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

«До захисту допущено»
В.о. завідувача кафедри

_____ М.В.Грайворонський
(підпис)

“___”_____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: _____«Покращення ефективності DLP з трасуванням системних викликів»___

Виконала: студентка __4__ курсу, групи __ФБ-52_____
(шифр групи)

Калиновська Валентина Вадимівна _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник __доцент кафедри ІБ, кандидат наук_ Барановський О.М. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

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

Київ - 2019 року

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

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)

«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

Калиновській Валентині Вадимівні_____
(прізвище, ім'я, по батькові)

1. Тема роботи «Покращення ефективності DLP з трасуванням системних викликів» _____
_____ ,
науковий керівник роботи Барановський Олексій Миколайович_____
доцент кафедри ІБ, кандидат наук_____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «____ » 2019 р. № _____

2. Термін подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи _____
_____
_____

4. Зміст роботи_____
_____
_____
_____
_____

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

6. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка

Студент _____ _____
(підпис) (ініціали, прізвище)

Науковий керівник роботи _____ _____
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Тема роботи розкрита в пояснювальній записці обсягом 58 сторінок та складається з 4 розділів. Містить 2 ілюстрації, 11 таблиць, 22 літературних посилань.

Основна мета даної роботи полягає у покращенні системи DLP. Для цього треба провести дослідження існуючих на ринку систем DLP, розкрити їх слабкі сторони і знайти шляхи поліпшення їх роботи.

Об'єктом дослідження виступають такі системи DLP, як Forcepoint, McAfee, Digital Guardian.

Як предмет дослідження розглядається здатність систем DLP реагувати на інциденти, пов'язані з витоком інформації.

При написанні роботи було проведено ознайомлення з літературою, тестування і аналіз DLP-систем, було знайдено гіпотетичні рішення існуючих проблем та розроблено доповнення, яке можна інтегрувати з DLP-системами для покращення їх функціонування.

Результатом є гіпотетичне рішення для усунення недоліків та підвищення ефективності систем DLP, а також розробка доповнення, яке можна інтегрувати з іншими рішеннями DLP.

Значення результатів роботи полягає у використанні даного додатку для підвищення ефективності систем DLP на рівні API.

DATA LEAKAGE PREVENTION (DLP), API HOOKING, FORCEPOINT, MCAFEE, DIGITAL GUARDIAN, ТРАСУВАННЯ СИСТЕМНИХ ВИКЛИКІВ

РЕФЕРАТ

Тема работы раскрыта в объяснительной записке объемом 58 страниц и состоит из 4 разделов. Содержит 2 иллюстрации, 11 таблиц, 22 литературных ссылок.

Основная цель данной работы состоит в улучшении системы DLP. Для этого нужно провести исследование существующих на рынке систем DLP, раскрыть их слабые стороны и найти пути улучшения их работы.

Объектом исследования выступают такие системы DLP, как Forcepoint, McAfee, Digital Guardian.

В качестве предмета исследования рассматривается способность систем DLP реагировать на инциденты, связанные с утечкой информации.

При написании работы было проведено ознакомление с литературой, тестирование и анализ DLP-систем, было найдено гипотетические решения существующих проблем и разработано дополнение, которое можно интегрировать с DLP-системами для улучшения их функционирования.

Результатом есть гипотетическое решение для устранения недостатков и повышение эффективности систем DLP, а также разработка дополнения, которое можно интегрировать с другими решениями DLP.

Значение результатов работы состоит в использовании данного дополнения для повышения эффективности систем DLP на уровне API.

DATA LEAKAGE PREVENTION (DLP), API HOOKING, FORCEPOINT, MCAFEE, DIGITAL GUARDIAN, ТРАССИРОВКА СИСТЕМНЫХ ВЫЗОВОВ

ABSTRACT

The work theme is opened in the explanatory note in volume of 58 pages and consists of four sections. Contains: 2 drawings, 11 tables and 22 references.

The purpose of this work is to improve the DLP system. In order to do this it's necessary to conduct a study of existing DLP systems on the market, reveal their weaknesses and find ways to improve their work.

The object of the study are DLP systems such as Forcepoint, McAfee, Digital Guardian.

The subject of the research is the ability of DLP systems to respond to information leakage incidents.

When writing the work a literature review, testing and analysis of DLP systems were carried out, hypothetical solutions to existing problems were found and an add-on was developed that can be integrated with DLP systems to improve their functioning.

The result is a hypothetical solution to eliminate the deficiencies and improve the efficiency of DLP systems, as well as the development of an add-on that can be integrated with other DLP solutions.

The value of the work is to use this add-on to improve the efficiency of DLP systems at the API level.

DATA LEAKAGE PREVENTION (DLP), API HOOKING, FORCEPOINT, MCAFEE, DIGITAL GUARDIAN, SYSTEM CALL TRACING

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DLP – Data Leakage (Loss) Prevention

HTTP – Hyper Text Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

API – Application Programming Interface

IDS – Intrusion Detection Systems

IPS – Intrusion Prevention Systems

MIME – Multipurpose Internet Mail Extension

ВСТУП

Інформація про кредитну картку та ідентифікаційний номер клієнтів є прикладами конфіденційної інформації, яка є дуже поширеною у корпоративному світі, це інформація, яка має бути обмеженою і не повинна пропускатись або поширюватися. Ситуація, коли працівник намагається надіслати таку інформацію конкуруючим компаніям, є прикладом витоку, який може мати руйнівні наслідки для компанії, наприклад, втрата клієнтів, призведення до судових справ та ін. Щоб запобігти подібним витокам, системи, призначені для виявлення та захисту конфіденційних даних, були запроваджені в 2006 році [6] і стали називатись Data Leakage (Loss) Prevention (DLP) – запобігання витоків інформації. Їх мета полягає в тому, щоб закупорювати існуючі точки витоку і блокувати несанкціоновані дії з конфіденційними даними.

Сьогодні існує багато вендорів з різними системами DLP для різних операційних систем і мобільних платформ. Але обмеження доступу працівників до повсякденної роботи іноді може викликати роздратування і фрустрацію. Це, у свою чергу, може призвести до того, що працівники намагатимуться знайти способи обійти систему.

Мета цієї роботи полягає в тому, щоб зрозуміти, як діють існуючі системи DLP, їх охоплення, їх слабкі місця і що може бути зроблено для їх поліпшення. Це робиться шляхом тестування трьох існуючих систем DLP і вивчення того, які точки витоку вони охоплюють, і чи ці системи можна обійти, та знайти метод, який покращить роботу системи.

Актуальність роботи зумовлюється тим, що в ній було впроваджено рішення, яке дозволяло покращити показники роботи системи, використовуючи трасування системних викликів, також зване API hooking, яке доповнює існуючу систему DLP. Це рішення базується на

запропонованих удосконаленнях, що внесені до кожної з перевірених систем DLP.

Для досягнення даної мети було поставлено такі завдання:

- встановлення систем DLP на віртуальні машини та їх налаштування;
- тестування систем, яке полягає в тому, щоб передати конфіденційні дані за допомогою веб-поштових клієнтів (Gmail), локальних поштових клієнтів (Windows Live Mail), програмного забезпечення для синхронізації хмар (Google диск), соціальних мереж (Facebook), програмного забезпечення для обміну миттєвими повідомленнями (Skype), програмного забезпечення для дистанційного керування (Teamviewer), принтери (мережеві та локальні) та пристрої USB;
- підведення підсумків тестування, які наведені в таблицях;
- створення додатку для покращення показників;
- тестування роботи систем з використанням створеного додатку.

Методами дослідження обрано: опрацювання літератури за даною темою, аналіз технічної документації, тестування систем.

Наукова новизна даної роботи полягає у створенні додатку до системи, який покращить показники роботи системи.

Практичне значення результатів роботи випливає з можливості використання створенного доповнення для покращення результатів роботи систем DLP, яке дозволить підвищити показники якості виявлення загроз витоку інформації.

Таким чином **об’єктом дослідження** є такі системи DLP, як Forcepoint, McAfee, Digital Guardian.

Предмет досліджень – здатність систем DLP реагувати на інциденти, пов'язані з витоком інформації.

1 ПРОБЛЕМАТИКА ВИТОКУ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ ТА DLP РІШЕННЯ

У цій главі буде проведене коротке ознайомлення з системами DLP, перераховано їх недоліки та існуючі методи, які дозволяють підвищити ефективність систем.

1.1 Існуючі недоліки систем DLP

DLP-система – це система запобігання витоку даних або запобігання втрат, яка попереджує втрату або просочування конфіденційних даних. Система DLP розроблена в основному для захисту від самих співробітників компанії, законних користувачів, які вже мають доступ до більшості внутрішніх систем і даних. Користувачам дозволяють минати більшу частину захисту системи, для того, щоб вони могли ефективно виконувати свою роботу [16]. Персонал можливо не хоче навмисно шкодити компанії, але й бувають випадки, коли робітники витягують інформацію в корисливих цілях.

Якщо несанкціонований користувач намагається пробити шлях до внутрішньої мережі, такі системи, як IDS, системи виявлення вторгнень, або IPS, система запобігання вторгнень, є відповідальними за виявлення та запобігання цього. Якщо зловмиснику вдасться пройти ці системи, система DLP оброблятиме цього користувача, як і будь-якого іншого користувача в межах внутрішньої мережі, блокувати будь-які дані, яким не дозволяється покидати межі системи. Іншими словами, система DLP є протилежною як до IPS, так і до IDS, контролюючи, що дії, які відбуваються.

Будь-яка інформація, що зберігається на жорсткому диску або на комп'ютері, вважається даними. Рішення DLP часто розглядають дані в трьох різних станах [3]:

- Data In Motion (DIM) - Дані, які в даний час передаються з одного місця в інше. Це можна вважати перенесенням файлу з одного жорсткого

диска на інший або, наприклад, надсилання електронної пошти через Інтернет.

- Data In Use (DIU) - Дані, які активно використовуються будь-яким програмним забезпеченням. Подібно текстовому документу, який не обов'язково було збережено на жорсткому диску.
- Data At Rest (DAR) - Дані, які зберігаються на жорсткому диску і не використовуються або передаються на цей момент.

Система DLP – це набір правил, які також називаються політиками і є принципом або протоколом для керівництва рішеннями і досягнення раціональних результатів. [20]

Ці політики встановлюються адміністратором і містять правила для мережі та кінцевих точок, повідомляють що дозволено, а що ні. Ці правила можуть бути встановлені на певній програмі або веб-сторінці. Також можуть бути встановлені різні правила в залежності від рівня доступу користувача. Чим конкретніше ці політики встановлюються, тим менше помилково позитивні та помилково негативні сигнали будуть видаватися системою DLP, що зробить систему більш точною.

Існує декілька дій, які може виконувати система DLP, тобто те, що система DLP повинна робити з певним типом файлу або вмістом. Це може бути блокування, відправлення в карантин або пропускання, це залежить від політики, встановленої адміністратором. Система може також реєструвати події з інформацією про те, хто це зробив, коли це сталося і які дії були вжиті.

Proxy – це сервер або комп'ютерна система, яка діє як посередник між клієнтами та іншими серверами. Він також може бути використаний для моніторингу та фільтрації мережевого трафіку, як вхідного, так і вихідного [21].

Підписи є більш складним способом ідентифікації типу або формату файлу, що включає аналіз файлу для шаблонів, відмінних від файлів магічних чисел, які відповідають певному типу.

Іншим способом ідентифікації типу файлу є перевірка MIME-типу файлу. MIME означає Multipurpose Internet Mail Extension, і одним з прикладів типу MIME є "application / zip", який призначений для файлів zip-архівів [19].

HTTP / HTTPS – це протокол зв'язку, який використовується для веб-серфінгу. HTTPS шифрує мережевий трафік і в основному використовується на веб-сайтах, що містять конфіденційну інформацію, наприклад, імена користувачів і паролі.

Різні системи DLP сьогодні пропонують в основному однакову функціональність у своїй системі. Деякі з цих функцій можуть забороняти користувачам додавати певний тип файлу в електронній пошті, блокувати та входити в журнал, коли хтось намагається скопіювати в буфер обміну або зробити знімок екрану, або відмовити у друці документів, які містять певне слово, речення, номер кредитної карти та ін. Система повинна дозволяти адміністратору вказувати політику з достатньою точністю, необхідною для своєї компанії.

Іншим важливим завданням як адміністратора, так і системи DLP є інформування працівників про те, що дозволено, а що ні. Якщо інцидент відбудеться випадково або навмисно, система повинна дати чітке пояснення того, що сталося, і чому цей файл був заблокований.

Однією з особливостей існуючих систем DLP є виявлення, де конфіденційні дані зберігаються в мережі за допомогою сканування файлової системи. Ці сканування повторюють всі доступні файли в мережі і в залежності від файлу, що виконує дію, відповідно до активних політик. Ці сканування дуже вимогливі до системи, яка сканується, і часто плануються, щоб вони працювали, а лише деякі використовують систему.

Сьогодні існує багато компаній, які надають DLP-системи для всіх типів операційних систем і мобільних пристроїв. Деякі з них є відкритими, а інші – комерційними, де компанія або посилатиме консультанта для встановлення та налаштування свого продукту для вас, або відправлятиме

джерела та програмне забезпечення, необхідні для встановлення самої системи. Важливо спочатку визначити, які можливі точки витоку в компанії і визначити, чи є ще яка-небудь система, яка може заважати системі DLP, перш ніж вирішити, яка з них найкраще підходить.

Насправді, рішення DLP повинні бути реалізовані та налаштовані так, щоб вони могли співіснувати і використовувати вже існуючі системи та дані [15]. Це означає, що система DLP повинна бути адаптована відповідно до існуючих систем, а не навпаки, щоб вона могла мати практичну реалізацію.

Кожна система DLP сьогодні відрізняється тим, як вони підходять до проблеми витоку даних, але багато важливих частин залишаються схожими. Більшість систем DLP мають заздалегідь визначені словники вмісту, які містять такі слова, як "Конфіденційно", "Секретний", "Номер кредитної картки / Інформація" тощо. Вони також йдуть зі словниками з шаблонами номерів кредитних карток для різних виробників, таких як Visa, Master Card, American express. Система DLP сканує файли, які потрібно передати, і якщо є номери, що відповідають будь-якому з цих шаблонів, це буде позначено як номер кредитної картки. Як згадувалося, залежно від політики, встановленої для такого типу події, система DLP виконуватиме визначену дію.

Деякі з них також мають можливість виконувати певні дії з певними типами файлів, наприклад, шляхом блокування розширення файлу або вивчення MIME-типу файлу. Якщо останній виконується, він буде цілісним, якщо розширення буде змінено. Якщо файл змінено з Smile.png на Smile.txt, система все одно розпізнає цей файл як зображення, а не текстовий файл через перевірку типу MIME.

Системи DLP не забезпечують повної безпеки для даних в інфраструктурі. Жодна з систем не дає захисту від витоків інформації на 100%. Наявність DLP не обов'язково має вплив на зменшення потенційної загрози, але може обмежити вплив атаки.

Велике значення в роботі систем DLP відіграє людський фактор. Часто відбуваються витоки даних через неправильне використання системи або через відсутність розуміння класифікації – що конфіденційне, а що ні.

Що стосується безпосередньо самої системи, то вона має свої слабкі місця. Наприклад, системи не підтримують повністю усі формати файлів; не розпаковують архіви та не аналізують вміст, навіть якщо всередині є конфіденційні дані; не ловлять витоки через тунелювання конфіденційної інформації в дозволені протоколи і взагалі не працюють з тунелюванням; не блокуються повністю усі способи знімків екранів; зашифровані файли не завжди виявляються системою; можна блокувати лише локально підключені принтери, а при друці за допомогою бездротового мережного принтера друк не блокується; системи DLP не розпізнають вмісту відеозаписів та аудіозаписів.

Також витоки даних можуть відбуватися через недобросовісних працівників компанії. При передачі текстових файлів або повідомлень може використовуватися примітивна стеганографія шляхом заміни нашого алфавіту англомовними або іншими символами. Це може створити купу складностей для DLP-рішень, адже передбачити усі комбінації і варіанти ключових слів просто неможливо. А у випадку використання відсоткового співпадіння кількість помилкових тривог істотно зросте, що серйозно ускладнить виявлення дійсно важливих або підозрілих операцій.

Обійти DLP систему також можна шляхом використання віртуальних машин, спеціалізованих стеганографічних утиліт, синтезаторів мови Text to Speech або навпаки Speech to Text, а також просто маючи гарну пам’ять, олівець чи фотоапарат.

1.2 Методи для підвищення ефективності DLP

Для попередження витоків інформації з мережі, яка захищається системою DLP найкраще було б використовувати комплексний підхід:

мінімізувати права та повноваження користувачів; відключення можливості вибору режимів завантаження ОС; приховування/перейменування служб, які відповідають за роботу сервісів DLP; контроль використання застосунків і форматів документів, які не прийняті в організації; використання чорних і білих списків для отримувачів документів незалежно від їх вмісту; використання корпоративних знімних носіїв і періодичний контроль на них слідів конфіденційних документів; використання електронного підпису в документах.

Висновки до розділу 1

На сьогодні системи DLP широко застосовуються для захисту від витоку інформації, які з розвитком інформаційних технологій трапляються все частіше. Є багато рішень, які різняться ціною і функціоналом. Компанії обирають системи залежно від своїх потреб і можливостей.

Системи DLP не забезпечують цілковитий захист інформації в мережі, на яку направлені. Інформація може поширюватися за межі компанії навмисно або ненавмисно, це можуть бути співробітники компанії, які за своєю необережністю втрачають контроль над даними за межами компанії, співробітники, які хочуть свідомо спричинити шкоди компанії або працівники компанії-конкурента, проникаючи в компанію під виглядом нового працівника вивідують необхідну для них інформацію.

В системі достатньо недоліків, щоб будь-який співробітник при бажанні міг обійти захист. Щоб це попередити, потрібно правильно впровадити систему, забезпечувати життєвий цикл роботи DLP-рішення та безкінечний моніторинг стану роботи DLP-системи та її компонентів.

Задача цієї роботи полягає в наступних діях:

- тестування і аналіз роботи існуючих DLP систем;
- ознайомлення з існуючими проблемами системи та шляхами, через які можуть відбуватися витоки інформації;

- огляд існуючих методів для покращення роботи DLP;
- пошук найкращого рішення;
- створення і впровадження методу, який покращить показники роботи системи.

2 АНАЛІЗ СПОСОБІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СИСТЕМ DLP

Для більш близького ознайомлення з недоліками та влучного підбору методу, який максимально ефективно покращить показники, в системі проведене тестування трьох систем DLP. Існують рішення для покращення результатів роботи системи, вони будуть розглянуті, проведеться порівняння їх ефективності та буде обрано найкращий варіант.

2.1 Тестування та оцінка ефективності DLP систем

Через обмежений доступ до ресурсів, тестове середовище складалося з віртуального середовища, де комп'ютери працювали на різних операційних системах:

- Windows 7 running service pack 1.
- Linux running Ubuntu 13.10
- Mac OS X Mavericks.

Використовуючи віртуальне середовище, перевагою є те, що досить легко створювати клони цілих віртуальних середовищ, що скоротить час, необхідний для налаштування всіх середовищ. Вона також дозволяє легко керувати і стрибати між середовищами, коли це необхідно.

Ці три системи DLP були встановлені і ретельно перевірені, по одному на цих комп'ютерах. Щоб покрити точку витоку зовнішніми жорсткими дисками та принтерами, використовується флеш-пам'ять USB і мережевий принтер. Експерименти також враховують, коли користувач має різні привілеї на локальному комп'ютері. Спроба експериментів полягала в тому, щоб організувати витік конфіденційних даних за допомогою веб-поштових клієнтів (Gmail), локальних поштових клієнтів (Windows Live Mail), програмного забезпечення для синхронізації хмар (Google диск), соціальних мереж (Facebook), програмного забезпечення для обміну миттєвими

повідомленнями (Skype), програмного забезпечення для дистанційного керування (Teamviewer)), принтери (мережеві та локальні) та пристрої USB. Дані для спроби витоку:

- Текстовий файл, що містить слова, які вважаються секретними: "Паспорт", "Ідентифікаційний код", "Кредитна картка" та "Конфіденційно". Потім вони були додані до власного словника вмісту, щоб побачити, чи може система виявляти файли, що містять ці слова. Номер кредитної картки: 4485 6305 9117 1087, також був доданий до текстового файлу для перевірки вбудованого CCN пошуку.
- Той самий текстовий файл, але стиснутий у zip-архів. Формати: .zip, .7z і .rar.
- Знову ж самий текстовий файл, але з зашифрованим файлом (він був збережений у контейнері файлів Truecrypt).
- Великий аудіофайл у розмірі 5370 кБ, .mp3.
- Велике .png зображення розміром 15 кБ.
- Стиснення зашифрованого текстового файлу в zip-архіві
- Файл Microsoft Office Word (.docx), оскільки цей тип файлу є архівом, який можна розпакувати, і файли можуть бути додані без відома.

Для кожного експерименту, один тип файлу блокувався, а всі інші були дозволені, щоб побачити, чи є який-небудь спосіб обійти і обдурити систему. Експерименти зі стиснутими файлами були зроблені двічі, у першому блокувалися стиснуті типи файлів, у другому – файли не блокувалися, але не дозволявся вміст текстового файлу. Те ж саме було зроблено з зашифрованими файлами. Тести з файлом Office були також зроблені двічі, один раз для блокування типу файлу .docx, а другий – для дозволу, але не дозволяючи зображення. Потім файл витягався .docx і зображення додавалось до архіву.

Тести проходили через три різні фази для кожного рішення DLP. На першому етапі була зроблена спроба встановити рішення DLP відповідно до наявних рекомендацій та інструкцій для запобігання витоку даних. Спроби переміщення/витоку даних були зроблені з використанням раніше згаданих програм і програмного забезпечення, а потім класифікували тестовий випадок за допомогою Passed / - Passed with comments або Failed залежно від того, чи вдалося запобігти витоку даних. Експерименти, які з будь-якої причини не могли бути проведені, були оцінені коментарем C, а тести, в яких система не охоплювала функціональність, відмітка була N/A. Цей етап також включав вивчення того, наскільки добре рішення повідомляє про запобігання як адміністраторам, так і користувачам.

Другий етап полягав у спробах вимкнути чи обійти DLP будь-яким способом, а потім розглядати, наскільки добре вона працює в розладнаному стані. Тести знову проводили і оцінювали рішення подібним чином на основі того, наскільки добре він виконувався. Для того, щоб досягти спроби нападу на DLP рішення, проводились наступні дії:

- Завантаження комп'ютера з іншого носія, подібного до компакт-диска, і таким чином здобувати доступ до жорсткого диска, також називається атакою “Live CD”.
- Пошкодження або маніпулювання важливими двійковими файлами або конфігураційними файлами, що використовуються системою DLP.
- Припинення/знищення DLP, що запобігає його запуску.

Третій і заключний етап – це запобігання комунікації між центральним сервером у DLP і кінцевою точкою. Для цього в локальному брандмауері були встановлені правила кінцевої точки для запобігання зв'язку між центральним сервером і кінцевою точкою. Тоді, як і на двох попередніх етапах, тести були запущені знову, щоб побачити, як це вплинуло на рішення DLP.

Для того, щоб добре зрозуміти, як працює поточна система DLP, було проведено дослідження трьох наступних систем DLP:

- Forcepoint;
- McAfee;
- Digital Guardian.

Таблиця 2.1 – Тест-кейси

Тест-кейс	**Конфігурація**
Передача текстового файлу	Політика встановлена для блокування текстових файлів
Передача текстового файлу із конфіденційними даними	Правила, призначені для блокування текстових файлів, видалено. Політика щодо конфіденційних даних активна
Передача стисненого файлу	Політики встановлюються для блокування стислих файлів
Передача стисненого файлу з конфіденційними даними	Політику, призначену для блокування стиснених файлів, було видалено. Політика щодо конфіденційних даних активна
Передача зашифрованого файлу	Політики встановлюються для блокування зашифрованих файлів
Передача зашифрованого файлу із конфіденційними даними	Політики, призначені для блокування зашифрованих файлів, видалено. Політика щодо конфіденційних даних активна
Передача медіа-файлу	Політика встановлена для блокування медіа-файлів
Передача зображення	Політика встановлена для блокування зображень
Запис конфіденційних даних	Введення заблокованих слів зі словника
Передача стисненого файлу з зашифрованим файлом.	Стискання зашифрованого файлу за допомогою політики, яка блокує зашифровані файли. Політику, призначену для блокування стислих файлів, було видалено.
Передача файлів Microsoft Office	1. Політика встановлена для блокування файлів .docx. 2. Додавання заблокованих типів файлів у розпаковані файли .docx. Політику, призначену для блокування файлу .docx, було видалено

Тести також включали те, як можна обробляти точки витоку, наприклад, блокувати або реєструвати.

2.2 Аналіз результатів тестування ефективності

Forcepoint

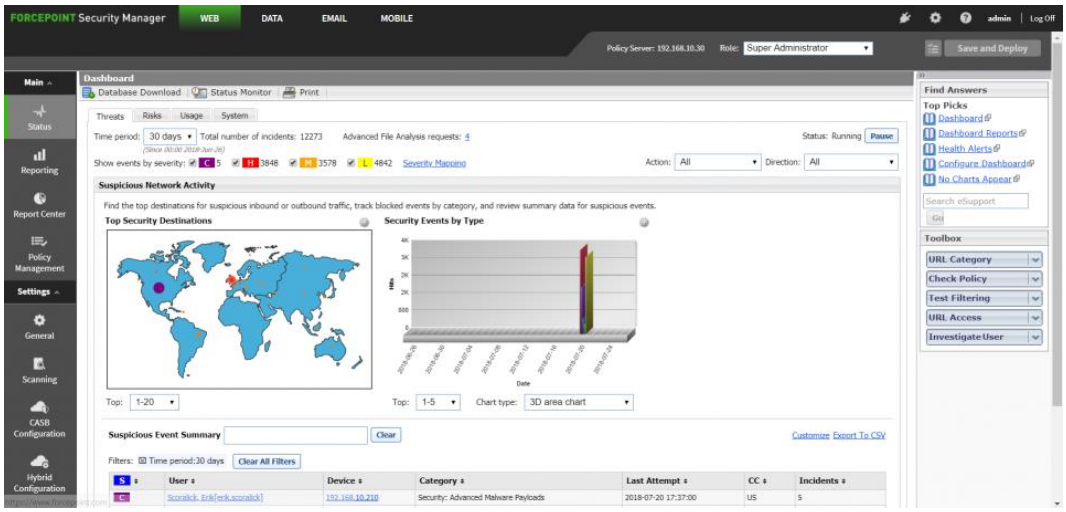

Рисунок 2.1 – Forcepoint для Windows

На рисунку 2.1 зображено інтерфейс системи Forcepoint для Windows. Параметри політики для Windows охоплюють найпоширеніші програми, які використовуються для обміну файлами або передачі, і це стосувалося програмного забезпечення, яке планувалося перевірити. Різні типи файлів для моніторингу є попередньо визначеними і охоплюють різноманітні типи файлів зображення, тексту, мультимедіа та стислих файлів. Це зручний інструмент адміністрування, де легко додати власний словник, заблокувати знімки екрану і скопіювати/вставити. Хоча, навіть якщо кнопка PrintScreen була заблокована, програмне забезпечення «Ножиці» на Windows не блокувалося.

Програми, що існують для macOS, не були настільки широко представлені, і адміністратор не може додати більше. На macOS не було проведено багато тестів, оскільки програми не підтримувалися для цієї

операційної системи, таких як Gmail і Google Drive. Типи файлів і словники контексту були однаковими, але блокувати друк екранів не вдалося.

Знайдені проблеми та рішення, запропоновані для них

Знайдені проблеми і вдосконалення спрямовані на систему для версії Windows.

Проблеми:

- Зображення було надіслано за допомогою програми Windows Live Mail, навіть якщо дія блокувалася, оскільки вона завантажує зображення до Microsoft SkyDrive, який не входить до списку програм для Windows.
- Агент DLP вимкнув кнопку Print screen, але не зміг вимкнути інструмент Ножиці для зняття скріншотів, який є попередньо встановленим на Windows.
- Агент не може розпаковувати файли та аналізувати вміст, навіть якщо він містить конфіденційні дані.
- Друк у мережевих принтерах агент не виявив. Тип файлу .xml був причиною блокування, а не самого друку.
- Агент не міг виявити, чи конфіденційний текст записувався в документі або чаті; міг лише в випадку якщо користувач намагався скопіювати текст.
- Коли з'єднання втрачається між агентом і сервером і відбуваються інциденти, журнали не надсилаються на сервер після повторного підключення.

Покращення:

- Зробити можливим додавати власний додаток для моніторингу.
- Оскільки агент аналізує MIME-тип файлів, адміністратор повинен мати можливість додавати більше типів файлів, якщо це необхідно.
- Зберігати журнали локально, але зашифровувати та надсилати на сервер, як тільки підключення буде відновлено.
- Бути більш уважним до стиснених файлів, розпаковувати архівні файли та аналізувати їх вміст.

McAfee

Знайдені проблеми та запропоновані їх рішення

Проблеми:

- Завантаження файлів .zip та .rar було заблоковано, але не файлового типу .7z.
- Зашифровані файли, що містять конфіденційні дані, не були виявлені системою DLP.
- При підключенні документа з конфіденційними даними в електронному листі єдине повідомлення, яке отримує користувач: «Щось пішло не так, не вдалося додати документ. Будь ласка спробуйте ще раз."

Покращення:

- McAfee отримав відгук про .7z і сказав, що вирішить цю проблему.
- Спростити додавання типів файлів з мови сценаріїв до додавання типів MIME.
- Подавати користувачеві більше інформації про те, який файл був заблокований і чому.

Digital Guardian

Знайдені проблеми та запропоновані їх рішення

Проблеми:

- Блокування вкладень у веб-поштових клієнтах, але не в локальних поштових клієнтах. Налаштування проксі-сервера було налаштовано, але блокування так і не відбулося.
- При надсиланні файлу .docx із зображенням, доданим у вилучену папку, в електронній пошті це було виявлено, але не виявлено при передачі його на Facebook.
- Можна блокувати лише локально підключені принтери. При друці за допомогою бездротового мережного принтера він не заблокував друк.
- Користувачеві не надсилаються сповіщення, коли щось заблоковано. Це може призвести до марного очікування, якщо користувач не знає, що ця інформація або тип файлу заблоковано.
- При застосуванні правила для блокування програми / октету, MIME-типи для зашифрованих файлів, система DLP блокувала всі веб-сторінки, які містили октет-потоки, що унеможливило тестування зашифрованих файлів як вкладень.
- Правило скріншоту змогло заблокувати програмне забезпечення «Інструмент Ножиці» у Windows, але не кнопку для екранів друку. Інші інструменти знімка не були заблоковані.

Покращення:

- Витягнути .docx файли для всіх типів передачі файлів.
- Надати інформацію користувачеві про те, що спричинило блокування.
- Розрізняти передачу файлів і перегляд веб-сторінок, коли мова йде про потоки октетів.
- Блокувати будь-які спроби скріншоту.

У таблицях 2.1-2.3 представлено резюме про те, як три різні системи DLP, Forcepoint (FP), McAfee (MA) і Digital Guardian (DG) пройшли тести:

- P – Passed, система успішно заблокувала спробу витоку.
- PC – Passed with comments, частково вдалося заблокувати або визначити спробу витоку.
- F – Failed, система не блокувала спроби витоку.
- C – Comment, тестовий випадок, який з будь-якої причини не може бути проведений.
- N/A – Not able to test

Етап 1: Тестування одразу після установки

Першим етапом наших експериментів було встановлення програмного забезпечення на кінцеві точки, налаштування політик відповідно до документів і посібників, початок передачі наших тестових файлів. На цьому етапі програмне забезпечення не втручалося. В рисунку 4.1 показані наші результати, як ці системи DLP впоралися з нашими тестами.

Таблиця 2.2 – Результати тестування після установки

Опис	Mail			Printer			Google Drive		
	FP	MA	DG	FP	MA	DG	FP	MA	DG
Прикріпити/відправити текстовий файл	P	N/A	PC	P	N/A	F	PC	N/A	F
Прикріпити/відправити текстовий файл з конфіденційними даними	P	P	PC	P	N/A	PC	P	C	F
Прикріпити/відправити стиснутий файл	PC	N/A	PC	N/A	N/A	N/A	PC	N/A	F
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	PC	PC	N/A	N/A	N/A	F	C	F
Прикріпити/відправити зашифрований файл	P	N/A	F	N/A	N/A	N/A	PC	N/A	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	F	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити медіа-файл	P	N/A	PC	N/A	N/A	N/A	PC	N/A	F
Прикріпити/відправити зображення	PC	N/A	PC	PC	N/A	F	PC	N/A	F
Написати конфіденційні дані	F	P	PC	N/A	N/A	N/A	F	C	F
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	F	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити файли Microsoft Office	P	N/A	PC	P	N/A	F	PC	N/A	F

Продовження таблиці 2.2

Опис	Facebook			Skype			Teamviewer			USB		
	FP	MA	DG	FP	MA	DG	FP	MA	DG	FP	MA	DG
Прикріпити/відправити текстовий файл	P	N/A	P	P	N/A	C	P	N/A	C	P	N/A	P
Прикріпити/відправити текстовий файл з конфіденційними даними	P	C	P	P	C	C	P	C	C	P	N/A	P
Прикріпити/відправити стиснутий файл	P	N/A	P	P	N/A	C	P	N/A	C	P	N/A	P
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	C	P	F	C	C	F	C	C	F	N/A	P
Прикріпити/відправити зашифрований файл	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	C	F	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити медіа-файл	P	N/A	P	P	N/A	C	P	N/A	C	P	N/A	P
Прикріпити/відправити зображення	P	N/A	P	P	N/A	C	P	N/A	C	P	N/A	P
Написати конфіденційні дані	F	C	P	F	C	C	N/A	N/A	N/A	N/A	N/A	N/A
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	C	F	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити файли Microsoft Office	P	N/A	PC	P	N/A	C	P	N/A	C	P	N/A	P

Результати:

Таблиця 2.3 – Підсумовування результатів з таблиці 2.2

	Forcepoint	McAfee	Digital Guardian
Pass	37	2	14
Pass with comments	8	1	10
Failed	22	2	23
Comments	0	19	21
N/A	9	53	9
Всього	77	77	77

Comments:

- McAfee – Через проблеми з SSL/HTTPS не вийшло їх перевірити. Крім того, вони не охоплюють кінцеві точки, тому N/A на принтерах і USB.
- Digital Guardian – Не в змозі змусити Skype і Teamviewer пройти через проксі.

Етап 2: Тестування при вимкненому агенті

На другому етапі здійснена спроба знайти способи відключити агент DLP на кінцевих точках. Таблиця 2.2 представляє результати цього експерименту.

Таблиця 2.4 – Результати тестування після вимкнення агенту

Опис	Mail			Printer			Google Drive		
	FP	MA	DG	FP	MA	DG	FP	MA	DG
Прикріпити/відправити текстовий файл	F	N/A	PC	F	N/A	F	F	N/A	F
Прикріпити/відправити текстовий файл з конфіденційними даними	F	C	PC	F	N/A	F	F	C	F
Прикріпити/відправити стиснутий файл	F	N/A	PC	N/A	N/A	N/A	F	N/A	F
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	C	PC	N/A	N/A	N/A	F	C	F
Прикріпити/відправити зашифрований файл	F	N/A	F	N/A	N/A	N/A	F	N/A	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	C	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити медіа-файл	F	N/A	PC	N/A	N/A	N/A	F	N/A	F
Прикріпити/відправити зображення	F	N/A	PC	F	N/A	F	F	N/A	F
Написати конфіденційні дані	F	C	PC	N/A	N/A	N/A	F	C	F
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	C	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити файли Microsoft Office	F	N/A	PC	F	N/A	F	F	N/A	F

Продовження таблиці 2.4

Опис	Facebook			Skype			Teamviewer			USB		
	FP	MA	DG	FP	MA	DG	FP	MA	DG	FP	MA	DG
Прикріпити/відправити текстовий файл	F	N/A	P	F	N/A	C	F	N/A	C	F	N/A	F
Прикріпити/відправити текстовий файл з конфіденційними даними	F	C	P	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити стиснутий файл	F	N/A	P	F	N/A	C	F	N/A	C	F	N/A	F
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	C	P	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити зашифрований файл	F	N/A	F	F	N/A	C	F	N/A	C	F	N/A	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	C	F	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити медіа-файл	F	N/A	P	F	N/A	C	F	N/A	C	F	N/A	F
Прикріпити/відправити зображення	F	N/A	P	F	N/A	C	F	N/A	C	F	N/A	F
Написати конфіденційні дані	F	C	P	F	C	C	N/A	N/A	N/A	N/A	N/A	N/A
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	C	F	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити файли Microsoft Office	F	N/A	PC	F	N/A	C	F	N/A	C	F	N/A	F

Результати:

Таблиця 2.5 – Підсумовування результатів з таблиці 2.4

	Forcepoint	McAfee	Digital Guardian
Pass	0	0	7
Pass with comments	0	0	9
Failed	68	0	31
Comments	0	24	21
N/A	9	53	9
Всього	77	77	77

Comments:

- Digital Guardian – Не в змозі змусити Skype і Teamviewer пройти через проксі.

Етап 3: Тестування при відключеному сервері

На третьому і заключному етапі спроба запобігти зв’язку між центральним сервером DLP і кінцевою точкою. Для результатів цієї фази див. табл. 2.3.

Таблиця 2.6 – Результати тестування при відключеному сервері

Опис	Mail			Printer			Google Drive		
	FP	MA	DG	FP	MA	DG	FP	MA	DG
Прикріпити/відправити текстовий файл	P	N/A	F	F	N/A	F	PC	N/A	F
Прикріпити/відправити текстовий файл з конфіденційними даними	P	C	F	F	N/A	PC	P	C	F
Прикріпити/відправити стиснутий файл	PC	N/A	F	N/A	N/A	N/A	PC	N/A	F
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	C	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити зашифрований файл	P	N/A	F	N/A	N/A	N/A	PC	N/A	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	C	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити медіа-файл	P	N/A	F	N/A	N/A	N/A	PC	N/A	F
Прикріпити/відправити зображення	PC	N/A	F	F	N/A	F	PC	N/A	F
Написати конфіденційні дані	F	C	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	C	F	N/A	N/A	N/A	F	C	F
Прикріпити/відправити файли Microsoft Office	P	N/A	F	F	N/A	F	PC	N/A	F

Продовження таблиці 2.6

Опис	Facebook			Skype			Teamviewer			USB		
	FP	MA	DG	FP	MA	DG	FP	MA	DG	FP	MA	DG
Прикріпити/відправити текстовий файл	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	P
Прикріпити/відправити текстовий файл з конфіденційними даними	P	C	F	P	C	C	P	C	C	P	N/A	P
Прикріпити/відправити стиснутий файл	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	P
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	C	F	F	C	C	F	C	C	F	N/A	P
Прикріпити/відправити зашифрований файл	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	C	F	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити медіа-файл	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	P
Прикріпити/відправити зображення	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	P
Написати конфіденційні дані	F	C	F	F	C	C	N/A	N/A	N/A	N/A	N/A	N/A
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	C	F	F	C	C	F	C	C	F	N/A	F
Прикріпити/відправити файли Microsoft Office	P	N/A	F	P	N/A	C	P	N/A	C	P	N/A	P

Результати:

Таблиця 2.7- Підсумування результатів з таблиці 2.6

	Forcepoint	McAfee	Digital Guardian
Pass	34	0	7
Pass with comments	8	0	1
Failed	26	0	39
Comments	0	24	21
N/A	9	53	9
Всього	77	77	77

Comments:

- McAfee - Віртуальне середовище зупинилося, коли було додано та активовано правило брандмауера. Також через проблеми з SSL/HTTPS не вийшло їх перевірити. Крім того, вони не охоплюють кінцеві точки, тому N/A на принтерах і USB.
- Digital Guardian – Не в змозі змусити Skype і Teamviewer пройти через проксі.

2.3 Загальний опис методів підвищення ефективності DLP систем

В системах DLP існує багато мілких шпаринок, через які може витекти інформація. Для забезпечення безпеки інформації в компанії не вистачить лише правильно впровадити систему та регулярно її моніторити. Для покращення роботи системи можна використовувати такі методи як:

- обмеження прав та повноважень користувачів до необхідних безпосередньо для їх роботи;
- контроль використання знімних носіїв, використання корпоративних знімних носіїв;
- використання чорних і білих списків;
- відключення можливості вибору режимів завантаження ОС;
- контроль застосунків і форматів документів, які не прийняті в організації;
- підвищена увага до стиснутих файлів, архівів, так як система не в змозі переглядати їх вміст;
- використання електронного підпису в документах.

2.4 Порівняльний аналіз та вибір оптимального методу

У системі достатньо шпаринок, які необхідно окремо розглядати і впливати на них. Усі вищеперераховані методи ефективні і зменшують шанси на витік. Але це все одно не забезпечує цілковитої безпеки для секретної інформації. Тому у цій дипломній роботі розглядається трасування системних викликів (System Call Tracing). Таким чином перехоплюються виклики функцій в операційній системі в програмі і перенаправляються на іншу функцію. Метод API hooking – це метод, за допомогою якого можна змінювати поведінку і потік викликів API. API підключення може бути зроблено з використанням різних методів у Windows. Метод включає в себе точку розриву пам'яті і введення інструкцій. Підключення може використовуватися для аналізу викликів у додатку Windows або може бути використано для захоплення деякої інформації, пов'язаної з API-викликами.

Висновки до розділу 2

В даному розділі було проведене тестування DLP систем. Згідно з результатами жодна з них ідеально не захищає інформацію в середині мережі.

Тому були розглянуті рішення, які покращують показники роботи системи. Для цього необхідно правильно впровадити систему, регулярно моніторити результати, не надавати користувачам більше прав, ніж їм потрібно для роботи, уважно слідкувати за файлами, формати яких не прийняті в організації, слідкувати за стиснутими документами, блокувати програмні засоби, які роблять знімки екранів, регулярно поповнювати словник системи і таке інше.

Ці методи ефективні, але жоден з них окремо від інших не закриває повністю весь обсяг вразливостей. Виходячи з цього було запропоновано трасування системних викликів за допомогою API hooking. Метод API hooking – це метод, за допомогою якого можна змінювати поведінку і потік викликів API.

3 ВПРОВАДЖЕННЯ МЕТОДУ API HOOKING

У цьому розділі буде розглядатися метод API hooking та процедура його впровадження. Буде здійснено опис процесу роботи методу. Буде проаналізовано переваги і недоліки методу.

3.1 API Hooking

Необхідно мати можливість ідентифікувати та класифікувати дані відповідно до набору правил/політик і мати можливість адаптувати та застосовувати ці правила відповідно до різних випадків. Також повинно відбуватися реєстрування дій, виконаних для кожного файлу, і надавати користувачеві пояснення щодо того, чому ця дія була зроблена. Використовуючи простий спосіб визначення правил за допомогою чорних і білих списків, файлів та файлів, що містять попередньо визначений підпис, дозволять можливу інтеграцію з рішеннями, подібними до тих, які було оцінено в розділі 2. Ідея полягає в тому, що існуючі рішення можуть надавати інформацію про те, які файли слід вважати конфіденційними.

Для того, щоб програми могли отримувати доступ до даних, що зберігаються на такому носії, як жорсткий диск, CD/DVD диск або знімні носії, програми запитують дані через операційну систему. Це також стосується зберігання або передачі даних між носіями, наприклад, надсилання файлів по мережі або копіювання файлів з жорсткого диска на диск USB. Розміщуючи інспекційний шар між додатком і операційною системою, можна керувати тим, хто і що отримає доступ до яких даних.

Одним із способів досягнення цього є техніка, яка перехоплює виклики функцій в операційній системі в програмі і перенаправляє її на іншу функцію. Ця методика відома як прив'язка інтерфейсу програмування додатків, підключення API, також зване трасування системних викликів

(System Call Tracing) або взаємодія системних викликів (System Call Intercepion).

Підключення API є досить вимогливим і ризикованим, якщо не виконати його правильно. Після того, як API підключений, це код, який повинен керувати всіма викликами функцій цього API, будь-які помилки можуть призвести до збоїв додатків або навіть до збою системи, наприклад BSOD, Blue Screen Of Death. Microsoft Research розробила структуру для Windows, щоб зробити підключення API більш легким і безпечним. Ця структура відома як Detours [10] і доступна у двох виданнях, експрес та професійний. Існують обмеження – у першому виданні можна, наприклад, обмежити підтримку процесорів. Перше видання працює тільки для архітектури x86. Тут було використано аналог з відкритим вихідним кодом для Detours, а саме фреймворк EasyHook, який спирається на концепцію Detours, продовжуючи вдосконалювати її.

Визначивши і підключивши Windows API, пов'язане з керуванням файлами [12] за допомогою фреймворку EasyHook, були реалізовані наступні функції:

- CopyFile
- CreateFile
- DeleteFile
- MoveFile
- ReadFile
- ReplaceFile
- WriteFile

Заздалегідь було визначено список файлів і каталогів, які слід заблокувати, і які повинні бути дозволені. Також було визначено список Magic Numbers і підписів, які повинні бути заблоковані. Вдалося запобігти доступу до файлів, які були заблоковані, що не розміщені у списку білих

списків або містять заблокований магічний номер або підпис через будь-який з визначених API.

Нижче наведено псевдокод, що описує процес перевірки підключеного API, коли він викликається:

Algorithm 1 Inspection process

```
1:  procedure FILEINSPECTION
2:      filetype ← type of requested file
3:      filepath ← path to requested file
4:      for all entries in blocked file types list do
5:          if identitfied filetype = entry then
6:              blockAPIcall
7:      for all entries in blocked directories and file list do
8:          if requested filepath begins with entry then
9:              blockAPIcall
10:     for all entries in allowed directories and file list do
11:         if requested filepath = entry then
12:             allowAPIcall
13:     blockAPIcall
```

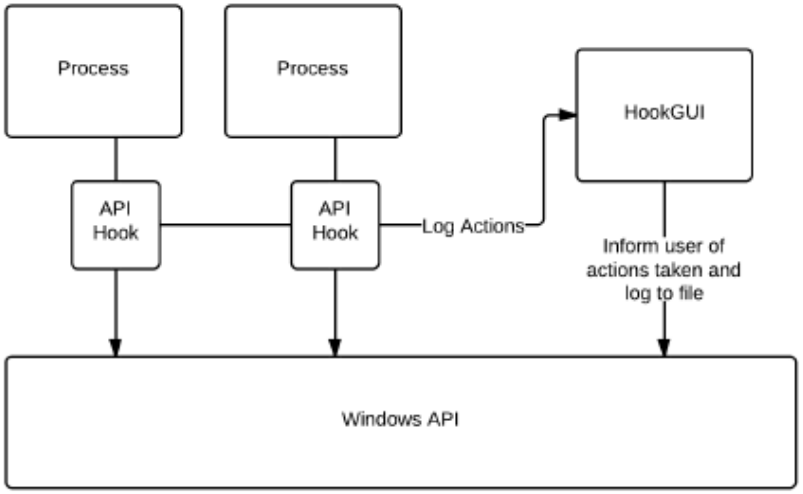

Рисунок 3.1 – Ілюстрація реалізованого рішення

3.2 Види API Hooking

Перехоплення перед запуском

Виконується за допомогою фізичного модуля (зазвичай це файли типу .exe або .dll), під час виконання функції, яку ми прагнемо перехопити та

перенаправити. Зазвичай є три можливості, які можна використати для досягнення бажаного результату.

Перша — знайти точку входу до цільової функції, та змінити чи переписати код. Цей спосіб обмежений розмірами самої функції. Проте в цьому випадку маємо змогу завантажити певні модулі динамічно, використовуючи API LoadLibrary. Функція Kernel (kernel32.dll) може бути використана майже в усіх випадках, оскільки кожен процес у системах Windows має власну копію даного модуля. Перевага також є в тому, коли ми знаємо, на якій операційній системі ми будемо змінювати модуль. В цьому випадку ми можемо використати прямі вказівники для API LoadLibrary. Використання таких можливе, оскільки адреса kernel модуля в пам’яті є статичною. Також можна використати відслідковування поведінки завантажених динамічних модулів. В цьому випадку частина ініціалізації запускається одразу, після завантаження пам’яті. Також не встановлені обмеження на ініціалізовані частини в нових модулях.

Друга можливість полягає в тому, що для заміни чи перенаправлення функції в модулі використовуємо розширення. Треба вибрати перші 5 байтів, та замінити їх, або переписати IAT. В першому випадку — відбудеться перенаправлення коду, що виконується на наш код. Коли викликається функція, котра змінію запис IAT, наш код буде виконано одразу після цього виклику. Проте, розширення модуля не є простим, оскільки ми повинні приділяти увагу DLL.

Третім способом є повна заміна цілого модуля. Це означає, що необхідно створити повністю свою версію модуля, який матиме змогу завантажити оригінальний та викликати оригінальні функції, в яких вже не буде потреби. Проте нові функції, які були додані з новою версією модуля, будуть повністю новими. Даний метод є не дуже ефективним для великих модулей, оскільки потребує багато часу та ресурсів для експорту.

Перехоплення під час роботи

Перехоплення перед запуском в основному є спеціалізованим та є дуже конкретизованим та націленим на конкретні програми чи модулі. Якщо замінити функцію в kernel32.dll, або в ntdll.dll, то ми отримаємо чудову можливість замінити дану функцію у всіх процесах, котрі будуть запущені пізніше. Проте такі процедури є дуже складними, оскільки потрібно відслідковувати точність та вдосконалення коду нової функції та цілого модуля. Проте головна проблема полягає в тому, що як той процес, який буде запущено в майбутньому буде перехоплено (тобто для всіх процесів необхідно буде перезапустити систему). Інша проблема полягає в складності отримання доступу до файлів системи. Набагато кращим виглядає рішення перехоплення процесів під час їх перебігу. Цей метод потребує більшої кількості інформації та знань, проте результат є найкращим. Перехоплення під час перебігу процесу можливе тільки для процесу, до якого ми маємо параметр доступу “writing” до пам’яті. Для запису можна використати функцію API WriteProcessMemory.

3.3 Потенційні випадки використання API Hooking

- Можливість контролю вхідного http-трафіку й “підміна” небажаного контенту на той, котрий ми приймаємо.
- Можливість виконання логування інформацію у випадку копіювання будь-яких файлів з підконтрольованої папки.
- Можливість виконання певних модифікацій коду в проекті, початковий код якого було певним чином втрачено.

3.4 Методи встановлення API Hooking

Різні методи встановлення такого прийому зумовлені різними підходами та використанням різних технологій та застосунків.

- Одним з таких методів полягає у використанні функції SetWindowsHookEx. Використання даного методу є доволі простим, проте несе в собі певні обмеження в роботі. Метод дозволяє перехоплювати лише конкретні функції, зазвичай пов'язані з вікном виконання програми (наприклад, перехоплення певних подій, які отримують доступ до вікна виконання, натискання комп'ютерної миші, або клавіатурного введення). Очевидною перевагою такого методу є потенціальна можливість встановлення та використання загального перехоплення (наприклад, є можливість одразу, на всіх програмах перехопити введення з клавіатури).
- Використання заміни адрес у відділі імпорту DLL. Даний метод полягає в тому, зо довільний модуль має відділ імпорту. В останньому перераховані все інші модулі, котрі використовуються в першому. Також, там зберігаються адреси пам’яті для функцій, які експортуються даним модулем. Виконання методу являє собою підміну адреси в нашому модулі на той, котрий є вигідним нам. Після виконання даних дій, керування буде передано на задану адресу.
- Третій метод полягає у застосуванні ключа реєстру HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_Dlls. Необхідно прописати в реєстрі шлях до DLL, але виконати таку дію має право лише користувач з правами адміністратора. Такий метод є простим та ефективним, якщо програма не використовує kernel32.dll (тобто не має змоги викликати функцію LoadLibrary).

Метод ін’єктування

Даний метод є можливим, оскільки функція ThreadStart, яка в свою чергу передається функції CreateThread, має схожу сигнатуру з функцією LoadLibrary. Це дає нам змогу вказати метод LoadLibrary в якості аргументу при створенні потоку.

Нижче наведено алгоритм використання ін’єктування DLL:

- Виконується пошук адруси функції LoadLibrary з Kernel32.dll для потоку, в який ми поставили за ціль інжектувати DLL.
- Необхідно виділити пам’ять для запису аргументів даної функції.
- Виконується створення потоку. В якості функції ThreadStart вказуємо функцію LoadLibrary та її аргумент.
- Відбувається виконання потока, який в свою чергу завантажує бібліотеку й завершується.
- Цільова бібліотека є ін’єктваною в адресний простір стороннього потоку. При цьому, при завантаженні DLL, буде викликано метод DllMain з флагом PROCESS_ATTACH. Останнє, якраз і є тим місцем, де ми отримали можливість встановлювати перехоплення на необхідні нам функції.

Далі детальніше опишемо процес встановлення перехоплення:

- Необхідно знайти адресу функції, виклик якої ми плануємо перехопити. Для цього можна використати MessageBox в user32.dll.
- Зберігажмо декілька перших байтів даної функції на іншій ділянці пам’яті.
- Замість збережених нами байтів функції вставляємо машинну команду JUMP для переходу до адреси підмінної функції. Для коректного виконання необхідно стежити за тим, щоб сигнатура функції була такою самою, як сигнатура вихідної функції. Тобто,

всі параметри, а також значення які повертаються й правила виклику повинні чітко співпадати з вихідною функцією.

- Останнім кроком відбувається наступне. Під час виклику потоком функції, яку ми плануємо перехопити, команда JUMP, яку було попередньо вставлено на місті перших байтів функції, перенаправить потік до підмінної нами функції. Наданому етапі ми отримаємо змогу виконання довільного коду.

Перехоплення Win32 API викликів завжди було не простим питанням та завданням. Термін hooking (перехоплення) являє собою фундаментальний метод отримання контролю над певними фрагментами, або фрагментом виконання коду. Перехоплення забезпечує простий механізм, який може змінити поведінку операційної системи, а також поведінку програм, які виконуються, не маючи при цьому доступу до вихідного коду.

Багато сучасних систем привертаються увагу до своєї спроможності використовувати існуючі програми Windows, використовуючи певні методи “шпигунства”. Ключовою ціллю та завданням перехоплення є не тільки сприяння більш розширеним функціональним можливостям, але й введення інверсійного коду для досягнення певних поставленних цілей.

Теперішні, новітні операційні системи забезпечують складні механізми задля розподілення адресного простору кожного процесу. Такі архітектури забезпечують захист пам’яті, тому жодна за програм не має змогу та нагоди пошкодити або змінити адресний простір іншого процесу.

Наведемо деякі переваги спостереження та перехоплення над програмами:

- **Моніторинг функції API**

  Можливість контролювати виклики функції API є надзвичайно корисною й дозволяє відслідковувати приховані дії, які можуть відбуватися під час виклику API. За рахунок цього відбувається всебічна перевірка параметрів функцій, а також повідомляє про проблеми, які зазвичай можуть залишитися поза увагою. Інколи, може

бути корисною можливість контролю функцій API, які пов’язані з пам’яттю, задля вилучення витоків ресурсів.

- **Налагодження та reverse engineering**

  Окрім стандартних методів налагодження API, перехоплення має репутацію як один з найпопулярніших механізмів налагодження. Багато розробників використовують техніку перехоплення та моніторингу API для ідентифікації різних компонентів та їх взаємозв'язків. Перехоплення API є дуже потужним способом отримання інформації про двійковий виконуваний файл.

- **Піринг всередині операційної системи**

  Часто розробники прагнуть знати операційну систему та її будову та архітектуру на доволі глибокому рівні. Перехоплення та моніторинг також є досить корисним методом для декодування недокументованих або погано документованих API.

- **Розширення пропонованих функціональних можливостей**

  Виконується шляхом вбудовування користувальницьких модулів у зовнішні програми Windows. Переустановка нормального коду для виконання за допомогою ін'єкційних гачків може забезпечити простий спосіб змінити та розширити існуючі функціональні можливості модуля. Наприклад, багато продуктів іноді не відповідають певним вимогам безпеки, і їх потрібно коригувати відповідно до ваших конкретних потреб. Перехоплення та моніторинг додатків дозволяє розробникам додавати складну попередню і пост-обробку навколо початкових функцій API. Ця здатність є надзвичайно корисною для зміни поведінки вже скомпільованого коду.

3.5 Функціональні вимоги до системи перехоплення та моніторингу

Існує кілька важливих рішень, які необхідно зробити, перш ніж приступити до впровадження будь-якої системи API перехоплення. Перш за

все, слід визначити, чи потрібно виконувати перехоплення та моніторинг однієї програми або необхідне використання системного движка. Наприклад, якщо ви хочете відстежувати лише одну програму, вам не потрібно встановлювати систему загальносистемного перехоплення та моніторингу, але якщо ваша робота полягає у відстеженні всіх викликів до TerminateProcess або WriteProcessMemory, єдиним способом зробити це - це мати системний засіб перехоплення та моніторингу.

3.6 Загальний вигляд структури засобу API перехоплення та моніторингу

Зазвичай система перехоплення складається принаймні з двох частин – сервер перехоплення тадрайверу. Сервер перехоплення відповідає за введення драйвера в цільові процеси у відповідний момент. Він також керує драйвером і додатково може отримувати інформацію від драйверу про свою діяльність, тоді як модуль драйвера, який виконує фактичне перехоплення. Ця конструкція є грубим утотожненням і поза сумнівом не охоплює всі можливі реалізації. Проте вона окреслює межі системи перехоплення та моніторингу.

Після того, як ви маєте специфікацію вимоги для системи перехоплення та моніторингу, існує кілька моментів, які необхідно враховувати:

- Які програми потрібно підключити?
- Як вводити DLL в цільові процеси або які методи імплантації дотримуватися?
- Який механізм перехоплення використовувати?

3.7 Обмеження і проблеми

З EasyHook можна зіткнутися з деякими обмеженнями в залежності від того, як він використовується, наприклад, програми, що потребують встановлення .NET 3.5 або 4.0, щоб мати змогу працювати. Проте він пропонує простий і безпечний спосіб підключення окремих процесів, що перенаправляють виклики API на ваш вибір. Можна масово підключити всі процеси, однак це може включати перезапис/заміну реального API. Це дозволить ефективно керувати всіма процесами за допомогою API, але будь-які проблеми, викликані переписаним API, вплинуть на всю систему. При підключенні кожного процесу окремо потрібно вирішувати будь-яку проблему, що виникає в перехідному API, впливаючи на процес підключення. Нижче наведений список інших проблем, з якими трапилося зіткнутися під час впровадження рішення:

- Бібліотек API, доступних в Windows, багато, і хоча багато API не мають доступу до файлів, не вдалося підключити ті, що пов'язані з доступом до файлів, що дозволить повністю охопити агент.
- Сам агент вразливий до підключення API від будь-яких привілейованих користувачів. Необхідно вжити заходів для забезпечення того, щоб агент не був підроблений, щоб забезпечити його безпеку. Це може бути зроблено за допомогою прихованого процесу, який спостерігає за статусом агента, як сторожовий таймер, і вживає таких дій, як перезавантаження системи, якщо агент з якоїсь причини відключений.
- Рішення не пропонує жодного захисту від живих атак CD.
- Оскільки кожен процес підключається окремо, рішення повинно забезпечити підключення кожного процесу миттєво, коли вони будуть створені. Якщо ні, то існуватиме проміжок часу, коли процес матиме можливість вільного доступу до даних, доки він не буде підключений.

- Створене рішення призначене лише для того, щоб розглядати API-підключення як середнє значення для DLP і не зосереджується на продуктивності. Варто зазначити, що кількість файлів і каталогів, перелічених для білого та чорного списку, безпосередньо вплине на продуктивність підключених процесів.
- Під час тестів iexplore.exe і chrome.exe все ще могли отримати доступ до всіх файлів, коли користувач намагався, серед іншого, завантажити файл на веб-сервер. Є припущення, що це пов'язано з використанням API, окрім тих, які були визначені.

Висновки до розділу 3

У цьому розділі описується процедура впровадження API hooking. Наведена ілюстрація (Рис. 3.1), яка доступно описує як працює цей метод. Метод був обраний завдяки його ефективності. Впровадження API hooking значно спрощує процедуру моніторингу за інцидентами.

Під час впровадження виникали труднощі, підключення є доволі вибагливим. У розділі детально описані обмеження, з якими довелося зіткнутися.

4 РЕЗУЛЬТАТИ ТЕСТУВАННЯ DLP ПІСЛЯ ЗАСТОСУВАННЯ ЗАПРОПОНОВАНИХ РІШЕНЬ

Самі по собі ці системи, які розглядаються, не працюють так добре, як очікувалося, однак комбінація знайдених рішень може спільно давати набагато кращі результати. Таблиці 4.1-4.4 демонструють гіпотетичні припущення та результати того, як комбінація знайденого рішення з Forcepoint та проксі з Digital Guardian буде виконуватися в цих експериментах. Спільна робота двох систем буде найбільш ефективним рішенням DLP. Тим не менш, результати не ідеальні, не всі точки витоку повністю покриті, але зона дії системи розширюється, а кількість невдач зменшується. Ці результати були розроблені з припущень, якщо один з рішень DLP проходить, а іншій не вдається, перший відповідає за цю точку витоку. Результати, представлені в цьому розділі, не приймають до уваги оцінки N/A. Проведено аналіз трьох фаз з відповідним результатом і порівняно з оригінальними результатами експериментів.

4.1 Етап 1: Тестування одразу після установки

У порівнянні з етапом 1 до вдосконалення, зараз отримано коефіцієнт пропуску 65%, де до цього було отримано 54%, використовуючи лише Forcepoint. Частота помилки Forcepoint спочатку становила 32%, а для Digital Guardian - 34%. З цією комбінацією отримано 15% відсоткової ставки. Таблиця 3.1 показує ці результати.

Таблиця 4.1 – Результат поєднання Forcepoint та Digital Guardian

Опис	Mail	Printer	Google Drive	Facebook	Skype	Teamviewer	USB
	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG
Прикріпити/відправити текстовий файл	P	P	PC	P	P	P	P
Прикріпити/відправити текстовий файл з конфіденційними даними	P	P	P	P	P	P	P
Прикріпити/відправити стиснутий файл	PC	N/A	PC	P	P	P	P
Прикріпити/відправити стиснутий файл з конфіденційними даними	PC	N/A	F	P	C	C	P
Прикріпити/відправити зашифрований файл	P	N/A	PC	P	P	P	P
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	N/A	F	F	C	C	F
Прикріпити/відправити медіа-файл	P	N/A	PC	P	P	P	P
Прикріпити/відправити зображення	PC	PC	PC	P	P	P	P
Написати конфіденційні дані	PC	N/A	F	P	C	N/A	N/A
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	N/A	F	F	C	C	F
Прикріпити/відправити файли Microsoft Office	P	P	PC	P	P	P	P

4.2 Етап 2: Тестування при вимкненому агенті

Оскільки в початковому етапі 2 було доведено, що Digital Guardian охоплює більшість точок витоку, цей експеримент дасть ті ж результати, швидкість проходження 10%. Forcepoint не пройшов 100% тестів у початковому етапі 1, при цьому ця швидкість не була б 46%. Таблиця 3.2 демонструє, як виглядатимуть ці результати.

Таблиця 4.2 – Результат поєднання Forcepoint та Digital Guardian з вимкненим агентом

Опис	Mail	Printer	Google Drive	Facebook	Skype	Teamviewer	USB
	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG
Прикріпити/відправити текстовий файл	PC	F	F	P	C	C	F
Прикріпити/відправити текстовий файл з конфіденційними даними	PC	F	F	P	C	C	F
Прикріпити/відправити стиснутий файл	PC	N/A	F	P	C	C	F
Прикріпити/відправити стиснутий файл з конфіденційними даними	PC	N/A	F	P	C	C	F
Прикріпити/відправити зашифрований файл	F	N/A	F	F	C	C	F
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	N/A	F	F	C	C	F
Прикріпити/відправити медіа-файл	PC	N/A	F	P	C	C	F
Прикріпити/відправити зображення	PC	F	F	P	C	C	F
Написати конфіденційні дані	PC	N/A	F	P	C	N/A	N/A
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	N/A	F	F	C	C	F
Прикріпити/відправити файли Microsoft Office	PC	F	F	PC	C	C	F

4.3 Етап 3: Тестування при відключеному сервері

Так само, як і на першому етапі, Forcepoint мав найкращі показники на третьому етапі серед інших систем. З цією комбінацією він дасть трохи кращий результат від початкового, швидкість проходження 51% порівняно з 50%. Найбільшою різницею тут була невдача Digital Guardian, з 57% до 25%. Це не бездоганно, але це крок у правильному напрямку.

Таблиця 4.3 – Результат поєднання Forcepoint та Digital Guardian при відключеному сервері

Опис	Mail	Printer	Google Drive	Facebook	Skype	Teamviewer	USB
	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG	FP+DG
Прикріпити/відправити текстовий файл	P	F	PC	P	P	P	P
Прикріпити/відправити текстовий файл з конфіденційними даними	P	PC	P	P	P	P	P
Прикріпити/відправити стиснутий файл	PC	N/A	PC	P	P	P	P
Прикріпити/відправити стиснутий файл з конфіденційними даними	F	N/A	F	F	C	C	P
Прикріпити/відправити зашифрований файл	P	N/A	PC	P	P	P	P
Прикріпити/відправити зашифрований файл з конфіденційними даними	F	N/A	F	F	C	C	F
Прикріпити/відправити медіа-файл	P	N/A	PC	P	P	P	P
Прикріпити/відправити зображення	PC	F	PC	P	P	P	P
Написати конфіденційні дані	F	N/A	F	F	C	N/A	N/A
Прикріпити/відправити стиснутий файл з зашифрованим файлом всередині	F	N/A	F	F	C	C	F
Прикріпити/відправити файли Microsoft Office	P	F	PC	P	P	P	P

Таблиця 4.4 – Результат поєднання Forcepoint та Digital Guardian

	Етап 1	Етап 2	Етап 3
Pass	44	7	35
Pass with comments	7	9	9
Failed	10	31	17
Comments	7	21	7
N/A	9	9	9
Всього	77	77	77

Висновки до розділу 4

У цьому розділі відбувається реалізація трасування системних викликів. Проводиться тестування після впровадження API hooking. У таблицях 4.1-4.4 наведені результати тестування, які показали, що результати роботи системи значно покращилися, а це означає що метод є дієвим. Код програми наведений у додатках.

ВИСНОВКИ

У цій роботі була проведена оцінка трьох існуючих системи DLP. Визначено їх слабкості та недоліки. Спираючись на результати тестування, можна зробити висновок, що жодне рішення DLP не є на 100% безпечним, оскільки всі рішення показали потенційно слабкі місця.

У аналізі видно збільшення охоплення, коли було зроблено об'єднання двох систем DLP у гібридне рішення. Якщо агент DLP на пристрої не працює, DLP-сервер системи все ще контролюватиме мережний трафік. Якщо ж пристрій буде переміщено за межі досяжності сервера, агент залишатиметься захищати дані в найближчих межах доступу.

Також було розроблено рішення для DLP. Його можна інтегрувати з іншими існуючими рішеннями DLP. Використання чорного і білого списку для каталогів, файлів, підписів і магічних чисел, що надаються іншими рішеннями DLP, дозволяє обмежувати доступ до файлів через виклики API управління файлами у Windows. Також було перераховано перелік виявлених обмежень і проблем під час впровадження цього рішення.

Результати показали, що існують різні загрози для самих агентів DLP і що необхідно вжити декількох заходів для забезпечення безпеки агентів, таких як обмеження привілеїв для користувача, використання повного шифрування диску та використання засобів для контролю стану самого процесу. Також правильна ідентифікація та категоризація даних, особливо при прихованих або заплутаних, не вдаючись до надмірно складних визначень при створенні політики, є важливим моментом. Через це трапляється велика кількість інцидентів.

ПЕРЕЛІК ДЖЕРЕЛ

1. H. Balinsky, D. S. Perez, and S. J. Simske. System call interception framework for data leak prevention. pages 139–148. IEEE, 2011.
2. Kevin Benton and Ty Bross. Timing analysis of ssl/tls man in the middle attacks. CoRR, abs/1308.3559, 2013.
3. Jorge Blasco, Julio Cesar Hernandez-Castro, Juan E Tapiador, and Arturo Ribagorda. Bypassing information leakage protection with trusted applications. computers & security, 31(4):557–568, 2012.
4. Vitor R Carvalho and William W Cohen. Preventing information leaks in email. In SDM. SIAM, 2007.
5. EasyHook. Easyhook. Режим доступу: http://easyhook.codeplex.com/ - 3.06.2019
6. Prathaben Kanagasingham. Data loss prevention. SANS Institute InfoSec Reading Room. Режим доступу: http://www. sans. org/reading_room/dlp/data lossprevention_ 32883, 2008.
7. Jinhyung Kim and Hyung Jong Kim. Design of internal information leakage detection system considering the privacy violation. In Information and Communication Technology Convergence (ICTC), 2010 International Conference on, pages 480–481. IEEE, 2010.
8. Matthias Luft. Can data leakage prevention prevent data leakage? 2009.
9. M. F. Marhusin, H. Larkin, C. Lokan, and D. Cornforth. An evaluation of api calls hooking performance. volume 2; 1, pages 315–319. IEEE, 2008.
10. Microsoft. Detours. Режим доступу: http://research.microsoft.com/en-us/projects/detours/ - 3.06.2019 р.
11. Rich Mogull and LLC Securosis. Understanding and selecting a data loss prevention solution. Technicalreport, SANS Institute, 2010.
12. MSDN. File management functions. Режим доступу: http://msdn.microsoft.com/enus/library/windows/desktop/aa364232.aspx.- 3.06.2019 р.

13. OpenDLP. Data loss prevention suite. Режим доступу: https://code.google.com/p/opendlp/ - 3.06.2019 p.
14. Eric Ouellet and Rob McMillan. Magic quadrant for content-aware data loss prevention. Gartner Group Research Note, 2011.
15. Walter Rogowski. The right approach to data loss prevention. Computer Fraud and Security, 2013(8):5, 2013.
16. Asaf Shabtai, Yuval Elovici, and Lior Rokach. A Survey of Data Leakage Detection and Prevention Solutions. Springer Publishing Company, Incorporated, 2012.
17. Mathias Thurman. Ins and outs of extending dlp. Computerworld, 47(18):36, 2013.
18. Jim Trocki. Pc administration tools: Using linux to manage personal computers. In LISA, pages 187–192, 1996.
19. Wikipedia. Mime-type. Режим доступу: http://en.wikipedia.org/wiki/MIME_type - 3.06.2019 p.
20. Wikipedia. Policy. Режим доступу: http://en.wikipedia.org/wiki/Policy - 3.06.2019 p.
21. Wikipedia. Proxy server. Режим доступу: http://en.wikipedia.org/wiki/Proxy_server - 3.06.2019 p.
22. Tobias Wuchner and Alexander Pretschner. Data loss prevention based on data-driven usage control. pages 151–160. IEEE, 2012.

ДОДАТКИ

ДОДАТОК А

Код програми

```
#include <string>
#include <iostream>
#include <Windows.h>
#include <easyhook.h>
using namespace std;

BOOL WINAPI actionHook(std::string params[]);

BOOL WINAPI actionHook(std::string params[])
{
    cout << "\n****Action catched!\n\n";
    return Action(params[]);
}

int _tmain(int argc, _TCHAR* argv[])
{
     std::string params[] = argv[];
    HOOK_TRACE_INFO hookTraceInfo = { NULL };
    cout << GetProcAddress(GetModuleHandle(TEXT(params[0])),
"Action");

    NTSTATUS initStatus = LhInstallHook(
        GetProcAddress(GetModuleHandle(TEXT(params[0])), "Action"),
           actionHook,
        NULL,
        &hookTraceInfo);
    if (FAILED(initStatus))
    {
        wstring s(RtlGetLastErrorString());
        wcout << "APIHooking init failed: ";
        wcout << s;
        cout << "\n\nPress any key to exit.";
        cin.get();
        return -1;
    }
    cout << "APIHooking successfully initiated. Enabling action
without APIHooking.\n";
    Action(params[]);
    cout << "Activate APIHooking for current thread.\n";
    ULONG ACLEntries[1] = { 0 };
    LhSetInclusiveACL(ACLEntries, 1, &hookTraceInfo);
    cout << "Enabling action after activation APIHooking.\n";
     Action(params[]);
    cout << "Uninstall APIHooking\n";
    LhUninstallHook(&hookTraceInfo);
    cout << "Enabling action after uninstalling APIHooking.\n";
     Action(params[]);
    cout << "\n\nWaiting for pending removals\n";
    LhWaitForPendingRemovals();
    cout << "Press any key to exit.";
    cin.get();
    return 0;
}
```