НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено
Завідувач кафедри

_____ **Дмитро ЛАНДЕ**
(підпис)

«_____»_____2022 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека»

на тему: Атрибуція APT груп та побудова залежностей на основі зібраних даних

Виконала: здобувачка вищої освіти **IV** курсу, групи **ФБ-83**
(шифр групи)

Мазуренко Віола Олександрівна
(прізвище, ім'я, по батькові) (підпис)

Керівник к.т.н., доцент Барановський Олексій Миколайович
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент к.т.н., доцент Астраханцев Андрій Анатолійович
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувачка вищої освіти _____
(підпис)

Київ – 2022 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Дмитро ЛАНДЕ
(підпис)

«___»_____2022 р.

ЗАВДАННЯ
на дипломну роботу здобувачці вищої освіти

Мазуренко Віолі Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Атрибуція APT груп та побудова залежностей на основі зібраних даних,
керівник роботи к.т.н., доцент Барановський Олексій Миколайович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від « » _____ 2022 р. №

2. Термін подання здобувачкою вищої освіти роботи 10 червня 2022 р.

3. Вихідні дані до роботи: перелік APT груп наданий ETDA та MITRE ATT&CK, праця Т. Стеффенса "Attribution of Advanced Persistent Threats: How to Identify the Actors Behind Cyber-Espionage".

4. Зміст роботи: аналіз APTs та існуючих методів атрибуції, розробка власної моделі атрибуції та її програмної реалізації, побудова залежностей між даними за APT групами.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація.

6. Дата видачі завдання 17.09.2021.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Вибір напрямку та теми дипломної роботи	06.09.21 – 17.09.21	
2	Аналіз наукової літератури стосовно обраної теми	18.09.21 – 01.04.22	
3	Дослідження існуючих проблем атрибуції та APTs	02.04.22 – 12.04.22	
4	Написання першого розділу роботи	12.04.22 – 15.04.22	
5	Пошук та аналіз наявних моделей проведення атрибуції	16.04.22 – 26.04.22	
6	Опис та порівняння існуючих моделей	27.04.22 – 01.05.22	
7	Проходження переддипломної практики та написання програмного коду	02.05.22 – 29.05.22	
8	Оформлення створеної моделі та розробленого програмного коду	29.05.22 – 09.06.22	
9	Підготовка презентації за дипломною роботою для передзахисту	10.06.22 – 12.06.22	
10	Передзахист дипломної роботи	13.06.22	
11	Виконання доопрацювання відповідно до зауважень комісії	14.06.22 – 19.06.22	
12	Захист дипломної роботи	20.06.22	

Здобувачка вищої освіти _____ Віола МАЗУРЕНКО
(підпис) **(Власне ім'я, ПРІЗВИЩЕ)**

Керівник роботи _____ Олексій БАРАНОВСЬКИЙ
(підпис) **(Власне ім'я, ПРІЗВИЩЕ)**

РЕФЕРАТ

Дипломна робота містить: 76 сторінок, 30 ілюстрацій, 2 додатки, 17 джерел використаної літератури.

Об'єкт дослідження: APTs.

Мета роботи: здійснення атрибуції APT груп, створення моделі атрибуції, аналіз та побудова залежностей між даними за кожною з груп.

Завдання роботи: проаналізувати наукову літературу за APTs та їхньою атрибуцією, зокрема оглянути існуючі проблеми атрибуції; розглянути наявні моделі та фреймворки, що застосовуються для проведення атрибуції, й оцінити недоліки та переваги кожної з них; розробити нову модель атрибуції і представити її програмну реалізацію.

Методи дослідження: аналіз наукової літератури стосовно визначеної теми, збір та вивчення відомих даних за APT групами, створення програмного коду й ілюстрація залежностей на основі створеної колекції даних та їхній аналіз.

Результати: вперше створена модель здійснення процесу атрибуції на основі колективної атрибуції та ілюстровані кореляції між даними за APT групами.

Рекомендації щодо використання: застосовувати для визначення зловмисників (APTs) і їхніх замовників та надання рекомендацій для обмеження можливості реалізації атаки у майбутньому, зменшення й усунення її наслідків.

Можливі напрямки розвитку: створення на основі запропонованої моделі міжнародного стандарту проведення атрибуції.

Ключові слова: APTS, АТРИБУЦІЯ, КІБЕРАТАКА, TTPS, МОДЕЛЬ, БАЗА ДАНИХ, ПРОГРАМНИЙ КОД.

ABSTRACT

The thesis contains: 76 pages, 30 illustrations, 2 appendixes, 17 references.

Object of study: APTs.

Objective: implementation of APT group attribution, creation of attribution model, analysis and building of dependencies between data for each of groups.

Tasks of work: to analyze the scientific literature on APTs and their attribution, in particular, to review the existing problems of attribution; to review the existing models and frameworks used for attribution and to assess the disadvantages and advantages of each; to develop a new attribution model and to present its software implementation.

Research: analysis of scientific literature on this topic, collection and study of known data by APT groups, creation of program code and illustration of dependencies based on the created data collection and their analysis.

Results: for the first time created a model of implementation of the attribution process based on the collective attribution and illustrated correlations between data of APT groups.

Recommendations for use: use to identify attackers (APTs) and their sponsors and provide recommendations to limit the possibility of future attacks, reduce and eliminate its consequences.

Possible directions of development: creation on the basis of the offered model of the international standard of carrying out attribution.

Key words: APTS, ATTRIBUTION, CYBER ATTACK, TTPS, MODEL, DATA BASE, PROGRAM CODE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КС – комп'ютерна система

КМ – комп'ютерна мережа

APT – Advanced Persistent Threat

IOC – Indicator of Compromise

TTPs – Tactics, Techniques, and Procedures

ПЗ – програмне забезпечення

ШПЗ – шкідливе програмне забезпечення

ETDA – Electronic Transactions Development Agency

ВСТУП

Виникнення Інтернету надало майже необмежені можливості для створення, втілення і поширення ідей та запустило невпинний процес цифровізації у світі. Щороку створюється і впроваджується дедалі більше різноманітних інноваційних технологій та методів, які базуються на використанні КС та КМ, у всі сфери життя людини. Але з шаленим успіхом міжнародної комп'ютерної мережі з'явилася значна кількість небезпек, зокрема таке поняття як APT.

Хоча перша згадка про появу цього терміну датується 2006 роком [1], самі APTs довго залишалися (а деякі з них залишаються і до сьогодні) непоміченими. Масштабного вжитку це поняття набуло лише починаючи з 2010 року, що є точкою відліку високої активності APT груп у світовому кіберпросторі (зокрема на теренах України це питання набуло особливого значення з 2014 року), де їхніми цілями є державні організації, енергетичні об'єкти, об'єкти військової інфраструктури та інше. Так виникла нагальна потреба визначення походження і мети виконаної атаки, дослідження поведінки та детального розгляду використаних інструментів і засобів кожної з груп для подальшого усунення чи пом'якшення результатів кіберінцидентів та уникнення можливих нападів у майбутньому.

Актуальність роботи полягає в тому, що APTs відіграють головну роль у ході кібервійни, і станом на сьогоднішній день їхній аналіз та атрибуція є критично важливими для державної та міжнародної безпеки.

Метою роботи є проведення атрибуції APT груп, створення моделі атрибуції, побудова й аналіз залежностей між даними за кожною з груп.

Завдання, які необхідно виконати для досягнення поставленої мети:

1. Проаналізувати наукову літературу за APTs та їхньою атрибуцією, зокрема оглянути існуючі проблеми атрибуції;
2. Розглянути наявні моделі та фреймворки, що застосовуються для проведення атрибуції, й оцінити недоліки та переваги кожної з них;

3. Розробити нову модель атрибуції і представити її програмну реалізацію:
    - проаналізувати збірки відомих APTs та обрати кілька з них за основу для подальшого проведення атрибуції;
    - здійснити пошук звітів за обраними APT групами, дослідити і виокремити часовий проміжок здійснених атак та IOCs (акцент на хешах використовуваних файлів) кожної групи;
    - створити потрібні бази даних з інформацією за APT групами і класи для їхнього опрацювання;
    - побудувати залежності між зібраними даними за допомогою графіків;
    - проаналізувати отримані результати та надати опис рекомендацій для усунення/зменшення наслідків здійснених кібернападів та уникнення у майбутньому схожих атак.

Об'єктом дослідження даної роботи є APTs.

Предметом дослідження є покращення процесу атрибуції.

Методами дослідження є аналіз наукової літератури стосовно визначеної теми, збір та вивчення відомих даних за APT групами, створення програмного коду й ілюстрація залежностей на основі створеної колекції даних та їхній аналіз.

Наукова новизна одержаних результатів. Було вперше запропоновано модель колективної атрибуції та зображено кореляцію даних, зібраних за APT групами.

1 ПРОЦЕС АТРИБУЦІЇ КІБЕРАТАК

1.1 Основні поняття та методи атрибуції

У широкому сенсі поняття *атрибуції* визначається як процес або дія пов'язування якогось наслідку з явищем або особою, що його зумовила. Розглядаючи наведений термін з погляду кіберпростору, його можна охарактеризувати як аналітичний процес, завдання якого полягає у наданні відповіді на запитання “Хто?” та “Чому?” вчинив кібератаку.

Атрибуція APTs робить значний внесок у забезпеченні національної та міжнародної безпеки. Насамперед вона надає змогу визначити відповідального: не лише у ролі фізичного виконавця кібернападу, але й у значенні країни, організації чи окремої особи, що доручила і спонсорувала завдання.

Завдяки детальному розгляду ситуації з боку постраждалих територій та секторів й аналізу використаних інструментів, атрибуція дозволяє сформулювати напрямок для встановлення мети проведеної операції та охарактеризувати поведінку зловмисника. За допомогою результатів проведеного процесу відбувається впровадження та створення нових процедур, засобів і способів захисту атакованого об'єкта.

Так формується чотири основні цілі проведення атрибуції [2]:

- *тактична* (як?) – відповідає за розгляд кіберінциденту з технічного погляду;
- *оперативна* (що?) – має на меті опису профілю зловмисника та високорівневої архітектури проведеної атаки;
- *стратегічна* (хто? і чому?) – надає завдання визначення відповідального за кібернапад і його мотивації, значимість самого інциденту та відповідну на нього відповідь;

- *комунікаційна* – акцентує увагу на важливості враховувати міжорганізаційне обговорення здійсненого аналізу та його результатів.

Атрибуцію можна представити у вигляді *шести етапів*: зібрання даних – забезпечення актуальності інформації, кластеризація – поділ даних на набори вторгнень (*intrusion sets*) завдяки ознакам певної технічної подібності, визначення мотивації – чи атака спонсорована державою, чи є злочинним шпигунством, ідентифікація країни-джерела атаки, ідентифікація зловмисної групи осіб чи організації та демонстрація результатів і висловлення гіпотез [1].

Оглянувши структуру описаного процесу виникає запитання про формування наборів вторгнень або, так званих кластерів. Вони є поєднанням паралельного використання зловмисниками *TTPs* та *IOCs*.

Перші відповідають за опис характеристики манер зловмисника: чому він використав задіяні методи? Які техніки він застосував? Як вони допомагають досягти поставленої мети? Другі ж, зі свого боку, представлені у вигляді конкретного об'єкту або дії, що свідчить про факт здійснення несанкціонованого доступу до КС чи КМ.

Тепер варто зрозуміти, що являє собою згадане вище поняття *APT* або *Advanced Persistent Threat*, і для цього потрібно розглянути кожен його компонент окремо [1]:

- *Advanced* (укр. розвинена) – злочинці, здійснюючи кібернапад, застосовують різноманіття технік і способів, які є поєднанням відразу кількох інструментів та методів атаки, постійно їх удосконалюючи та модифікуючи.
- *Persistent* (укр. стійка) – злочинці усвідомлено і заздалегідь обирають свою ціль та атакують її протягом довготривалого проміжку часу залишаючись непоміченими, безперервно спостерігаючи і взаємодіючи з нею.
- *Threat* (укр. загроза) – злочинці мають певну стратегію, конкретну мотивацію та відповідну організацію.

Тобто, APT не є шкідливою програмою, фрагментом коду, певною практикою чи технічним прийомом, а кваліфікованою групою осіб (APT групою) з визначеним наперед наміром та планом отримання несанкціонованого доступу до КС чи КМ та перебування у ній якомога довше.

1.2 Атрибуція APT-атак

Нині кібератаки є частиною повсякденного життя. Неминучість їхнього існування та розвитку прямолінійно залежить від науково-технічного прогресу, який успішно долає всі труднощі та досягає все нових висот.

APTs є небезпекою нового покоління у кіберпросторі. Кожна з груп є складним механізмом, що поєднує у собі детальну підготовку, суттєве фінансування, передові технології і високий професійний рівень.

Розглянемо, чому варто відокремлювати APTs від класичних кібератак, що нам так добре знайомі:

1. APT – це злагоджений колектив фахівців, зловмисник у кібератаці – зазвичай виступає однією людиною або групою недосвідчених початківців.
2. APTs – цілеспрямовані та зосереджені на перевагах економічного та політичного характеру (загалом глобального), зловмисник у кібератаці – мотивований персональною або фінансовою користю.
3. APT – забезпечена ресурсами та провідним інструментарієм, створює самостійно чи купує ПЗ в інших груп, зловмисник у кібератаці – використовує загальнодоступні утиліти, ПЗ та методи реалізації атак.
4. APT – заздалегідь обирає ціль та проводить підготовку для отримання доступу до неї, зловмисник у кібератаці – випадкова жертва і/або непродуманий план виконання нападу.

5. APTs – мають на меті залишитися у КС чи КМ якомога довше непоміченими та зібрати якнайбільше потрібної інформації (чи заподіяти якомога сильніших втрат), зловмисник у кібератаці – лімітований у часі через швидкий та непродуманий до деталей підхід, нетерплячий.
6. APT – постійно взаємодіють з обраною жертвою (КС/КМ) та пристосовується до будь-яких модифікацій у ній, зловмисник у кібератаці – зазвичай здається після першої невдачі.

Незважаючи на те, що APT є загрозою передового рівня з швидким розвитком, перевершуючи стандартні атаки в рази, не варто недооцінювати базові рекомендації та способи для підтримки захисту КС та КМ, адже вони добре працюють та створюють додатковий шар безпеки цільового об'єкту. Наприклад: наявність антивірусу та забезпечення необхідних оновлень у системі, підтримка безпеки вебдодатків, вебраузерів та вебсторінок (як внутрішньоорганізаціних, так і зовнішніх), встановлення й обслуговування системи запобігання вторгненням (*Intrusion Prevention System*) і фаєрволу (*firewall*), захист електронних повідомлень та перевірка їх вмісту перед відкриттям (автоматична), і т.д.

APTs використовують різноманітні техніки для проникнення та отримання несанкціонованого доступу до КС чи КМ, і не перестають застосовувати та перевіряти базові способи.

Наразі спеціалісти з кібербезпеки з усього світу пропонують нові методики для сповільнення зростання кількості таких груп, зокрема різні моделі атрибуції, що надає можливість не лише зловити винних, а й висунути звинувачення та присвоїти відповідне покарання.

1.3 Проблеми атрибуції

За останні десять років атрибуція набула ще більшого політичного та поширеного характеру завдяки методам та моделям її перебігу, які постійно удосконалюються [2]. Тому є необхідним, щоб результати цього процесу були якомога точнішими та об'єктивнішими.

Велику роль при виконанні атрибуції відіграє сприйняття знайдених відомостей, тобто висунені припущення щодо отриманих результатів, які тісно пов'язані з психологічним аспектом кожної людини, а саме: світогляду і різноманітних упереджень. Тому завжди потрібно брати до уваги кому це може бути вигідним і дати відповіді на запитання вигляду: “Яка прихована мотивація розкриття інформації чи оприлюднення цього звіту?”, “Що спонукало висловити саме таку думку?”, “Які наміри є передумовою для висловленого твердження?” і тому подібне [3].

Авторка роботи “The Ultimate Challenge: Attribution for Cyber Operations” Аманда Хілл виділяє чотири основні обмеження атрибуції [4]:

1) природа кіберпростору (*nature of cyberspace*) – кіберсердовище має складну структуру, яка постійно розвивається. Мережа Інтернет надає змогу швидкого і зручного транспортування даних, але також містить у собі найрізноманітніші небезпеки, загрози і вразливості. Однією з першочергових переваг для її користувачів є анонімність, що водночас є найбільшою проблемою для спеціалістів при пошуку зловмисників;
2) дефіцит закону (*deficiency of law*) – нестача прийнятого міжнародного законодавства стосовно її проведення та визначення типу відповідальності за скоєні злочини на її основі;
3) державний суверенітет (*state sovereignty*) – проблематика її неточного виконання та, як результат, звинувачення невинних осіб, організацій та країн;

4) обмежені технології (*limited technology*) – особливості архітектури мережі Інтернет надають можливість зловмисникам залишатися невідомими.

Тобто можна виділити дві найбільші проблеми атрибуції: *технічну* і *юридичну* [5].

Перша базується на з'ясуванні осіб виконавців кібератаки, їхнього місцезнаходження та характеру зв'язків, що поєднують між собою одне і друге. Її вирішення безпосередньо полягає у змозі спеціалістів подолати труднощі, що виникають на шляху відстеження зловмисників. Різноманітні засоби і техніки, які забезпечують анонімність, будують багатошаровий захист та утворюють складну архітектуру задіяних пристроїв, створюють багаторівневу платформу для успішної атаки та стають на заваді знаходження відповідальних осіб за її реалізацію.

Друга залежить від коректності одержаних результатів при розв'язанні першої проблеми та визначає, яку відповідальність за завдану шкоду будуть нести зловмисники.

Хибна атрибуція підвищує ризики виникнення чи ескалації конфлікту (зокрема збройного конфлікту) між постраждалою та підозрюваною в атаці країною.

Наразі кількість організацій, які займаються розвідкою та дослідженням кібератак і APTs, зростає як у приватному, так і у публічному секторі економіки. Відсутність єдиного офіційного міжнародного стандарту надає можливість кожній установі здійснювати атрибуцію за своїми правилами та способами, що створює не лише простір для вільного модифікування цього процесу та формування нових підходів до нього, але й високий ризик малоефективного, недостатнього та упередженого проведення оцінки загрози.

Висновки до розділу 1

На сьогоднішній день APT групи є однією з найнебезпечніших проблем захисту мережі Інтернет. Зловмисники, що до них входять, є кваліфікованими спеціалістами своєї справи з великим досвідом, високоякісним набором інструментів та спонсоруванням.

Усе більше країн інвестують кошти та час у розвиток і покращення атрибуції та створення різноманітних моделей її процесу виконання. Вона є необхідним компонентом для надання гарантій безпеки кожної держави і підтримки міжнародних відносин.

Співпраця та проведення колективної атрибуції є ключем до вирішення її проблем та зниження активності APT груп у світовому кіберпросторі (сповільнення їхньої роботи, затримання осіб, які до них входять і тих, які ними керують, створення єдиних інтернаціональних законів та відповідних покарань, тощо).

2 СУЧАСНІ МЕТОДИ МОДЕЛЮВАННЯ АТРИБУЦІЇ

Ажіотаж навколо вирішення питання визначення APT груп нині не стихає та навіть набирає обертів, незважаючи на те, що це поняття було введене більше п'ятнадцяти років тому. З кожним роком на горизонті кіберпростору з'являються все нові способи виконання атрибуції, а дослідники, підтримуючи гонку, змагаються між собою, видозмінюючи та модернізуючи старі.

Щоб краще зрозуміти як відбувається процес атрибуції та скільки ресурсів необхідно на нього витратити, потрібно детально розглянути запропоновані моделі його реалізації останніх років. Підхід до поставленого завдання кожного з авторів (або об'єднання авторів) є унікальним, з розставленням акцентів на різних елементах аналізу, що надає можливість фахівцям обрати саме ту модель, яка відповідає методам їхньої роботи.

2.1 Теоретичні моделі атрибуції

2.1.1 The Q Model

Запропонована у 2014 році Томасом Рідом і Беном Бьюкененом модель “Q Model” (Рисунок 2.1), що була розроблена для “пояснення, орієнтування та покращення виконання атрибуції” [2], заклала фундамент для удосконалення і надала розуміння організації самого процесу атрибуції. Автори поділили його перебіг на три основні рівні: *тактичний* (виключно технічний аспект аналізу), *оперативний* (розуміння поведінки зловмисника та структури атаки) та *стратегічний* (визначення мети й особи зловмисника), і один додатковий, але ні в

якому разі не менш важливий, *комунікаційний* (обмін отриманою інформацію та її обговорення).

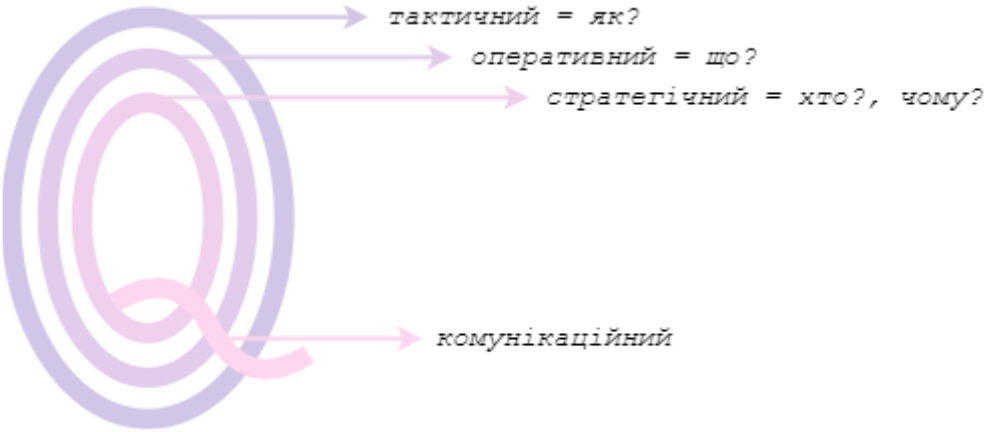


Рисунок 2.1 – Ілюстрація рівнів “The Q Model” [2]

Перший зазвичай є початком для розслідування кіберінциденту, що відбувся. На цьому етапі фахівці вивчають усі можливі деталі, що стосуються використаних технік, тактик та інструментів, як наприклад: IOCs, спосіб отримання доступу до КС чи КМ, кількість задіяних методів, відмітки часу файлів, мова, заголовки запитів і т.і. Наступний рівень сконцентрований на поясненні організації виконання атаки та опису профілю зловмисника, наприклад: оцінка підготовки нападу, його розгляд на кожному етапі Kill Chain, врахування можливостей осіб, що його здійснили та їхніх геополітичних інтересів. На третьому рівні спеціалісти зводять усі дані, що були отримані на попередніх, аналізують їхню сукупність і надають кінцеві твердження. Зокрема сюди входить: оцінка завданої шкоди, характеристика її форми, розгляд наслідків і т.п. Останній етап, значущість якого часто не враховують, забезпечує не лише поліпшення атрибуції, а й підвищення довіри та зменшення критики з боку суспільства до оприлюднених результатів.

Ця модель є легкою для сприйняття, тому не лише кваліфіковані особи, а й пересічні люди мають змогу зрозуміти хід розглянутого процесу, що є важливим для усвідомлення необхідності підтримки кібергігієни. Також саме “The Q Model” використовується як фундамент при створенні нових способів атрибуції.

2.1.2 Cyber Attribution Model (CAM)

Натхнення для створення даної моделі (Рисунок 2.2) Тімеї Пахі і Флоріана Скопіка у 2019 році лежить в основі “The Diamond Model” та знаменитої “Kill Chain”. Вона складається з двох залежних одна від одної частин – аналізу профілю зловмисника та експертизи кібератаки [6].

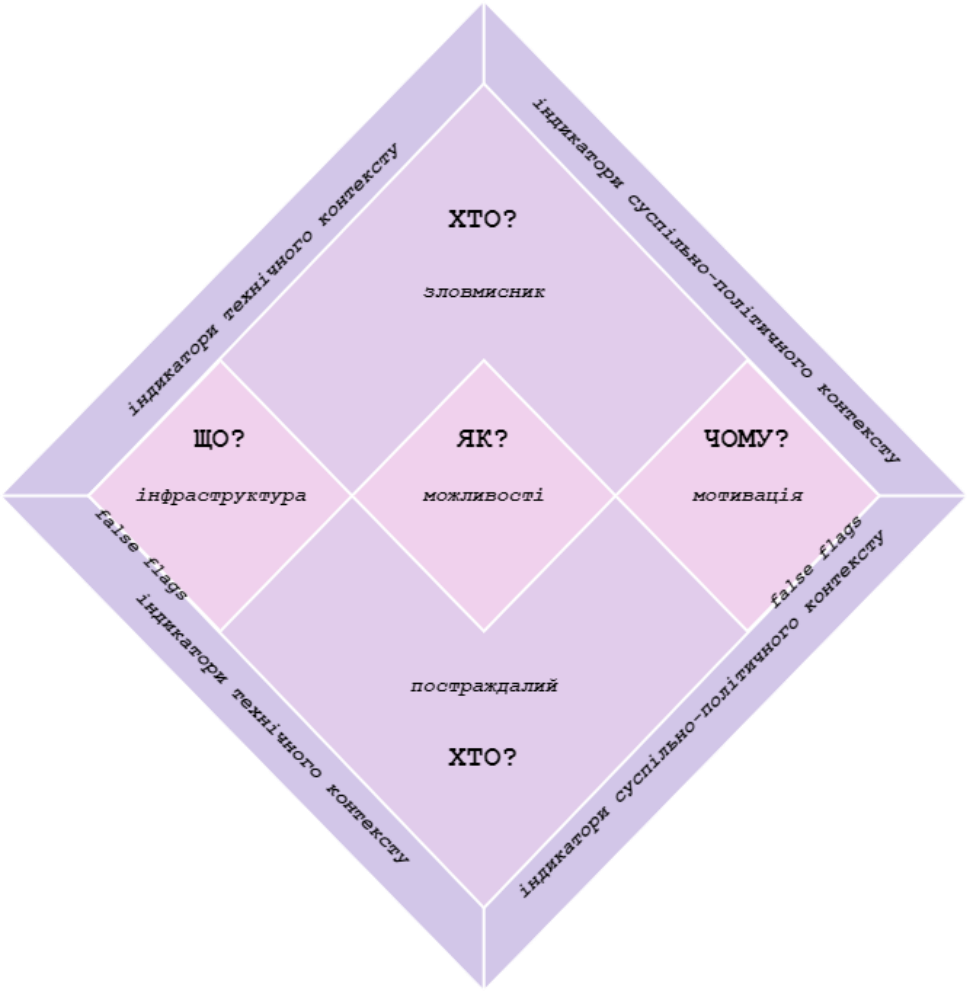


Рисунок 2.2 – Вигляд моделі CAM та її складових [6]

Фахівці неодноразово вказували на проблему застосування APT групами операцій фальшивих прапорів (*false flags operations*) під час кібератак та проведення коректної атрибуції у таких випадках. Тому дана модель окремо їх виділяє і поділяє на два види: *соціально-політичні* та *технічні*. За такою класифікацією відбираються індикатори з відповідним контекстом (технічним або соціально-політичним), які використовуються у вищезгаданих елементах CAM

разом з компонентами, що описані у “The Diamond Model”, та допомагають віднайти подібності між зібраними доказами.

Розгляд профілю правопорушника є верхньою частиною покращеної “діамантової” моделі. Він надає відповідь на такі запитання:

- “Хто є зловмисником?”;
- “Що він використовував при здійсненні атаки?”;
- “Які можливості він має?”;
- “Якою є мотивація скоєного нападу?”.

Хоча дослідження самого кіберінциденту і виступає нижньою частиною CAM, саме з нього починається процес атрибуції. На цьому етапі визначається:

- “Хто є жертвою?”;
- “Чому було обрано цю ціль?”;
- “Що сталося?”;
- “Як це сталося?”.

Перевага цієї моделі полягає у тому, що вона розбиває процес атрибуції на дві компоненти, які виконуються по черзі: спочатку з погляду зловмисника, а потім – жертви, у такий спосіб описуючи його повністю.

2.2 Практичні фреймворки атрибуції

2.2.1 The Triangle Model

“The Triangle Model” (Рисунок 2.3) була створена та опублікована Аруном Варіку у 2021 році. На відміну від попередньо розглянутої CAM, дана модель зосереджена на здійсненні атрибуції з боку порушника. В її основі закладені індикатори атаки, які знаходяться на трьох найвищих рівнях “Pyramid of Pain” Девіда Б'янко, а саме: TTPs, використані інструменти та цільовий економічний

сектор, який безпосередньо пов'язаний з жертвою кібернападу [7]. Автор також підкреслює властивість моделі надавати рівень впевненості (*confidence level*) проведеної атрибуції.

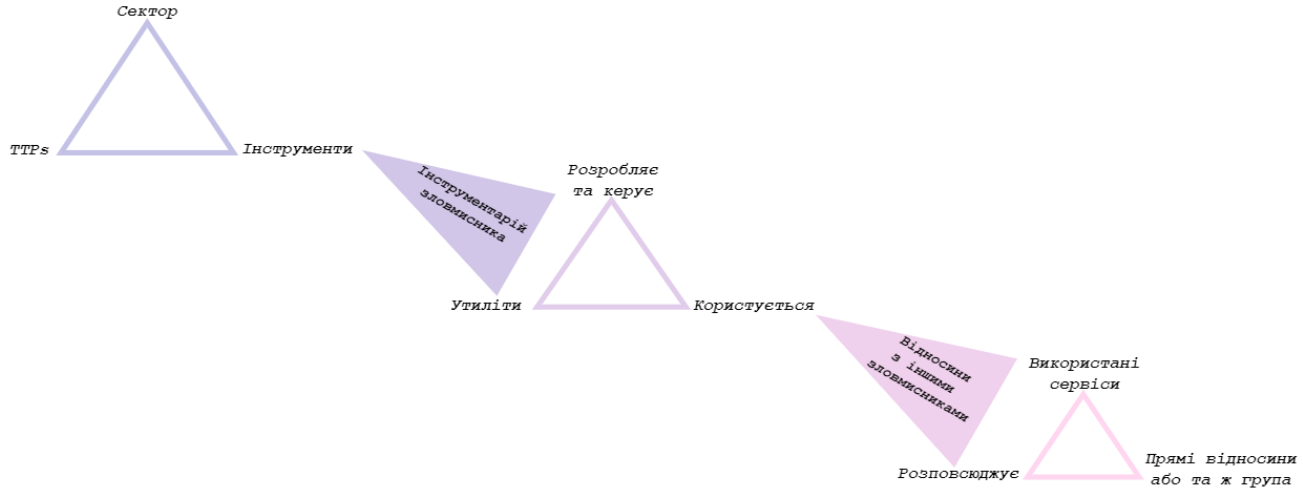


Рисунок 2.3 – Демонстрація “The Triangle Model” [7]

Модель складається з трьох трикутників, які утворюють своєрідний ланцюжок [7]: найбільший і основний трикутник формується з трьох індикаторів згаданих вище: *TTPs* – є найважчим для модифікації і найінформативнішим від усіх інших, застосовані *інструменти* при реалізації нападу, *сектор жертви* – є ключовим для формулювання мети атаки; середній – демонструє структуру інструментарію зловмисника: *розробляє та керує* – ті ШПЗ, які він створює і якими оперує, *користується* – характеризує відносини між ним й іншими APT групами та визначає, якими чужими ШПЗ він користується, утиліти – описує ті ПЗ, які є загальнодоступними (або безкоштовні, або платні); останній і найменший трикутник вказує на підкатегорії того, чим порушник ‘користується’: *використані сервіси* – описує ШПЗ, яке він придбав в іншого зловмисника, *розповсюджує* – відповідає за сумісну працю між APT групами, *прямі відносини або та ж група* – ШПЗ створене групою, що входить до розглянутої.

Підхід до виконання атрибуції цієї моделі цікавий тим, що автор враховує існуючі зв'язки розглянутої групи з іншими та акцентує увагу на їхній можливій співпраці: як самому проведенні атаки, так і у забезпеченні потрібних ресурсів для здійснення нападу.

2.2.2 Intelligent Cyber Attribution (InCA) фреймворк

У 2015 році було представлено *фреймворк* (framework – це “багаторівнева структура, яка вказує, які програми можна або потрібно створювати і як вони будуть взаємопов'язані” [8]) під назвою “Intelligent Cyber Attribution” Пауло Шакаряном, Гегардо Сімарі, Джеффрі Мурсом та Саймоном Парсонсом, основне завдання якого полягає у виконанні коректної атрибуції, незважаючи на сумнівність чи/і суперечливість наявної інформації [9].

Фундаментом для створення InCA слугує дві моделі [9]: середовища (*environmental model*) – призначення якої ґрунтується на опису ймовірнісної базової інформації про середовище, що моделюється, й аналітична (*analytical model*) – зосереджена на аналізі та репрезентації припущень стосовно середовища, що моделюється, конкуруючих між собою, тобто демонструє використання діалектичного методу (*dialectic method*) з критеріями, що допомагають порівняти запропоновані твердження та обрати найбільш правдоподібні.

Фреймворк має вигляд математичної моделі: автори сформували змінні, які являють собою КС, типи кібероперацій, зловмисників і т.і., символи для кожної з зазначених вище моделей, операції над ними та їхні визначення. Його структура складається з великої кількості різноманітних елементів, що забезпечують повноцінність та коректність процесу.

InCA пропонує гнучке здійснення атрибуції за допомогою додавання додаткових до наявних базових констант і правил, що дозволить моделювання не лише APT груп, а й інших порушників, співпраці між різними організаціями (замовниками та виконавцями кібератаки) та врахування можливого використання фальшивих прапорів і т.п.

Висновки до розділу 2

Різноманіття способів реалізації процесу атрибуції нині незліченне. Професіонали та початківці з усього світу пропонують концепції його покращення, застосовуючи все нові й нові методики і розглядаючи його з різних перспектив, як наприклад: окремо з боку зловмисника та окремо з погляду постраждалого.

Існування такої варіативності створює, окрім переваг, ще й головну проблему атрибуції (Розділ 1.3), а саме: відсутність єдиного міжнародного стандарту її проведення. Одні моделі є легкими для сприйняття, але недостатньо інформативними для гарантування коректності результатів, інші, враховуючи кожну деталь, є доволі складними і вимагають витрати великої кількості часу для розуміння усіх нюансів їхнього застосування. Тому на сьогодні питання створення універсальної моделі залишається відкритим.

3 УДОСКОНАЛЕНА МОДЕЛЬ АТРИБУЦІЇ CSIAC

3.1 Опис структури моделі

Сучасні методи атрибуції не надають відповіді вирішення її основних проблем (Розділ 1.3) та базуються на знаходженні доказів у межах однієї організації чи команди фахівців.

Зважаючи на поставлені завдання для знаходження рішення до цих питань, мною була розроблена модель CSIAC (Рисунок 3.1), основою якої є концепція колективної атрибуції, а конкретно: поєднання інформації з різних джерел.

Головною ціллю розробки є вдосконалення технічного аспекту процесу атрибуції та створення платформи для укладання загальноприйнятих міжнародних норм його проведення у майбутньому.

Рисунок 3.1 – Кроки моделі CSIAC

Так як атрибуція – це аналітичний процес, CSIAC має вигляд ланцюжка дій, які описують його перебіг, але варто зауважити, що модель передбачає повернення на вже пройдені етапи, як наприклад: додавання нових відомостей після сортування вже наявних чи повторного відбору для здійснення аналізу, враховуючи потрібні зміни та розставляючи, відмінні від попередніх, акценти.

Завдяки її динамічній структурі у спеціалістів є можливість підвищення якості процесу за рахунок покращення зібраних даних, модифікації їхнього аналізу, визначення нових правил/методів класифікації і т.і, не змінюючи саму архітектуру CSIAC та її елементів.

Розглянемо компоненти цієї моделі:

1. ***C*** або ***C***ollect – збір уже відомої на цей момент інформації за APT групою або групами з різних джерел: статей, звітів, книг, збірок і т.д. До такої інформації входять: назва групи, інші назви групи (*aliases*), країна походження, цільові країни і сектори, що були нею атаковані, TTPs, інструменти, IOCs, час проведення кібероперації(-й) (*timeline*) і т.д.
2. ***S*** або ***S***ort – сортування накопичених даних за критеріями наповненості і достовірності та створення набору відфільтрованих даних (*dataset*).
3. ***I*** або ***I***llustrate – ілюстрація отриманих даних та існуючих між ними залежностей за допомогою графіків.
4. ***A*** або ***A***nalyze – аналіз зібраної інформації на основі побудованих зв'язків.
5. ***C*** або ***C***onclude – надання результатів та рекомендацій для обмеження можливості здійснення повторних атак й усунення/зменшення наслідків успішно проведених.

CSIAC надає можливість підвищення точності і коректності атрибуції та демонстрації наявних кореляцій між відібраними даними шляхом побудови графіків залежностей.

Також, на останньому кроці, вона не лише концентрує увагу на висновках проведеного аналізу, а й дозволяє спеціалістам запропонувати методи і способи для удосконалення захисту атакованої КС чи КМ.

3.2 Програмна реалізація CSIAC

Для демонстрації ефективності створеної моделі як приклад її застосування було розроблено програмний код на мові Python з використанням властивостей об'єктно-орієнтованого програмування та системи керування базами даних SQLite.

3.2.1 Пошук та збір інформації

Першим кроком за описаною моделлю CSIAC є збір необхідних даних. За основні джерела інформації після ретельного пошуку було узято загальнодоступні переліки APT груп, надані ETDA [10], що містить: відомі назви кожної з груп та посилання на авторів, які їх присвоїли, країну походження зловмисників, країни, у яких спостерігалася їхня активність, сектори, що були ними атаковані, використані інструменти і т.п. та MITRE ATT&CK [11], яка надає список відомих назв групи і/або можливих зв'язків з іншими групами, технік, тактик та її інструментарій.

Особливість ETDA полягає у тому, що організація демонструє зібрані дані з різноманітних ресурсів та вказує посилання, де можна віднайти інформацію, що була відібрана, та її авторів. Також, це чи не єдина колекція APTs, яка постійно оновлюється та зазначає ті групи, що були активні останні три місяці.

Перевага ATT&CK насамперед складається з подання переліку застосованих певною групою тактик і технік. Обоє понять описують поведінку зловмисника, що допомагає у подальшому сформувати його профіль та значно зменшує об'єм роботи фахівців та аналітиків при знаходженні ідентичних характеристик під час проведення атрибуції чи звичайній перевірці КС або КМ. Техніка є більш поглибленим поясненням тактики. MITRE визначає останню як відповідь на питання “Чому було виконано конкретну дію?”, а першу – на питання “Як було виконано цю дію?”.

ETDA пропонує перегляд своєї енциклопедії APT груп за допомогою вебраузера, але її також можна завантажити у форматі JSON чи MISP. Зокрема, є можливість окремо зберегти на свій пристрій “картку” конкретної групи або у розширенні JSON, або у PDF. Мною було вирішено скористатися пропозицією завантаження усієї збірки у форматі JSON.

Для отримання необхідних даних з ATT&CK було застосовано можливості модуля Python pyattck [12], який знаходиться у вільному доступі для всіх бажаючих ним скористатися.

Функція ***read_data(file_name)*** (Рисунок 3.2) відповідає за зчитування даних зі збереженого файлу, де його ім'я передається до неї як параметр, та запису цих даних до словника, який вона повертає як своє значення.

```
17  #зчитування даних з файлу у словник
18  def read_data(file_name):
19      data = open(file_name, encoding = "utf8")
20      data_dict = dict()
21      data_dict = json.load(data)['values']
22      data.close()
23      return data_dict
```

Рисунок 3.2 – Запис інформації, представленої ETDA, у словник

Далі необхідно створити окремий перелік назв APT груп та зафіксувати його у вигляді окремої змінної. Для цього використовується функція ***get_names(data)*** (Рисунок 3.3), яка приймає на вхід отримані дані з read_data(file_name), та повертає список APT груп за їхньою назвою.

```
25  #складання переліку apt груп за назвою
26  def get_names(data):
27      names_list = []
28      for apt_group in data:
29          names_list.append(apt_group['actor'].split(', ')[0])
30      return names_list
```

Рисунок 3.3 – Запис назв усіх APT груп до окремого списку

3.2.2 Фільтрація і структуризація даних

CSIAC визначає другий етап проведення атрибуції як сортування зібраних даних та створення відфільтрованого датасету. Для зручності проведення відбору й упорядкування зібраної інформації було застосовано об'єктно-орієнтовне програмування та створено два класи: APT_World – для APT груп, що працювали та працюють зараз у всьому світі, та APT_Ukraine – для тих груп, які безпосередньо атакували кіберпростір України. Вони поєднанні між собою відношенням успадкування, а саме: APT_Ukraine є дочірнім класом APT_World, що вказує на

можливість використання його методів і властивостей. Такий підхід забезпечує мінімізацію розмірів коду та надає легкість і зручність доповнення його у майбутньому.

Клас APT_World містить змінні (Рисунок 3.4) та функції (Рисунок 3.5), що необхідні не лише для структуризації і відбору даних, а й для створення бази APT_Groups на їхній основі, яка міститиме дві таблички – одна для APT_World (Рисунок 3.6), а інша для його дочірнього класу APT_Ukraine (Рисунок 3.10).

```
32  #усі APT групи зі списку
33  class APT_World:
34
35      def __init__(self, data, name):
36          self.data = data
37          self.name = name
38          self.aliases_etda = []
39          self.aliases_mitre = []
40          self.country_of_origin = ""
41          self.target_countries = []
42          self.target_sectors = []
43          self.techniques = []
44          self.tactics = []
45          self.tools = []
```

Рисунок 3.4 – Змінні класу APT_World

```
35  def __init__(self, data, name):
46
47      #знаходження усіх назв за ETDA
48  def get_aliases_etda(self):
60
61      #знаходження усіх назв за Mitre Att&ck
62  def get_aliases_mitre(self):
82
83      #знаходження країни походження
84  def get_country_of_origin(self):
90
91      #знаходження усіх постраждалих країн
92  def get_target_countries(self):
102
103     #знаходження усіх постраждалих секторів
104 def get_target_sectors(self):
114
115     #знаходження усіх інструментів
116 def get_tools(self):
126
127     #знаходження усіх технік та тактик
128 def get_techniques_and_tactics(self):
145
146     #пошук за вказаною постраждалою країною
147 def check_target_country(self, country):
151
152     #перевірка достатньості кількості інформації
153 def info_sufficiency(self):
164
165     #запис даних у базу даних
166 def insert_data(self):
175
176 def get_all(self):
184
185 def main(self):
190
```

Рисунок 3.5 – Функції класу APT_World

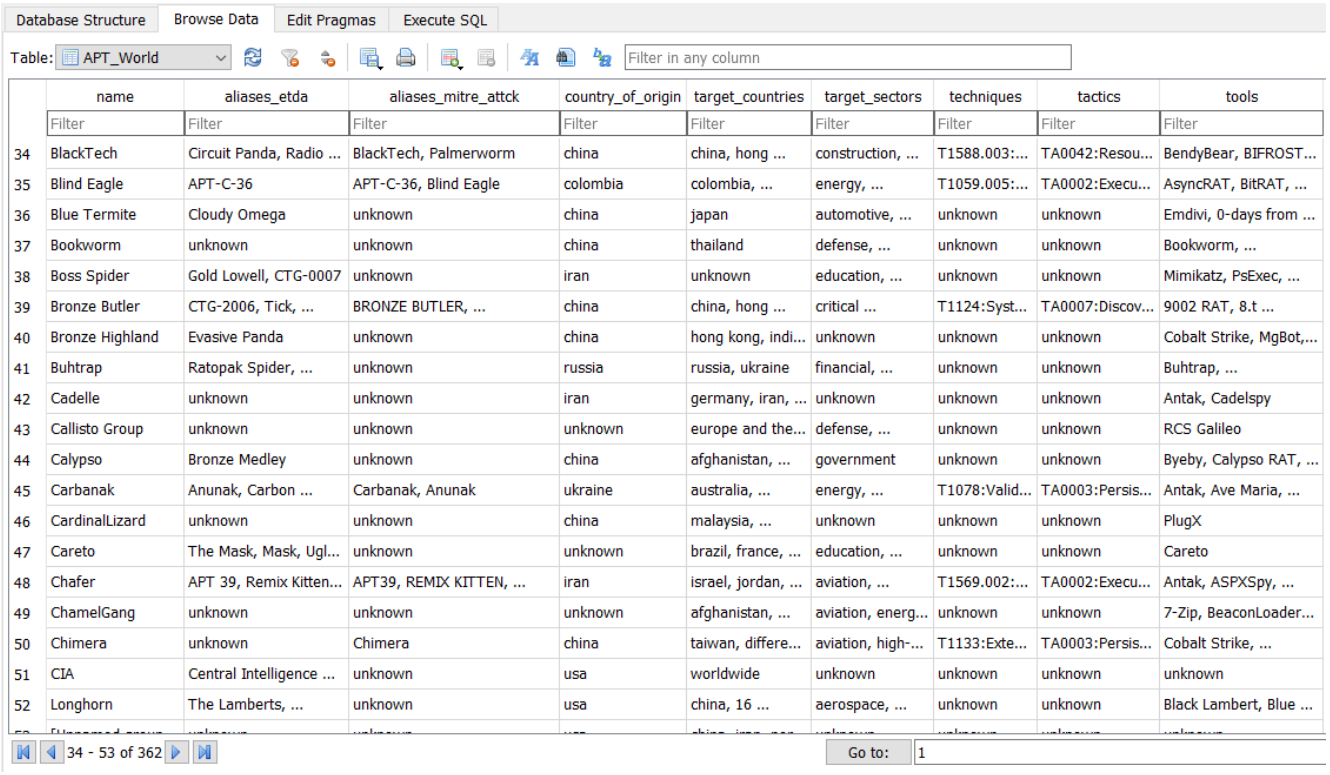
Розглянемо детальніше структуру APT_World:

- Властивості:
  1) ***data*** – змінна, у яку записується словник, отриманий за допомогою read_data(file_name) та переданий до класу як параметр;
  2) ***name*** – назва APT групи для якої створюється об'єкт класу, аналогічно data вона є параметром, що передається. За циклом for відбувається перебір усіх груп, а конкретно – їхніх назв, з попередньо отриманого переліку завдяки get_names(data);
  3) ***aliases_etda*** **–** усі відомі назви (псевдоніми) вказаної APT, що зібрані та записані організацією ETDA;
  4) ***aliases_mitre*** **–** псевдоніми та інші групи, що мають схожу характеристику з поточною, за ATT&CK;
  5) ***country_of_origin*** **–** країна походження вказаної групи (або територія);
  6) ***target_countries*** **–** перелік тих країн (чи/і територій), у яких були знайдені докази роботи розглянутої APT;
  7) ***target_sectors*** **–** список цільових економічних секторів, що зазнали витрат, у яких була реалізована атака чи атаки;
  8) ***techniques*** **–** застосовані групою техніки за ATT&CK;
  9) ***tactics*** **–** здійснені APT тактики за ATT&CK;
  10) ***tools*** **–** використані інструменти для успішного проведення кібернападу.
- Методи:
  1) ***__init__(self, data, name)*** – визначена функція для ініціалізації атрибутів об'єкта класу (у нашому випадку – вказаної APT). До неї передаються попередньо зчитані дані як data та обрана назва групи як name;

2) ***get_aliases_etda(self)*** – відповідає за відбір усіх можливих псевдонім даної групи, які надає ETDA, записує та повертає їх у вигляді змінної aliases_etda;
3) ***get_aliases_mitre(self)*** – перебирає усі можливі назви APT, надані ETDA, та шукає збіг серед переліку, що представляє ATT&CK фреймворк. Повертає знайдені значення як aliases_mitre;
4) ***get_country_of_origin(self)*** – пошук за попередньо зчитаними даними з файлу країни походження поточної групи та повертає властивість класу country_of_origin;
5) ***get_target_countries(self)*** **–** здійснює пошук за назвою групи по файлу та записує перелік атакованих країн у змінну target_coutries;
6) ***get_target_sectors(self)*** **–** аналогічно до попередньої функції. Повертає зібрані значення як змінну target_sectors;
7) ***get_tools(self)*** **–** відбирає інструменти, що використовувала вказана APT група, зі списку, наданого ETDA;
8) ***get_techniques_and_tactics(self)*** **–** пошук за знайденими псевдонімами поточної групи у збірці aliases_mitre технік та тактик, що вона застосовувала;
9) ***check_target_country(self, country)*** **–** перевіряє чи країна, що є переданим параметром у функцію, є у переліку цільових країн даної APT та повертає назву групи;
10) ***info_sufficiency(self)*** **–** перевіряє, чи є достатньою кількість інформації, що була отримана з функцій описаних вище, а саме: country_of_origin, target_countries, target_sectors, tools, techniques. Вона рахує і повертає кількість полів, що мають значення “unknown”;
11) ***insert_data(self)*** **–** метод для заповнення бази даних, а конкретно – таблички APT_World у ній. Записує отримані дані з перерахованих функцій до неї;

12) ***get_all(self)*** **–** викликає у собі всі функції, що починаються на get;

13) ***main(self)*** **–** викликає get_all() та info_sufficiency(), яку записує у змінну *counter*. Якщо значення counter менше 4, тобто в діапазоні від 0 до 3 включно, відбувається запис отриманих даних до таблиці.



Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: APT_World — Filter in any column

	name	aliases_etda	aliases_mitre_attck	country_of_origin	target_countries	target_sectors	techniques	tactics	tools
34	BlackTech	Circuit Panda, Radio ...	BlackTech, Palmerworm	china	china, hong ...	construction, ...	T1588.003:...	TA0042:Resou...	BendyBear, BIFROST...
35	Blind Eagle	APT-C-36	APT-C-36, Blind Eagle	colombia	colombia, ...	energy, ...	T1059.005:...	TA0002:Execu...	AsyncRAT, BitRAT, ...
36	Blue Termite	Cloudy Omega	unknown	china	japan	automotive, ...	unknown	unknown	Emdivi, 0-days from ...
37	Bookworm	unknown	unknown	china	thailand	defense, ...	unknown	unknown	Bookworm, ...
38	Boss Spider	Gold Lowell, CTG-0007	unknown	iran	unknown	education, ...	unknown	unknown	Mimikatz, PsExec, ...
39	Bronze Butler	CTG-2006, Tick, ...	BRONZE BUTLER, ...	china	china, hong ...	critical ...	T1124:Syst...	TA0007:Discov...	9002 RAT, 8.t ...
40	Bronze Highland	Evasive Panda	unknown	china	hong kong, indi...	unknown	unknown	unknown	Cobalt Strike, MgBot,...
41	Buhtrap	Ratopak Spider, ...	unknown	russia	russia, ukraine	financial, ...	unknown	unknown	Buhtrap, ...
42	Cadelle	unknown	unknown	iran	germany, iran, ...	unknown	unknown	unknown	Antak, Cadelspy
43	Callisto Group	unknown	unknown	unknown	europe and the...	defense, ...	unknown	unknown	RCS Galileo
44	Calypso	Bronze Medley	unknown	china	afghanistan, ...	government	unknown	unknown	Byeby, Calypso RAT, ...
45	Carbanak	Anunak, Carbon ...	Carbanak, Anunak	ukraine	australia, ...	energy, ...	T1078:Valid...	TA0003:Persis...	Antak, Ave Maria, ...
46	CardinalLizard	unknown	unknown	china	malaysia, ...	unknown	unknown	unknown	PlugX
47	Careto	The Mask, Mask, Ugl...	unknown	unknown	brazil, france, ...	education, ...	unknown	unknown	Careto
48	Chafer	APT 39, Remix Kitten...	APT39, REMIX KITTEN, ...	iran	israel, jordan, ...	aviation, ...	T1569.002:...	TA0002:Execu...	Antak, ASPXSpy, ...
49	ChamelGang	unknown	unknown	unknown	afghanistan, ...	aviation, energ...	unknown	unknown	7-Zip, BeaconLoader...
50	Chimera	unknown	Chimera	china	taiwan, differe...	aviation, high-...	T1133:Exte...	TA0003:Persis...	Cobalt Strike, ...
51	CIA	Central Intelligence ...	unknown	usa	worldwide	unknown	unknown	unknown	unknown
52	Longhorn	The Lamberts, ...	unknown	usa	china, 16 ...	aerospace, ...	unknown	unknown	Black Lambert, Blue ...

34 - 53 of 362 — Go to: 1

Рисунок 3.6 – Таблиця APT_World у базі даних APT_Groups

Для того, щоб відібрати ті APT групи, що виконували кібернапади на об'єкти на території України, необхідно використати згадану вище функцію класу APT_World check_target_country(self, country), до якої передати параметр country зі значенням “ukraine”, та створити перелік цих груп.

Як вже було зазначено у розгляді структури моделі CSIAC, ми можемо повертатися на попередні кроки, тому здійснюючи перехід на перший етап – *collect*, було виконано пошук звітів за сформованим списком APTs, з яких були виділені: часовий проміжок реалізованих кібератак (у вигляді року виконання) і хеші файлів, що використовували розглянуті групи. Дані та посилання на джерела, звідки було їх зібрано, було записано у базу даних з назвою APT_Ukraine_timeline_and_hashes (Рисунок 3.7).

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: APT_Ukraine_hashes_and_timelines — Filter in any column

	name	timeline	timeline_source	hashes_source
	Filter	Filter	Filter	Filter
1	APT 29	2020	https://www.kyivpost.com/technology/microsoft-...	https://www.mandiant.com/resources/tracking-apt29-...
2	Berserk Bear	2019, 2020	https://vblocalhost.com/uploads/VB2021-...	https://symantec-enterprise-blogs.security.com/blogs/...
3	Buhtrap	2016, 2022	https://cert.gov.ua/article/37246	https://cert.gov.ua/article/37246,https://github.com/...
4	Calypso	2021	https://cyware.com/news/calypso-apt-eyes-...	https://github.com/RedDrip7/APT_Digital_Weapon/blo...
5	Carbanak	2014-2015	https://www.mandiant.com/resources/behind-th...	https://www.secureworks.com/blog/excel-add-ins-...
6	Circus Spider	?	?	https://www.crowdstrike.com/blog/analysis-of-ecrime...
7	Cobalt Group	?	?	https://github.com/RedDrip7/APT_Digital_Weapon/blo...
8	Cold River	2022	https://blog.google/threat-analysis-group/trackin...	https://github.com/RedDrip7/APT_Digital_Weapon/blo...
9	Confucius	2018	https://www.trendmicro.com/en_us/research/18...	https://www.lookout.com/blog/lookout-discovers-nove...
10	Corkow	2013	https://www.welivesecurity.com/2014/02/27/...	https://www.welivesecurity.com/2014/02/27/corkow-...
11	Curious Gorge	2022	https://blog.google/threat-analysis-group/update...	?
12	Cyber Berkut	2014	https://www.kyivpost.com/article/content/may-2...	?
13	Desert Falcons	2014-2015	https://media.kasperskycontenthub.com/wp-...	https://github.com/RedDrip7/APT_Digital_Weapon/blo...
14	El Machete	2014-2022	https://blogs.blackberry.com/en/2017/03/el-...	https://github.com/RedDrip7/APT_Digital_Weapon/blo...
15	Energetic Bear	2016, 2021-2022	https://query.prod.cms.rt.microsoft.com/cms/api...	https://github.com/RedDrip7/APT_Digital_Weapon/blo...
16	Evilnum	2019-2020	https://symantec.broadcom.com/hubfs/SED-...	https://github.com/eset/malware-ioc/blob/master/...
17	Fishing Elephant	2019-2020	https://securelist.com/apt-trends-report-...	?
18	Gamaredon Group	2014-2022	https://cert.gov.ua/article/18365,https://...	https://cert.gov.ua/article/18365,https://cert.gov.ua/...
19	Hades	2018	https://securelist.com/olympic-destroyer-is-still-...	https://securelist.com/olympicdestroyer-is-here-to-tric...

1 - 20 of 54 — Go to: 1

Рисунок 3.7 – Вигляд додаткової бази даних APT_Ukraine_timeline_and_hashes

Клас APT_Ukraine створений спеціально для розгляду поточної ситуації в Україні, що стосується питання атрибуції та APTs. Окрім успадкованих атрибутів та методів від батьківського класу, він містить й інші змінні (Рисунок 3.8) й функції (Рисунок 3.9), які несуть у собі додаткову інформацію, що надає повноти для коректної оцінки стану вищезгаданих проблем і розуміння необхідності їхнього вирішення та може бути використаною для перевірки захищеності КС чи КМ як спеціалістами з кібербезпеки, так і звичайними користувачами мережі Інтернет.

```
191  #усі APT групи зі списку, що атакували кіберпростір України
192 class APT_Ukraine(APT_World):
193
194     def __init__(self, data, name):
195         super().__init__(data, name)
196         self.timeline = []
197         self.hashes = []
198         self.timeline_source = []
199         self.hashes_source = []
```

Рисунок 3.8 – Властивості класу APT_Ukraine

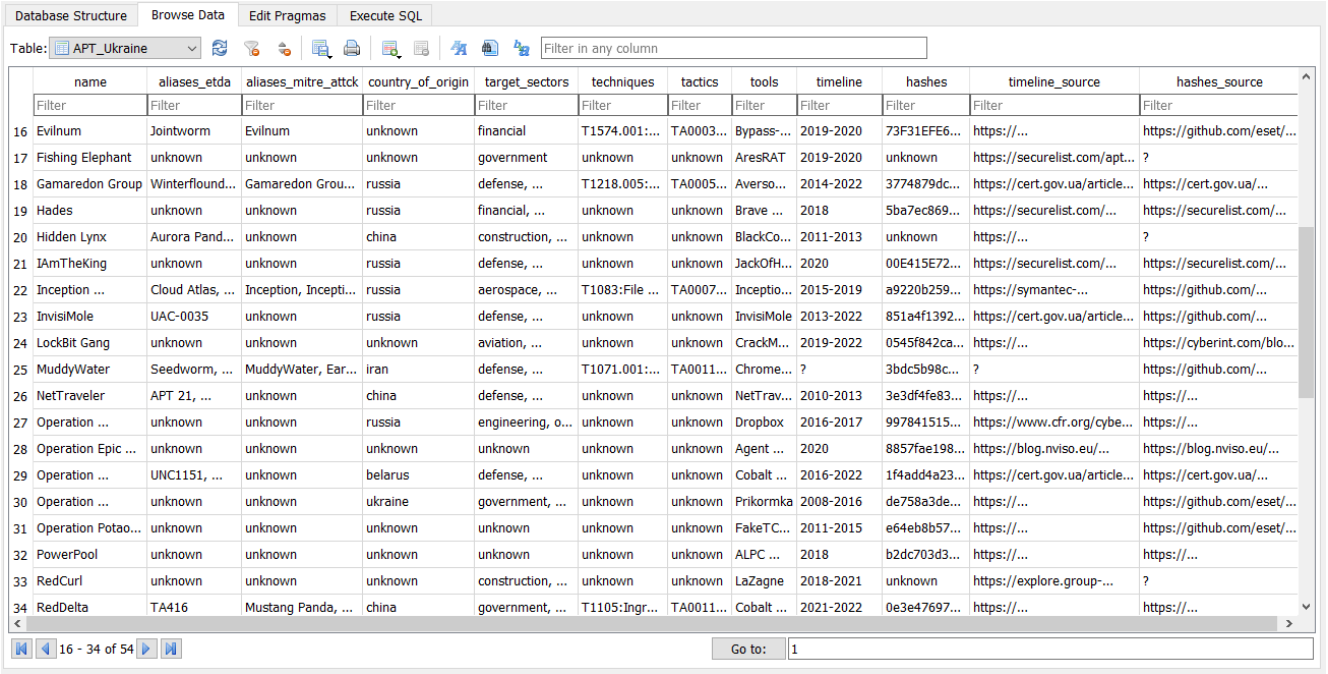
```
194     def __init__(self, data, name):
200
201     #зчитування посилань на timeline атак групи
202     def get_timeline_source(self):
207
208     #зчитування timeline
209     def get_timeline(self):
214
215     #зчитування посилань на хеші
216     def get_hashes_source(self):
221
222     #пошук усіх хешів шкідливих файлів за посиланням
223     def get_hashes_from_url(self):
246
247     #запис даних у базу даних
248     def insert_data_ukraine(self):
260
261     #запуск для отримання значень (додаткових) атрибутів класу
262     def main(self):
271
```

Рисунок 3.9 – Методи класу APT_Ukraine

Вивчимо побудову цього класу:

- Властивості:
  1) ***timeline*** – змінна, що містить перелік усіх років, коли було зафіксовано діяльність зазначеної групи;
  2) ***hashes*** – збірка хешів файлів, які були використані під час атак(и);
  3) ***timeline_source*** – список посилань на звіти, статті, збірок і т.д., що стосуються даної APT;
  4) ***hashes_source*** – аналогічний перелік посилань до попереднього, з відмінністю, що акцент ставиться на наявність IOCs у розглянутих джерелах.
- Методи:
  1) ***get_timeline_source(self)*** – функція, що відповідає за зчитування комірки бази даних APT_Ukraine_timeline_and_hashes під назвою timeline_source та запису у змінну класу з такою ж назвою;
  2) ***get_timeline(self)*** **–** аналогічно до попереднього методу, лише з урахуванням, що потрібна комірка має назву timeline;
  3) ***get_hashes_source(self)*** – функція, що забезпечує створення списку посилань на джерела, що містять хеші шкідливих файлів;

4) ***get_hashes_from_url(self)*** – за допомогою циклу for перебирає кожне з посилань у hashes_source, використовує Python бібліотеку BeautifulSoup[13] для зчитування тексту з вказаного url та, завдяки функціям бібліотеки iocextract[14], відбирає хеші з цього тексту;
5) ***insert_data_ukraine(self)*** – відповідає за запис отриманої інформації до таблички APT_Ukraine бази даних APT_Groups;
6) ***main(self)*** **–** аналогічно до однойменної функції батьківського класу викликає необхідні методи.



	name	aliases_etda	aliases_mitre_attck	country_of_origin	target_sectors	techniques	tactics	tools	timeline	hashes	timeline_source	hashes_source
16	Evilnum	Jointworm	Evilnum	unknown	financial	T1574.001:...	TA0003...	Bypass-...	2019-2020	73F31EFE6...	https://...	https://github.com/eset/...
17	Fishing Elephant	unknown	unknown	unknown	government	unknown	unknown	AresRAT	2019-2020	unknown	https://securelist.com/apt...	?
18	Gamaredon Group	Winterflound...	Gamaredon Grou...	russia	defense, ...	T1218.005:...	TA0005...	Averso...	2014-2022	3774879dc...	https://cert.gov.ua/article...	https://cert.gov.ua/...
19	Hades	unknown	unknown	russia	financial, ...	unknown	unknown	Brave ...	2018	5ba7ec869...	https://securelist.com/...	https://securelist.com/...
20	Hidden Lynx	Aurora Pand...	unknown	china	construction, ...	unknown	unknown	BlackCo...	2011-2013	unknown	https://...	?
21	IAmTheKing	unknown	unknown	russia	defense, ...	unknown	unknown	JackOfH...	2020	00E415E72...	https://securelist.com/...	https://securelist.com/...
22	Inception ...	Cloud Atlas, ...	Inception, Incepti...	russia	aerospace, ...	T1083:File ...	TA0007...	Inceptio...	2015-2019	a9220b259...	https://symantec-...	https://github.com/...
23	InvisiMole	UAC-0035	unknown	russia	defense, ...	unknown	unknown	InvisiMole	2013-2022	851a4f1392...	https://cert.gov.ua/article...	https://github.com/...
24	LockBit Gang	unknown	unknown	unknown	aviation, ...	unknown	unknown	CrackM...	2019-2022	0545f842ca...	https://...	https://cyberint.com/blo...
25	MuddyWater	Seedworm, ...	MuddyWater, Ear...	iran	defense, ...	T1071.001:...	TA0011...	Chrome...	?	3bdc5b98c...	?	https://github.com/...
26	NetTraveler	APT 21, ...	unknown	china	defense, ...	unknown	unknown	NetTrav...	2010-2013	3e3df4fe83...	https://...	https://...
27	Operation ...	unknown	unknown	russia	engineering, o...	unknown	unknown	Dropbox	2016-2017	997841515...	https://www.cfr.org/cybe...	https://...
28	Operation Epic ...	unknown	unknown	unknown	unknown	unknown	unknown	Agent ...	2020	8857fae198...	https://blog.nviso.eu/...	https://blog.nviso.eu/...
29	Operation ...	UNC1151, ...	unknown	belarus	defense, ...	unknown	unknown	Cobalt ...	2016-2022	1f4add4a23...	https://cert.gov.ua/article...	https://cert.gov.ua/...
30	Operation ...	unknown	unknown	ukraine	government, ...	unknown	unknown	Prikormka	2008-2016	de758a3de...	https://...	https://github.com/eset/...
31	Operation Potao...	unknown	unknown	unknown	unknown	unknown	unknown	FakeTC...	2011-2015	e64eb8b57...	https://...	https://github.com/eset/...
32	PowerPool	unknown	unknown	unknown	unknown	unknown	unknown	ALPC ...	2018	b2dc703d3...	https://...	https://...
33	RedCurl	unknown	unknown	unknown	construction, ...	unknown	unknown	LaZagne	2018-2021	unknown	https://explore.group-...	?
34	RedDelta	TA416	Mustang Panda, ...	china	government, ...	T1105:Ingr...	TA0011...	Cobalt ...	2021-2022	0e3e47697...	https://...	https://...

Рисунок 3.10 – Таблиця APT_Ukraine у базі даних APT_Groups

Створена база даних APT_Groups містить усю необхідну інформацію в організованому вигляді для подальшого проведення дослідження та визначення зв'язків між її елементами.

3.2.3 Візуалізація даних і їхній аналіз

Мною було вирішено об'єднати наступні два кроки запропонованої моделі CSIAC: *illustrate* – демонстрації даних у форматі ілюстрацій – графіків, для чіткого уявлення діяльності APT груп, та *analyze* – проведення їхнього аналізу.

Для побудови графіків було створено окремий клас з назвою Graph (Рисунок 3.11), який не містить жодних атрибутів, а лише методи, які представляють алгоритми формування окремих діаграм та додаткові операції для цього:

1) ***count_items(self, item_list_without_repeats, item_list)*** – забезпечує підрахунок об'єктів зі списку, де вони повторюється, за допомогою переліку, де вони існують в єдиному екземплярі, та повертає перелік з кількістю кожного об'єкта окремо;
2) ***select_all_names(self, table_name)*** – зчитує назви APTs з вказаної таблиці та повертає їх перелік;
3) ***select_all_countries_of_origin(self, table_name)*** – повертає перелік усіх країн походження APT груп (відсортований та ні) із запропонованої таблиці;
4) ***select_all_sectors(self, table_name)*** – аналогічно до попередньої функції, повертає список, що містить у собі два переліки цільових секторів: у єдиному екземплярі і просто зчитані з таблиці, назву якої було передано як параметр;
5) ***select_all_techniques(self, table_name)***, ***select_all_tactics(self, table_name)*** та ***select_all_tools(self, table_name)*** – виконуються аналогічно до попередніх двох описаних методів;
6) ***country_of_origin_graph(self, choice)*** – демонстрація залежності між кількістю APT груп та країнами їхнього походження. На вхід приймає параметр choice, що має значення ‘1’ або ‘2’ та вказує на вибір таблиці з якої необхідно обрати та обробити дані;

7) ***top_20_targeted_countries_graph(self)*** – кругова діаграма 20-и країн світу, що найбільше потерпають від кібератак APTs;
8) ***targeted_sectors_graph(self, choice)*** – шкала залежності секторів економіки від кількості здійснених на них нападів APT групами за обраною таблицею;
9) ***top_15_used_techniques_graph(self, choice)*** – демонстрація 15-и найчастіше використовуваних технік зловмисниками. Як вхідний параметр приймається назва таблиці з потрібними даними;
10) ***top_10_used_tactics_graph(self, choice)*** – аналогічна до вищеописаної функція, яка ілюструє 10-ть найбільше застосовуваних тактик при кібератаці;
11) ***top_used_tools_graph(self, choice)*** – кругова діаграма інструментів, які є найпопулярнішими серед APTs світу або України;
12) ***timeline_graph(self)*** – графік залежності часу (конкретно років) здійснення атак(и) та груп, діяльність яких була зафіксована в кіберпросторі України;
13) ***sectors_vs_tactics(self, choice)*** – ілюструє залежність між цільовими секторами та найчастіше використовуваними для нападу тактиками;
14) ***sectors_vs_countries_of_origin(self, choice)*** – графік залежності секторів економіки, що атакують APTs, та країнами походження груп.

```
272  #графіки залежностей
273 class Graphs:
274
275     def count_items(self, item_list_without_repeats, item_list):
284
285     #перелік усіх APT груп з таблички
286     def select_all_names(self, table_name):
293
294     #вибір усіх країн походження APT груп з таблички
295     def select_all_countries_of_origin(self, table_name):
303
304     #вибір усіх секторів з таблички
305     def select_all_sectors(self, table_name):
316
317     #вибір усіх технік з табличці
318     def select_all_techniques(self, table_name):
329
330     #вибір усіх тактик з табличці
331     def select_all_tactics(self, table_name):
342
343     #вибір усіх інструментів з табличці
344     def select_all_tools(self, table_name):
355
356     #демонстрація шкали країн-походження APT груп
357     def country_of_origin_graph(self, choice):
393
394     #демонстрація 20 країн, які найбільше потрапляють під атаки APT груп
395     def top_20_targeted_countries_graph(self):
430
431     #демонстрація шкали секторів, які атакують APT групи
432     def targeted_sectors_graph(self, choice):
469
470     #демонстрація 15 технік, які найбільше використовують APT групи
471     def top_15_used_techniques_graph(self, choice):
501
502     #демонстрація 10 тактик, які найбільше використовують APT групи
503     def top_10_used_tactics_graph(self, choice):
537
538     #демонстрація топ програм, які найбільше використовують APT групи
539     def top_used_tools_graph(self, choice):
570
571     #демонстрація часу атак APT груп на різноманітні об'єкти та структури України
572     def timeline_graph(self):
615
616     #демонстрація залежності між секторами та використовуваними тактиками
617     def sectors_vs_tactics(self, choice):
687
688     #демонстрація залежності між секторами та країнами походження APT груп
689     def sectors_vs_countries_of_origin(self, choice):
750
```

Рисунок 3.11 – Вигляд класу Graph

Розглянувши структуру класу, що дає можливість побудувати необхідні графіки залежностей, потрібно дослідити, що візуалізує кожна з діаграм. Як вже згадувалося раніше ті методи описаного класу, що приймають параметр *choice*, створюють ілюстрації даних окремо для світу та окремо для України. Графік, що демонструє час здійснення атак(и) APT групами будується з інформації, що записана винятково до таблиці APT_Ukraine. Відповідно діаграма, яка репрезентує 20-ть країн, на території яких найчастіше реєструють кібернапади APTs, створена за допомогою даних, отриманих з APT_World.

Щоб зрозуміти масштаб небезпеки, яку нині викликає існування APT груп у кіберсередовищі, потрібно оглянути, яким територіям (Рисунок 3.12) та сферам індустрії (Рисунок 3.13) для проведення атак вони віддають перевагу.

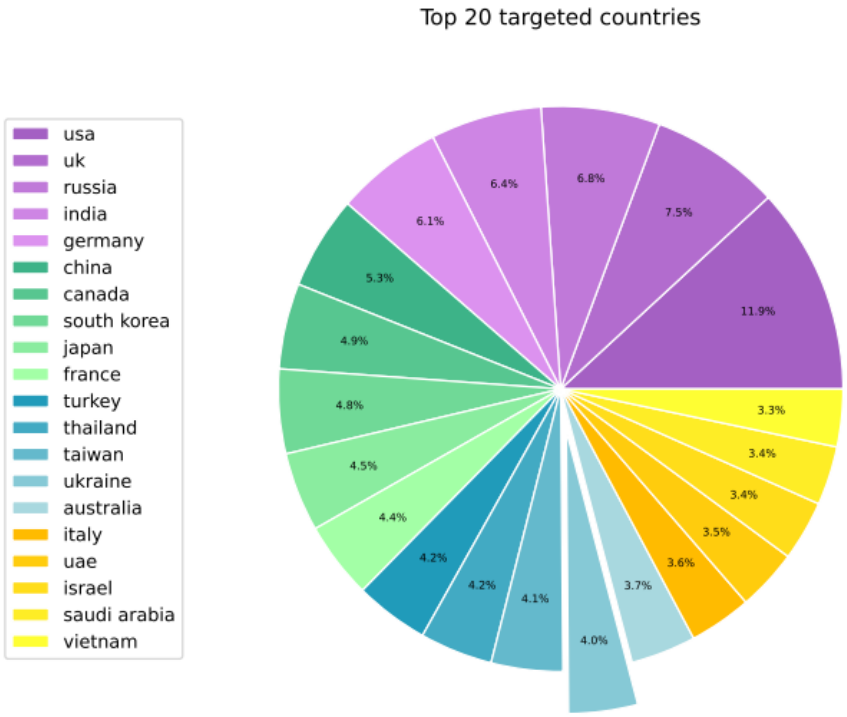


Рисунок 3.12 – Діаграма топ-20 цільових країн, що APT групи атакують

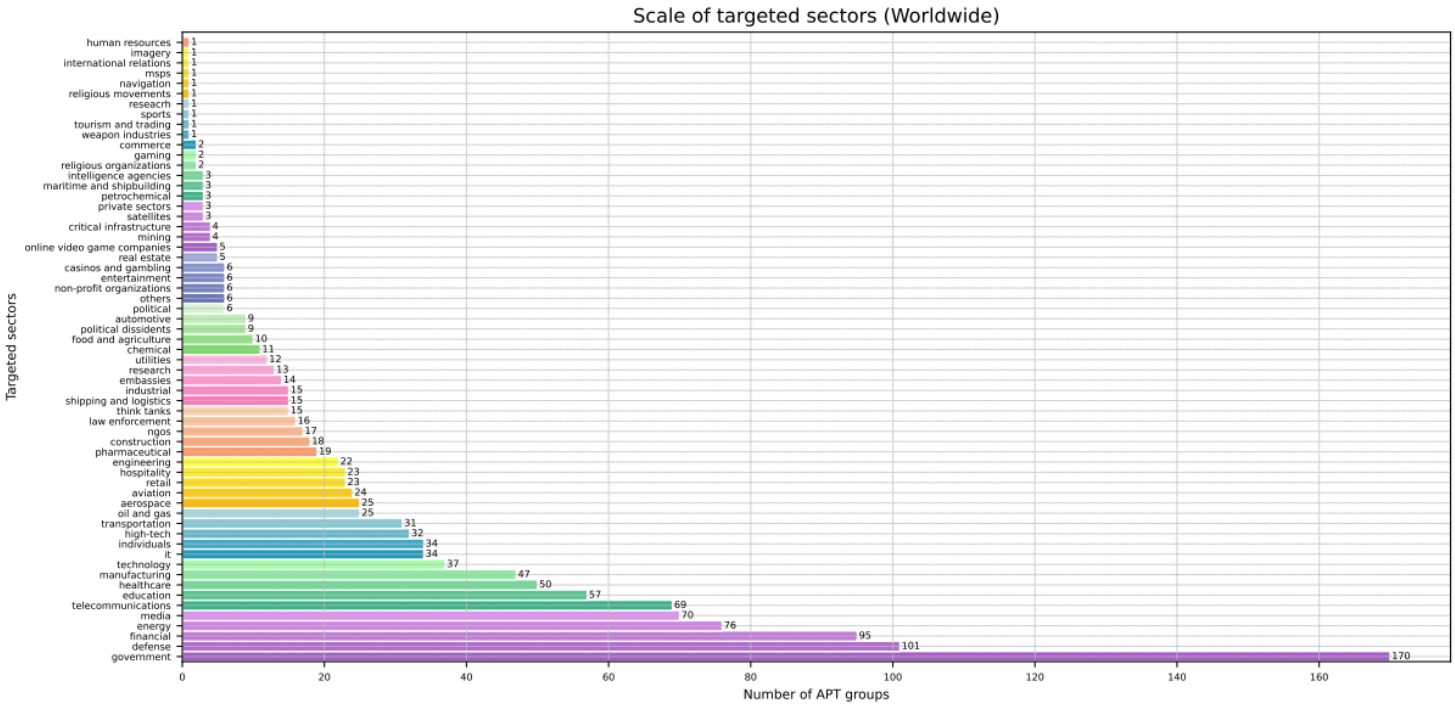


Рисунок 3.13 – Шкала залежності цільових секторів від кількості ATPs, що їх атакують/атакували (світ)

Проаналізуємо продемонстровані графіки у сукупності. Як ми бачимо, Сполучені Штати Америки, Велика Британія, росія, Індія та Німеччина є державами з найбільшим відсотком здійснених на них кібернападів, зокрема Україна посідає 14-е місце у списку. З боку секторів економіки, що найчастіше потрапляють у поле діяльності різноманітних APT груп, з великим відривом на першому місці розміщується державний, який складається з місцевих і центральних апаратів управління та відповідає за забезпечення продовольства країни, фінансування урядових організацій та установ і т.п., далі безпеки й оборони, головна задача якого пов'язана з виробництво та підтримкою військових організацій, спорядження і т.д., фінансовий, зосереджений на наданні фінансових послуг, енергетичний, що конкретизується на видобутку і постачанні різних видів енергії, та медійний, який займається публікуванням, створенням та поширенням інформації.

З урахуванням геополітичної ситуації та світових тенденцій у всіх сферах суспільства можна зрозуміти чому виник такий результат, візуалізований у діаграмах, описаних вище. Атакуючи державний сектор, APT групи дізнаються державні таємниці, отримують інформацію про конкретних високопосадовців, розробляють платформу для проведення диверсій і подібне в інтересах країни-агресорки. Здійснюючи кібернапади у медіапросторі, вони можуть впливати на думку суспільства та корегувати її напрямок у своїх цілях. У зв'язку з науково-технічним прогресом питання, що вирішує сфера енергетики, є надважливими: той хто володіє енергією, володіє світом. Її застосування існує у будь-якій сфері життя людини, як наприклад: подача гарячої води до будинку, робота апарату штучної вентиляції легень, створення побутової техніки, транспортування і т.д.

Зважаючи на передові сектори економіки перелічених країн можна зробити висновки щодо головних цілей APT груп, що їх атакують. Основною ідеєю, що об'єднує їх усіх, є жага до влади та грошей.

Аналогічно можна дослідити кореляцію країн походження APTs та сфер, що групи атакують (Рисунок 3.14-3.15).

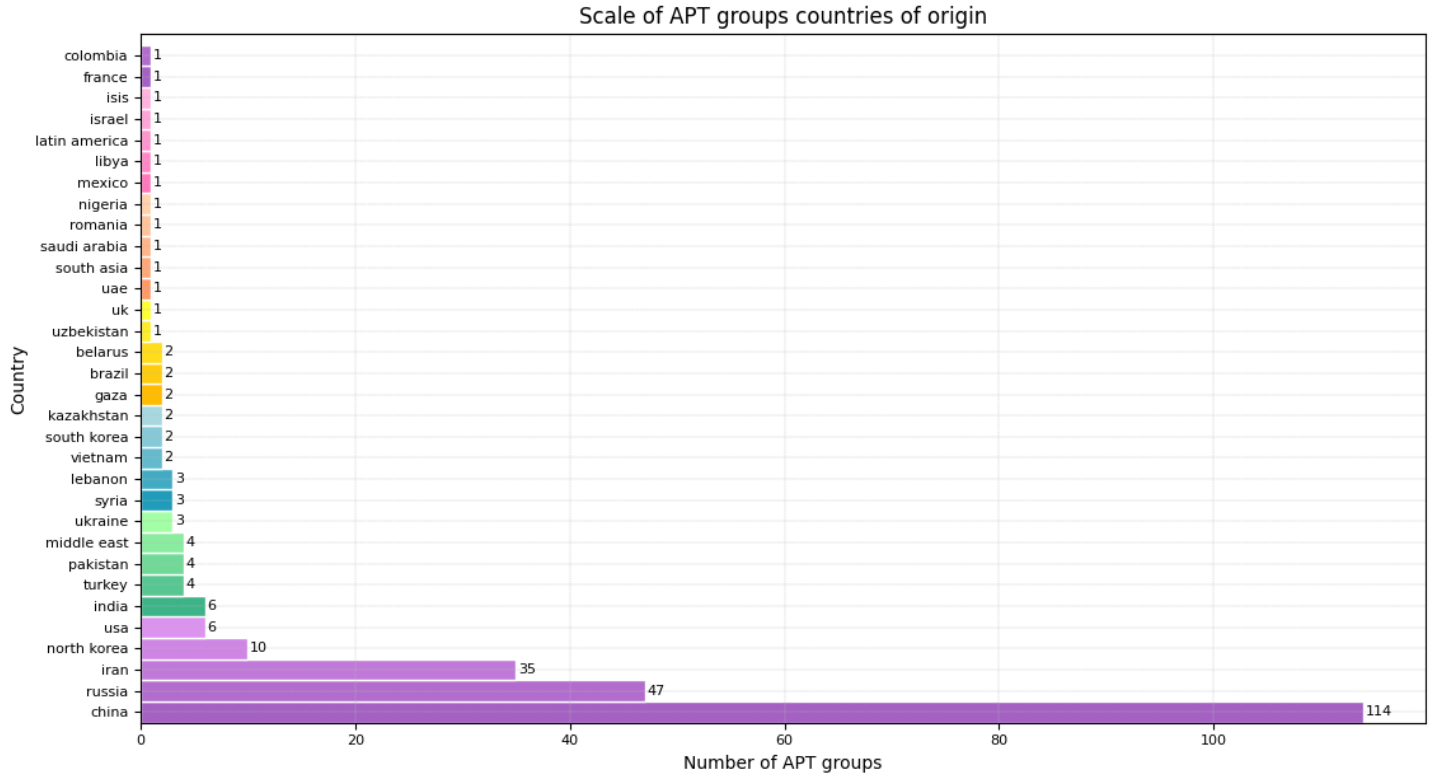


Рисунок 3.14 – Шкала країн походження APT груп та їхня кількість (світ)

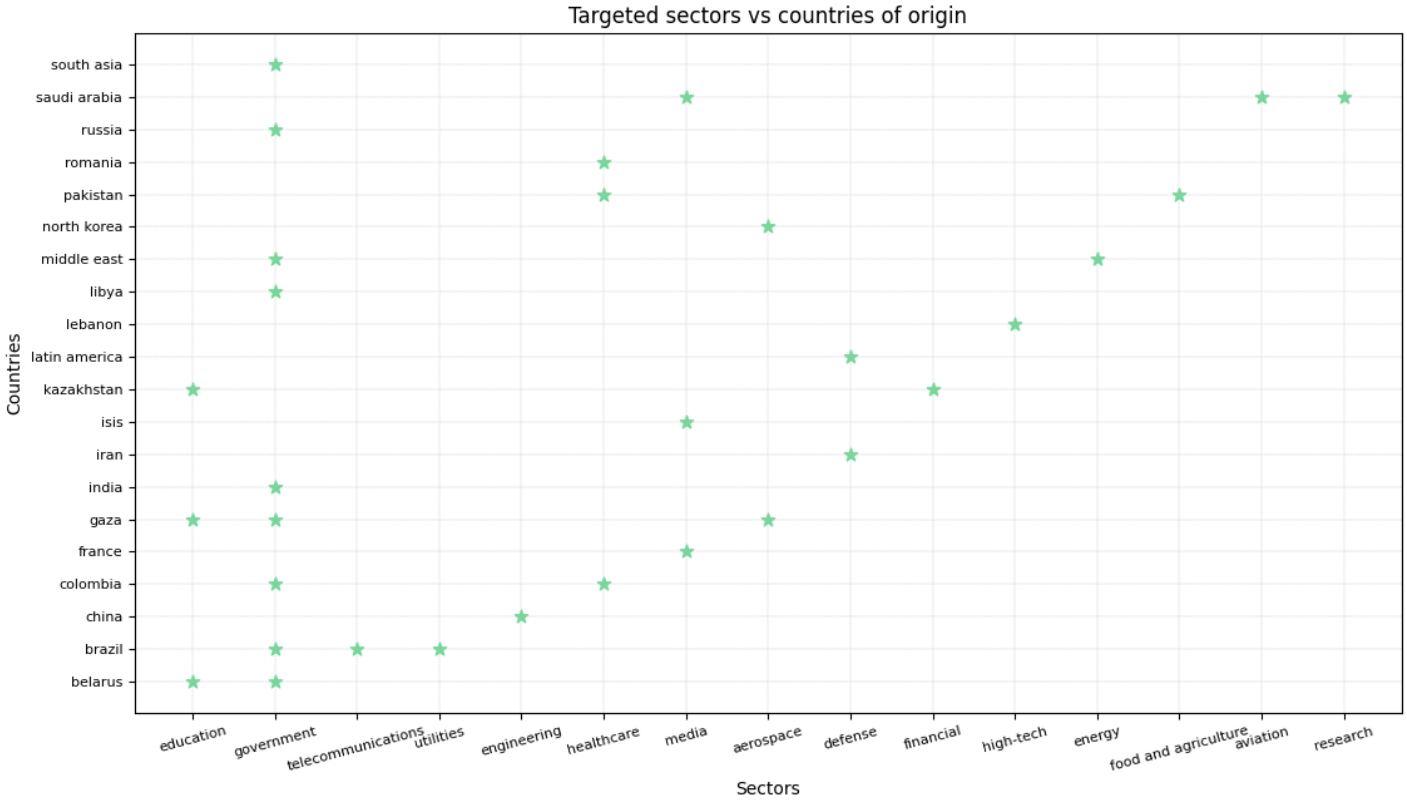


Рисунок 3.15 – Залежність цільових секторів від країн походження APT груп (світ)

З цих графіків можна визначити, чим мотивовані APT групи, об'єкти яких секторів вони атакують найбільше. Беручи до уваги загальний розвиток держави та її інтереси на міжнародному ринку, ми можемо надати певну характеристику зловмисників, які на неї працюють.

Наприклад: Китай – є батьківщиною найбільшої кількості APT груп, які найчастіше атакують сектор інженерії, до якої входять різноманітні галузі промисловості, зокрема електроніка, обладнання для транспортування і виробничого устаткування та будівництво. Китайська компанія China State Construction Engineering Corporation (CSCEC) є “найбільшою будівельною компанією світу станом на 2019 рік” [15]. Склавши ці два твердження разом, ми можемо зробити припущення, що APTs Китаю атакують своїх конкурентів, щоб залишатися на передовій позиції на ринку.

Також варто звернути увагу на найбільш поширені застосовані тактики (Рисунок 3.16) та техніки (Рисунок 3.17) APTs та детально їх розглянути [16, 17].

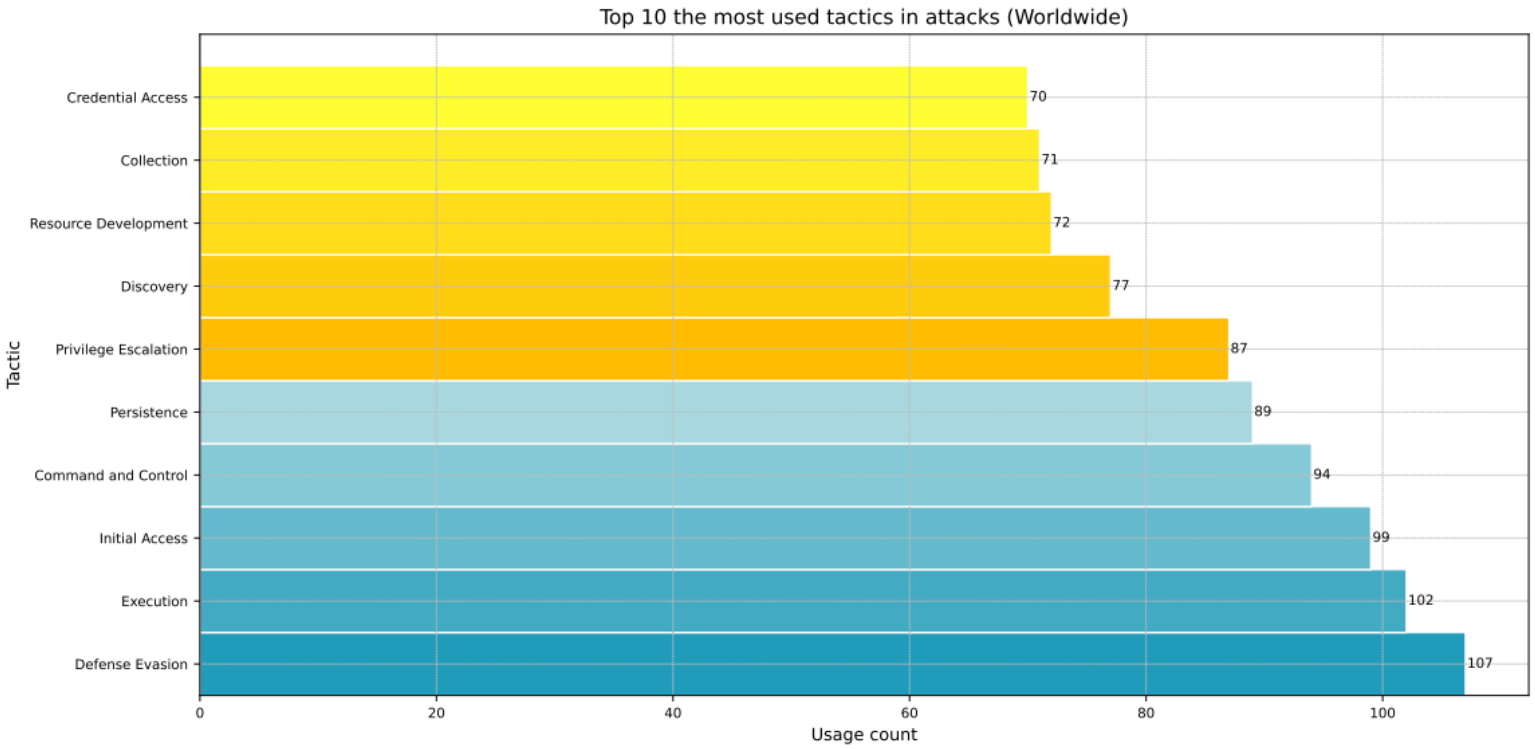


Рисунок 3.16 – Топ-10 найчастіше використовуваних тактик для проведення атак (світ)

Розберемо тактики зловмисників – вони надають відповідь на питання в чому полягає виконання певної дії, якою є ціль порушника:

1. **Defense Evasion** (укр. обхід захисту) – відповідає за перелік усіх наявних технік, які забезпечують зловмисникам можливість залишатися непоміченими під час несанкціонованого доступу до КС чи КМ;
2. **Execution** (укр. виконання) – надає список прийомів, які використовуються безпосередньо для запуску ШПЗ, виконуваних файлів та різноманітних команд, як у локальній, так і віддаленій системі керування.
3. **Initial Access** (укр. отримання початкового доступу) – тактика, яка описує першочергову задачу APT: увійти до КС чи КМ та зафіксуватися в ній для початку проведення кібероперації.
4. **Command and Control** (укр. командування і керування) – забезпечує техніки, які надають можливість безпосереднього керування КС чи КМ обраної жертви.
5. **Persistence** (укр. закріплення) **–** описує усі можливі техніки, які допомагають порушникам закріпитися у КС чи КМ, незважаючи на можливі переривання зв'язку.
6. **Privilege Escalation** (укр. ескалація привілеїв) – тактика, яка надає список методів та способів для отримання зловмисниками дозволів вищого рівня системи.
7. **Discovery** (укр. виявлення) – головне завдання полягає у зібранні якомога більшої кількості інформації про систему, в яку зайшов порушник, для розуміння її структури та, які засоби необхідно використати, для успішної атаки.
8. **Resource Development** (укр. розвиток ресурсів) **–** є збіркою різноманітних технік, які використовуються для отримання (купівлі чи викрадення вже існуючих, створення власних) ресурсів, що дозволять підтримку кібератаки протягом довгого часу.

9. **Collection** (укр. збір даних) – перелік методів, які надають змогу зловмисникам зібрати цільову інформацію, що зберігається у системі.
10. **Credential Access** (укр. отримання облікових даних) – відповідає за ті техніки, які дають зловмисникам змогу викрасти облікові записи: паролі та імена користувачів.

APT групи завжди використовують сукупність тактик під час здійснення кібероперації. Поодинці вони не мають такої ефективності як у комплексі. Наприклад: увійшовши до системи і закріпившись у ній (*Initial Access*) та відразу почати виконання ШПЗ (*Execution*), може призвести до миттєвого виявлення зловмисника у КС або КМ, тому передусім необхідно обійти її захист (*Defense Evasion*); поєднання фіксації у системі (*Persistence*) та отримання облікових даних (*Credential Access*) дозволить зібрати нову інформацію облікових записів, навіть після зміни паролю чи скидання налаштувань системи.

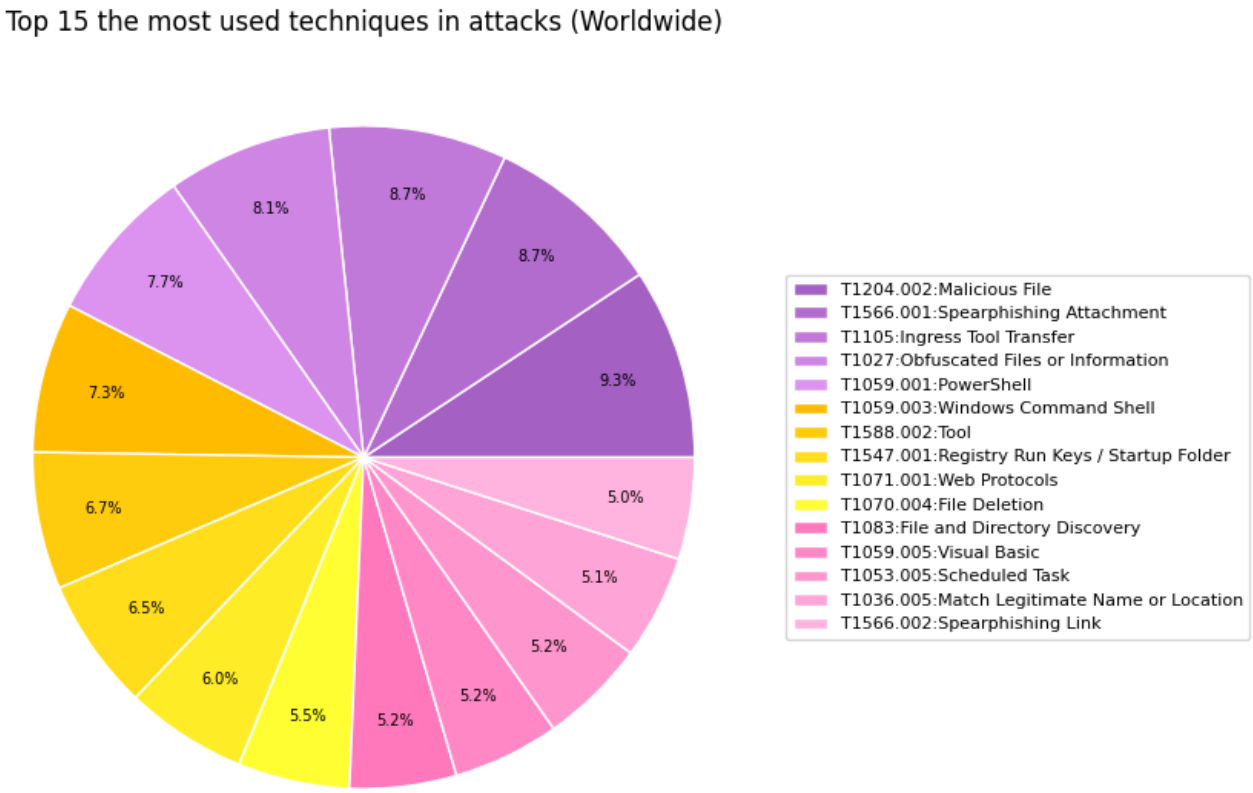


Рисунок 3.17 – Топ-20 найбільше застосованих технік APT групами (світ)

Тепер зупинимося на детальному розгляді технік, які розкривають суть окремої тактики, а саме: надають пояснення як була виконана поставлена зловмисниками ціль:

1. **Malicious File** (укр. шкідливий файл) – підтехніка User Execution (укр. виконання користувача), яка вказує на те, що зловмисник покладається на користувача: він сам відкриє/запустить шкідливий файл.
2. **Spearphishing Attachment** (укр. цільовий фішинг зі шкідливим вкладенням) – є підвидом техніки Phishing (укр. фішинг), який описує спосіб отримання доступ до КС чи КМ за допомогою електронних листів зі шкідливим прикріпленим файлом, які надсилаються заздалегідь визначеним особам.
3. **Ingress Tool Transfer** – техніка, яка відповідає за надання можливості порушникам копіювання та передачі різних файлових об’єктів та інструментів від їхньої робочої системи до КС або КМ жертви.
4. **Obfuscated Files or Information** (укр. обфусковані файли або інформація) – файли та інформація можуть бути закодованими/зашифрованими/і т.п. зловмисниками, щоб їх було важче виявити антивірусом чи іншим ПЗ для детектування небажаних змін у системі.
5. **PowerShell** – підтехніка Command and Scripting Interpreter (укр. інтерпретатор команд і сценаріїв), яка відповідає за використання зловмисниками можливостей PowerShell.
6. **Windows Command Shell** (укр. командна оболонка Windows) – аналогічно до техніки описаної вище є частиною сім’ї технік Command and Scripting Interpreter, та описує застосування *cmd* для керування/виконання порушниками різноманітних файлів КС чи КМ.
7. **Tool** (укр. інструмент) – підтехніка Obtain Capabilities (укр. отримання можливостей), яка зазначає, що інструментарій зловмисників може складатися з різних знарядь атаки (куплених, створених, викрадених).

8. **Registry Run Keys / Startup Folder** (укр. ключі запуску реєстру / папка запуску) – підтехніка Boot or Logon Autostart Execution (укр. автозапуск під час завантаження або входу в систему), яка описує можливість запуску шкідливого виконуваного файлу користувачем при вході у систему, якщо зловмисник додасть його до папки автозапуску або створить посилання на нього за допомогою ключа запуску реєстру.
9. **Web Protocols** (укр. вебпротоколи) – підтехніка Application Layer Protocol (укр. протокол прикладного рівня). Використовується порушниками у цілях запобігання розкриття присутності у системі та підтримки з'єднання з нею за допомогою протоколів прикладного рівня.
10. **File Deletion** (укр. видалення файлу) – підтехніка Indicator Removal on Host (укр. видалення індикатору на хості). Зловмисники видаляє файли, які були ним створені або передані до КС чи КМ.
11. **File and Directory Discovery** (укр. виявлення файлів та каталогів) – застосовується для двох основних задач: пошук певної інформації або перерахування (enumeration) директорій та файлів у системі жертви.
12. **Visual Basic** – підтехніка згаданої вище Command and Scripting Interpreter, яка описує використання зловмисниками мови програмування Visual Basic у цілях успішного проведення атаки.
13. **Scheduled Task** (укр. заплановане завдання) – підтехніка Scheduled Task/Job (укр. заплановане завдання), яке базується на виконанні ШПЗ чи шкідливого коду на запланований порушником час за допомогою зловмисного використання можливостей Windows Task Scheduler.
14. **Match Legitimate Name or Location** (укр. збіг із законним ім'ям або місцезнаходженням) – підтехніка Masquerading (укр. маскування). Зловмисники приховують ШПЗ шляхом перейменування його у назву легітимного ПЗ або переміщувати у довірений каталог системи.
15. **Spearphishing Link** (укр. цільовий фішинг зі шкідливим посиланням) – підтехінка раніше згадуваної Phishing, яка полягає у надсиланні

електронних листів зі шкідливим посиланням жертвам та допомагає запобігти виявленню при перевірці вкладень у повідомленнях.

Зважаючи на методи і засоби, що найчастіше використовують зловмисники для успішної реалізації кібероперацій, ми можемо підтримати рівень захисту нашої КС чи КМ завдяки створенню додаткових шарів безпеки. Наприклад: одними з найпопулярніших технік на сьогодні є Spearphishing Attachment/Link, щоб уникнути імплементації атаки, можна встановити сканер, що переглядає електронні повідомлення на вміст шкідливих файлів і посилань, завжди перевіряти адресанта та відкривати листи лише від знайомих/надійних джерел. Якщо це корпоративна мережа, то потрібно підтримувати технології і процедури захисту в актуальному стані. Але загрозу не завжди можна виявити заздалегідь, у такому випадку необхідно пам'ятати про: шифрування/кодування конфіденційної і важливої інформації, збереження резервних копій системи/чи окремих її компонентів, увімкнення багатофакторної автентифікації і т.д.

Цікавою також є кореляція між секторами, щоб були атаковані, та тактик, які було застосовано найбільше (Рисунок 3.18).

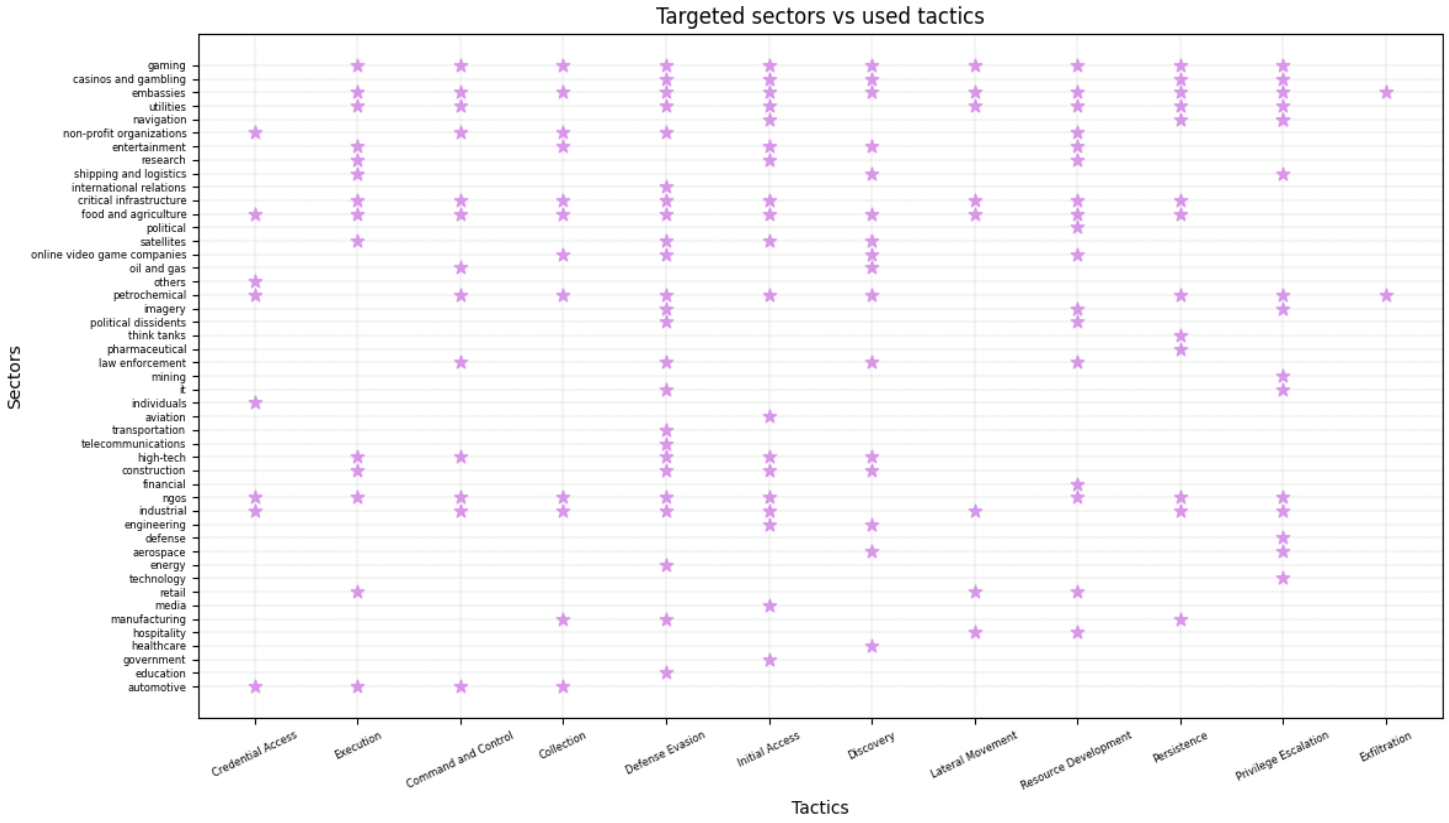


Рисунок 3.18 – Графік залежності використаних тактик від цільових секторів

Сконцентруємо нашу увагу на тих APT групах, що здійснювали кібернапади на різноманітні об'єкти, що розташовані на території України і проаналізуємо країни їхнього походження (Рисунок 3.19), рейтинг цільових секторів (Рисунок 3.20) та їх у комплексі (Рисунок 3.21).

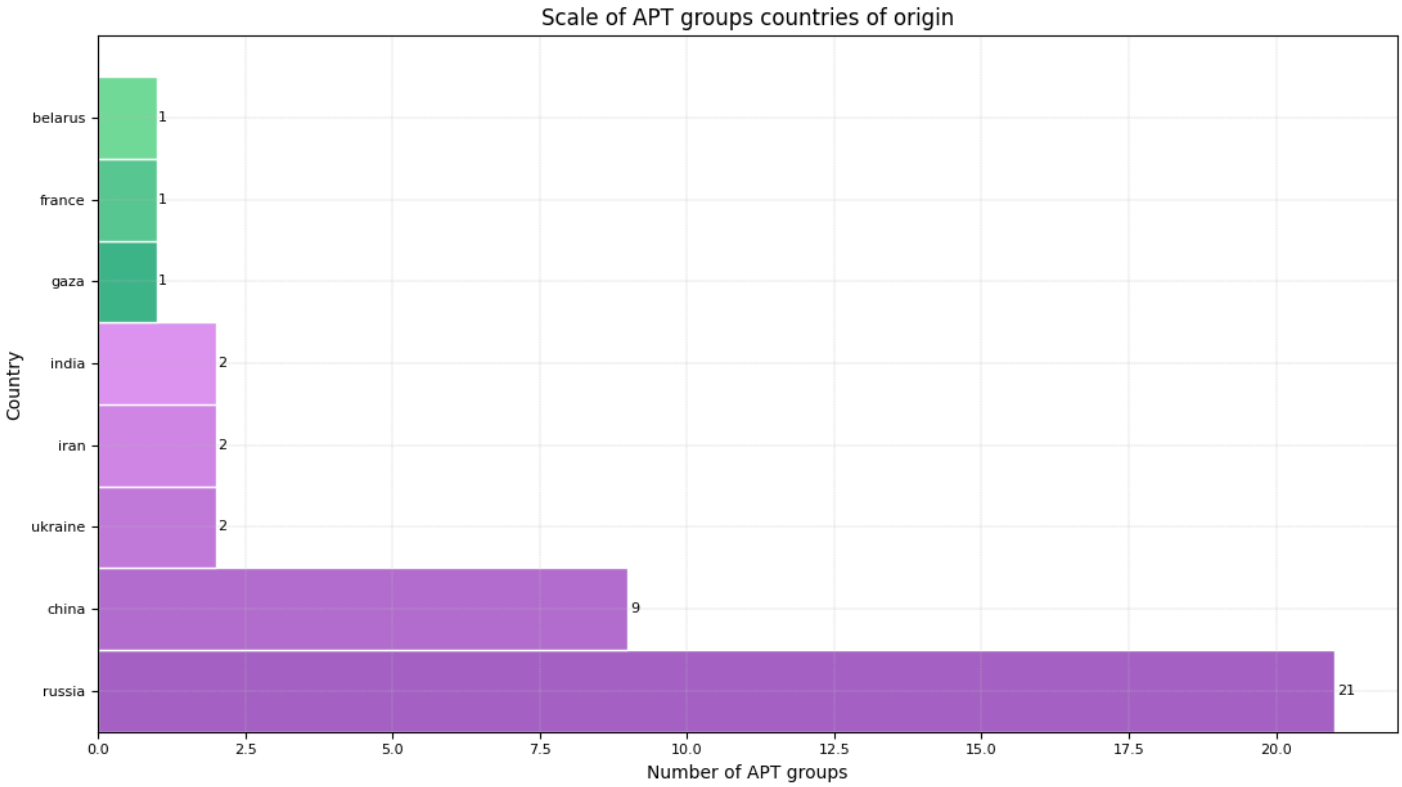


Рисунок 3.19 – Шкала кількості атакуючих APTs кіберпростір України

Країн, що наймають APTs для виконання кібератак на інфраструктуру, логістику та суверенітет України, є небагато, порівняно з загальною чисельністю держав у світі, проте варто звернути увагу на кількість самих груп, що походять від них. З великим відривом лідерську позицію займають APTs росії (21 група), на другому місці – спонсоровані Китаєм (9 груп), далі – Україною, Іраном, Індією з кількістю 2, Смугою Гази, Францією та Білоруссю з кількістю 1.

Оглядаючи шкалу атакованих секторів економіки України стає зрозуміло, що розміщення за рейтингом більшості з них (особливо перші 3-є) співпадає з ситуацією у світі. Цікаво відзначити, що кібернапади на сектор під назвою *ngos* (non-governmental organizations або не державні організації) займають майже половину (7 з 17) з загальної кількості атак на нього.

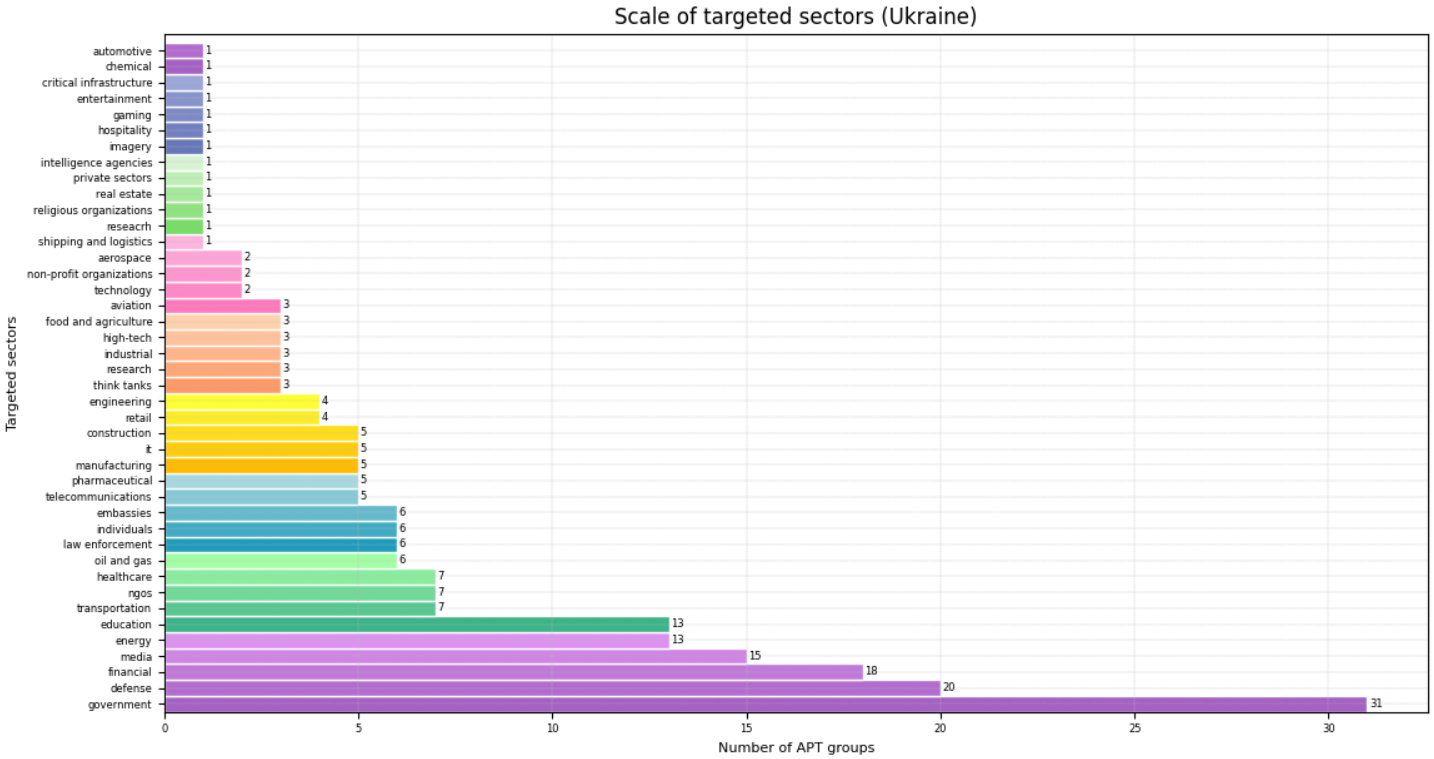


Рисунок 3.20 – Графік, що демонструє залежність секторів від кількості APTs, що їх атакують (Україна)

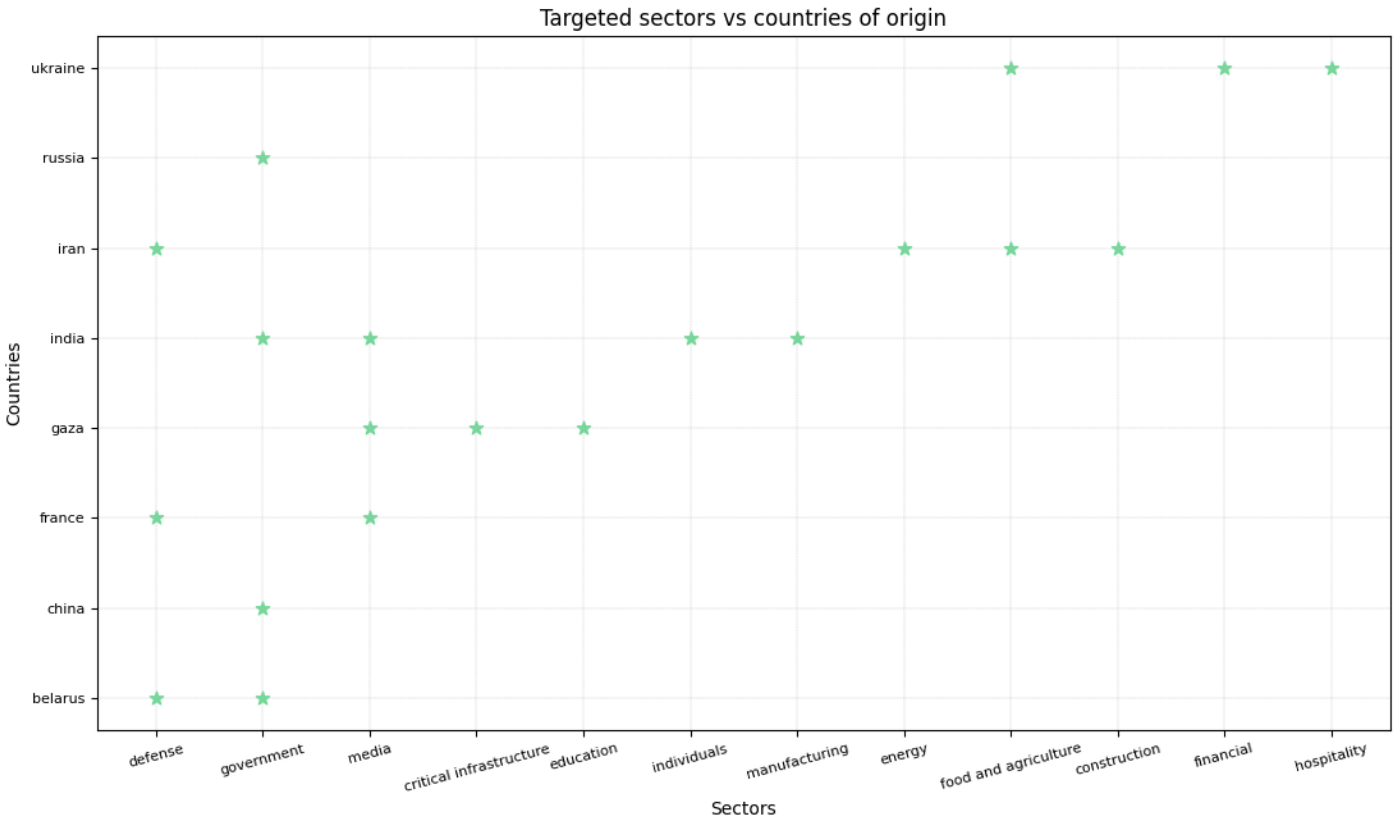


Рисунок 3.21 – Графік, що демонструє залежність цільових секторів від країн походження APTs (Україна)

З діаграми, ілюстрованої вище, ми бачимо, яка кореляція виникає між країнами-замовниками APT груп і сферами економіки, на які було завдано нападів. Зокрема, варто зауважити, що деякі території мають відразу кілька позначень – це означає, що найбільша кількість здійснених атак на певні сектори є однаковою для кількох з них одночасно. Наприклад: APTs росії та Китаю зацікавлені найбільше у державному секторі України, у той час, як Іран спрямовує свій інтерес рівномірно на сфери енергетики, будівництва, оборони і харчування та сільського господарства.

Тактики (Рисунок 3.22) і техніки (Рисунок 3.23), які застосовують зловмисники в операціях, проведених у кіберпросторі України, у своїй більшості збігаються зі світовими тенденціями, проте суттєво відрізняються за рейтингом використання.

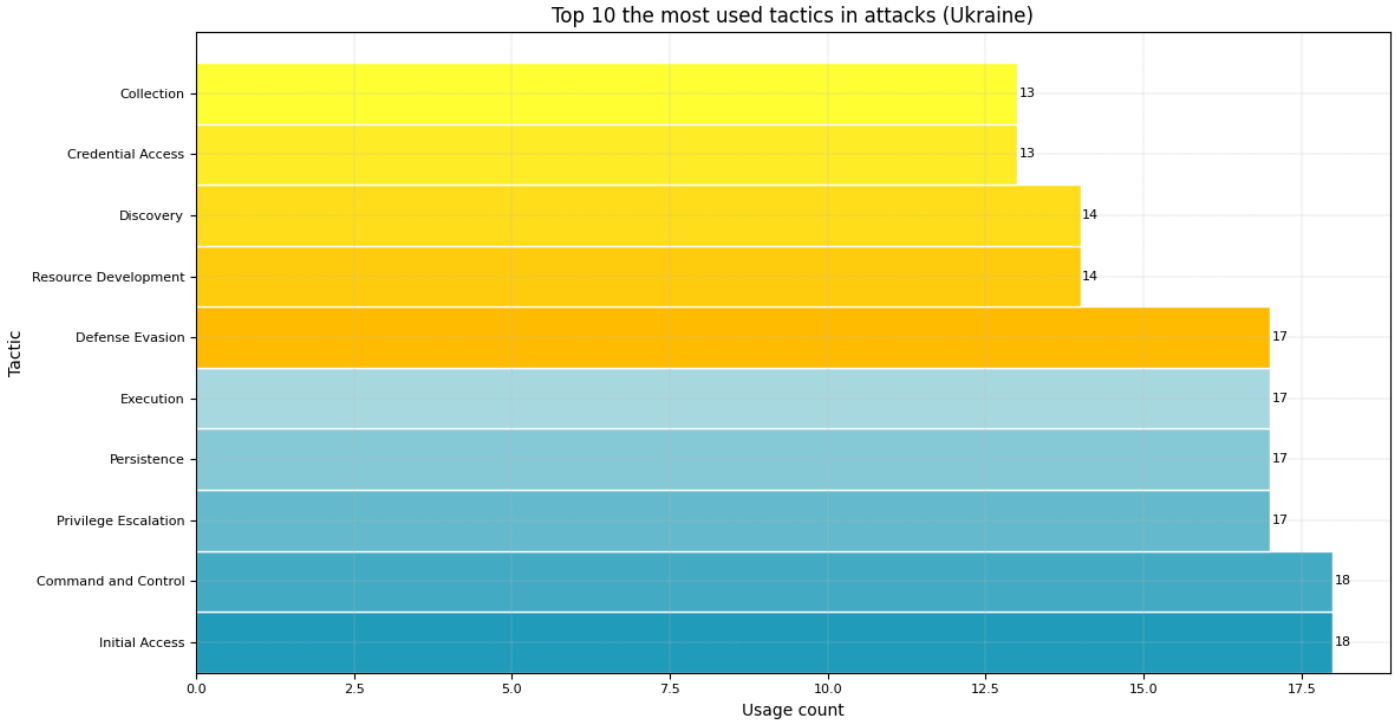


Рисунок 3.22 – Шкала застосованих тактик при кібератаках APTs (Україна)

Як можна спостерігати на графіку вище, головними завданнями APT груп є вхід до КС чи КМ, можливість безпосереднього виконання та керування машиною-жертвою, отримання високих привілеїв, якомога довше закріплення у системі і т.д.

Можна припустити, що кібератаки, спрямовані насамперед на отримання конфіденційної інформації і довготривалої розвідки того чи іншого об'єкту.

Розглядаючи топ-15 найчастіше використовуваних технік (Рисунок 3.23), можна підмітити, що майже всі з них сходяться зі світовими. Одним зі способів, який ще не був описаний, є ***Malicious Link*** (укр. шкідливе посилання) – підтехніка User Execution, яка аналогічно до Malicious File, означає покладання зловмисника на дії користувача, єдина різниця полягає у відкритті шкідливого *посилання*, що призведе до виконання шкідливого *файлу/коду*.

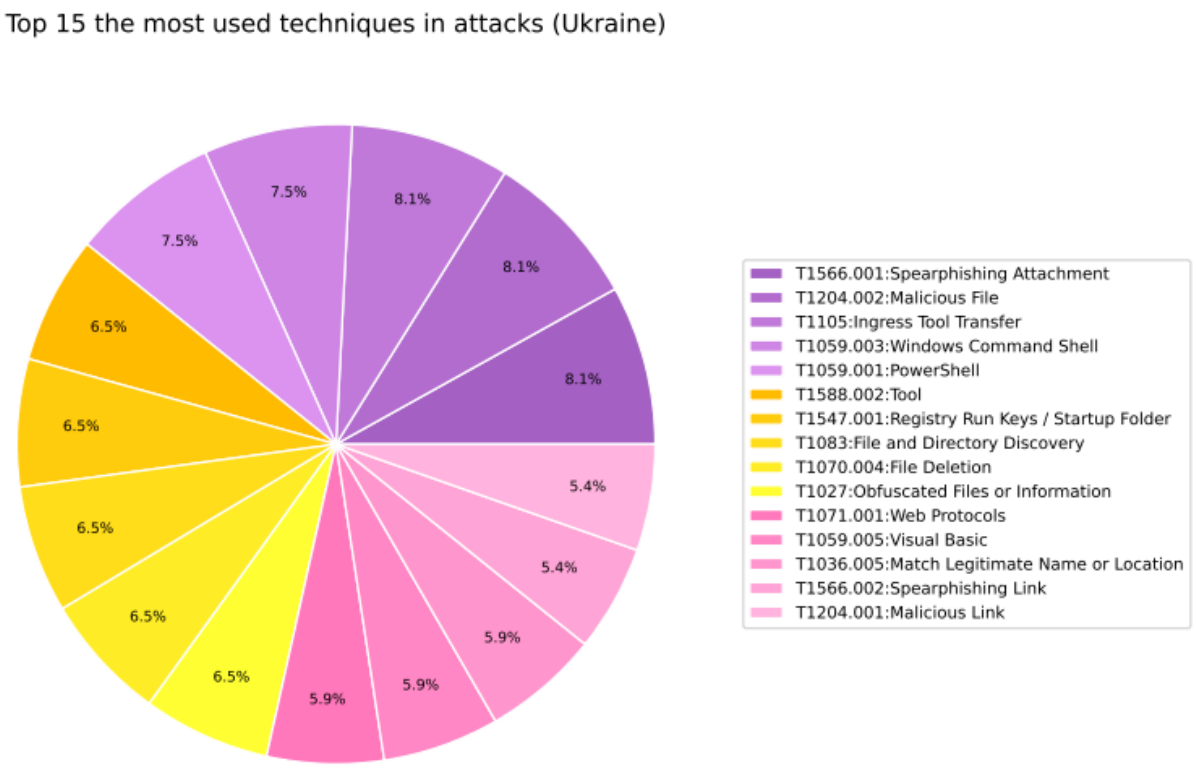


Рисунок 3.23 – Топ-15 найчастіше використовуваних технік APTs у кіберпросторі України

Аналізуючи кореляцію між сферами, на які було здійснено атаки, і тактиками, що були для цього використані (Рисунок 3.24), можна помітити, що для всіх секторів одними з найпопулярніших методів є Command and Control, Initial Access, Defense Evasion, не у значенні загальної кількості використання, а для кожного сектору окремо.

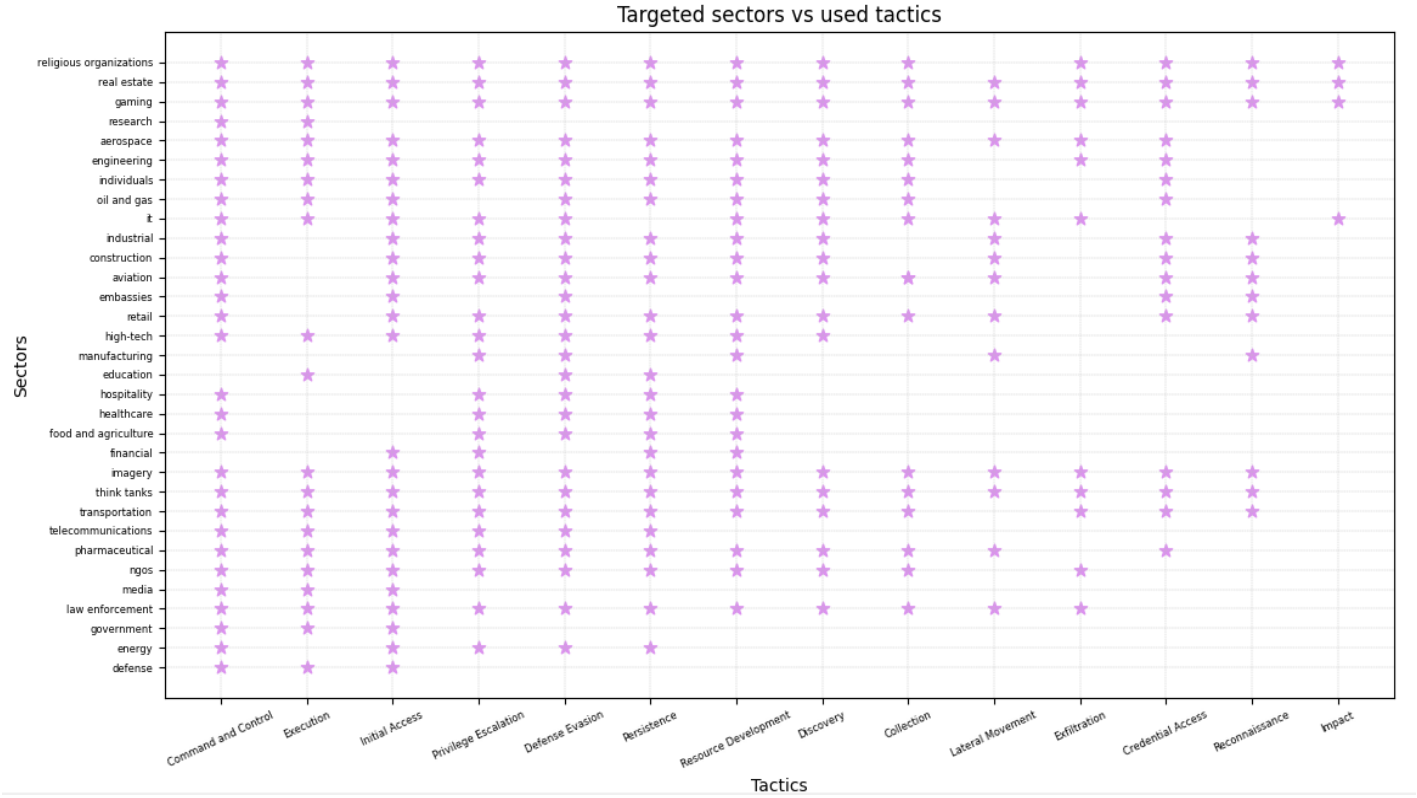


Рисунок 3.24 – Співвідношення атакованих секторів та найчастіше застосованих для них тактик (Україна)

Важливо розглядати не лише способи і підходи реалізації кіберінцидентів, а й засоби, які для цього використовують, щоб мати розуміння, які методи і процедури захисту потрібно імплементувати.

Порівняємо найпопулярніші інструменти серед APT груп світу (Рисунок 3.25) та України (Рисунок 3.26) (у значенні територій, що вони атакують), які вони застосовують при проведенні кібероперацій.

Перші 6 серед інструментів, які використовуються в атаках в Україні, також є у переліку для світу, проте інші 4 не користуються такою популярністю. Розглянемо детальніше інструментарій APTs для кіберпростору України:

1. ***Mimikatz*** – є програмою з відкритим кодом. Головними завданнями, що вона виконує, є перегляд та зберігання облікових записів, зокрема у вигляді звичайного тексту.
2. ***Living off the Land*** – колективна назва для різноманітних кібератак, під час яких APTs використовують наявні функції у системі та законне ПЗ.

3. ***Cobalt Strike*** – один із виду комерційного ПЗ, яке використовується для тестувань на проникнення (*penetration testing*).
4. ***PsExec*** – легітимний продукт компанії Microsoft, що був створений для можливості виконання процесів на віддалених системах за допомогою командного рядка.
5. ***LaZagne*** – аналогічно до Mimikatz є програмою з відкритим кодом. Зловмисники використовують її для отримання (відновлення) паролів з кінцевих точок.
6. ***PlugX*** – є трояном, що імплементується для отримання віддаленого доступу та контролю КС чи КМ.
7. ***Meterpreter*** – пейлоад (англ. *payload* – корисне навантаження) проєкту Metasploit, що використовує DLL ін'єкції, які за допомогою динамічних бібліотек процесів дозволяють в їхньому адресному просторі запуск ШПЗ/коду. APTs користуються ним як програмною оболонкою для аналізу КС або КМ жертви.
8. ***Impacket*** – надана у вільному доступі колекція класів Python для аналізу, створення і т.п. пакетів мережевих протоколів.
9. ***FlawedAmmyy*** – троян, що використовується з ціллю отримання віддаленого доступу за допомогою встановлення бекдору (англ. backdoor).
10. ***EmpireProject*** – інструмент, що застосовується зловмисниками вже після входу в систему та відкриття сесії PowerShell для віддаленого керування та надається вільному доступі.

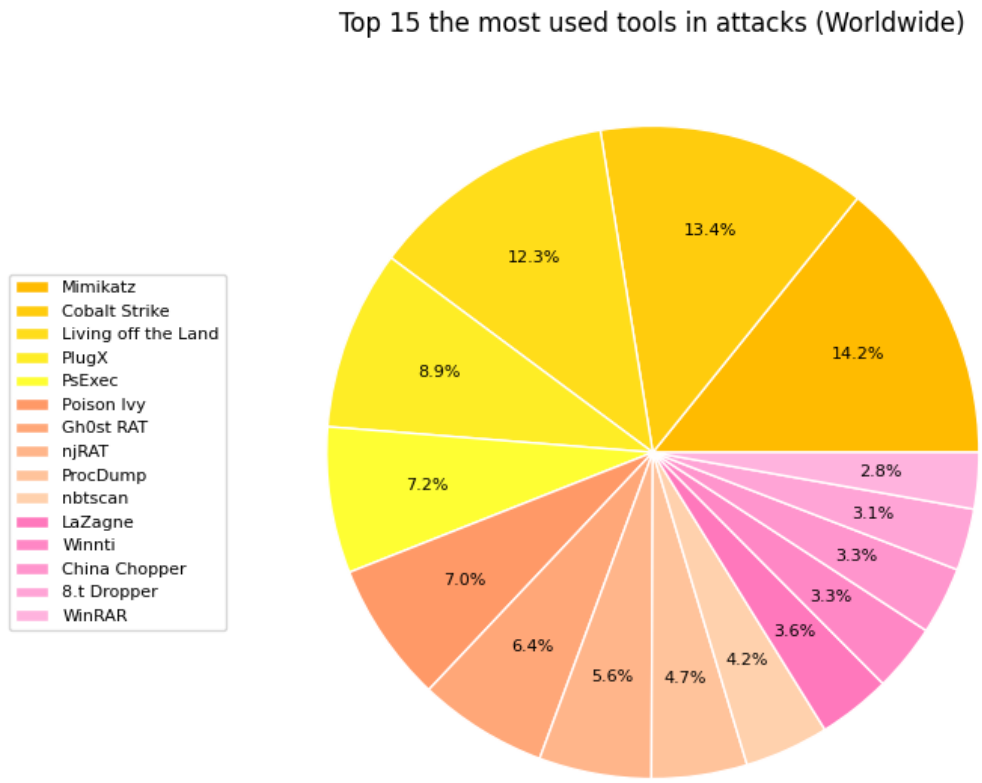


Рисунок 3.25 – Топ-15 найбільше використовуваних інструментів зловмисниками (світ)

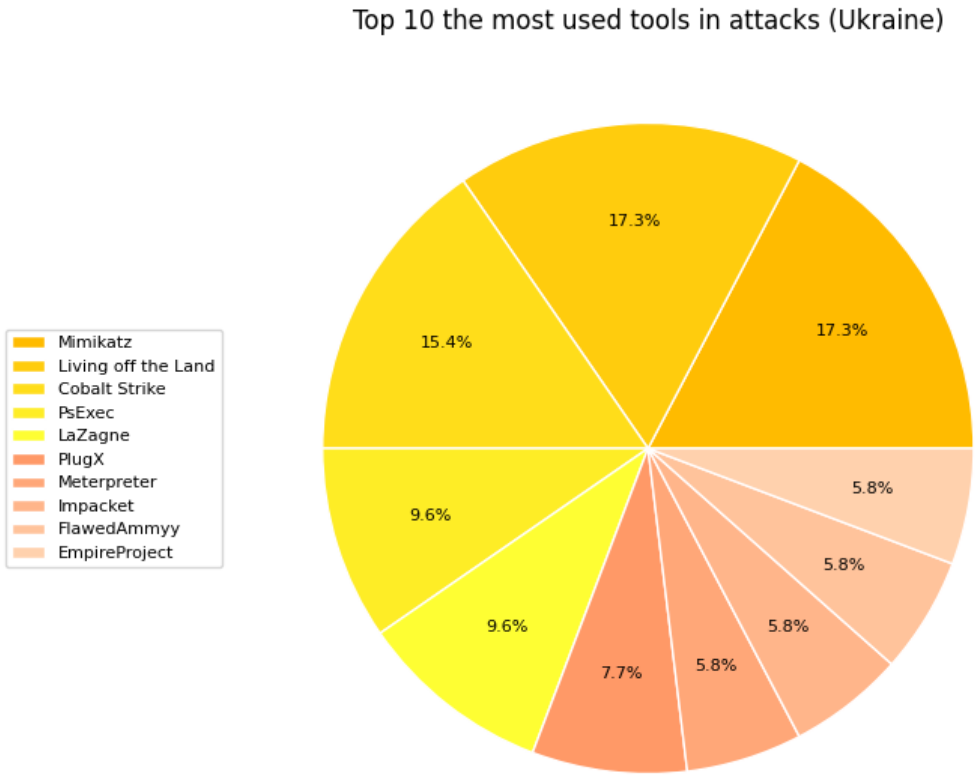


Рисунок 3.26 – Топ-10 найчастіше використовуваних APTs інструментів (Україна)

Останньою, але ні в якому разі не менш важливою, є діаграма, що демонструє залежність між атакуючими APT групами та часом реалізації нападів на кібертерени України (Рисунок 3.27). Варто зауважити, що згаданий час або *timeline* представлений у вигляді року, а не конкретної дати, тому що в цілях безпеки або через брак інформації, команди фахівців, які займаються аналізом APTs, не вказують чіткої дати початку проведення кібероперації.

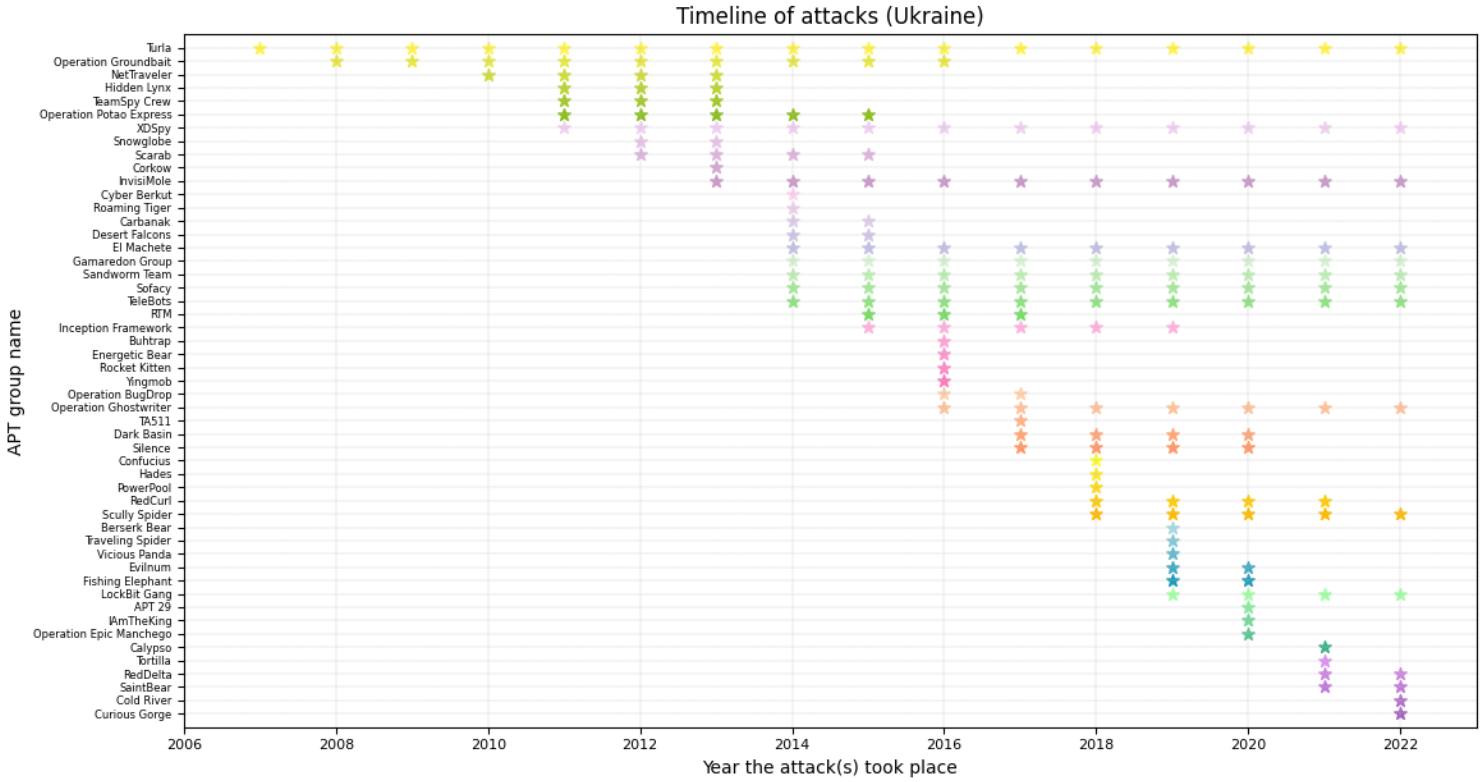


Рисунок 3.27 – Графік, що ілюструє роки проведених атак вказаними APT групами (Україна)

Як можна побачити з наведеного вище графіку, активність зловмисних груп значно зросла з 2014 року і продовжує набирати темпів, зокрема більшість цих APTs походять з росії та Китаю. У 2022 році, після початку повномасштабного вторгнення російських військ на територію України, з’явилися нові групи, що не були поміченими до цього.

3.2.4 Обговорення результатів

Останнім кроком моделі CSIAC є *conclude* – підбиття підсумків та надання рекомендацій, що допоможуть створити додатковий захист системи.

Проаналізувавши зібрані і відсортовані дані за APTs, було встановлено тактики і техніки, що найчастіше застосовують групи, інструменти, що є найпопулярнішими для використання при проведенні атак, залежності між країнами походження груп та цільових секторів і т.д. як у світовому кіберпросторі, так і конкретно в Україні.

Беручи до уваги окремі компоненти, що були описані, та розглядаючи їх у сукупності, спеціалісти можуть надати рекомендації, необхідні для впровадження правил, процедур, методів і способів для підвищення рівня захисту кожної КС окремо та КМ загалом. Як наприклад: фільтрація доменів країн, APTs яких найчастіше атакують територію отримувача, своєчасне оновлення і постійна підтримка систем та їхніх програмних компонентів, встановлення антивірусу та сканерів для електронних листів і вебсайтів, проведення лекцій та практик щодо кібергігієни для працівників організаці, тощо.

Висновки до розділу 3

Запропонована модель CSIAC надає фундамент для створення не лише універсального способу атрибуції, а й відкриває можливості для її покращення та введення рекомендацій, що допоможуть усунути/зменшити наслідки здійснених атак, запровадити політики і процедури для завчасного виявлення підозрілої активності, що може вказувати на перебування чи вхід APTs до системи, та реагування на кіберінциденти такого характеру, тощо. Модель CSIAC є зручною у користуванні та зрозумілою навіть для звичайних користувачів.

CSIAC не вирішує юридичної проблеми атрибуції у тому вигляді, в якому вона зараз існує, проте у майбутньому, сформувавши закони та правила цього процесу на міжнародному рівні, їх можна буде імплементувати до самої моделі.

Програмна реалізація даної моделі ілюструє її можливості та вказує на її зручність у використанні, зокрема, наголошуючи на тому, що процес описаний CSIAC стає зрозумілим навіть для звичайних користувачів.

ВИСНОВКИ

1. Актуальність роботи зумовлена тим, що з кожним роком небезпека APT груп зростає, адже зловмисники набувають усе більше досвіду, пристосовуються до змін, що відбуваються у цільовій системі, модифікуючи або змінюючи інструментарій і підхід до виконання атак, залишаються непоміченими довгий проміжок часу, що може дорівнювати кільком рокам. Вирішення цього питання на сьогоднішній день є пріоритетним завданням спеціалістів з кібербезпеки у всьому світі. Одним з методів його рішення є атрибуція, мета якої полягає у визначенні зловмисників та їхніх замовників.
2. Існує варіація різних моделей проведення процесу атрибуції, але вони не надають рішення її основних проблем: технічної – коректність результатів та юридичної – притягнення винних до відповідальності за скоєні злочини.
3. У ході виконання даної роботи було проаналізовано наявні методи атрибуції, оцінено переваги і недоліки, врахувавши які, було розроблено модель під назвою CSIAC, що є науковою новизною роботи та базується на колективній атрибуції, у такий спосіб надаючи рішення для її технічного аспекту та закладаючи фундамент для впровадження її міжнародного стандарту.

Варто відзначити особливість запропонованої моделі – введення ілюстрацій залежностей між інформацією, що була зібрана за кожною APT групою окремо, та надання рекомендацій щодо усунення наслідків і запобігання повторних кібернападів.

Також була продемонстрована програмна реалізація даної моделі на основі мови програмування Python, зі створенням бази даних за APT групами за допомогою SQLite та використанням властивостей об'єктно-орієнтованого програмування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Steffens T. Attribution of Advanced Persistent Threats: How to Identify the Actors Behind Cyber-Espionage [Текст] / T. Steffens. – Heidelberg : Springer Vieweg, July 22nd, 2020. – с. 5, 35-40.
2. Rid T., Buchanan B. Attributing Cyber Attacks [Текст] / T. Rid, B. Buchanan // Journal of Strategic Studies. – London : Department of War Studies, King's College London, 2015. – Vol. 38. – с. 4-10.
3. Berghel H. On the Problem of (Cyber) Attribution [Електронний ресурс] // Computer. – March 2017. – Vol. 55. – с. 85. – Режим доступу: https://www.computer.org/csdl/magazine/co/2017/03/mco2017030084/13rRUxjyX7t. – Дата звернення: 01.06.2022.
4. Hill A.G. The Ultimate Challenge: Attribution for Cyber Operations [Текст] / A. G. Hill // Wright Flyer Paper. – Alabama : Air University Press, The United States Air Force Air Command and Staff College, November 2019. – No. 70. – с. 20-24.
5. Payne C. N., Finlay L. Addressing Obstacles to Cyber-Attribution: A Model Based on State Response to Cyber-Attack [Текст] // George Washington International Law Review. – Washington : George Washington University Law School, 2017. – Vol. 49. – с. 559-563.
6. Pahi T., Skopik F.Cyber Attribution 2.0: Capture the False Flag [Текст] // In Proceedings of the 18th European Conference on Cyber Warfare and Security (ECCWS 2019). – Coimbra, July 4th-5th 2019. – с. 338-346.
7. Warikoo A. The Triangle Model for Cyber Threat Attribution [Текст] // Journal of Cyber Security Technology. – March 2021. – с. 4-7.
8. Lutkevich B. Definition of Framework [Електронний ресурс]. – August 2020. – Режим доступу: https://www.techtarget.com/whatis/definition/framework. – Дата звернення: 02.06.2022.

9. Shakarian P., Simari G. I., Moores G., Parsons S. Cyber Attribution: An Argumentation-Based Approach [Текст] // Cyber Warfare: Building the Scientific Foundation. – Springer, April 2015. – с. 151-164.
10. Threat Group Cards: A Threat Actor Encyclopedia [Електронний ресурс]. – Bangkok : Electronic Transactions Development Agency. – Режим доступу: https://apt.etda.or.th/cgi-bin/listgroups.cgi. – Дата звернення: 08.05.2022.
11. MITRE ATT&CK [Електронний ресурс]. – The MITRE Corporation, 2015-2022. – Режим доступу: https://attack.mitre.org/. – Дата звернення: 05.06.2022.
12. pyattck [Електронний ресурс]. – Режим доступу: https://github.com/swimlane/pyattck. – Дата звернення: 08.05.2022.
13. Beautiful Soup [Електронний ресурс]. – Режим доступу: https://www.crummy.com/software/BeautifulSoup/. – Дата звернення: 15.05.2022.
14. Defanged Indicator of Compromise (IOC) Extractor [Електронний ресурс]. – Режим доступу: https://github.com/InQuest/python-iocextract. – Дата звернення: 15.05.2022.
15. ENR 2019 Top 250 Global Contractors 1-100 [Електронний ресурс]. – Engineering News-Record (ENR), 2019. – Режим доступу: https://www.enr.com/toplists/2019-Top-250-Global-Contractors-1. – Дата звернення: 06.06.2022.
16. MITRE ATT&CK Enterprise Tactics [Електронний ресурс]. – The MITRE Corporation, 2015-2022. – Режим доступу: https://attack.mitre.org/tactics/enterprise/. – Дата звернення: 07.06.2022.
17. MITRE ATT&CK Enterprise Techniques [Електронний ресурс]. – The MITRE Corporation, 2015-2022. – Режим доступу: https://attack.mitre.org/techniques/enterprise/. – Дата звернення: 07.06.2022.

ДОДАТКИ

ДОДАТОК А ПРОГРАМНИЙ КОД

```
import json, sqlite3, iocextract, os, re
from pyattck import Attck
from urllib.request import Request, urlopen
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
import numpy as np

attack = Attck()

connection_1 = sqlite3.connect('APT_Groups.db')
c_1 = connection_1.cursor()
c_1.execute("CREATE TABLE IF NOT EXISTS APT_World (name TEXT, aliases_etda TEXT, aliases_mitre_attck
TEXT, country_of_origin TEXT, target_countries TEXT, target_sectors TEXT, techniques TEXT, tactics
TEXT, tools TEXT)")
c_1.execute("CREATE TABLE IF NOT EXISTS APT_Ukraine (name TEXT, aliases_etda TEXT,
aliases_mitre_attck TEXT, country_of_origin TEXT, target_sectors TEXT, techniques TEXT, tactics
TEXT, tools TEXT, timeline TEXT, hashes TEXT, timeline_source TEXT, hashes_source TEXT)")
connection_2 = sqlite3.connect('APT_Ukraine_hashes_and_timelines.db')
c_2 = connection_2.cursor()

#зчитування даних з файлу у словник
def read_data(file_name):
        data = open(file_name, encoding = "utf8")
        data_dict = dict()
        data_dict = json.load(data)['values']
        data.close()
        return data_dict

#складання переліку apt груп за назвою
def get_names(data):
        names_list = []
        for apt_group in data:
                names_list.append(apt_group['actor'].split(', ')[0])
        return names_list

#усі APT групи зі списку
class APT_World:

        def __init__(self, data, name):
                self.data = data
                self.name = name
                self.aliases_etda = []
                self.aliases_mitre = []
                self.country_of_origin = ""
                self.target_countries = []
                self.target_sectors = []
                self.techniques = []
                self.tactics = []
                self.tools = []

        #знаходження усіх назв за ETDA
        def get_aliases_etda(self):
                if self.name.startswith('APT'):
                        self.aliases_etda.append(self.name.replace(" ", ""))
                for apt_group in self.data:
```

```
                        if apt_group['actor'].split(', ')[0].lower() == self.name.lower():
                                for item in apt_group['names']:
                                        if item['name'].lower() != self.name.lower():
                                                self.aliases_etda.append(item['name'])
                                if not self.aliases_etda:
                                        self.aliases_etda.append("unknown")
                                break
                return self.aliases_etda

        #знаходження усіх назв за Mitre Att&ck
        def get_aliases_mitre(self):
                if self.aliases_etda[0] != "unknown":
                        etda_names = self.aliases_etda.copy()
                else:
                        etda_names = []
                etda_names.append(self.name)
                etda_names.append(self.name + " APT")
                etda_names.append(self.name + " Group")
                etda_names.append(self.name + " Gang")
                etda_names.append(self.name + " Team")
                for etda_name in etda_names:
                        for apt_group in attack.enterprise.actors:
                                mitre_names = apt_group.alias.copy()
                                for mitre_name in mitre_names:
                                        if etda_name.lower() == mitre_name.lower():
                                                self.aliases_mitre = mitre_names.copy()
                                                break
                if not self.aliases_mitre:
                        self.aliases_mitre.append("unknown")
                return self.aliases_mitre

        #знаходження країни походження
        def get_country_of_origin(self):
                for apt_group in self.data:
                        if apt_group['actor'].split(', ')[0] == self.name:
                                self.country_of_origin = ((apt_group['country'][0].replace('[',
'')).replace(']', '')).lower()
                                break
                return self.country_of_origin

        #знаходження усіх постраждалих країн
        def get_target_countries(self):
                for apt_group in self.data:
                        if apt_group['actor'].split(', ')[0] == self.name:
                                if 'observed-countries' in apt_group.keys():
                                        for country in apt_group['observed-countries']:

        self.target_countries.append(country.lower())
                                else:
                                        self.target_countries.append("unknown")
                                break
                return self.target_countries

        #знаходження усіх постраждалих секторів
        def get_target_sectors(self):
                for apt_group in self.data:
                        if apt_group['actor'].split(', ')[0] == self.name:
                                if 'observed-sectors' in apt_group.keys():
                                        for sector in apt_group['observed-sectors']:
                                                self.target_sectors.append(sector.lower())

                                else:
                                        self.target_sectors.append("unknown")
                                break
                return self.target_sectors

        #знаходження усіх інструментів
        def get_tools(self):
```

```
                for apt_group in self.data:
                        if apt_group['actor'].split(', ')[0] == self.name:
                                if 'tools' in apt_group.keys():
                                        for tool in apt_group['tools']:
                                                self.tools.append(tool)
                                else:
                                        self.tools.append("unknown")
                                break
                return self.tools

        #знаходження усіх технік та тактик
        def get_techniques_and_tactics(self):
                for actor in attack.enterprise.actors:
                        for name in self.aliases_mitre:
                                if actor.name == name:
                                        for technique in actor.techniques:
                                                technique_info = technique.id + ":" +
technique.name
                                                self.techniques.append(technique_info)
                                                for tactic in technique.tactics:
                                                        tactic_info =  tactic.id + ":" +
tactic.name
                                                        self.tactics.append(tactic_info)
                                                self.tactics =
list(dict.fromkeys(self.tactics))
                                        break
                if not self.techniques:
                        self.techniques.append("unknown")
                if not self.tactics:
                        self.tactics.append("unknown")
                return self.techniques, self.tactics

        #пошук за вказаною постраждалою країною
        def check_target_country(self, country):
                for target_country in self.target_countries:
                        if target_country == country:
                                return self.name

        #перевірка достатньості кількості інформації
        def info_sufficiency(self):
                helper = 0
                attributes_list = [self.country_of_origin, self.target_countries,
self.target_sectors, self.tools, self.techniques]
                for item in attributes_list:
                        if type(item) is list:
                                if item[0] == "unknown":
                                        helper += 1
                        else:
                                if item == "unknown":
                                        helper += 1
                return helper

        #запис даних у базу даних
        def insert_data(self):
                aliases_etda_str = ', '.join([str(alias) for alias in self.aliases_etda])
                aliases_mitre_str = ', '.join([str(alias) for alias in self.aliases_mitre])
                target_countries_str = ', '.join([str(target_country) for target_country in
self.target_countries])
                target_sectors_str = ', '.join([str(target_sector) for target_sector in
self.target_sectors])
                techniques_str = ', '.join([str(technique) for technique in self.techniques])
                tactics_str = ', '.join([str(tactic) for tactic in self.tactics])
                tools_str = ', '.join([str(tool) for tool in self.tools])
                c_1.execute("INSERT INTO APT_World (name, aliases_etda, aliases_mitre_attck,
country_of_origin, target_countries, target_sectors, techniques, tactics, tools) VALUES
(?,?,?,?,?,?,?,?,?);", (self.name, aliases_etda_str, aliases_mitre_str, self.country_of_origin,
target_countries_str, target_sectors_str, techniques_str, tactics_str, tools_str))
```

```
        def get_all(self):
                self.get_aliases_etda()
                self.get_aliases_mitre()
                self.get_country_of_origin()
                self.get_target_countries()
                self.get_target_sectors()
                self.get_techniques_and_tactics()
                self.get_tools()

        def main(self):
                self.get_all()
                counter = self.info_sufficiency()
                if counter < 4:
                        self.insert_data()

#усі APT групи зі списку, що атакували кіберпростір України
class APT_Ukraine(APT_World):

        def __init__(self, data, name):
                super().__init__(data, name)
                self.timeline = []
                self.hashes = []
                self.timeline_source = []
                self.hashes_source = []

        #зчитування посилань на timeline атак групи
        def get_timeline_source(self):
                c_2.execute("SELECT timeline_source FROM APT_Ukraine_hashes_and_timelines WHERE
name=?", [self.name])
                source = c_2.fetchone()
                self.timeline_source = ''.join(source).split(',')
                return self.timeline_source

        #зчитування timeline
        def get_timeline(self):
                c_2.execute("SELECT timeline FROM APT_Ukraine_hashes_and_timelines WHERE name=?",
[self.name])
                timeline = c_2.fetchone()
                self.timeline = ''.join(timeline).split(',')
                return self.timeline

        #зчитування посилань на хеші
        def get_hashes_source(self):
                c_2.execute("SELECT hashes_source FROM APT_Ukraine_hashes_and_timelines WHERE
name=?", [self.name])
                source = c_2.fetchone()
                self.hashes_source = ''.join(source).split(',')
                return self.hashes_source

        #пошук усіх хешів шкідливих файлів за посиланням
        def get_hashes_from_url(self):
                for url in self.hashes_source:
                        if url != "?":
                                request = Request(url, headers = {'User-Agent': 'Mozilla/5.0
Chrome/41.0.2228.0', 'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'})
                                webpage = urlopen(request).read()
                                soup = BeautifulSoup(webpage, features = "html.parser",
from_encoding = "iso-8859-1")
                                #видалення усіх script та style елементів
                                for script in soup(["script", "style"]):
                                        script.extract()
                                #отримання тексту з вказаного посилання
                                text = soup.get_text()
                                #розбивання на рядки та видалення пробілів на початку та в
кінці кожного рядка
                                lines = (line.strip() for line in text.splitlines())
                                #розбивання мульти-заголовків на окремі рядки
```

```
                                        chunks = (phrase.strip() for line in lines for phrase in
line.split("  "))
                                        #видалення порожніх рядків
                                        text = '\n'.join(chunk for chunk in chunks if chunk)
                                        for hash_value in iocextract.extract_hashes(text):
                                                if hash_value not in self.hashes:
                                                        self.hashes.append(hash_value)
                        if not self.hashes:
                                self.hashes.append("unknown")
                        return self.hashes

                #запис даних у базу даних
                def insert_data_ukraine(self):
                        aliases_etda_str = ', '.join([str(alias) for alias in self.aliases_etda])
                        aliases_mitre_str = ', '.join([str(alias) for alias in self.aliases_mitre])
                        target_sectors_str = ', '.join([str(target_sector) for target_sector in
self.target_sectors])
                        techniques_str = ', '.join([str(technique) for technique in self.techniques])
                        tactics_str = ', '.join([str(tactic) for tactic in self.tactics])
                        tools_str = ', '.join([str(tool) for tool in self.tools])
                        timeline_str = ', '.join([str(time) for time in self.timeline])
                        hashes_str = ', '.join([str(hash_value) for hash_value in self.hashes])
                        timeline_source_str = ', '.join([str(t_source) for t_source in
self.timeline_source])
                        hashes_source_str = ', '.join([str(h_source) for h_source in
self.hashes_source])
                        c_1.execute("INSERT INTO APT_Ukraine (name, aliases_etda, aliases_mitre_attck,
country_of_origin, target_sectors, techniques, tactics, tools, timeline, hashes, timeline_source,
hashes_source) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?);", (self.name, aliases_etda_str, aliases_mitre_str,
self.country_of_origin, target_sectors_str, techniques_str, tactics_str, tools_str, timeline_str,
hashes_str, timeline_source_str, hashes_source_str))

                #запуск для отримання значень (додаткових) атрибутів класу
                def main(self):
                        super().get_all()
                        counter = self.info_sufficiency()
                        if counter < 4:
                                self.get_timeline_source()
                                self.get_timeline()
                                self.get_hashes_source()
                                self.get_hashes_from_url()
                                self.insert_data_ukraine()

#графіки залежностей
class Graphs:

                def count_items(self, item_list_without_repeats, item_list):
                        item_count = []
                        for item in item_list_without_repeats:
                                counter = 0
                                for element in item_list:
                                        if item == element:
                                                counter += 1
                                item_count.append(counter)
                        return item_count

                #перелік усіх APT груп з таблички
                def select_all_names(self, table_name):
                        names = []
                        c_1.execute("SELECT name FROM {}".format(table_name))
                        apt_names = c_1.fetchall()
                        for name in apt_names:
                                names.append(''.join(str(name)).replace('(', '').replace(')',
'').replace("'", '').replace(',', ''))
                        return names

                #вибір усіх країн походження APT груп з таблички
                def select_all_countries_of_origin(self, table_name):
```

```
                countries = []
                c_1.execute("SELECT country_of_origin FROM {}".format(table_name))
                countries_of_origin = c_1.fetchall()
                for country in countries_of_origin:
                        countries.append(str(country).replace('(', '').replace(')',
'').replace(',', '').replace("'", ''))
                countries_without_repeats = list(dict.fromkeys(countries))
                return countries_without_repeats, countries

        #вибір усіх секторів з таблички
        def select_all_sectors(self, table_name):
                sectors = []
                c_1.execute("SELECT target_sectors FROM {}".format(table_name))
                target_sectors = c_1.fetchall()
                for item in target_sectors:
                        s_list = ''.join(item).split(', ')
                        for sector in s_list:
                                if sector != 'unknown':
                                        sectors.append(sector)
                sectors_without_repeats = list(dict.fromkeys(sectors))
                return sectors_without_repeats, sectors

        #вибір усіх технік з таблиці
        def select_all_techniques(self, table_name):
                techniques = []
                c_1.execute("SELECT techniques FROM {}".format(table_name))
                used_techniques = c_1.fetchall()
                for item in used_techniques:
                        t_list = ''.join(str(item).replace('(', '').replace(')',
'').replace("'", '').replace(", ", ',')).split(',')
                        for element in t_list:
                                if element.startswith('T1'):
                                        techniques.append(element)
                techniques_without_repeats = list(dict.fromkeys(techniques))
                return techniques_without_repeats, techniques

        #вибір усіх тактик з таблиці
        def select_all_tactics(self, table_name):
                tactics = []
                c_1.execute("SELECT tactics FROM {}".format(table_name))
                used_tactics = c_1.fetchall()
                for item in used_tactics:
                        t_list = ''.join(str(item).replace('(', '').replace(')',
'').replace("'", '').replace(", ", ',')).split(',')
                        for element in t_list:
                                if element.startswith('TA'):
                                        tactics.append(re.split(":", element)[1])
                tactics_without_repeats = list(dict.fromkeys(tactics))
                return tactics_without_repeats, tactics

        #вибір усіх інструментів з таблиці
        def select_all_tools(self, table_name):
                tools = []
                c_1.execute("SELECT tools FROM {}".format(table_name))
                used_tools = c_1.fetchall()
                for item in used_tools:
                        t_list = ''.join(item).split(', ')
                        for tool in t_list:
                                if tool != 'unknown':
                                        tools.append(tool)
                tools_without_repeats = list(dict.fromkeys(tools))
                return tools_without_repeats, tools

        #демонстрація шкали країн-походження APT груп
        def country_of_origin_graph(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_country_of_origin.png"
```

```
                        pdf_name = "APT_World_country_of_origin.pdf"
                        title = "Scale of APT groups countries of origin (Worldwide)"
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_country_of_origin.png"
                        pdf_name = "APT_Ukraine_country_of_origin.pdf"
                        title = "Scale of APT groups countries of origin (Ukraine)"
                else:
                        exit()
                countries = self.select_all_countries_of_origin(table_name)[0]
                c_list = self.select_all_countries_of_origin(table_name)[1]
                countries.remove('unknown')
                country_count = self.count_items(countries, c_list)
                country_count, countries = zip(*sorted(zip(country_count, countries), reverse =
True)) #сортування списків відносно один одного
                y = np.array(countries)
                x = np.array(country_count)

                fig, ax = plt.subplots(figsize = (14, 7))
                bar_colors = ['#A460C3', '#B26CCE', '#C079D9', '#CE85E4', '#DC92EF', '#3DB388',
'#57C690', '#70D997', '#8AEC9F', '#A3FFA6',
                                                '#209BBA', '#42AAC3', '#64B9CC', '#86C9D6',
'#A8D8DF', '#FFBB00', '#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33',
                                                '#FF9967', '#FFA778', '#FFB58A', '#FFC39B',
'#FFD1AD', '#FF78BC', '#FF87C5', '#FF95CD', '#FFA4D6', '#FFB3DE']
                bars = ax.barh(y, x, height = 1, color = bar_colors, edgecolor = "white")
                ax.bar_label(bars, padding = 1, fontsize = 8)
                ax.set_ylabel('Country', fontsize = 10)
                ax.set_xlabel('Number of APT groups', fontsize = 10)
                ax.set_title('Scale of APT groups countries of origin', fontsize = 12)
                ax.set_ylim(-0.5, len(y))
                plt.yticks(fontsize = 8)
                plt.xticks(fontsize = 8)
                plt.grid(linestyle = '--', linewidth = 0.3, color = '#BFBFBF')
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

        #демонстрація 20 країн, які найбільше потрапляють під атаки APT груп
        def top_20_targeted_countries_graph(self):
                country_list = []
                country_count = []
                c_1.execute("SELECT target_countries FROM APT_World")
                countries = c_1.fetchall()
                for item in countries:
                        countries_list = ''.join(item).split(', ')
                        for country in countries_list:
                                country_list.append(country)
                countries = list(dict.fromkeys(country_list)) #видалення повторень
                for country in countries:
                        counter = 0
                        for item in country_list:
                                if item == country:
                                        counter += 1
                        country_count.append(counter)
                country_count, countries = zip(*sorted(zip(country_count, countries), reverse =
True)) #сортування списків відносно один одного
                x = countries[:20]
                y = country_count[:20]

                fig, ax = plt.subplots(figsize = (14, 7))
                element_explode = []
                for item in x:
                        if item == 'ukraine':
                                element_explode.append(0.15)
                        else:
                                element_explode.append(0)
```

```
                pie_colors = ['#A460C3', '#B26CCE', '#C079D9', '#CE85E4', '#DC92EF', '#3DB388',
'#57C690', '#70D997', '#8AEC9F', '#A3FFA6',
                                                '#209BBA', '#42AAC3', '#64B9CC', '#86C9D6',
'#A8D8DF', '#FFBB00', '#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33']
                ax.pie(y, autopct = '%.1f%%', pctdistance = 0.75, colors = pie_colors, textprops
= {'fontsize':6},  wedgeprops = {'edgecolor':'#FFFFFF'}, explode = element_explode)
                ax.legend(labels = x, loc = 'center left', bbox_to_anchor = (-0.3, 0.5), fontsize
= 10)
                ax.set_title('Top 20 targeted countries', fontsize = 12)
                plt.savefig('Top_20_targeted_countries.png')
                plt.savefig('Top_20_targeted_countries.pdf')
                plt.show()

        #демонстрація шкали секторів, які атакують APT групи
        def targeted_sectors_graph(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_targeted_sectors.png"
                        pdf_name = "APT_World_targeted_sectors.pdf"
                        title = "Scale of targeted sectors (Worldwide)"
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_targeted_sectors.png"
                        pdf_name = "APT_Ukraine_targeted_sectors.pdf"
                        title = "Scale of targeted sectors (Ukraine)"
                else:
                        exit()
                sectors = self.select_all_sectors(table_name)[0]
                s_list = self.select_all_sectors(table_name)[1]
                sector_count = self.count_items(sectors, s_list)
                sector_count, sectors = zip(*sorted(zip(sector_count, sectors), reverse = True))
#сортування списків відносно один одного
                y = np.array(sectors)
                x = np.array(sector_count)

                fig, ax = plt.subplots(figsize = (14, 7))
                bar_colors = ['#A460C3', '#B26CCE', '#C079D9', '#CE85E4', '#DC92EF', '#3DB388',
'#57C690', '#70D997', '#8AEC9F', '#A3FFA6',
                                                '#209BBA', '#42AAC3', '#64B9CC', '#86C9D6',
'#A8D8DF', '#FFBB00', '#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33',
                                                '#FF9967', '#FFA778', '#FFB58A', '#FFC39B',
'#FFD1AD', '#FF78BC', '#FF87C5', '#FF95CD', '#FFA4D6', '#FFB3DE',
                                                '#77DD66', "#8FE381", '#A7E99C', '#BFEEB7',
'#D7F4D2', '#6777B8', '#7280BF', '#7C8AC5', '#8793CC', '#9CA6D9']
                bars = ax.barh(y, x, height = 1, color = bar_colors, edgecolor = "white")
                ax.bar_label(bars, padding = 1, fontsize = 6)
                ax.set_ylabel('Targeted sectors', fontsize = 8)
                ax.set_xlabel('Number of APT groups', fontsize = 8)
                ax.set_title(title, fontsize = 12)
                ax.set_ylim(-0.5, len(y))
                plt.yticks(fontsize = 6)
                plt.xticks(fontsize = 6)
                plt.grid(linestyle = '--', linewidth = 0.3, color = '#BFBFBF')
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

        #демонстрація 15 технік, які найбільше використовують APT групи
        def top_15_used_techniques_graph(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_top_15_used_techniques.png"
                        pdf_name = "APT_World_top_15_used_techniques.pdf"
                        title = "Top 15 the most used techniques in attacks (Worldwide)"
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_top_15_used_techniques.png"
                        pdf_name = "APT_Ukraine_top_15_used_techniques.pdf"
```

```
                        title = "Top 15 the most used techniques in attacks (Ukraine)"
                else:
                        exit()
                techniques = self.select_all_techniques(table_name)[0]
                t_list = self.select_all_techniques(table_name)[1]
                technique_count = self.count_items(techniques, t_list)
                technique_count, techniques = zip(*sorted(zip(technique_count, techniques),
reverse = True)) #сортування списків відносно один одного
                x = techniques[:15]
                y = technique_count[:15]

                fig, ax = plt.subplots(figsize = (14, 7))
                pie_colors = ['#A460C3', '#B26CCE', '#C079D9', '#CE85E4', '#DC92EF', '#FFBB00',
'#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33',
                                                '#FF78BC', '#FF87C5', '#FF95CD', '#FFA4D6',
'#FFB3DE']
                ax.pie(y, autopct = '%.1f%%', pctdistance = 0.8, colors = pie_colors, textprops =
{'fontsize':7},  wedgeprops = {'edgecolor':'#FFFFFF'})
                ax.legend(labels = x, loc = 'center left', bbox_to_anchor = (1, 0.5), fontsize =
8)
                ax.set_title(title, fontsize = 12)
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

        #демонстрація 10 тактик, які найбільше використовують APT групи
        def top_10_used_tactics_graph(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_top_10_used_tactics.png"
                        pdf_name = "APT_World_top_10_used_tactics.pdf"
                        title = "Top 10 the most used tactics in attacks (Worldwide)"
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_top_10_used_tactics.png"
                        pdf_name = "APT_Ukraine_top_10_used_tactics.pdf"
                        title = "Top 10 the most used tactics in attacks (Ukraine)"
                else:
                        exit()
                tactics = self.select_all_tactics(table_name)[0]
                t_list = self.select_all_tactics(table_name)[1]
                tactic_count = self.count_items(tactics, t_list)
                tactic_count, tactics = zip(*sorted(zip(tactic_count, tactics), reverse = True))
#сортування списків відносно один одного
                x = tactics[:10]
                y = tactic_count[:10]

                fig, ax = plt.subplots(figsize = (14, 7))
                bar_colors = ['#209BBA', '#42AAC3', '#64B9CC', '#86C9D6', '#A8D8DF', '#FFBB00',
'#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33']
                bars = ax.barh(x, y, height = 1, color = bar_colors, edgecolor = "white")
                ax.bar_label(bars, padding = 1, fontsize = 8)
                ax.set_ylabel('Tactic', fontsize = 10)
                ax.set_xlabel('Usage count', fontsize = 10)
                ax.set_title(title, fontsize = 12)
                ax.set_ylim(-0.5, len(y))
                plt.yticks(fontsize = 8)
                plt.xticks(fontsize = 8)
                plt.grid(linestyle = '--', linewidth = 0.3, color = '#BFBFBF')
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

        #демонстрація топ програм, які найбільше використовують APT групи
        def top_used_tools_graph(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_top_15_used_tools.png"
```

```
                        pdf_name = "APT_World_top_15_used_tools.pdf"
                        title = "Top 15 the most used tools in attacks (Worldwide)"
                        n = 15
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_top_10_used_tools.png"
                        pdf_name = "APT_Ukraine_top_10_used_tools.pdf"
                        title = "Top 10 the most used tools in attacks (Ukraine)"
                        n = 10
                else:
                        exit()
                tools = self.select_all_tools(table_name)[0]
                t_list = self.select_all_tools(table_name)[1]
                tool_count = self.count_items(tools, t_list)
                tool_count, tools = zip(*sorted(zip(tool_count, tools), reverse = True))
                y = np.array(tools[:n])
                x = np.array(tool_count[:n])

                fig, ax = plt.subplots(figsize = (14, 7))
                pie_colors = ['#FFBB00', '#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33', '#FF9967',
'#FFA778', '#FFB58A', '#FFC39B', '#FFD1AD',
                                                '#FF78BC', '#FF87C5', '#FF95CD', '#FFA4D6',
'#FFB3DE']
                ax.pie(x, autopct = '%.1f%%', pctdistance = 0.7, colors = pie_colors, textprops =
{'fontsize':8},  wedgeprops = {'edgecolor':'#FFFFFF'})
                ax.legend(labels = y, loc = 'center left', bbox_to_anchor = (-0.3, 0.5), fontsize
= 8)
                ax.set_title(title, fontsize = 12)
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

        #демонстрація часу атак APT груп на різноманітні об'єкти та структури України
        def timeline_graph(self):
                timeline = []
                names = []
                c_1.execute("SELECT name, timeline FROM APT_Ukraine")
                data = c_1.fetchall()
                for pair in data:
                        time = []
                        if pair[1] != "?":
                                names.append(pair[0])
                                years = pair[1].split(', ')
                                for item in years:
                                        if len(item) == 9:
                                                t = list(range(int(re.split("-", item)[0]),
int(re.split("-", item)[1]) + 1))
                                                for element in t:
                                                        time.append(element)
                                        elif len(item) == 4:
                                                time.append(int(item))
                                timeline.append(time)
                timeline, names = zip(*sorted(zip(timeline, names), reverse = True))

                scatter_colors = ['#A460C3', '#B26CCE', '#C079D9', '#CE85E4', '#DC92EF',
'#3DB388', '#57C690', '#70D997', '#8AEC9F', '#A3FFA6',
                                                '#209BBA', '#42AAC3', '#64B9CC',
'#86C9D6', '#A8D8DF', '#FFBB00', '#FFCC0D', '#FFDD1A', '#FEED26', '#FEFE33',
                                                '#FF9967', '#FFA778', '#FFB58A',
'#FFC39B', '#FFD1AD', '#FF78BC', '#FF87C5', '#FF95CD', '#FFA4D6', '#FFB3DE',
                                                '#77DD66', "#8FE381", '#A7E99C',
'#BFEEB7', '#D7F4D2', '#C3C1E6', '#D2C6E8', '#E1CCEB', '#F0D1ED', '#FFD6EF',
                                                '#CB99C9', '#D4A7D3', '#DEB5DE',
'#E7C3E8', '#F0D1F2', '#8BBE1C', '#A2C926', '#B9D42F', '#D1DE39', '#E8E942',
                                                '#FFF44C']
                fig, ax = plt.subplots(figsize = (14, 7))
                counter = 0
                for x, y in zip(timeline, names):
```

```
                        color = scatter_colors[0]
                        for item_x in x:
                                plt.scatter(item_x, y, s = 50, marker = '*', c = color)
                        scatter_colors.pop(0)
                ax.set_xlabel('Year the attack(s) took place', fontsize = 10)
                ax.set_ylabel('APT group name', fontsize = 10)
                ax.set(xlim = (2006, 2023), ylim = (-1, len(names)))
                ax.set_title('Timeline of attacks (Ukraine)', fontsize = 12)
                plt.xticks(fontsize = 8)
                plt.yticks(fontsize = 6)
                plt.grid(linestyle = '--', linewidth = 0.3, color = '#BFBFBF')
                plt.savefig('Timeline of attacks (Ukraine).png')
                plt.savefig('Timeline of attacks (Ukraine).pdf')
                plt.show()

        #демонстрація залежності між секторами та використовуваними тактиками
        def sectors_vs_tactics(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_sectors_vs_tactics.png"
                        pdf_name = "APT_World_sectors_vs_tactics.pdf"
                        title = "Sectors vs tactics (Worldwide)"
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_sectors_vs_tactics.png"
                        pdf_name = "APT_Ukraine_sectors_vs_tactics.pdf"
                        title = "Sectors vs tactics (Ukraine)"
                else:
                        exit()
                names = self.select_all_names(table_name)
                sectors = self.select_all_sectors(table_name)[0]
                tactics = self.select_all_tactics(table_name)[0]
                sectors_best_tactics = []
                for sector in sectors:
                        tactics_for_sector = []
                        for name in names:
                                c_1.execute("SELECT target_sectors FROM {} WHERE
name=?".format(table_name), [name])
                                one_apt_sectors = c_1.fetchone()
                                sectors_for_current_apt =
''.join(str(one_apt_sectors)).replace('(', '').replace(')', '').replace("'", '').replace(", ",
',').split(',')
                                sectors_for_current_apt.pop()
                                if sector in sectors_for_current_apt:
                                        c_1.execute("SELECT tactics FROM {} WHERE
name=?".format(table_name), [name])
                                        one_apt_tactics = c_1.fetchone()
                                        tactics_for_current_apt =
''.join(str(one_apt_tactics)).replace('(', '').replace(')', '').replace("'", '').replace(", ",
',').split(',')
                                        tactics_for_current_apt.pop()
                                        for tactic in tactics_for_current_apt:
                                                if tactic != 'unknown':

        tactics_for_sector.append(re.split(":", tactic)[1])
                                                else:
                                                        tactics_for_sector.append(tactic)
                        if tactics_for_sector:
                                best_tactic = []
                                tactic_count = self.count_items(tactics, tactics_for_sector)
                                for t_c, tactic in zip(tactic_count, tactics_for_sector):
                                        if t_c == max(tactic_count):
                                                if tactic != 'unknown':
                                                        best_tactic.append(tactic)
                                if best_tactic:
                                        sectors_best_tactics.append(best_tactic)
                                else:
                                        sectors.remove(sector)
```

```
                fig, ax = plt.subplots(figsize = (14, 7))
                for x, y in zip(sectors_best_tactics, sectors):
                        for item_x in x:
                                plt.scatter(item_x, y, s = 60, marker = '*', c = '#DC92EF')
                ax.set_xlabel('Tactics', fontsize = 10)
                ax.set_ylabel('Sectors', fontsize = 10)
                ax.set_title('Targeted sectors vs used tactics', fontsize = 12)
                plt.xticks(fontsize = 6, rotation = 25)
                plt.yticks(fontsize = 6)
                plt.grid(linestyle = '--', linewidth = 0.3, color = '#BFBFBF')
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

        #демонстрація залежності між секторами та країнами походження APT груп
        def sectors_vs_countries_of_origin(self, choice):
                if choice == '1':
                        table_name = "APT_World"
                        png_name = "APT_World_sectors_vs_countries_of_origin.png"
                        pdf_name = "APT_World_sectors_vs_countries_of_origin.pdf"
                        title = "Sectors vs countries of origin (Worldwide)"
                elif choice == '2':
                        table_name = "APT_Ukraine"
                        png_name = "APT_Ukraine_sectors_vs_countries_of_origin.png"
                        pdf_name = "APT_Ukraine_sectors_vs_countries_of_origin.pdf"
                        title = "Sectors vs countries of origin (Ukraine)"
                else:
                        exit()
                names = self.select_all_names(table_name)
                sectors = self.select_all_sectors(table_name)[0]
                countries = sorted(self.select_all_countries_of_origin(table_name)[0])
                countries.remove('unknown')
                countries_most_sectors = []
                for country in countries:
                        sectors_for_country = []
                        for name in names:
                                c_1.execute("SELECT country_of_origin FROM {} WHERE
name=?".format(table_name), [name])
                                country_of_origin = c_1.fetchone()
                                country_of_origin =
''.join(str(country_of_origin)).replace('(', '').replace(')', '').replace("'", '').replace(", ",
',')
                                if country in country_of_origin:
                                        c_1.execute("SELECT target_sectors FROM {} WHERE
name=?".format(table_name), [name])
                                        one_apt_sectors = c_1.fetchone()
                                        sectors_for_current_apt =
''.join(str(one_apt_sectors)).replace('(', '').replace(')', '').replace("'", '').replace(", ",
',').split(',')
                                        sectors_for_current_apt.pop()
                                        for sector in sectors_for_current_apt:
                                                sectors_for_country.append(sector)
                        most_sector = []
                        sector_count = self.count_items(sectors, sectors_for_country)
                        for s_c, sector in zip(sector_count, sectors_for_country):
                                if s_c == max(sector_count):
                                        if sector == 'unknown':
                                                countries.remove(country)
                                        else:
                                                most_sector.append(sector)
                        if most_sector:
                                countries_most_sectors.append(most_sector)

                fig, ax = plt.subplots(figsize = (14, 7))
                for x, y in zip(countries_most_sectors, countries):
                        for item_x in x:
                                plt.scatter(item_x, y, s = 60, marker = '*', c = '#70D997')
```

```
                ax.set_xlabel('Sectors', fontsize = 10)
                ax.set_ylabel('Countries', fontsize = 10)
                ax.set_title('Targeted sectors vs countries of origin', fontsize = 12)
                plt.xticks(fontsize = 8, rotation = 15)
                plt.yticks(fontsize = 8)
                plt.grid(linestyle = '--', linewidth = 0.3, color = '#BFBFBF')
                plt.savefig(png_name)
                plt.savefig(pdf_name)
                plt.show()

def results():
        data = read_data('APT.json')
        apt_groups = get_names(data)
        apt_target_ukraine = []
        for apt_group in apt_groups:
                apt = APT_World(data, apt_group)
                apt.main()
                check_target_country_ukraine = apt.check_target_country("ukraine")
                if check_target_country_ukraine != None:
                        apt_target_ukraine.append(check_target_country_ukraine)
        for apt_group in apt_target_ukraine:
                apt = APT_Ukraine(data, apt_group)
                apt.main()

        graph = Graphs()

        choice = '1'
        graph.top_20_targeted_countries_graph()
        graph.country_of_origin_graph(choice)
        graph.targeted_sectors_graph(choice)
        graph.top_15_used_techniques_graph(choice)
        graph.top_10_used_tactics_graph(choice)
        graph.top_used_tools_graph(choice)
        graph.sectors_vs_tactics(choice)
        graph.sectors_vs_countries_of_origin(choice)

        choice = '2'
        graph.country_of_origin_graph(choice)
        graph.targeted_sectors_graph(choice)
        graph.top_15_used_techniques_graph(choice)
        graph.top_10_used_tactics_graph(choice)
        graph.top_used_tools_graph(choice)
        graph.timeline_graph()
        graph.sectors_vs_tactics(choice)
        graph.sectors_vs_countries_of_origin(choice)
        print("About the program: \nAuthor: Viola Mazurenko, FB-83.\n©2022 Viola Mazurenko All
Rights Reserved.\n")

results()
connection_2.commit()
connection_2.close()
connection_1.commit()
connection_1.close()
```

ДОДАТОК Б ДОДАТКОВА БАЗА ДАНИХ

Таблиця Б.1 – Додаткова база даних для APT груп України під назвою APT_Ukraine_hashes_and_timelines

name	timeline	timeline_source	hashes_source
APT 29	2020	https://www.kyivpost.com/technology/microsoft-says-russia-hacked-us-aid-agency-that-works-in-ukraine.html	https://www.mandiant.com/resources/tracking-apt29-phishing-campaigns,https://www.mandiant.com/resources/not-so-cozy-an-uncomfortable-examination-of-a-suspected-apt29-phishing-campaign,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/APT29/APT29_hash.md
Berserk Bear	2019, 2020	https://vblocalhost.com/uploads/VB2021-Slowik.pdf	https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/dragonfly-energy-sector-cyber-attacks
Buhtrap	2016, 2022	https://cert.gov.ua/article/37246	https://cert.gov.ua/article/37246,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Buhtrap/Buhtrap_hash.md
Calypso	2021	https://cyware.com/news/calypso-apt-eyes-microsoft-exchange-vulnerabilities-2ab3f520	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Calypso/Calypso_hash.md
Carbanak	2014-2015	https://www.mandiant.com/resources/behind-the-carbanak-backdoor,https://www.welivesecurity.com/2015/09/08/carbanak-gang-is-back-and-packing-new-guns/	https://www.secureworks.com/blog/excel-add-ins-deliver-jssloader-malware,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/FIN7/FIN7_hash.md,https://www.welivesecurity.com/2015/09/08/carbanak-gang-is-back-and-packing-new-guns/
Circus Spider	?	?	https://www.crowdstrike.com/blog/analysis-of-ecrime-menu-style-toolkits/,https://www.cynet.com/attack-techniques-hands-on/netwalker-ransomware-report/,https://github.com/sophoslabs/IoCs/blob/master/Ransomware-Netwalker
Cobalt Group	?	?	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Cobalt%20Group/Cobalt%20Group_hash.md
Cold River	2022	https://blog.google/threat-analysis-group/tracking-cyber-activity-eastern-europe/	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Cold%20River/Cold%20River_hash.md
Confucius	2018	https://www.trendmicro.com/en_us/research/18/e/confucius-update-new-tools-and-techniques-further-connections-with-patchwork.html	https://www.lookout.com/blog/lookout-discovers-novel-confucius-apt-android-spyware-linked-to-india-pakistan-conflict,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Confucius/Confucius_hash.md
Corkow	2013	https://www.welivesecurity.com/2014/02/27/corkow-analysis-of-a-business-oriented-banking-trojan/	https://www.welivesecurity.com/2014/02/27/corkow-analysis-of-a-business-oriented-banking-trojan/
Curious Gorge	2022	https://blog.google/threat-analysis-group/update-on-cyber-activity-in-eastern-europe/	?
Cyber Berkut	2014	https://www.kyivpost.com/article/content/may-25-presidential-election/authorities-hackers-foiled-in-bid-to-rig-ukraine-presidential-election-results-349288.html,https://carnegieendowment.org/specialprojects/protectingfinancialstability/timeline	?
Desert Falcons	2014-2015	https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08064309/The-Desert-Falcons-targeted-attacks.pdf	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/APT-Q-63/APT-Q-63_hash.md
El Machete	2014-2022	https://blogs.blackberry.com/en/2017/03/el-machete-malware-attacks-cut-through-latam	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/El%20Machete/El%20Machete_hash.md,https://blogs.blackberry.com/en/2017/03/el-machete-malware-attacks-cut-through-latam
Energetic Bear	2016, 2021-2022	https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE4Vwwd,https://www.welivesecurity.com/2017/06/12/industroyer-biggest-threat-industrial-control-systems-since-stuxnet/	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Energetic%20Bear/Energetic%20Bear_hash.md

Продовження таблиці Б.1

Evilnum	2019-2020	https://symantec.broadcom.com/hubfs/SED-Threats-Financial-Sector.pdf	https://github.com/eset/malware-ioc/blob/master/evilnum/samples.md5,https://github.com/eset/malware-ioc/blob/master/evilnum/samples.sha1,https://github.com/eset/malware-ioc/blob/master/evilnum/samples.sha256,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Evilnum/Evilnum_hash.md
Fishing Elephant	2019-2020	https://securelist.com/apt-trends-report-q1-2020/96826/	?
Gamaredon Group	2014-2022	https://cert.gov.ua/article/18365,https://cert.gov.ua/article/39138,https://cert.gov.ua/article/40240,https://cip.gov.ua/en/news/khto-stoyit-za-kiberatakami-na-ukrayinsku-kritichnu-informaciinu-infrastrukturu-statistika-15-22-bereznya,https://ssu.gov.ua/uploads/files/DKIB/Technical%20report%20Armagedon.pdf	https://cert.gov.ua/article/18365,https://cert.gov.ua/article/39138,https://cert.gov.ua/article/39386,https://cert.gov.ua/article/40240,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Gamaredon%20Group/Gamaredon%20Group_hash.md
Hades	2018	https://securelist.com/olympic-destroyer-is-still-alive/86169/	https://securelist.com/olympicdestroyer-is-here-to-trick-the-industry/84295/
Hidden Lynx	2011-2013	https://docs.broadcom.com/doc/hidden_lynx	?
IAmTheKing	2020	https://securelist.com/iamtheking-and-the-slothfulmedia-malware-family/99000/	https://securelist.com/iamtheking-and-the-slothfulmedia-malware-family/99000/
Inception Framework	2015-2019	https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/inception-framework-hiding-behind-proxies,https://securelist.com/recent-cloud-atlas-activity/92016/	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Inception%20Framework/Inception%20Framework_hash.md,https://unit42.paloaltonetworks.com/unit42-inception-attackers-target-europe-year-old-office-vulnerability/
InvisiMole	2013-2022	https://cert.gov.ua/article/37829,https://www.welivesecurity.com/2018/06/07/invisimole-equipped-spyware-undercover/	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/InvisiMole/InvisiMole_hash.md,https://cert.gov.ua/article/37829,https://github.com/eset/malware-ioc/blob/master/invisimole/samples.md5,https://github.com/eset/malware-ioc/blob/master/invisimole/samples.sha1,https://github.com/eset/malware-ioc/blob/master/invisimole/samples.sha256
LockBit Gang	2019-2022	https://www.kaspersky.com/resource-center/threats/lockbit-ransomware,https://www.advintel.io/post/from-russia-with-lockbit-ransomware-inside-look-preventive-solutions	https://cyberint.com/blog/research/lockbit-ransomware-hits-again/,https://www.trendmicro.com/en_us/research/21/h/lockbit-resurfaces-with-version-2-0-ransomware-detections-in-chi.html
MuddyWater	?	?	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/MuddyWater/MuddyWater_hash.md,https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/seedworm-espionage-group,https://github.com/artemissnx/AutonomousThreatSweep/tree/main/Muddywater_attacks_U.S_Worldwide
NetTraveler	2010-2013	https://www.kaspersky.com/about/press-releases/2013_kaspersky-lab-uncovers--operation-nettraveler--a-global-cyberespionage-campaign-targeting-government-affiliated-organizations-and-research-institutes	https://unit42.paloaltonetworks.com/nettraveler-spear-phishing-email-targets-diplomat-of-uzbekistan/
Operation BugDrop	2016-2017	https://www.cfr.org/cyber-operations/operation-bugdrop,https://securityaffairs.co/wordpress/56517/intelligence/operation-bugdrop-ukraine.html	https://blogs.blackberry.com/en/2017/03/threat-spotlight-operation-bugdrop
Operation Epic Manchego	2020	https://blog.nviso.eu/2020/09/01/epic-manchego-atypical-maldoc-delivery-brings-flurry-of-infostealers/	https://blog.nviso.eu/2020/09/01/epic-manchego-atypical-maldoc-delivery-brings-flurry-of-infostealers/
Operation Ghostwriter	2016-2022	https://cert.gov.ua/article/37626,https://www.fireeye.com/content/dam/fireeye-www/blog/pdfs/unc1151-ghostwriter-update-report.pdf,https://www.mandiant.com/resources/unc1151-linked-to-belarus-government	https://cert.gov.ua/article/37626,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Ghostwriter/Ghostwriter_hash.md
Operation Groundbait	2008-2016	https://www.welivesecurity.com/2016/05/18/groundbait/	https://github.com/eset/malware-ioc/blob/master/groundbait/samples.md5,https://github.com/eset/malware-ioc/blob/master/groundbait/samples.sha1,https://github.com/eset/malware-ioc/blob/master/groundbait/samples.sha256

Продовження таблиці Б.1

Operation Potao Express	2011-2015	https://www.welivesecurity.com/wp-content/uploads/2015/07/Operation-Potao-Express_final_v2.pdf	https://github.com/eset/malware-ioc/blob/master/potao/samples.md5,https://github.com/eset/malware-ioc/blob/master/potao/samples.sha1,https://github.com/eset/malware-ioc/blob/master/potao/samples.sha256
PowerPool	2018	https://www.welivesecurity.com/2018/09/05/powerpool-malware-exploits-zero-day-vulnerability/	https://www.welivesecurity.com/2018/09/05/powerpool-malware-exploits-zero-day-vulnerability/
RedCurl	2018-2021	https://explore.group-ib.com/redcurl-english-reports/report-redcurl-eng	?
RedDelta	2021-2022	https://www.proofpoint.com/us/blog/threat-insight/good-bad-and-web-bug-ta416-increases-operational-tempo-against-european	https://www.proofpoint.com/us/blog/threat-insight/good-bad-and-web-bug-ta416-increases-operational-tempo-against-european
Roaming Tiger	2014	https://2014.zeronights.org/assets/files/slides/roaming_tiger_zeronights_2014.pdf	?
Rocket Kitten	2016	https://securelist.com/freezer-paper-around-free-meat/74503/	?
RTM	2015-2017	https://www.welivesecurity.com/wp-content/uploads/2017/02/Read-The-Manual.pdf	https://unit42.paloaltonetworks.com/russian-language-malspam-pushing-redaman-banking-malware/,https://github.com/eset/malware-ioc/blob/master/rtm/samples.md5,https://github.com/eset/malware-ioc/blob/master/rtm/samples.sha1,https://github.com/eset/malware-ioc/blob/master/rtm/samples.sha256
SaintBear	2021-2022	https://cert.gov.ua/article/18273,https://cert.gov.ua/article/37704,https://cert.gov.ua/article/39882	https://cert.gov.ua/article/18273,https://cert.gov.ua/article/37704,https://cert.gov.ua/article/39882
Sandworm Team	2014-2022	https://cert.gov.ua/article/39518,https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE4Vwwd,https://www.welivesecurity.com/2022/05/20/sandworm-ukraine-new-version-arguepatch-malware-loader/	https://cert.gov.ua/article/39518,https://www.welivesecurity.com/2022/05/20/sandworm-ukraine-new-version-arguepatch-malware-loader/
Scarab	2012-2015, 2020, 2022	https://cert.gov.ua/article/38097,https://web.archive.org/web/20150124025612/http:/www.symantec.com:80/connect/blogs/scarab-attackers-took-aim-select-russian-targets-2012	https://cert.gov.ua/article/38097
Silence	2017-2020	https://www.group-ib.com/resources/threat-research/silence_2.0.going_global.pdf,https://www.group-ib.com/resources/threat-research/silence_moving-into-the-darkside.pdf	https://securelist.com/the-silence/83009/,https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Silence/Silence_hash.md
Snowglobe	2012-2013	https://securelist.com/animals-in-the-apt-farm/69114/	https://www.welivesecurity.com/2015/03/05/casper-malware-babar-bunny-another-espionage-cartoon/,https://www.welivesecurity.com/2015/06/30/dino-spying-malware-analyzed/,https://unit42.paloaltonetworks.com/unit42-analysing-10-year-old-snowball/
Sofacy	2014-2022	https://www.mandiant.com/sites/default/files/2021-09/APT28-Center-of-Storm-2017.pdf,https://symantec-enterprise-blogs.security.com/blogs/election-security/apt28-espionage-military-government,https://blogs.microsoft.com/on-the-issues/2022/04/07/cyberattacks-ukraine-strontium-russia/	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/APT28/APT28_hash.md,https://symantec-enterprise-blogs.security.com/sites/default/files/2018-10/APT28_IOCs.txt
TeamSpy Crew	2011-2013	https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/20134928/theteamspystory_final_t2.pdf,https://www.crysys.hu/publications/files/technical-reports/teamspy/teamspy.pdf	?
TeleBots	2014-2022	https://www.welivesecurity.com/2017/06/30/telebots-back-supply-chain-attacks-against-ukraine/,https://www.welivesecurity.com/2018/10/11/new-telebots-backdoor-linking-industroyer-notpetya/,https://www.eset.com/int/about/newsroom/press-releases/announcements/eset-research-ukraine-and-poland-targeted-by-sophisticated-blackenergy-trojan/	https://www.welivesecurity.com/2017/06/30/telebots-back-supply-chain-attacks-against-ukraine/,https://github.com/eset/malware-ioc/blob/master/telebots/samples.md5,https://github.com/eset/malware-ioc/blob/master/telebots/samples.sha1,https://github.com/eset/malware-ioc/blob/master/telebots/samples.sha256,https://www.welivesecurity.com/2016/01/03/blackenergy-sshbeardoor-details-2015-attacks-ukrainian-news-media-electric-industry/
Tortilla	2021	https://blog.talosintelligence.com/2021/11/babuk-exploits-exchange.html	https://blog.talosintelligence.com/2021/11/babuk-exploits-exchange.html
Traveling Spider	2019	https://www.bleepingcomputer.com/news/security/new-nemty-ransomware-may-spread-via-compromised-rdp-connections/	https://www.trendmicro.com/vinfo/fr/security/news/cyber-attacks/nemty-ransomware-possibly-spreads-through-exposed-remote-desktop-connections,https://any.run/malware-trends/nemty

Кінець таблиці Б.1

Turla	2007-2022	https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/,https://blog.talosintelligence.com/2021/09/tinyturla.html	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Turla/Turla_hash.md,https://github.com/eset/malware-ioc/blob/master/turla/samples.md5,https://github.com/eset/malware-ioc/blob/master/turla/samples.sha1,https://github.com/eset/malware-ioc/blob/master/turla/samples.sha256,https://blog.talosintelligence.com/2021/09/tinyturla.html
Vicious Panda	2019	https://research.checkpoint.com/2020/vicious-panda-the-covid-campaign/	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/Vicious%20Panda/Vicious%20Panda_hash.md,https://bbs.pediy.com/thread-256810.htm
XDSpy	2011-2022	https://www.welivesecurity.com/2020/10/02/xdspy-stealing-government-secrets-since-2011/,https://cip.gov.ua/en/news/khto-stoyit-za-kiberatakami-na-ukrayinsku-kritichnu-informaciinu-infrastrukturu-statistika-15-22-bereznya	https://github.com/RedDrip7/APT_Digital_Weapon/blob/master/XDSpy/XDSpy_hash.md,https://github.com/eset/malware-ioc/blob/master/xdspy/samples.sha1,https://github.com/eset/malware-ioc/blob/master/xdspy/samples.sha256
Dark Basin	2017-2020	https://www.nortonlifelock.com/blogs/security-response/mercenary-amanda-professional-hackers-hire	?
Scully Spider	2018-2022	https://www.cisa.gov/uscert/ncas/alerts/aa22-110a,https://www.proofpoint.com/us/blog/threat-insight/new-year-new-version-danabot	https://www.proofpoint.com/us/blog/threat-insight/new-year-new-version-danabot,https://www.zscaler.com/blogs/security-research/spike-danabot-malware-activity
TA511	2017	https://unit42.paloaltonetworks.com/threat-brief-hancitor-actors/	https://unit42.paloaltonetworks.com/hancitor-infections-cobalt-strike/,https://unit42.paloaltonetworks.com/ta551-shathak-icedid/
Yingmob	2016	https://blog.checkpoint.com/wp-content/uploads/2016/07/HummingBad-Research-report_FINAL-62916.pdf	https://digital.nhs.uk/cyber-alerts/2017/cc-1094,https://blog.checkpoint.com/2017/01/23/hummingbad-returns/