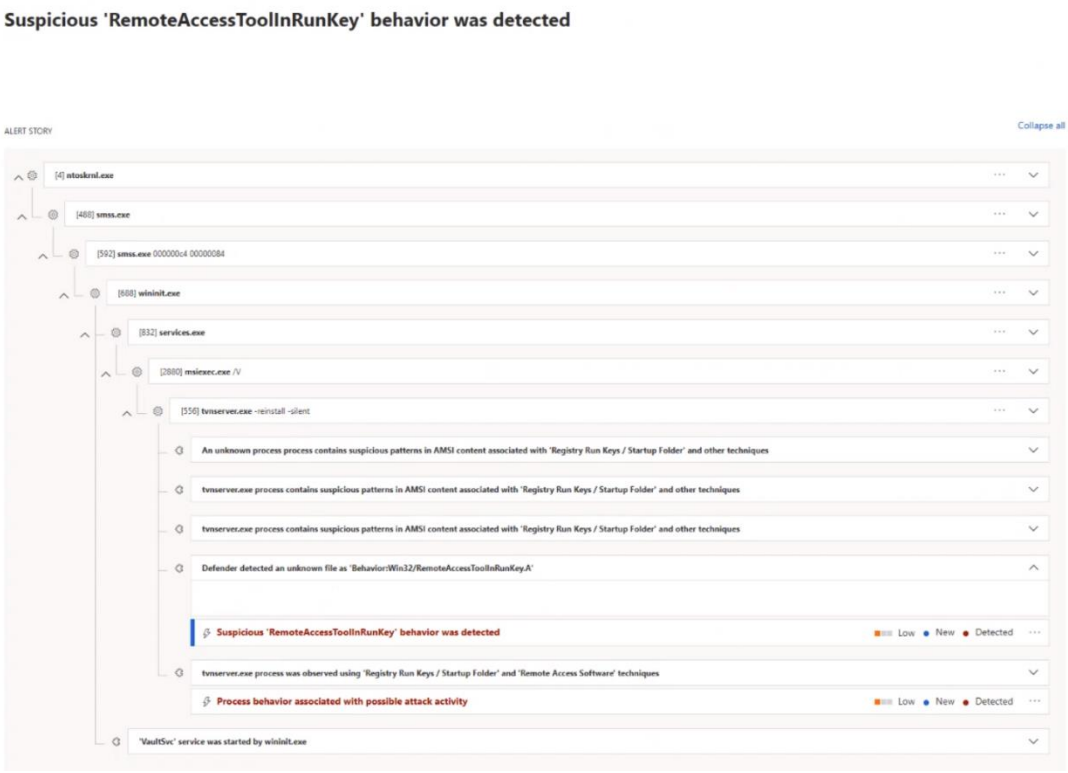


Рисунок 2.7 – Виявлення техніки «Виконання автозапуску» запитом ППЗ

Застосування розробленого запиту ППЗ дозволило виявити всю гілку подій від додавання ключа до елементу реєстру Windows – HKLM \ Software \ Microsoft \ CurrentVersion \ Run до запуску програмного забезпечення віддаленого доступу [76; 77] шляхом його додавання як ключа запуску реєстру.

2.8.6 Техніка: Виявлення облікових записів

Техніка виявлення облікових записів застосовується зловмисниками для спроб отримати список облікових записів домену. Ця інформація може допомогти їм визначити, які облікові записи домену існують, щоб забезпечити успішне проведення подальшої атаки. В операційній системі Windows для таких цілей використовуються команди, такі як net user і net group утиліти Net [79].

Утиліта Net – це компонент операційної системи Windows, який використовується в операціях командного рядка для керування користувачами,

групами, службами та мережевими підключеннями. Net має функції, багато з яких корисні для зловмисника, такі як збір системної і мережевої інформації, горизонтальне переміщення через загальні ресурси адміністратора Windows за допомогою команд мережевого використання і взаємодія зі службами [90].

Доцільно зазначити, що виявлення облікових записів як частина методів виявлення системи і мережі зазвичай використовуються протягом усієї операції, коли зловмисник вивчає середовище. Таким чином, дані та події, пов'язані із цією технікою, слід розглядати не ізольовано, а як частину ланцюжка дій, який може призвести до інших, таких як горизонтальне переміщення в системі, на основі отриманої інформації. Корисним у детектування даної техніки є відстеження процесів і аргументів командного рядка для дій, які можуть бути здійснені для збору системної і мережевої інформації. Інструменти віддаленого доступу з вбудованими функціями також можуть безпосередньо взаємодіяти з Windows API для збору інформації. Додатково, інформацію можна отримати за допомогою інструментів управління системою Windows, таких як Windows Management Instrumentation (далі – WMI) і PowerShell [91].

Пропонований запит ППЗ для детектування техніки виявлення облікових записів виявляє спроби перерахування облікових записів домену, спираючись на команди утиліти Net, які використовуються найчастіше, та на інструментарій PowerShell, який може бути задіяний у даній атаці (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки виявлення облікових записів, доцільно брати до уваги наступні міркування:

- застосування команд утиліти Net адміністраторами системи саме по собі не є інцидентом інформаційної безпеки, тому для успішного виявлення техніки, необхідно враховувати контекст;
- у деяких випадків можуть знадобитися виключення для WMI, адже деяке програмне забезпечення, наприклад мережевий моніторинг, може застосовувати даний інструмент.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.8).

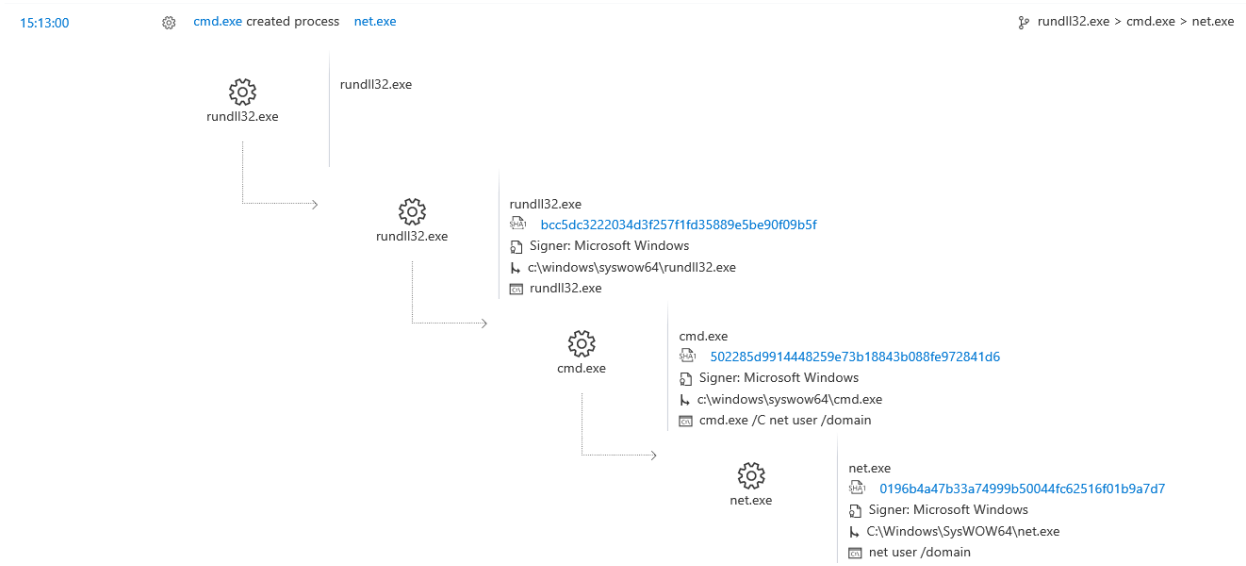


Рисунок 2.8 – Виявлення техніки «Виявлення облікових записів» запитом ППЗ

Застосування ППЗ дозволило виявити послідовність виконання cmd.exe, що запускає net.exe з аргументами командного рядка [76; 77], що містили команди для перерахування облікових записів домену.

2.8.7 Техніка: Дампінг облікових даних

Техніка «Дампінг облікових даних» – це процес отримання логінів і паролів, як правило, у формі хешу або текстового пароля з операційної системи.

Дана техніка застосовується зловмисниками для спроб скидання облікових даних, щоб отримати дані для входу в обліковий запис, які зазвичай представлені у формі хешу або паролю у вигляді відкритого тексту, з операційної системи і програмного забезпечення. Потім облікові дані можна використовувати для виконання горизонтального переміщення і доступу до інформації з обмеженим доступом [79].

Успішне детектування техніки дампінгу облікових даних у системах Windows можливе шляхом відслідковування несподіваних процесів, що взаємодіють з lsass.exe. Зазвичай програми-дампери облікових даних отримують доступ до процесу LSA Subsystem Service (LSASS), відкривають його, знаходять секретний

ключ LSA і розшифровують розділи в пам’яті, де зберігаються облікові дані. Дампери облікових даних можуть також використовувати методи відбивання ін’єкції процесів, щоб зменшити потенційні індикатори компрометації шкідливої активності [92].

Хеш-дампи відкривають диспетчер облікових записів безпеки (SAM) в локальній файловій системі або створюють дамп ключа SAM реєстру для доступу до збережених хешів паролів облікових записів. Деякі хеш-дампи відкривають локальну файлову систему і виконують синтаксичний аналіз таблиці SAM, щоб уникнути захисту доступу до файлів. Інші роблять копію таблиці SAM в пам’яті перед читанням хешів [93].

Пропонований запит ППЗ виявляє дампінг облікових даних шляхом пошуку прапорів, що вказують на запис у всій пам’яті процесу LSASS, та за допомогою перевірки бази даних SAM на предмет крадіжки облікових даних (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки виявлення облікових записів, доцільно брати до уваги наступне міркування:

- не всі прапори запису у пам’яті процесу LSASS можуть свідчити про дампінг облікових даних, тому з метою уникнення помилкових спрацьовувань слід враховувати додатковий контекст інциденту.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.9).

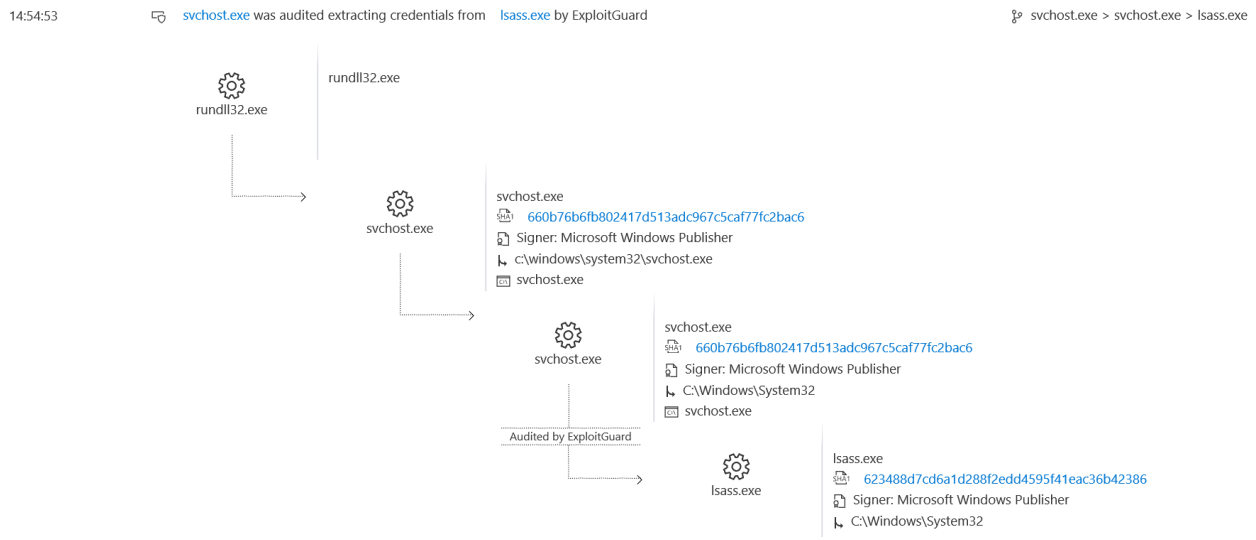


Рисунок 2.9 – Виявлення техніки «Дампінг облікових даних» запитом ППЗ

Здійснення пошуку розробленим запитом ППЗ дозволило виявити виконання вбудованої можливості дампера облікових даних [76; 77], що забезпечувався процесом svchost.exe, який в свою чергу відкриває lsass.exe з метою отримання доступу до облікового запису.

2.8.8 Техніка: Обфускація файлів або інформації

Техніка обфускації файлів або інформації включає такі методи як: шифрування, кодування та інші методи перетворення файлів і їх вмісту в системі або при їх передачі [79].

Корисні навантаження можуть бути додані в архів або зашифровані, іноді для проведення зворотного перетворення і подальшого запуску необхідною є активна дія з боку користувача, наприклад ввести пароль для відкриття архіву, підготовленого зловмисником. Щоб приховати рядки простого тексту закодованими можуть бути і частини файлів. Корисні навантаження можуть бути розділені на окремі «безпечні» файли, які при складанні в єдине ціле виконують шкідливий функціонал [94].

Кіберзлочинці також можуть заплутувати команди, що викликаються з корисних навантажень безпосередньо або через інтерфейс командного рядка. Змінні

середовища, псевдоніми і символи, характерні для семантики платформи або мови, можуть використовуватися для обходу виявлення шкідливого коду на основі сигнатур і білих списків. Також обхід засобів захисту у більшості випадків забезпечується тим, що зловмисники часто зіставляють кодовані файли або приблизно відповідають імені або місцю розташування легітимних файлів, щоб уникнути правил виявлення, заснованих на довірі до певних процесів операційної системи [95, с. 6-7].

Ще одним прикладом обфускації є використання стенографії – техніки приховування даних або коду в зображеннях, звукових доріжках, відео і текстових файлах.

З огляду на вище сказане, детектування техніки обфускації можливе за рахунок застосування запиту ППЗ, що направлений на виявлення невідповідності у батьківсько-дочірніх відносинах основних процесів операційної системи, щоб виявити різні спроби маскування кодованих або шифрованих файлів (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки обфускації файлів або інформації, доцільно брати до уваги наступні міркування:

- деякі із батьківських процесів запускають свої дочірні процеси і відразу ж завершуються, що в деяких випадках може призвести до того, що батьківське поле залишиться порожнім, тому такі спрацьовування слід сприймати як хибні;
- деякі процеси, такі як dllhost можуть мати інші легітимні батьківські процеси, що можуть бути не зазначені у пропонованому запиті;
- очевидно існує обмеження у виявленні для тих процесів, які не зазначені у пропонованому запиті ППЗ.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.10).

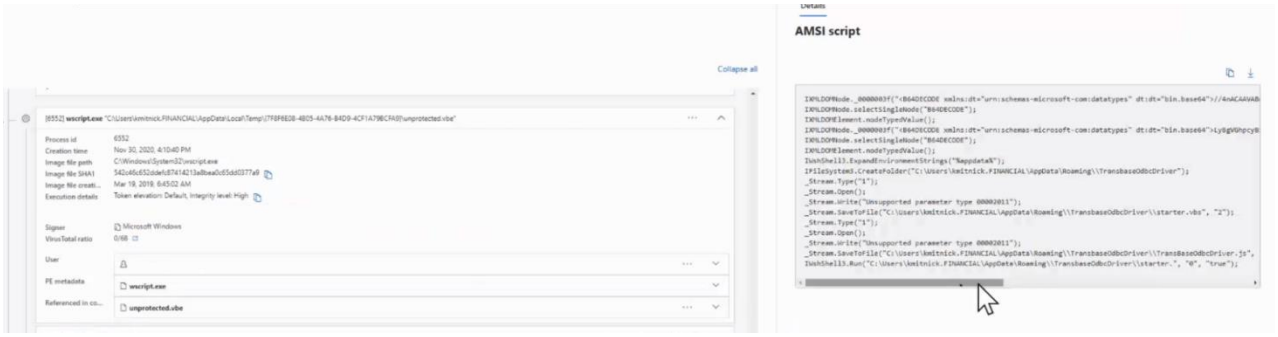


Рисунок 2.10 – Виявлення техніки «Обфускація файлів або даних» запитом ППЗ

Використання запиту ППЗ дозволило виявити шкідливий файл unprotected.vbe, що є закодованим та впроваджується легітимним процесом Windows wscript.exe ще на етапі його первинного впровадження процесом [76; 77]. Додатково, були отримані більш докладні відомості щодо файлу шляхом отримання інформації із AMSI інтерфейсу Windows.

2.8.9 Техніка: Проксі-виконання коду через підписані бінарні файли

Техніка використовується з метою обходу захисту на основі процесів або сигнатур, шляхом проксіювання (перенаправлення трафіку) виконання шкідливого вмісту за допомогою підписаних бінарних файлів [79].

Бінарні файли, підписані довіреними цифровими сертифікатами, можуть виконуватися в системах Windows, захищених перевіркою цифрового підпису. Деякі підписані Microsoft виконавчі файли, які використовуються за замовчуванням в установках Windows, можуть використовуватися для проксі-виконання інших файлів.

Одним із найчастіше використовуваних для даних цілей виконавчим файлом є rundll32.exe. Це хост-процес Windows, який є компонентом операційної системи та запускає програми, що знаходяться бібліотеках, що динамічно підключаються (DLL) [96].

Використання rundll32.exe, виконуваного безпосередньо (тобто, через загальні модулі), допомагає уникнути запуску інструментів безпеки, які можуть не відслідковувати виконання процесу rundll32.exe через списки дозволів або помилкові спрацьовування при нормальній роботі. rundll32.exe зазвичай асоціюється з виконанням корисних навантажень DLL [96].

Однак, особливу небезпеку застосовування даної техніки становить, коли rundll32.exe використовується для виконання файлів елементів панелі управління Windows (.cpl) за допомогою незадокументованих функцій shell32.dll [97, с. 4].

Таким чином, у більшості інструментів захисту направлених на боротьбу із застосовуванням даної техніки, використовується моніторинг процесів для відстеження виконання і аргументів rundll32.exe. Далі відбувається порівняння недавніх викликів rundll32.exe з попередньою історією відомих аргументів і завантажених бібліотек DLL, щоб визначити аномальну і потенційно ворожу активність. Аргументи команди, використовувані при виклику rundll32.exe, також можуть бути корисні при визначенні джерела і цілі DLL [97, с. 8].

Однак, у такому підході є суттєвий недолік, адже файли CPL – це особливий вид DLL, які мають пріоритетне значення в Windows. Коли Windows бачить файл CPL, операційна система передбачає, що це елемент панелі керування, і виконує його. Тому, зловмисники використовують файли CPL як спосіб уникнути виявлення техніки. Якщо правила виявлення спрямовані на виявлення файлів .DLL і .EXE, CPL може залишитися непоміченим [97, с. 14].

Саме тому пропонований запит ППЗ для виявлення техніки «Проксі-виконання коду через підписані бінарні файли» направлений на виявлення подій ImageLoad в CPL, які мають «низьку репутацію». Спочатку складається перелік всіх окремих хешів завантажених файлів CPL. Для кожного файлу перевіряється база даних Microsoft Threat Intelligence (база знань, розроблена експертами з кібербезпеки Microsoft), щоб зрозуміти, як часто Microsoft переглядала його раніше. Далі встановлюються деякі пороги для підписаних і непідписаних CPL. Якщо розмір файлу нижче порогового значення, спрацьовує попередження (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Проксі-виконання коду через підписані бінарні файли», доцільно брати до уваги наступні міркування:

- граничні значення у пропонованому запиті ППЗ вимагають точного налаштування конкретного ІТ-середовища. Чим нижче є значення, тим краще, але необхідно врівноважити кількість помилкових спрацьовувань. У даному випадку слід встановити максимально можливі низькі значення, при яких імовірність помилкових спрацьовувань буде прийнятною;
- часовий інтервал, за яким здійснюється пошук, повинен бути досить коротким, щоб кількість унікальних хешів CPL була меншою за 1000 через обмеження функції FileProfile;
- деякі безпечні програми можуть додавати допустимі файли CPL. Якщо програмне забезпечення досить нерозповсюджене, глобальна поширеність встановлена Microsoft може бути низькою, що може стати потенційним джерелом помилкових спрацьовувань.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.11).

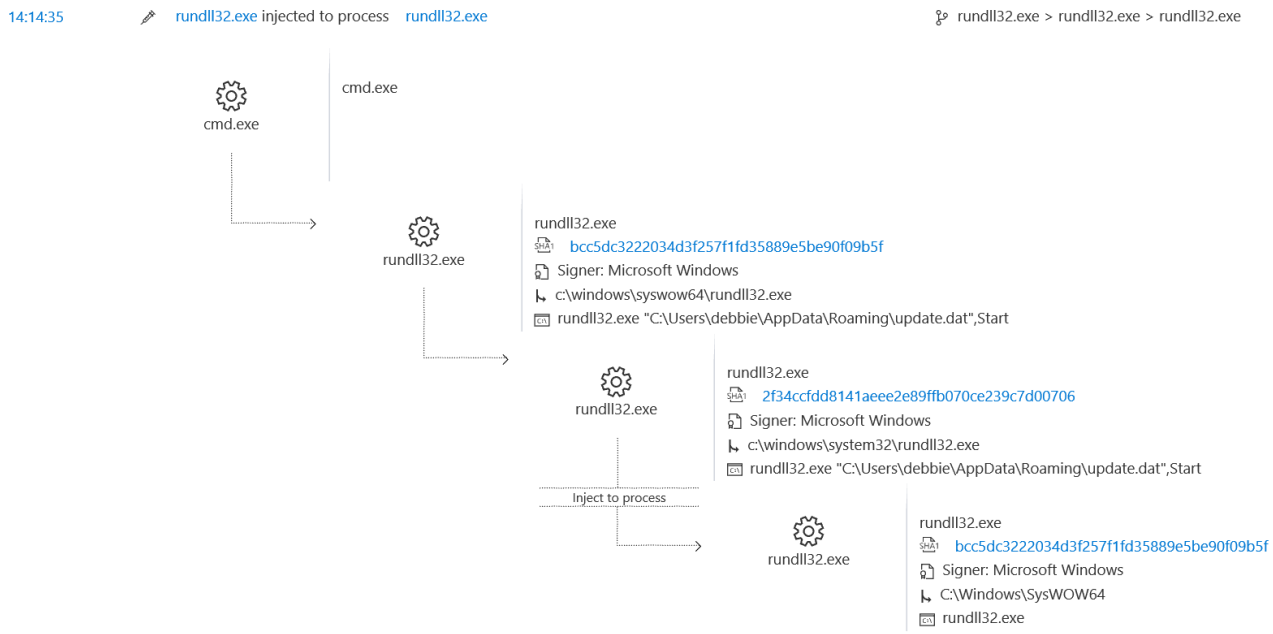


Рисунок 2.11 – Виявлення техніки «Проксі-виконання коду через підписані бінарні файли» запитом ППЗ

Використання запиту ППЗ дозволило успішно виявити DLL з низькою репутацією [76; 77], що покроково завантажується підписаним виконуваним файлом через виконання rundll32.exe файлу update.dat, спираючись на елементи панелі управління – Windows CPL.

2.8.10 Техніка: Заплановане завдання

Техніка виконання запланованих завдань застосовується для полегшення початкового або повторюваного виконання шкідливого коду кіберзлочинцями. У всіх основних операційних системах існують службові програми для планування виконання програм або сценаріїв у зазначені дату і час. Завдання також може бути заплановане у віддаленій системі за умови виконання належної автентифікації (наприклад, за допомогою віддаленого виклику процедур або загального доступу до файлів і принтерів в середовищах Windows). Для планування завдання у віддаленій системі зазвичай необхідне членство в групі адміністраторів або іншій привілейованій групі [79].

В сімействі операційних систем Windows дана техніка застосовується через планувальник завдань Windows – компонент, що дозволяє запланувати запуск програм та скриптів за розкладом. Зазвичай атакуючими використовуються декілька способів отримання доступу до планувальника завдань Windows. Перший спосіб передбачає запуск процесу планувальника – schtasks.exe безпосередньо з командного рядка, другий – відкриття планувальника за допомогою графічного інтерфейсу панелі управління Windows та її інструментів адміністратора. У деяких випадках можливе використання .NET-оболонки для планувальника завдань Windows або бібліотеки Windows netapi32 для створення запланованої задачі [98].

Таким чином, основним способом протидії даній техніці є відстеження виконання процесів за допомогою svchost.exe у Windows. У більшості випадків порушники не зберігають заплановані задачі в системі, а видаляють їх одразу після завершення необхідної дії, тому необхідно також здійснювати моніторинг записів про зміни, пов’язані із запланованими завданнями, які не корелюють з відомим програмним забезпеченням та циклами виправлень [99].

Розроблений запит ППЗ для виявлення техніки запланованих завдань направлений на отримання унікальних хешів бінарних файлів, які виконуються батьківським процесом svchost і аргументами командного рядка netsvcs. Для виявлення потенційно шкідливих бінарних файлів зберігаються тільки непідписані виконавчі файли з низькою глобальною поширеністю, встановленою Microsoft (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Заплановане завдання», доцільно брати до уваги наступні міркування:

- не кожен непідписаний файл, що виконується процесом планувальника задач є шкідливим, однак додаткова перевірка таких фалів не завадить;
- глобальна поширеність, встановлена Microsoft, не є абсолютно вірним джерелом, адже існують довірені виконавчі файли, які не мають цифрового підпису, наприклад виконавчі файли власної розробки, тому це є потенційною причиною можливих помилкових спрацьовувань;

- в рамках даної техніки атакуючий може використовувати підписаний бінарний файл, такий як cmd.exe або LOLBIN, для непрямого запуску шкідливого коду, у такому випадку пропонований запит ППЗ не спрацьовує.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.12).

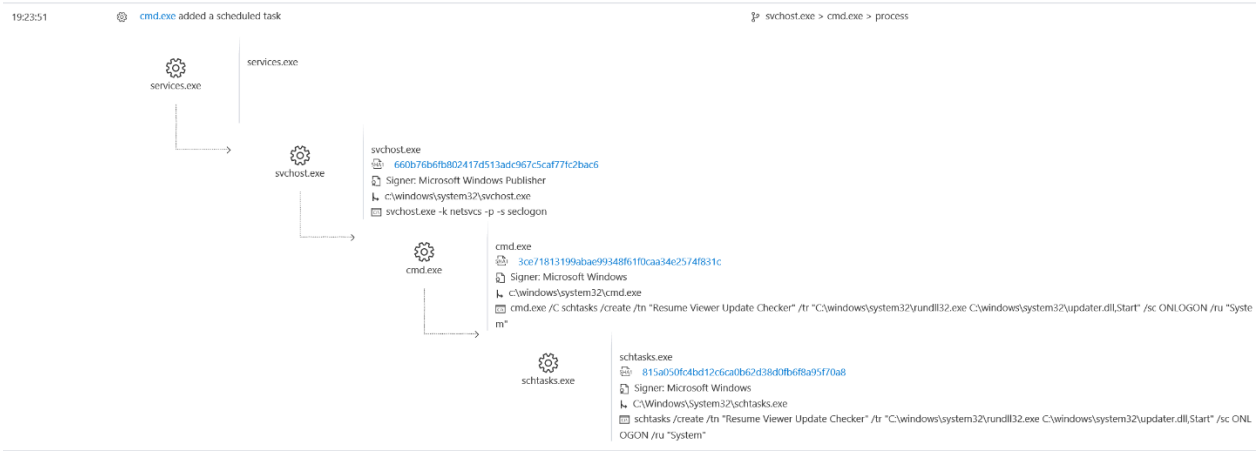


Рисунок 2.12 – Виявлення техніки «Заплановане завдання» запитом ППЗ

Здійснення пошуку за допомогою запиту ППЗ дозволило виявити ланцюжок подій [76; 77], під час яких процес планувальника задач Windows через командний рядок створив завдання, яке виконало корисне навантаження зловмисного файлу.

2.8.11 Техніка: Дані з локальної системи

Техніка витягнення даних з локальної системи є однією із найскладніших для виявлення, адже відповідно до MITRE ATT&CK, вона відноситься до групи тактик «Збір» і не є повноцінною атакою. На етапі застосування даної техніки порушники можуть виконувати пошук в джерелах локальної системи, таких як файлові системи або локальні бази даних, щоб знайти критичні файли і конфіденційні дані для проведення подальшої ексфільтрації (виділення безпечної зони для проведення безпосередньої атаки на систему) [79].

Способи виконання даної техніки включають: використання інтерпретатора команд і сценаріїв (наприклад, звичайного командного рядка), який має функції для взаємодії з файловою системою для збору інформації; автоматизований збір в локальній системі за допомогою інструментів керування Windows (WMI) або PowerShell [100].

Таким чином, для детектування даної техніки слід звертати увагу на загальні команди файлової системи і параметри в інтерфейсі командного рядка в пакетних файлах або сценаріях. Послідовність подібних дій може вказувати на потенційний автоматичний збір інформації. З огляду на це, моніторинг доступу до файлів, який показує незвичайний процес, що виконує послідовне відкриття файлу і копіювання до інших розположень файлової системі одночасно для багатьох файлів, може вказувати на поведінку збору даних з локальної системи [101, с. 19-30].

Запит ППЗ для детектування даної техніки було розроблено шляхом виявлення виконання незвичних для середовища команд PowerShell, які пов’язані із визначенням зв’язків у файловій системі та виявлення планомірного завантаження файлів до деякого розположення у файловій системі (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Дані з локальної системи», доцільно брати до уваги наступні міркування:

- моніторинг виконання команд PowerShell, які пов’язані із визначенням зв’язків у файловій систем, слід розглядати у контексті інших подій IT-середовища та виключити можливість їх застосування справжнім адміністратором системи;
- планомірне завантаження даних до певних розположень може бути частиною робочого процесу, тому дані події також доцільно розглядати у контексті;
- у той же час, пропонований запит ППЗ має низький рівень помилкових спрацьовувань, адже рідко легітимні процеси в системі поєднують одночасне розслідування зв’язків та завантаження файлів.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.13).

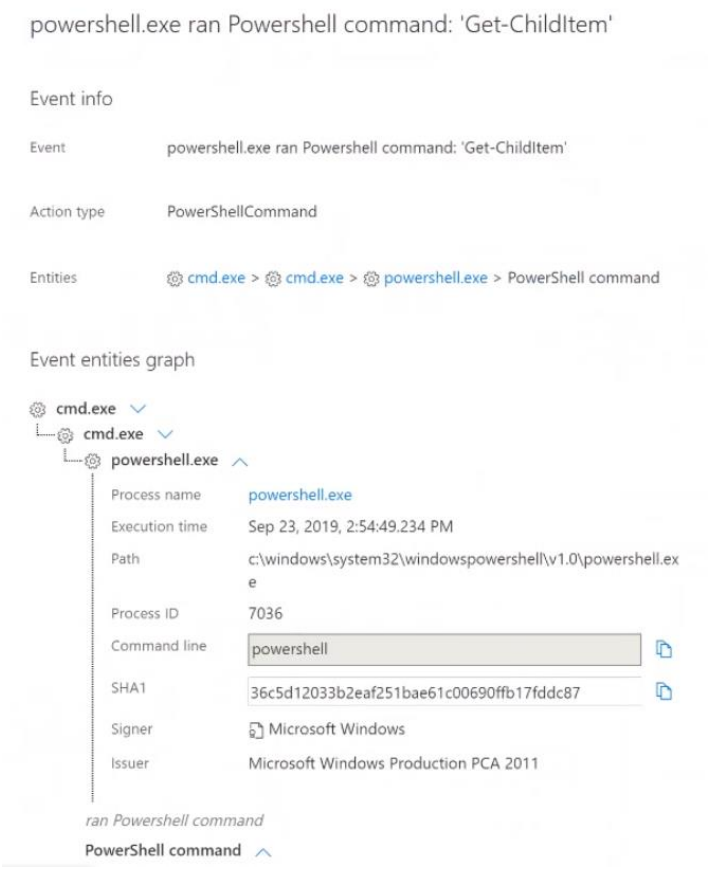


Рисунок 2.13 – Виявлення техніки «Дані з локальної системи» запитом ППЗ

Використання запиту ППЗ дозволило виявити здійснення скриптового пошуку у файловій системі з використанням PowerShell [76; 77], що виконував команду пошуку дочірніх зв'язків – (Get-) ChildItem. Додатково, ППЗ показав взаємозв'язок між завантаженням файлів з командного рядка та PowerShell.

2.8.12 Техніка: Діючі облікові записи

Техніка використання діючих облікових записів базується на скомпрометованих облікових даних справжніх користувачів. Отримання зловмисниками скомпрометованих облікових даних можливе внаслідок проведення фішингових кампаній, необережного розкриття конфіденційних даних самими користувачами тощо. Дана техніка зловживання діючими обліковими даними застосовується порушниками для широкого спектру задач: отримання початкового доступу, збереження, підвищення привілеїв чи ухилення від захисту [79].

Скомпрометовані облікові дані можуть використовуватися для обходу засобів управління доступом, розміщених на різних ресурсах в системах та мережі, і навіть для постійного доступу до віддалених систем і доступним ззовні службам, таким як віртуальні приватні мережі, поштові системи, віддалений доступ до робочих столів тощо. Діючі облікові дані також можуть надавати підвищені привілеї для певних систем або доступ до обмежених областей мережі. Такий прийом застосовується для обережної компрометації системи без застосування активних засобів для зламу паролів [102].

Найчастішою причиною успішного застосування даної техніки є неправильна конфігурація рольової моделі доступу в системі або її відсутність. Суміщення ролей або перекриття дозволів для локальних, доменних і хмарних облікових записів в мережі систем дозволяє порушнику мати змогу перемикатися між обліковими записами і системами для досягнення високого рівня доступу [102].

Тому, успішне детектування техніки використання діючих облікових записів пов’язане із налаштуванням надійних політик аудиту дій з обліковими записами всередині організації та із зовнішніми службами. Пошук підозрілої поведінки слід зосередити насамперед у системах, в яких застосовуються загальні облікові записи користувачів, адміністраторів або служб. Прикладами таких випадків можуть бути: ситуація, коли один обліковий запис одночасно підключений до декількох систем; або кілька облікових записів одночасно підключені до однієї машини; облікові записи, що входять в систему в неробочий час тощо. Підозрілі дії можуть відбуватися з інтерактивних сеансів входу в систему, які використовуються для виконання бінарних файлів у віддаленій системі в якості конкретного облікового запису. Саме тому, при детектуванні слід здійснювати зіставлення інформації інших систем безпеки та інформацію для входу в систему (наприклад, у користувача є активний сеанс входу в систему, але він не здійснив підключення до віртуальної приватної мережі організації) [103].

Додатково, дану техніку можливо визначити шляхом проведення регулярного аудиту облікових записів домену та локальної системи, для виявлення тих, які могли бути створені зловмисником для забезпечення стійкості доступу до системи. Такі

перевірки також можуть включати підозрілу активацію облікових записів, що використовуються за замовчуванням, наприклад гостьові [103].

Запит ППЗ для детектування даної техніки було розроблено шляхом використання декількох ітерацій. По-перше, запит визначає підозріле перерахування користувачів у системі облікових даних Windows (для виявлення найбільш критичних або таких, що мають найширший перелік дозволів у системі). По-друге, запит відслідковує додавання нового типу облікових даних до системних програм, у тому числі утиліт. Насамкінець, ППЗ визначає нетиповий вхід користувача в систему, тобто такий, що виходить за межі звичайного робочого процесу користувача (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Діючі облікові записи», доцільно брати до уваги наступні міркування:

- перерахування користувачів у системі є цілком нормальним процесом, якщо ця процедура здійснюється легітимним доменним адміністратором, проте даний запит ППЗ звертає увагу на детектування такої процедури адміністраторами середньої ланки, для яких ця операція є нетиповою;
- для скорочення можливих помилкових спрацьовувань запиту ППЗ в частині нетипового входу користувачів, повинні бути присутні дані поведінкової аналітики за певний історичний період.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.14).

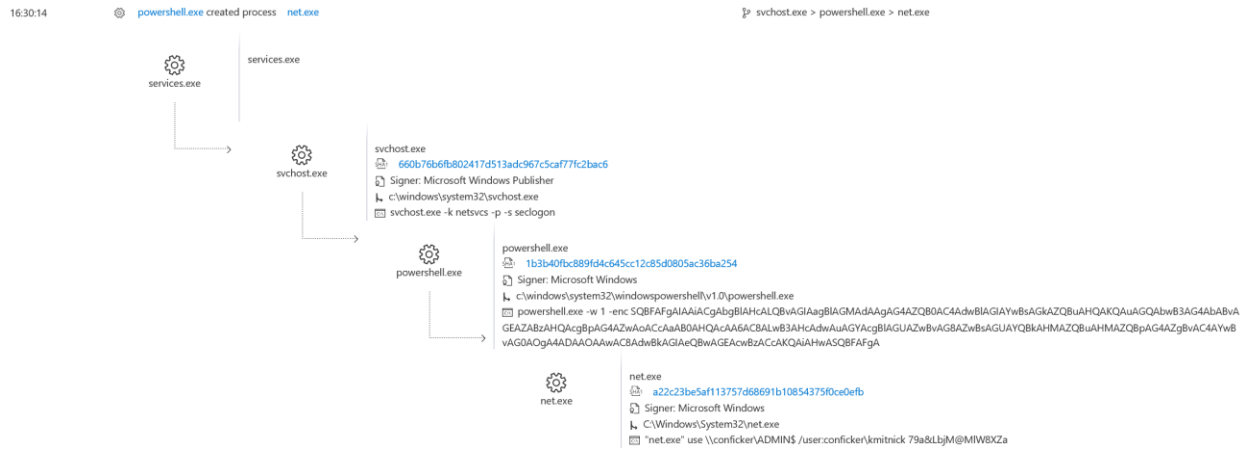


Рисунок 2.14 – Виявлення техніки «Діючі облікові записи» запитом ППЗ

Пропонований запит ППЗ виявив запуск системної утиліти net.exe Windows через PowerShell [76; 77] для успішної автентифікації новоствореного користувача, за допомогою облікових даних іншого користувача, які були раніше скомпрометовані.

2.8.13 Техніка: Виявлення системної інформації

Техніка виявлення системної інформації застосовується на етапі підготовки до основної атаки. Даний прийом характерний для спроб отримання детальної інформації про операційну систему і обладнання, включаючи версію, встановлені виправлення, пакети оновлень і архітектуру. Автоматичне виявлення використовується кіберзлочинцями для формування плану подальшої атаки, наприклад, необхідно скомпрометувати систему повністю або лише здійснити крадіжку певних даних тощо [79].

Для збору докладної інформації про систему найчастіше використовується процес systeminfo.exe. Даний процес запускає утиліту командного рядка, яка відображає системну інформацію щодо апаратних і програмних компонентів, які працюють в середовищі Windows [104, с. 43-51].

Методи виявлення інформації щодо системи і мережі зазвичай використовуються протягом всього циклу дослідження середовища порушником. Тому, при спробах детектувати дану техніку, дані та події слід розглядати не ізольовано, а як частину ланцюжка поведінки, який може призвести до суміжних дій, заснованих на отриманій інформації [104, с. 51-71].

Для ефективної протидії техніці корисне відстеження процесів та аргументів командного рядка для дій, які можуть бути зроблені для збору системної і мережевої інформації. Інструменти віддаленого доступу з вбудованими функціями можуть безпосередньо взаємодіяти з Windows для збору інформації. Інформацію також можна отримати за допомогою інструментів керування Windows (WMI) або PowerShell [105, с. 11].

Розроблений запит ППЗ передбачає моніторинг системної утиліти systeminfo.exe на предмет несподіваного її запуску сторонніми виконавчими файлами. Додатково, відслідковуються запити на предмет пошуку об'єктів облікових даних, комп'ютерів, а також мережевих сканувань (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Виявлення системної інформації», доцільно брати до уваги наступні міркування:

- на перших етапах використання пропонованого запиту ППЗ слід заздалегідь додати до виключень легітимні системні процеси, що можуть викликати запуск утиліти systeminfo.exe;
- процеси ошуку об'єктів облікових даних та комп'ютерів слід розглядати лише у поєднанні із викликами утиліти systeminfo.exe, інакше запит ППЗ демонструватиме високий рівень помилкових спрацьовувань.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.15).

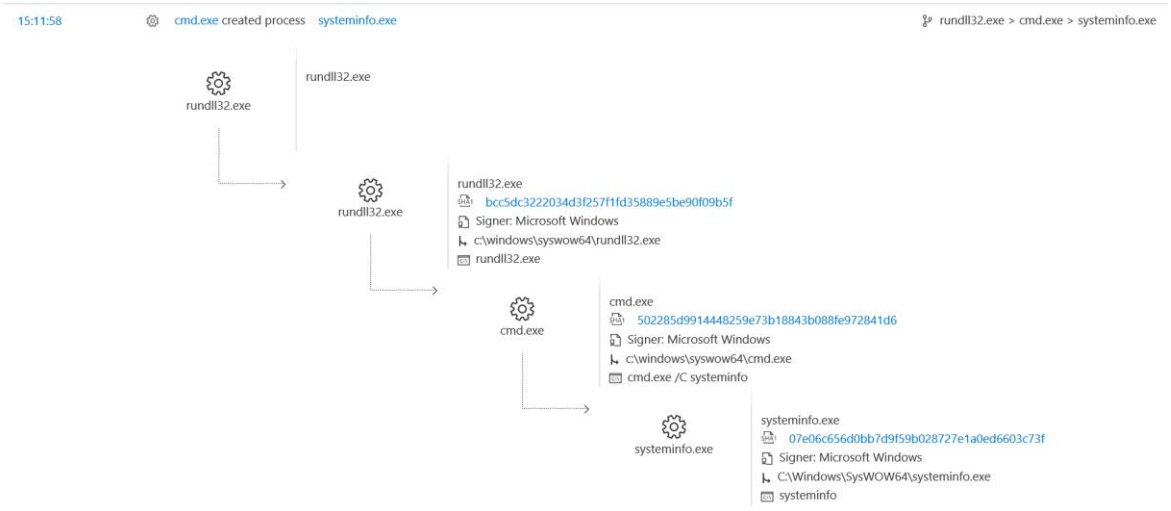


Рисунок 2.15 – Виявлення техніки «Виявлення системної інформації» запитом ППЗ

Встановлений запит ППЗ дозволив виявити ланцюжок подій використання командного рядка для запуску процесу systeminfo.exe [76; 77], який ініціює збір основної критичної інформації про систему.

2.8.14 Техніка: Виконання за подією

Встановлення сталості в системі та підвищення привілеїв є одними із головних задач при здійсненні повномасштабної атаки на систему. Саме тому, кіберзлочинці часто використовують системні механізми, які запускають виконання шкідливого програмного забезпечення на основі певних подій. Різні операційні системи мають засоби для моніторингу та підписки на такі події як: вхід в систему або інші дії користувача, наприклад запуск певних додатків або бінарних файлів [79].

Порушники можуть використовувати ці механізми як засоби підтримки постійного доступу до системи, що атакується, за допомогою багаторазового виконання шкідливого коду. Після отримання доступу до системи з'являється можливість створювати та змінювати тригери подій, щоб вказувати на шкідливий контент, який буде виконуватися при кожному виклику події [106, с. 35-43].

Оскільки виконання може виконуватися через проксі-сервер за допомогою облікового запису з більш високими привілеями такими як системні або службові облікові записи, кіберзлочинець може використовувати ініційовані механізми виконання для підвищення власних привілеїв [106, с. 51-56].

Одним із популярних способів експлуатації даної техніки є бінарне захоплення ключів. Це метод збереження доступу до системи, який дозволяє зловмисникові отримати доступ без автентифікації. Застосовується звичайна функція доступності, яка дозволяє активувати доступ безпосередньо з екрану входу в систему, натиснувши клавішу shift на клавіатурі декілька разів. Якщо порушнику вдасться замінити базовий виконавчий файл, він зможе виконати цей бінарний файл без автентифікації [106, с. 43-49].

Спираючись на вище сказане, при виявленні техніки виконання файлів за подіями в системі необхідно здійснювати моніторинг додавання або модифікації

механізмів, які можуть бути використані для запуску виконання на основі подій, особливо додавання аномальних команд, таких як виконання невідомих програм, відкриття мережевих сокетів або виходів до мережі. Також корисним є пошук змін, які не збігаються з оновленнями, виправленнями або іншими запланованими адміністративними діями [107, с. 32-34].

Механізми виконання за подіями можуть відрізнятися в залежності від операційної системи, але зазвичай знаходяться в центральних репозиторіях, в яких зберігається інформація щодо конфігурації: реєстр Windows, загальна інформаційна модель або певні іменовані файли, над якими можна здійснювати хешування і порівнювати із стандартно встановленими значеннями [107, с. 19-23].

Інструменти автозапуску Windows також можуть використовуватися для виявлення змін в тригерах виконання, які можуть бути спробами збереження зловмисного доступу до системи. Додатково, аномальні дерева викликів процесів для виконання інших команд можуть мати відношення до шкідливих дій [107, с. 23-30].

Пропонований запит ППЗ передбачає виявлення техніки шляхом здійснення декількох етапів. Спочатку відбувається запит відкладок, підключених за допомогою параметрів реєстру до процесів доступності. Далі здійснюється пошук запитів на перезапис файлів спеціальних можливостей. Насамкінець, виконується порівнювання стандартних значень хешів іменованих системних файлів із метаданими тих, що завантажуються через командний рядок, інструменти PowerShell тощо. Це дозволяє виявити можливу підробку легітимного програмного забезпечення для цілей розвідувальної діяльності зловмисника (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Виконання за подією», доцільно брати до уваги наступні міркування:

- стандартні значення хешів не завжди є постійними для певних системних файлів Windows, тому можливий невеликий відсоток помилкових спрацьовувань через мінливість певних значень;

- перелік ключів реєстру, представлений у запиті ППЗ, які відповідають за процеси доступності в системі, може не бути вичерпним, це потрібно враховувати як можливу невидиму зону для ППЗ та уточнювати значення.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.16).

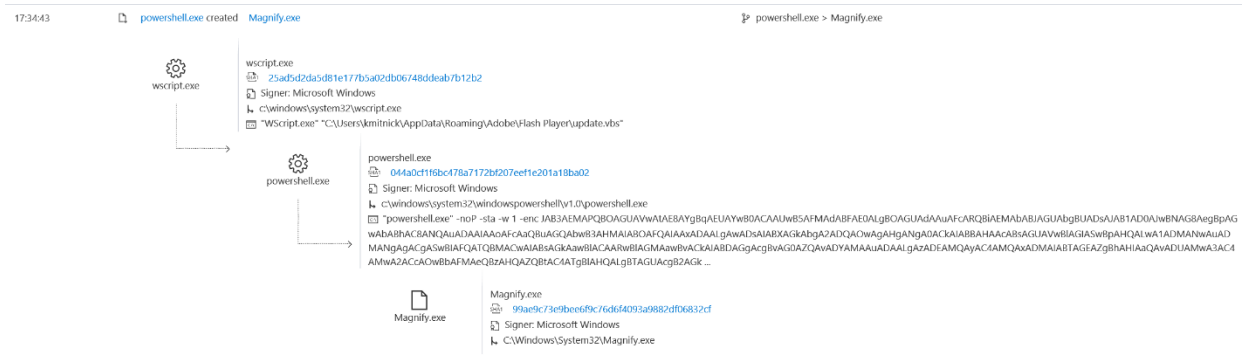


Рисунок 2.16 – Виявлення техніки «Виконання за подією» запитом ППЗ

Пропонований запит ППЗ дозволив виявити перезапис зловмисного файлу magnify.exe новим файлом, який містить той самий хеш, що і легітимний процес cmd.exe [76; 77]. Запит також показав інструментарій PowerShell, що застосовувався для здійснення всього процесу.

2.8.15 Техніка: Виконання користувачем

Застосування техніки, за якої відбувається виконання певної дії користувачем спирається на два прості механізми: запуск шкідливого файлу або натискання на підроблене посилання. За таких умов, користувачі можуть піддаватися соціальній інженерії, фішинговим кампаніям тощо [108].

З точки зору тактик MITRE ATT&CK, виконання користувачем часто відбувається незабаром після початкового доступу зловмисника до системи, однак ця техніка може здійснюватися і на інших етапах вторгнення [79].

Таким чином, для успішного детектування техніки необхідно здійснювати відстеження виконання і аргументів командного рядка для програм, які можуть бути

використані порушником для отримання початкового доступу, що вимагає взаємодії з користувачем. До даної категорії входять додатки для архівування інформації, які можна в перспективі декодувати у корисне навантаження [109, с. 9-11].

Розроблений запит ППЗ передбачає виявлення двох методів виконання зловмисного коду із офісних документів (макросів) та документів формату .pdf. Перший етап запиту визначає випадки, коли документ запускає інший процес і отримує інформацію про запущений бінарний файл. Далі запит перевіряє, чи є запущений виконавчий файл: підписаним Microsoft, непідписаним із низькою глобальною поширеністю або відомим LOLBIN. Заключний етап запиту визначає випадки, коли процес запуску документу впроваджує код в інший процес через віддалений виклик або запис безпосередньо до пам'яті системи (Додаток Б).

При використанні даного розробленого запиту ППЗ для детектування техніки «Виконання користувачем», доцільно брати до уваги наступні міркування:

- граничні значення глобальної поширеності зазначені у пропонованому запиті ППЗ вимагають точного налаштування для конкретного ІТ-середовища. Необхідно встановити максимально можливі низькі значення, при яких імовірність помилкових спрацьовувань буде прийнятною;
- часовий інтервал наявних даних за якими запит ППЗ здійснює пошук повинен бути досить коротким, щоб кількість хешів була менше 1000, що обумовлено обмеженнями функції FileProfile;
- деякі легітимні процеси можуть викликати LOLBIN, наприклад процес, що відповідає за запуск бібліотек, що динамічно підключаються;
- впровадження процесів із використанням методів, які не ґрунтуються на створенні віддаленого виклику, не виявляються пропонованим ППЗ.

Після застосування даного розробленого запиту із ППЗ для відповідної емульованої загрози були отримані наступні результати (Рисунок 2.17).

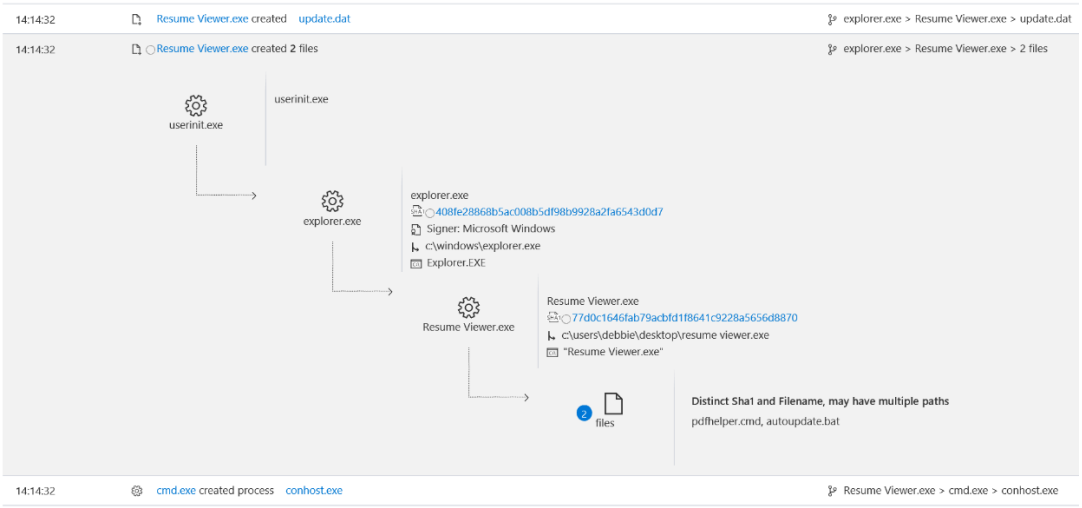


Рисунок 2.17 – Виявлення техніки «Виконання користувачем» запитом ППЗ

Розроблений запит ППЗ забезпечив детектування повної процедури того, як звичайний користувач системи відкрив шкідливий архів [76; 77], що містив функцію саморозпакування та здійснив доставку двох безпосередньо небезпечних файлів – pdfhelper.cmd та autoupdate.bat.

2.9 Висновки за розділом 2

У даному розділі було проведено ряд досліджень та розробок із наступними практичними результатами:

1. організовано лабораторне середовище для здійснення наукового дослідження, визначено обмеження обумовлені проведенням експерименту на базі операційних систем сімейства Windows;

2. проведено аналіз чотирьох методологій виявлення актуальних ТТП, на пошук яких має бути спрямоване дослідження;

3. розроблено критерії оцінки актуальності ТТП (поширеність у звітах різних років, присутність у кампаніях APT3 та APT29) та сформовано перелік технік, які складуть основу керівного зводу правил із проведення ППЗ;

4. здійснено емуляцію загроз у лабораторному середовищі для наповнення його даними і створення бази для написання запитів ППЗ;

5. розроблено та протестовано запити ППЗ для виявлення технік атак;

6. надано підтвердження визначення емульованих загроз пропонованими запитами ППЗ;

7. визначено обмеження та можливі помилкові спрацьовування розроблених запитів ППЗ.

РОЗДІЛ 3
ОЦІНКА ЕФЕКТИВНОСТІ РОБОТИ КЕРІВНОГО ЗВОДУ ПРАВИЛ ІЗ ПРОВЕДЕННЯ ППЗ

3.1 Виявлення загроз традиційними засобами

Для проведення належної оцінки ефективності роботи керівного зводу правил із проведення ППЗ, було здійснено порівняння виявлення емульованих загроз, зазначених у пункті 2.7, розробленими запитами ППЗ та традиційними засобами інформаційної безпеки.

Під традиційними засобами маються на увазі технології виявлення файлових та безфайлових атак, що базуються на автоматичному виявленні та не передбачають активного втручання з боку адміністратора системи. У розрізі технологій захисту кінцевих точок, зокрема під операційною системою Windows, такими рішеннями є клас Endpoint Detection and Response (EDR) [110, с. 81].

Технології класу EDR поєднують функціонал класичного антивірусу із реагуванням на інциденти, а також містять можливості: виявлення підозрілої поведінки, зменшення поверхні атаки, вбудованої «пісочниці», ідентифікації вразливостей, контролю мережевої активності, захисту від експлойтів, ізоляції на рівні обладнання, запобігання вторгненням [110, с. 82-83].

Microsoft Defender for Endpoint без застосовування функціоналу ППЗ є типовим представником рішень класу EDR та використовується у даній частині дослідження для оцінки виявлення емульованих загроз традиційним засобом інформаційної безпеки.

Нижче наведені результати виявлення визначених у пункті 2.6 технік традиційним функціоналом.

Техніка: Маскування. При застосуванні стандартного антивірусного функціоналу Microsoft Defender for Endpoint, механізм не виявив шкідливої активності. Адже, бінарна репутація та метадані для recycler.exe показують хеш і підпис видавця як WinRAR GmbH, що вказує на те, що файл насправді є службовою

програмою WinRAR, тому стандартному функціоналу не вдалося розпізнати маскування (Рисунок 3.1).

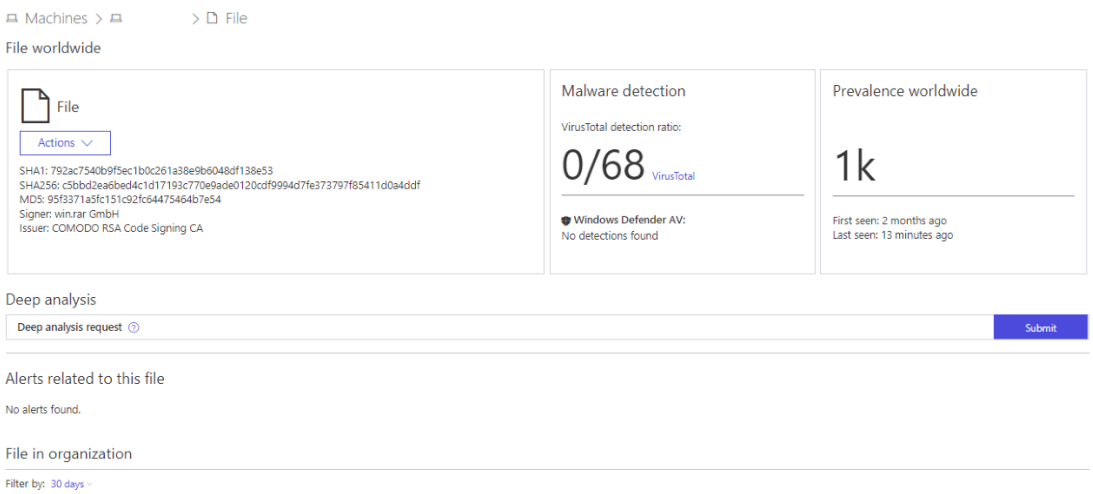


Рисунок 3.1 – Виявлення техніки «Маскування» антивірусом

Техніка: Інтерпретатор команд і скриптів. Антивірусний функціонал Microsoft Defender for Endpoint виявив лише частину шкідливої активності. А саме, спрацювало сповіщення про шкідливий PowerShell командлет. Однак, безпосередній зловмисний скрипт VBScript та використання DDL стандартний функціонал не зміг виявити (Рисунок 3.2).

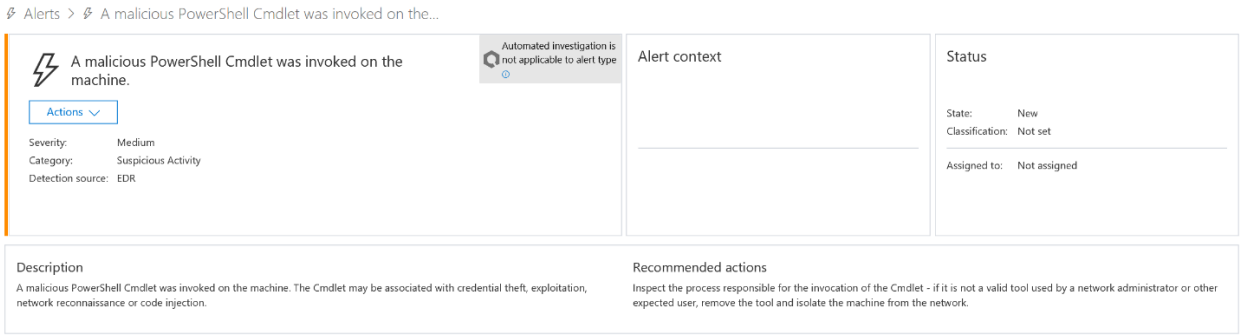


Рисунок 3.2 – Виявлення техніки «Інтерпретатор команд і скриптів» антивірусом

Техніка: Ін’єкція процесу. Функціонал захисту від експлойтів Microsoft Defender for Endpoint, запущений у режимі моніторингу, виявив здійснення доступу

від командного рядка до таблиці адрес експорту (масив покажчиків на функції у Windows, які містять адресу експортованої функції). Було виявлено доступ до модулю ntdll.dll, який містить деякі функції ядра, які допомагають в нормальному функціонуванні операційної системи Windows. Цей файл може одночасно обслуговувати різні програми, надаючи їм різні функції ядра, які підтримують продуктивність програми. Безпосередній зловмисний процес WerFault.exe традиційному антивірусу виявити не вдалося, однак детектування підозрілої активності у файлі, що відповідає за функції ядра, свідчить про задовільний результат. Однак, моніторинг таблиці адрес експорту не завжди є увімкненим за замовчуванням у системах Windows (Рисунок 3.3).

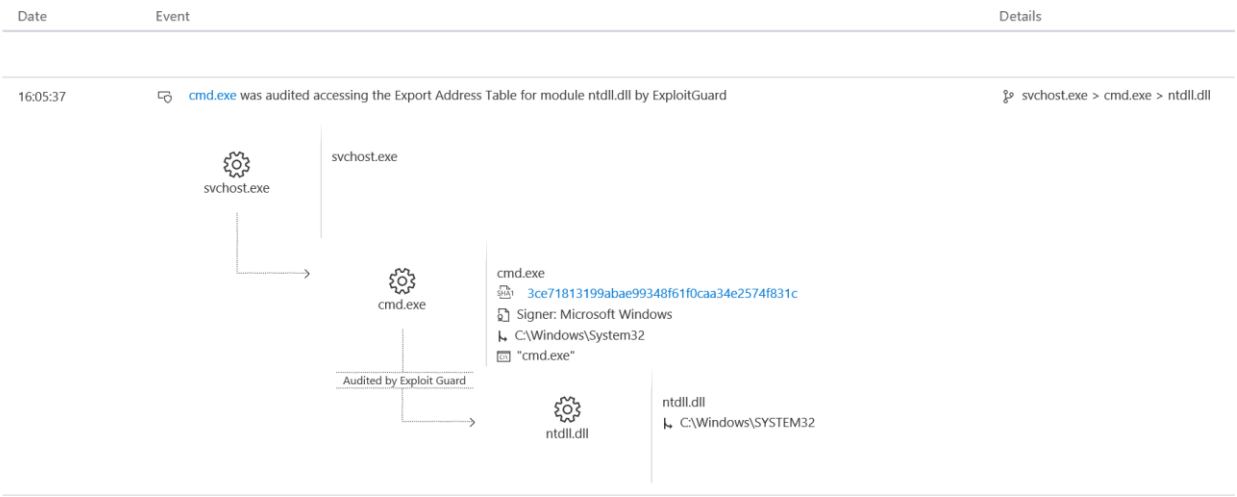


Рисунок 3.3 – Виявлення техніки «Ін'єкція процесу» антивірусом

Техніка: Виконання сценаріїв. Функціонал виявлення уникнення детектування антивірусним захистом Microsoft Defender for Endpoint виявив ініціалізацію сценаріїв від cmd.exe, а також ознаки атаки типу «Right-to-Left Override», яка характеризується зловживанням маловідомими методами кодування, які дозволяють легко замаскувати шкідливі виконувані файли (.exe) під відносно нешкідливі документи, такі як текст або файли оновлень. Таким чином, хоча і традиційний функціонал не дозволяє отримати настільки докладні відомості про техніку як процес ППЗ, даний інцидент був цілком виявлений і може бути повноцінно

зупинений звичайним антивірусним захистом Microsoft Defender for Endpoint (Рисунок 3.4).

Рисунок 3.4 – Виявлення техніки «Виконання сценаріїв» антивірусом

Техніка: Виконання автозапуску. Антивірусний компонент Microsoft Defender for Endpoint виявив підозрілу поведінку пов’язану із програмним забезпеченням віддаленого доступу, що використовувало реєстр. Однак, даний функціонал виявив загрозу на останньому етапі і пропустив значну частину підозрілих шаблонів асоційованих з реєстром, які успішно виявив запит ППЗ. З огляд на це, детектування інциденту можна вважати неповним (Рисунок 3.5).

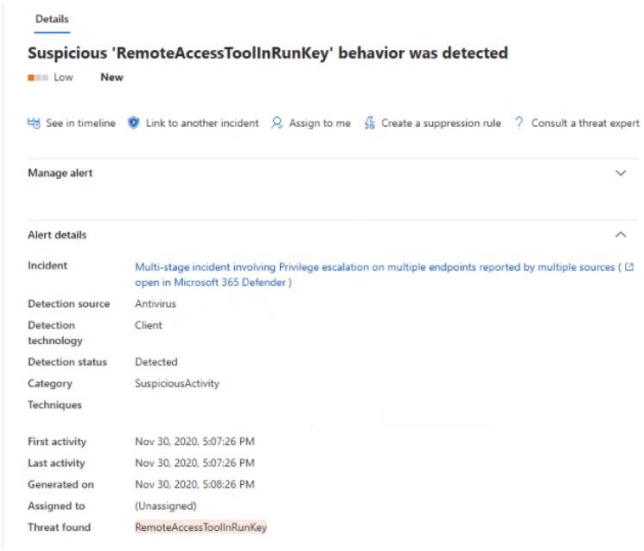


Рисунок 3.5 – Виявлення техніки «Виконання автозапуску» антивірусом

Техніка: Виявлення облікових записів. Компонент моніторингу підозрілої активності Microsoft Defender for Endpoint виявив послідовність розвідувальної діяльності в системі. Однак, більше подробиць щодо командного рядку та аргументів, які запускалися з нього, традиційному антивірусу виявити не вдалось (Рисунок 3.6).

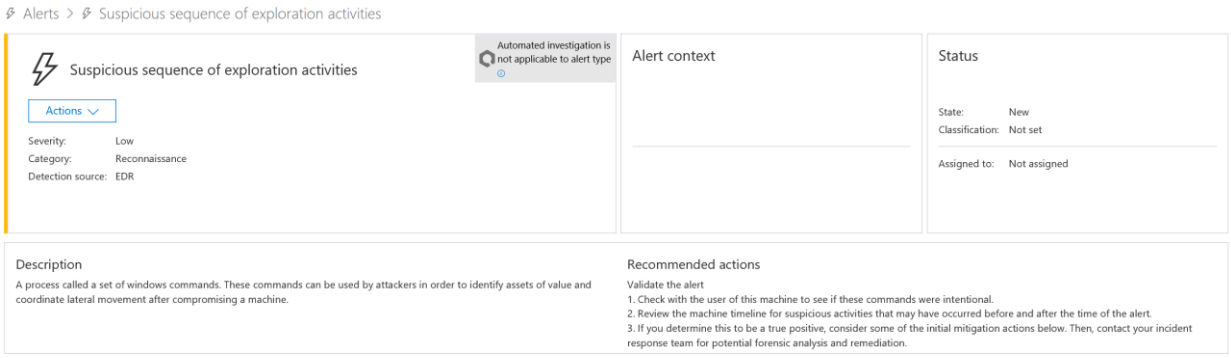


Рисунок 3.6 – Виявлення техніки «Виявлення облікових записів» антивірусом

Техніка: Дампінг облікових даних. Функціонал запобігання експлойтам, вбудований до Microsoft Defender for Endpoint, виявив читання пам’яті із критичними обліковими даними, в рамках сканування процесу LSASS. Таким чином, хоча традиційний функціонал і не дозволяє отримати настільки докладні відомості про техніку як процес ППЗ, даний інцидент був цілком виявлений і може бути повноцінно зупинений звичайним антивірусним захистом (Рисунок 3.7).

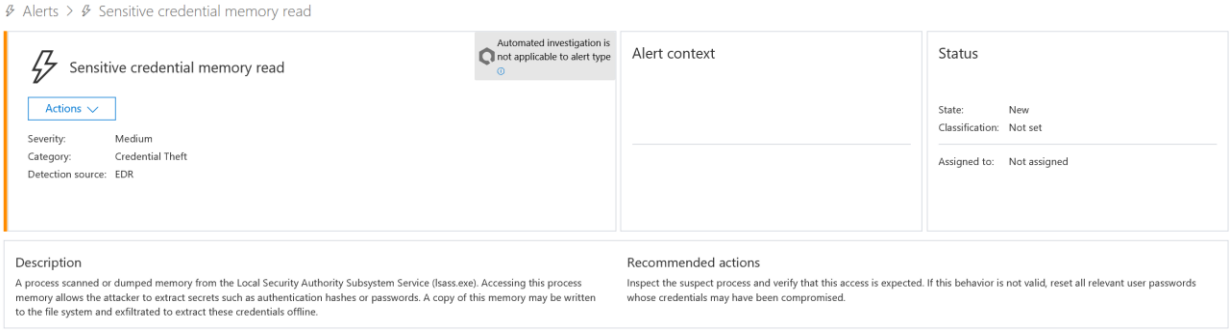


Рисунок 3.7 – Виявлення техніки «Дампінг облікових даних» антивірусом

Техніка: Обфускація файлів або інформації. Функціонал зменшення поверхні атаки Microsoft Defender for Endpoint виявив шкідливий файл, впроваджений легітимним процесом у закодованому вигляді, тільки після того, як в системі була здійснена спроба його використання програмою winword.exe. Таким чином, стандартний функціонал не виявив шкідливе використання легітимного процесу, а також сам закодований файл. З огляду на це, можна зробити висновок, що інцидент залишився би непоміченим, якби не спроба запуску програмою (Рисунок 3.8).

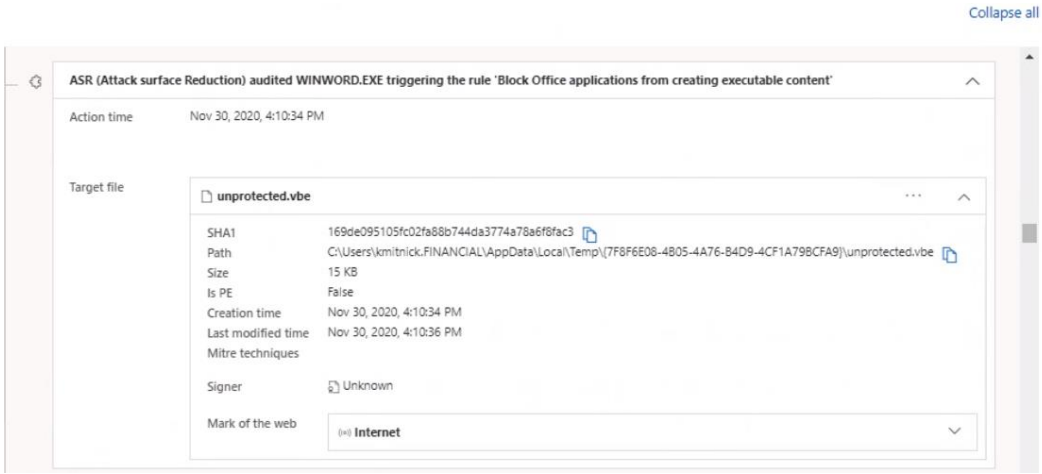


Рисунок 3.8 – Виявлення техніки «Обфускація файлів або інформації» антивірусом

Техніка: Проксі-виконання коду через підписані бінарні файли. Функціонал виявлення підозрілої активності Microsoft Defender for Endpoint виявив завантаження DLL із низькою репутацією підписаним виконавчим файлом і

присвоїв цьому інциденту низький пріоритет. Однак, традиційному функціоналу не вдалося виявити несанкціоноване виконання rundll32.exe файлу update.dat, спираючись на елементи панелі управління Windows CPL. Тому, можна зробити висновок, що атаку не було повноцінно виявлено, а був помічений лише один із її компонентів (Рисунок 3.9).

Рисунок 3.9 – Виявлення техніки «Проксі-виконання коду через підписані бінарні файли» антивірусом

Техніка: Заплановане завдання. Стандартний антивірусний функціонал Microsoft Defender for Endpoint виявив виконання шкідливого файлу через заплановану задачу. Однак, традиційному функціоналу не вдалося виявити атаку на ранніх стадіях – компрометація планувальника задач та виконання через командний рядок залишилися непоміченими (Рисунок 3.10).

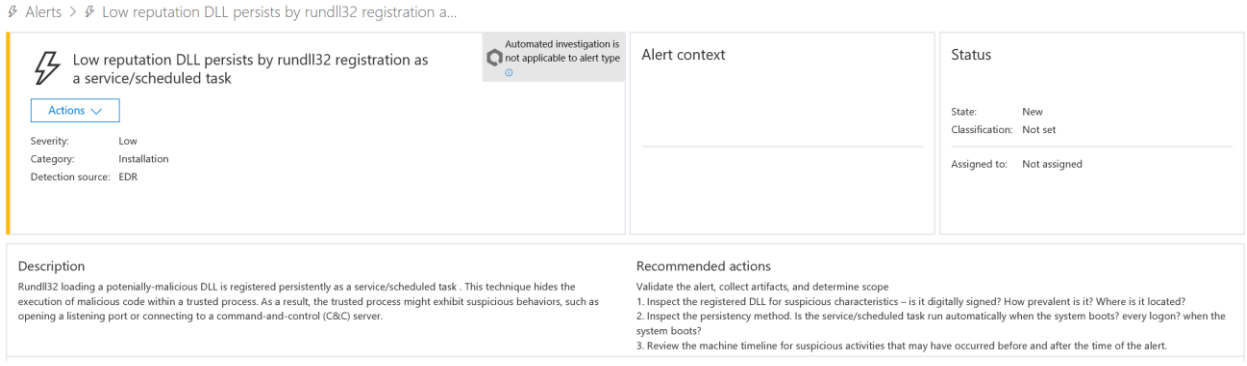


Рисунок 3.10 – Виявлення техніки «Заплановане завдання» антивірусом

Техніка: Дані з локальної системи. Антивірусний функціонал Microsoft Defender for Endpoint взагалі не зреагував на дану техніку. Мінімальні відомості про запуск PowerShell були знайдені в журналі подій, однак ніяких інцидентів

традиційний функціонал не зафіксував. До того ж в журналі не містилося відомостей щодо головної команди, пов'язаної із отриманням даних з локальної системи, а саме – пошуком дочірніх зв'язків. Таким чином, техніку не було виявлено (Рисунок 3.11).

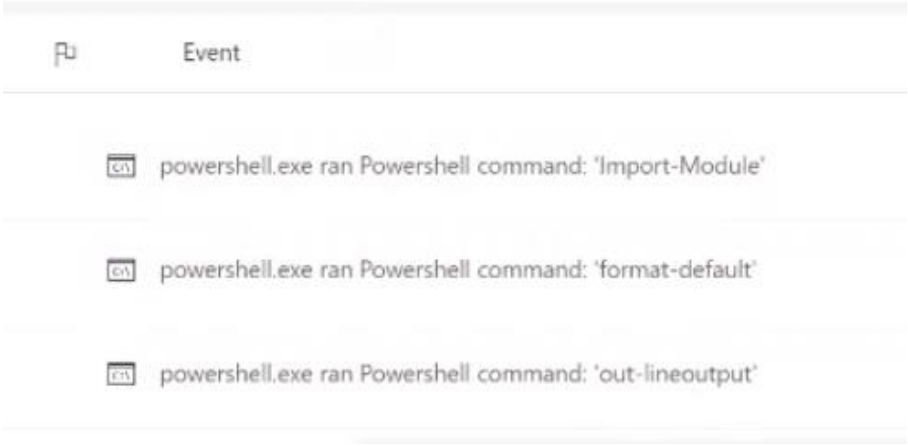


Рисунок 3.11 – Виявлення техніки «Дані з локальної системи» антивірусом

Техніка: Діючі облікові записи. Функціонал виявлення підозрілої активності Microsoft Defender for Endpoint спромігся виявити лише виконання скрипта PowerShell із зловмисним вмістом. Однак, процедура успішної автентифікації новоствореного користувача, за допомогою облікових даних іншого користувача через запуск системної утиліти net.exe Windows не була визначена. Таким чином, інцидент не було зафіксовано традиційним функціоналом (Рисунок 3.12).

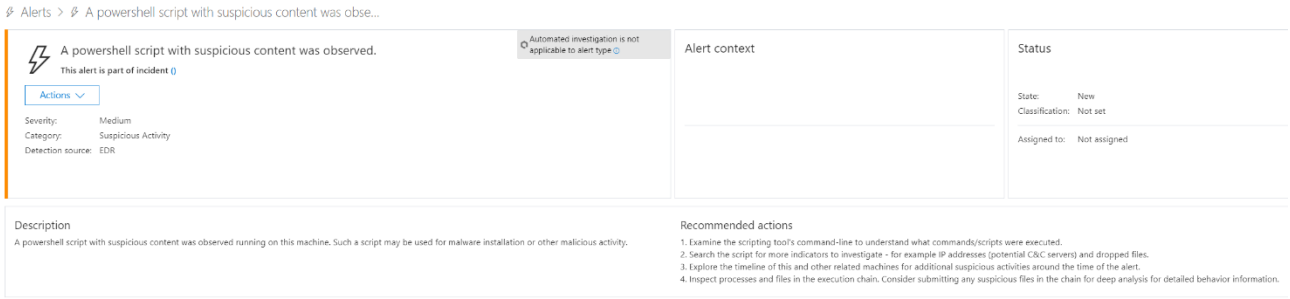


Рисунок 3.12 – Виявлення техніки «Діючі облікові записи» антивірусом

Техніка: Виявлення системної інформації. Антивірусний функціонал Microsoft Defender for Endpoint виявив підозрілу послідовній розвідувальної діяльності.

Повноцінно ідентифікувати ланцюжок подій використання командного рядка для запуску процесу systeminfo.exe, який ініціює збір основної критичної інформації про систему, традиційному функціоналу не вдалося, однак інцидент цілком можна вважати визначеним (Рисунок 3.13).

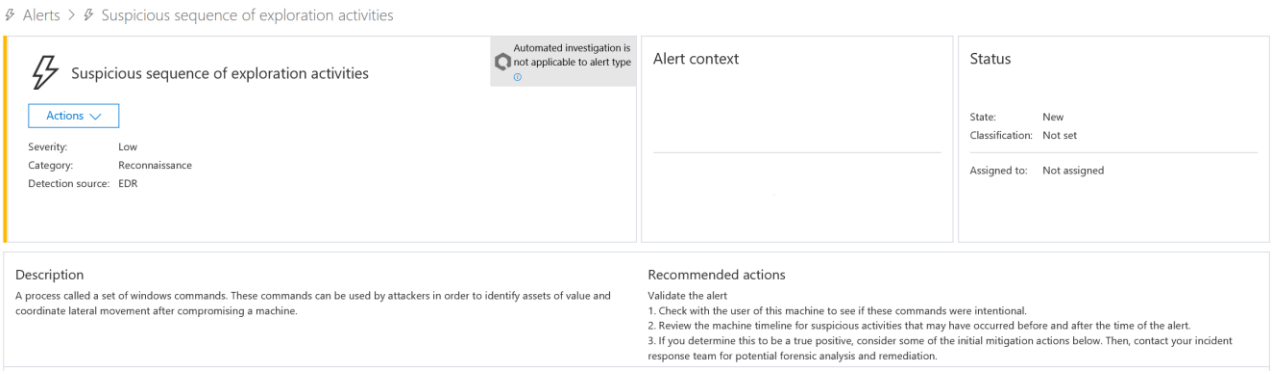


Рисунок 3.13 – Виявлення техніки «Виявлення системної інформації» антивірусом

Техніка: Виконання за подією. Антивірусний функціонал Microsoft Defender for Endpoint не виявив шкідливої активності. На додачу до цього, антивірусу не вдалося виявити, що файл magnify.exe є перезаписаним на легітимний процес. Тому, результат розпізнавання даної техніки можна вважати незадовільним (Рисунок 3.14).

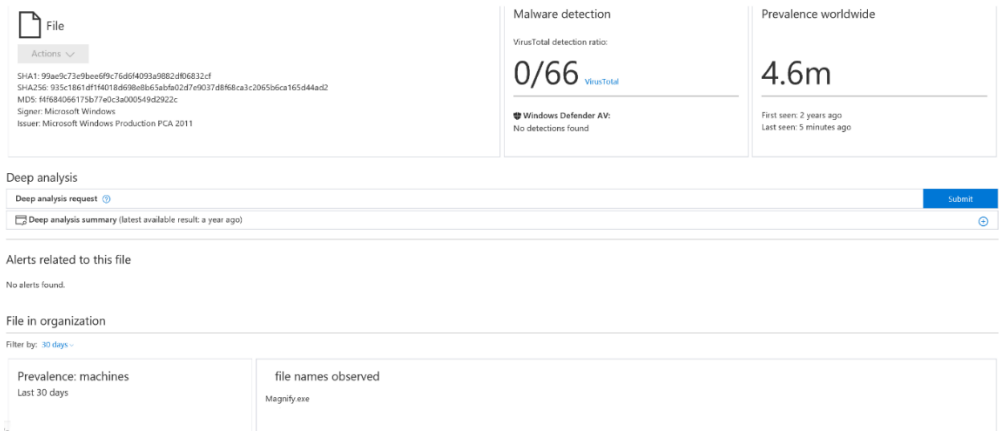


Рисунок 3.14 – Виявлення техніки «Виконання за подією» антивірусом

Техніка: Виконання користувачем. Функціонал отримання інформації щодо файлів антивірус виявив лише присутність архіву Resume Viewer.exe в системі, однак не зазначив ніякого інциденту, пов’язаного з ним (Рисунок 3.15).

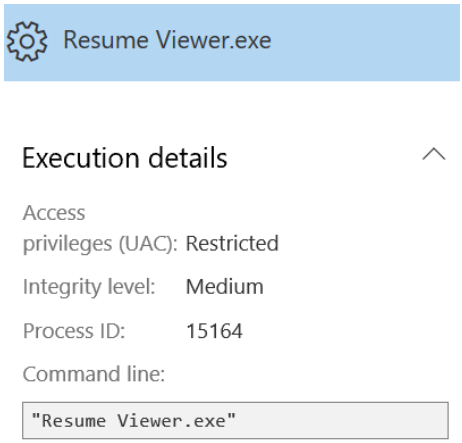


Рисунок 3.15 – Виявлення техніки «Виконання користувачем» антивірусом

До того ж два безпосередньо небезпечних файли, що були доставлені архівом – pdfhelper.cmd та autoupdate.bat, залишилися непоміченими. З огляду на це, детектування техніки традиційним функціоналом можна вважати нездійсненим.

3.2 Порівняння результатів виявлення загроз

Для складання детального порівняння результатів виявлення загроз традиційними засобами та запитами ППЗ, тактики, до яких відносяться перевірені техніки були розділені на групи за критерієм можливості заподіяння шкоди системі, яка піддається зловмисному впливу. Групи визначені від найменш шкідливих до найбільш небезпечних:

- група I – охоплює тактики початкового доступу і виконання, які є життєво важливими для кіберзлочинця та забезпечують планування подальших етапів атаки і підтримки контролю доступу до системи, у той же час не завдаючи значної шкоди цільовій системі;
- група II – охоплює тактики: постійності, ухилення від захисту, доступу з обліковими даними, які необхідні для досягнення проміжної мети атакуючого –

залишатися непоміченим і планомірно перейти до тактик безпосереднього впливу на систему;

- група III – охоплює тактики: підвищення привілеїв, виявлення і збору, які здійснюють безпосередній вплив і є прямою загрозою для цільової системи. Успішне виконання даних тактик може не тільки призвести до втрати облікових даних і постійного доступу до системи, а й забезпечити підвищення рівня привілеїв зловмисника на одній кінцевій робочій станції і створити умови для подальшого поширення в мережі.

Для кожної із груп тактик результати виявлення загроз були оцінені за всіма техніками, які до них відносяться, та за наступними параметрами:

- кількість стадій – величина, що вказує на кількість етапів загрози, які заявлені використовуваними джерелами для емуляції – MITRE ATT&CK Arsenal та Red Canary Atomic Red Team [76; 77];
- інцидент – вказує на відсоток спрацьовування традиційного функціоналу з генеруванням інциденту в системі та сповіщенням. Розраховується у відповідності до кількості зафіксованих стадій загрози;
- телеметрія – вказує на відсоток спрацьовування традиційного функціоналу із фіксуванням події в журналі, але без генерування інциденту в системі та сповіщення. Розраховується у відповідності до кількості зафіксованих стадій загрози;
- ППЗ – вказує на відсоток спрацьовування розроблених запитів ППЗ. Розраховується у відповідності до кількості зафіксованих стадій загрози;
- вага – величина від 1 до 3, що присвоюється кожній із технік в групі. Якщо техніка зустрічається лише в одній із груп, то вона має вагу 1, якщо в двох – 2, якщо в трьох – 3. Такий підхід обумовлений тим, що детектування технік, які можуть застосовуватися на різних етапах компрометації системи більш важливе, ніж детектування техніки, що може бути присутньою лише на одному із етапів.

Для параметрів: інцидент, телеметрія та ППЗ за всіма техніками в групі було вирахувано середнє (ситуація, коли всі ваги рівні) та середнє зважене виявлення (спирається на параметр ваги техніки) загроз (3.1) [111, с. 118-124].

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i \times x_i}{\sum_{i=1}^{n} w_i}, \quad (3.1)$$

де $\bar{x}$ – середнє та середнє зважене арифметичне виявлень загроз для параметрів: інцидент, телеметрія та ППЗ;

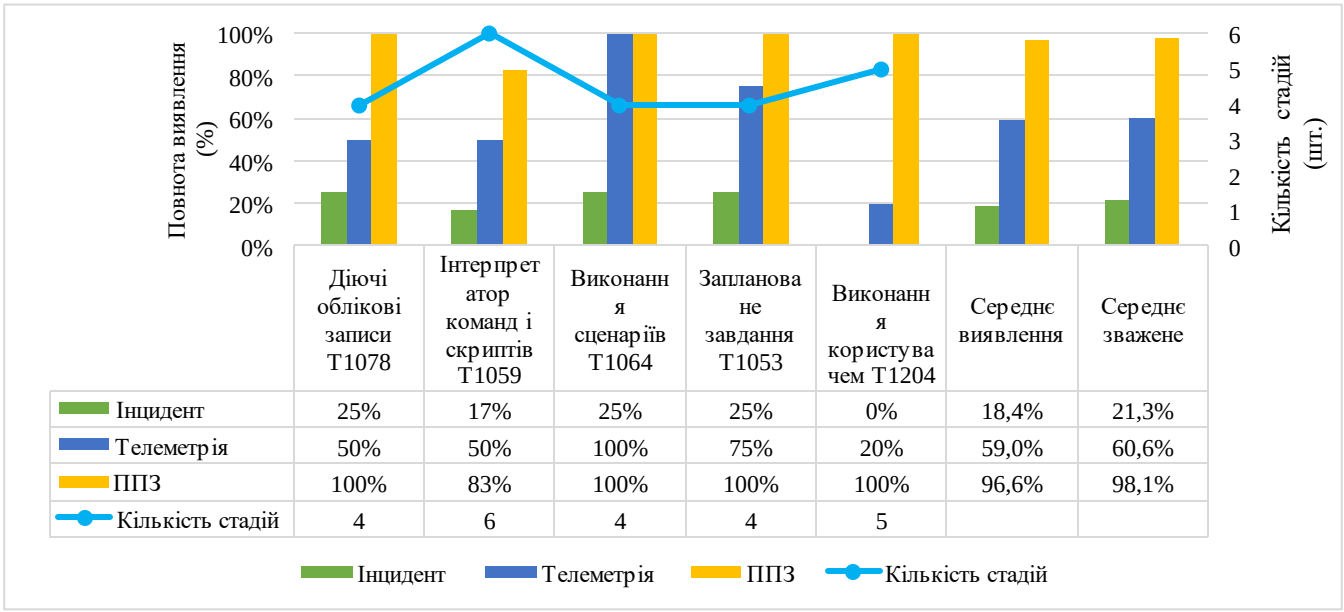
w_i – вага від 1 до 3 для кожної із технік в групі;

x_i – значення виявлень загрози по кожній із технік в групі за параметрами: інцидент, телеметрія та ППЗ;

i – лічильник виявлень загроз за техніками в групі (від 1);

n – загальна кількість виявлень загроз.

Результати оцінки виявлення загроз (Рисунки 3.16 – 3.18) наведені для трьох груп тактик. Для кожної із технік, які входять до групи тактик, наведені відсотки спрацьовувань із детектування загроз різним функціоналом. Показники середнього та середнього зваженого виявлення наведені для розуміння загальної спроможності кожного із видів функціоналу протидіяти загрозам, що входять до певної групи.



	Діючі облікові записи T1078	Інтерпретатор команд і скриптів T1059	Виконання сценаріїв T1064	Запланована завдання T1053	Виконання користувачем T1204	Середнє виявлення	Середнє зважене
Інцидент	25%	17%	25%	25%	0%	18,4%	21,3%
Телеметрія	50%	50%	100%	75%	20%	59,0%	60,6%
ППЗ	100%	83%	100%	100%	100%	96,6%	98,1%
Кількість стадій	4	6	4	4	5		

Рисунок 3.16 – Порівняння результатів виявлення загроз для групи I

Результати повноти виявлення технік групи I показують доволі низький рівень спрацьовування інцидентів і 60,6% збору телеметрії, в той час як для ППЗ результат

становить 98,1%. Це свідчить про те, що техніки, пов’язані із початковою розвідкою, які не є повноцінними атаками, слабко виявляються традиційним функціоналом.

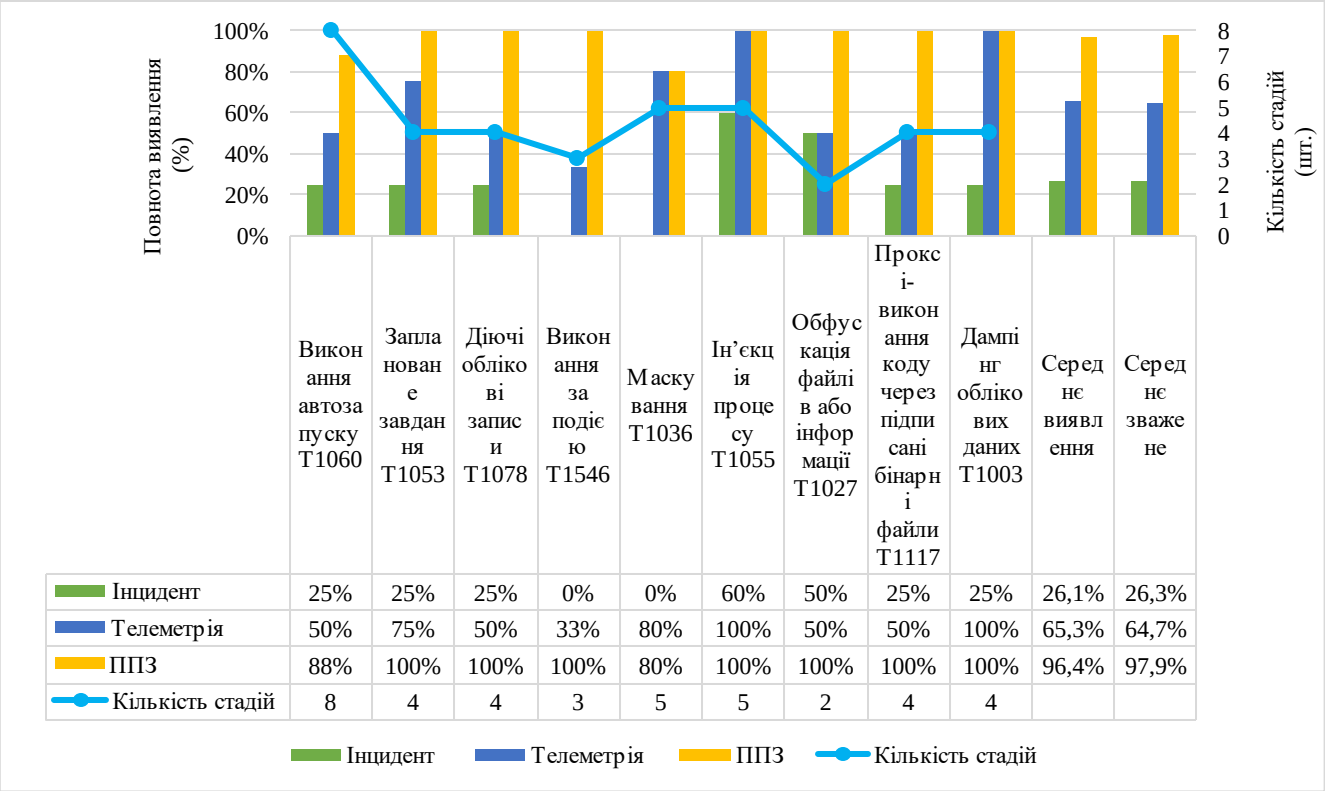


Рисунок 3.17 – Порівняння результатів виявлення загроз для групи II

Результати повноти виявлення технік групи II демонструють кращі результати, порівняно із групою I у розрізі виявлення традиційним функціоналом. Рівень виявлення ППЗ трохи нижче за попередній результат. Тобто, для технік, що пов’язані із закріпленням в системі, проміжним етапом атаки, традиційний функціонал проявляє себе більш надійно.

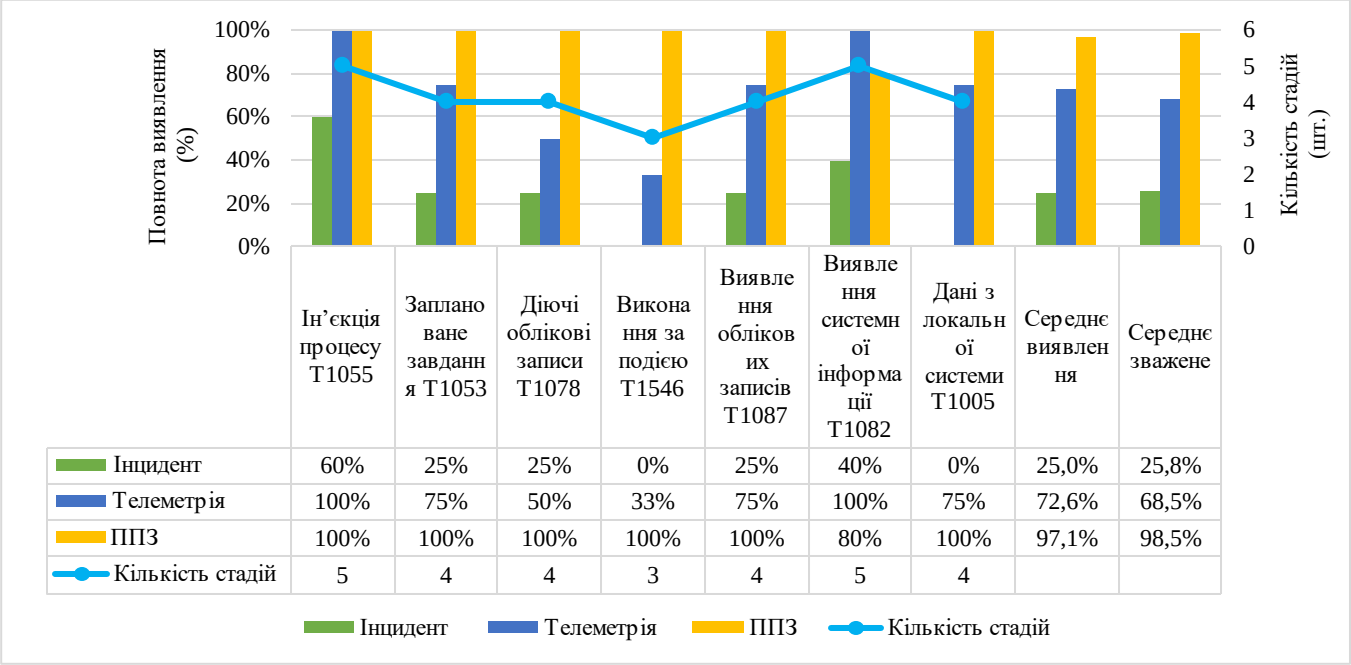


Рисунок 3.18 – Порівняння результатів виявлення загроз для групи III

Результати повноти виявлення технік групи III визначаються найбільш високим рівнем фіксації телеметрії, дане свідчить про те, що атакуючі залишають найбільше слідів у системі під час активного впливу на неї, тому краще ідентифікуються традиційними інструментами. Відповідно до зростання кількості телеметрії, рівень виявлення ППЗ також показав кращі результати, ніж у попередніх групах.

Для приведення показників інцидентів та телеметрії до єдиного значення виявлення традиційним функціоналом, для них були розраховані середні зважені значення виявлення (3.1) за кожною із технік із вагою 1 для телеметрії та 2 для інцидентів, оскільки автоматичне виявлення і сповіщення, важливіше за простий збір даних про подію.

Результати оцінки ефективності виявлення загроз традиційним функціоналом та розробленими запитами ППЗ наведені у вигляді графіків (Рисунок 3.19).

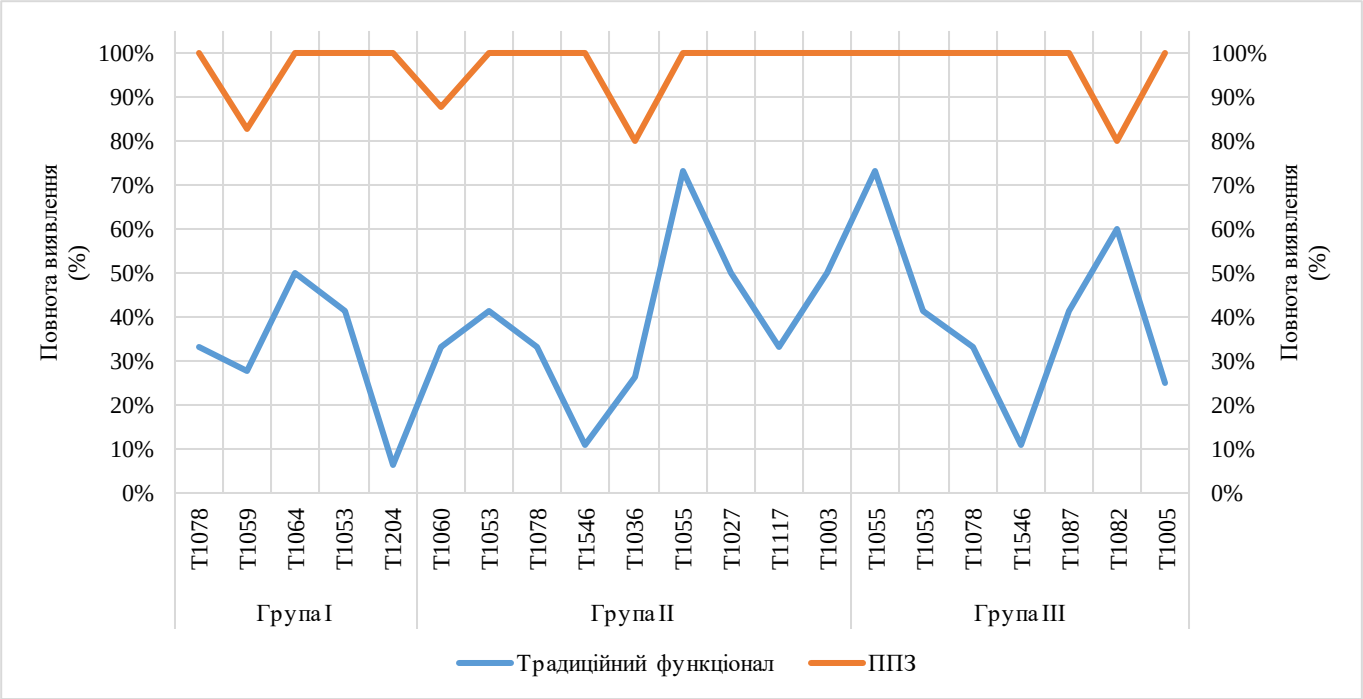


Рисунок 3.19 – Оцінка ефективності виявлення загроз традиційним функціоналом та розробленими запитами ППЗ

Для агрегації результатів та формування остаточної оцінки ефективності, для кожної із груп тактик було вирахувано середнє зважене виявлення (3.1) для традиційного функціоналу та ППЗ.

Додатково, для кожної із груп тактик та для кожного із підходів, було розраховано стандартне відхилення, тобто показник розсіювання значень повноти виявлень або змінність можливостей детектування кожного із підходів (3.2) [112, с. 15-18].

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}, \quad (3.2)$$

де S – стандартне відхилення у детектуванні технік традиційним функціоналом та ППЗ;

x_i – значення показників виявлення загроз для всіх технік у групі кожного із підходів;

$\bar{x}$ – середнє арифметичне виявлення загроз для всіх технік у групі кожного із підходів;

i – лічильник виявлень загроз за техніками в групі (від 1);

n – загальна кількість виявлень загроз у групі.

Результати остаточної оцінки ефективності виявлення загроз традиційним функціоналом та запитами ППЗ наведені в Таблиці 3.1.

Таблиця 3.1

Оцінка ефективності виявлення загроз традиційним функціоналом та ППЗ

	Група I	Група II	Група III
Традиційний функціонал (середнє зважене виявлення)	34,4%	39,1%	40,0%
ППЗ (середнє зважене виявлення)	98,1%	97,9%	98,5%
Ефективність ППЗ (кількість разів)	2,85	2,50	2,46
Традиційний функціонал (стандартне відхилення)	16,41%	17,54%	20,90%
ППЗ (стандартне відхилення)	7,60%	7,33%	7,56%
Стабільність ППЗ (кількість разів)	2,16	2,40	2,76

Отже, показники свідчать про те, що в середньому ППЗ є ефективнішим за виявлення загроз традиційними засобами у 2,6 разів. На додачу до цього, підхід на базі ППЗ є стабільнішим за традиційний функціонал у 2,4 разів, що підтверджує більшу впевненість у виявленні загроз.

3.3 Формалізація керівного зводу правил із проведення ППЗ

Для підбиття підсумків наукового дослідження даної дипломної роботи, керівний звід правил із проведення ППЗ формалізовано за пунктами:

- назва техніки та її маркування у відповідності до MITRE ATT&CK;

- критичність техніки, з точки зору її присутності у групах тактик I–III (див. пункт 3.2);
- помилкові спрацьовування, які можуть виникнути при застосуванні ППЗ для виявлення техніки;
- обмеження, які необхідно враховувати при здійсненні ППЗ для виявлення техніки;
- середній відсоток виявлення загроз, які відносяться до певної техніки за допомогою ППЗ та традиційного функціоналу (Таблиця 3.2).

Таблиця 3.2

Формалізований керівний звід правил із проведення ППЗ

Техніка	Критичність	Помилкові спрацьовування	Обмеження	ППЗ (%)	Традиційний функціонал (%)
Маскування T1036	II	• Установники програмного забезпечення, які запускають виконавчі файли, використовуючи інші імена (потрібні виключення).	• Програмні пакети, які мають власну копію LOLBIN, створюють додаткову поверхню для атак; • зловмисник може раніше змінити заголовок метаданих і вихідне ім'я файлу (необхідні додаткові правила).	80	26,7
Інтерпретатор команд і скриптів T1059	I	• Може знадобитись фільтрація за всіма виконавчими файлами, підписаними Microsoft; • може знадобитись фільтрація за всіма підписаними виконавчими файлами.	• Може знадобитись включення повного шляху і його зіставлення до запиту; • присутнє обмеження у виявленні додатків, які завантажили DLL з блоку вбудованої пам'яті.	83	28
Ін'єкція процесу T1055	II, III	–	• Можливе виявлення лише процесів, через виклик API NtAllocate VirtualMemoryRemote; • обмежене виявлення за наявності перевіреного сертифіката.	100	73,3

продовження табл. 3.2

Виконання сценаріїв T1064	I	• Граничні значення вимагають підлаштування до конкретного IT-середовища; • існують легітимні додатки, які використовують CreateRemoteThread для впровадження процесу.	• Часовий інтервал повинен бути досить коротким, щоб кількість хешів була менше 1000.	100	50
Виконання автозапуску T1060	II	• Зміни деяких параметрів конфігурації автозапуску можуть відбуватися при легітимному програмному забезпеченні.	• Рекомендується поєднувати запит ППЗ з іншими спрацьовуваннями антивірусу, для кращого виявлення ознак бічного переміщення.	88	33,3
Виявлення облікових записів T1087	III	• Можуть знадобитися виключення для WMI.	• Необхідно враховувати контекст застосування команд утиліти Net.	100	41,7
Дампінг облікових даних T1003	II	• Не всі прапори запису у пам’яті процесу LSASS можуть свідчити про атаку (необхідний додатковий контекст).	–	100	50
Обфускація файлів або інформації	II	• Деякі із батьківських процесів запускають свої дочірні процеси і відразу ж завершуються.	• Деякі процеси можуть мати інші легітимні батьківські процеси; • обмеження у виявленні для тих процесів, які не зазначені у запиті ППЗ.	100	50
Заплановане завдання T1053	I - III	• Не кожен непідписаний файл, що виконується процесом планувальника задач є шкідливим; • глобальна поширеність, встановлена Microsoft, не є абсолютно вірною.	• Атакуючий може використовувати підписаний бінарний файл, для непрямого запуску шкідливого коду.	100	41,7

продовження табл. 3.2

Проксі-виконання коду через підписані бінарні файли T1117	II	• Граничні значення вимагають підлаштування до конкретного IT-середовища; • деякі безпечні програми можуть додавати допустимі файли CPL.	• Часовий інтервал повинен бути досить коротким, щоб кількість унікальних хешів CPL була менше 1000.	100	33,3
Дані з локальної системи T1005	III	• При моніторингу виконання команд PowerShell, слід виключити можливість їх застосування справжнім адміністратором системи.	–	100	25
Дійчі облікові записи T1078	I - III	• Перерахування користувачів у системі інколи є цілком нормальним процесом.	• Повинні бути присутні дані поведінкової аналітики за певний історичний період.	100	33,3
Виявлення системної інформації T1082	III	• Слід заздалегідь додати до виключень легітимні системні процеси, що можуть викликати запуск утиліти systeminfo.exe; • процеси слід розглядати у поєднанні із викликами утиліти systeminfo.exe.	–	80	60
Виконання за подією T1546	II, III	• Стандартні значення хешів не завжди є постійними для певних системних файлів Windows.	• Перелік ключів реєстру може не бути вичерпним.	100	11

Виконання користувачем T1204	I	• Граничні значення вимагають підлаштування до конкретного IT-середовища; • деякі легітимні процеси можуть викликати LOLBIN.	• Часовий інтервал повинен бути досить коротким, щоб кількість хешів була менше 1000; • обмежене виявлення із використанням методів, які не створюють віддаленого виклику.	100	6,7

Пропонований керівний звід правил із проведення ППЗ позиціонується як базовий для виявлення зловмисних технік, що наразі присутні у більшості кампаній атак.

Даний звід правил є допоміжним інструментом для традиційного функціоналу протидії загрозам (антивірусним рішенням та рішенням класу EDR). Адже, для технік, що виявляються традиційним функціоналом, наприклад на 70%, ППЗ може бути не обов'язковим для успішної ліквідації загрози. У той час, як для інших загроз із низьким середнім рівнем спрацьовування сповіщень традиційного функціоналу (особливо тих технік, які відносяться до початкової компрометації системи), виконання запитів ППЗ є вирішальний інструментом для своєчасного детектування та ліквідації інциденту.

3.4 Висновки за розділом 3

У даному розділі було проведено оцінку ефективності пропонованого керівного зводу правил із проведення ППЗ із наступними науковими результатами:

1. продемонстровано результати виявлення загроз, що пов'язані із досліджуваними техніками, традиційним (антивірусним) функціоналом із зазначенням інцидентів, які були автоматично сформовані, та зібраної телеметрії;

2. проведено порівняння результатів виявлення загроз за допомогою ППЗ та традиційних засобів з огляду детектування технік різних груп критичності;

3. доведено, що ефективність ППЗ у виявленні загроз перевищує виявлення традиційними засобами у 2,6 разів і є стабільнішим за традиційний функціонал у 2,4 разів;

4. формалізовано керівний звід правил із проведення ППЗ із зазначенням: обмежень, можливих помилкових спрацьовувань, очікуваних результатів, правил / рекомендацій щодо застосування.

ВИСНОВКИ

У дипломній роботі розв'язано актуальне завдання створення керівного зводу правил із проведення проактивного пошуку загроз. В ході розв'язання поставлених задач були отримані наступні наукові та практичні результати:

1. Проведено аналітичний огляд проблеми проактивного пошуку кіберзагроз. Здійснено аналіз наявних методів проактивного пошуку та створено їх порівняльну характеристику із виділенням особливостей та недоліків.

2. Обґрунтовано актуальність та необхідність створення керівного зводу правил із проведення проактивного пошуку загроз. Створено лабораторне середовище на базі хмарної служби Microsoft Defender for Endpoint для проведення наукового експерименту виявлення кіберзагроз та визначено обмеження обумовлені проведенням експерименту на базі операційних систем сімейства Windows.

3. Обґрунтовано підбір набору кіберзагроз із розробкою критеріїв оцінки їх актуальності (поширеність, присутність у кампаніях) шляхом аналізу чотирьох методологій. Здійснено емуляцію загроз у лабораторному середовищі для наповнення його даними і створення бази для написання запитів проактивного пошуку загроз.

4. Розроблено та протестовано запити проактивного пошуку для виявлення технік атак. Адаптовано мову запитів KQL для вирішення даної задачі.

5. Надано підтвердження визначення емульованих загроз пропонованими запитами пошуку із визначенням їхніх обмежень та можливих помилкових спрацьовувань.

6. Виконано порівняння результатів виявлення загроз за допомогою проактивного пошуку та традиційних засобів із зазначенням інцидентів, які були автоматично сформовані, та зібраної телеметрії, з огляду детектування технік різних груп критичності.

7. Формалізовано керівний звід правил із проведення проактивного пошуку загроз. Доведено ефективність пропонованого методу проактивного пошуку загроз у виявленні атак за показниками кількості правильних спрацьовувань та стабільності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fuchs M. SANS 2019 Threat Hunting Survey: The Differing Needs of New and Experienced Hunters / M. Fuchs, J. Lemon. // SANS Institute. – 2019. – 18 P.

2. Fuchs M. SANS 2020 Threat Hunting Survey Results / M. Fuchs, J. Lemon. // SANS Institute. – 2020. – 17 P.

3. Brown R. 2021 SANS Cyber Threat Intelligence (CTI) Survey / R. Brown, R. M. Lee. // SANS Institute. – 2021. – 19 P.

4. Lee R. M. The Who, What, Where, When, Why and How of Effective Threat Hunting [Electronic resource] / R. M. Lee, R. Lee // SANS Institute. – 2016. – Resource access mode: https://www.sans.org/reading-room/whitepapers/analyst/membership/36785.

5. Shoard P. Hype Cycle for Security Operations [Electronic resource] / Pete Shoard // Gartner, Inc. – 2020. – Resource access mode: https://www.gartner.com/doc/reprints?id=1-24D23P6S&ct=201013&st=sb.

6. Жилін А. Функціональна модель ситуаційного центру кіберзахисту / А. Жилін, М. Худинцев, М. Літвінов. // P-ISSN 2411-1031. Information Technology and Security. – July-December 2018. – Vol. 6. Iss. 2 (11). – С. 51–67.

7. Жилін А. Функціональна модель оцінювання рівня зрілості SOC на основі моделі зрілості / А. Жилін, Г. Голич, М. Худинцев. // Захист інформації. – Липень-Вересень 2019. – Том 21, №3. – С. 182–193.

8. Гахов С. О. Аналіз методів виявлення подій та інцидентів інформаційної та кібернетичної безпеки SIEM-системами / С. О. Гахов. // Сучасний захист інформації. – 2018. – №4 (36). – С. 11–16.

9. Кифоренко І. В. Метод аналізу APT атак за допомогою технік MITRE та алгоритму нечіткого пошуку : дис.: 125 Кібербезпека / Кифоренко І. В. – Київ, 2019. – 129 с.

10. Lee R. M. Collection Management Frameworks: Looking Beyond Asset Inventories In Preparation for and Response to Cyber Threats / R. M. Lee, B. Miller, M. Stacey. // Dragos, Inc. – 2018. – 13 P.

11. Johansson T. Exploring the Human Fingerprints on Malware [Electronic resource] / T. Johansson, R. M. Lee // SANS Institute. – 2019. – Resource access mode: https://www.sans.org/reading-room/whitepapers/threats/paper/39275.

12. Caltagirone S. The Four Types of Threat Detection With Case-Studies in Industrial Control Systems (ICS) / S. Caltagirone, R. M. Lee. // Dragos, Inc. – 2018. – 15 P.

13. Lee R. M. Generating Hypotheses for Successful Threat Hunting [Electronic resource] / R. M. Lee, D. Bianco // SANS Institute. – 2016. – Resource access mode: https://www.sans.org/reading-room/whitepapers/threats/generating-hypothesessuccessful-threat-hunting-37172.

14. Crowley C. Hunting in Network Telemetry [Electronic resource] / C. Crowley // SANS Institute. – 2021. – Resource access mode: https://www.sans.org/reading-room/whitepapers/analyst/membership/40170.

15. Van Os R. TaHiTI Threat Hunting Methodology [Electronic resource] / R. van Os, M. Bakker, R. Bouman et al. // FI-ISAC NL Publication. – 2018. – 38 P. – Resource access mode: https://www.betaalvereniging.nl/wp-content/uploads/DEF-TaHiTI-Threat-Hunting-Methodology.pdf

16. Gunter D. A Practical Model for Conducting Cyber Threat Hunting [Electronic resource] / D. Gunter, M. Seitz // SANS Institute. – 2018. – 15 P. – Resource access mode: https://www.sans.org/reading-room/whitepapers/threathunting/practical-model-conducting-cyber-threat-hunting-38710.

17. Schmitt S. Intelligent Threat Hunting in Software-Defined Networking / S. Schmitt, F. I. Kandah, D. Brownell. // IEEE International Conference on Consumer Electronics (ICCE). – 2019. – P. 1–5, doi: 10.1109/ICCE.2019.8661952.

18. Karev D. Cyber threat hunting through the use of an isolation forest / D. Karev, C. McCubbin, R. Vaulin. // Proceedings of the 18th International Conference on Computer Systems and Technologies. – 2017. – P. 163–170, doi: 10.1145/3134302.3134319.

19. Wen S. Detecting and Predicting APT Based on the Study of Cyber Kill Chain with Hierarchical Knowledge Reasoning / S. Wen, N. He, H. Yan. // Proceedings of the

2017 VI International Conference on Network, Communication and Computing. – 2017. – P. 115–119, doi: 10.1145/3171592.3171641.

20. The Design of Cyber Threat Hunting Games: A Case Study / [M. N. S. Miazi, M. M. A. Pritom, M. Shehab et al.] // 26th International Conference on Computer Communication and Networks (ICCCN). – 2017. – P. 1 – 6, doi: 10.1109/ICCCN.2017.8038527.

21. Chuvakin A. How to Hunt for Security Threats [Electronic resource] / A. Chuvakin // Gartner, Inc. – 2017. – Resource access mode: https://www.gartner.com/en/documents/3664330.

22. Wafula K. CARVE: A Scientific Method-Based Threat Hunting Hypothesis Development Model / K. Wafula, Y. Wang. // IEEE International Conference on Electro Information Technology (EIT). – 2019. – P. 1 – 6, doi: 10.1109/EIT.2019.8833792.

23. Akacki D. 5 types of Threat Hunting [Electronic resource] / D. Akacki // Sqrrl Data – Resource access mode: https://sqrrl.com/5-types-threat-hunting/.

24. An Enhanced Stacked LSTM Method With No Random Initialization for Malware Threat Hunting in Safety and Time-Critical Systems / [A. N. Jahromi, S. Hashemi, A. Dehghantanha et al.] // IEEE Transactions on Emerging Topics in Computational Intelligence. – 2020. – Vol. 4. No. 5 – P. 630 – 640, doi: 10.1109/TETCI.2019.2910243.

25. Long II M. C. Scalable Methods for Conducting Cyber Threat Hunt Operations [Electronic resource] / M. C. Long II // SANS Institute. – 2017. – 20 P. – Resource access mode: https://www.sans.org/reading-room/whitepapers/threats/scalable-methods-conducting-cyber-threat-hunt-operations-37090.

26. A Multi-Kernel and Meta-heuristic Feature Selection Approach for IoT Malware Threat Hunting in the Edge Layer / [H. Haddadpajouh, A. Mohtadi, A. Dehghantanaha et al.] // IEEE Internet of Things Journal. – 2021. – Vol. 8. No. 6 – P. 4540 – 4547, doi: 10.1109/JIOT.2020.3026660.

27. A cyber security data triage operation retrieval system / [C. Zhong, T. Lin, P. Liu et al.] // Computers & Security. – 2018. – Vol. 76. – P. 12 – 31, doi: 10.1016/j.cose.2018.02.011.

28. Caltagirone S. The Diamond Model of Intrusion Analysis [Electronic resource] / S. Caltagirone, A. Pendergast. – 2013. – 61 P. – Resource access mode: https://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf.

29. Samtani S. Developing Proactive Cyber Threat Intelligence from the Online Hacker Community: A Computational Design Science Approach: PhD Thesis / Samtani – Tucson, 2018. – 205 P.

30. The Sage Advice Guide to Cyber Threat Hunting [Electronic resource] // Tyler Cybersecurity. – 2020. – Resource access mode: https://www.tylercybersecurity.com/cyber-threat-hunting#tools.

31. Detecting the Unknown: A Guide to Threat Hunting [Electronic resource] // Home Office Digital, Data and Technology. – Version 2.0. – 2019. – 50 P. –Resource access mode: https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf.

32. Huntpedia: Your Threat Hunting Knowledge Compendium [Electronic resource] / [D. Akacki, D. Bianco, R. Bejtlich et al.] // Sqrrl Data. – 2018. – 107 P. – Resource access mode: https://www.threathunting.net/files/huntpedia.pdf.

33. Bornholm B. Network-based APT profiler: Thesis / Bornholm – Rochester Institute of Technology, 2019. – 189 P.

34. Delgado P. Developing an Adaptive Threat Hunting Solution: The Elasticsearch Stack: Thesis / – College of Information and Logistics Technology, 2018. – 120 P.

35. Liliengren T. Threat hunting, definition and framework: Thesis / T. Liliengren, P. Löwenadler // – School of Information Technology, Halmstad University, 2018. – 81 P.

36. Velazquez C. Detecting and Preventing Attacks Earlier in the Kill Chain [Electronic resource] / C. Velazquez // SANS Institute. – 2015. – 22 P. – Resource access mode: https://www.sans.org/reading-room/whitepapers/infosec/detecting-preventing-attacks-earlier-kill-chain-36230.

37. Nese A. Improving Security Posture by Learning from Intrusions: Thesis / Nese – Norwegian University of Science and Technology, 2018. – 144 P.

38. Pure I. An automated methodology for validating web related cyber threat intelligence by implementing a honeyclient: Thesis / Pure – Institute of Computer Science, University of Tartu, 2019. – 64 P.

39. Roberts S. J. Intelligence-Driven Incident Response: Outwitting the Adversary / S. J. Roberts, R. Brown. // O'Reilly Media, Inc. – 2017. – 420 P.

40. White paper: A Framework for Cyber Threat Hunting [Electronic resource] // Sqrrl Data. – 2018. – 10 P. – Resource access mode: https://www.threathunting.net/files/framework-for-threat-hunting-whitepaper.pdf.

41. Horta Neto A. J. Cyber Threat Hunting Through Automated Hypothesis and Multi-Criteria Decision Making / A. J. Horta Neto, A. Fernandes Pereira dos Santos. // 2020 IEEE International Conference on Big Data (Big Data). – 2020. – P. 1823 – 1830, doi: 10.1109/BigData50022.2020.9378213.

42. Wei J. A laboratory for hands-on cyber threat hunting education / J. Wei, B.-T. Chu, D. Cranford-Wesley, J. Brown. // Journal of The Colloquium for Information Systems Security Education. –2020. – Vol. 7. No. 1. – P. 1–7.

43. Mavroeidis V. Cyber threat intelligence model: An evaluation of taxonomies sharing standards and ontologies within cyber threat intelligence / V. Mavroeidis, S. Bromander. // 2017 European Intelligence and Security Informatics Conference (EISIC). – 2017. – P. 91–98.

44. Desmeules R. E. Exploring Methods for Linked Data Model Evaluation in Practice / R. E. Desmeules, C. Turp, A. Senior. // Journal of Library Metadata. – 2020. – P. 65–89.

45. Janjua F. Handling Insider Threat Through Supervised Machine Learning Techniques / F. Janjua, A. Masood, H. Abbas, I. Rashid. // Procedia Computer Science. – 2020. – Vol. 177. – P. 64–71.

46. Noor U. A machine learning-based fintech cyber threat attribution framework using high-level indicators of compromise / U. Noor, Z. Anwar, T. Amjad, K.-K. R. Choo. // Future Generation Computer Systems. –2019. – Vol. 96. – P. 227–242.

47. Brown S. From cyber security information sharing to threat management / S. Brown, J. Gommers, O. Serrano. // Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security. –2015. – P. 43–49.

48. Rasheed H. Threat Hunting Using GRR Rapid Response / H. Rasheed, A. Hadi, M. Khader. // 2017 International Conference on New Trends in Computing Sciences (ICTCS). – 2017. – P. 155–160, doi: 10.1109/ICTCS.2017.22.

49. Cole E. Automating the Hunt for Hidden Threats [Electronic resource] / E. Cole // SANS Institute. – 2016. – Resource access mode: https://www.sans.org/reading-room/whitepapers/analyst/membership/36282.

50. Poputa-Clean P. Automated Defense - Using Threat Intelligence to Augment [Electronic resource] / P. Poputa-Clean // SANS Institute. – 2015. – P. 38 – Resource access mode: https://www.sans.org/reading-room/whitepapers/threats/automated-defense-threat-intelligence-augment-35692.

51. Reading the tea leaves: A comparative analysis of threat intelligence / [C V. G. Li, M. Dunn, P. Pearce et al.] // Proceedings of the 28th USENIX Conference on Security Symposium. – 2019. – P. 851 – 867.

52. Samtani S. Cybersecurity as an industry: A cyber threat intelligence perspective / S. Samtani, M. Abate, V. Benjamin, W. Li. // The Palgrave Handbook of International Cybercrime and Cyberdeviance. – 2020. – P. 135–154.

53. Hallberg J. Event-driven Analysis of Cyber Kill Chain: Thesis / Hallberg – JAMK University of Applied Sciences, 2020 – 111 P.

54. Scarfone K. The Hunter’s Handbook: Endgame’s Guide to Adversary Hunting. // CyberEdge. – 2016. – 61 P.

55. Miloslavskaya N. Security Operations Centers for Information Security Incident Management / N. Miloslavskaya. // 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). – 2016. – P. 131–136, doi: 10.1109/FiCloud.2016.26.

56. Overview of Azure Cloud Services (classic) [Electronic resource] // Microsoft. – 2020. – Resource access mode: https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-choose-me.

57. Microsoft Defender for Endpoint [Electronic resource] // Microsoft. – 2021. – Resource access mode: https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-endpoint?view=o365-worldwide.

58. Getting started with Kusto [Electronic resource] // Microsoft. – 2020. – Resource access mode: https://docs.microsoft.com/en-us/azure/data-explorer/kusto/concepts/.

59. CrowdStrike Global Threat Report: adversary tradecraft and the importance of speed [Electronic resource] // CrowdStrike. – 2019. – 75 P. – Resource access mode: https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2019GlobalThreatReport.pdf.

60. CrowdStrike Global Threat Report [Electronic resource] // CrowdStrike. – 2020. – 68 P. – Resource access mode: https://afrasrv.ir/wp-content/uploads/2020/06/GLOBAL-THREAT-REPORT-2020.pdf.

61. CrowdStrike Global Threat Report [Electronic resource] // CrowdStrike. – 2021. – 53 P. – Resource access mode: https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2021GTR.pdf.

62. Red Canary Threat Detection Report [Electronic resource] // Red Canary. – 2019. – 31 P. – Resource access mode: https://resource.redcanary.com/rs/003-YRU-314/images/ThreatDetectionReport-2019.pdf.

63. Red Canary Threat Detection Report [Electronic resource] // Red Canary. – 2020. – 93 P. – Resource access mode: https://resource.redcanary.com/rs/003-YRU-314/images/2020-Threat-Detection-Report.pdf.

64. Red Canary Threat Detection Report [Electronic resource] // Red Canary. – 2021. – 122 P. – Resource access mode: https://resource.redcanary.com/rs/003-YRU-314/images/2021-Threat-Detection-Report.pdf?mkt_tok=MDAzLVlSVS0zMTQAAAF78_PiecrvPxUua_xUfr6YoG1T6n8AXnP9rKnbEPTenTo8AE5rvONj5FcsXE7MZ-PtpvfYRB-RMy-KX7FG9K1fSJsGGiAMQUrDWL0D2HIidg.

65. Recorded Future: Defense Evasion Dominant in Top MITRE ATT&CK Tactics of 2019 [Electronic resource] // Recorded Future. – 2019. – 19 P. – Resource access mode: https://www.recordedfuture.com/mitre-attack-tactics/.

66. Recorded Future: Top MITRE ATT&CK Techniques Identified in 2020, Defense Evasion Tactics Prevail [Electronic resource] // Recorded Future. – 2021. – 11 P. –

Resource access mode: https://go.recordedfuture.com/hubfs/reports/cta-2021-0203.pdf?hsCtaTracking=caa7e5e6-ac59-4f94-8c81-3da49fae3fa4%7Cfa1f5f1f-3a62-4624-95e6-1f4c80bd1a7f.

67. Rapid7 Threat Report [Electronic resource] // Rapid7. – 2019. – 25 P. – Resource access mode: https://www.rapid7.com/research/report/2019-q3-threat-report/.

68. Rapid7 Threat Report [Electronic resource] // Rapid7. – 2020. – 34 P. – Resource access mode: https://www.rapid7.com/research/report/2020-threat-report/.

69. Rapid7 Threat Report [Electronic resource] // Rapid7. – 2021. – 37 P. – Resource access mode: https://www.rapid7.com/research/report/2020Q2-threat-report/.

70. MITRE ATT&CK Framework [Electronic resource] // MITRE Corporation. – 2021. – Resource access mode: https://attack.mitre.org/tactics/enterprise/.

71. Enterprise APT3 Overview [Electronic resource] // MITRE Corporation. – 2018. – Resource access mode: https://attackevals.mitre-engenuity.org/enterprise/APT3/.

72. Enterprise APT29 Overview [Electronic resource] // MITRE Corporation. – 2019. – Resource access mode: https://attackevals.mitre-engenuity.org/enterprise/APT29/.

73. Early Detection of the Advanced Persistent Threat Attack Using Performance Analysis of Deep Learning / [J. Hassannataj Joloudari, M. Haderbadi, A. Mashmool et al.] // IEEE Access. – 2020. – Vol. 8. – P. 186125 – 186137, doi: 10.1109/ACCESS.2020.3029202.

74. Finding Cyber Threats with ATT&CK™-Based Analytics [Electronic resource] / [B. E. Strom, J. A. Battaglia, M. S. Kemmerer et al.] // MITRE Corporation. – 2017. – Resource access mode: https://www.mitre.org/sites/default/files/publications/16-3713-finding-cyber-threats%20with%20att%26ck-based-analytics.pdf.

75. Rünkaru H. Endpoint Detection and Response Solution as a Security as a Service Platform by the Example of FireEye HX: Thesis / Rünkaru – Tallin University of Technology, 2020 – 60 P.

76. MITRE ATT&CK Arsenal [Electronic resource] // MITRE Corporation. – 2020. – Resource access mode: https://github.com/mitre-attack/attack-arsenal.

77. Red Canary Atomic Red Team [Electronic resource] // Red Canary. – 2020. – Resource access mode: https://github.com/redcanaryco/atomic-red-team.

78. Falcon Force Team Falcon Friday [Electronic resource] // Falcon Force. – 2021. – Resource access mode: https://github.com/FalconForceTeam/FalconFriday.

79. MITRE ATT&CK Enterprise Techniques [Electronic resource] // MITRE Corporation. – 2021. – Resource access mode: https://attack.mitre.org/techniques/.

80. Living Off The Land Binaries and Scripts (and now also Libraries) [Electronic resource] // LOLBAS Project. – 2021. – Resource access mode: https://github.com/LOLBAS-Project/LOLBAS.

81. Demmer S. Detecting „Living-off-the-Land" Attacker Techniques in Microsoft Windows Using publicly available technology to spot modern adversaries: Thesis / Demmer – St. Pölten University of Applied Sciences, 2019 – 50 P.

82. New Targeted Attack in the Middle East by APT34, a Suspected Iranian Threat Group, Using CVE-2017-11882 Exploit [Electronic resource] / [M. Sardiwal, V. Cannon, N. Fraser et al.] // FireEye Threat Research. – 2017. – Resource access mode: https://www.fireeye.com/blog/threat-research/2017/12/targeted-attack-in-middle-east-by-apt34.html.

83. "Windows PowerShell Logging Cheat Sheet [Electronic resource] // Malware Archeology. – 2016. – Resource access mode: https://static1.squarespace.com/static/552092d5e4b0661088167e5c/t/5760096ecf80a129e0b17634/1465911664070/Windows+PowerShell+Logging+Cheat+Sheet+ver+June+2016+v2.pdf.

84. Hosseini A. Ten process injection techniques: A technical survey of common and trending process injection techniques [Electronic resource] / A. Hosseini // Elastic. – 2017. – Resource access mode: https://www.elastic.co/blog/ten-process-injection-techniques-technical-survey-common-and-trending-process.

85. Desimone J. Hunting In Memory [Electronic resource] / J. Desimone // Elastic. – 2019. – Resource access mode: https://www.elastic.co/blog/hunting-memory.

86. Sutherland S. 15 Ways to Bypass the PowerShell Execution Policy [Electronic resource] / S. Sutherland // Netspi. – 2014. – Resource access mode: https://www.netspi.com/blog/technical/network-penetration-testing/15-ways-to-bypass-the-powershell-execution-policy/.

87. Babinec K. Executing PowerShell scripts from C# [Electronic resource] / K. Babinec // Microsoft. – 2014. – Resource access mode: https://docs.microsoft.com/ru-ru/archive/blogs/kebab/executing-powershell-scripts-from-c.

88. Russinovich M. Autoruns for Windows v13.100 [Electronic resource] / M. Russinovich // Microsoft. – 2021. – Resource access mode: https://docs.microsoft.com/ru-ru/sysinternals/downloads/autoruns.

89. Moe O. Persistence using RunOnceEx - Hidden from Autoruns.exe. [Electronic resource] / 2018. – Resource access mode: https://oddvar.moe/2018/03/21/persistence-using-runonceex-hidden-from-autoruns-exe/.

90. Sharma N. Net Commands On Operating Systems [Electronic resource] / N. Sharma // Microsoft. – 2020. – Resource access mode: https://docs.microsoft.com/en-US/troubleshoot/windows-server/networking/net-commands-on-operating-systems.

91. Hidden Administrative Accounts: BloodHound to the Rescue [Electronic resource] // CrowdStrike. – 2018. – Resource access mode: https://www.crowdstrike.com/blog/hidden-administrative-accounts-bloodhound-to-the-rescue/.

92. French D. Detecting Attempts to Steal Passwords from Memory [Electronic resource] / D. French // Threat Punter. – 2018. – Resource access mode: https://medium.com/threatpunter/detecting-attempts-to-steal-passwords-from-memory-558f16dce4ea.

93. [MS-SAMR]: Security Account Manager (SAM) Remote Protocol (Client-to-Server) [Electronic resource] // Microsoft. – 2021. – Resource access mode: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-samr/4df07fab-1bbc-452f-8e92-7853a3c7e380.

94. Bohannon D. Obfuscation in the Wild: Targeted Attackers Lead the Way in Evasion Techniques [Electronic resource] / D. Bohannon, N. Carr // FireEye Threat Research. – 2017. – Resource access mode: https://www.fireeye.com/blog/threat-research/2017/06/obfuscation-in-the-wild.html.

95. Bohannon D. Revoke-Obfuscation: PowerShell Obfuscation Detection Using Science [Electronic resource] / D. Bohannon, L. Holmes // FireEye Threat Research. –

2017. – 20 P. – Resource access mode: https://www.fireeye.com/content/dam/fireeye-www/blog/pdfs/revoke-obfuscation-report.pdf.

96. Ancel B. Poweliks – Command Line Confusion [Electronic resource] / B. Ancel // Thisissecurity. – 2014. – Resource access mode: https://thisissecurity.stormshield.com/2014/08/20/poweliks-command-line-confusion/.

97. Mercês F. CPL Malware Malicious Control Panel Items [Electronic resource] / F. Mercês // Trend Micro Research Paper. – 2014. – 18 P. – Resource access mode: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-cpl-malware.pdf.

98. Carvey H. Where You AT?: Indicators of Lateral Movement Using at.exe on Windows 7 Systems [Electronic resource] / H. Carvey // Secureworks. – 2014. – Resource access mode: https://www.secureworks.com/blog/where-you-at-indicators-of-lateral-movement-using-at-exe-on-windows-7-systems.

99. Audit Other Object Access Events [Electronic resource] // Microsoft. – 2017. – Resource access mode: https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-other-object-access-events.

100. MCMD Malware Analysis [Electronic resource] // Secureworks. – 2019. – Resource access mode: https://www.secureworks.com/research/mcmd-malware-analysis.

101. Nesbit B. Malware Analysis Report RawPOS Malware: Deconstructing an Intruder’s Toolkit [Electronic resource] / B. Nesbit, D. Ackerman // Kroll. – 2017. – 44 P. – Resource access mode: https://www.kroll.com/-/media/kroll/pdfs/publications/kroll-malware-analysis-report-rawpos-malware-deconstructing-an-intruders-toolkit.pdf.

102. Implementing Least-Privilege Administrative Models [Electronic resource] // Microsoft. – 2018. – Resource access mode: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models.

103. Miller S. TRITON Actor TTP Profile, Custom Attack Tools, Detections, and ATT&CK Mapping [Electronic resource] / S. Miller, N. Brubaker, D. Kapellmann Zafra, D. Caban // FireEye Threat Research. – 2019. – Resource access mode:

https://www.fireeye.com/blog/threat-research/2019/04/triton-actor-ttp-profile-custom-attack-tools-detections.html.

104. Microsoft Security Intelligence Report [Electronic resource] / [C. Anthe, E. Argyle, E. Douglas et al.] // Microsoft. – 2016. – 180 P. – Resource access mode: http://download.microsoft.com/download/E/B/0/EB0F50CC-989C-4B66-B7F6-68CD3DC90DE3/Microsoft_Security_Intelligence_Report_Volume_21_English.pdf.

105. Ransomware Uncovered: Attackers' Latest Methods [Electronic resource] // GroupIB. – 2019. – 24 P. – Resource access mode: https://www.group-ib.com/whitepapers/ransomware-uncovered.html.

106. Murphy B. The Elastic Guide to Threat Hunting [Electronic resource] / B. Murphy, D. French // CyberEdge Group, LLC. – 2020. – 74 P. – Resource access mode: https://www.elastic.co/pdf/elastic-guide-to-threat-hunting.

107. Ballenthin W. Windows Management Instrumentation (WMI) Offense, Defense, and Forensics [Electronic resource] / W. Ballenthin, M. Graeber, C. Teodorescu // FireEye, Inc. – 2015. – 90 P. – Resource access mode: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-windows-management-instrumentation.pdf.

108. Reverse engineering DUBNIUM – Stage 2 payload analysis [Electronic resource] // Microsoft. – 2016. – Resource access mode: https://www.microsoft.com/security/blog/2016/07/14/reverse-engineering-dubnium-stage-2-payload-analysis/.

109. Technical Analysis of Operation Diànxùn [Electronic resource] // McAfee. – 2021. – 24 P. – Resource access mode: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-operation-dianxun.pdf.

110. Endpoint Protection: Measuring the Effectiveness of Remediation Technologies and Methodologies for Insider Threat / [S. Chandel, S. Yu, T. Yitian et al.] // 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). – 2019. – P. 81 – 89, doi: 10.1109/CyberC.2019.00023.

111. Статистика: учебник для бакалавров / Н. А. Садовникова [и др.]; под ред. В. Г. Минашкина. – М.: Издательство Юрайт, 2014. – 448 с. – Серия: Бакалавр. Базовый курс.

112. Теория вероятностей: в 3 ч. Ч. 2. Одномерные случайные величины. Предельные теоремы: учеб.-метод. пособие / авт.-сост.: С. Е. Демин, Е. Л. Демина; – НТИ (филиал) УрФУ, 2017. – 286 с.

ДОДАТОК А

Перелік наукових публікацій

1. H. Papirna. Threat Hunting as a method of protection against cyber threats / N. Lukova-Chuiko, A. Fesenko, H. Papirna, S. Gnatyuk // CEUR Selected Papers of the 7th International Conference "Information Technology and Interactions" (IT&I-2020). Conference Proceedings, Kyiv, Ukraine, December 02-03, 2020, Vol. 2833, pp. 103-113.

2. H. Papirna. Modern approaches to the security evaluation: a roadmap to secure and usable systems/ A. Fesenko, H. Papirna// Scientific and Practical Cyber Security Journal (SPCSJ) 2(2):13-17 Scientific Cyber Security Association (SCSA), 2018, pp. 13-17.

3. H. Papirna. Threat Hunting as a method of protection against cyber threats / N. Lukova-Chuiko, A. Fesenko, H. Papirna, S. Gnatyuk // Information Technology and Interactions (Satellite): Conference Proceedings, December 04, 2020, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). Kyiv: Stylos, 2020. 388 p. P. 79-82.

4. Г. К. Папірна. Виявлення фінансових шахрайств засобами машинного навчання / А.О. Фесенко, Г. К. Папірна, М. Бауиржан // Фаховий науково-технічний журнал «Захист інформації». – 2019, Том 21, №2, с. 104-111, doi: 10.18372/2410-7840.21.13768.

5. Г.К. Папірна. Машинне навчання як інструмент виявлення інтернет-шахрайств / А.О. Фесенко, Г.К. Папірна // IT&I:V Міжнародна наук.-практ. конф. 20-21 листопада 2018р.: тези доп. – К.: Видавнично-поліграфічний центр «Київський університет», 2018. – 320 с. С.330-331.

6. Г.К. Папірна. Концепція частково розподілених файлових систем як напрям забезпечення безпеки даних у хмарному середовищі / А.О. Фесенко, Г.К. Папірна // Матеріали науково-практичної конференції «Сучасні інформаційні технології та кібербезпека». – К.: ІСЗЗІ КПІ ім. Ігоря Сікорського, 2018. – 264 с. С. 43-46.

7. Г.К. Папірна. Проблема співвідношення ефективності та економічної вигідності при виборі засобів захисту інформаційних активів /А. О. Фесенко, Г. К. Папірна, Я. В. Шестак// IT&I:IV Міжнародна наук.-практ. конф. 8-10 листопада 2016р.: тези доп.- К.:Вид-во Виданично-поліграфічний центр «Київський університет», 2017. – 310 с. С.284-286.

8. Г.К. Папірна. Сучасні підходи до оцінки захищеності систем/ Г.К. Папірна, А.О. Фесенко // Проблеми кібербезпеки інформаційно-телекомунікаційних систем: Збірник матеріалів доповідей та тез; м. Київ, 05-06 квітня 2018 року; Київський національний університет імені Тараса Шевченка / Редкол.: Оксіюк О.Г. (голова) та ін. – К.: ВПЦ «Київський університет», 2018. – 510 с. С. 155-159.

9. Фесенко А.А. Аутентификация по радужной оболочке глаза/ А.А.Фесенко, А.Г.Оксиюк, Швец В.А., Папирная А.К.// Современные средства связи: материалы XXIII Междунар. Науч.-техн. Конф., 18-19 окт. 2018 года, Минск, Респ. Беларусь; редкол.: А.О.Зеневич [и др.]. - Минск: Белорусская государственная академия связи, 2018.-304 с. С.226-228.

ДОДАТОК Б

Лістинг запитів ППЗ

Техніка: Маскування.

```
// Техніка: Маскування. Папірна Г.К.
// Проведення запиту на основі даних за тижневий період
let timeframe=7d;
// Перелік LOLBIN, які найчастіше використовуються в системі Windows
let lolbins = dynamic(["at.exe", "atbroker.exe", "bash.exe", "bitsadmin.exe", "certreq.exe", "certutil.exe", "cmd.exe", "cmdkey.exe", "cmstp.exe", "control.exe",
"csc.exe", "cscript.exe", "desktopimgdownldr.exe", "dfsvc.exe", "diantz.exe", "diskshadow.exe", "dnscmd.exe", "esentutl.exe", "eventvwr.exe", "expand.exe",
"extexport.exe", "extrac32.exe", "findstr.exe", "forfiles.exe", "ftp.exe", "gfxdownloadwrapper.exe", "gpscript.exe", "hh.exe", "ie4uinit.exe", "ieexec.exe",
"ilasm.exe", "infdefaultinstall.exe", "installutil.exe", "jsc.exe", "makecab.exe", "mavinject.exe", "microsoft.workflow.compiler.exe", "mmc.exe", "mpcmdrun.exe",
"msbuild.exe", "msconfig.exe", "msdt.exe", "mshta.exe", "msiexec.exe", "netsh.exe", "odbcconf.exe", "pcalua.exe", "pcwrun.exe", "pktmon.exe",
"presentationhost.exe", "print.exe", "psr.exe", "rasautou.exe", "reg.exe", "regasm.exe", "regedit.exe", "regini.exe", "register-cimprovider.exe", "regsvcs.exe",
"regsvr32.exe", "replace.exe", "rpcping.exe", "rundll32.exe", "runonce.exe", "runscripthelper.exe", "sc.exe", "schtasks.exe", "scriptrunner.exe",
"syncappvpublishingserver.exe", "ttdinject.exe", "tttracer.exe", "vbc.exe", "verclsid.exe", "wab.exe", "wmic.exe", "wscript.exe", "wsreset.exe", "xwizard.exe",
"agentexecutor.exe", "appvlp.exe", "bginfo.exe", "cdb.exe", "csi.exe", "devtoolslauncher.exe", "dnx.exe", "dotnet.exe", "dxcap.exe", "excel.exe", "mftrace.exe",
"msdeploy.exe", "msxsl.exe", "ntdsutil.exe", "powerpnt.exe", "rcsi.exe", "sqldumper.exe", "sqlps.exe", "sqltoolsps.exe", "squirrel.exe", "te.exe", "tracker.exe",
"vsjitdebugger.exe", "winword.exe", "wsl.exe"]);
// Перелік відомих вихідних імен файлів, які можуть бути вбудовані в заголовок метаданих, які відрізняються від самого імені файлу LOLBIN в системі Windows
let binaries_of_interest = dynamic(["net.exe", "net1.exe", "whoami.exe", "ipconfig.exe", "tasklist.exe", "quser.exe", "tracert.exe", "route.exe", "runas.exe",
"klist.exe", "wevtutil.exe", "wmiprvse.exe", "powershell.exe", "bash.exe", "qwinsta.exe", "rwinsta.exe", "replace.exe", "findstr.exe", "icacls.exe", "cacls.exe",
"xcopy.exe", "robocopy.exe", "takeown.exe", "vssadmin.exe", "nltest.exe", "nltestk.exe", "sctasks.exe", "nbtstat.exe", "nbtinfo.exe", "mofcomp.exe", "nltestrk.exe",
"dnscmd.exe", "registercimprovider.exe", "registercimprovider2.exe", "procdump", "ru.exe", "pspasswd.exe", "psexec.c", "psexec.exe", "pslist.exe", "regsize",
"pskill.exe", "pkill.exe", "wsmprovhost.exe", "fltmc.exe", "sdbinst.exe"]);
// Об'єднання двох масивів в один список
let original_file_name_set=array_concat(lolbins,binaries_of_interest);
DeviceProcessEvents
| where Timestamp > ago(timeframe)
| extend process_name=tolower(FileName)
| extend original_file_name=tolower(ProcessVersionInfoOriginalFileName)
| where original_file_name in (original_file_name_set)
| where original_file_name != ""
// Фільтрація деяких відомих невідповідностей між заголовком метаданих імені файлу і самим іменем файлу
| where not(process_name=="schtasks.exe" and original_file_name=="sctasks.exe" and
(FolderPath=~@"C:\Windows\System32\schtasks.exe" or FolderPath=~@"C:\Windows\SysWOW64\schtasks.exe"))
| where not(process_name=="nbtstat.exe" and original_file_name=="nbtinfo.exe" and
FolderPath=~@"C:\Windows\System32\nbtstat.exe")
| where not(process_name=="bginfo64.exe" and original_file_name=="bginfo.exe" and
(FolderPath=~@"C:\Windows\System32\Bginfo64.exe" or FolderPath =~@"C:\Program Files\SysInternals BGInfo\Bginfo64.exe"))
// Фільтрація конвертеру форматів файлів MS Excel
| where not(process_name=="excelcnv.exe" and original_file_name=="excel.exe" and
(FolderPath startswith @"C:\Program Files\Microsoft Office Web Apps\ExcelServicesEcs\" or
FolderPath  startswith @"C:\Program Files\Microsoft Office\" or FolderPath startswith @"C:\Program Files (x86)\Microsoft Office\"))
// Опціональна фільтрація, у випадку якщо psexec дуже поширений у конкретному середовищі
//| where not(process_name=="psexec.exe" and original_file_name=="psexec.c")
//| where not(process_name=="psexec64.exe" and original_file_name=="psexec.c")
| where process_name != original_file_name
| project Timestamp,DeviceName,AccountName,process_name, original_file_name, FolderPath, ProcessCommandLine, InitiatingProcessFileName,
InitiatingProcessVersionInfoOriginalFileName, InitiatingProcessCommandLine, InitiatingProcessParentFileName, ReportId
```

Техніка: Інтерпретатор команд і скриптів.

```
// Техніка: Інтерпретатор команд і скриптів. Папірна Г.К.
// Проведення запиту на базі PowerShell та System.Management.Automation.dll
DeviceImageLoadEvents
| where FileName =~ "System.Management.Automation.dll" or FileName =~ "System.Management.Automation.ni.dll"
| where InitiatingProcessFolderPath !~ "C:\\Windows\\System32\\WindowsPowerShell\\v1.0" and InitiatingProcessFolderPath
!~ "C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0"  and (InitiatingProcessFileName !~ "powershell.exe" or InitiatingProcessFileName
!~ "powershell_ise.exe")
// RemoteFXvGPUDisablement.exe призначений для віртуалізації графічного процесора, MS рекомендує видалити цю службу
| where InitiatingProcessFolderPath !~ "C:\\Windows\\system32" and InitiatingProcessFileName !~ "RemoteFXvGPUDisablement.exe"
// Виключення зазначені нижче можуть бути включені, якщо використовується Visual Studio
//| where InitiatingProcessFolderPath !contains "C:\\Windows\\Microsoft.NET\\Framework" and InitiatingProcessFileName !~ "devenv.exe"
//| where InitiatingProcessFolderPath !contains "\\Microsoft Visual Studio\\2019\\Community\\Common7\\ServiceHub\\Hosts\\ServiceHub.Host.CLR.x86"
//and InitiatingProcessFileName !startswith "servicehub"
//| where InitiatingProcessFolderPath !contains "\\Microsoft Visual Studio\\2019\\Community\\Common7\\IDE" and InitiatingProcessFileName
//!~ "mscorsvw.exe" and InitiatingProcessParentFileName !~ "ngen.exe"
| project Timestamp,DeviceName,InitiatingProcessAccountName,ActionType,InitiatingProcessFileName,InitiatingProcessCommandLine,
InitiatingProcessIntegrityLevel,FileName,InitiatingProcessParentId,InitiatingProcessId
```

```
// Техніка: Ін’єкція процесу. Папірна Г.К.
// Проведення запиту на основі даних за добу
let timeframe = (24h);
let remoteAlloc = DeviceEvents
| where Timestamp > ago(timeframe)
// Пошук викликів NtAllocateVirtualMemoryRemote
| where ActionType == "NtAllocateVirtualMemoryRemoteApiCall"
| where InitiatingProcessFileName !~ FileName
// Пошук непідписаних процесів
| where not(isempty(InitiatingProcessSHA1))
| summarize count(), make_set(InitiatingProcessFileName) ,make_set(FileName) by InitiatingProcessSHA1
| invoke FileProfile(InitiatingProcessSHA1, 1000)
| where (GlobalPrevalence <= 200 or isempty(GlobalPrevalence)) and IsCertificateValid != 1;
DeviceEvents
| where Timestamp > ago(timeframe)
| where InitiatingProcessSHA1 in ((remoteAlloc | project InitiatingProcessSHA1)) and ActionType == "NtAllocateVirtualMemoryRemoteApiCall"
| join kind=leftouter remoteAlloc on InitiatingProcessSHA1
| summarize count() by FileName,InitiatingProcessFolderPath, InitiatingProcessSHA256,InitiatingProcessCommandLine, InitiatingProcessAccountName,
InitiatingProcessVersionInfoProductName, InitiatingProcessVersionInfoOriginalFileName
```

Техніка: Ін’єкція процесу.

```
// Техніка: Виконання сценаріїв. Папірна Г.К.
let suspiciousCRT = DeviceEvents
// Фільтрація всіх зареєстрованих використань API CreateRemoteThread, в яких процес впровадження не дорівнює процесу, який впроваджується
| where ActionType =~ "CreateRemoteThreadApiCall" and ProcessId != InitiatingProcessId
// Фільтрація унікальних впроваджень на базі хешу SHA1 і збагачення даних за допомогою функції FileProfile
| summarize by InitiatingProcessSHA1
| invoke FileProfile(InitiatingProcessSHA1, 1000)
// Список всіх унікальних хешів SHA1 підсумовується, де Global Prevalence нижче стандартно визначеного порогу
| where GlobalPrevalence < 200 or ((isempty(Signer) or not(IsCertificateValid)) and GlobalPrevalence < 500)
| summarize by InitiatingProcessSHA1;
// Фільтрація всіх подій журналу CreateRemoteThread, де інжектор був в раніше створеному списку підозрілих Global Prevalence
DeviceEvents
| where ActionType =~ "CreateRemoteThreadApiCall" and ProcessId != InitiatingProcessId and InitiatingProcessSHA1 in (suspiciousCRT)
```

Техніка: Виконання сценаріїв.

Техніка: Виконання автозапуску.

```
// Техніка: Виконання автозапуску. Папірна Г.К.
// Пошук здійснюється за подіями в реєстрі Windows
RegistryEvents
// Відслідковування додавання ключів до елементів реєстру
| where InitiatingProcessFileName == "powershell.exe"
| project EventTime, InitiatingProcessId, InitiatingProcessFileName, RegistryValueName, RegistryValueData
// Виявлення паралельного спрацьовування дій із боку Antimalware Scan Interface (AMSI) Windows
MiscEvents
| where ActionType == "AmsiScriptContent"
| where AdditionalFields contains "Webcache"
| project EventTime, InitiatingProcessId, InitiatingProcessFileName, AdditionalFields
MiscEvents
| where ActionType == "AmsiScriptContent"
| where AdditionalFields contains "IEX"
| project EventTime, InitiatingProcessId, InitiatingProcessFileName, AdditionalFields
```

Техніка: Виявлення облікових записів.

```
// Техніка: Виявлення облікових записів. Папірна Г.К.
// Відслідковування команд перерахування облікових записів утилітою Net Windows
DeviceProcessEvents
| where ActionType = "NetUtilityCommand"
| where AdditionalFields contains "Get-NetUser"
| project Timestamp, DeviceName, InitiatingProcessId, InitiatingProcessFileName, InitiatingProcessCommandLine, AdditionalFields
// Виявлення паралельного спрацьовування дій із боку Antimalware Scan Interface (AMSI) Windows
| where ActionType = "AmsiScriptContent"
| extend content = tostring(parse_json(AdditionalFields).ScriptContent)
| where content contains "Get-NetUser"
| project Timestamp, DeviceName, InitiatingProcessId, InitiatingProcessFileName, InitiatingProcessCommandLine, content
```

```
// Техніка: Дампінг облікових даних. Папірна Г.К.
// Перевірка дампінгу пам'яті процесу LSASS
DeviceProcessEvents
| where Timestamp > ago(7d)
// Command lines that include "lsass" and -accepteula or -ma flags used in procdump
| where (ProcessCommandLine has "lsass" and (ProcessCommandLine has "-accepteula" or
ProcessCommandLine contains "-ma"))
// Виключення можливих помилкових спрацьовувань
or (FileName in~ ('procdump.exe','procdump64.exe') and ProcessCommandLine has 'lsass')
// Пошук ознак крадіжки облікових даних через експорт бази даних SAM
DeviceProcessEvents
| where Timestamp > ago(7d)
| where FileName =~ 'reg.exe'
and ProcessCommandLine has 'save'
and ProcessCommandLine has 'hklm'
and ProcessCommandLine has 'sam'
| project DeviceId, Timestamp, InitiatingProcessId, InitiatingProcessFileName, ProcessId, FileName, ProcessCommandLine
```

Техніка: Дампінг облікових даних.

```
// Техніка: Обфускація файлів або інформації. Папірна Г.К.
// Пошук невідповідності у батьківсько-дочірніх відносинах основних процесів операційної системи,
// щоб виявити різні спроби маскування кодованих або шифрованих файлів
let ProcessRelations=datatable(ImageFile:string,ExpectedParent:dynamic) [
"smss.exe", dynamic(["smss.exe", "ntoskrnl.exe", ""]),
"crmss.exe", dynamic(["smss.exe"]),
"wininit.exe", dynamic(["smss.exe"]),
"winlogon.exe", dynamic(["smss.exe"]),
"services.exe", dynamic(["wininit.exe"]),
"lsaiso.exe", dynamic(["wininit.exe"]),
"lsass.exe", dynamic(["wininit.exe"]),
"spoolsv.exe", dynamic(["services.exe"]),
"dllhost.exe", dynamic(["svchost.exe"]),
"lsm.exe", dynamic(["wininit.exe"]),
"svchost.exe", dynamic(["services.exe", "msmpeng.exe"]),
"wscript.exe", dynamic(["vbscript.exe"]),
"runtimebroker.exe", dynamic(["svchost.exe"]),
"taskhostw.exe", dynamic(["svchost.exe"]),
"userinit.exe", dynamic(["winlogon.exe"])];
// Пошук шифрованих або кодованих файлів у процесах та зображеннях
DeviceProcessEvents
| extend ImageFile = tostring(tolower(parse_path(tostring(FolderPath)).Filename))
| extend ParentFile = tostring(tolower(parse_path(tostring(InitiatingProcessFolderPath)).Filename))
| project Timestamp, ImageFile, ParentFile
| lookup kind=inner ProcessRelations on ImageFile
| where not(set_has_element(ExpectedParent,ParentFile))
| summarize count() by ImageFile, ParentFile
```

Техніка: Обфускація файлів або інформації.

Техніка: Проксі-виконання коду через підписані бінарні файли.

```
// Техніка: Проксі-виконання коду через підписані бінарні файли. Папірна Г.К.
// Складання переліку всіх окремих хешів завантажених файлів CPL
let suspiciousCPLs = DeviceImageLoadEvents
| where FileName endswith ".cpl"
| summarize by SHA1
// Для кожного файлу перевіряється база даних Microsoft Threat Intelligence
| invoke FileProfile(SHA1, 1000)
// Далі встановлюються деякі пороги для підписаних і непідписаних CPL. Якщо розмір файлу нижче порогового значення, спрацьовує попередження
| where ((isempty(Signer) or not(IsCertificateValid)) and GlobalPrevalence < 100) or GlobalPrevalence < 50;
DeviceImageLoadEvents
| where SHA1 has_any (suspiciousCPLs) and ActionType == "ImageLoaded"
```

```
// Техніка: Заплановане завдання. Папірна Г.К.
// Отримання унікальних хешів бінарних файлів, які виконуються батьківським процесом svchost і аргументами командного рядка netsvcs
let scheduled_binaries = DeviceProcessEvents
| where InitiatingProcessCommandLine == "svchost.exe -k netsvcs -p -s Schedule"
| distinct SHA1;
let untrusted_binaries = scheduled_binaries
| join kind=leftanti (DeviceFileCertificateInfo | summarize max_trusted=max(IsTrusted) by SHA1 | where max_trusted==1) on SHA1;
untrusted_binaries
| invoke FileProfile(SHA1,1000)
// Збереження тільки непідписаних виконавчих файлів з низькою глобальною поширеністю
| where IsCertificateValid != 1
| where GlobalPrevalence < 1000
| join (DeviceProcessEvents | where InitiatingProcessCommandLine == "svchost.exe -k netsvcs -p -s Schedule") on SHA1
// Надання результату проранжованого за деталями SHA1
| summarize arg_max(Timestamp, *) by SHA1
```

Техніка: Заплановане завдання.

Техніка: Дані з локальної системи.

```
// Техніка: Дані з локальної системи. Папірна Г.К.
// Виявлення виконання незвичних для середовища командлетів PowerShell
let powershellCommands =
DeviceEvents
| where ActionType == "PowerShellCommand"
// Витягнення імені команди powershell з поля Command в стовпці AdditionalFields JSON
| project PowershellCommand=extractjson("$.Command", AdditionalFields, typeof(string)), InitiatingProcessCommandLine,
InitiatingProcessParentFileName, Timestamp, DeviceId
| where PowershellCommand !endswith ".ps1" and PowershellCommand !endswith ".exe";
// Фільтрація командлетів Powershell, що виконуються на відповідному комп'ютері і в певний період часу
powershellCommands | where DeviceId == DeviceId and Timestamp between ((timestamp-5min) .. 10min)
// Filter out common powershell cmdlets
| join kind=leftanti (powershellCommands | summarize MachineCount=dcount(DeviceId) by PowershellCommand | where MachineCount > 20) on PowershellCommand
// Виявлення планомірного завантаження файлів до деякого розположення у файловій системі за 445 локальним портом
let ToleranceInSeconds = 5;
DeviceNetworkEvents
| where LocalPort == 445 and isnotempty(RemoteIP)
| join kind = inner DeviceLogonEvents on DeviceId
| where Timestamp1 between (Timestamp .. datetime_add('second',ToleranceInSeconds,Timestamp)) and RemoteIP endswith RemoteIP1
| join kind=inner (
DeviceFileEvents
| where ActionType in ('FileModified','FileCreated') and (InitiatingProcessFileName =~ 'System' or InitiatingProcessFolderPath endswith "ntoskrnl.exe")
) on DeviceId
| where Timestamp2 between (Timestamp .. datetime_add('second',ToleranceInSeconds,Timestamp))
| join kind=inner DeviceProcessEvents on DeviceId, FolderPath
| where Timestamp3 between (Timestamp .. datetime_add('second',ToleranceInSeconds,Timestamp))
| project Timestamp, DeviceName, RemoteIP, RemotePort, AccountDomain, AccountName, AccountSid, Protocol, LogonId, RemoteDeviceName, IsLocalAdmin, FileName,
FolderPath, SHA1, SHA256, MD5, ProcessCommandLine
```

Техніка: Діючі облікові записи.

```
// Техніка: Діючі облікові записи. Папірна Г.К.
// Визначення підозрілого перерахування користувачів у системі за десять днів
let startdate = 10d;
let lookupwindow = 2m;
let threshold = 3; //number of commandlines in the set below
// Перелік імен серверів контролерів домену залежить від конфігурації конкретної системи облікових даних
let DCADFSServersList = dynamic (["DCServer10", "DCServer11", "ADFSServer13"]);
let tokens = dynamic(["objectcategory","domainlist","dcmodes","adinfo","trustdmp","computers_pwdnotreqd",
"Domain Admins", "objectcategory=person", "objectcategory=computer", "objectcategory=*"]);
DeviceProcessEvents
| where Timestamp between (ago(startdate) .. now())
| where ProcessCommandLine  has_any (tokens)
| where ProcessCommandLine matches regex "(.*)>(.*)"
| summarize Commandlines = make_set(ProcessCommandLine), LastObserved=max(Timestamp) by bin(Timestamp, lookupwindow),
AccountName, DeviceName, InitiatingProcessFileName, FileName
| extend Count = array_length(Commandlines)
| where Count > threshold
```

```
// Відслідковування додавання нового типу облікових даних до системних програм, у тому числі утиліт
CloudAppEvents
| where Application == "net.exe"
| where ActionType in ("Add service principal credentials.", "Update application – Certificates and secrets management ")
| project Timestamp, RawEventData, AccountDisplayName, ActionType, AccountObjectId
| extend ModifiedProperties = RawEventData.ModifiedProperties
| extend NewValue = ModifiedProperties.NewValue, OldValue = ModifiedProperties.OldValue, Name = ModifiedProperties.Name
| project Timestamp, AccountDisplayName, ActionType, NewValue, OldValue, RawEventData, AccountObjectId
| where (NewValue has "KeyType=Password" and OldValue !has "KeyType=Password" and OldValue has "AsymmetricX509Cert") or
(NewValue has "AsymmetricX509Cert" and OldValue !has "AsymmetricX509Cert" and OldValue has "KeyType=Password")
| extend NewSecret = set_difference(todynamic(parse_json(tostring(NewValue))), todynamic(parse_json(tostring(OldValue))))
| project Timestamp,ActionType,ActorType = RawEventData.Actor[-1].ID, ObjectId = RawEventData.Actor[-2].ID, AccountDisplayName,
AccountObjectId, AppnName = RawEventData.Target[3].ID, AppObjectId = RawEventData.Target[1].ID, NewSecret = NewSecret[0], RawEventData
// Відслідковування нетипових входів користувачів в систему, тобто таких, що виходять за межі звичайного робочого процесу
let relevant_computers=
DeviceInfo
| where MachineGroup == "MachineGroup"
| summarize make_list(DeviceName);
let relevant_users=
IdentityInfo
| where EmailAddress endswith "@allowed.users"
| summarize make_list(AccountName);
DeviceLogonEvents
| where Timestamp > ago(1d)
| where DeviceName in (relevant_computers)
| where AccountName !in (relevant_users)
| project DeviceName, AccountName
```

```
// Техніка: Виявлення системної інформації. Папірна Г.К.
// Моніторинг системної утиліти systeminfo.exe на предмет несподіваного її запуску сторонніми виконавчими файлами
let systeminfo = pack_array;
union DeviceProcessEvents , DeviceFileEvents
| where Timestamp > ago(7d)
| where FileName =~ "nbtscan.exe" or SHA1 in (systeminfo)
| project FolderPath, FileName, InitiatingProcessAccountName,
InitiatingProcessFileName, ProcessCommandLine, Timestamp
// Відслідковування запитів на предмет пошуку об’єктів облікових даних та комп’ютерів
let ComputerObject = "objectCategory=computer";
let ComputerClass = "objectClass=computer";
let SamAccountComputer = "sAMAccountType=805306369";
let OperatingSystem = "operatingSystem=";
IdentityQueryEvents
| where ActionType == "LDAP query"
| parse Query with * "Search Scope: " SearchScope ", Base Object:" BaseObject ", Search Filter: " SearchFilter
| where (SearchFilter contains ComputerObject or SearchFilter contains ComputerClass or SearchFilter contains SamAccountComputer) and
SearchFilter contains OperatingSystem
// Пошук великих обсягів мережевих запитів за параметрами: IP-адреси, імені хоста, порта і процесу
// Діапазони підмереж та часові інтервали можна змінювати
let remotePortCountThreshold = 10;
// Необхідно підлаштувати мінімальне значення для хоста, що звертається до віддалених портів на віддалену IP-адресу до такого,
// яке вважається пороговим для підозрілої поведінки у конкретному середовищі
DeviceNetworkEvents
| where Timestamp > ago(1d) and RemoteIP startswith "172.16" or RemoteIP startswith "192.168"
| summarize
by DeviceName, RemoteIP, RemotePort, InitiatingProcessFileName
| summarize RemotePortCount=dcount(RemotePort) by DeviceName, RemoteIP, InitiatingProcessFileName
| where RemotePortCount > remotePortCountThreshold
```

Техніка: Виявлення системної інформації.

Техніка: Виявлення за подією.

```
// Техніка: Виявлення за подією. Папірна Г.К.
let accessibilityProcessNames = dynamic(["utilman.exe","osk.exe","magnify.exe","narrator.exe","displayswitch.exe","atbroker.exe","sethc.exe", "helppane.exe"]);
// Здійснення запитів відкладок, підключених за допомогою параметрів реєстру до процесів доступності
let attachedDebugger =
DeviceRegistryEvents
| where Timestamp > minTime
and RegistryKey startswith @"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\"
and RegistryValueName =~ "debugger"
// Розбір імен налагоджених процесів із розділу реєстру
| parse RegistryKey with @"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\" FileName
| where FileName in~ (accessibilityProcessNames) and isnotempty(RegistryValueData)
| project Technique="AttachedDebugger", FileName, AttachedDebuggerCommandline=RegistryValueData, InitiatingProcessCommandLine, Timestamp, DeviceName;
```

```
// Пошук запитів на перезапис файлів спеціальних можливостей
let fileOverwriteOfAccessibilityFiles =
DeviceFileEvents
| where Timestamp > minTime
and FileName in~ (accessibilityProcessNames)
and FolderPath contains @"Windows\System32"
| project Technique="OverwriteFile", Timestamp, DeviceName, FileName, SHA1, InitiatingProcessCommandLine;
// Порівнювання стандартних значень хешів іменованих системних файлів із метаданими тих, що завантажуються через командний рядок, інструменти PowerShell
let executedProcessIsPowershellOrCmd =
DeviceProcessEvents
| project Technique="PreviousOverwriteFile", Timestamp, DeviceName, FileName, SHA1
| where Timestamp > minTime
| where FileName in~ (accessibilityProcessNames)
| join kind=leftsemi(
DeviceProcessEvents
| where Timestamp > ago(14d) and (FileName =~ "cmd.exe" or FileName =~ "powershell.exe")
| summarize MachinesCount = dcount(DeviceName) by SHA1
| where MachinesCount > 5
| project SHA1
) on SHA1;
// Використовується зовнішнє об’єднання всіх результатів, тому що схеми трохи розрізняються між таблицями , а необхідно отримати множину всіх таблиць разом
attachedDebugger
| union kind=outer fileOverwriteOfAccessibilityFiles
| union kind=outer executedProcessIsPowershellOrCmd
```

```
// Техніка: Виконання користувачем. Папірна Г.К.
// Виявлення двох методів виконання зловмисного коду із офісних документів (макросів) та документів формату .pdf.
let lolbins = dynamic(["At.exe", "Atbroker.exe", "Bash.exe", "Bitsadmin.exe", "CertReq.exe", "Certutil.exe", "Cmd.exe", "Cmdkey.exe", "Cmstp.exe", "Control.exe",
"Csc.exe", "Cscript.exe", "Desktopimgdownldr.exe", "Dfsvc.exe", "Diantz.exe", "Diskshadow.exe", "Dnscmd.exe", "Esentutl.exe", "Eventvwr.exe", "Expand.exe",
"Extexport.exe", "Extrac32.exe", "Findstr.exe", "Forfiles.exe", "Ftp.exe", "GfxDownloadWrapper.exe", "Gpscript.exe", "Hh.exe", "Ie4uinit.exe", "Ieexec.exe",
"Ilasm.exe", "Infdefaultinstall.exe", "Installutil.exe", "Jsc.exe", "Makecab.exe", "Mavinject.exe", "Microsoft.Workflow.Compiler.exe", "Mmc.exe", "MpCmdRun.exe",
"Msbuild.exe", "Msconfig.exe", "Msdt.exe", "Mshta.exe", "Msiexec.exe", "Netsh.exe", "Odbcconf.exe", "Pcalua.exe", "Pcwrun.exe", "Pktmon.exe", "Presentationhost.exe",
"Print.exe", "Psr.exe", "Rasautou.exe", "Reg.exe", "Regasm.exe", "Regedit.exe", "Regini.exe", "Register-cimprovider.exe", "Regsvcs.exe", "Regsvr32.exe", "Replace.exe",
"Rpcping.exe", "Rundll32.exe", "Runonce.exe", "Runscripthelper.exe", "Sc.exe", "Schtasks.exe", "Scriptrunner.exe", "SyncAppvPublishingServer.exe", "Ttdinject.exe",
"Tttracer.exe", "vbc.exe", "Verclsid.exe", "Wab.exe", "Wmic.exe", "Wscript.exe", "Wsreset.exe", "Xwizard.exe", "AgentExecutor.exe", "Appvlp.exe", "Bginfo.exe", "Cdb.exe",
"csi.exe", "Devtoolslauncher.exe", "dnx.exe", "Dotnet.exe", "Dxcap.exe", "Excel.exe", "Mftrace.exe", "Msdeploy.exe", "msxsl.exe", "ntdsutil.exe", "Powerpnt.exe", "rcsi.exe",
"Sqldumper.exe", "Sqlps.exe", "SQLToolsPS.exe", "Squirrel.exe", "te.exe", "Tracker.exe", "Update.exe", "vsjitdebugger.exe", "Winword.exe", "Wsl.exe"]);
DeviceImageLoadEvents
| where FileName in~ ("mscoree.dll", "mscorlib.dll", "mscorlib.ni.dll") and InitiatingProcessFileName in~ ("winword.exe", "excel.exe", "powerpnt.exe")
and InitiatingProcessCommandLine has_any (".doc", ".wbk", ".docm", ".dot", ".dotm" ".xls", ".xlsm", ".xltm", ".xla",".xll", ".xlam", ".ppt", ".pptm",
".pdf", ".pot", ".potm", ".ppsm", ".sldm") and not(InitiatingProcessCommandLine has_any (".docx", "dotx",".xlsx", ".xltx", ".pptx", ".pdf"))
| extend InitiatingProcessFileName=tolower(InitiatingProcessFileName)
| summarize by DeviceId, InitiatingProcessId, InitiatingProcessFileName
// Визначення випадків, коли документ запускає інший процес і отримує інформацію про запущений бінарний файл
| join kind=inner hint.strategy=broadcast (
DeviceProcessEvents
| where InitiatingProcessFileName in~ ("winword.exe", "excel.exe", "powerpnt.exe") and FileName !in~ ("winword.exe", "excel.exe", "powerpnt.exe")
// Перевірка, чи є запущений виконавчий файл: підписаним Microsoft, непідписаним із низькою глобальною поширеністю або відомим LOLBIN
| extend InitiatingProcessFileName=tolower(InitiatingProcessFileName)
) on DeviceId, InitiatingProcessFileName, InitiatingProcessId
| where not(isempty(FileName))
| invoke FileProfile("SHA1", 1000)
| where FileName in~(lolbins) or (
not (IsCertificateValid and IsRootSignerMicrosoft)
and not (GlobalPrevalence >= 500 and IsCertificateValid))
// Визначення випадків, коли процес запуску документу впроваджує код в інший процес через віддалений виклик або запис безпосередньо до пам’яті системи
DeviceImageLoadEvents
| where FileName in~ ("mscoree.dll", "mscorlib.dll", "mscorlib.ni.dll") and InitiatingProcessFileName in~ ("winword.exe", "excel.exe", "powerpnt.exe")
and InitiatingProcessCommandLine has_any (".doc", ".wbk", ".docm", ".dot", ".dotm" ".xls", ".xlsm", ".xltm", ".xla",".xll", ".xlam", ".ppt", ".pptm", ".pot", ".potm",
".pdf", ".ppsm", ".sldm")
and not(InitiatingProcessCommandLine has_any (".docx", "dotx",".xlsx", ".xltx", ".pdf", ".pptx"))
| extend InitiatingProcessFileName=tolower(InitiatingProcessFileName)
| summarize by DeviceId, InitiatingProcessId, InitiatingProcessFileName
| join kind=inner hint.strategy=broadcast (
DeviceEvents
| where ActionType in ("CreateRemoteThreadApiCall", "WriteProcessMemory")
| where InitiatingProcessFileName in~ ("winword.exe", "excel.exe", "powerpnt.exe", "pdf.exe",)
| extend InitiatingProcessFileName=tolower(InitiatingProcessFileName))
on DeviceId, InitiatingProcessFileName, InitiatingProcessId
```

Техніка: Виконання користувачем.