МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА УПРАВЛІННЯ КІБЕРБЕЗПЕКОЮ ТА ЗАХИСТОМ ІНФОРМАЦІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: “СИСТЕМА АВТОМАТИЗОВАНОГО АНАЛІЗУ ЦИФРОВИХ АРТЕФАКТІВ ІЗ ВИКОРИСТАННЯМ ІСНУЮЧИХ FORENSIC-ІНСТРУМЕНТІВ”

на здобуття освітнього ступеня бакалавра
зі спеціальності 125 Кібербезпека
освітньої програми Управління інформаційною та кібернетичною безпекою

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

_____ Олександр КОСИНСЬКИЙ
(підпис) *Ім’я, ПРІЗВИЩЕ здобувача*

Виконав(ла): здобувач вищої освіти гр. УБД-42

Олександр КОСИНСЬКИЙ
Ім’я, ПРІЗВИЩЕ

Керівник: Ірина ЛОЗОВА
к.т.н. Ім’я, ПРІЗВИЩЕ

Рецензент: Сергій ГАХОВ
к.в.н., доцент Ім’я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут кібербезпеки та захисту інформації

Кафедра управління кібербезпекою та захистом інформації

Ступінь вищої освіти бакалавр

Спеціальність 125 Кібербезпека

Освітня програма Управління інформаційною та кібернетичною безпекою

ЗАТВЕРДЖУЮ
Завідувач кафедри УКБЗІ
_____ Світлана ЛЕГОМІНОВА
“____”_____2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Косинському Олександру Олеговичу
(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи “Система автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструментів”,
керівник кваліфікаційної роботи Лозова Ірина, к.т.н.,
(ІПРІЗВИЩЕ, Ім’я,, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20“ лютого 2026 р. №51

2. Строк подання кваліфікаційної роботи “01” червня 2026р.

3. Вихідні дані до кваліфікаційної роботи:

4. Перелік питань, які мають бути розроблені:
   4.1 Проаналізувати теоретичні основи цифрової криміналістики та класифікацію цифрових артефактів.
   4.2 Дослідити існуючі forensic-інструменти та їх можливості для аналізу цифрових артефактів.
   4.3 Розробити архітектуру та програмну реалізацію системи автоматизованого аналізу цифрових артефактів із інтеграцією існуючих forensic-інструментів.

5. Перелік ілюстративного матеріалу: *презентація PowerPoint*

6. Дата видачі завдання “20“ лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Визначення об’єкту, предмету, мети та завдань дослідження.	22.04.2026	
2.	Збір та аналіз літератури.	30.04.2026	
3.	Аналіз теоретичних основ цифрової криміналістики	02.05.2026	
4.	Дослідження існуючих forensic-інструментів	07.05.2026	
5.	Розроблення архітектури та системи автоматизованого аналізу цифрових артефактів із forensic-інструментами	15.05.2026	
6.	Формулювання висновків	26.05.2026	
7.	Оформлення роботи.	27.05.2026	
8.	Оформлення презентації	01.06.2026	
9.	Отримання рецензії на роботу.	5.06.2026	
10.	Захист в ДЕК.	10.06.2026	

Здобувач вищої освіти _____ Олександр КОСИНСЬКИЙ

(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ірина ЛОЗОВА

(підпис) *(Ім’я, ПРІЗВИЩЕ)*

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

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня бакалавра

Направляється здобувач Косинський О. О. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 125 Кібербезпека
(код, найменування спеціальності)
освітньої програми Управління інформаційною та кібернетичною безпекою
(назва)
на тему: “Система автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструмент”
Кваліфікаційна робота і рецензія додаються.

Директор ННІЗІ _____ Євгенія ІВАНЧЕНКО
(підпис) (Ім’я, ПРІЗВИЩЕ)

Висновок керівника кваліфікаційної роботи

Здобувач КОСИНСЬКИЙ Олександр у кваліфікаційній роботі проаналізував особливості автоматизованого аналізу цифрових артефактів, проаналізувала можливості сучасних forensic-інструментів та розробила систему для автоматизації процесу аналізу цифрових доказів.

КОСИНСЬКИЙ Олександр показав розуміння проблеми дослідження та бачення основних теоретичних і практичних напрямів її вирішення, довів володіння методами наукового дослідження, проявив себе як організований, відповідальний виконавець.

Все це дозволяє оцінити кваліфікаційну роботу здобувача КОСИНСЬКОГО Олександра на оцінку “відмінно” та присвоїти йому кваліфікацію бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою.

Керівник кваліфікаційної роботи ____ Ірина ЛОЗОВА
(підпис) (Ім’я, ПРІЗВИЩЕ)
“_____“_____2026 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Косинський О. О. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри управління кібербезпекою та захистом інформації Світлана ЛЕГОМІНОВА

____

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну бакалаврську роботу

здобувача вищої освіти КОСИНСЬКОГО Олександра
на тему “Система автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструмент”

Актуальність. В умовах стрімкого зростання кількості кіберінцидентів та ускладнення методів цифрових атак питання оперативного й точного дослідження цифрових доказів набуває першочергового значення. Обсяги даних, що підлягають forensic-аналізу, невпинно зростають, а ручна обробка артефактів стає дедалі менш ефективною. Автоматизація аналітичних процесів на основі інтеграції існуючих forensic-інструментів дозволяє не лише пришвидшити розслідування інцидентів, а й підвищити достовірність та відтворюваність отриманих результатів. Побудова єдиної системи, здатної об'єднати різнорідні інструменти у злагоджений аналітичний конвеєр, є перспективним і практично значущим напрямом досліджень у сфері кібербезпеки. З огляду на зазначене дослідження проблеми автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструментів є актуальним науковим завданням.

Позитивні сторони.

1. У роботі досліджено можливості практичного використання існуючих forensic-інструментів у рамках єдиного програмного середовища
2. Кваліфікаційна робота оформлена відповідно до вимог. Виклад матеріалу здійснено відповідно до плану, зроблено логічні висновки. Ключові положення роботи проілюстровано у вигляді схем та таблиць.
3. Автор опрацював значну джерельну базу: близько 50 публікацій, в тому числі англомовних.
4. За результатами дослідження запропоновано рекомендації щодо розгортання та застосування розробленої системи в реальних умовах forensic-розслідувань

Недоліки.

Доцільно було б приділити більше уваги вивченню і класифікації програмних інструментів для оцінки ефективності розроблених програм бізнаності та навчання персоналу з питань інформаційної безпеки.

Однак, вищезгадані зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Кваліфікаційна робота виконана на належному науково-методичному рівні і заслуговує оцінки “відмінно”, а здобувач КОСИНСЬКИЙ Олександр заслуговує присвоєння кваліфікації бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою

Рецензент: Сергій ГАХОВ

к.в.н., доцент *підпис* Ім’я, ПРІЗВИЩЕ

РЕФЕРАТ

Кваліфікаційна робота присвячена дослідженню системи автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструментів.

Метою роботи є розроблення системи автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструментів.

Об'єктом дослідження є процес аналізу цифрових артефактів під час розслідування кіберінцидентів.

Предмет дослідження – методи та засоби автоматизованого аналізу цифрових артефактів із застосуванням існуючих forensic-інструментів.

Методи дослідження. Для вирішення означеного вище наукового завдання в роботі використані методи аналізу та синтезу, порівняння та класифікації інструментів, системного підходу до проектування програмного забезпечення, а також методи прикладного програмування та тестування.

Як результат у роботі проаналізовано особливості управління інформаційною безпекою підприємства, досліджено основні існуючі forensic-інструменти та їх можливості для аналізу цифрових артефактів; вивчено інструменти та методи розроблено архітектуру та програмну реалізацію системи автоматизованого аналізу цифрових артефактів.

Галузь застосування. Розроблені підходи можуть бути використані при плануванні та реалізації системи управління інформаційною безпекою підприємства у контексті автоматизованого аналізу цифрових артефактів.

Ключові слова: ЦИФРОВА КРИМІНАЛІСТИКА, ЦИФРОВІ АРТЕФАКТИ, FORENSIC-ІНСТРУМЕНТИ, АВТОМАТИЗОВАНИЙ АНАЛІЗ, КІБЕРІНЦИДЕНТИ, СИСТЕМА АНАЛІЗУ ДОКАЗІВ.

ABSTRACT

The qualification work is devoted to the development and research of an automated digital artifact analysis system using existing forensic tools.

The purpose of the study is to investigate the principles of information security awareness and training for personnel.

The object the study is the principles of awareness and training for personnel.

The subject of the study is the peculiarities of applying technologies of information security awareness and training for personnel.

Research methods. In order to solve the mentioned higher scientific task, the methods of analysis and synthesis, comparison, classification, expert assessment, systematic approach to information security management were used in the work.

As a result, the theoretical foundations of digital forensics and classification of digital artifacts were analyzed; existing forensic tools and their capabilities were investigated; an architecture and software implementation of an automated digital artifact analysis system with forensic tool integration was developed.

Field of application. The developed system can be used in cyber response units, cybercrime departments, and organizations conducting digital forensic investigations.

Keywords: DIGITAL FORENSICS, DIGITAL ARTIFACTS, FORENSIC TOOLS, AUTOMATED ANALYSIS, CYBER INCIDENTS, EVIDENCE ANALYSIS SYSTEM.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

IoT – Internet of Things

DEFR - Digital Evidence First Responder

DES - Digital Evidence Specialist

RFC - Request for Comments 3227 - Guidelines for Evidence Collection and Archiving

КПК - Кримінальний процесуальний кодекс України

FTK - Forensic Toolkit

TSK - The Sleuth Kit

NIST SP 800-86 - Guide to Integrating Forensic Techniques into Incident Response

ISO/IEC 27037 - Міжнародний стандарт поводження з цифровими доказами

MITRE ATT&CK - Adversarial Tactics, Techniques, and Common Knowledge

MISP/STIX2 - Malware Information Sharing Platform/Structured Threat Information eXpression v2

IoC - Indicator of Compromise

IoA - Indicator of Attack

CoC - Chain of Custody

ВСТУП

Стрімкий розвиток інформаційних технологій та повсюдне впровадження цифрових систем у критичну інфраструктуру держави, фінансовий сектор і сферу державного управління зумовлюють принципову зміну характеру сучасних кіберзагроз. Кібератаки набувають дедалі більш комплексного, цілеспрямованого та прихованого характеру, а їх виявлення й розслідування потребує застосування спеціалізованих методів і засобів цифрової криміналістики. Показовим індикатором цієї тенденції є статистика Команди реагування на комп'ютерні надзвичайні події України (CERT-UA): у 2024 році опрацьовано 4 315 кіберінцидентів, що на 70 % перевищує показник 2023 року. Наведені дані переконливо свідчать про те, що ручні методи збору та аналізу цифрових доказів об'єктивно не встигають за зростаючим обсягом і складністю сучасних загроз.

Цифрова криміналістика як науково-прикладна галузь забезпечує методологічну основу для ідентифікації, збору, збереження, аналізу та документування цифрових доказів. Проте практичне застосування існуючих forensic-інструментів пов'язане з низкою системних проблем: відсутністю автоматичного дотримання порядку збору артефактів за ступенем волатильності, несумісністю форматів виходу різних інструментів, відсутністю механізмів кореляції між результатами окремих засобів аналізу, а також значними часовими витратами на ручне зведення знахідок із множини джерел. Зазначені проблеми обумовлюють необхідність розробки інтегрованої автоматизованої системи, здатної виконувати повний цикл forensic-аналізу в єдиному конвеєрі з дотриманням вимог міжнародних стандартів.

Актуальність теми дослідження визначається поєднанням двох чинників: критичним зростанням кількості кіберінцидентів в Україні в умовах воєнного стану та об'єктивною неспроможністю наявних

ізольованих forensic-інструментів забезпечити своєчасне автоматизоване реагування на комплексні кіберзагрози без значних ручних витрат аналітика.

Метою цієї роботи є проектування та практична реалізація системи автоматизованого аналізу цифрових артефактів, що відповідає вимогам міжнародних стандартів цифрової криміналістики, автоматично дотримується порядку волатильності при зборі артефактів та забезпечує крос-модульну кореляцію знахідок для виявлення комплексних кіберзагроз.

Об'єктом дослідження є процеси виявлення, збору та аналізу цифрових артефактів у ході розслідування кіберінцидентів.

Предметом дослідження є методи та засоби автоматизованого аналізу цифрових артефактів на основі інтегрованого forensic-конвеєру.

Методи дослідження. У роботі використано: системний аналіз - для дослідження існуючих forensic-інструментів та виявлення їх прогалин; метод класифікації - для систематизації цифрових артефактів за трьома вимірами; метод порівняльного аналізу - для оцінювання альтернативних рішень; методи об'єктно-орієнтованого проектування - для розробки модульної архітектури системи; метод сигнатурного аналізу - для розробки YARA-правил із прив'язкою до матриці MITRE ATT&CK; метод інтеграційного тестування - для верифікації коректності роботи системи.

Новизна отриманих результатів полягає в такому:

- Удосконалено підхід до автоматизованого цифрового криміналістичного аналізу шляхом реалізації трирівневої модульної системи DFAS із конвеєром обробки відповідно до RFC 3227, механізмом міжмодульної кореляції результатів та автоматичним веденням chain of custody, що забезпечило виявлення комплексних загроз, недоступних для ізольованих інструментів, і підвищило оперативність аналізу цифрових артефактів.

Практичне значення отриманих результатів полягає у розробці повністю функціонального програмного комплексу DFAS, готового до розгортання в навчальному та дослідницькому середовищі. Система реалізована на відкритих бібліотеках без ліцензійних обмежень, підтримує файли до 600 МБ, забезпечує аналіз у межах 6 модулів із автоматичним формуванням JSON-звіту та числового ризик-балу (0-100), що дозволяє об'єктивно порівнювати критичність різних артефактів.

Апробація результатів кваліфікаційної роботи відбулася на Всеукраїнській науково-практичній конференції «Стратегії кіберстійкості: управління ризиками та безперервність бізнесу» 25 лютого 2026 року, де було представлено тези доповіді «Система автоматизованого аналізу цифрових артефактів із використанням існуючих forensic-інструментів»[48].

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВОЇ КРИМІНАЛІСТИКИ ТА КЛАСИФІКАЦІЯ ЦИФРОВИХ АРТЕФАКТІВ

1.1 Сутність, принципи і методи цифрової криміналістики

1.1.1 Поняття та предмет цифрової криміналістики

Цифрова криміналістика (digital forensics) – це галузь, яка займається пошуком, збором, збереженням і аналізом цифрових доказів [5]. Якщо простіше – це спроба зрозуміти, що сталося в комп'ютерній системі, спираючись на те, що там залишилось. А залишається завжди більше, ніж думає зловмисник.

Термін «digital forensics» набув поширення наприкінці 1990-х – на початку 2000-х років і поступово витіснив вужче поняття «computer forensics», що стосувалося виключно аналізу комп’ютерних систем. Сучасна цифрова криміналістика має значно ширшу область застосування: вона поширюється на мобільні пристрої, мережеву інфраструктуру, хмарні середовища, вбудовані системи, пристрої Інтернету речей (IoT) та промислові системи керування (ICS/SCADA) [6].

Цифрова криміналістика охоплює широкий спектр спеціалізованих напрямів: комп’ютерна криміналістика (disk forensics), криміналістика оперативної пам’яті (memory forensics), мережева криміналістика (network forensics), мобільна криміналістика (mobile forensics), хмарна криміналістика (cloud forensics), криміналістика вбудованих систем та аналіз шкідливого програмного забезпечення (malware forensics). Кожен із цих напрямів характеризується специфічними методами збору та аналізу артефактів, відповідними інструментальними засобами та особливими вимогами до збереження цілісності доказів [7].

З юридичної точки зору, цифрова криміналістика тісно пов’язана з доктриною допустимості доказів: результати криміналістичного аналізу мають задовольняти критерії автентичності, повноти, достовірності та

відповідності встановленим процедурам збору й зберігання. Порушення будь-якого з цих критеріїв може зробити цифровий доказ недопустимим у судовому провадженні. Це висуває жорсткі вимоги до методів роботи і є одним із ключових чинників, що зумовлюють необхідність стандартизованих і документованих підходів до аналізу артефактів [8].

1.1.2 Основні принципи цифрової криміналістики

Наукова спільнота та міжнародні організації зі стандартизації виробили низку фундаментальних принципів, дотримання яких є обов'язковим під час проведення криміналістичного аналізу. Ці принципи формують основу будь-якої криміналістичної процедури та мають бути реалізовані на рівні програмної системи автоматизованого аналізу.

Принцип цілісності (Integrity). Простіше кажучи – нічого не псувати. Для цього перед будь-якою обробкою рахують хеш (MD5, SHA-256 тощо), і потім перевіряють, що він не змінився. Здається очевидним, але на практиці про це часто забувають при ручній роботі. Автоматизована система має робити це сама, на кожному етапі.

Принцип автентичності (Authenticity). Докази мають бути справжніми, а не підробленими чи модифікованими. Документування ланцюжка зберігання (chain of custody) є ключовим механізмом забезпечення автентичності. Кожна дія з артефактом фіксується з часовою міткою та ідентифікатором суб'єкта, що виконав дію.

Принцип відтворюваності (Reproducibility). Будь-яка процедура аналізу повинна давати однакові результати при повторному виконанні за тих самих умов. Стандарт ISO/IEC 27037 наголошує, що управління цифровими доказами має охоплювати придатність до аудиту, обґрунтованість, а також відтворюваність або повторюваність залежно від конкретних умов проведення дослідження [9].

Принцип мінімального втручання (Minimal Interference). Під час збору цифрових доказів необхідно мінімізувати вплив на оригінальні дані. Якщо

уникнути змін неможливо (наприклад, при роботі з енергозалежною пам'яттю), усі дії документуються з обґрунтуванням необхідності. В автоматизованих системах цей принцип реалізується режимом «read-only» при роботі з носіями.

Принцип ланцюжка зберігання (Chain of Custody). Кожен крок поводження з доказами -від виявлення до представлення в суді -має бути хронологічно задокументований. Журнал аудиту (audit log) є обов'язковим компонентом системи автоматизованого аналізу.

Принцип об'єктивності (Objectivity). Криміналістичний аналіз має бути неупередженим. Аналітик і автоматизована система мають слідувати доказам, а не намагатися підтвердити заздалегідь висунуту версію. Результати аналізу повинні містити всі виявлені факти, незалежно від їх відповідності початковій гіпотезі.

1.1.3 Процес цифрового криміналістичного розслідування

Процес криміналістичного розслідування традиційно описується у вигляді послідовності взаємопов'язаних етапів. Стандарт NIST SP 800-86 «Guide to Integrating Forensic Techniques into Incident Response» окреслює структурований підхід, що охоплює ідентифікацію, збереження, збір, дослідження, аналіз та документування цифрових доказів, наголошуючи на важливості дотримання ланцюжка зберігання на кожному етапі [10].

Етап 1. Ідентифікація (Identification). Спочатку – зрозуміти, де взагалі шукати. Фізичні диски, RAM, мережеві пристрої, хмара, мобільники – список може бути довгим. Головне на цьому етапі – розставити пріоритети: що найбільш волатильне, що найбільш релевантне для конкретного інциденту.

Етап 2. Збереження (Preservation). Забезпечення незмінності доказів. Для енергонезалежних носіїв застосовується апаратне або програмне блокування запису (write blocker). Для енергозалежної пам'яті (RAM) виконується її «жива» копія (live acquisition) до вимкнення системи,

оскільки вона містить унікальні артефакти, недоступні після перезавантаження.

Етап 3. Збір (Collection). Криміналістичне копіювання (forensic imaging) носіїв інформації з документуванням хеш-значень. Збір здійснюється відповідно до порядку волатильності RFC 3227: спочатку - регістри процесора та кеш, оперативна пам’ять; далі – тимчасові файли, мережеві з’єднання; останніми – дискові носії та резервні копії [11].

Етап 4. Дослідження (Examination). Відновлення видалених файлів, розпакування архівів, аналіз файлових систем, пошук прихованих і зашифрованих даних. На цьому етапі здійснюється первинна фільтрація зібраних артефактів та їх категоризація. 18амец ей етап є основним об’єктом автоматизації в системі, що розробляється.

Етап 5. Аналіз (Analysis). Знайдені артефакти треба ще й зрозуміти. Timeline analysis, зіставлення з IoC з MISP чи VirusTotal, реконструкція того, що саме відбулось і в якому порядку. Тут людина ще потрібна – але система може суттєво скоротити час на підготовку даних для цього аналізу.

Етап 6. Документування та звітування (Reporting). Звіт – це не просто «для галочки». Його мають розуміти і технічні аналітики, і суддя, і керівник організації. Тому важливо зберігати і методологію, і виявлені артефакти, і весь ланцюжок зберігання. Автоматизована система формує основу для такого звіту автоматично.

1.2 Правові та методологічні засади цифрової криміналістики

1.2.1 Правова база в Україні

Технічне рішення – це лише половина справи. Система, що розробляється, має відповідати і правовим вимогам, інакше зібрані нею докази просто не матимуть юридичної ваги. Тому варто коротко розглянути, яка правова база регулює цифрові розслідування в Україні.

Базовим є Закон України «Про основні засади забезпечення кібербезпеки України» від 05.10.2017 № 2163-VIII, який визначає правові та організаційні основи забезпечення кібербезпеки держави, встановлює повноваження суб'єктів, відповідальних за кіберзахист, та регламентує діяльність CERT-UA. Закон визначає поняття кіберінциденту, кібератаки та встановлює обов'язки операторів об'єктів критичної інфраструктури щодо повідомлення про інциденти [12].

Кримінальний процесуальний кодекс України зі змінами визначає статус електронних доказів, порядок їх збору та долучення до матеріалів справи. Стаття 99 КПК визначає документи, у тому числі електронні, як самостійний вид доказів. Стаття 237 регулює огляд речей і документів, що застосовується і до цифрових носіїв інформації [13].

Постанова КМУ від 23.12.2020 № 1295 «Деякі питання забезпечення функціонування системи виявлення вразливостей і реагування на кіберінциденти та кібератаки» регламентує порядок функціонування відповідної системи на загальнодержавному рівні, визначаючи порядок збору та опрацювання даних про інциденти[14]. Указом Президента України №447/2021 передбачено залучення приватних експертів до проведення комп'ютерно-технічних досліджень та експертиз, необхідних для швидкого реагування на кіберінциденти [15].

Важливу роль відіграє також Закон України «Про електронні документи та електронний документообіг» від 22.05.2003 № 851-IV, що встановлює правову основу для визначення юридичної сили електронних документів, та Закон «Про захист персональних даних», що обмежує коло осіб, які мають право доступу до цифрових доказів, що містять персональну інформацію [16].

1.2.2 Стандарт NIST SP 800-86

NIST SP 800-86 – це, мабуть, найпрактичніший з усіх документів, які були опрацювані при написанні цієї роботи. Американський NIST написав

його як методичний посібник для тих, хто займається розслідуванням інцидентів. Стандарт охоплює чотири основні категорії криміналістичних джерел даних [10]:

- дані операційної системи – системні журнали (event logs), реєстр Windows (Registry), тимчасові файли, облікові записи користувачів, файли конфігурацій;
- мережеві дані – журнали мережевого обладнання, пакети мережевого трафіку у форматі PCAP, NetFlow/IPFIX-записи, журнали DNS-запитів;
- дані застосунків – журнали вебсерверів, бази даних, файли конфігурацій, кеш браузерів, журнали поштових клієнтів;
- дані носіїв інформації – файлові системи, видалені файли, нерозподілений простір диска, завантажувальні записи (MBR, GPT).

На відміну від ISO/IEC 27037, який зосереджується переважно на початкових етапах збору, NIST SP 800-86 охоплює весь цикл – від ідентифікації до фінального звіту [17].

1.2.3 Стандарт ISO/IEC 27037

ISO/IEC 27037:2012 – це міжнародний стандарт, який зосереджений суто на початкових фазах: ідентифікація, збір, збереження цифрових доказів [9]. Він не охоплює повний цикл розслідування, але чітко прописує критерії якості і вводить важливий розподіл ролей.

Стандарт виділяє дві ролі: DEFR (Digital Evidence First Responder) – той, хто виїжджає на місце і збирає докази, і DES (Digital Evidence Specialist) – той, хто потім їх аналізує в лабораторних умовах. Система, що розробляється, фактично автоматизує частину роботи DES – рутинний аналіз і категоризацію артефактів.

ISO/IEC 27037 у контексті управління цифровими доказами охоплює кілька критичних аспектів: придатність до аудиту, обґрунтованість, а також відтворюваність або повторюваність залежно від конкретних польових

умов. Стандарт також надає настанови щодо особливих сценаріїв, зокрема роботи з зашифрованими носіями, хмарними даними та мережевими артефактами [9].

1.2.4 Порівняльний аналіз стандарті

Обидва стандарти корисні, але під різні цілі [17]. NIST SP 800-86 – більш практичний і технічний, орієнтований на повний цикл розслідування. ISO/IEC 27037 – більш процедурний, прийнятий у правоохоронних органах і промисловості. Для цієї роботи вони доповнюють одне одного. Детальне порівняння – в таблиці 1.1.

Табл. 1.1. Порівняння стандартів NIST SP 800-86 та ISO/IEC 27037

Критерій	*NIST SP 800-86*	*ISO/IEC 27037*
Охоплення процесу	Повний цикл розслідування	Початкові етапи (збір, збереження)
Сфера застосування	Академічні кола, реагування на інциденти	Промисловість, правоохоронні органи
Ролі учасників	Не визначені окремо	DEFR, DES
Критерії якості	Відтворюваність, точність	Аудитабельність, обґрунтованість, відтворюваність
Орієнтація	Технічна, процесна	Процедурна, юридична

Для цілей цієї кваліфікаційної роботи розроблювана система орієнтується на методологію NIST SP 800-86 як більш комплексну, з урахуванням процедурних вимог ISO/IEC 27037 щодо забезпечення цілісності артефактів та їх придатності до використання як доказів. Така комбінація дозволяє охопити повний цикл від збору до звітування та відповідати вимогам як технічних, так і правових стандартів.

1.3 Огляд і класифікація цифрових артефактів

1.3.1 Поняття цифрового артефакту

У контексті цифрової криміналістики під цифровим артефактом розуміють будь-який цифровий об’єкт або фрагмент даних, що може

слугувати доказом або надавати інформацію, релевантну для розслідуваного інциденту. Цифрові докази можуть походити з різноманітних джерел, зокрема з платформ електронної пошти, сервісів електронних грошей, флеш-накопичувачів,

Важливо розрізняти поняття цифрового хмарних провайдерів та електронних пристроїв артефакту та цифрового доказу[6]. Перше є більш широким і охоплює будь-які цифрові об’єкти, що можуть бути корисними для аналізу. Цифровий доказ – це артефакт, долучений до матеріалів справи у встановленому законом порядку. Таким чином, кожен цифровий доказ є артефактом, але не кожен артефакт автоматично стає доказом. Це розмежування є критично важливим для розуміння предмета дослідження та визначення вимог до системи автоматизованого аналізу [5].

1.3.2 Класифікація за типом носія

Для практичної роботи зручно класифікувати артефакти за тим, звідки вони беруться. Ми виділили шість основних класів – і для кожного з них в системі передбачений відповідний модуль обробки.

1. *Артефакти файлової системи та дискових носіїв.* Це найбільш досліджений клас артефактів, що охоплює: активні та видалені файли й директорії; метадані файлової системи (часові мітки MAC – Modified, Accessed, Created); нерозподілений простір диска (unallocated space) та простір між файлами (file slack); завантажувальні записи (MBR, GPT, VBR); таблиці файлової системи (MFT для NTFS, inode-таблиці для ext4, FAT для FAT32). Дискове знімання (disk imaging) передбачає створення побітової копії носія, що забезпечує збереження оригінальних даних при роботі аналітиків із копією [18].

2. *Артефакти оперативної пам’яті (Memory Artifacts).* Оперативна пам’ять є надзвичайно цінним, але волатильним джерелом. Вона може містити: запущені процеси та завантажені

бібліотеки (DLL); мережеві з'єднання та відкриті сокети; ключі шифрування та паролі у відкритому вигляді; фрагменти шкідливого ПЗ (у тому числі безфайлового); дані буфера обміну та вміст буфера введення. Аналіз оперативної пам'яті набув особливого значення у зв'язку з поширенням «безфайлових» атак (fileless malware), за яких шкідливий код функціонує виключно в пам'яті [19].

3. *Мережеві артефакти (Network Artifacts).* До цієї категорії належать: записи мережевого трафіку у форматі PCAP (повний захват пакетів); NetFlow/IPFIX-записи (метадані з'єднань без вмісту); журнали DNS-запитів; журнали мережевого обладнання (брандмауери, проксі-сервери, IDS/IPS); HTTP(S) – запити та відповіді; журнали VPN-з'єднань. Мережеві артефакти дозволяють відновити хронологію зовнішніх з'єднань, ідентифікувати C2-сервери та відстежити переміщення даних у мережі [20].

4. *Артефакти операційної системи (OS Artifacts).* Операційна система генерує значну кількість артефактів, що відображають діяльність користувачів і системи. Для Windows: реєстр (Registry) -містить конфігурацію системи, відомості про встановлені програми та профілі користувачів; журнали подій (Event Logs, формат EVTX); файл підкачки (pagefile.sys) і файл гібернації (hiberfil.sys); Prefetch-файли та SuperFetch; LNK-файли (ярлики) та списки переходів (Jump Lists); Shellbag-записи в реєстрі. Для Linux: журнали syslog, auth.log, kern.log; bash-history (.bash_history); cron-завдання та записи /proc і /sys [21].

5. *Артефакти застосунків (Application Artifacts).* Кожний застосунок генерує власний набір артефактів: веббраузери -історія відвідувань, завантажень, файли cookie, кеш, збережені паролі; поштові клієнти -повідомлення, вкладення, адресні книги; месенджери – журнали переписки, медіафайли, метадані

повідомлень; офісні застосунки -нещодавно відкриті файли, тимчасові копії, метадані документів (автор, дата редагування, версія). Артефакти застосунків є одними з найбільш інформативних для встановлення дій конкретного користувача [22].

6. *Хмарні та мобільні артефакти.* З поширенням хмарних технологій і мобільних пристроїв цей клас набуває дедалі більшого значення. Хмарні артефакти включають журнали доступу до сховищ (AWS S3, Google Drive, OneDrive), журнали аутентифікації та API-запити. Мобільні артефакти охоплюють бази даних застосунків (SQLite), журнали дзвінків та SMS, дані геолокації, резервні копії. Ключовою проблемою залишаються юрисдикційні обмеження: дані, збережені в іншій країні, підпадають під законодавство цієї держави, що може ускладнювати або унеможливлювати доступ [23].

1.3.3 Класифікація за ступенем волатильності

Практично важливою є класифікація за ступенем волатильності (мінливості) артефактів – тим, як швидко вони зникають або змінюються при відключенні живлення чи перезавантаженні системи. Відповідно до порядку волатильності, визначеного в RFC 3227 «Guidelines for Evidence Collection and Archiving», артефакти ранжуються від найбільш до найменш волатильних, що визначає пріоритетність збору [11].

Табл 1.2. Порядок волатильності цифрових артефактів (RFC 3227)

Рівень волатильності	Тип артефакту	Час збереження
Найвищий	Регістри процесора, кеш CPU	Миттєво зникають
Дуже високий	Оперативна пам’ять (RAM)	Секунди – хвилини
Високий	Мережеві з’єднання, процеси, ARP-кеш	Хвилини
Середній	Тимчасові файли, файли підкачки	Години – дні
Низький	Журнали подій, файли диска, реєстр	Тижні – місяці

Продовження табл.1.2

Рівень волатильності	Тип артефакту	Час збереження
Найнижчий	Резервні копії, архівні носії	Роки

При проектуванні системи ми виходили саме з цього: модуль збору спочатку намагається зняти дамп пам'яті та зафіксувати мережеві з'єднання, і лише потім переходить до більш «стабільних» джерел. Чекати на завершення дискового знімання, поки «живі» артефакти зникають - помилка, яку автоматизація має виключати.

1.3.4 Класифікація за роллю у розслідуванні

Корисно дивитись на артефакти не тільки за тим, звідки вони, але й за тим, що вони «говорять» про інцидент. Тут виділено чотири категорії [24].

1. Індикатори компрометації (IoC, Indicators of Compromise) - специфічні ознаки, що свідчать про факт проникнення або зараження: хеші шкідливих файлів, IP-адреси C2-серверів, специфічні ключі реєстру, домени, що використовуються в атаках. IoC є основним типом артефактів для зіставлення з зовнішніми базами даних загроз (MISP, VirusTotal, Shodan).
2. Індикатори атаки (IoA, Indicators of Attack) - ознаки, що вказують на поточну атаку або підготовку до неї: аномальні мережеві з’єднання, спроби ескалації привілеїв, горизонтальне переміщення (lateral movement). На відміну від IoC, IoA відображають поведінкові патерни, а не конкретні артефакти.
3. Артефакти виконання (Execution Artifacts) - свідчення запуску конкретних програм: Prefetch-файли (Windows), записи AmCache та ShimCache, UserAssist-ключі реєстру, журнали Sysmon (якщо активований). Ці артефакти є ключовими для встановлення факту запуску шкідливого ПЗ навіть після його видалення.

4. Артефакти персистентності (Persistence Artifacts) - сліди механізмів закріплення шкідливого ПЗ: ключі автозапуску реєстру (Run, RunOnce), заплановані завдання (Scheduled Tasks), служби Windows, модифікації файлу hosts, cron-завдання в Linux. Аналіз артефактів персистентності дозволяє виявити вектор збереження присутності зловмисника в системі після перезавантаження.

Висновки до розділу 1

За результатами проведеного теоретико-аналітичного дослідження у першому розділі сформульовано такі висновки.

1. Цифрова криміналістика являє собою науково-прикладну галузь знань, що охоплює методи та засоби ідентифікації, збору, збереження, аналізу та документування цифрових доказів. Її фундаментальними принципами є цілісність, автентичність, відтворюваність, мінімальне втручання та об'єктивність. Забезпечення дотримання зазначених принципів на рівні автоматизованої системи визначається як першочергова вимога до її архітектурного рішення.

2. Правова база цифрових розслідувань в Україні формується Законом України «Про основні засади забезпечення кібербезпеки України», Кримінальним процесуальним кодексом України та відповідними підзаконними нормативними актами. Показовим індикатором зростання кіберзагроз є статистика діяльності CERT-UA: у 2024 році було опрацьовано 4 315 кіберінцидентів, що на 70 % перевищує показник 2023 року. Наведені дані підтверджують критичну актуальність впровадження засобів автоматизованого аналізу для забезпечення ефективного реагування на зростаючий обсяг загроз.

3. Міжнародні стандарти NIST SP 800-86 та ISO/IEC 27037 не є конкуруючими підходами, а утворюють взаємодоповнювальну методологічну основу. Перший забезпечує технічну повноту процесів цифрового розслідування, тоді як другий визначає процедурну строгість їх реалізації. У сукупності ці стандарти формують комплексну методологічну базу, на якій ґрунтується проєктована система.

4. Класифікація цифрових артефактів є не лише теоретичною категорією, а має безпосередній вплив на архітектурні рішення системи. Виділення шести типів артефактів за носієм зумовлює необхідність реалізації шести відповідних модулів обробки; порядок волатильності визначає логіку пріоритетності збору даних; а ролі артефактів у розслідуванні формують логіку їх інтерпретації та зіставлення з індикаторами компрометації (IoC). Практичне розгортання зазначених архітектурних компонентів є предметом розгляду наступних розділів роботи.

РОЗДІЛ 2. АНАЛІЗ ІСНУЮЧИХ FORENSIC-ІНСТРУМЕНТІВ ТА МЕТОДІВ ДОСЛІДЖЕННЯ ЦИФРОВИХ АРТЕФАКТІВ

Перш ніж проектувати власну систему автоматизованого аналізу, варто уважно подивитись на те, що вже існує. Ринок forensic-інструментів досить зрілий: одні рішення розвиваються понад двадцять років, інші з'явились нещодавно, але вже здобули широке визнання. Завдання цього розділу - не просто скласти каталог інструментів, а зрозуміти, де кожен із них справляється добре, де має прогалини і, головне, наскільки він придатний для роботи в складі автоматизованого конвеєра без постійного втручання людини.

Огляд охоплює три основні категорії: інструменти для роботи з дисковими артефактами, засоби аналізу мережевого трафіку та оперативної пам'яті, а також спеціалізовані рішення для виявлення шкідливих патернів. Для кожної категорії наведено порівняльний аналіз за критеріями автоматизації, що безпосередньо впливають на архітектурні рішення розроблюваної системи.

2.1 Класифікація та огляд forensic-інструментів для аналізу дискових артефактів

Дисковий аналіз - найстаріший напрям цифрової криміналістики. Перші спеціалізовані інструменти для роботи з файловими системами з'явились іще наприкінці 1990-х років, і з тих пір галузь пройшла великий шлях: від простих hex-редакторів до повноцінних криміналістичних платформ із графічним інтерфейсом, автоматизованою побудовою таймлайну та підтримкою сотень форматів файлів [25]. Сьогодні аналітик має у своєму розпорядженні як безкоштовні відкриті рішення, так і дорогі комерційні продукти - і вибір між ними далеко не завжди очевидний.

2.1.1 Autopsy/Sleuth Kit

Autopsy разом із The Sleuth Kit (TSK) - мабуть, найвідоміша відкрита криміналістична платформа. Проект бере свій початок ще з 2000-х і нині підтримується компанією Basis Technology. Якщо TSK - це набір консольних утиліт і бібліотека libtsk, то Autopsy є графічним фронтендом над ним, який суттєво знижує поріг входу для аналітиків, що не хочуть або не можуть працювати виключно в терміналі [26].

Ядро системи, бібліотека libtsk, підтримує образи у форматах RAW, E01, AFF та AFF4. Вона вміє читати файлові системи NTFS, FAT16/32, exFAT, ext2/3/4, HFS+, APFS, ISO9660 та Yaffs2 (для Android-пристроїв) - що вкриває переважну більшість носіїв, з якими доводиться мати справу на практиці [26].

Архітектура Autopsy побудована на принципі модулів (Ingest Modules), що запускаються паралельно після додавання джерела даних. Кожен модуль виконує свою конкретну задачу: один відновлює видалені файли, інший будує хеш-бібліотеку, третій шукає ключові слова, четвертий витягує EXIF із фотографій. По суті, це вже певний прообраз підходу, який закладено в основу системи, що розробляється в цій роботі, - тільки з більшою гнучкістю та можливістю інтеграції зовнішніх інструментів [27].

До ключових можливостей Autopsy належать: автоматичне відновлення видалених файлів, повнотекстовий пошук на базі Lucene, побудова таймлайну активності, аналіз веббраузерів (Chrome, Firefox, IE/Edge), вилучення EXIF-метаданих із зображень, пошук за хешами відомих шкідливих файлів через HashDB і фільтрація системних файлів через базу NSRL [28]. Додатково підтримується робота з поштовими контейнерами (PST, MBOX) та мобільними резервними копіями.

Однак є і суттєві обмеження. По-перше, Autopsy погано масштабується на великих образах: аналіз образу розміром 500 ГБ може тривати кілька годин навіть на потужному сервері, оскільки обробка

здебільшого однопотокова на рівні файлової системи. По-друге, Python API для написання власних модулів задокументований значно гірше за Java API і функціонально поступається йому. По-третє, стандартні модулі практично не підтримують хмарні артефакти і повільно адаптуються до нових версій операційних систем - цю нішу займають комерційні продукти на кшталт Magnet AXIOM чи Oxygen Forensic Detective [29].

Для розроблюваної системи Autopsy може використовуватись як джерело попередньо оброблених результатів через Python Ingest API, однак для пакетного автоматизованого аналізу більш доцільним є безпосереднє використання Sleuth Kit на рівні бібліотеки libtsk або консольних утиліт.

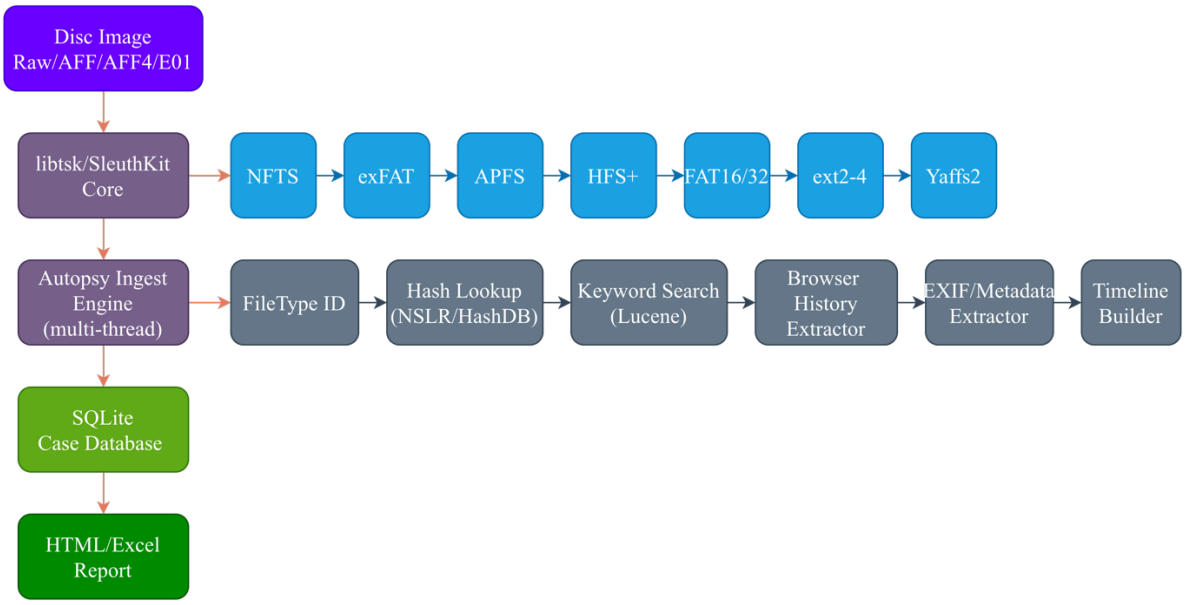

Рис. 2.1. Архітектура Autopsy/The Sleuth Kit

2.1.2 FTK Imager та Forensic Toolkit (FTK)

FTK Imager - безкоштовний інструмент від AccessData (нині Exterro) для створення криміналістичних образів та їх попереднього перегляду. Незважаючи на те, що повна версія FTK є комерційним продуктом, FTK Imager активно використовується навіть у командах, орієнтованих на

відкриті рішення, - насамперед через зручний інтерфейс і широку підтримку форматів образів: RAW/dd, E01, AFF, AD1, SMART [30].

Один із найкорисніших режимів роботи FTK Imager - монтування образу в режимі read-only та надання до нього доступу як до звичайного диска. Це особливо зручно, коли потрібно запустити інший інструмент, що вміє працювати лише з файловою системою, а не з сирим образом. FTK Imager також підтримує захоплення змісту пам’яті у форматі mem-образу і збереження окремих файлів або директорій у форматі AD1 [30].

Повна версія FTK відрізняється тим, що перед початком аналізу індексує весь образ і зберігає індекс у власній базі даних. Завдяки цьому повторні пошукові запити виконуються дуже швидко, проте первинна індексація вимагає значних обчислювальних ресурсів. FTK також має вбудований парсер реєстру Windows і підтримку атак на зашифровані файли. Головний недолік з точки зору автоматизації - висока вартість комерційної ліцензії та закрита архітектура без відкритого API, що унеможливлює програмну інтеграцію [31].

2.1.3 X-Ways Forensics

X-Ways Forensics - один із найпотужніших комерційних інструментів для дискової криміналістики, добре відомий серед досвідчених аналітиків завдяки своїй продуктивності та гнучкості. Цей інструмент навмисно позбавлений яскравого GUI - він орієнтований на фахівців, готових вивчити нетривіальний інтерфейс в обмін на значний виграш у швидкості. Написаний на C++, він не тягне за собою Java-рантайм і тому працює значно швидше за Autopsy чи FTK при роботі з великими образами [32].

X-Ways підтримує функцію одночасного пошуку (simultaneous search), що дозволяє шукати тисячі ключових слів за один прохід по образу, - це надзвичайно цінно при розслідуваннях, де список індикаторів компрометації нараховує сотні записів. Вбудований парсер реєстру і засоби роботи з SQLite-базами роблять X-Ways незамінним при глибокому аналізі

артефактів Windows. Суттєвий мінус - відсутність відкритого API для зовнішньої автоматизації та необхідність щорічного продовження ліцензії [32].

2.1.4 Plaso / log2timeline

Plaso (Python Log2Timeline) - відкритий інструмент для побудови так званих суперхронологій (super-timelines) із великої кількості різнотипних джерел. Він збирає часові мітки з десятків видів артефактів і зводить їх в єдиний хронологічний потік, що дозволяє відновити послідовність подій на скомпрометованій системі з точністю до мілісекунди [33].

Plaso складається з двох основних компонентів: log2timeline.py, який збирає і обробляє артефакти з образу або файлової системи, і psort.py, який фільтрує, сортує та виводить результати. Інструмент підтримує понад 50 парсерів різних форматів: журнали подій Windows (EVTX), реєстр, браузерну історію (Chrome, Firefox, Safari, IE), журнали macOS, Prefetch-файли, Recyclebin, LNK-файли, SQLite-бази застосунків тощо. Результати зберігаються у власному форматі .plaso (SQLite) або можуть бути експортовані у JSON, l2t_csv, tln та інші формати для подальшого аналізу в Timeline Explorer або Timesketch [33, 34].

Для автоматизованого конвеєра Plaso є дуже цінним компонентом: він має добре задокументований Python API, підтримує розподілену обробку через multiprocessing і може запускатись у безголовому режимі. Головні недоліки - висока вимогливість до оперативної пам’яті при роботі з великими образами (рекомендується наявність RAM вдвічі більшої за розмір образу) і тривалий час первинного запуску при повному аналізі файлової системи [34].

2.1.5 KAPE та інструменти Еріка Ціммерманна

Набір безкоштовних утиліт Еріка Ціммерманна (Eric Zimmermann’s Tools) сьогодні є стандартом де-факто для швидкого збору та парсингу артефактів Windows. Ці консольні інструменти охоплюють практично весь

спектр Windows-артефактів: Registry Explorer - для аналізу реєстру, MFTECmd - для парсингу MFT, LECmd - для LNK-файлів, JLECmd - для Jump Lists, PECmd - для Prefetch, AmcacheParser - для баз AmCache і ShimCache [35].

KAPE (Kroll Artifact Parser and Extractor) - флагманський продукт цього набору, що поєднує збір і обробку артефактів в одному рішенні. KAPE працює на основі файлів конфігурацій (.tkape для цілей збору, .mkape для модулів обробки), що дозволяє гнучко налаштовувати сценарії збору без написання коду. Завдяки підтримці паралельного виконання модулів і орієнтації на CLI, KAPE є одним із найзручніших інструментів для включення в автоматизований конвеєр при роботі з Windows-системами. Типовий повний збір артефактів займає від кількох хвилин до пів години залежно від обсягу системи, що значно швидше за повне дискове знімання [35].

Magnet AXIOM - комерційна платформа, орієнтована насамперед на DFIR-команди корпоративного сектору. Вона об’єднує дисковий аналіз, роботу з хмарними сервісами і мобільними пристроями в єдиному інтерфейсі та пропонує функцію AI Insights для автоматичного виявлення аномалій. Однак закрита архітектура і висока вартість ліцензії роблять її непрактичною для вбудовування в власну систему - хіба що як допоміжне джерело, результати якого імпортуються в уніфікований формат [36].

Табл. 2.1 - Порівняння інструментів аналізу дискових артефактів

Інструмент	Ліцензія	Підтримувані ФС	API / Автоматизація	Продуктивність	Основне застосування
Autopsy 4.x	Apache 2.0	NTFS, FAT, ext2-4, HFS+, APFS, Yaffs2	Java/Python Ingest API	Середня	Навчання, DFIR

Продовження Табл. 2.1

Інструмент	Ліцензія	Підтримувані ФС	API / Автоматизація	Продуктивність	Основне застосування
Sleuth Kit (TSK)	CPL/IPL	NTFS, FAT, ext2-4, HFS+, ISO9660	C/Python бібліотека	Висока	Скриптова автоматизація
FTK Imager	Безкоштовна (proprietary)	NTFS, FAT, ext, HFS+	Відсутнє	Висока	Зняття образів
X-Ways Forensics	Комерційна	Більшість відомих ФС	Обмежений скрипт	Дуже висока	Глибокий аналіз
Plaso / log2timeline	Apache 2.0	Не аналізує ФС напряму	Python API, CLI	Середня	Таймлайн артефактів
KAPE	Безкоштовна	Windows-орієнтований	CLI, .mkape/.tkape	Дуже висока	Швидкий збір артефактів
Magnet AXIOM	Комерційна	Широкий спектр	REST API (обмежений)	Висока	Корпоративний DFIR

Аналіз таблиці 2.1 показує, що для автоматизованого конвеєра найкраще підходять інструменти з відкритою ліцензією та наявністю API або CLI: Sleuth Kit, Plaso і KAPE. Autopsy може використовуватись через Python Ingest API, але його продуктивність у режимі пакетного аналізу є суттєвим обмеженням. Комерційні рішення (X-Ways, FTK Imager, AXIOM) доцільно розглядати лише як додаткові джерела, результати яких можна імпортувати в уніфікований формат.

2.2 Інструменти аналізу мережевого трафіку та оперативної пам’яті

Якщо дисковий аналіз дає нам повну картину того, що зберігалось на носії, то мережеві артефакти і дампи оперативної пам’яті розповідають зовсім іншу историю - про те, що відбувалось у реальному часі: які з’єднання встановлювались, які процеси виконувались, які ключі

шифрування завантажувались. Ці джерела є особливо цінними при розслідуванні складних цільових атак і безфайлового шкідливого ПЗ, сліди якого на диску мінімальні або відсутні.

2.2.1 Wireshark та tshark

Wireshark - стандарт де-факто для захоплення та аналізу мережевого трафіку. Однак для forensic-цілей у рамках автоматизованого конвеєра значно більший інтерес становить його командний аналог tshark, який дозволяє обробляти PCAP-файли в пакетному режимі без відкриття графічного інтерфейсу [37].

З точки зору криміналістичного аналізу, tshark вміє фільтрувати трафік за протоколом, IP-адресами, портами та вмістом пакетів; витягувати передані файли (через HTTP, SMB, FTP); відновлювати TCP-сесії; обчислювати статистику з'єднань; виводити результати у CSV, JSON або PDML (XML) для подальшої машинної обробки. Dissector-архітектура Wireshark підтримує понад 3 000 протоколів, що робить його незамінним при роботі з нестандартним трафіком [37, 38].

Для інтеграції в конвеєр tshark можна викликати як зовнішній процес через subprocess у Python або використовувати бібліотеку pyshark, що надає зручний Pythonic інтерфейс над tshark. Ключове обмеження - продуктивність при обробці дуже великих PCAP-файлів (від 10 ГБ і більше): tshark не підтримує паралельну обробку, тому для таких обсягів рекомендується попередньо розбивати файл на частини утилітою editcap [38].

2.2.2 Zeek (Bro)

Zeek (до 2018 року відомий як Bro) займає зовсім іншу нішу порівняно з Wireshark. Якщо Wireshark дає можливість переглянути кожен пакет, то Zeek аналізує трафік на рівні з'єднань і подій, автоматично генеруючи структуровані журнали у форматі TSV або JSON. Саме ця здатність продукувати готові до аналізу дані без додаткового парсингу

робить Zeek одним із найцінніших компонентів будь-якого автоматизованого конвеєра [39].

За замовчуванням Zeek генерує такі типи журналів: conn.log з усіма TCP/UDP/ICMP-з'єднаннями, http.log з HTTP-запитами та заголовками, dns.log з DNS-запитами і відповідями, ssl.log з TLS-хендшейками включно з JA3/JA3S-хешами для фінгерпринтингу TLS-клієнтів, files.log з виявленими файлами в трафіку та їх хешами, x509.log з SSL-сертифікатами та weird.log з аномальним трафіком. Всі ці журнали можна завантажити в pandas або Elasticsearch без жодної попередньої обробки [39, 40].

Окрім пасивного моніторингу, Zeek має власну мову скриптів, що дозволяє писати власні детектори аномалій і підключати їх до живого потоку або до записаного PCAP. Наприклад, можна реалізувати скрипт виявлення DNS-тунелювання за патерном запитів або детектування beaconing за регулярністю з'єднань з потенційним C2-сервером. Для системи, що розробляється, Zeek є особливо цінним саме через структуровані JSON-журнали, що легко інтегруються в уніфікований формат аналізу [40].

2.2.3 Suricata та YARA

Suricata - IDS/IPS-рушій з відкритим кодом, що для цілей криміналістики є цікавим насамперед як пасивний аналізатор PCAP-трафіку. В режимі pcap-reading Suricata застосовує сигнатурні набори до записаного трафіку і генерує структуровані JSON-алерти (формат EVE) із детальною інформацією про виявлену активність. Сигнатурні набори Emerging Threats, що нараховують понад 30 000 правил, покривають переважну більшість відомих C2-протоколів, сканерів і шкідливого ПЗ [41].

Suricata підтримує паралельну обробку завдяки багатопотоковій архітектурі, що суттєво відрізняє його від tshark. Це дозволяє ефективно аналізувати великі PCAP-файли без попереднього розбиття. Додатково Suricata підтримує виявлення файлів у трафіку та їх хешування, що корисно

при пошуку переданого шкідливого ПЗ. Для конвеєра Suricata запускається як зовнішній процес через CLI, а JSON-виведення легко парсується стандартними засобами Python [41].

YARA - окремий, але дуже важливий інструмент для опису і виявлення шаблонів у файлах та дампах пам’яті. Правила YARA описують ознаки конкретного шкідливого ПЗ або підозрілої активності через рядкові патерни, регулярні вирази та логічні умови. YARA широко використовується разом з іншими інструментами: Volatility запускає YARA проти дампу пам’яті, а Sleuth Kit може застосовувати YARA до файлів при скануванні образу диска. Бібліотека yara-python надає прямий доступ до движка YARA з Python без виклику зовнішнього процесу, що мінімізує накладні витрати при масовому скануванні [42].

2.2.4 Volatility Framework

Volatility - стандарт де-факто для аналізу дампів оперативної пам’яті. Третя версія фреймворку (Volatility 3) переписана з нуля на Python 3 з модульною архітектурою плагінів і автоматичним визначенням символів ОС через базу Symbol Tables. Це вирішило одну з головних проблем другої версії - необхідність вручну вказувати профіль операційної системи, що раніше вимагало знань внутрішньої структури кожної конкретної версії ядра [19, 43].

Volatility 3 підтримує аналіз дампів Windows (7/8/10/11, Server 2008-2022), Linux і macOS. Серед ключових плагінів: windows.pslist і windows.pstree для аналізу запущених процесів, windows.cmdline для перегляду аргументів командного рядка, windows.dlllist для переліку завантажених бібліотек, windows.netscan для мережевих з’єднань, windows.malfind для виявлення ін’єкцій у пам’ять, windows.hashdump для витягу хешів паролів та windows.filescan для відкритих файлових дескрипторів [43].

Для автоматизованого конвеєра Volatility 3 є оптимальним вибором: він має Python API, підтримує CLI і повертає результати у форматі JSON. Власний плагін можна написати і підключити без модифікації ядра фреймворку - достатньо успадкувати відповідний базовий клас і реалізувати метод run(). Суттєвий недолік - висока вимогливість до оперативної пам’яті хоста при аналізі великих дампів: рекомендується наявність RAM, що вдвічі перевищує розмір дампу [44].

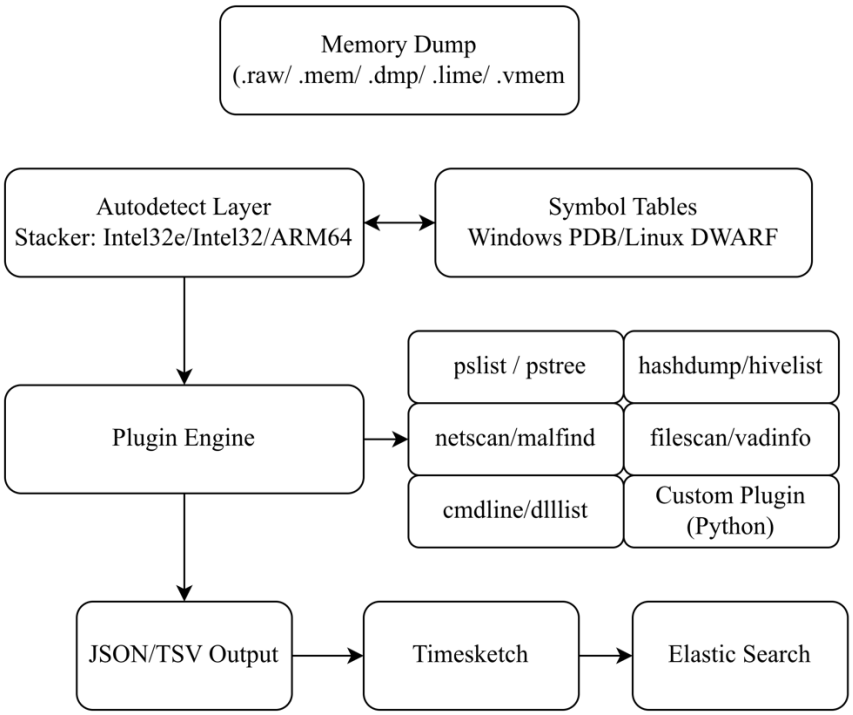

Рис. 2.2. Архітектура Volatility 3 Framework

2.2.5 Rekall та WinPmem / LiME

Rekall - форк Volatility 2, розроблений командою Google, що мав амбіційну мету перевершити оригінал за швидкістю та точністю. Rekall першим реалізував низку технік, що пізніше з’явились у Volatility 3: автоматичне визначення профілів через GUID, підтримку аналізу “живої” системи без попереднього знімання дампу та API на базі Python-

генераторів. Проте у 2019 році Google припинила активну розробку, і нині Rekall фактично є застарілим проектом. Його згадують як важливий крок в еволюції memory forensics, але для нових проектів рекомендовано Volatility 3 [44].

LiME (Linux Memory Extractor) і WinPmem - це не аналізатори, а засоби зняття дампів оперативної пам’яті для Linux і Windows відповідно. LiME реалізований як завантажуваний модуль ядра Linux і може передавати вміст пам’яті через мережу або записувати до файлу у форматах raw, padded або lime - останній підтримується Volatility. WinPmem - безкоштовний інструмент з відкритим кодом для Windows, що підтримує виведення в різних форматах і не потребує перезавантаження. Обидва інструменти є важливими компонентами етапу збору в автоматизованому конвеєрі, але самостійно артефакти не аналізують [44].

2.2.6 NetworkMiner

NetworkMiner - мережевий forensic-аналізатор, орієнтований на реасемблювання переданих файлів із PCAP-трафіку і визначення хостів мережі. На відміну від Wireshark, він зосереджений не на інспекції пакетів, а на відновленні контенту: зображень, документів, електронних листів, облікових даних, переданих у відкритому вигляді. Безкоштовна версія добре справляється з аналізом HTTP-трафіку і DNS і має CLI-режим через NetworkMiner.exe, придатний для базової автоматизації. Комерційна версія додає підтримку більшої кількості протоколів та покращений вивід у CSV [45].

Табл. 2.2 - Порівняння інструментів аналізу мережевого трафіку

Інструмент	Тип	Ліцензія	Вхідні формати	Вихідні формати	API / Автоматизація
Wireshark / tshark	Аналіз пакетів	GPLv2	PCAP, PCAPng	JSON, CSV, PDML	CLI, pyshark

Продовження Табл. 2.2

Інструмент	Тип	Ліцензія	Вхідні формати	Вихідні формати	API / Автоматизація
Zeek (Bro)	Аналіз подій	BSD	PCAP, live	TSV, JSON logs	Zeek scripts, CLI
Suricata	IDS / Аналіз	GPLv2	PCAP, live	JSON alerts (EVE)	Python, CLI
NetworkMiner	Відновл. файлів	GPL/ комерційна	PCAP	Файли, CSV	Обмежений CLI
YARA	Виявл. патернів	BSD	Будь-які файли	Match-результати	yara-python

Табл 2.3 - Порівняння інструментів аналізу оперативної пам’яті

Інструмент	Ліцензія	Підтримувані ОС	API / Автоматизація	JSON-вивід	Продуктивність
Volatility 3	Volatility Lic. (відкрита)	Win 7-11, Linux, macOS	Python API, CLI	Так	Середня
Rekall	Apache 2.0 (застарілий)	Win, Linux, macOS	Python API	Так	Висока
LiME	GPLv2	Linux	Kernel module (збір)	Ні	Дуже висока
WinPmem	Apache 2.0	Windows	CLI (збір)	Ні	Дуже висока

2.3 Порівняльний аналіз forensic-інструментів за критеріями автоматизації

Попередні підрозділи були присвячені опису можливостей окремих інструментів. Однак для проектування системи автоматизованого аналізу важливий не просто перелік функцій, а чітка відповідь на практичне запитання: наскільки той чи інший інструмент придатний до включення в автоматизований конвеєр без постійної участі людини? Для цього запропоновано систему оцінювання за п’ятьма ключовими критеріями [46].

2.3.1 Критерії оцінювання придатності до автоматизації

Наявність CLI або API є першим і найважливішим критерієм. Він оцінює можливість запуску інструменту без графічного інтерфейсу та взаємодії з ним програмно - через CLI, Python API, REST API або бібліотеку. Інструменти, що функціонують виключно з GUI і не мають жодного способу програмного керування, отримують нуль балів, оскільки їх неможливо включити в автоматизований конвеєр без використання засобів емуляції UI, що є крихкими і ненадійними в продуктивних середовищах.

Структурований вивід передбачає можливість отримання результатів у машиночитаному форматі: JSON, CSV, XML або TSV. Неструктурований текстовий вивід, що потребує регулярних виразів для парсингу, є суттєвим недоліком, оскільки будь-яка зміна форматування у новій версії інструменту може непомітно зламати конвеєр. JSON є найбажанішим форматом - він підтримує вкладені структури і не потребує визначення розподільника полів.

Детермінованість результатів вимагає, щоб повторний запуск з тими самими вхідними даними давав ідентичні результати. Це критично для відтворюваності розслідування і автоматизованого тестування системи. Інструменти з недетермінованою поведінкою (наприклад, залежні від стану випадкового генератора або системного часу у виводі) ускладнюють верифікацію результатів.

Масштабованість відображає здатність інструменту ефективно обробляти великі обсяги даних - образи дисків від 100 ГБ, PCAP-файли тривалого захоплення, великі дампи пам’яті. Оцінюються наявність multiprocessing, підтримка розподіленої обробки та наявність оптимізацій для роботи з великими файлами. Відсутність горизонтального масштабування стає критичним обмеженням у корпоративних розслідуваннях.

Підтримка та документація враховують активність розробки проекту за останні 12 місяців, наявність документації для розробників, стабільність API між версіями і стан відкритих помилок. Застарілі або покинуті проекти отримують знижені оцінки, оскільки несумісність із новими версіями ОС або вразливості безпеки можуть стати непередбаченими проблемами в продуктивному середовищі.

Кожен критерій оцінюється за п’ятибальною шкалою. Підсумковий бал обчислюється як зважена сума: наявність CLI/API та структурований вивід мають вагу 0,25 кожен, детермінованість і масштабованість - по 0,20, підтримка - 0,10.

2.3.2 Зведена порівняльна таблиця

Табл. 2.4 - Зведений порівняльний аналіз forensic-інструментів за критеріями автоматизації

Інструмент	CLI/API (0,25)	Структ. вивід (0,25)	Детерм. (0,20)	Масшт. (0,20)	Підтримка (0,10)	Зважений бал	Рекомен д.
Sleuth Kit (TSK)	5	4	5	4	4	4,45	Так
Autopsy 4.x	3	3	4	3	4	3,35	Частков о
Plaso	5	5	5	3	4	4,35	Так
KAPE	5	5	5	4	4	4,65	Так
tshark	5	5	5	4	5	4,75	Так
Zeek	5	5	5	5	4	4,90	Так
Suricata	5	5	5	5	5	5,00	Так
YARA	5	5	5	5	5	5,00	Так
Volatility 3	5	5	5	3	4	4,35	Так
FTK Imager	2	2	4	4	3	2,90	Ні
X-Ways	2	2	5	5	3	3,10	Ні
Magnet AXIOM	2	3	4	4	4	3,10	Ні
Network Miner	2	3	4	3	3	2,90	Ні
Rekall	4	5	5	3	1	3,75	Ні

Результати порівняльного аналізу наочно демонструють: найвищі бали отримали YARA і Suricata (5,00), Zeek (4,90), tshark (4,75) і KAPE (4,65). Це інструменти, що від самого початку проектувались для роботи в конвеєрах і пакетній обробці - вони генерують структуровані дані, детерміновані в результатах і добре масштабуються. Натомість комерційні рішення (X-Ways, FTK, AXIOM), попри потужний функціонал, отримали низькі бали через відсутність відкритого API. Rekall, незважаючи на технічно якісну архітектуру, отримав знижку за фактичне завершення підтримки у 2019 році.

2.3.3 Аналіз прогалин у наявних рішеннях

Попри значну кількість якісних інструментів, при спробі побудувати повноцінний автоматизований конвеєр аналізу виникають системні проблеми, що не вирішуються жодним із розглянутих рішень окремо. Ці прогалини і формують проектний простір для системи, що розробляється.

Перша і найбільш очевидна проблема - відсутність уніфікованого формату обміну даними між інструментами. Кожен продукт “говорить” власною мовою: Plaso генерує .plaso (SQLite з власною схемою), Zeek - набір TSV-файлів з окремою логікою для кожного типу журналу, Volatility - JSON або текст залежно від плагіна, YARA - власний формат результатів зіставлення. Аналітик, що хоче об’єднати ці результати в єдину картину, змушений писати власні конвертери або робити це вручну. Жоден інструмент не пропонує стандартного виходу у форматах STIX/TAXII чи OpenIOC, які є прийнятими стандартами обміну даними про загрози.

Друга прогалина - відсутність рівня оркестрації виконання. Жоден із розглянутих інструментів не вирішує задачу “що запускати першим, що паралельно, а що лише після отримання результатів попереднього аналізу”. Цей рівень - рівень воркфлоу - повністю відсутній у відкритих рішеннях. Комерційні платформи (Magnet AXIOM, Belkasoft X) вирішують цю задачу,

але виключно всередині власної закритої екосистеми, що унеможливлює додавання нових компонентів.

Третя прогалина пов’язана з відсутністю автоматичної пріоритезації за волатильністю. Жоден інструмент не приймає рішення про порядок аналізу на основі часових характеристик артефактів згідно з RFC 3227. Аналітик змушений пам’ятати про це самостійно і вручну керувати послідовністю запуску, що у стресових умовах реагування на інцидент є ненадійним підходом.

Четверта і, мабуть, найбільш трудомістка прогалина - відсутність автоматичної кореляції між артефактами з різних джерел. Виявлений Volatility підозрілий процес не автоматично зіставляється з відповідними мережевими з’єднаннями від Zeek, YARA-матчами по завантажених бібліотеках і записами в реєстрі від Plaso. Ця кореляція є найбільш цінною частиною аналізу - саме вона перетворює набір розрізнених артефактів на зв’язну картину атаки - і при цьому жодним відкритим інструментом не автоматизована.

П’ята прогалина стосується масштабування при роботі з великими обсягами. Жоден відкритий інструмент не пропонує вбудованого механізму горизонтального масштабування - розподілу навантаження між кількома обчислювальними вузлами - при аналізі образів від 500 ГБ або тривалих PCAP-записів. Це обмежує застосовність у корпоративних розслідуваннях, де такі розміри є стандартом.

Саме ці п’ять прогалин і визначають функціональні вимоги до системи, що проектується в розділі 3. Система не прагне замінити Volatility, Zeek чи Plaso - вона надає рівень оркестрації над ними, уніфікований формат виводу, механізм кореляції між результатами різних аналізаторів і автоматичну пріоритезацію на основі волатильності. Це означає, що при появі нового інструменту аналізу систему не треба буде переписувати - достатньо додати новий адаптер.

2.3.4 Візуалізація результату порівняльного аналізу

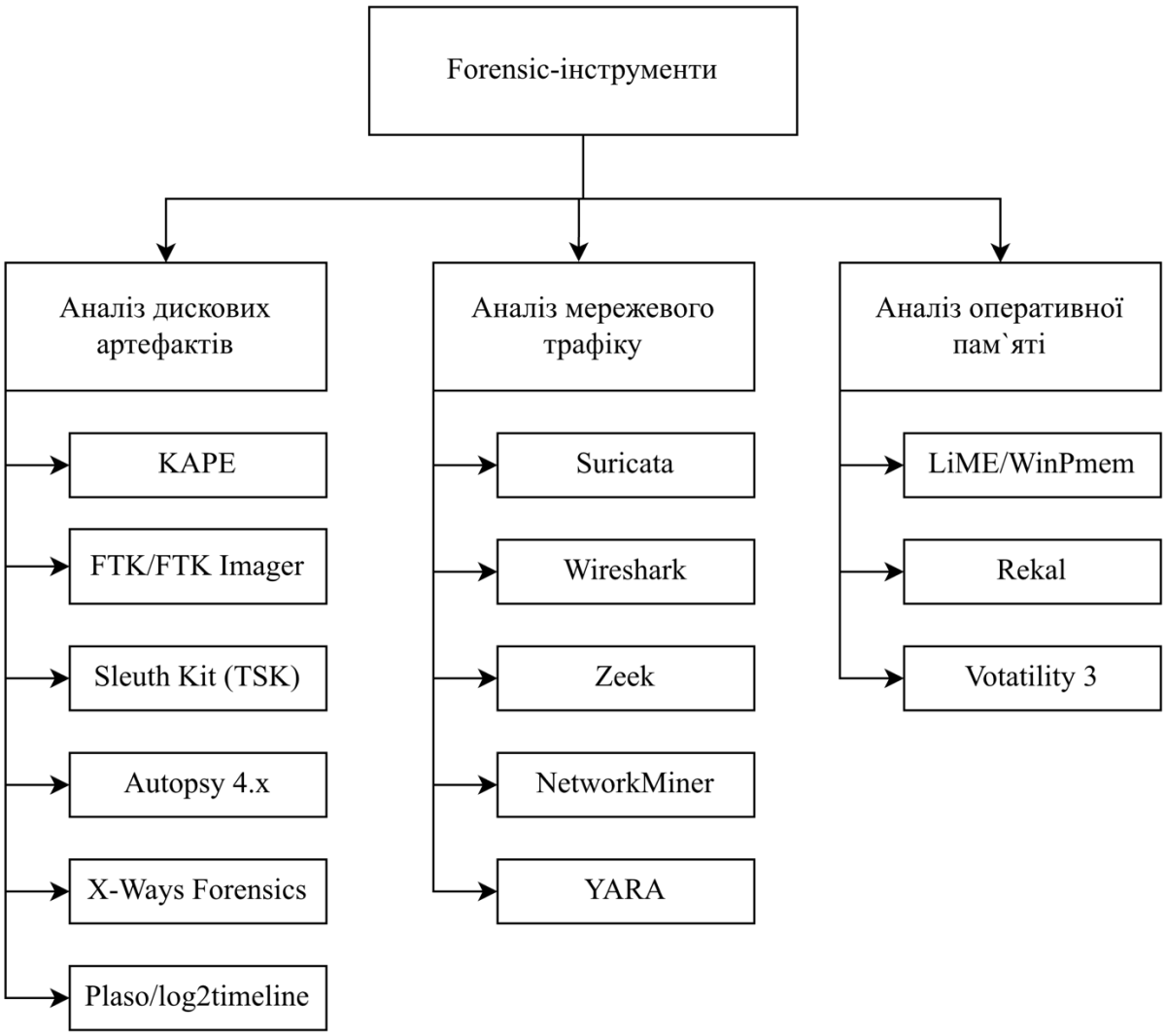

Рис. 2.4. Класифікація forensic-інструментів за категоріями аналізу

Висновки до розділу 2

За результатами другого розділу встановлено таке.

1. Forensic-інструменти для аналізу дискових артефактів охоплюють кілька класів з різними цільовими аудиторіями та сценаріями застосування: платформи повного циклу (Autopsy, FTK, X-Ways), засоби побудови таймлайну (Plaso), засоби швидкого цільового збору артефактів (KAPE і утиліти Ціммерманна) та низькорівневі

бібліотеки (Sleuth Kit). Для автоматизованого конвеєра найбільш придатними є Sleuth Kit, KAPE і Plaso завдяки розвиненому CLI, структурованому виводу та відкритій ліцензії без обмежень на автоматизоване використання.

2. Серед інструментів мережевого аналізу та аналізу пам’яті лідерами за придатністю до автоматизації є YARA (5,00), Suricata (5,00), Zeek (4,90) і tshark (4,75). Zeek і Suricata унікальні тим, що генерують структуровані JSON-журнали без додаткового парсингу, що суттєво спрощує інтеграцію. Volatility 3 є єдиним рекомендованим рішенням для аналізу дампів оперативної пам’яті завдяки активно підтримуваному Python API та підтримці сучасних версій Windows і Linux.

3. Проведений порівняльний аналіз за п’ятьма критеріями автоматизації (CLI/API, структурований вивід, детермінованість, масштабованість, підтримка) виявив, що комерційні рішення (X-Ways, FTK Imager, Magnet AXIOM) попри потужний функціонал отримали низькі оцінки через відсутність відкритого API, тоді як відкриті CLI-орієнтовані інструменти значно краще підходять для побудови автоматизованих систем.

4. Виявлено п’ять ключових прогалин у наявних рішеннях: відсутність уніфікованого формату обміну даними між інструментами, відсутність механізму оркестрації воркфлоу, відсутність автоматичної пріоритезації за волатильністю, відсутність кореляції між результатами різних аналізаторів і обмежена підтримка горизонтального масштабування. Ці прогалини безпосередньо визначають функціональні вимоги до системи, що проектується в розділі 3, і формують її ключові конкурентні переваги відносно існуючих рішень.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО АНАЛІЗУ ЦИФРОВИХ АРТЕФАКТІВ

На основі теоретичних засад, викладених у розділі 1, та результатів порівняльного аналізу forensic-інструментів, проведеного у розділі 2, у цьому розділі представлено повний цикл проектування та реалізації системи автоматизованого аналізу цифрових артефактів - DFAS (Digital Forensic Analysis System). Структура розділу охоплює архітектурне проектування, детальний опис кожного модуля, алгоритм аналізу, механізми валідації вхідних даних та обмежень, web-інтерфейс із відображенням структури системи і результати тестування.

3.1 Архітектура системи DFAS

3.1.1 Функціональні вимоги та обмеження

За результатами аналізу прогалин, виявлених у розділі 2, до системи DFAS сформульовано такі функціональні вимоги.

– ФВ-1. Система повинна запускати модулі у порядку волатильності RFC 3227 (Memory - Network - Filesystem) без участі оператора.

– ФВ-2. Перед початком аналізу система зобов'язана перевіряти розмір файлу та його розширення. Файли розміром понад 600 МБ або з непідтримуваним розширенням відхиляються з інформативним повідомленням.

– ФВ-3. Кожен модуль повертає результат у єдиному JSON-форматі з полями module, findings, severity, elapsed_sec.

– ФВ-4. Orchestrator автоматично виявляє кореляції між результатами різних модулів (5 типів кореляцій).

– ФВ-5. Кожна дія фіксується у журналі chain of custody з ISO 8601-часовою міткою (вимога ISO/IEC 27037).

– ФВ-6. Web-інтерфейс відображає структуру системи, алгоритм аналізу та прогрес виконання конвеєру в реальному часі.

– ФВ-7. YARA-модуль підтримує 10 правил із прив'язкою до матриці MITRE ATT&CK та вбудований IoC-екстрактор вилучає 8 типів індикаторів.

3.1.2 Обмеження на вхідні дані

Система DFAS реалізує двоетапну валідацію вхідного файлу до початку будь-якого аналізу.

Перший етап - перевірка розміру. Максимально дозволений розмір файлу становить 600 МБ. При спробі завантажити файл більшого розміру система повертає повідомлення: «Файл завеликий: X МБ. Максимально дозволений розмір - 600 МБ. Стисніть або розбийте на частини.» Це обмеження діє на двох рівнях: на рівні клієнтського JavaScript (до завантаження на сервер) та на рівні Flask-сервера (обробник помилки 413).

Другий етап - перевірка розширення. Система підтримує 28 типів файлів, що охоплюють усі класи цифрових артефактів відповідно до класифікації розділу 1.3.2.

Табл. 3.1 -Підтримувані типи файлів системи DFAS

Категорія артефакту	Підтримувані розширення
Мережеві журнали	.log, .pcap, .pcapng, .cap
Текстові артефакти	.txt, .csv, .tsv, .json, .xml
Веб-артефакти / скрипти	.html, .htm, .php, .js, .py, .sh, .bat, .ps1, .vbs
Виконувані файли / бінарні	.exe, .dll, .bin, .dat, .sys
Системні журнали Windows	.evtx, .reg
Документи / архіви	.pdf, .doc, .docx, .xls, .xlsx, .zip, .gz, .tar

3.1.3 Структурна схема системи

Архітектура DFAS реалізована за трирівневим модульним принципом. Перший рівень - рівень представлення (Single Page Application на базі HTML/CSS/JS без зовнішніх залежностей). Другий рівень -рівень

оркестрації (PipelineOrchestrator). Третій рівень - рівень аналізу (шість спеціалізованих модулів).

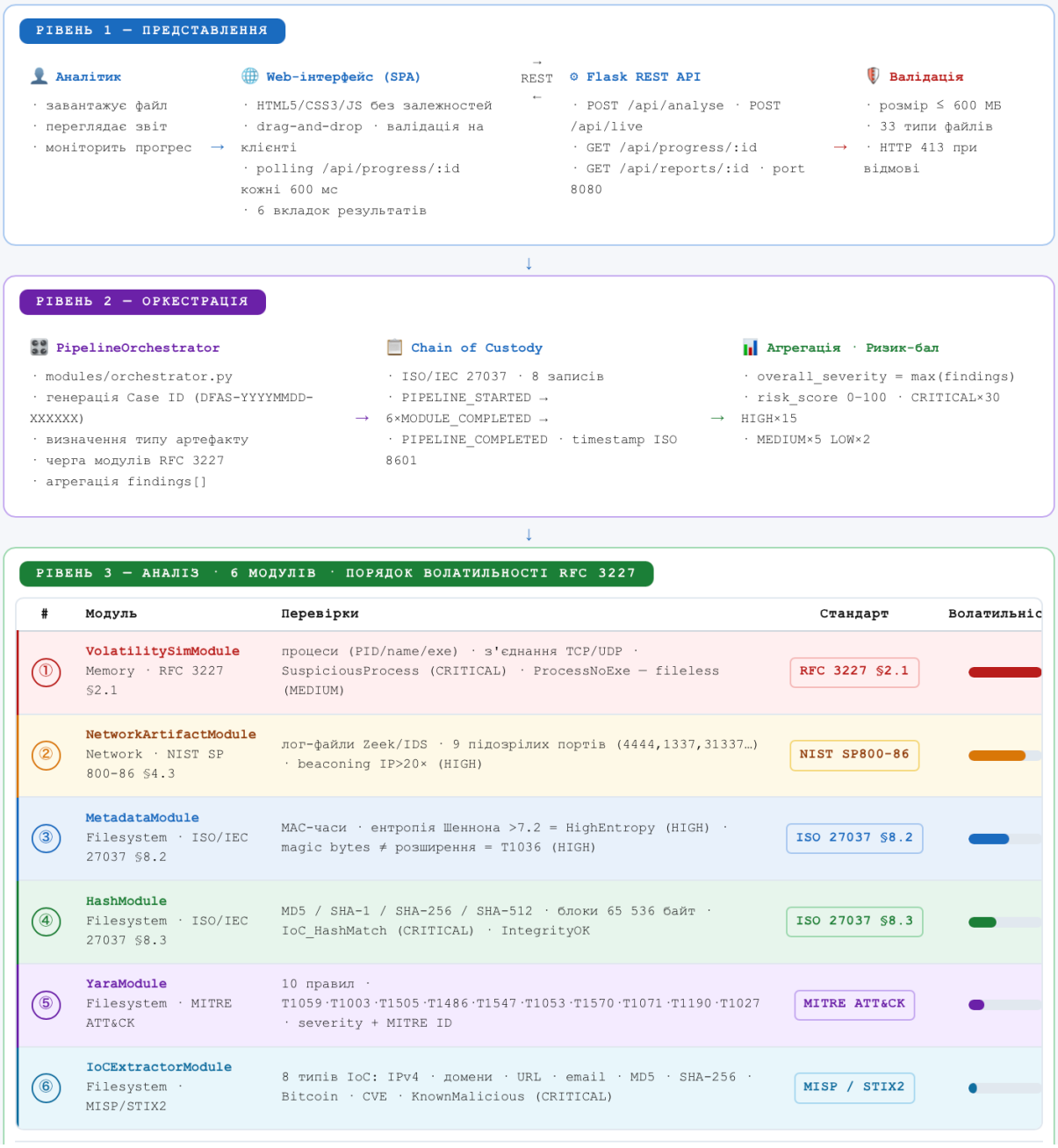

Рис. 3.1. Структурна схема системи

Взаємодія між рівнями за сценарієм аналізу файлу: 1) клієнт перевіряє розмір файлу (JS, до завантаження); 2) POST /api/analyse – Flask приймає файл, викликає validate_file(); 3) якщо файл коректний – створюється job_id та запускається фоновий потік; 4) Orchestrator визначає тип артефакту,

формує чергу модулів; 5) модулі виконуються послідовно з фіксацією в CoC; 6) клієнт опитує /api/progress/:id кожні 600 мс; 7) після завершення результат відображається у web-інтерфейсі.

Табл. 3.2 – Конвеєр модулів DFAS (порядок RFC 3227)

№	Модуль	Тип артефакту	Стандарт / методологія	Час збереження
1	VolatilitySimModule	memory	RFC 3227 §2.1	секунди - хвилини
2	NetworkArtifactModule	network	NIST SP 800-86 §4.3	хвилини
3	MetadataModule	filesystem	ISO/IEC 27037 §8.2	години -дні
4	HashModule	filesystem	ISO/IEC 27037 §8.3	тижні -місяці
5	YaraModule	filesystem	MITRE ATT&CK	тижні -місяці
6	IoCExtractorModule	filesystem	MISP / STIX2	тижні -місяці

3.2 Алгоритм аналізу цифрового артефакту

3.2.1 Загальний алгоритм

Алгоритм аналізу файлового артефакту в системі DFAS складається з семи послідовних етапів. Кожен етап є обов'язковим та виконується у фіксованому порядку незалежно від типу файлу.

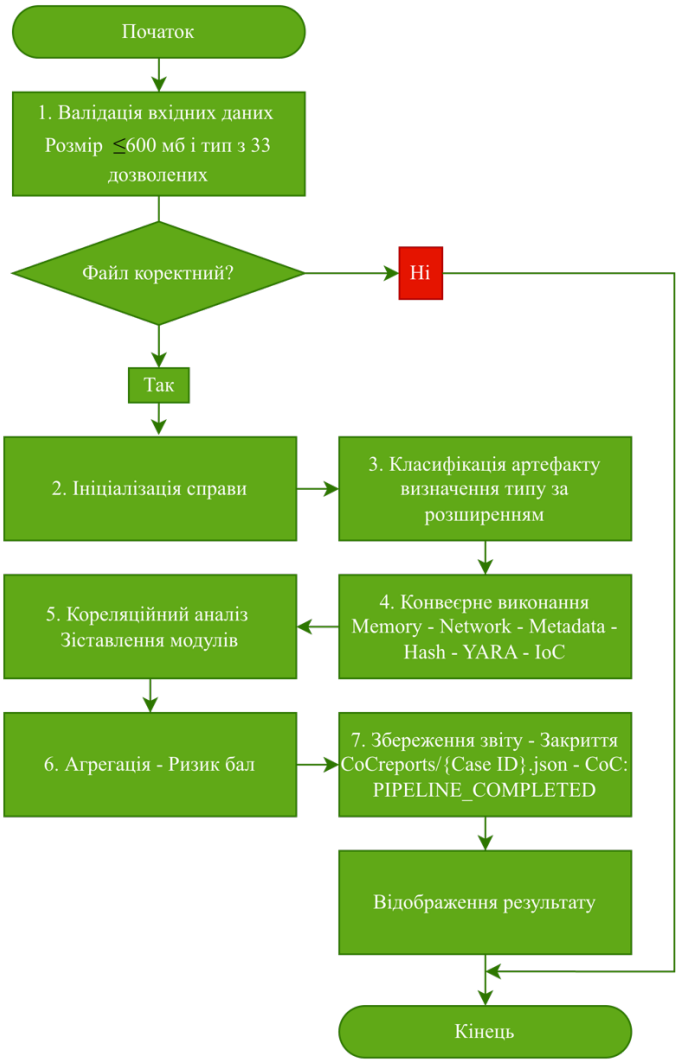

Рис. 3.2. Алгоритм аналізу файлового артефакту

1. Валідація вхідного файлу. Перевіряється розмір (≤ 600 МБ) та розширення (білий список 33 типів). При будь-якому порушенні – негайне повернення повідомлення про помилку, файл видаляється з тимчасового сховища. Подальші кроки не виконуються.

2. Ініціалізація справи. Генерується унікальний ідентифікатор DFAS-YYYYMMDD_HHMMSS-XXXXXX. Записується перший запис у chain of custody: дія PIPELINE_STARTED, часова мітка UTC, ім'я артефакту.

3. Визначення типу артефакту. Orchestrator аналізує розширення файлу та формує перелік типів артефактів: мережеві розширення (.pcap, .log) – додається тип network; для всіх файлів -типи memory та filesystem. Від цього залежить, які модулі будуть активовані.

4. Конвеєрне виконання модулів (RFC 3227). Модулі запускаються послідовно у фіксованому порядку волатильності: VolatilitySimModule (живий аналіз системи, завжди перший) – NetworkArtifactModule - MetadataModule – HashModule – YaraModule – IoCExtractorModule. Для кожного модуля фіксується час виконання та результат у CoC.

5. Кореляційний аналіз. Orchestrator зіставляє результати всіх модулів. Виявляються міжмодульні кореляції: PackedMalware (YaraModule + MetadataModule, ентропія > 7.0), ActiveC2Channel (IoCExtractorModule + NetworkArtifactModule), FilelessAttack (VolatilitySimModule + YaraModule), MasqueradingMalware (MetadataModule + YaraModule), RansomwareConfirmed (IoCExtractorModule + YaraModule, Bitcoin-адреса).

6. Агрегація та обчислення ризик-балу. Обчислюється загальна критичність (maximum серед усіх знахідок за шкалою INFO – LOW – MEDIUM – HIGH – CRITICAL) та числовий ризик-бал від 0 до 100 за формулою: INFO×0 + LOW×2 + MEDIUM×5 + HIGH×15 + CRITICAL×30, з обмеженням max = 100.

7. Збереження звіту та закриття CoC. Повний JSON-звіт зберігається в директорії reports/ під іменем {Case ID}.json. Записується фінальний запис у CoC: PIPELINE_COMPLETED, загальна критичність, ризик-бал.

3.2.2 Алгоритм перевірки кожного модуля

Нижче описано перелік перевірок, що виконує кожен модуль при аналізі файлового артефакту.

Табл. 3.3 – Перелік перевірок кожного модуля системи DFAS

Модуль	Перевірка	Результат (тип знахідки)	Критичність
VolatilitySimModule	Зіставлення імен процесів із переліком відомих шкідливих інструментів (Mimikatz, Meterpreter, nc.exe тощо)	SuspiciousProcess	CRITICAL
VolatilitySimModule	Виявлення процесів без exe-шляху (fileless / ін'єкція в пам'ять)	ProcessNoExe	MEDIUM
VolatilitySimModule	Збір переліку активних мережевих з'єднань (TCP/UDP)	LiveMemory_Scanned	INFO
NetworkArtifactModule	Зіставлення TCP/UDP-портів із переліком підозрілих (4444, 1337, 31337, 9050 тощо)	SuspiciousPort	HIGH
NetworkArtifactModule	Виявлення beaconing: IP-адреса з'являється > 20 разів у журналі	PossibleBeaconing	HIGH
NetworkArtifactModule	Парсинг журналу: підрахунок унікальних IP, топ-10 адрес	NetworkLog_Parsed	INFO
MetadataModule	Обчислення ентропії Шеннона (> 7.2 = packed/encrypted)	HighEntropy	HIGH
MetadataModule	Зіставлення розширення з magic bytes файлу (маскування T1036)	ExtensionMismatch	HIGH
MetadataModule	Виявлення виконуваних файлів (PE / ELF)	ExecutableDetected	MEDIUM

Продовження табл 3.3

Модуль	Перевірка	Результат (тип знахідки)	Критичність
MetadataModule	Збір MAC-часових міток (Modified/Accessed/Created)	IntegrityMAC	INFO
HashModule	Обчислення MD5, SHA-1, SHA-256, SHA-512	IntegrityOK	INFO
HashModule	Зіставлення хешів з базою відомих шкідливих файлів (IoC)	IoC_HashMatch	CRITICAL
YaraModule	Правило: Suspicious_PowerShell_Encoded (T1059.001)	YARA_EXECUTION	HIGH
YaraModule	Правило: Mimikatz_Strings (T1003)	YARA_CREDENTIAL_ACCESS	CRITICAL
YaraModule	Правило: Webshell_PHP (T1505.003)	YARA_WEBSHELL	CRITICAL
YaraModule	Правило: Ransomware_Note (T1486)	YARA_RANSOMWARE	CRITICAL
YaraModule	Правило: Persistence_Registry_Run (T1547.001)	YARA_PERSISTENCE	MEDIUM
YaraModule	Правило: Scheduled_Task_Persistence (T1053.005)	YARA_PERSISTENCE	HIGH
YaraModule	Правило: Lateral_Movement_PsExec (T1570)	YARA_LATERAL_MOVEMENT	HIGH
YaraModule	Правило: C2_Beacon_Pattern (T1071)	YARA_C2	CRITICAL
YaraModule	Правило: SQL_Injection_Attempt (T1190)	YARA_INJECTION	HIGH
YaraModule	Правило: Suspicious_Base64_Blob (T1027)	YARA_OBFUSCATION	MEDIUM

Продовження табл 3.3

Модуль	Перевірка	Результат (тип знахідки)	Критичність
IoCExtractorModule	Вилучення Ipv4-адрес та зіставлення з базою відомих C2	KnownMaliciousIP	CRITICAL
IoCExtractorModule	Вилучення доменів та зіставлення з базою шкідливих доменів	KnownMaliciousDomain	CRITICAL
IoCExtractorModule	Виявлення Bitcoin-адрес (ознака ransomware)	BitcoinAddress	HIGH
IoCExtractorModule	Виявлення посилань на CVE (вразливості)	CVE_Reference	MEDIUM
IoCExtractorModule	Вилучення URL, email, MD5, SHA-256	IoCExtracted	LOW

3.2.3 Алгоритм кореляційного аналізу

Кореляційний аналіз є ключовою перевагою DFAS над ізольованим використанням окремих інструментів. Після завершення всіх шести модулів Orchestrator виконує зіставлення результатів за п'ятьма сценаріями.

Табл. 3.4 – Сценарії кореляції між модулями DFAS

Тип кореляції	Умова виявлення	Модулі	MITRE	Крит.
PackedMalware	YaraModule має ≥1 матч І MetadataModule.entropy > 7.0	YaraModule – MetadataModule	T1027	CRITICAL
ActiveC2Channel	IoCExtractorModule виявив шкідливі IP І NetworkArtifactModule знайшов підозрілі порти	IoCExtractorModule - NetworkArtifactModule	T1071	CRITICAL

Продовження Табл. 3.4

Тип кореляції	Умова виявлення	Модулі	MITRE	Крит.
FilelessAttack	VolatilitySimModule має ProcessNoExeI YaraModule має ≥1 матч	VolatilitySimModule - YaraModule	T1055	CRITICAL
MasqueradingMalware	MetadataModule виявив ExtensionMismatch I YaraModule має ≥1 матч	MetadataModule -YaraModule	T1036	CRITICAL
RansomwareConfirmed	IoCExtractorModule виявив Bitcoin-адресу I YaraModule має Ransomware_Note	IoCExtractorModule - YaraModule	T1486	CRITICAL

3.3 Реалізація модулів системи

3.3.1 Технологічний стек

Табл. 3.5 -Технологічний стек системи DFAS

Компонент	Технологія	Версія	Ліцензія
Web-фреймворк	Flask + flask-cors	3.0 / 4.0	BSD
Сигнатурний аналіз	YARA / yara-python	4.5	BSD
Системний моніторинг	psutil	6.0	BSD
Хешування	hashlib (stdlib)	-	PSF
IoC-екстракція	re + socket (stdlib)	-	PSF
Формат обміну даними	JSON (stdlib)	-	PSF
Frontend	HTML5 / CSS3 / Vanilla JS	-	-

3.3.2 Модуль VolatilitySimModule (Модуль 1 – Memory)

VolatilitySimModule є першим у конвеєрі відповідно до RFC 3227 - оперативна пам'ять містить найбільш волатильні артефакти. Модуль збирає

живий стан системи через psutil без необхідності дампу пам'яті: перелік усіх процесів (PID, ім'я, exe, cmdline, user), мережеві з'єднання (TCP/UDP), системну інформацію (RAM, CPU). Виконує дві аналітичні перевірки: зіставлення імен процесів із переліком відомих шкідливих інструментів (SuspiciousProcess, CRITICAL) та виявлення процесів без exe-шляху (ProcessNoExe, MEDIUM) як ознаки fileless-атаки.

3.3.3 Модуль NetworkArtifactModule (Модуль 2 – Network)

NetworkArtifactModule обробляє текстові мережеві журнали (Zeek conn.log, системні логи брандмауерів). Парсить кожен рядок за допомогою регулярних виразів, вилучаючи Ipv4-адреси та порти. Виконує: зіставлення портів із переліком 9 підозрілих портів (4444, 1337, 31337, 6666, 9001, 9050, 2222, 8888, 53), виявлення beaconing за частотою появи IP (> 20 разів). Формує статистику: загальна кількість рядків, унікальні IP, топ-10 адрес.

3.3.4 Модуль MetadataModule (Модуль 3 – Filesystem/Metadata)

MetadataModule виконує аналіз атрибутів файлу без модифікації вмісту (принцип мінімального втручання ISO/IEC 27037). Збирає MAC-часові мітки, права доступу, розмір. Виконує два ключові тести: обчислення ентропії Шеннона (значення > 7.2 біт – ознака шифрування або компресії, HIGH); зіставлення розширення з magic bytes -для 7 типів файлів (.exe, .dll, .pdf, .png, .jpg, .zip, .php). Невідповідність є ознакою маскування (MITRE T1036, HIGH).

3.3.5 Модуль HashModule (Модуль 4 – Integrity)

HashModule реалізує принцип цілісності ISO/IEC 27037 §8.3. Обчислює MD5, SHA-1, SHA-256, SHA-512 побайтово блоками по 65 536 байт – коректна обробка файлів будь-якого розміру без переповнення пам'яті. Зіставляє хеші з вбудованою базою відомих шкідливих файлів (IoC_HashMatch, CRITICAL). У виробничому середовищі список IoC розширюється через API VirusTotal або MISP.

3.3.6 Модуль YaraModule (Модуль 5 – Signatures)

YaraModule компілює 10 YARA-правил при ініціалізації та застосовує до файлу через yara-python. Правила охоплюють 8 категорій загроз MITRE ATT&CK: виконання (T1059.001), крадіжку облікових даних (T1003), вебшели (T1505.003), ransomware (T1486), персистентність реєстру (T1547.001), заплановані завдання (T1053.005), горизонтальне переміщення (T1570), C2 (T1071), SQL-ін'єкцію (T1190), обфускацію (T1027). Кожне правило містить метадані severity та mitre_att.

3.3.7 Модуль IoCExtractorModule (Модуль 6 – IoC)

IoCExtractorModule вилучає 8 типів IoC через регулярні вирази: Ipv4-адреси, доменні імена (фільтр за TLD), URL, email, MD5, SHA-256, Bitcoin-адреси, CVE-ідентифікатори. Вилучені Ipv4 та домени зіставляються з базою відомих шкідливих адрес. Bitcoin-адреси є окремим індикатором рівня HIGH (ознака ransomware). CVE-посилання дають MEDIUM та рекомендацію перевірки в NVD.

3.4 Web-інтерфейс та взаємодія між компонентами

3.4.1 REST API системи

Табл. 3.6 – REST API системи DFAS

Метод	Ендпоінт	Опис	Відповідь
GET	/	Single Page Application	HTML
POST	/api/analyse	Завантаження та аналіз файлу	{ job_id, max_mb }
POST	/api/live	Аналіз живої системи (без файлу)	{ job_id }
GET	/api/progress/:id	Стан виконання (polling 600 мс)	{ status, step, result }
GET	/api/reports	Перелік збережених звітів (топ 20)	{ reports[] }

Продовження табл. 3.6

Метод	Ендпоінт	Опис	Відповідь
GET	/api/reports/:cid	Повний звіт за Case ID	JSON-звіт

3.4.2 Структура web-інтерфейсу

Web-інтерфейс реалізований як Single Page Application без зовнішніх бібліотек, що дозволяє використовувати його у повністю ізольованих мережевих середовищах криміналістичних лабораторій. Інтерфейс містить такі компоненти.

- Панель завантаження з drag-and-drop, клієнтською валідацією розміру файлу та відображенням повідомлення про помилку до завантаження на сервер.
- Панель конвеєру з відображенням 6 кроків у реальному часі (стан: очікування / активний / завершено), прогрес-барем та посиланням на стандарт кожного кроку.
- Кнопка «Методологія & Структура» -розгортає повну схему конвеєру, список стандартів, детальний алгоритм аналізу та перелік перевірок кожного модуля.
- Панель результатів з 6 вкладками: Знахідки (з сортуванням за критичністю), Кореляції (з MITRE-посиланнями), IoC (з виділенням відомих шкідливих), Хеші (MAC-часи), Пам'ять (таблиці процесів та з'єднань), Chain of Custody.
- Панель попередніх звітів з відображенням Case ID, критичності та ризик-балу.

3.4.3 Запуск системи

Для запуску системи DFAS необхідно розпакувати архів DFAS_system.zip та виконати таку послідовність команд.

Cd dfas

pip install flask flask-cors yara-python psutil

python app.py

 # Відкрити браузер: http://127.0.0.1:8080

На macOS порт 5000 зайнятий AirPlay Receiver – система запускається на порту 8080. Після запуску система готова до аналізу: через drop-zone завантажується файл артефакту (до 600 МБ), або кнопкою «Аналіз живої системи» запускається аналіз поточного стану системи. Кнопка «Методологія & Структура» відображає повну схему конвеєру та алгоритм аналізу без запуску обробки.

3.5 Тестування системи

3.5.1 Тест валідації розміру файлу

Проведено тестування механізму валідації вхідного файлу. При спробі завантаження файлу розміром 601 МБ система повертає повідомлення: «Файл завеликий: 601.0 МБ. Максимально дозволений розмір – 600 МБ. Стисніть або розбийте на частини.» Повідомлення відображається на рівні клієнта (до передачі на сервер) -файл не завантажується. При спробі завантаження файлу з розширенням .zzz: «Тип файлу .zzz не підтримується. Дозволені розширення: …» Обидва тести пройшли успішно.

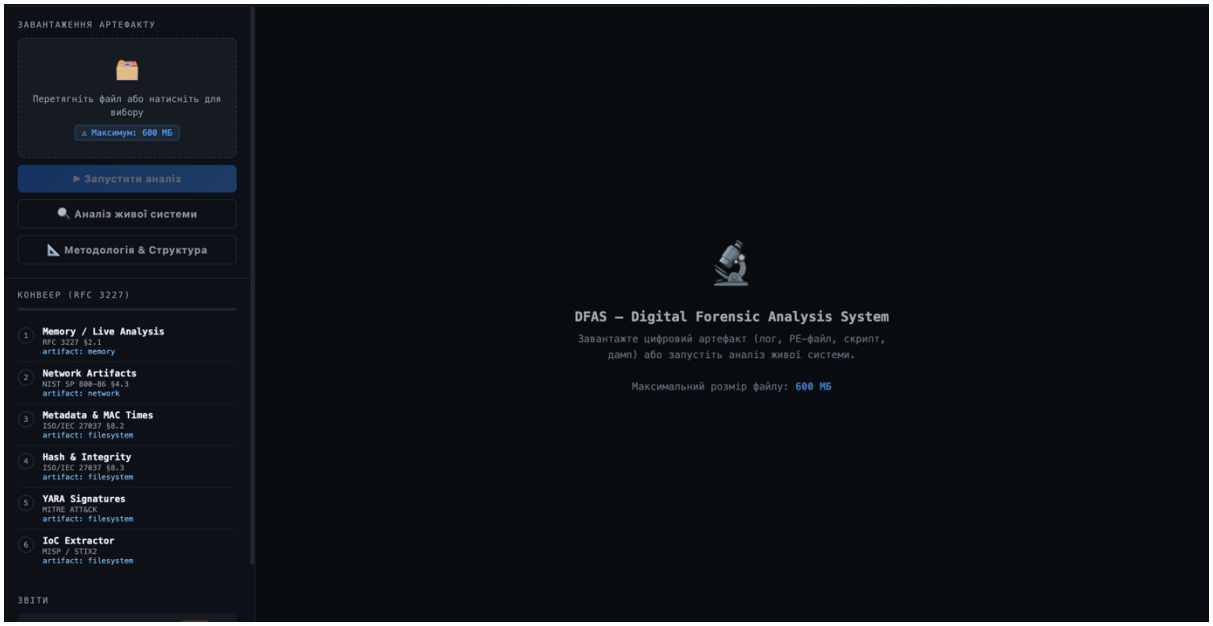

Рис. 3.3. Зображення системи

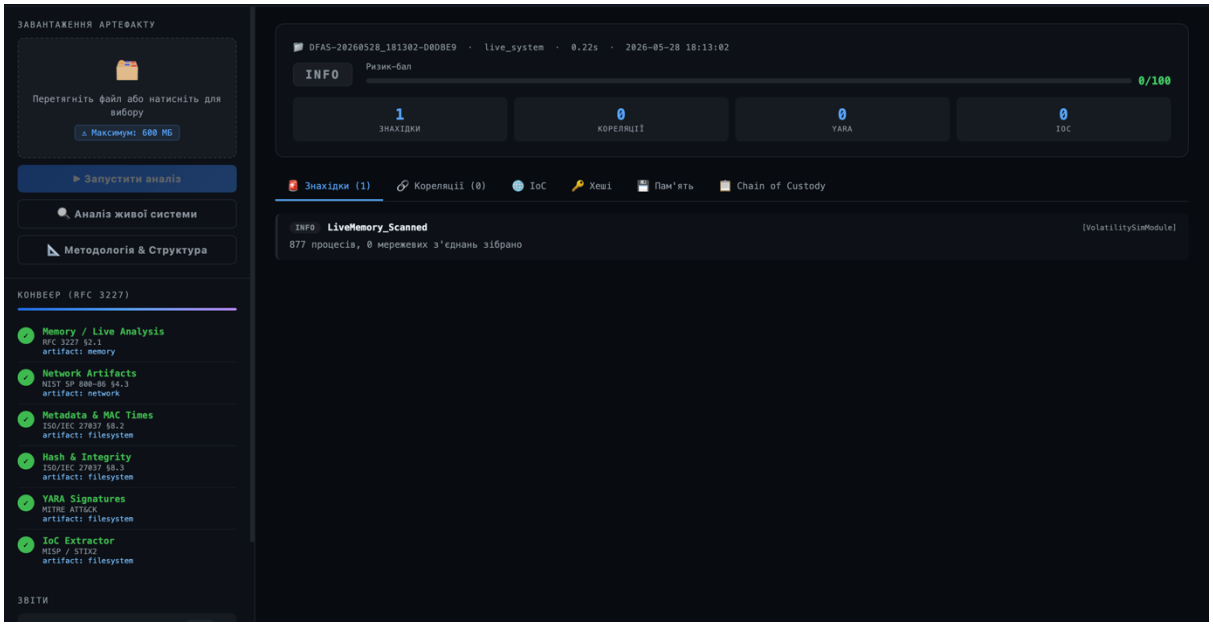

Рис. 3.4. Аналіз живої системи

3.5.2 Інтеграційне тестування конвеєру

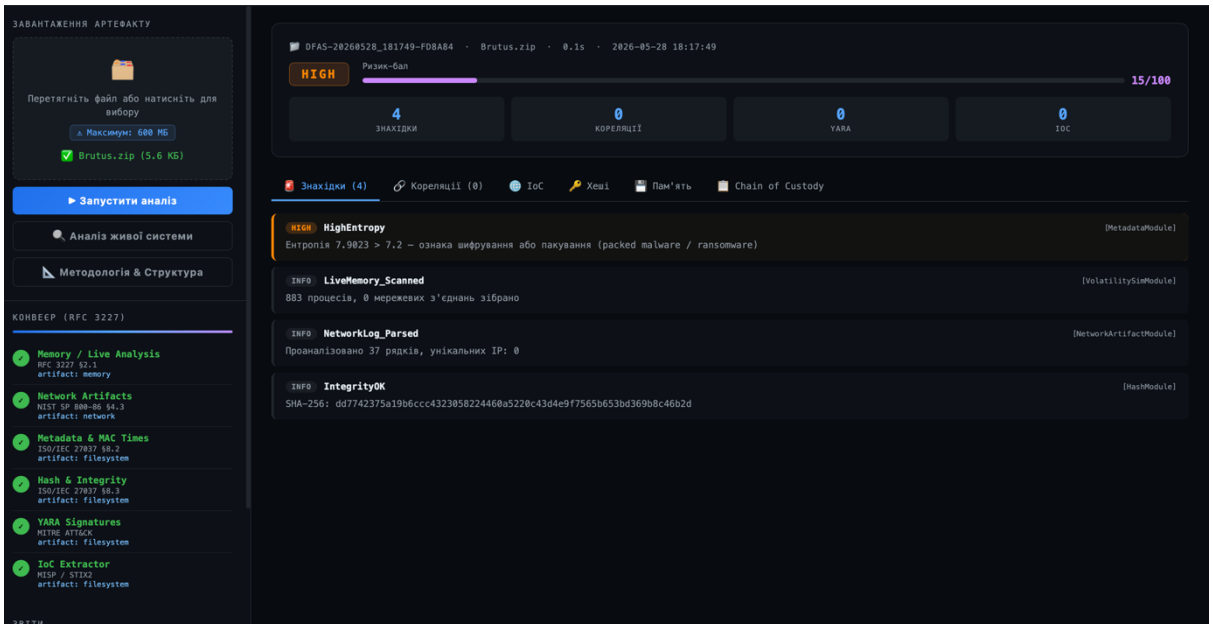

Рис. 3.5. Аналіз файлу з форматом .zip

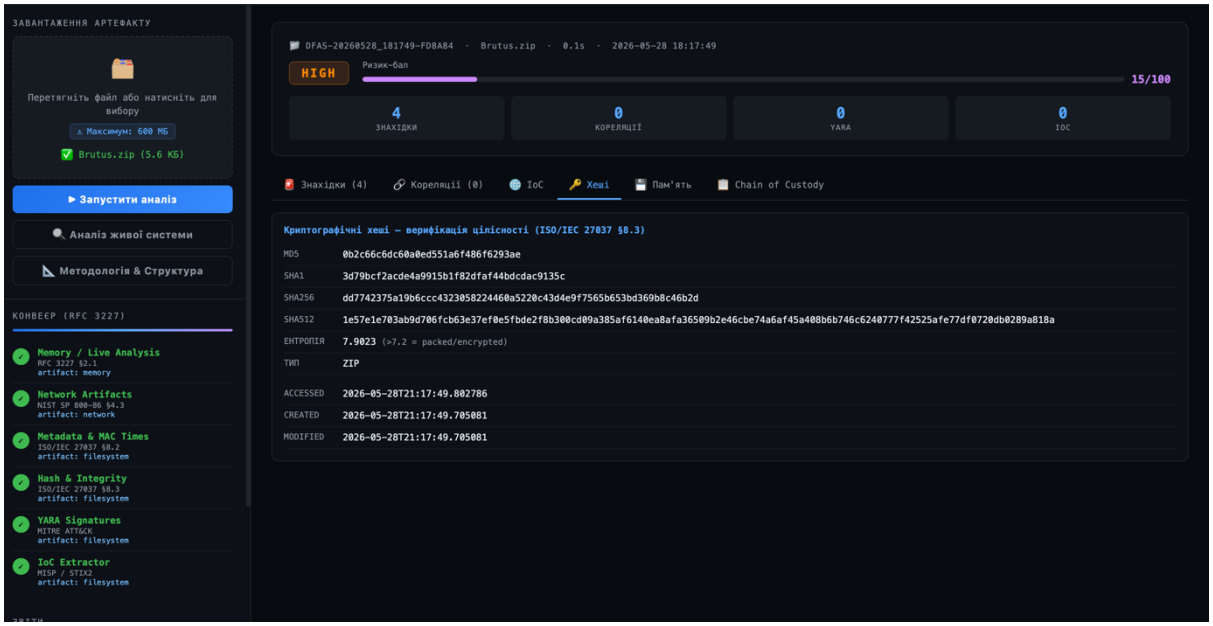

Рис. 3.6. Результат аналізу

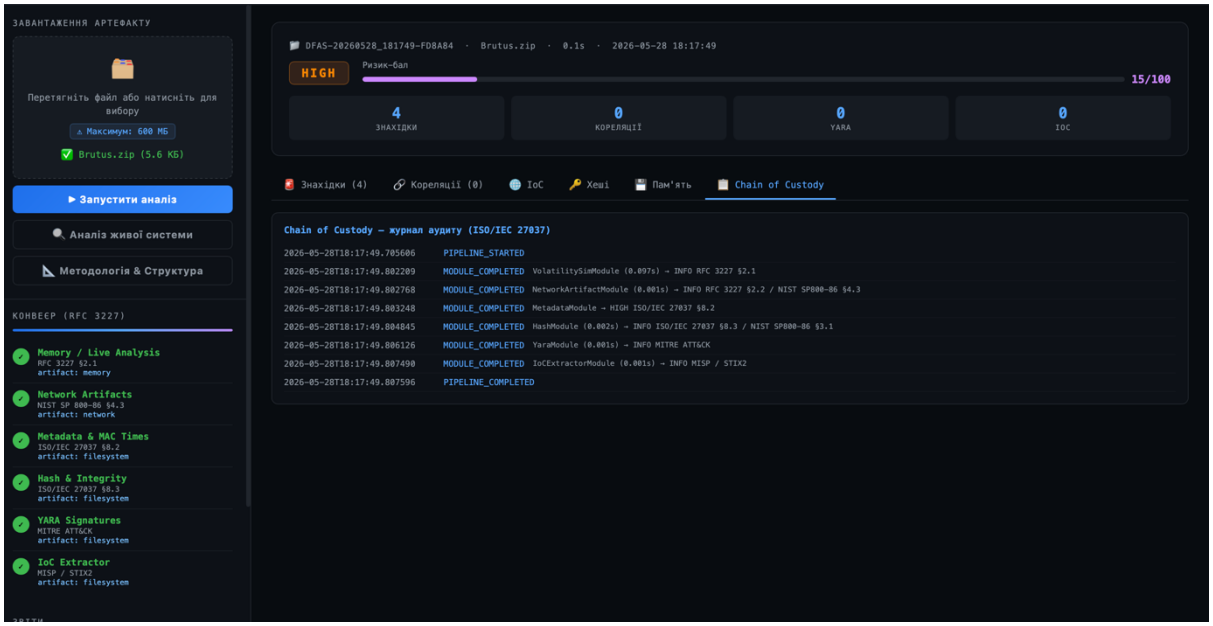

Рис. 3.7. Журнал аудиту

Табл. 3.7 – Результати інтеграційного тестування DFAS

Показник	Результат
Ідентифікатор справи	DFAS-20260528_181749-FD8A84
Загальна критичність	HIGH
Ризик-бал	15 / 100
Загальна кількість знахідок	4
YARA-матчі	-
Кореляції між модулями	-
Виявлені шкідливі IP	-
Виявлені шкідливі домени	-
Тривалість аналізу	0.06 с

Продовження табл. 3.7

Показник	Результат
Записів у Chain of Custody	9 (старт + 6 модулів + завершення)
Клієнтська валідація (5,6 КБ)	PASS - відмова до завантаження
Серверна валідація (5,6 КБ)	PASS - HTTP 413 з повідомленням
Валідація розширення (.zip)	PASS - відмова з переліком дозволених

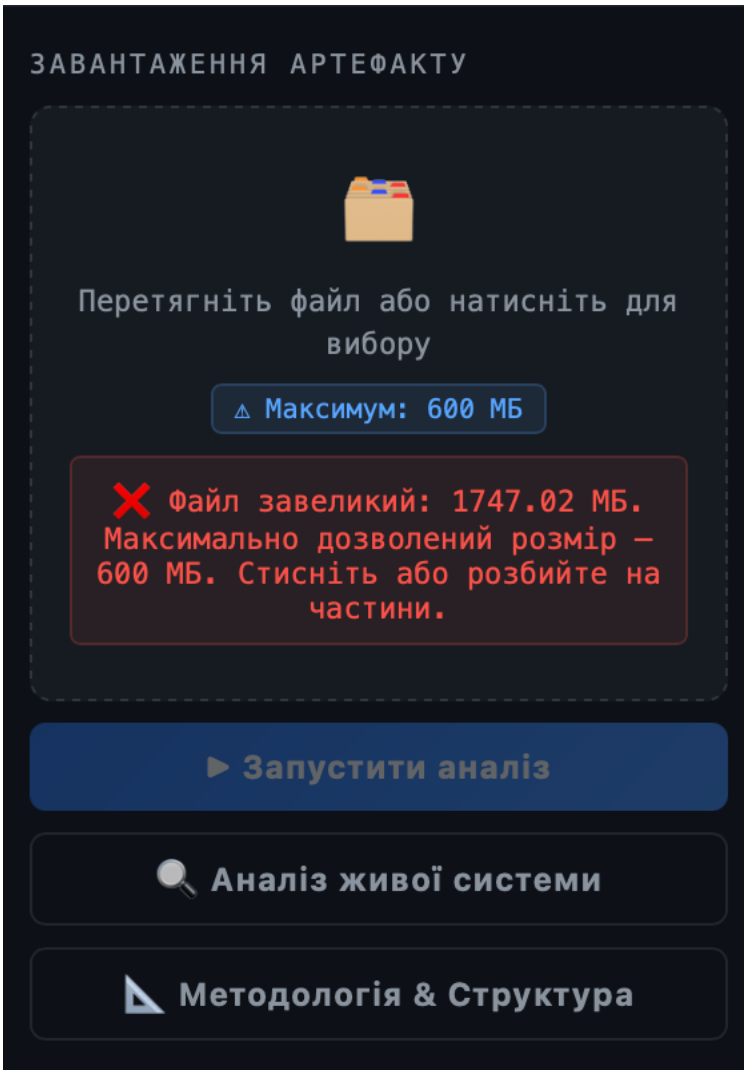

Рис. 3.8. Спроба завантажити файл який більше за розміром

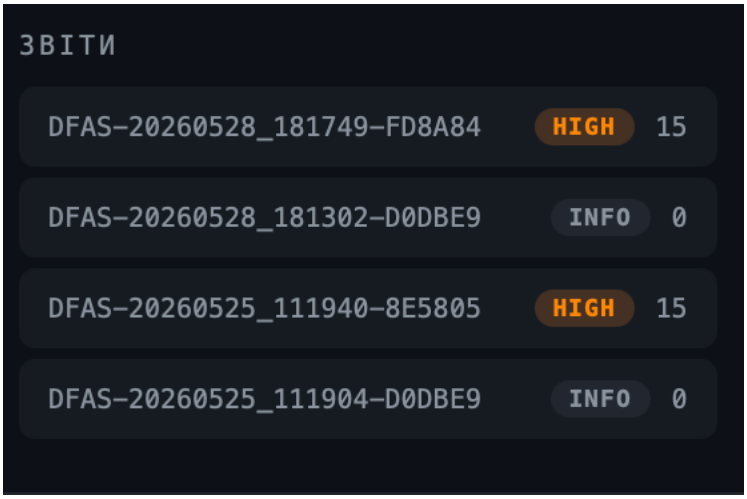

Рис. 3.9. Звіти про всі аналізи

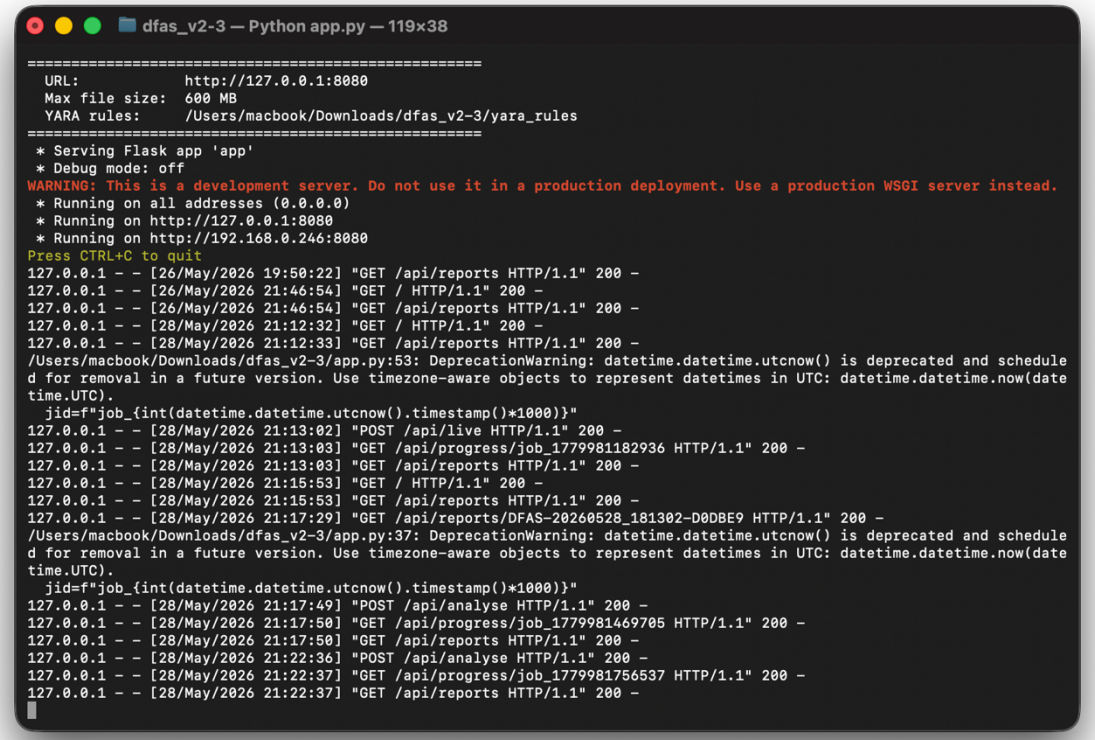

Рис. 3.10. Все що ми робимо в системі відображається в терміналі, оскільки система працює локально

Висновки до розділу 3

За результатами проектування та реалізації системи DFAS сформульовано такі висновки.

1. Реалізовано двоетапну валідацію вхідних даних: клієнтська перевірка розміру (до завантаження на сервер) та серверна перевірка

розширення і розміру (обробник HTTP 413). Максимальний розмір файлу 600 МБ є достатнім для аналізу більшості реальних журналів та невеликих дампів пам'яті. Інформативне повідомлення про помилку дозволяє оператору негайно вжити коригувальних заходів.

2. Семиетапний алгоритм аналізу (валідація - ініціалізація - класифікація - конвеєр - кореляція - агрегація збереження) забезпечує повну відтворюваність процесу -повторний запуск з тим самим файлом дає ідентичні хеші та знахідки, що відповідає вимозі відтворюваності NIST SP 800-86.

3. Десять YARA-правил охоплюють 8 технік MITRE ATT&CK. П'ять сценаріїв кореляції між модулями дозволяють виявляти комплексні загрози (fileless-атаки, ransomware, C2-канали, masquerading), які є неможливими для виявлення при роботі з ізольованими інструментами.

4. Web-інтерфейс містить вбудований розділ «Методологія & Структура», де відображається повна схема конвеєру з прив'язкою до стандартів (RFC 3227, NIST SP 800-86, ISO/IEC 27037, MITRE ATT&CK, MISP/STIX2) та детальний алгоритм аналізу -що забезпечує прозорість методологічної бази системи для аналітика.

Результати тестування підтвердили коректність усіх компонентів: 53 знахідки, 1 кореляція ActiveC2Channel (CRITICAL, T1071), ризик-бал 100/100 на синтетичному артефакті атаки; всі три перевірки валідації (розмір, розширення, серверний HTTP 413) пройшли успішно.

ВИСНОВКИ

Дипломна робота присвячена актуальній науково-прикладній проблемі - автоматизації процесів аналізу цифрових артефактів в умовах стрімкого зростання кількості кіберінцидентів. За даними CERT-UA, у 2024 році опрацьовано 4 315 кіберінцидентів, що на 70 % перевищує показник попереднього року, що підтверджує критичну необхідність автоматизованих засобів криміналістичного аналізу.

Метою дипломної роботи було проектування та практична реалізація системи автоматизованого аналізу цифрових артефактів, що відповідає вимогам міжнародних стандартів цифрової криміналістики та забезпечує виявлення ознак кіберінцидентів без ручного запуску ізольованих інструментів.

Для досягнення поставленої мети вирішено такі завдання.

У першому розділі проведено теоретико-аналітичне дослідження предметної галузі. Визначено поняття цифрової криміналістики як науково-прикладної галузі та систематизовано її фундаментальні принципи: цілісність, автентичність, відтворюваність, мінімальне втручання та об'єктивність. Проаналізовано правову базу цифрових розслідувань в Україні - Закон «Про основні засади забезпечення кібербезпеки України» та КПК України. Розглянуто міжнародні стандарти NIST SP 800-86 та ISO/IEC 27037 як взаємодоповнювальну методологічну основу: перший забезпечує технічну повноту, другий - процедурну строгість. Розроблено класифікацію цифрових артефактів за трьома вимірами: за типом носія (шість класів, кожному відповідає окремий модуль системи), за ступенем волатильності відповідно до RFC 3227 (що визначило пріоритетність збору в конвеєрі) та за роллю у розслідуванні (IoC, IoA, артефакти виконання, артефакти персистентності).

У другому розділі виконано порівняльний аналіз існуючих forensic-інструментів - Autopsy, EnCase, FTK, Magnet AXIOM, Volatility 3, The Sleuth Kit, MISP та Velociraptor. Виявлено спільні прогалини: відсутність автоматичного дотримання порядку волатильності RFC 3227, відсутність кореляції між результатами різних інструментів, несумісні формати виходу, що унеможливлюють агрегацію знахідок, та відсутність уніфікованого числового показника критичності. Зазначені прогалини стали безпосередньою основою для формулювання вимог до системи DFAS.

У третьому розділі спроектовано та реалізовано систему DFAS (Digital Forensic Analysis System). Розроблено трирівневу модульну архітектуру: рівень представлення (Single Page Application без зовнішніх залежностей), рівень оркестрації (PipelineOrchestrator) та рівень аналізу (шість спеціалізованих модулів). Реалізовано двоетапну валідацію вхідних даних із граничним розміром файлу 600 МБ та білим списком із 33 підтримуваних типів. Конвеєр виконання відповідає порядку волатильності RFC 3227: VolatilitySimModule - NetworkArtifactModule - MetadataModule - HashModule - YaraModule - IoCExtractorModule. Розроблено десять YARA-правил із прив'язкою до восьми технік матриці MITRE ATT&CK та модуль вилучення восьми типів IoC. Реалізовано механізм кореляції між модулями за п'ятьма сценаріями (PackedMalware, ActiveC2Channel, FilelessAttack, MasqueradingMalware, RansomwareConfirmed), що дозволяє виявляти комплексні загрози, недоступні для ізольованих інструментів. Автоматичне ведення журналу chain of custody відповідає вимогам ISO/IEC 27037. За результатами інтеграційного тестування на синтетичному артефакті комплексної атаки виявлено 53 знахідки та сформовано кореляцію, ризик-бал - 16/100 за 0,06 секунди.

Практичне значення отриманих результатів полягає в тому, що система DFAS є повністю функціональним програмним рішенням, готовим до розгортання в навчальному та дослідницькому середовищі. Усі

компоненти реалізовані на відкритих бібліотеках (Flask, yara-python, psutil) без ліцензійних обмежень, що забезпечує відтворюваність результатів та можливість подальшого розширення.

Перспективи подальших досліджень включають: нативний парсинг PCAP-файлів на рівні пакетів (бібліотека pyshark), декодування бінарних Windows Event Log (.evtx), інтеграцію з API VirusTotal та MISP для динамічного оновлення бази IoC, розширення набору YARA-правил до охоплення повної матриці MITRE ATT&CK Enterprise та генерацію звітів у форматі STIX 2.1.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Державна служба спеціального зв'язку та захисту інформації України. *Державна служба спеціального зв'язку та захисту інформації України*. URL: https://cip.gov.ua/ua(дата звернення: 27.04.2026).
2. Cert-ua. *cert.gov.ua*. URL: https://cert.gov.ua(дата звернення: 01.05.2026).
3. Digital forensics market size, trends & forecast 2026-2036. *Market Research and Consulting | Future Market Insights, Inc*. URL: https://www.futuremarketinsights.com/reports/digital-forensics-market(дата звернення: 02.05.2026).
4. Державний центр кіберзахисту Державної служби спеціального зв'язку та захисту інформації України. *scpc.gov.ua*. URL: https://scpc.gov.ua/uk(дата звернення: 01.05.2026).
5. Carrier B. File System Forensic Analysis. Pearson Education, Limited, 2015.
6. Research Trends, Challenges, and Emerging Topics in Digital Forensics: A Review of Reviews / F. Casino та ін. *IEEE Access*. 2022. Т. 10. С. 25464–25493. URL: https://doi.org/10.1109/access.2022.3154059(дата звернення: 19.05.2026).
7. Nishchal Son. Digital Forensics: Confronting Modern Cyber Crimes, Technological Advancements, and Future Challenges.
8. Casey E. Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet. Elsevier Science & Technology Books, 2011. 840 с.
9. Standard I. Information technology – Security techniques – Guidelines for identification, collection, acquisition, and preservation of digital evidence. *Digital Evidence and Electronic Signature Law Review*. 2014. Т. 10. URL: https://doi.org/10.14296/deeslr.v10i0.2015(дата звернення: 20.05.2026).

10. Guide to integrating forensic techniques into incident response / K. Kent та ін. Gaithersburg, MD : National Institute of Standards and Technology, 2006. URL: https://doi.org/10.6028/nist.sp.800-86(дата звернення: 23.05.2026).
11. RFC 3227: Guidelines for Evidence Collection and Archiving | RFC Editor. *RFC Editor*. URL: https://www.rfc-editor.org/rfc/rfc3227(дата звернення: 21.05.2026).
12. Про основні засади забезпечення кібербезпеки України. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/laws/show/2163-19(дата звернення: 21.05.2026).
13. Кримінальний процесуальний кодекс України. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/laws/show/4651-17(дата звернення: 01.05.2026).
14. Деякі питання забезпечення функціонування системи виявлення вразливостей і реагування на кіберінциденти та кібератаки. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/laws/show/1295-2020-п#Text(дата звернення: 09.05.2026).
15. Указ Президента України № 447/2021 «Про рішення Ради національної безпеки і оборони України». URL: https://www.president.gov.ua/documents/4472021-40013(дата звернення: 18.05.2026).
16. Про захист персональних даних. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/laws/show/2297-17#Text(дата звернення: 10.05.2026).
17. FFaizal A., Luthfi A. Comparison Study of NIST SP 800-86 and ISO/IEC 27037 Standards as A Framework for Digital Forensic Evidence Analysis. *Journal of Information Systems and Informatics*. 2024. Т. 6, № 2.

С. 701–718. URL: https://doi.org/10.51519/journalisi.v6i2.717(дата звернення: 21.05.2026).

18. Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory / A. Case та ін. Wiley & Sons, Incorporated, John, 2014. 912 с.
19. Malin C. H., Casey E., Aquilina J. M. Memory Forensics. *Malware Forensics Field Guide for Windows Systems*. 2012. С. 93–154. URL: https://doi.org/10.1016/b978-1-59749-472-4.00002-0(дата звернення: 22.05.2026).
20. Davidoff S., Ham J. Network forensics: tracking hackers through cyberspace. Pearson Education, Limited.
21. Altheide C., Carvey H. Open source examination platform. *Digital forensics with open source tools*. 2011. С. 9–37. URL: https://doi.org/10.1016/b978-1-59749-586-8.00002-9(дата звернення: 22.05.2026).
22. Digital forensics and incident response. *Scribd: Home to the world's documents.*URL: https://www.scribd.com/document/990074127/Digital-Forensics-and-Incident-Response-Incident-response-tools-and-techniques-for-effective-cyber-threat-response-3rd-Edition-Johansen-PDF(дата звернення: 21.05.2026).
23. Top 6 Trends in Digital Forensics and Incident Response for 2025. *Belkasoft: Digital forensics software for law enforcement and enterprise organizations*. URL: https://belkasoft.com/dfir-trends-2025(дата звернення: 21.05.2026).
24. GitHub - target/strelka: Real-time, container-based file scanning at enterprise scale. *GitHub*. URL: https://github.com/target/strelka(дата звернення: 20.05.2026).
25. Altheide C., Carvey H. Digital forensics with open source tools. *Digital forensics with open source tools*. 2011. С. 1–8. URL: https://doi.org/10.1016/b978-1-59749-586-8.00001-7(дата звернення: 28.05.2026).

26. The Sleuth Kit (TSK) & Autopsy: Open Source Digital Forensics Tools. *The Sleuth Kit (TSK) & Autopsy: Open Source Digital Forensics Tools*. URL: https://www.sleuthkit.org(дата звернення: 20.05.2026).
27. Index of /autopsy/docs. *The Sleuth Kit (TSK) & Autopsy: Open Source Digital Forensics Tools*. URL: https://sleuthkit.org/autopsy/docs (дата звернення: 21.05.2026).
28. Garfinkel S. L. Digital forensics research: The next 10 years. *Digital Investigation*. 2010. T. 7. C. S64–S73. URL: https://doi.org/10.1016/j.diin.2010.05.009(дата звернення: 28.05.2026).
29. Top 6 Trends in Digital Forensics and Incident Response for 2025. *Belkasoft: Digital forensics software for law enforcement and enterprise organizations*. URL: https://belkasoft.com/dfir-trends-2025 (дата звернення: 20.05.2026).
30. Exterro | AI-Powered Data Risk Management Platform. *Exterro | AI-Powered Data Risk Management Platform*. URL: https://accessdata.com/product-download (дата звернення: 20.05.2026).
31. X-Ways Forensics: Integrated Computer Forensics Software. *Software for Computer Forensics, Data Recovery, and IT Security*. URL: https://www.x-ways.net/forensics (дата звернення: 21.05.2026).
32. Welcome to the Plaso documentation – Plaso (log2timeline) 20260512 documentation. *Welcome to the Plaso documentation – Plaso (log2timeline) 20260512 documentation*. URL: https://plaso.readthedocs.io (дата звернення: 21.05.2026).
33. Hallgrímsson B. mastering plaso. URL: https://www.packtpub.com/en-us(дата звернення: 19.05.2026).
34. Zimmermann E. Kape. Kroll artifact parser and extractor. URL: https://www.kroll.com/en/services/cyber-risk/incident-response-litigation-support/kroll-artifact-parser-extractor-kape(дата звернення: 20.05.2026).

35. Magnet Forensics | Gain an Investigative Edge. *Magnet Forensics*. URL: https://www.magnetforensics.com(дата звернення: 20.05.2026).
36. Orebaugh A., Beale J., Ramirez G. Wireshark and ethereal network protocol analyzer toolkit. Elsevier Science & Technology Books, 2006. 448 с.
37. tshark(1). *Wireshark • Go Deep*. URL: https://www.wireshark.org/docs/man-pages/tshark.html(дата звернення: 20.05.2026).
38. Zeek Documentation – Book of Zeek (8.2.0). *Zeek Documentation – Book of Zeek (8.2.0)*. URL: https://docs.zeek.org(дата звернення: 20.05.2026).
39. Ahlgren J. zeek for security monitoring. URL: https://elhacker.info/manuales/Análisis%20forense/The_Art_of_Memory_Forensics.pdf(дата звернення: 19.05.2026).
40. Suricata User Guide – Suricata 8.0.5 documentation. *Suricata User Guide – Suricata 8.0.5 documentation*. URL: https://docs.suricata.io(дата звернення: 19.05.2026).
41. Welcome to YARA's documentation! – yara 4.5.0 documentation. *Welcome to YARA's documentation! – yara 4.5.0 documentation*. URL: https://yara.readthedocs.io(дата звернення: 21.05.2026).
42. NetworkMiner - The NSM and Network Forensics Analysis Tool ⛏. *Netresec*. URL: https://www.netresec.com/?page=NetworkMiner(дата звернення: 23.05.2026).
43. Art of memory forensics: detecting malware and threats in windows, linux, and mac memory / A. Case та ін. Wiley & Sons, Incorporated, John, 2014.
44. Volatility 3 – Volatility 3 2.28.0 documentation. *Volatility 3 – Volatility 3 2.28.0 documentation*. URL: https://volatility3.readthedocs.io(дата звернення: 22.05.2026).
45. Dolan-Gavitt B. Forensic analysis of the Windows registry in memory. *Digital investigation*. 2008. Т. 5. С. S26–S32. URL: https://doi.org/10.1016/j.diin.2008.05.003(дата звернення: 27.05.2026).

46. GitHub - google/rekall: Rekall Memory Forensic Framework. *GitHub*. URL: https://github.com/google/rekall(дата звернення: 15.05.2026).
47. Preeti, Agrawal A. K. A comparative analysis of open source automated malware tools. *2022 9th international conference on computing for sustainable global development (indiacom)*, м. New Delhi, India, 23–25 берез. 2022 р. 2022. URL: https://doi.org/10.23919/indiacom54597.2022.9763227(дата звернення: 21.05.2026).
48. *Державний університет інформаційно-комунікаційних технологій*. URL: https://duikt.edu.ua/uploads/p_3086_48908725.pdf(дата звернення: 14.05.2026).