НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Кафедра кібербезпеки

Методичні вказівки до лабораторних робіт з дисципліни

«Комплексні системи захисту інформації»

для студентів спеціальності 125 Кібербезпека та захист інформації

Лаптєв О.А., Король О.Г., Корольов Р.В. Воропай Н.І.

Харків-2023

УДК 004.056(075.8)

Рецензенти: Директор навчально-наукового інституту захисту інформації
Державного університету телекомунікацій
доктор технічних наук, професор Савченко В.А.

Начальник кафедри інституту спеціального зв'язку
та захисту інформації НТУ "КПІ".
доктор технічних наук, доцент Субач І.Ю.

Рекомендовано до друку вченою радою факультету
інформаційних технологій Національного технічного університету
«Харківський політехнічний інститут»
протокол № 3 від 31 березня 2023 р.

Програмні методи захисту інформації: методичні вказівки до лабораторних робіт з дисципліни" Комплексні системи захисту інформації " укл. Лаптєв О.А., Король О.Г., Корольов Р.В., Воропай Н.І. – Харків. 2023.-103 с.

Анотація

У методичних вказівках до лабораторних робіт зроблено спробу подати логічно структурований комплексний виклад теоретичних основ та практичного використання сучасних методів комплексного захисту інформації згідно із загальними підходами в Україні та світі. Розглядаються поняття, класифікація, сфери застосування, складові і ознаки, різні типи підходів, систем і способів забезпечення безпеки інформаційних ресурсів, комплексними системами захисту інформації, та їх застосування в різних сферах сучасного життя.

Для фахівців у галузі захисту інформації, а також студентів, аспірантів що навчаються за напрямом підготовки «Кібербезпека».

УДК 004.056(075.8)
© Євсеєв С.П.,
© Лаптєв О.А.,

ЗМІСТ

Вступ

Комплексні системи захисту інформації - взаємопов'язані сукупності організаційних та інженерно-технічних заходів, програмно-апаратних засобів і методів захисту інформації на об’єктах інформаційної діяльності, в автоматизованих системах обробки інформації та інформаційно технічних системах

Інформація це сила. Інформація стала одним з найбільш потужних важелів економічного розвитку. Володіння інформацією у потрібному місці та у потрібний час є запорукою успіху в будь-якій галузі суспільства. Монопольне володіння певною інформацією виявляється вирішальною перевагою в конкурентній боротьбі і зумовлює, тим самим, високу ціну інформаційної складової.

Створення індустрії переробки інформації, даючи об'єктивні передумови для грандіозного підвищення ефективності життєдіяльності людства, породжує цілий ряд складних і великомасштабних проблем. Однією з таких проблем є надійне забезпечення збереження встановленого статусу використання інформації, що циркулює і обробляється в інформаційно-обчислювальних установках, центрах, системах і мережах, або коротко – в автоматизованих системах обробки даних. Тому виникає актуальне наукове завдання побудова ефективної системи захисту інформації. Ефективне рішення цього наукового завдання можливо тільки застосував комплексну систему захисту інформації. Дисципліна «Комплексна система захисту інформації» присвячена вирішенню саме цього наукового завдання. **Головна функція комплексної системи захисту інформації**-забезпечення конфіденційності, цілісності та доступності інформації.

Раніше проблема захисту інформації вирішувалася досить ефективно застосуванням, в основному, організаційних заходів. До них належали передусім, режимні заходи, охорона, сигналізація і найпростіші програмні засоби захисту інформації. Ефективність використання зазначених заходів досягалася за рахунок концентрації інформації на обчислювальних центрах, як правило, автономних, що сприяло забезпеченню захисту відносно малими засобами. Проблема забезпечення необхідного рівня захисту інформації виявилася досить складною, що вимагає для свого вирішення не просто здійснення деякої сукупності наукових, науково-технічних і організаційних заходів та застосування специфічних засобів і методів, а створення цілісної системи організаційних заходів та застосування специфічних засобів і методів захисту інформації. Координація робіт стосовно захисту інформації в

державному масштабі традиційно здійснювалася і здійснюється Департаментом Держспецзв’язку України, яка створювалася як головна організація з протидії іноземним технічним розвідкам. У зв'язку з викладеними вище об'єктивними причинами до теперішнього часу відбулося переосмислення функцій Департаменту Держспецзв’язку України. Роботи з захисту інформації у нас у країні ведуться досить інтенсивне і вже тривалий час. Накопичено певний досвід. Його аналіз показав, що весь період робіт із захисту інформації досить чітко ділиться на три напрямка:

1. Організаційно-правовий.
2. Інженерно-технічний.
3. Програмно-апаратний.

Кожен з яких характеризується своїми особливостями в принципових підходах до захисту інформації. Концептуальність підходу передбачає розробку єдиної концепції як повної сукупності науково обґрунтованих поглядів, положень і рішень, необхідних і достатніх для оптимальної організації та забезпечення надійності захисту інформації, а також для цілеспрямованої організації всіх робіт із захисту інформації.

Враховуючи різноманіття потенційних загроз інформації в інформаційних системах, складність їх структури і функцій, а також участь людини в технологічному процесі обробки інформації, цілі захисту інформації можуть бути досягнуті тільки шляхом створення системи захисту інформації на основі комплексного підходу. Комплексна система захисту інформації є сукупністю методів і засобів, об'єднаних єдиним цільовим призначенням, які забезпечують необхідну ефективність захисту інформації в автоматизованих системах та мережах. Комплексність системи захисту інформації досягається охопленням всіх можливих загроз і узгодженням між собою різнорідних методів і засобів, що забезпечують захист всіх елементів інформаційної системи.

Тому дисципліна комплексні системи захисту інформації є необхідної складової загальної системи захисту інформації.

Лабораторна робота №1

Тема: Налагодження віддаленого доступу до комп'ютера користувача

Мета: Навчитися встановлювати віддалене з'єднання з ПК за допомогою програми TeamViewer.

Час проведення: - 2 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

TeamViewer є швидким і безпечним комплексним рішенням для одержання віддаленого доступу до комп'ютерів і мереж. Завдяки безлічі ефективних функцій для віддаленого доступу, які спрощують віддалене управління, проведення онлайн-нарад і роботу хмарної служби підтримки, існує цілий ряд ресурсів для демонстрації всіх можливостей TeamViewer.

Провести первинне налаштування TeamViewer дуже легко: просто встановіть програмне забезпечення, вкажіть мету використання (комерційне або приватне використання), створіть ім'я і пароль для вашого комп'ютера і запишіть їх для використання в подальшій роботі. Після завершення процесу установки ви будете перенаправлені на головну сторінку TeamViewer, яка розділена на дві вкладки: віддалене управління та онлайн-наради. Звідси ви зможете управляти різними функціями.

TeamViewer - це спеціальна програма для віддаленого підключення, дозволяє надати тимчасовий доступ до свого комп'ютера. Дуже популярна у всьому світі - за словами розробників модуль встановлений понад 1 мільярд раз.

Завантажити програму можна з офіційного сайту: **https://www.teamviewer.com/ru/download/windows/**

Крім того, TeamViewer дозволяє також створювати нові файли, передавати дані, спілкуватися, проводити конференції. Головна його особливість - відсутність зовнішнього IP, досить буквально двох ідентифікацій в програмі.

Наприклад, з її допомогою можна проводити конференції в цілому до 25 осіб, що живуть в різних країнах і континентах. При цьому вам навіть не потрібно буде додавати їх, спілкування почнеться за допомогою одного «кліка». Це дозволить вам усунути територіальна відстань, об'єднавши необхідних людей за допомогою Віртуального простору.

Саме за допомогою цієї програми проводиться віддалена комп'ютерна допомога - послуга, яку надають різні IT-компанії. І це, в першу чергу, говорить про її безпеки.

Багато хто думає, що дана програма для віддаленого доступу - небезпечна. Однак слід вас в цьому переконати. Так, по-перше, процес передачі даних проходить з шифруванням всіх сеансів, а також відбувається обмін

ключами. Захищені канали застосовують протоколи https / SSL, що і дозволяє говорити про питання безпеки.

1. Без знання вашого ID і пароля під'єднатися до вашого комп'ютера не вийде!

2. Всі дії фахівця тех. підтримки ви будете бачити на своєму екрані. Всі рухи мишки, відкриття вікон звантаження та установку ви спостерігаєте і контролюєте в режимі реального часу.

3. Завжди є можливість натиснути кнопку «відключити». При наступному підключенні вже буде створений новий пароль. А це означає, що повторно під'єднатися без вашого відома не вийде.

Основна проблема віддаленого доступу через дану програму - брандмауери, маршрутизація локальних адрес, заблоковані порти. Саме це іноді не дає грамотно обійти всі заслони, приєднатися до комп'ютера необхідного людини. В такій ситуації ви можете використовувати TeamViewer Web Connector, який працює через браузер, а не на основі ОС. Є тут ще один плюс програми - вона самостійно вибирає найкращу швидкість і якість для обміну інформацією між ПК.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Інсталює програму TeamViewer на комп'ютери.
2. Запуск TeamViewer на комп'ютері, до якого ви будите приєднатися.
3. Запишіть дані, які з'являться у вікні TeamViewer на віддаленому комп'ютері в полях «Ваш ID» і «Пароль» (Рис 1.2).

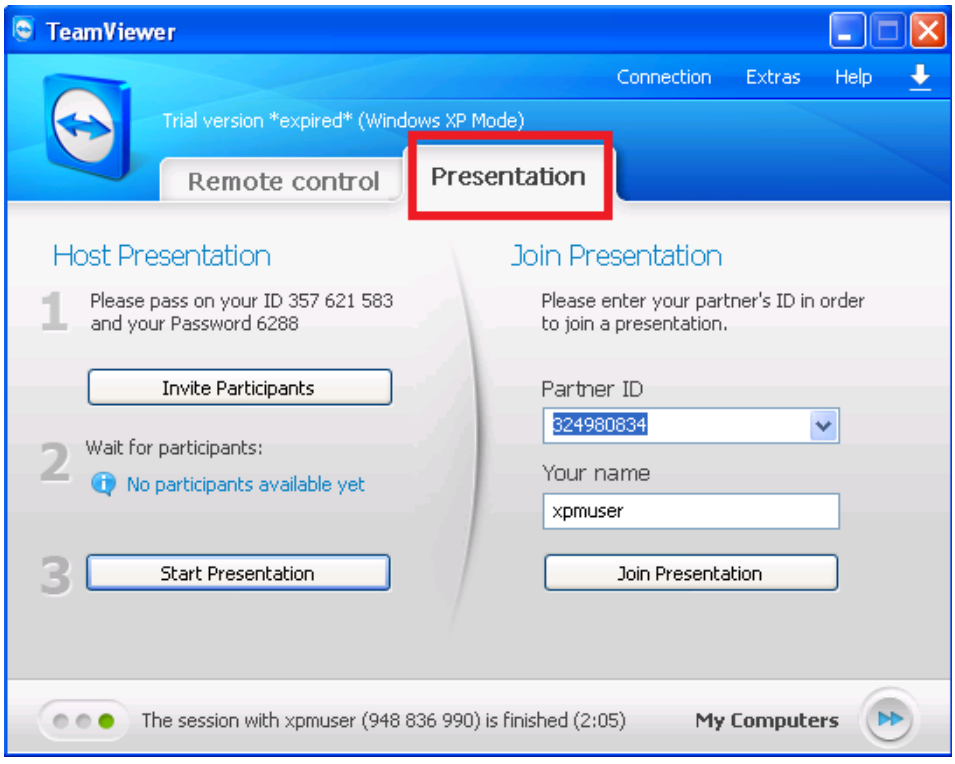

Рисунок 1.2 - Вікно TeamViewer

4. Скріншот даного вікна помістити у звіт.

5. Активуйте TeamViewer на своєму комп'ютері.

6. В поле ID партнера введіть той код, який відображався в поле «Ваш ID» на віддаленому ПК.

Прі тому має бути активна кнопка «Віддалене керування».

7. Скриншот даного вікна помістити у звіт.

8. Натисніть кнопку «Приєднатися до партнера».

9. У вікні, введіть пароль з віддаленого комп'ютера (даний код відображався в поле «Пароль» на віддаленому комп'ютере).

10. Скріншот даного вікна помістити у звіт.

11. Після введення вказаного значення в єдине поле віконця натисніть кнопку «Вхід в систему». «Робочий стіл» вибраного комп'ютера зобразиться в окремому віконці на даному ПК.

12. Скріншот даного вікна помістити у звіт.

13. На робочому столі віддаленого комп'ютера натисніть «Пуск» → «Комп'ютер».

14. Скріншот даного вікна помістити у звіт.

15. Не деінсталюйте програму TeamViewer на комп'ютерах

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Програма TeamViewer: призначення, можливості.
2. Чому TeamViewer безпечна?
3. Як в TeamViewer організована передача і копіювання файлів і папок?
4. Охарактеризуйте TeamViewer Web Connector.
5. Які існують проблеми віддаленого підключення?

Лабораторна робота №2

Тема: Критерії оцінки інформаційної безпеки та аспекти захисту інформації.

Мета: Ознайомитися з поняттями захисту інформації, інформаційної безпеки, властивостями інформації, аспектами захисту інформації та критеріями оцінки інформаційної безпеки.

Час проведення: - 4 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Захист інформації (англ. Data protection) — сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації. Термін вживається в Україні для опису комплексу заходів по забезпеченню інформаційної безпеки [9,10,12].

Інформаційна безпека (information security) — збереження конфіденційності, цілісності та доступності інформації; крім того, можуть враховуватися інші властивості, такі, як автентичність, відстежуваність, неспростовність та надійність [9,10,12].

Конфіденційність (англ. confidentiality, privacy) — властивість не підлягати розголосові; довірливість, секретність, суто приватність [9,10,12,14].

Типологія конфіденційності

Конфіденційність адміністративна [mandatory confidentiality] — послуга безпеки, що забезпечує конфіденційність інформації відповідно до принципів керування доступом адміністративного [11,13,17].

Конфіденційність довірча [discretionary confidentiality] — послуга безпеки, що забезпечує конфіденційність інформації відповідно до принципів керування доступом довірчого[11,13,17].

Конфіденційність інформації [information confidentiality] — властивість інформації, яка полягає в тому, що інформація не може бути отримана неавторизованим користувачем і (або) процесом[11,13,17].

Інформація зберігає конфіденційність, якщо дотримуються встановлені правила ознайомлення з нею.

Цілісність (англ. integrity) — внутрішня єдність, пов'язаність усіх частин чого-небудь, єдине ціле [11,13,17]. В інформаційній системі — стан даних або інформаційної системи, в якій дані та програми використовуються встановленим чином, що забезпечує:

- стійку роботу системи;

- автоматичне відновлення у випадку виявлення системою потенційної помилки;

- автоматичне використання альтернативних компонентів замість тих, що вийшли з ладу.

Для інформаційної системи можна розглядати такі поняття як цілісність даних, цілісність інформації, цілісність бази даних, цілісність інформаційної системи і таке інше.

Цілісність даних [data integrity] — в інформаційній системі — стан при якому дані, що зберігаються в системі, в точності відповідають даним у вихідних документах; властивість, що має відношення до набору даних і означає, що дані не можуть бути змінені або зруйновані без санкції на доступ. Цілісність даних вважається збереженою, якщо дані не спотворені і не зруйновані (стерті) [17,18,19].

Семантична цілісність даних [semantic data integrity] — стан даних, коли вони зберігають свій інформаційний зміст та однозначність інтерпретації в умовах випадкових впливів[17,18,19].

Цілісність інформації [information integrity] — властивість інформації, яка полягає в тому, що інформація не може бути модифікована неавторизованим користувачем і (або процесом). Інформація зберігає цілісність, якщо дотримуються встановлені правила її модифікації (видалення) [17,18,19].

Цілісність бази даних [database integrity] — стан бази даних, коли всі значення даних правильні в тому сенсі, що відображають стан реального світу (в межах заданих обмежень по точності та часовій узгодженості) і підпорядковуються правилам взаємної не суперечливості [17,18,19]. Підтримка цілісності бази даних включає перевірку цілісності і відновлення з будь-якого неправильного стану, який може бути виявлено; це входить у функції адміністратора бази даних.

Цілісність системи [system integrity] — властивість системи, яка полягає в тому, що жоден її компонент не може бути усунений, модифікований або доданий з порушенням політики безпеки [17,18,19]..

Цілісність адміністративна [mandatory integrity] — послуга безпеки, яка забезпечує цілісність інформації відповідно до принципів адміністративного керування доступом.

Цілісність довірча [discretionary integrity] — послуга безпеки, яка забезпечує цілісність інформації відповідно до принципів керування доступом довірчого.

Цілісність об'єкта [object integrity] — властивість об'єкта доступу, що характеризує його авторизований стан.

Доступність (англ. Availability) — властивість інформаційного ресурсу, яка полягає в тому, що користувач та/або процес, який володіє відповідними повноваженнями, може використовувати цей ресурс відповідно до правил, встановлених політикою безпеки не очікуючи довше заданого (прийнятного) інтервалу часу.

Суть властивості полягає в тому, що потрібний інформаційний ресурс знаходиться у вигляді, необхідному користувачеві, в місці, необхідному користувачеві, і в той час, коли він йому необхідний.

Апелювання (англ. non-repudiation) — можливість довести, що автором є саме заявлена людина (юридична особа), і ніхто інший [18].

Підзвітність (англ. accountability) — властивість інформаційної системи, що дозволяє фіксувати діяльність користувачів, використання ними пасивних об‘єктів та однозначно встановлювати авторів певних дій в системі [18].

Достовірність (англ. reliability)- властивість інформації, яка визначає ступінь об'єктивного, точного відображення подій, фактів, що мали місце [18].

Автентичність (англ. authenticity) — властивість, яка гарантує, що суб'єкт або ресурс ідентичні заявленим [18].

Відповідно до властивостей інформації, виділяють такі загрози її безпеці:

- **загрози цілісності:**
- знищення;
- модифікація;
- **загрози доступності:**
- блокування;
- знищення;
- **загрози конфіденційності:**
- несанкціонований доступ (НСД);
- витік;
- розголошення.

Аспекти захисту інформації

Конфіденційність — захист від несанкціонованого ознайомлення з інформацією.

Цілісність — захист інформації від несанкціонованої модифікації.

Доступність — захист (забезпечення) доступу до інформації, а також можливості її використання. Доступність забезпечується як підтриманням систем в робочому стані так і завдяки способам, які дозволяють швидко відновити втрачену чи пошкоджену інформацію.

Кожен вид ЗІ забезпечує окремі аспекти ІБ:

Технічний — забезпечує обмеження доступу до носія повідомлення апаратно-технічними засобами (антивіруси, фаєрволи, маршрутизатори, токіни, смарт-карти тощо):

- попередження витоку по технічним каналам;
- попередження блокування;

Інженерний — попереджує руйнування носія внаслідок навмисних дій або природного впливу інженерно-технічними засобами (сюди відносять обмежуючі конструкції, охоронно-пожежна сигналізація).

Криптографічний — попереджує доступ до носія інформації за допомогою математичних перетворень повідомлення:

- попередження несанкціонованої модифікації ;
- попередження НС розголошення.

Організаційний — попередження доступу на об’єкт інформаційної діяльності сторонніх осіб за допомогою організаційних заходів (правила розмежування доступу).

Інформаційні системи можна розділити на три частини: ***програмне*** забезпечення, ***апаратне*** забезпечення та ***комунікації*** з метою цільового застосування (як механізму захисту і попередження) стандартів інформаційної безпеки. Самі механізми захисту реалізуються на трьох рівнях або шарах:

– Фізичний.
– Особистісний.
– Організаційний.

По суті, реалізація політик і процедур безпеки покликана надавати інформацію адміністраторам, користувачам і операторам про те як правильно використовувати готові рішення для підтримки безпеки.

Об'єктивно категорія «інформаційна безпека» виникла з появою засобів інформаційних комунікацій між людьми, а також з усвідомленням людиною наявності у людей і їхніх співтовариств інтересів, яким може бути завданий

збитку шляхом дії на засоби інформаційних комунікацій, наявність і розвиток яких забезпечує інформаційний обмін між всіма елементами соціуму.

Враховуючи вплив на трансформацію ідей інформаційної безпеки, в розвитку засобів інформаційних комунікацій можна виділити декілька етапів:

I етап — до 1816 року — характеризується використанням природно виникаючих засобів інформаційних комунікацій. В цей період основне завдання інформаційної безпеки полягало в захисті відомостей про події, факти, майно, місцезнаходження і інші дані, що мають для людини особисто або співтовариства, до якого вона належала, життєве значення [20,21].

II етап — починаючи з 1816 року — пов'язаний з початком використання штучно створюваних технічних засобів електро- і радіозв'язку. Для забезпечення скритності і перешкодостійкості радіозв'язку необхідно було використовувати досвід першого періоду інформаційної безпеки на вищому технологічному рівні, а саме застосування перешкодостійкого кодування повідомлення (сигналу) з подальшим декодуванням прийнятого повідомлення (сигналу) [20,21].

III етап — починаючи з 1935 року — пов'язаний з появою засобів радіолокацій і гідроакустики [20,21].

Основним способом забезпечення інформаційної безпеки в цей період було поєднання організаційних і технічних заходів, направлених на підвищення захищеності засобів радіолокацій від дії на їхні приймальні пристрої активними маскуючими і пасивними імітуючими радіоелектронними перешкодами.

IV етап — починаючи з 1946 року — пов'язаний з винаходом і впровадженням в практичну діяльність електронно-обчислювальних машин (комп'ютерів). Завдання інформаційної безпеки вирішувалися, в основному, методами і способами обмеження фізичного доступу до устаткування засобів добування, переробки і передачі інформації [20,21].

V етап — починаючи з 1965 року — обумовлений створенням і розвитком локальних інформаційно-комунікаційних мереж. Завдання інформаційної безпеки також вирішувалися, в основному, методами і способами фізичного захисту засобів добування, переробки і передачі інформації, об'єднаних в локальну мережу шляхом адміністрування і управління доступом до мережевих ресурсів [20,21].

VI етап — починаючи з 1973 року — пов'язаний з використанням надмобільних комунікаційних пристроїв з широким спектром завдань. Загрози інформаційній безпеці стали набагато серйознішими. Для забезпечення інформаційної безпеки в комп'ютерних системах з безпровідними мережами

передачі даних потрібно було розробити нові критерії безпеки. Утворилися співтовариства людей — хакерів, що ставлять собі за мету нанесення збитку інформаційній безпеці окремих користувачів, організацій і цілих країн. Інформаційний ресурс став найважливішим ресурсом держави, а забезпечення його безпеки — найважливішою і обов'язковою складовою національної безпеки.

Формується інформаційне право — нова галузь міжнародної правової системи [20,21].

VII етап — починаючи з 1985 року — пов'язаний із створенням і розвитком глобальних інформаційно-комунікаційних мереж з використанням космічних засобів забезпечення. Можна припустити що черговий етап розвитку інформаційної безпеки, очевидно, буде пов'язаний з широким використанням надмобільних комунікаційних пристроїв з широким спектром завдань і глобальним охопленням у просторі та часі, забезпечуваним космічними інформаційно-комунікаційними системами [20,21].

Для вирішення завдань інформаційної безпеки на цьому етапі необхідним є створення макросистеми інформаційної безпеки людства під егідою ведучих міжнародних форумів.

Базові поняття інформаційної безпеки:

Інформаційна безпека держави — стан захищеності життєво важливих інтересів людини, суспільства і держави, при якому запобігається нанесення шкоди через: неповноту, невчасність та невірогідність інформації, що використовується; негативний інформаційний вплив; негативні наслідки застосування інформаційних технологій; несанкціоноване розповсюдження, використання і порушення цілісності, конфіденційності та доступності інформації.

Інформаційна безпека організації — цілеспрямована діяльність її органів та посадових осіб з використанням дозволених сил і засобів по досягненню стану захищеності інформаційного середовища організації, що забезпечує її нормальне функціонування і динамічний розвиток.

Інформаційна безпеки особистості характеризується як стан захищеності особистості, різноманітних соціальних груп та об'єднань людей від впливів, здатних проти їхньої волі та бажання змінювати психічні стани і психологічні характеристики людини, модифікувати її поведінку та обмежувати свободу вибору.

Забезпечення ІБ держави

Згідно з українським законодавством, вирішення проблеми інформаційної безпеки має здійснюватися шляхом:

- створення повнофункціональної інформаційної інфраструктури держави та забезпечення захисту її критичних елементів;

- підвищення рівня координації діяльності державних органів щодо виявлення, оцінки і прогнозування загроз інформаційній безпеці, запобігання таким загрозам та забезпечення ліквідації їхніх наслідків, здійснення міжнародного співробітництва з цих питань;

- вдосконалення нормативно-правової бази щодо забезпечення інформаційної безпеки, зокрема захисту інформаційних ресурсів, протидії комп'ютерній злочинності, захисту персональних даних, а також правоохоронної діяльності в інформаційній сфері;

- розгортання та розвитку Національної системи конфіденційного зв'язку як сучасної захищеної транспортної основи, здатної інтегрувати територіально розподілені інформаційні системи, в яких обробляється конфіденційна інформація [13].

Забезпечення ІБ підприємства/організації

В Україні забезпечення ІБ здійснюється шляхом захисту інформації — у випадку, коли необхідність захисту інформації визначена законодавством в галузі ЗІ. Для реалізації захисту інформації створюється Комплексна система захисту інформації (КСЗІ), або, у випадку, коли суб'єкт ІБ має наміри розробити і реалізувати політику ІБ і може реалізовувати їх без порушення вимог законодавства:

- міжнародними стандартами ISO: ISO/IEC 17799:2005, ISO/IEC 27001:2005 та ін. — для підтримки рішень на основі ITIL та COBIT і виконання вимог англ. Sarbanes-Oxley Act (акту Сербайнза-Оклі про відповідальність акціонерів за обізнаність про стан своїх активів). Тоді на підприємстві створюється Система управління інформаційною безпекою (СУІБ), яка повинна відповідати усім вимогам міжнародних стандартів в галузі ІБ.

- власними розробками.

Забезпечення ІБ особистості

Органи (підрозділи) забезпечення ІБ

1. Міжнародні організації.

2. Державні органи.

- Відділи спецслужб держави.

- Спеціально уповноважений орган держави з питань захисту інформації (зараз в Україні — це Державна служба спеціального зв'язку та захисту інформації (скор. ДССЗЗІ)

3. Підрозділи підприємства.

На підприємстві функцію забезпечення ІБ може виконувати як окремий відділ Служби безпеки підприємства, так і окрема Служба (Служба захисту інформації).

Для контролю за КСЗІ в обов'язковому порядку створюється Служба захисту інформації в інформаційно-телекомунікаційній системі (сама назва «Служба» не є обов'язковою).

Функції з контролю за СУІБ покладаються на певний відділ підприємства.

Законодавчі вимоги і регулювання ІБ

Загальнозаконодавчі вимоги – інформаційне законодавство держави, спеціалізовані нормативні акти (в Україні — це Нормативні документи в галузі технічного захисту інформації (скор. НД ТЗІ).

Галузеві вимоги (галузеві стандарти тощо).

Критерії оцінки інформаційної безпеки (англ. Common Criteria) є методологічною базою для визначення вимог захисту комп'ютерних систем від несанкціонованого доступу, створення захисних систем та оцінки ступені захищеності.

За допомогою критеріїв можливо порівняти різні механізми захисту інформації та визначити необхідну функціональність таких механізмів у розробці захищених комп'ютерних систем.

Для характеристики основних критеріїв інформаційної безпеки застосовують модель тріади CIA.

Ця система передбачає такі основні характеристики інформаційної безпеки як конфіденційність, цілісність, доступність.

Інформаційні системи аналізуються в трьох головних секторах: технічних засобах, програмному забезпеченні і комунікаціях, з метою ідентифікування і застосування промислових стандартів інформаційної безпеки, як механізми захисту і запобігання, на трьох рівнях або шарах: фізичний, особистий і організаційний. По суті, процедури або правила запроваджуються для інформування адміністраторів, користувачів та операторів щодо використовування захисної продукції для гарантування інформаційної безпеки в межах організацій.

В Україні також розробляються і використовуються критерії інформаційної безпеки. Наприклад департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України прийняв нормативний документ технічного захисту інформації 2.5-004-99 «Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу» який подібний до моделі тріади CIA (https://tzi.ua/ua/nd_tz_2.5-004-99.html#:~:text=Цей%20нормативний%20документ%20—%20установлює%20критерії,ютерних%20системах%2C%20від%20несанкціонованого%20доступу).

Функціональні критерії

Конфіденційність. Загрози, що відносяться до несанкціонованого ознайомлення з інформацією, становлять загрози конфіденційності. Якщо існують вимоги щодо обмеження можливості ознайомлення з інформацією, то відповідні послуги відносяться до критеріїв конфіденційності.

Цілісність. Загрози, що відносяться до несанкціонованої модифікації інформації, становлять загрози цілісності. У випадку, якщо існують вимоги щодо обмеження можливості модифікації інформації, то їх відносяться до критеріїв цілісності.

Доступність. Загрози, що відносяться до порушення можливості використання комп'ютерних систем або оброблюваної інформації, становлять загрози доступності. Якщо існують вимоги щодо захисту від відмови в доступі або захисту від збоїв, то їх відносяться до критеріїв доступності.

Спостереженість. Ідентифікація і контроль за діями користувачів, керованість комп'ютерною системою становлять предмет спостереженості і керованості. Якщо існують вимоги щодо контролю за діями користувачів або легальністю доступу і за спроможністю комплексу засобів захисту виконувати свої функції, то відповідні функції відносяться до критеріїв спостереженості.

Окрім функціональних критеріїв захищеності існують такі критерії гарантій, що дозволяють оцінити коректність реалізації систем захисту. Ці критерії включають вимоги до архітектури комплексу засобів захисту, середовища розробки, послідовності розробки, випробування комплексу засобів захисту, середовища функціонування і експлуатаційної документації.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустіть браузер Інтернет (Це може бути Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, чи будь-який інший, що встановлений на вашому комп'ютері).

2. Користуючись однією з пошуковх систем (Yahoo!, Google, чи будь-якою іншою) ознайомтеся із законодавчою базою України, що стосується захисту інформації та інформаційної безпеки. Назви основних законів, указів президента, постанов, положень *запишіть до звіту (не менше 15)*.

3. На офіційному сайті Верховної ради «Законодавство України» (http://zakon2.rada.gov.ua/laws) знайдіть Закон України «Про інформацію», ознайомтесь з його основними положеннями та занотуйте до звіту такі відомості:

- визначення основних термінів;
- основні види інформації;
- основні аспекти відповідальності за порушення законодавства про інформацію.

4. На офіційному сайті Верховної ради «Законодавство України» (http://zakon2.rada.gov.ua/laws) знайдіть Закон України «Про захист інформації в інформаційно-телекомунікаційних системах», ознайомтесь з його основними положеннями та *занотуйте до звіту такі відомості*:

- визначення основних термінів (не менше 7);
- об'єкти захисту та суб'єкти відносин;
- умови обробки державної інформації в системі;
- повноваження державних органів у сфері захисту інформації в системі;
- основні аспекти відповідальності за порушення законодавства про захист інформації в системах.

5. Зробіть та запишіть до звіту висновки по роботі.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що називається інформаційною безпекою?
2. Дайте визначення поняття захисту інформації.
3. Назвіть та охарактеризуйте основні властивості інформації.
4. Назвіть та охарактеризуйте основні аспекти захисту інформації та інформаційної безпеки.
5. Які виділяють загрози безпеки інформації відповідно до її властивостей?
6. Назвіть та дайте визначення базовим поняттям інформаційної безпеки.

7. Охарактеризувати шляхи забезпечення інформаційної безпеки держави, підприємства, особистості.

8. Назвати та охарактеризувати основні критерії інформаційної безпеки.

9. Що являють собою законодавчі вимоги до інформаційної безпеки?

10. Що являє собою модель тріади CIA?

Лабораторна робота №3

Тема: Паролі, правила роботи з ними.

Мета: Ознайомитися з поняттям паролю, основними засобами збереження та доступу до паролів, правилами роботи з паролями. Набуття практичних навичок аудиту парольного захисту документів.

Час проведення: - 6 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Паролі - це ключі, які відкривають доступ до особистих даних, що зберігаються на комп'ютері та в облікових записах в Інтернеті.

Якщо зловмисники вкрадуть ці дані, вони можуть скористатися ними для відкриття нових рахунків кредитних карт, отримання кредиту або виконання через Інтернет інших дій від вашого імені. Дуже часто ви можете не підозрювати про такі дії до тих пір, поки не стане занадто пізно.

З величезною кількістю системних паролів та сайтів, доступ до яких надається лише зареєстрованим користувачам, неможливо запам'ятати всі пари «ім'я користувача / пароль», якщо тільки ви не використовуєте одні й ті самі ім'я та пароль для всіх облікових записів або не записуєте їх. Обидва ці методи мають знижений рівень безпеки.

Той, хто використовує одні й ті ж ім'я користувача та пароль, починаючи електронною поштою і закінчуючи особистими банківськими рахунками, надають повний доступ до власного життя кожному, хто зможе зламати один з використовуваних акаунтів, так як таким чином, зловмисники зможуть використовувати отримані дані для будь-яких інших облікових записів. Отримати доступ до важливої інформації можна через файли cookie, які зберігаються на комп'ютері (це залежить від ступеня уразливості використовуваного браузера), і за допомогою сайтів, які не можуть використовувати протокол http або технологію SSL (відкритий стандарт для створення безпечних каналів підключення, які запобігають витоку важливих конфіденційних відомостей, таких як номери кредитних карт). В даному випадку, ваші ім'я користувача та пароль виявляються у відкритому доступі в Інтернеті і можуть бути отримані будь-якою кількістю комп'ютерів.

При записуванні ім'я користувача та пароль на папір завжди існує можливість втратити цей листок, або хто-небудь може знайти його і отримати

доступ до ваших акаунтів. Ймовірність такої загрози значно зростає, якщо ви залишаєте листок з даними під клавіатурою, приклеюєте його на монітор, ноутбук або просто залишаєте його на столі. Пароль використовується для забезпечення безпеки комп'ютерних програм, і тому, записуючи його на папір, ви просто видаляєте всяку безпеку. З іншого боку, ім'я користувача та пароль можуть бути легко викрадені, якщо вони надсилаються через Інтернет відкритим текстом, але зазвичай ті користувачі, які записують дані на папір для різних акаунтів, використовують різні паролі. Тому при втраті листка з паролем вони не втратять контроль над усіма обліковими записами.

Одним з кращих способів забезпечити безпеку при відкритті акаунтів є використання різних імен користувача та паролів при кожній реєстрації. Запам'ятати їх буває дуже складно (тому часто використовуються методи, описані вище). Все ж існує кращий спосіб збереження паролів - використання інструменту управління паролями. Таким чином, значно спрощується збереження паролів для кожного облікового запису. Іншою важливою перевагою використання менеджера паролів є те, що можна використовувати безпечніші паролі і не боятися забути їх.

Існує безліч менеджерів паролів, доступних в Інтернеті, - деякі платні, інші безкоштовні. Вибір типу програм залежить від користувача.

Менеджер паролів - програмне забезпечення, яке допомагає користувачеві працювати з паролями і PIN-кодами. У подібного програмного забезпечення зазвичай є в наявності місцева база даних або файли, які містять зашифровані дані пароля. Багато менеджерів паролів також працюють як заповнювач форми, тобто вони заповнюють поле користувач і дані пароля автоматично в формах.

Зазвичай вони реалізовані як розширення браузера.

Менеджери паролів діляться на три основні категорії:

Десктоп - зберігають паролі до програмного забезпечення, встановленого на жорсткому диску комп'ютера.

Портативні - зберігають паролі до програмного забезпечення на мобільних пристроях, таких як КПК, смартфон або до портативних додатків на USB флеш-накопичувачі.

Мережеві - менеджери паролів онлайн, де паролі збережені на веб-сайтах провайдерів.

Менеджери паролів можуть також використовуватися як захист від фішингу. На відміну від людей, програма менеджер паролів може звертатися з автоматизованим скриптом логіна не сприйнятливі до візуальних імітацій, які

схожі на веб-сайти. З цією вбудованою перевагою використання менеджера паролів вигідно, навіть якщо у користувача є всього кілька паролів, які він пам'ятає. Однак не всі менеджери паролів можуть автоматично звертатися з більш складними процедурами ідентифікації, накладеними багатьма банківськими веб-сайтами.

Менеджери паролів зазвичай використовують вибраний користувачем основний пароль, або секретну фразу (passphrase), щоб сформувати ключ, використовуваний для зашифровки збережених паролів. Цей основний пароль повинен бути досить складним, щоб встояти при атаках зловмисників (наприклад повний перебір).

Якщо основний пароль буде зламаний, то будуть розкриті всі збережені в базі даних програми паролі. Це демонструє зворотний зв'язок між зручністю використання і безпекою: єдиний пароль може бути більш зручний, але якщо він буде зламаний, то поставить під загрозу всі збережені паролі [2].

Основний пароль може також бути атакований і виявлений при використанні кейлоггера або акустичного криптоаналізу (acoustic cryptanalysis). Така загроза може бути знижена шляхом використання віртуальної клавіатури, як, наприклад, в KeePass.

Деякі менеджери паролів включають генератор паролів. Згенеровані паролі можуть бути відгадані, якщо менеджер пароля не використовує криптографічно безпечний генератор випадкових чисел.

Онлайн менеджер паролів - веб-сайт, який надійно зберігає дані логіна. Таким чином це мережева версія звичайного десктоп-менеджера паролів.

Переваги онлайн менеджерів паролів над десктоп-версіями - це мобільність (вони можуть використовуватися на будь-якому комп'ютері з web-браузером і інтернет-з'єднанням, без необхідності встановлювати програмне забезпечення) і менший ризик втрати паролів через злодійство або пошкодження PC. Ризик пошкодження може бути в значній мірі знижений, якщо заздалегідь будуть створені резервні копії.

Головний недолік онлайн менеджерів паролів - необхідна довіра хостингу сайту. Неодноразові зломи і втрати централізовано збереженої інформації на сервері не вселяють довіри.

Існують змішані рішення. Ряд ресурсів, таких як FortNotes, що надають послуги онлайн-зберігання паролів та інших секретних даних, поширюють вихідні коди цих систем. Можливість провести аудит коду та встановити таку систему на захищений фаєрволом сервер або на сервер, що не має прямий вихід в Інтернет, дозволяє вирішити проблему з можливою компрометацією даних.

Використання мережевого менеджера паролів - альтернатива технології єдиного входу (Single Sign On), такий як OpenID або Microsoft's Windows Live ID, і може використовуватися як тимчасова міра, поки не буде прийнятий кращий метод.

Також існує менеджери паролів з бар'єрним захистом. У цьому випадку захищається інтернет-акаунт користувача в цілому. Периметр захисту будується починаючи від протидії клавіатурним і екранним шпигунам, і закінчуючи захистом від підміни IP-адреси мережевого ресурсу. Прикладом є Keepy Internet Password Security. Для мережевого захисту використовується Google Public DNS, а протидія шпигунам забезпечується автоматичною підстановкою авторизаційних даних в web-форми.

Правила роботи з паролями.

Для зловмисника найнадійний пароль виглядає як випадковий набір знаків. Наступні критерії допоможуть у виборі пароля.

Використовуйте якомога більше символів. Кожен додатковий знак збільшує ступінь захисту пароля. Пароль повинен містити не менш 8 знаків; 14 знаків і більше є ідеальним варіантом.

Так як багато систем дозволяють використовувати знак пробілу при створенні пароля, можна скласти пароль з декількох слів - парольну фразу. Такі фрази легше запам'ятати і важче підібрати.

Використовуйте комбінацію з літер, цифр та інших символів. Чим більше різних знаків містить пароль, тим важче його підібрати. Інші важливі відомості:

Чим менше різних символів ви використовуєте, тим довше повинен бути ваш пароль. Пароль з 15 випадково вибраних літер і цифр приблизно в 33 тисячі разів надійніше, ніж пароль з 8 знаків, що містить різні типи наявних на клавіатурі знаків. Якщо немає можливості включити в пароль символи, слід зробити його значно довше, щоб забезпечити ту ж ступінь захисту. Ідеальний пароль поєднує в собі довжину і різноманітність знаків.

Використовуйте всі символи клавіатури, а не тільки часто використовувані. Цифри і символи, що вводяться за допомогою клавіші Shift, також часто використовуються при створенні паролів. Пароль буде надійніше, якщо ви використовуєте всі наявні на клавіатурі символи, включаючи знаки пунктуації, розташовані не в верхньому ряду клавіатури, і символи, характерні тільки для вашої мови. Використовуйте слова і фрази, легкі для запам'ятовування, але не очевидні для зловмисників.

Найпростіше запам'ятати паролі і парольні фрази, записавши їх. Всупереч загальноприйнятій думці, немає нічого страшного в записі пароля, якщо дані при цьому захищені належним чином.

Паролі, записані на папері, звичайно важче зламати через Інтернет, ніж паролі, що зберігаються в диспетчері паролів, на веб-сайті або в іншій програмі для зберігання даних.

Шість етапів створення надійного пароля що легко запам’ятовується

1. Придумайте речення, яке точно не забудете, Ця пропозиція і буде основою для надійного пароля або парольної фрази. Пропозиція повинна бути незабутньою (наприклад, "Моєму синові Павлу три роки").

2. Переконайтеся, що обрана вами система перевірки пароля допускає використання ідентифікаційних фраз. Якщо є можливість використовувати парольну фразу (з пробілами між знаками), скористайтеся нею.

3. Якщо використання ідентифікаційних фраз недопустимо в даній системі, скористайтеся звичайним паролем. Складіть нове безглузде слово з перших букв усіх слів, що входять до створеної пропозиції. У нашому прикладі вийде: "мсптр".

4. Ускладніть комбінацію. Використовуючи великі літери, малі літери і цифри. Можна поміняти місцями букви в слові або навмисно допустити орфографічні помилки. Наприклад, в парольній фразі, наведеній вище, можна допустити помилку в імені або замінити слово "три" на цифру 3. Є безліч можливих підстановок, і чим довше пропозиція, тим більш надійним буде пароль. Наш приклад можна перетворити так: "Моєму синові Па8лУ 3 роки". Якщо комп'ютер або система не підтримують парольні фрази, той же метод можна застосувати і до простого паролю. Наприклад, "мСп3Р".

5. Нарешті, замініть окремі символи. Можна використовувати знаки, схожі на літери, об'єднувати слова (видаляючи пробіл між ними) і т. п. Дотримуючись нашого прикладу, ми отримуємо:

"Моєму $иНуП@в8лУ 3 року" або "м$п3Р!".

6. Перевірте свій новий пароль за допомогою програми перевірки паролів. Програма перевірки паролів на цьому веб-сайті визначить надійність вибраного пароля, як тільки ви його введете і не збережете при цьому.

Методи створення пароля, які не слід використовувати.

Вище ми розглянули як створити надійний пароль. Тепер розглянемо те, що не потрібно робити при підборі надійних паролів. Існують загальноприйняті методи, про які можуть знати і зловмисники.

Щоб уникнути створення ненадійного пароля: не використовуйте послідовні комбінації і повторювані символи. Такі поєднання (наприклад, "12345678", "222222", "abcdefg" або поєднання сусідніх букв на клавіатурі) не є надійними паролями.

Уникайте використання тільки замін схожих цифр і символів. Злочинців та інших зловмисників, що володіють достатніми знаннями для підбору і злому пароля, не вдасться ввести в оману подібними замінами, наприклад "і" на "1" або "а" на "@" в словах "M1cr0$0ft" або "П@р0ль ".

Однак не варто нехтувати такими замінами в поєднанні з іншими методами підвищення надійності пароля, такими як збільшення довжини, неправильне написання, використання великих і малих букв.

Не застосовуйте своє ім'я користувача в якості пароля. Уникайте також використання інших особистих даних (своїх або своїх близьких), таких як ім'я, дата народження, код соціального страхування і т. д. Ці відомості використовуються зловмисниками в першу чергу.

Уникайте словникових слів на будь-якій мові. Зловмисники володіють досконалими засобами, що дозволяють швидко підібрати паролі, в основі яких лежать слова з різних мов; слова, написані задом наперед; поширені орфографічні помилки і заміни, а також всі види лайки та інших слів, які не вимовляють при дітях.

Використовуйте кілька паролів. При зломі одного комп'ютера або системи, де використовується певний пароль, небезпеки піддаються всі інші дані, захищені тим же паролем. Рекомендується використовувати різні паролі для різних систем.

Не зберігайте пароль в Інтернеті. Зловмисник, який отримав доступ до вашого паролю в Інтернеті або в комп'ютерній мережі, отримує доступ до всіх даних.

Використання порожнього пароля

Порожній пароль (відсутність пароля) більш ефективний, ніж ненадійний, такий як, наприклад, "1234". Простий пароль легко розгадати. Порожній пароль для облікового запису на комп'ютері можна використовувати, якщо виконані перераховані нижче вимоги.

У вас один або кілька комп'ютерів, але вам не потрібен доступ з одного комп'ютера на інший.

Ваш комп'ютер фізично захищений (ви довіряєте всім, хто має доступ до вашого комп'ютера).

Не завжди рекомендується використовувати порожній пароль. Наприклад, переносний комп'ютер швидше за все фізично не захищений, і на ньому краще використовувати самий надійний пароль.

Облікові записи в Інтернеті

Веб-сайти мають різні політики, що регулюють доступ до облікового запису і зміна пароля. На домашній сторінці веб-сайту знайдіть посилання (наприклад "Рахунок"), що служить для переходу на сторінку веб-сайту, де виконується управління паролем і обліковим записом.

Паролі на комп'ютері

Відомості про створення і зміну облікових записів, захищених паролями, а також про доступ до них і про те, як встановити захист паролем при завантаженні комп'ютера, зазвичай маються на файлах довідки операційної системи. Можна також спробувати знайти ці відомості на веб-сайті виробника програмного забезпечення. Наприклад, в ОС Microsoft Windows відомості про управлінні паролями, їх зміну і т. д. можна знайти в системі інтерактивної довідки.

Зберігання паролів в секреті

Ставтеся до паролів та парольним фразам так само серйозно, як до даних, які вони захищають.

Нікому не повідомляйте пароль. Тримайте пароль в секреті від своїх близьких (особливо від дітей) та друзів, які можуть повідомити його кому-небудь ще. Винятком є паролі, які необхідно знати вашим близьким, наприклад пароль до вашого банківського рахунку в Інтернеті, який можна повідомити дружині або чоловікові.

Ніколи не пересилайте пароль по електронній пошті. Будь-яке повідомлення електронної пошти, що містить запит пароля або вимагає перейти на веб-сайт для підтвердження пароля, майже напевно є шахрайським. Це відноситься і до повідомлень такого типу, отриманим від надійної компанії або людини. Повідомлення електронної пошти може бути перехоплено, а відправник запиту може бути зовсім не тим, за кого себе видає. В фішинг-аферах використовуються шахрайські повідомлення електронної пошти, обманним шляхом змушують розкрити ім'я користувача та пароль і дозволяють зловмисникам заволодіти ідентифікаційними даними і т. п. Додаткові відомості про фішинг-аферах і про захист від шахрайства в Інтернеті.

Регулярно змінюйте паролі. Це допоможе ввести зловмисників в оману. Чим надійніше пароль, тим довше можна його використовувати.

Пароль з 8 знаків-і менше можна застосовувати протягом тижня, у той час як поєднання з 14 і більше знаків може служити кілька років, якщо воно складено за всіма правилами, наведеними вище.

Не вводьте паролі на чужих комп'ютерах. Комп'ютери в інтернет-кафе і лабораторіях, системи загального доступу, інтерактивні термінали, а також комп'ютери на конференціях і в залах очікування аеропортів не можуть вважатися безпечними і підходять тільки для анонімного виходу в Інтернет. Не користуйтеся такими комп'ютерами для перевірки електронної пошти, банківського рахунку, доступу в віртуальні кімнати для розмов і доступу до інших облікових записів, де запитується ім'я користувача та пароль. Зловмисники застосовують недорогі і швидко встановлюються пристрої, що записують послідовність натиснень на клавіші. Такі пристрої дозволяють шахраям отримувати через Інтернет всі дані, введені в комп'ютер. Пам'ятайте, що ваші паролі і парольні фрази так само важливі, як і дані, які вони захищають.

Дії у разі викрадення пароля

Відстежуйте всі дані, захищені паролями: фінансові звіти за місяць, звіти про кредитні операції, дані про покупки через Інтернет і т. д. Надійні та легко запам'ятовуючі паролі допомагають захиститися від шахрайства і розкрадання ідентифікаційних даних, але не є абсолютною гарантією захисту.

Незалежно від того, наскільки надійним є пароль, якщо шахраям вдасться зламати систему, де він зберігається, вони його дізнаються. Якщо ви помітили підозрілі дії, які можуть означати, що хтось отримав доступ до ваших даних, як можна швидше повідомте про це у відповідні органи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

(Всі кроки виконання занотуйте до звіту у вигляді скріншотів)

Завдання 1.

1. Запустіть браузер Інтернет (Це може бути Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera чи будь-який інший, що встановлений на вашому комп’ютері).
2. Відвідайте ресурс http://fortnotes.com/.
3. Створіть власний акаунт.
4. Проаналізуйте можливості даної служби.
5. Створіть кілька паролів електронної пошти, бази даних, сайтів.
6. Перегляньте створені записи.

Завдання 2.

1. Запустіть браузер Інтернет (Це може бути Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera чи будь-який інший, що встановлений на вашому комп'ютері).

2. Відвідайте ресурс http://websecurity.com.ua/password/.

3. Ознайомтесь з можливостями та призначенням ресурсу.

4. Введіть власні паролі в поле для діагностики рівня їх надійності. Запишіть в звіт результати.

5. Створіть 10 нових паролів (не користуючись відповідним сервісом) з різним рівнем надійності. Запишіть їх та висновки до звіту.

6. Скористуйтеся сервісом створення паролю. Створіть таким чином та запишіть до звіту 7 паролів.

Завдання 3.

1. Завантажить Advanced Office Password Recovery (або інше ПЗ для аудиту парольного захисту) та встановимо його.

Програми підбору паролів можливо скачати зі спеціалізованих сайтів, наприклад, таких як:

− http://www.passwords.com,

− http://www.elcomsoft.com,

− http://lastbit.com/mso.

2. Створить документ Word і Excel, стандартними засобами ПЗ встановити на ці документи пароль.

3. Запустіть Advanced Archive Password Recovery Professional і спробуйте підібрати пароль методом суцільного перебору. Запам'ятайте час, яке програма витратила для цього.

4. Дослідить залежність часу підбору від довжини пароля. Для цього збільшуйте його довжину, не змінюючи категорії (тобто використовуйте тільки цифри) і кожен раз визначайте час, витрачений програмою для його підбору. Побудуйте графік залежності часу зламу від довжини пароля для однієї категорії. Зробіть висновок про доцільність використання однієї категорії для створення паролів.

5. Поставте пароль з використанням трьох з чотирьох доступних категорій. Для початку виберіть короткий пароль, наприклад, 3-4 символу. Поступово збільшуючи довжину пароля, дізнайтеся час, який витрачає зломник на подолання встановленого пароля. Побудуйте графік отриманої залежності.

5. Задайте пароль українським словом, набраними англійською розкладкою, наприклад, слово «пароль» буде виглядати як «gfhjkm» і досліджуйте час, необхідний для його підбору.

6. Порівняйте отримані результати та зробіть висновок про оптимальної стійкості досліджених паролів для заданих обчислювальних ресурсів нападника.

7. Зробить архівний документ та встановити на нього пароль.

8. Повторіть все дослідження (пп.3-6), використовуючи замість повного перебору паролів атаку по словнику. Словники можна звантажити з Інтернету або скласти самостійно. Як змінилися Ваші результати?

9. Побудуйте графік залежності стійкості паролю від часу взлому.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що називається паролем?
2. Яке програмне забезпечення називається менеджером паролів?
3. На які категорії діляться менеджери паролів.
4. Який принцип роботи менеджерів паролів?
5. Що називається онлайн менеджером паролів?
6. Назвіть основні переваги та недоліки онлайн менеджерів паролів.
7. Що являють собою менеджери паролів з бар’єрним захистом?
8. Назвіть правила створення та користування паролями.
9. Назвіть етапи створення надійного паролю.
10. Які програми управління паролями ви знаєте? Опишіть їх основні можливості.
11. Яке властивість інформації захищають за допомогою паролів?
12. Де переважно використовують парольний захист?
13. Які вимоги пред'являються до пральний захисту?
14. Чи підтверджуються ці вимоги результатом Ваших досліджень?
15. Які Ви знаєте способи захисту від атаки повного перебору паролів і атаки по словнику? Чому Ви вважаєте їх ефективними? Які з них використовують в операційних системах і другом програмному забезпеченні?
16. Які Ви можете запропонувати способи вдосконалення парольний захисту?

Лабораторна робота №4

Тема:. Засоби аутентифікації користувачів і аналізу безпеки системи

Мета: Вивчення засобів аналізу безпеки систем, роботи методів аутентифікації, ознайомлення з методами і завданнями підсистем управління доступом

Час проведення: - 4 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Засоби аналізу захищеності

Засоби аналізу захищеності (ЗАЗ), або сканери безпеки, призначені для автоматизації роботи адміністратора безпеки шляхом пошуку потенційних порушень політики безпеки та вразливостей ОС. При цьому ЗАЗ можуть виявляти такі види порушень (вразливості):

- “люки” (back door) в програмах и програми типу “троянський кінь”;
- слабкі паролі й неправильні налаштовування механізмів автентифікації;
- сприйнятливість до проникнення внаслідок неявних довірчих відношень між системами;
- сприйнятливість до атак на відмову в обслуговуванні;
- неправильні налаштовування міжмережевих екранів, мережевих і прикладних сервісів;
- нездатність засобів захисту системи адекватно реагувати на спроби збору інформації;

ЗАЗ можуть здійснювати сканування системи як “ззовні”, з використанням для доступу до системи мережевих засобів (network-based), так і “зсередини”, працюючи безпосередньо на хості, який аналізують (host-based, application-based). Як правило, мережеві сканери також здатні здійснювати сканування локальної системи через “зворотний” інтерфейс (127.0.0.1).

Для мережевих ЗАЗ існують дві групи способів виявлення вразливостей – сканування й зондування.

Сканування (banner check) – механізм пасивного аналізу, використовуючи який сканер намагається визначити існування вразливості за непрямими ознаками без фактичного підтвердження її наявності. Цей метод є найшвидшим і найпростішим для реалізації. Цей процес ідентифікує відкриті порти, що знайдені на кожному мережевому пристрої, і збирає пов’язані з портами

заголовки (banner), знайдені під час сканування кожного порту. Аналіз заголовків дозволяє ідентифікувати операційну систему й використовувані сервіси аж до конкретної версії. На підставі апріорних знань про наявність вразливостей в тій чи іншій версії ПЗ робиться висновок про наявність або відсутність вразливості в системі, яку аналізують[15].

Варто зазначити, що певна функціональність щодо аналізу заголовків включається навіть в ПЗ, яке призначене лише для сканування мереж. ПЗ, що спеціально призначене для пошуку вразливостей, обов'язково має базу даних щодо вразливостей, і механізми її оновлення.

Зондування (active check) – механізм активного аналізу, який дозволяє впевнитись, чи присутня вразливість на вузлі, що аналізують. Зондування виконується шляхом імітування атаки, що використовує вразливість, яку перевіряють. Цей метод повільніший, чим сканування, але майже завжди значно точніший. Цей процес використовує інформацію, що була отримана в процесі сканування, для детального аналізу кожного мережевого пристрою.

Наприклад, в ході сканування можна отримати відомості про відкриті TCP порти №№ 135, 139. Це ознака використання в мережі служби NetBIOS (майже напевно, на системі, яку сканують, встановлено операційну систему від Microsoft). При цьому можна одразу підозрювати численні вразливості, притаманні системам Windows і протоколу NetBIOS. Далі логічною є взаємодія з системою за протоколом NetBIOS для вивчення наявності загальнодоступних (shared) ресурсів, захищеності цих ресурсів паролем. При цьому можна отримати відомості про користувачів системи, наявність можливості адміністрування системи через мережу, тощо. За цими даними можна отримати детальні й достатньо достовірні відомості про наявність вразливостей.

Також під час зондування можуть використовуватись відомі методи реалізації атак для того, щоб остаточно підтвердити або спростувати наявність тих вразливостей, які припускаються за результатами сканування, а також виявити інші вразливості, які не можуть бути виявлені пасивними методами, як, наприклад, нестійкість до атак типу "відмова в обслуговуванні" ("denial of service").

Необхідною складовою сканера безпеки є система підготовки звітів. Оскільки основним призначенням цього ПЗ є надання інформації системним адміністраторам, доброю ознакою якості програми є її здатність не лише виявляти вразливості, але й надавати рекомендації з їх усунення.

Одним з перших мережевих сканерів був SATAN – System Administrator Tool for Analysis of the Network. В часи його появи (перша половина 90-х)

ситуація з безпекою в Internet була просто катастрофічна (достатньо пригадати успішне розповсюдження “віруса Морріса”, який автоматично, без втручання користувачів, за одну ніч спромігся здійснити проникнення на тисячі хостів). Тому поява SATAN – засобу, доступного для використання усіма зацікавленими, спричинила величезний галас. Зрештою, переміг здоровий глузд

– неможливо заборонити розробку і розповсюдження подібних засобів через погану захищеність хостів, навпаки, слід підтримувати достатній рівень захищеності хостів і для контролю цього періодично застосовувати сканери безпеки.

Ніколи не можна сподіватись, що ваша система не буде знайдена зловмисниками, не буде для них цікавою, або вони не будуть достатньо оснащені або компетентні. Як правило, все відбувається навпаки. Зловмисників багато, вони добре оснащені, серед них є дуже талановиті і досвідчені. Успіх адміністратора безпеки ніколи не буває остаточним, оскільки успішними повинні бути всі без виключення його дії протягом всього часу роботи системи. Але система може бути скомпрометована внаслідок лише однієї з дуже багатьох спроб лише одного з дуже багатьох порушників, навіть якщо всі інші спроби виявились невдалими. Результат – успіх зловмисника. Тому слід розуміти – адміністратор безпеки повинен постійно випробувати свою систему “на міцність” (звичайно ж, з дотриманням мір обачності). Він повинен бути на крок попереду зловмисників, користуючись своєю обізнаністю щодо особливостей системи.

Пошук вразливостей є ще більш загрозливою активністю в мережі, ніж сканування портів. Якщо його проводить неуповноважена або некомпетентна особа, можуть виникнути наслідки, такі як відмова систем або їх компонентів, втрата або псування інформаційних ресурсів, розголошення конфіденційної інформації. Але ті ж дії у виконанні грамотних адміністраторів є невід’ємною складовою мережевої безпеки.

Сканери безпеки бувають комерційні і вільні (безкоштовні). Мабуть, найвідоміший з комерційних сканерів – Internet Scanner, продукт компанії Internet Security Systems [16].

З некомерційних сканерів слід відзначити сканер nessus, який складається з ядра, що працює під керуванням Unix, та клієнтської частини, яка існує для різних систем. Ядро здійснює сканування, використовуючи базу даних по вразливостям. Клієнтська частина реалізує інтерфейс користувача. Сканер розповсюджується безкоштовно, з відкритим кодом.

Дуже широкого поширення набула програма російських розробників

XSpider. Хоча її бази даних були розраховані на пошук вразливостей переважно в системах Windows (для інших платформ програма також надавала певний обсяг інформації, але він поступався і детальністю, і точністю), програма мала зручний інтерфейс, надавала детальні звіти, а при застосуванні імітації атак ефективно використовувала вразливості, в тому числі підбирала слабкі паролі. На жаль, успіх програми дозволив розробникам перевести її на комерційну основу, остання безкоштовна версія датована груднем 2002 року.

Також слід пам’ятати, що для платформи Windows існує безкоштовний сканер безпеки від розробників – Microsoft Baseline Security Analyzer (MBSA). MBSA – вільно розповсюджуваний програмний продукт, засіб аналізу захищеності операційних систем Windows та ряду програмних продуктів компанії, таких як Internet Information Services, SQL Server, Internet Explorer та ін. Термін “Baseline” у назві слід розуміти як деякий рівень, при якому безпеку ОС можна вважати задовільною. MBSA дозволяє сканувати комп’ютери під керуванням ОС Windows на предмет знаходження основних вразливостей та наявності рекомендованих оновлень системи безпеки. MBSA виконує цю перевірку, звертаючись до бази даних Windows, яка містить потрібну інформацію.

Його застосування вимагає повноважень адміністратора, і хоча ніхто не знає вади Windows краще за їх творців з Microsoft, деякі вразливості цей сканер може не помічати через те, що розробники не бажають їх такими визнавати. З іншого боку, цей продукт дуже зручний для оцінювання поточного стану системи, особливо стосовно встановлення необхідних оновлень і виправлень.

Для виконання лабораторної роботи можна використовувати сканер XSpider. Деяка інформація про порядок роботи з програмою наведена в Додатку.

Використання сканера безпеки – досить критична операція, оскільки крім сканування він здатний здійснювати також і зондування цілі, а це вже фактично є здійсненням атаки. Тому в жодному разі не використовуйте сканери безпеки у мережах, в яких ви не є адміністратором, а використання сканера не погоджено з відповідальними особами (не слід забувати, що адміністратор фактично є технічним обслуговуючим персоналом, а не безроздільним господарем мережі, тому навіть адміністратор не повинен здійснювати сканування будь-коли за власним розсудом).

Аутентифікація користувачів на основі токенів безпеки

Правильне функціонування підсистеми безпеки комп‘ютерної системи вимагає реалізації ряду функцій загального призначення, пов‘язаних з

перетворенням вмісту об‘єктів системи (файлів, записів бази даних тощо) або з обчислення деяких спеціальних функцій, які суттєво залежать від вмісту об‘єктів. До таких функцій належать алгоритми контролю цілісності об‘єктів, аутентифікації та авторизації об‘єктів, що керують процесами, а також алгоритми підтримання конфіденційності інформації, що міститься в об‘єктах комп‘ютерної системи.

Міжнародні та національні стандарти описують ряд добре відомих та вивчених функцій захисного характеру, зокрема алгоритми хешування MD5, MD2, SHA тощо; алгоритми генерування та перевірки електронного цифрового підпису RSA, DSS та інших. Усі ці алгоритми мають різні механізми викликів (зокрема, різну довжину аргументів). Це, у свою чергу, означає, що вони несумісні між собою.

Тому задача вбудовування тих чи інших захисних механізмів в операційну систему на основі якогось одного алгоритму буде виглядати неефективною, особливо, якщо ця ОС розповсюджується в різних регіонах земної кулі. В цьому випадку логічним є побудова “шаруватої” структури, де окремий шар, реалізований, скажемо, як набір динамічних бібліотек, відповідає за захист інформації. Цей спосіб досить універсальний і широко застосовується у сімействі операційних систем Windows. Таким способом можна розв‘язати великий клас задач, пов‘язаних з універсалізацією ОС: від національних налаштувань системи до реалізації різноманітних засобів безпеки.

Зрозуміло, що такі структури повинні мати т. зв. “відкритий інтерфейс”, тобто бути детально документованими для того, щоби програмісти могли використати засоби цієї структури при створенні прикладного програмного забезпечення, в тому числі і для захисту інформації.

Сьогодні є достатня кількість криптографічних інтерфейсів, однак найбільшої популярності набув інтерфейс від Microsoft – Microsoft CryptoAPI.

Зараз використовується CryptoAPI версії 2.0. Причина популярності цього інтерфейсу полягає в тому, що Microsoft інтенсивно впровадила захисні механізми CryptoAPI у свої операційні системи та прикладне програмне забезпечення. Сучасні ОС сімейства Windows містять багато криптографічних підсистем різного призначення як прикладного рівня, так і рівня ядра. Провідну роль в цьому грають якраз функції CryptoAPI, зокрема базові криптографічні функції, сукупність яких створює інтерфейс CryptoAPI 1.0.

Інтерфейс CryptoAPI 2.0 містить як базові криптографічні функції, так і функції, що реалізують перетворення вищого рівня – роботу з сертифікатами X.509, обробку криптографічних повідомлень PKCS#7 та інші функції, що

підтримують інфраструктуру відкритих ключів. Однак набір базових криптографічних функцій цього інтерфейсу утворює CryptoAPI 1.0. Таким чином, функції CryptoAPI 1.0 утворюють криптографічне ядро прикладного рівня для сучасних операційних систем лінійки Windows.

Загальну архітектуру CryptoAPI 1.0 подано на рис. 4.1.

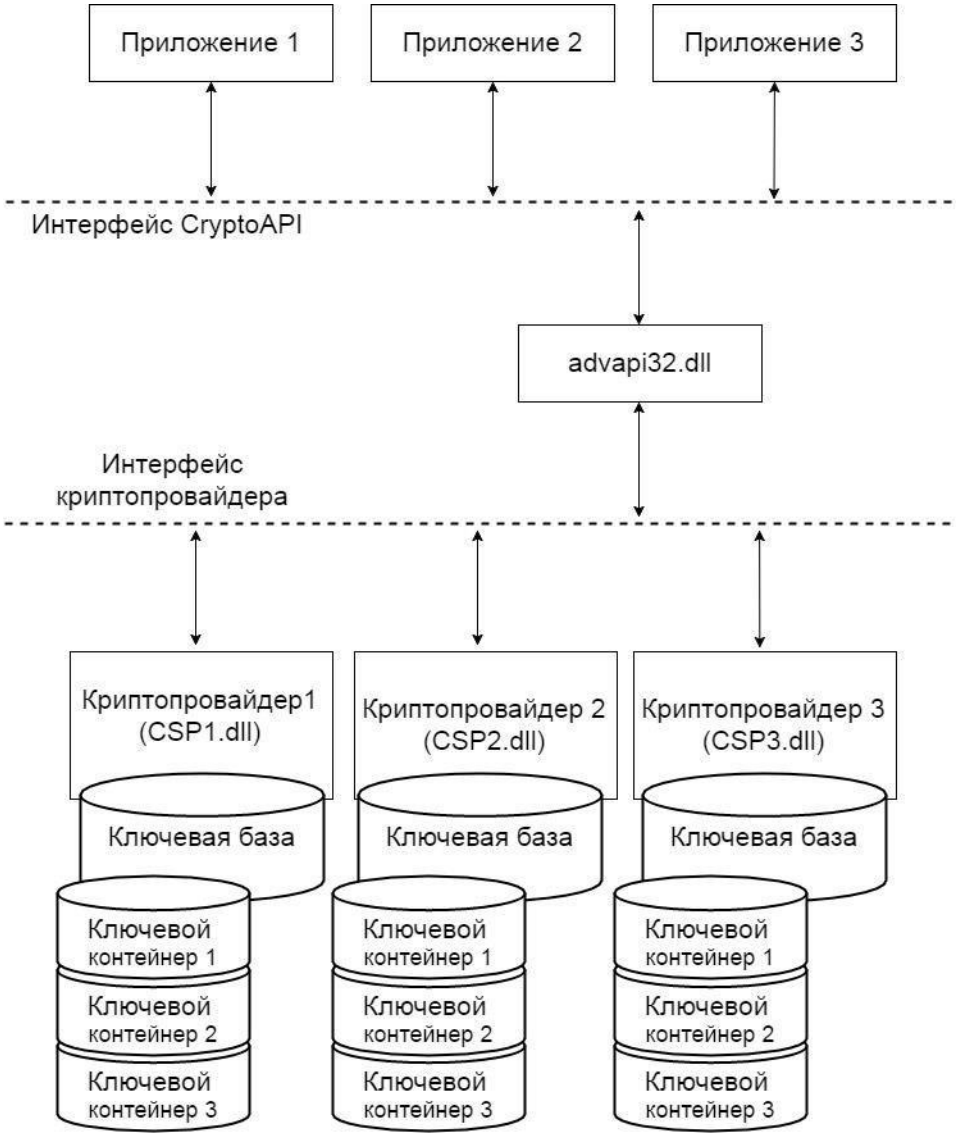

Рис. 4.1. Загальна архітектура CryptoAPI

Будь який криптопровайдер повинен експортувати набір обов'язкових функцій, які формують системний програмний інтерфейс CryptoAPI, при цьому кожна з цих функцій відповідає деякої функції CryptoAPI. Також криптопровайдер повинен забезпечувати реалізацію стандартного інтерфейсу криптопровайдера, роботу з ключами шифрування, призначеними для забезпечення роботи алгоритмів, специфічних для даного криптопровайдера, неможливість втручання третіх осіб в схеми роботи алгоритмів.

Додатки не працюють безпосередньо з криптопровайдером. Замість цього вони викликають функції CryptoAPI з бібліотек Advapi32.dll і Crypt32.dll. Операційна система фільтрує виклики цих функцій і викликає відповідні функції CryptoAPI, які безпосередньо працюють з криптопровайдером.

Мінімальний склад криптопровайдера – одна DLL. Зазвичай ця бібліотека зберігається в папці \ WINDOWS \ system32 \. Обов'язковою є контроль цілісності цієї DLL.

Крім стандартних функцій CryptoAPI, криптопровайдер зазвичай підтримує ряд власних функцій. Якщо власні функції не реалізовані, то DLL діє, по суті, як проміжний шар між операційною системою і виконавцем криптографічних операцій.

Функції криптопровайдерів представлені в таблиці 4.1.

Таблиця 4.1

Функції криптопровайдерів

Функція	Короткий опис
1	2
CryptAcquireContext	Використовується для створення дескриптора ключового контейнера у рамках визначеного криптопровайдера (КП).
CryptContextAddRef	Збільшує на одиницю лічильник посилань на дескриптор КП.
CryptEnumProviders	Використовується для отримання першого та наступних доступних КП.
CryptEnumProviderTypes	Використовується для отримання першого та наступних доступних типів КП.
CryptGetDefaultProvider	Повертає КП, встановлений за замовчуванням для вказаного типу КП.
CryptGetProvParam	Повертає параметри КП
CryptReleaseContext	Вивільняє дескриптор КП
CryptSetProvider CryptSetProviderEx	Задає тип та назву КП за замовчуванням
CryptSetProvParam	Встановлює параметри КП
CryptDeriveKey	Створює сесійні криптографічні ключі з ключового матеріалу
CryptDestroyKey	Звільняє дескриптор ключа
CryptDuplicateKey	Робить копію криптографічного ключа
CryptExportKey	Експортує криптографічні ключі із заданого контейнера
CryptGenKey	Генерує випадкові криптографічні ключі та ключові пари
CryptGenRandom	Генерує випадкову послідовність та зберігає її в буфері
CryptGetKeyParam	Повертає параметри ключа

CryptImportKey	Імпортує криптографічні ключі з ключового блоба у контейнер КП
CryptSetKeyParam	Встановлює параметри ключа
CryptDecrypt	Виконує операцію розшифрування даних
CryptEncrypt	Виконує операцію за шифрування даних
CryptCreateHash	Створює хешований потік даних
CryptDestroyHash	Знищує об‘єкт хеш функції
CryptDuplicateHash	Створює точну копію хеш-об‘єкта
CryptGetHashParam	Повертає параметри хеш-об‘єкта
CryptHashData	Додає дані до хеш-об‘єкта
CryptHashSessionKey	Підмішує до хеш-об‘єкта сесійний ключ
CryptSetHashParam	Встановлює параметри хеш-об‘єкту
CryptSignHash	Обчислює значення ЕЦП від значення хешу
CryptVerifySignature	Перевіряє ЕЦП заданого значення хешу

Як бачимо з таблиці, CryptoAPI 2.1 підтримує усі основні методи криптографічного перетворення даних: від генерування криптографічних послідовностей випадкових чисел до операцій з електронним цифровим підписом. Таким чином, знаючи інтерфейс CryptoAPI 2.1, програміст може досить легко реалізувати усі популярні криптографічні алгоритми у своїх прикладних програмах.

Програміст, який працює з цим інтерфейсом, може отримати усю необхідну інформацію про певного криптопровайдера засобами функції CryptGetProvParam. Перше, що необхідно знати при цьому – це набір криптографічних стандартів, які реалізують встановлені у системі крипто-провайдери.

Окрім різниці у стандартах, криптопровайдери відрізняються способом фізичної організації збереження ключової інформації. З точки зору програмування спосіб зберігання ключів значення не має, однак він дуже важливий з точки зору експлуатації та безпеки комп‘ютерної системи. Існуючі криптопровайдери Microsoft зберігають ключову інформацію на жорсткому диску (у реєстрі або у файлах), а провайдери інших фірм (GemPlus, Schlumberger та Infineon) – на смарт-картках.

Якщо способи фізичної організації збереження ключової інформації у криптопровайдерів відрізняється, то логічна структура, яка визначається

інтерфейсами та з якою мають справу програмісти, однакова для будь-якого типу провайдера. Ключова база визначається набором ключових контейнерів, кожен з яких має ім‘я, що привласнюється йому при створенні, а потім використовується для роботи з ним. У ключовому контейнері зберігається довготривала ключова інформація, наприклад, ключові пари для цифрового підпису або несиметричної системи шифрування.

Тепер розглянемо детально, як функції інтерфейсу CryptoAPI викликають бібліотеки конкретного криптопровайдера. Кожен криптопровайдер має своє власне ім‘я та тип. Його ім‘я – просто рядок, за допомогою якого система його ідентифікує. Так, базовий криптопровайдер Microsoft має назву Microsoft Base Cryptographic Provider v1.0. Тип криптопровайдера – ціле число (у нотації C – DWORD), значення якого ідентифікує набір криптографічних алгоритмів, що підтримуються. Криптопровайдер Microsoft має тип 1, цей тип провайдера реалізує в якості алгоритмів цифрового підпису та обміну ключів алгоритм RSA. Інший базовий криптопровайдер Microsoft, „Microsoft Base DSS and Diffie-Hellman Cryptographic Provider”, має тип 13. Цей тип криптопровайдера реалізує алгоритм цифрового підпису DSS, а в якості алгоритму обміну ключами – протокол Діффі-Хелмана.

Отже, для роботи з набором криптопровайдерами у системному реєстрі міститься список імен усіх криптопровайдерів. З кожним ім‘ям пов‘язаний тип криптопровайдера та ім‘я бібліотеки, яка реалізує його алгоритми.

Окрім цього в системі міститься інформація про те, який криптопровайдер треба застосовувати, якщо користувач явно не вказав конкретне його ім‘я, лише визначивши тип провайдера. Такий криптопровайдер називають провайдером за замовчуванням для заданого типу. Наприклад, для типу 1 провайдером за замовчуванням є Microsoft Base Cryptographic Provider v1.0, а для типу 13 – Microsoft Base DSS and Diffie-Hellman Cryptographic Provider. Для визначення криптопровайдерів за замовчуванням використовують функцію CryptGetDefaultProvider, а для зміни цього параметру – функції CryptSetProvider або CryptSetProviderEx. Функції дозволяють встановити провайдера за замовчуванням як для поточного користувача, так і для системи в цілому (усіх користувачів). Ці параметри зберігаються у вулику реєстру HKEY_LOCAL_MACHINE. Параметри, встановлені для поточного користувача, мають пріоритет над параметрами, встановленими для усієї системи, та зберігаються у вулику реєстру HKEY_CURRENT_USER. Якщо параметри для поточного користувача відсутні, застосовуються загальносистемні.

Тепер розглянемо, яким чином користувач починає працювати з конкретним криптопровайдером, і як система викликає конкретну бібліотеку, що відповідає обраному криптопровайдеру.

Робота з певним провайдером починається з виклику функції CryptAcquireContext, де користувач визначає тип потрібного криптопровайдера, його назву та назву робочого ключового контейнера. В результаті роботи функція повертає користувачу дескриптор криптопровайдера (handle), за допомогою якого користувач в подальшому буде звертатися до нього та передавати його у процедури для виконання усіх необхідних криптографічних операцій.

Детальний опис контексту роботи з криптопровайдерами та приклади (мовою програмування С) дивіться у книжці Щербакова Л.Ю., Домашева А.В. “Прикладная криптография”.

Власне бібліотеки CryptoAPI разом з файлами заголовків та допомоги постачаються у складі бібліотек MSDN.

Відомості про способи аутентифікації.

Однією з основних функцій систем захисту від несанкціонованого доступу є ідентифікація та аутентифікація. Вона полягає в тому, що жоден суб’єкт (сутність обчислювальної системи, здатна ініціювати виконання операцій) не може отримати доступ до об’єктів (сутностей обчислювальної системи, що захищаються) без надання системі захисту певного обсягу інформації про себе.

При цьому ідентифікація суб’єкта полягає в тому, що суб’єкт повідомляє системі захисту свій унікальний ідентифікатор в обчислювальній системі; аутентифікація суб'єкта полягає в тому, що суб’єкт надає системі захисту окрім ідентифікуючої інформації ще й певну інформацію, за допомогою якої система перевіряє, що він дійсно є тим суб’єктом, якого стосується ідентифікуюча інформація; авторизація суб’єкта відбувається після вдалих ідентифікації та аутентифікації і полягає в тому, що обчислювальна система виконує дії, необхідні для того, щоб суб’єкт мав можливість почати роботу.

Таким чином, щоб отримати доступ в обчислювальну систему, користувач має спочатку ідентифікувати себе, а механізми захисту, в свою чергу, мають підтвердити істинність користувача, тобто підтвердити, що він дійсно є тим, кого з себе удає. Існує три групи способів підтвердження істинності користувача. Відповідно, для кожної групи механізми підсистеми ідентифікації та аутентифікації мають перевірити:

- щось, що користувач знає (паролі, ідентифікаційні коди, інші

відомості);

- щось, що користувач має (ключі, магнітні чи смарт-картки і т.п.);ось, чим користувач є (особисті характеристики користувача: відбитки пальців, малюнок сітківки ока, характеристики голосу, особливості користування клавіатурою та маніпуляторами).

Далі розглядатимуться способи, що належать до першої групи, як найбільш поширені.

Якщо перевіряється істинність тільки користувача, то таку процедуру називають одностороннім (peer-entity) підтвердженням істинності. В іншому випадку, тобто коли користувач має підтвердити свою істинність системі, а система, в свою чергу, має підтвердити свою істинність користувачеві, така процедура носить назву двосторонньої (peer-to-peer) аутентифікації.

В разі використання аутентифікації за простим паролем кожен користувач обчислювальної системи отримує пару значень – ідентифікатор (ім'я в системі) та пароль. Користувач отримує доступ, якщо вказаний ним в процесі входу в систему ідентифікатор є зареєстрованим, а відповідний пароль – вірним.

Така схема вразлива щодо втрати або розголошення пароля, внаслідок чого одні користувачі можуть видавати себе за інших, тим самим здійснюючи несанкціонований доступ.

Іншим способом є аутентифікація на основі списку паролів. При цьому користувачеві разом з ідентифікатором надається список паролів. Перший пароль використовується при першому вході систему, другий – при другому і т. д. Незважаючи на те, що така схема є більш стійкою до втрати окремих паролів, вона має суттєві недоліки, а саме:

- користувачеві незручно запам'ятовувати список паролів;
- у випадку помилки або збою при аутентифікації користувач не знає, користуватись йому поточним чи наступним паролем.

Ще одним способом аутентифікації є метод одноразових паролів.

Під час реєстрації користувач генерує певну послідовність, наприклад:

F999(x), …, F(F(F(x))), F(F(x)), F(x), x,

де *x* – випадкове число.

При цьому в системі в цей час зберігається значення *F1000(x)*, на першому кроці в якості паролю користувач використовує значення *F999(x)*. Отримавши його, система обчислює *F(F999(x))* та перевіряє його на відповідність тому *F1000(x)*, що зберігається. В разі відповідності користувач отримує доступ до системи, а в системі в якості поточного зберігається вже значення *F999(x)*. На другому кроці перевіряється *F(F998(x)) = F999(x)* і так далі. Таким чином, пароль, що вже був використаний, а також всі інші, що знаходяться у списку перед ним, стають недійсними. При цьому у випадку порушення синхронізації користувач має можливість перейти до наступного в списку значення, або навіть “перескочити” через один чи кілька паролів, а система вираховує *F(F(…Fn(x)…))* поки не отримає значення, відповідне тому, що зберігається.

Перевірити істинність користувача також можна за допомогою методу рукостискання (handshake). При цьому існує процедура *f*, що відома лише користувачеві та обчислювальній системі. При вході в систему генерується випадкове значення x і обчислюється *f(x)*. Користувач, отримавши x, також обчислює *y = f(x)* та надсилає його системі. Система порівнює власне значення з отриманим від користувача і робить висновок про його (користувача) істинність. При використанні методу рукостискання ніякої конфіденційної інформації між користувачем і обчислювальною системою не передається взагалі, навіть у шифрованому вигляді. Щодо самої функції *f(x)*, то вона має бути досить складною, щоб зловмисник не міг її вгадати, навіть накопичивши велику кількість пар *(x, f(x))*. В якості процедури *f(x)* можна використовувати шифрування x на таємному ключі, який є спільним секретом (або шифрування таємного "магічного рядка" на ключі x).

Підсистема керування доступом

Задачею підсистеми керування доступом в системах захисту інформації є виконання політики безпеки як певного набору правил розмежування доступу (ПРД).

Згідно матричної моделі безпеки, запропонованої Д. Деннінг, обчислювальну систему з точки зору безпеки можна описати у вигляді кортежу

<*S,O,A*>, де S-множина суб'єктів системи, O-множина об'єктів системи, A- матриця доступу. Суб'єкти *S* є активними сутностями, здебільшого це користувачі або процеси. Об'єкти O є пасивними сутностями, тобто такими, що потребують захисту. Це можуть бути, наприклад, файли, записи баз даних, сегменти оперативної пам'яті. У деяких операціях доступу суб'єкти можуть виступати як пасивні сутності, до яких здійснюють доступ інші суб'єкти, тому $S \subset O$. У матриці доступу A кожен рядок відповідає певному суб'єктові S_i, а кожен стовпчик – об'єктові O_i. Елементом матриці $A(S_i,O_i)$ є список прав доступу, або повноважень суб'єкта S_i стосовно об'єкта O_i. Ці права, власне, і визначають, що може робити суб'єкт з об'єктом.

Оскільки матриця доступу, як правило дуже розріджена (і, отже, неефективна з точки зору використання пам'яті), вона практично не використовується в реальних системах у повному вигляді. Замість того використовуються її наявні представлення, а саме списки доступу та списки повноважень. Список доступу асоціюється з кожним захищеним об'єктом в системі і містить в собі ідентифікатори різних суб'єктів разом з їх правами доступу до даного об'єкту (список доступу, таким чином, відповідає стовпчику матриці доступу). На відміну від списку доступу, список повноважень асоціюється з кожним суб'єктом в системі і містить в собі ідентифікатори об'єктів разом з повноваженнями цього суб'єкта стосовно цих об'єктів. Список повноважень, таким чином, відповідає рядкові матриці доступу.

При використанні матричної моделі безпеки політика безпеки інформації набуває вигляду обмежень, що накладаються на спосіб модифікації матриці доступу. Так, у випадку довірчого управління доступом (згідно НД ТЗІ 1.1-003-99) всі права на зміну прав доступу до об'єкту надаються суб'єктові, що є

власником цього об'єкту. Тобто, якщо список прав доступу суб'єкта S_i до об'єкта O_i містить право власника, то суб'єкт S_i отримує повний контроль над стовпчиком матриці доступу, що відповідає O_i.

У випадку адміністративного управління доступом (НД ТЗІ 1.1-003-99) система захисту сама визначає можливість доступу суб'єктів до об'єктів, базуючись на певних мітках або атрибутах доступу, які може встановлювати або змінювати лише спеціально призначений адміністратор. Так, наприклад, в класичній моделі Белла-ЛаПадула такими атрибутами є рівень конфіденційності L та категорія *C*. При цьому мають виконуватись два правила: просте правило безпеки та *-правило. Просте правило безпеки встановлює, що суб'єкт *S* може читати об'єкт *O* тоді і тільки тоді, коли рівень конфіденційності $l_S \geq l_O$ та категорія $c_S \subseteq c_O$. * правило встановлює, що суб'єкт *S* може писати в об'єкт O тоді і тільки тоді, коли рівень конфіденційності $l_S \leq l_O$ та категорія $c_O \subseteq c_S$.

Подальшим розвитком моделі Белла-ЛаПадула є модель ватерлінії (Low Water Mark, LWM). В цій моделі атрибути доступу об'єктів та суб'єктів можуть змінюватись у процесі роботи системи. Так, якщо суб'єкт *S* читає об'єкт *O*, рівень конфіденційності якого $l_S < l_O$, то рівень конфіденційності суб'єкта підвищується, так що $l_S = l_O$. І навпаки, якщо суб'єкт пише в об'єкт, коли $l_S > l_O$, то рівень конфіденційності об'єкта підвищується таким чином, що $l_S = l_O$.

Моделі Белла-ЛаПадула та ватерлінії сконцентровані на питаннях захисту обчислювальних систем від загрози конфіденційності інформації.

Інша група моделей безпеки (адміністративного управління доступом) розглядає питання захисту обчислювальних систем від загроз цілісності інформації. Так, в моделі Біба існують рівні цілісності *I* та категорії *C*. при цьому доступ суб'єкту до об'єкту на читання можливий тоді, коли $i_S \leq i_O$ та $c_S \subseteq c_O$. Доступ на запис, в свою чергу, можливий тоді і лише тоді, коли $i_S \geq i_O$ та $c_O \subseteq c_S$. Ці два правила, по аналогії з моделлю Белла-ЛаПадула, носять назву просте правило цілісності та *-правило цілісності. Існують також моделі Біба з пониженням рівня цілісності об'єкта та Біба з пониженням цілісності суб'єкта. В першій після операції запису суб'єктом *S* в об'єкт *O* рівень цілісності об'єкта падає до рівня цілісності суб'єкта $(i_S = i_O)$. В другій внаслідок операції читання рівень цілісності падає до рівня цілісності прочитаного об'єкта $(i_S = i_O)$.

Композитна модель адміністративного управління доступом об'єднує в собі модель конфіденційності Белла-ЛаПадула та модель цілісності Біба.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

(Всі кроки виконання занотуйте до звіту у вигляді скріншотів)

Завдання 1.

1. Встановить та запустити програму Microsoft Baseline Security Analyzer (MBSA).
2. Проскануйте локальний комп’ютер.

3. В результаті сканування отримайте звіт, де наведено знайдені вразливості безпеки.

4. Збережіть результати у PDF-форматі.

Завдання 2.

1. Встановіть на комп'ютері або у віртуальну машину діагностичне програмне забезпечення Microsoft Baseline Security Analyzer.

2. Проскануйте локальний комп'ютер за допомогою кнопки «сканування».

3. Збережіть результати у PDF-форматі.

4. Запустити сканування на localhost та зберегти репорт у форматі html

5. Перегляньте створені записи. Зробить висновок.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Назвіть типи засобів аналізу захищеності.
2. Назвіть види вразливостей, виявлених засобами аналізу захищеності
3. Назвіть способи виявлення вразливостей.
4. Охарактеризуйте етапи роботи сканерів безпеки.
5. Порівняйте можливості використаних в цій лабораторній роботі сканерів безпеки. Вкажіть їх переваги та недоліки, а також визначте дії цих програмних продуктів.
6. Дайте визначення процесів ідентифікації, аутентифікації і авторизації.
7. Які способи підтвердження особи користувача існують? Чим вони відрізняються? Охарактеризуйте їх переваги та недоліки.
8. Де використовуються маркери безпеки? Які типи токенів Ви знаєте?
9. Охарактеризуйте призначення і можливості інтерфейсу CryptoAPI.
10. Охарактеризуйте основні способи аутентифікації, розглянуті в цій лабораторній роботі. Назвіть основні області їх застосування, переваги і недоліки
11. Опишіть криптографічні функції з набору CryptoAPI, які Ви використовували в Вашій програмі.
12. Проаналізуйте недоліки довірчого управління доступом.

Лабораторна робота №5

Тема: Шкідливе програмне забезпечення. Основні типи та загальний огляд комп’ютерних вірусів.

Мета: Отримати навички виявлення на комп'ютері шкідливих програм, вивчити основні методи по усуненню наслідків вірусних атак без використання антивірусного програмного забезпечення.

Час проведення: - 8 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Вірус - програма, здатна створювати свої копії (необов'язково співпадаючі з оригіналом) і впроваджувати їх у файли, системні області комп'ютера, комп'ютерних мереж, а також здійснювати інші деструктивні дії. При цьому копії зберігають здатність подальшого поширення. Комп'ютерний вірус відноситься до шкідливим програмам.

Ще одна проблема, пов'язана з визначенням комп'ютерного вірусу криється в тому, що сьогодні під вірусом найчастіше розуміється не "традиційний" вірус, а практично будь-яка шкідлива програма.

Це призводить до плутанини в термінології, ускладненої ще й тим, що сьогодні практично будь-який антивірус здатний виявляти всі типи шкідливих програм, таким чином асоціація "шкідлива програма-вірус" стає все більш стійкою.

Шкідлива програма - комп'ютерна програма або переносний код, призначений для реалізації загроз інформації, що зберігається в комп'ютерній системі (КС), або для прихованого нецільового використання ресурсів КС, або іншого впливу, що перешкоджає нормальному функціонуванню КС. До шкідливих програмам відносяться комп'ютерні віруси, трояни, мережні хробаки і інші.

Віруси

До вірусів відносяться:

– ***завантажувальні віруси*** - віруси, що заражають завантажувальні сектори постійних і змінних носіїв.;

– ***файлові віруси*** - віруси, що заражають файли;

– ***макровіруси*** - віруси, написані мовою макрокоманд і виконують у середовищі якого-небудь додатка. У переважній більшості випадків мова йде про макроси в документах Microsoft Office;

– ***скрипт-віруси*** - віруси, що виконуються у середовищі певної командної оболонки: раніше - bat-файли в командній оболонці DOS, зараз частіше VBS-і Java-скрипти в командній оболонці Windows Scripting Host (WSH)[5].

Окремо варто сказати пару слів про макровіруси. Більшість електронних документів створюються й обробляються у форматі MS Office, інструмент VBA (Visual Basic for Application), який можна використовувати для створення макровірусів поставляється разом з додатком MS Office. Такий стан речей призводить до того, що на сьогоднішній день макровіруси - найбільш розповсюджений тип вірусів. Однак боротьба з ними не викликає особливих проблем і зводиться до вивчення тіла шкідливого макросу за допомогою того ж VBA на предмет виконуваних операцій, контролю стартових макросів AutoOpen, AutoClose, AutoSave, глобальних макросів FileOpen, FileSaveAs, FileSave, FileClose і ряду стандартних операцій, таких як виклик API-функцій, виконання команд Shell і т. д. Процедура лікування макровірусів зводиться до видалення тіла макросу з документів і шаблонів MS Office.

Мережеві черв'яки

Черв'як (мережний черв'як) - тип шкідливих програм, що поширюються по мережних каналах, здатних до автономного подолання систем захисту автоматизованих і комп'ютерних систем, а також до створення й подальшого поширення своїх копій.

У залежності від шляхів проникнення в операційну систему чирви діляться на:

– *поштові черв'яки* (Mail-Worm) - черв'яки, що поширюються у форматі повідомлень електронної пошти;

– *IM черв'яки* (IM-Worm) - черв'яки, що використовують Інтернет-пейджери;

– *P2P черв'яки* (P2P-Worm) - черв'яки, що розповсюджуються за допомогою пірінгових (peer-to-peer) файлообмінних мереж;

– *мережеві черв'яки* (Net-Worm) - інші мережеві черв'яки, серед яких має сенс додатково виділити Інтернет-хробаки і LAN-черв'яки.

Інтернет черв'яки - черв'яки, що використовують для поширення протоколи Інтернет. Переважно цей тип черв'яків поширюється з

використанням неправильної обробки деякими додатками базових пакетів стека протоколів TCP / IP.

LAN черв'яки - черв'яки, що поширюються по протоколах локальних мереж.

Трояни

Троян (троянський кінь) - тип шкідливих програм, основною метою яких є шкідливий вплив стосовно комп'ютерної системи. Трояни відрізняються відсутністю механізму створення власних копій.

Деякі трояни здатні до автономного подолання систем захисту КС, з метою проникнення й зараження системи. У загальному випадку, троян попадає в систему разом з вірусом або хробаком, у результаті необачних дій користувача або ж активних дій зловмисника. Найбільш поширені наступні види троянів:

- ***Клавіатурні шпигуни*** (Trojan-SPY) - трояни, що постійно перебувають у пам'яті і зберігають всі дані, що надходять від клавіатури з метою наступної передачі цих даних зловмисникові. Зазвичай в такий спосіб зловмисник намагається довідатися паролі або іншу конфіденційну інформацію.
- ***Викрадачі паролів*** (Trojan-PSW) - трояни, також призначені для одержання паролів, але не використають спостереження за клавіатурою. Зазвичай в таких троянах реалізовані способи добування паролів з файлів, в яких ці паролі зберігаються різними додатками.
- ***Утиліти віддаленого керування*** (Backdoor) - трояни, що забезпечують повний віддалений контроль над комп'ютером користувача. Існують легальні утиліти такої ж властивості, але вони відрізняються тим, що повідомляють про своє призначення при установці або ж постачаються з документацією, у якій описані їхні функції. Троянські утиліти вилученого керування ніяк не видають свого реального призначення, так що користувач і не підозрює про те, що його комп'ютер підконтрольний зловмисникові. Найбільш популярна утиліта віддаленого управління - Back Orifice.
- ***Анонімні smtp-сервери й проксі*** (Trojan-Proxy) - трояни, що виконують функції поштових серверів або проксі й, що використаються в першому випадку для спам-розсилок, а в другому для замітання слідів хакерами.
- ***Модифікатори настроювань браузера*** (Trojan-Cliker) - трояни, які міняють стартову сторінку в браузері, сторінку пошуку або ще які-небудь настроювання, для організації несанкціонованих звертань до Інтернет-ресурсів.

- ***Інсталятори інших шкідливих програм*** (Trojan-Dropper) - трояни, що представляють можливість зловмисникові здійснювати приховану установку інших програм.
- ***Завантажувачі шкідливих програм*** (Trojan Downloader) - трояни, призначені для завантаження на комп'ютер-жертву нових версій шкідливих програм, або рекламних систем.
- ***Повідомлювачі про успішну атаку*** (Trojan-Notifier) - трояни даного типу призначені для повідомлення своєму "господарю" про зараження комп'ютеру.
- ***"Бомби" в архівах*** (ARCBomb) - трояни, що представляють собою архіви, спеціально оформлені таким чином, щоб викликати нештатну поведінку архіваторів при спробі розархівувати дані - зависання або істотне уповільнення роботи комп'ютера, заповнення диска великою кількістю "порожніх" даних.
- ***Логічні бомби*** - частіше не стільки трояни, скільки троянські складові черв'яків і вірусів, суть роботи яких полягає в тому, щоб за певних умов (дата, час доби, дії користувача, команда ззовні) зробити певну дію: наприклад, знищення даних.
- ***Утиліти дозвону*** - тип троянів, що представляє собою утиліти dial-up доступу в Інтернет через платні поштові служби. Такі трояни прописуються в системі як утиліти дозвону за замовчуванням і спричиняють за собою великі рахунки за користування Інтернетом [2,3,5].

Життєвий цикл шкідливих програм

Процес розмноження вірусів може бути умовно розділений на кілька стадій:

- Активація вірусу.
- Пошук об'єктів для зараження.
- Підготовка вірусних копій.
- Впровадження вірусних копій.

Так само як для вірусів, життєвий цикл хробаків можна розділити на певні стадії:

Проникнення в систему.

- Активація.
- Пошук "жертв".
- Підготовка копій.
- Поширення копій.

У троянів внаслідок відсутності функцій розмноження й поширення, їхній життєвий цикл менше ніж у вірусів - усього три стадії:

- Проникнення на комп'ютер.
- Активація.
- Виконання закладених функцій.

Це, само собою, не означає малого часу життя троянів. Навпаки, троян може непомітно перебувати в пам'яті комп'ютера тривалий час, ніяк не видаючи своєї присутності, до тих пір, поки не виконає свою шкідливу функцію.

Основні шляхи проникнення в систему і активації

Існує твердження - будь-яку шкідливу програму користувач може перемогти самостійно, тобто не вдаючись до використання антивірусних програм. Це дійсно так, за успішними діями будь-якої антивірусної програми стоїть праця вірусних аналітиків, які вручну розбираються з алгоритмами роботи нових вірусів, виділяють сигнатури, описують алгоритм роботи вірусу.

Сигнатура вірусу - в широкому сенсі, інформація, що дозволяє однозначно визначити наявність даного вірусу у файлі або іншому коді.

Прикладами сигнатур є:

- унікальна послідовність байт, присутня в даному вірусі і не зустрічається в інших програмах;
- контрольна сума такої послідовності.

Таким чином, антивірусну програму можна розглядати як засіб автоматизації боротьби з вірусами. Слід зауважити, що аналіз вірусів потребує від користувача володіння більшим обсягом специфічних знань в області програмування, роботи операційних систем і т.д.

Сучасні шкідливі програми використовують складні технології маскування і захисту своїх копій, які обумовлюють необхідність застосування спеціальних засобів для їх аналізу.

Процес підготовки шкідливою програмою своїх копій для поширення може істотно відрізнятися від простого копіювання. Автори найбільш складних у технологічному плані вірусів намагаються зробити різні копії максимально несхожими для ускладнення їх виявлення антивірусними засобами. Як наслідок, складання сигнатури для такого вірусу вкрай ускладнено.

При створенні копій для маскування можуть застосовуватися наступні технології:

Шифрування - вірус складається із двох функціональних блоків: власне вірусу і шифратора. Кожна копія вірусу складається із шифратора, випадкового ключа й вірусного блоку, зашифрованого цим ключем

Метаморфізм - створення різних копій вірусу шляхом заміни груп команд на еквівалентні, перестановки місцями блоків коду, вставки між

значущими шматками коду "сміттєвих" команд, які практично нічого не роблять.

Поєднання цих двох технологій приводить до появи наступних типів вірусів.

Шифрований вірус - вірус, що використовує просте шифрування з випадковим ключем і незмінний шифратор. Такі віруси легко виявляються по сигнатурі шифратора.

Метаморфний вірус - вірус, що застосовує метаморфізм до всього свого тіла для створення нових копій.

Поліморфний вірус - вірус, що використовує метаморфний шифратор для шифрування основного тіла вірусу з випадковим ключем. При цьому частина інформації, використовуваної для одержання нових копій шифратора також може бути зашифрована. Наприклад, вірус може реалізовувати кілька алгоритмів шифрування і при створенні нової копії міняти не тільки команди шифратора, але і сам алгоритм

Розглядаючи сучасні вірусні загрози необхідно відзначити, що більше 90% відсотків вірусних погроз останнім часом пов'язані з хробаками. Найбільш численну групу в цьому класі шкідливих програм становлять поштові черв'яки. Інтернет-черв'яки також є помітним явищем, але не стільки через кількість, скільки через якість: епідемії, викликані Інтернет-хробаками найчастіше відрізняються високою швидкістю розповсюдження і великими масштабами. IRC і P2P черв'яки зустрічаються досить рідко, частіше IRC та P2P служать альтернативними каналами поширення для поштових і Інтернет-черв'яків.

Поширення через LAN також використовується переважно як додатковий спосіб поширення. Крім того, на етапі активації хробаків можна розділити на дві великі групи, що відрізняються як за технологіями активації, так і по тривалістю життя:

Для активації необхідно активна участь користувача.

Для активації участь користувача не потрібна зовсім або досить лише пасивної участі.

Активація мережного хробака без участі користувача завжди означає, що хробак використовує пролом в безпеці програмного забезпеченні комп'ютера. Це призводить до дуже швидкого поширення хробака у середині корпоративної мережі з більшим числом станцій, істотно збільшує завантаження каналів зв'язку і може повністю паралізувати мережу. Саме цей метод активації використовували черв'яки Lovesan і Sasser. Під пасивною участю користувача розуміється, наприклад, перегляд листів у поштовому клієнті, при якому

користувач не відкриває вкладені файли, але його комп'ютер тим не менше виявляється зараженим.

Активна участь користувача в активації хробака означає, що користувач був уведений в оману методами соціальної інженерії. У більшості випадків основним фактором служить форма подачі інфікованого повідомлення: воно може імітувати лист від знайомої людини (включаючи електронну адресу, якщо знайомий уже заражений), службове повідомлення від поштової системи або ж що-небудь подібне, настільки ж часто зустрічається в потоці звичайної кореспонденції.

При зараженні комп'ютера хробаки зазвичай виробляють наступні дії:

Створюють виконуваний файл з розширенням *.exe з довільним ім'ям або ім'ям дуже схожим на ім'я системних файлів Windows. У деяких черв'яках можуть використовуватися технології притаманні вірусам, в такому випадку черв'яки інфікують вже існуючий файл програми (наприклад WSOCK32.DLL) або замінюють його на свій файл (наприклад I-Worm.MTX записує одну зі своїх процедур в файл WSOCK32.DLL таким чином, що вона перехоплює відсилання даних в Інтернет (процедура send). Внаслідок черв'як в зараженій бібліотеці WSOCK32.DLL отримував управління кожен раз, коли дані відправляються в Інтернет).

Крім цього, разом з додаванням в систему виконуваних файлів, в ряді випадків черв'яки поміщають в систему файли бібліотек, які зазвичай виконують функції Backdoor компонентів (наприклад один з клонів хробака MyDoom - I-Worm.Mydoom.aa створював системному каталозі Windows файл tcp5424.dll , що є Backdoor-компонентом і реєстрував його в системному реєстрі: HKCR \CLSID \ {E6FB5E20-DE35-11CF-9C87-00AA005127ED} \ InProcServer32 {Default} = "% SysDir% \tcp5424.dll").

Шкідлива програма може вносити зміни в системні файли win.ini і system.ini. Наприклад Email-Worm.Win32.Toil при установці в заражаємо системі копіює себе в папку Windows з випадковим ім'ям і записує у файл system.ini наступні значення:

[Boot]

shell = Explorer.exe% ім'я хробака%

що забезпечує йому автозапуск при кожному перезавантаженні Windows (але тільки під Win9x/Me).

Email-Worm.Win32.Atak.h в процесі інсталяції копіює себе з ім'ям dec25.exe в системний каталог Windows і модифікує файл win.ini для свого

подальшого запуску - додає повний шлях до файлу dec25.exe в ключ run секції [windows]:

[Windows]

run =% SystemDir% \ dec25.exe)

Слід так само відзначити, що у файлі system.ini крім секції [boot] шкідливі програми можуть використовувати секцію [Drivers].

Шкідливі програми можуть вносити зміни в наступні гілки реєстру:

HKEY_LOCAL_MACHINE \ Software \ Microsoft \ Windows \ CurrentVersion в ключі Run, RunOnce, RunOnceEx, RunServices, RunServicesOnce - для того щоб система запускала створені хробаком файли.

HKEY_CURRENT_USER \ Software \ Microsoft \ Windows \ CurrentVersion в ключ Run.

Наприклад, Email-Worm.Win32.Bagle.ax після запуску копіює себе в системний каталог Windows, після чого реєструє в реєстрі скопійований файл: HKEY_CURRENT_USER \ Software \Microsoft \ Windows \ CurrentVersion \ Run "Sysformat" = "% System% \ sysformat. exe

HKEY_CLASSES_ROOT \ exefile \ shell \ open \ command

Крім вище перерахованих гілок і ключів реєстру шкідливі програми можуть вносити зміни і в інші гілки і ключі реєстру, наприклад:

HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ WOW \boot

HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ Drivers32

HKEY_LOCAL_MACHINE \ Software \ Microsoft \ Windows NT \ CurrentVersion \ WinLogon

HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Services

HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Active Setup \ Installed Components

HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ AeDebug.

Усунення наслідків зараження

У зв'язку з тим що, що черв'яки практично не використовують технології шифрування і метаморфізму для маскування своїх копій, боротьба з ним вручну дещо спрощується і зводиться до наступного алгоритму дій:

1. Аналіз і виявлення за допомогою Диспетчера завдань Windows підозрілих процесів.

2. Аналіз відкритих портів за допомогою команди Netstat.

3. Вивантаження підозрілих процесів.

4. Аналіз реєстру за допомогою утиліти Regedit.exe в вище перерахованих гілках і ключах.

5. Відновлення та правка ключів реєстру.

6. Пошук інфікованих файлів по імені на основі даних аналізу процесів операційної системи і даних аналізу реєстру.

7. Видалення або заміна інфікованих файлів.

8. Перевантаження системи.

9. Контрольний аналіз процесів, ключів реєстру, відкритих портів.

Якщо підозрілих процесів не виявлено, ключі реєстру не змінилися, значить, процедуру дезінфекції комп'ютера можна вважати успішною, в іншому випадку алгоритм доведеться повторити. Тим не менше, враховуючи той факт, що сучасні хробаки можуть використовувати технології притаманні вірусам, встановлювати backdoor-компоненти, ускладнюючи тим самим, процедуру аналізу та виявлення своїх файлів, для повної впевненості комп'ютер після дезінфекції вручну рекомендується здійснити перевірку антивірусним засобом.

Слід також зазначити, що боротися вручну з шкідливими програмами можна тільки постфактум - після того, як вони вразили комп'ютер, в той же час використання антивірусного програмного забезпечення в переважній більшості випадків не допустить активації шкідливої програми на комп'ютері.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустіть Диспетчер завдань Windows і проаналізуйте список запущених процесів, навантаження, яку вони надають на ЦП. Запишіть в звіт лабораторної роботи назву підозрілих процесів. Поясніть, на підставі чого були зроблені такі висновки.

2. Використовуючи Диспетчер завдань Windows вивантажити підозрілі процеси.

3. Запустіть інтерфейс командного рядка Windows. Ознайомтеся з ключами команди **netstat**. Отримайте статистку поточних мережевих підключень Вашого комп'ютера з параметрами відображення всіх підключень і чекаючих портів, відображення послідовності компонентів, що беруть участь у створенні підключення, відображення виконуваного файлу, який бере участь у створенні кожного підключення. Збережіть статистику в файл з назвою netstat.log. Запишіть протокол лабораторної роботи формат команди для цієї дії. Проаналізує дані netstat.log. Запишіть в протокол лабораторної роботи підозрілі

відкриті порти і програми що їх використовують. Поясніть, на підставі чого були зроблені такі висновки.

4. Запустіть утиліту роботи з реєстром Windows. Проаналізуйте вміст перерахованих в теоретичній частині гілок і ключів реєстру.

5. При необхідності внесіть зміни в параметри ключів з метою видалення підозрілих записів, створені шкідливою програмою. Запишіть в протокол лабораторної роботи всі внесені зміни.

6. Проаналізуйте зміст файлу win.ini на предмет підозрілих записів. При необхідності внесіть корективи в файл win.ini. Запишіть в протокол лабораторної роботи внесені коректування.

7. Проаналізуйте файл system.ini на предмет підозрілих записів. При необхідності внесіть корективи в файл system.ini. Запишіть в протокол лабораторної роботи внесені коректування.

8. Використовуючи стандартні засоби пошуку, знайдіть файли, які породжували підозрілі процеси і видаліть їх. Запишіть в протокол лабораторної роботи імена цих файлів.

9. Перезавантажте комп'ютер, виконайте пункти 1-6, щоб переконатися, що видалення шкідливої програми пройшло успішно, в оперативній пам'яті немає підозрілих процесів, ключі реєстру не змінилися, на комп'ютері відсутні підозріло відкриті порти.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що називається комп’ютерним вірусом?

2. Які існують основні групи вірусів?

3. Опишіть основні типи мережевих черв’яків.

4. Опишіть основні типи троянів.

5. Зробіть аналіз життєвого циклу шкідливих програм.

6. Які існують способи проникнення шкідливої програми на персональний комп’ютер?

7. Назвіть основні ознаки враження вірусом.

8. Поясніть у чому різниця між шифрованим і поліморфним вірусом?

9. Чи достатньо для захисту від зараження шкідливою програмою встановити файлам дозвіл

тільки для читання? Обґрунтуйте відповідь.

10. Поясніть у чому відмінність понять вірус і шкідлива програма.

Лабораторна робота №6

Тема: Шкідливе програмне забезпечення. Використання вразливості «Переповнення буферу»

Мета: Ознайомитися зі шкідливим програмним забезпеченням, набуття компетенцій з пошуку вразливостей в програмному коді, що можуть призводити до реалізації загрози «Переповнення буферу».

Час проведення: - 4 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Переповнення буфера – це аномалія, яка виникає, коли програмне забезпечення, що записує дані в буфер, переповнює ємність буфера, в результаті чого сусідні місця пам’яті перезаписуються. Іншими словами, занадто багато інформації передається в контейнер, якому не вистачає місця, і ця інформація замінює дані в сусідніх контейнерах.

Вразливості переповнення буфера можуть бути використані зловмисниками з метою зміни пам’яті комп’ютера, щоб підірвати або взяти під контроль виконання програми.

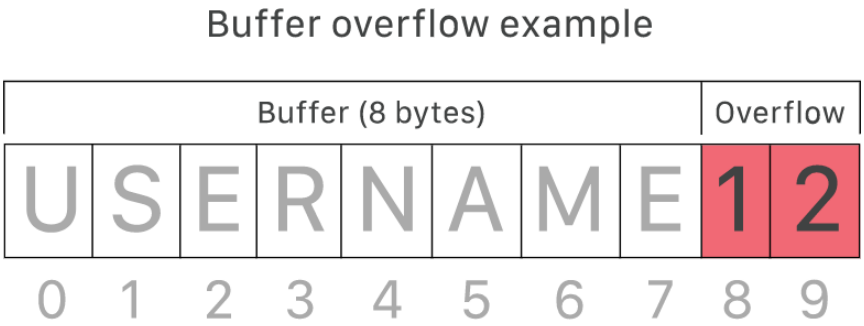

Рисунок 6.1 – Приклад переповнення буферу

Буфер даних

Буфер або буфер даних – це область фізичної пам'яті, яка використовується для тимчасового зберігання даних під час їх переміщення з одного місця на інше. Ці буфери зазвичай знаходяться в оперативній пам'яті.

Комп’ютери часто використовують буфери для покращення продуктивності; більшість сучасних жорстких дисків використовують буферизацію для ефективного доступу до даних, а багато онлайн-сервісів також використовують буфери. Наприклад, буфери часто використовуються в онлайн-потоковому відео, щоб запобігти перериванню. Під час потокової передачі відео програвач завантажує та зберігає, можливо, 20% відео одночасно в буфері, а потім потоково передає його з цього буфера. Таким чином, незначні падіння швидкості з’єднання або швидкі перебої в обслуговуванні не вплинуть на продуктивність потоку відео.

Буфери призначені для того, щоб містити певну кількість даних. Якщо програма, яка використовує буфер, не має вбудованих інструкцій щодо відхилення даних, коли надсилається занадто багато даних до буфера, програма перезапише дані в пам’яті, яка прилегла до буфера.

Переповнення буфера може бути використано зловмисниками для пошкодження програмного забезпечення. Незважаючи на ясність причин цієї вразливості, атаки переповнення буфера все ще є основною проблемою безпеки. У 2014 році загроза, відома як «Heartbleed», піддала атакам сотні мільйонів користувачів через уразливість переповнення буфера в програмному забезпеченні SSL.

3. Як зловмисники використовують переповнення буфера?

Зловмисник може навмисно подати ретельно продуманий вхідний сигнал до програми, що змусить програму зберегти цей вхідний сигнал у недостатньо великому буфері, перезаписуючи частини пам'яті, підключені до буферного простору. Якщо макет пам’яті програми чітко визначений, зловмисник може навмисно перезаписати області, відомі тим, що містять виконуваний код. Потім зловмисник може замінити цей код своїм власним виконуваним кодом, який може різко змінити спосіб роботи програми.

Наприклад, якщо перезаписана частина в пам’яті містить вказівник (об’єкт, який вказує на інше місце в пам’яті), код зловмисника міг би замінити

цей код іншим покажчиком, що вказує на корисне навантаження. Це може передати управління всією програмою в код зловмисника.

4. Хто вразливий до атак переповнення буфера?

Деякі мови кодування більш схильні до переповнення буфера, ніж інші. C та C ++ – дві популярні мови з високою вразливістю, оскільки вони не містять вбудованих засобів захисту від доступу або перезапису даних у їхній пам’яті. Windows, Mac OSX і Linux містять код, написаний однією або обома мовами.

Більш сучасні мови, такі як Java, PERL та C#, мають вбудовані функції, які допомагають зменшити ймовірність переповнення буфера, але не можуть повністю запобігти цьому.

5. Як захиститися від атак переповнення буфера?

На щастя, сучасні операційні системи мають засоби захисту під час виконання, які допомагають пом’якшити атаки переповнення буфера. Розглянемо 2 поширені засоби захисту, які допомагають зменшити ризик експлуатації:

Рандомізація адресного простору (ASLR) – це випадкова зміна розташування адресного простору ключових областей даних процесу. Атаки переповнення буфера зазвичай залежать від знання точного розташування важливого виконуваного коду, проте рандомізація адресних просторів робить це майже неможливим.

Запобігання виконанню даних позначає певні області пам’яті як виконувані, так і невиконані, запобігаючи запуску коду, знайденого у невиконаній зоні.

Розробники програмного забезпечення також можуть вжити запобіжних заходів щодо вразливостей переповнення буфера, пишучи мовами, які мають вбудовані засоби захисту, або використовуючи спеціальні процедури безпеки у своєму коді.

Передумови виконання роботи

Для виконання лабораторних робіт по визначенню шкидливого програмного коду ви повинні використовувати надану вам віртуальну машину. Використання підготовленої віртуальної машини має дві мети По -перше, цільові вразливі програми містять реальні вразливі місця, які можна експлуатувати і автори машини не рекомендують встановлювати їх з **root setuid** на вашій машині. По -друге, все, починаючи від конкретної версії компілятора, закінчуючи операційною системою та встановленими версіями бібліотек, впливатиме на точне розташування коду в стеці. Віртуальна машина забезпечує ідентичне середовище для перевірки та оцінювання завдання.

Віртуальну машину конфігуровано з Ubuntu Linux 16.04 LTS, та з вимкненою рандомізацією адрес ASLR. В системі є один користувач *account user ma пароль cs155*", ви також можете тимчасово стати супер користувачем за допомогою sudo.

Віртуальна машина має єдиний обліковий запис користувача «**user**» з паролем «**cs155**», але можна тимчасово стати користувачем root. Посилання на машину де можливо виконувати роботи https://cs155.stanford.edu/

VM вміщує набір попередньо встановлених інструментів, (curl, wget, openssh, gcc, vim тощо).

Усі програмні застосунки необхідно встановлювати без застосування привілейованих повноважень (root). Версії програмного забезпечення будуть впливати на точне місце розташування коду в стеці. Початкові коди програмних застосунків розташовані в директорії **proj1**. Їх необхідно встановити на віртуальній машині та ідентифікувати наявні вразливості у процесі виконання лабораторної роботи (buffer overflow, double free, format string vulnerability, etc.).

Директорія /targets містить початковий код цільових вразливостей.

Розроблені експлойти необхідно запускати від імені користувача user в результаті вони повинні бути причиною виконання оболонки (/bin/sh), що працює як root. Віртуальна машина має набір попередньо встановлених

інструментів (curl, wget, openssh, gcc, vim тощо), але ви можете додатково встановлювати необхідне програмне забезпечення.

Наприклад, для встановлення mc (GNU Midnight Commander), ви можете запустити як root:

$ apt-get install mc

Коли ви вперше запустите віртуальну машину, на ній буде працювати сервер OpenSSH, відповідно ви зможете під'єднатися з вашої хост -машини, а також передавати файли за допомогою, наприклад, ssh та scp. Під'єднатися до віртуальної машини зі своєї хост -машини також можна за допомогою команди:

$ ssh user@192.168.56.155

Матеріали необхідні для початку роботи з віртуальною машиною розміщені в каталозі proj1.

Код цільових програм знаходиться в каталозі targets/. Використовуйте Makefile для їх зборки та встановлення на віртуальну машину. Зверніть увагу, що виконання цільового коду повинно відбуватися без використання повноважень root. Команди

$ cd target

$ make

$ sudo make install

Дозволять скомпілювати цільові програми та встановити прапорець виконуваного стеку. При розробці експлоїтів використовуйте шлях до цільових програм ../tmp/ (наприклад /tmp/target1, / tmp/target3 і т.п.)

Каталог sploits/ в tar-архіві завдань вміщує скелет коду експлоїтів, які можна використовувати для написання експлоїту (імена sploit1.c, sploit2.c… sploit5 відповідають цільовим файлам). В експлоїти також включено shellcode Aleph One для статичної константи char*

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. **Виконати інсталяцію необхідних програм, які було зазначено в інструкції до лабораторної роботи, як користувач «user»**:

$ cd targets

$ make

$ sudo make install

Спочатку перейдить до директорії proj1, а звідти до ./targets – каталогу, який містить коди всіх цільових програм, проте зараз зосередимо увагу лише на target1.c.

Скриншот треба у звіт

Утиліта **make** автоматично визначає, які частини великої програми повинні бути перекомпільовані і команди для їх перекомпіляції. Найбільш часто **make** використовується для компіляції С-програм.

Запуск команди make-**Скриншот треба у звіт**

Результат виконання роботи команди make- **Скриншот треба у звіт**

Після виконання «make install», програма make бере виконавчі файли з попереднього кроку і копіює їх в деякі відповідні місця, щоб до них можна було отримати доступ.

Запуск команди make install-**Скриншот треба у звіт**

Перевірити наявність в директорії /tmp встановлених цільових програм з правами root.

Індикація наявності цільових програм-**Скриншот треба у звіт**

2. Аналіз програмного коду target1

За допомогою редактора **nano** представити лістинг коду з файлу **target1.c** (скомпільованого файлу) – цільової вразливості. **-Скриншот треба у звіт**

Дослідити представлені функції та команди. Виявити, що в цільовій програмі використовується небезпечна функція **strcpy**, яка дозволяє записати більше даних, ніж вміщує виділений під них масив. Оскільки функція не перевіряє довжину рядка і розмір буферу, вона не повинна використовуватися для роботи з даними, розмір яких невідомий, щоб уникнути переповнення буфера **dst**.

3. Розробка експлойту

Каталог sploits/ вміщує шаблони кодів для експлойтів. Файли програмних розробок названо у відповідності до цілей: sploit1.c, sploit3.c sploit5.c, у заголовок кожного включено файл shellcode.h.

Індикація експлойтів--**Скриншот треба у звіт**

Шаблон експлойту для цільової програми target1-**Скриншот треба у звіт**

Далі потрібно з'ясувати адресу комірки початку змінної buf (представлена в коді вразливої програми), і там будемо виконувати shellcode. Для такої ситуації зручно використати gdb.

GDB, налагоджувач проектів GNU, дозволяє бачити, що відбувається «всередині» іншої програми під час її виконання – або що інша програма робила в момент її аварійного завершення роботи.

Отримати значення **0xbffffc5c** після запуску налагоджувача-**Скриншот треба у звіт**

Ідентифікована адреса комірки-**Скриншот треба у звіт**

Наповнити sploit1.c.

1) Для використання вразливості «Переповнення буфера» треба створити глобальну змінну SEED довжиною 265 байт (256 байтів функціїї buf в target1.c + 4 байти – sfp (відповідний регістр EBP), 4 байти будуть передані інструкції ret, 1 байт – розмір «нульового» байта для завершення програми).

2) Потім потрібно передати змінну SEED в масив buffer.

3) Відповідно масив buffer передамо в функцію memset. Дана функція додає NOP (no operation) інструкції асемблера.

4) Використайте функцію memcpy, яка перезапише початок shellcode в початок масиву buffer, поки не закінчиться довжина shellcode.

5) *(unsigned long)* – неоголошена змінна типу long, що записує в позицію buffer[260] (початок EIP регістру) значення, що відповідає початку стеку, виділеному під buf. Це значення було отримано за допомогою gdb – 0xbffffc5c. Саме на це місце запишемо навантаження, таким чином вказівник

EIP вкаже програмі повернутися до shellcode, і буде отримано права адміністратора.

6) Щоб відобразити кінець масиву треба додати до останнього елементу масиву buffer значення 0 - buffer [SEED-1] = 0.

7) І останнім кроком передати параметри для цільової програми target1.

Лістинг експлойту sploit1.c-**Скриншот треба у звіт**

Отримати новий код. Для його компіляції використайте утиліту **gcc**, і запустити команду: **gcc -Wall –o sploit1 sploit1.c**. Дана команда згенерує виконуваний файл elf (Executable and Linkable Format). Запускаємо скомпільований експлойт: **./sploit1**

Компіляція і запуск експлойту **-Скриншот треба у звіт**

Перевірити за допомогою команди «whoami» отримання прав-root

Зробити висновок.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке переповнення буфера?
2. Що таке буфер?
3. Як зловмисники використовують переповнення буфера?
4. Хто вразливий до атак переповнення буфера?
5 Як захиститися від атак переповнення буфера?
6. Які ви знаєте сучасні найпоширеніші антивірусні програми?
7. Охарактеризуйте можливості шкідливого програмного коду.

Лабораторна робота №7

Тема: Шкідливе програмне забезпечення. Використання вразливості «Помилка на одиницю»

Мета: набуття компетенцій з пошуку вразливостей в програмному коді, що можуть призводити до реалізації загрози «Помилка на одиницю».

Завдання: розробити експлойт для вразливої цільової програми, скомпілювати її та встановити з правами root в директорій /tmp.

Час проведення: - 4 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Помилка на одиницю або **помилка неврахованої одиниці** (англ. Off-by-one error) – це логічна помилка в алгоритмі, що включає зокрема дискретний варіант порушення граничних умов. Помилка часто зустрічається в програмуванні, коли кількість ітерацій покрокового циклу виявляється на одиницю менше або більше необхідного. Наприклад, програміст при порівнянні вказує «менше або дорівнює», замість «менше», або помиляється, відраховуючи початок послідовності не з нуля, а з одиниці (індексація масивів у багатьох мовах програмування починається з нуля).

The loop should run n times. n can be any integer value.

```
for (int i = 1; i < n; i++) {
  /* Body of the loop */
}
```

i	n	loop
1	5	runs
2	5	runs
3	5	runs
4	5	runs
5	5	no run

Рисунок 7.1 – Приклад помилки на одиницю

2. Проблема безпеки

Звичайна помилка на одиницю, яка призводить до проблеми безпеки – це некоректне використання функції strncat() зі стандартної бібліотеки C. Звичайне

непорозуміння, пов'язане з strncat (), полягає в тому, що null-байт не може бути записаний далі, ніж довжина рядка. Насправді функція пише null-байт за вказаною довжиною рядка, якщо третій параметр дорівнює або перевищує цю довжину.

Помилка на одиницю є поширеною при використанні стандартної бібліотеки C, тому що в ній не реалізований єдиний підхід щодо того, віднімати чи ні 1: функції подібні fgets() і strncpy() ніколи не вийдуть за вказану їм довжину (fgets () сама віднімає 1 і витягує (length-1) байтів), тоді як інші функції, подібні strncat(), здійснюють запис далі зазначеної для рядка довжини. З цієї причини програміст повинен пам'ятати, для яких функцій потрібно віднімати 1.

У деяких системах (особливо в архітектурі з порядком байтів від молодшого до старшого) це може призвести до перезапису значущих байтів в стеці процесу, що може створити умову, коли зловмисник отримує дані, що дозволяють йому викликати процедуру процесу.

Один з підходів, за допомогою якого можна вирішувати такі проблеми – використовувати модифікацію зазначених функцій, які рахують кількість записаних байтів з урахуванням довжини буфера замість того, щоб писати або читати максимальну кількість байтів. Прикладом можуть служити функції strlcat() і strlcpy(), які часто розглядаються як «безпечні», тому що виключають випадкову можливість запису за кінцем буфера.

При великих числах помилка на одиницю часто не настільки важлива в конкретному випадку. При невеликих числах, однак, і в деяких специфічних випадках, де точність першорядна, поява помилки на одиницю може бути катастрофічною. Іноді помилка може повторитися і, тому, посилена, тим, хто виконує некоректні обчислення, якщо наступна людина робить цю помилку знову (звичайно, ця помилка може бути здійснена і в зворотний бік).

Передумови виконання роботи

Для виконання лабораторних робіт по визначенню шкидливого програмного коду ви повинні використовувати надану вам віртуальну машину. Використання підготовленої віртуальної машини має дві мети По -перше, цільові вразливі програми містять реальні вразливі місця, які можна експлуатувати і автори машини не рекомендують встановлювати їх з **root setuid** на вашій машині. По -друге, все, починаючи від конкретної версії компілятора,

закінчуючи операційною системою та встановленими версіями бібліотек, впливатиме на точне розташування коду в стеці. Віртуальна машина забезпечує ідентичне середовище для перевірки та оцінювання завдання.

Віртуальну машину конфігуровано з Ubuntu Linux 16.04 LTS, та з вимкненою рандомізацією адрес ASLR. В системі є один користувач *account user та пароль cs155*", ви також можете тимчасово стати супер користувачем за допомогою sudo.

Віртуальна машина має єдиний обліковий запис користувача «**user**» з паролем «**cs155**», але можна тимчасово стати користувачем root. Посилання на машину де можливо виконувати роботи https://cs155.stanford.edu/

VM вміщує набір попередньо встановлених інструментів, (curl, wget, openssh, gcc, vim тощо).

Усі програмні застосунки необхідно встановлювати без застосування привілейованих повноважень (root). Версії програмного забезпечення будуть впливати на точне місце розташування коду в стеці. Початкові коди програмних застосунків розташовані в директорії **proj1**. Їх необхідно встановити на віртуальній машині та ідентифікувати наявні вразливості у процесі виконання лабораторної роботи (buffer overflow, double free, format string vulnerability, etc.).

Директорія /targets містить початковий код цільових вразливостей.

Розроблені експлойти необхідно запускати від імені користувача user в результаті вони повинні бути причиною виконання оболонки (/bin/sh), що працює як root. Віртуальна машина має набір попередньо встановлених інструментів (curl, wget, openssh, gcc, vim тощо), але ви можете додатково встановлювати необхідне програмне забезпечення.

Каталог sploits/ в tar-архіві завдань вміщує скелет коду експлоїтів, які можна використовувати для написання експлоїту (імена sploit1.c, sploit2.c… sploit5 відповідають цільовим файлам). В експлоїти також включено shellcode Aleph One для статичної константи char*

ПОРЯДОК ВИКОНАННЯ РОБОТИ

Попередні налаштування

Виконаємо інсталяцію необхідних програм, які було зазначено в інструкції до лабораторної роботи, як користувач «user»:

$ cd targets

$ make

$ sudo make install

Спочатку перейдить до директорії proj1, а звідти до ./targets – каталогу, який містить коди всіх цільових програм, проте зараз зосередимо увагу лише на target2.c.

Перехід в каталог targets й індикація встановлених файлів- **Скриншот треба у звіт**

Утиліта **make** автоматично визначає, які частини великої програми повинні бути перекомпільовані і команди для їх перекомпіляції. Найбільш часто **make** використовується для компіляції С-програм.

Запуск команди make - **Скриншот треба у звіт**

Результат виконання роботи команди make **- Скриншот треба у звіт**

Після виконання «make install», програма make бере виконавчі файли з попереднього кроку і копіює їх в деякі відповідні місця, щоб до них можна було отримати доступ.

Запуск команди make install- **Скриншот треба у звіт**

Перевірьте наявність в директорії /tmp встановлених цільових програм з правами root.

Індикація наявності цільових програм- **Скриншот треба у звіт**

2. Аналіз програмного коду target2

За допомогою редактора **nano** лістинг коду з файлу **target2.c** (скомпільованого файлу) – цільової вразливості. Представте **- Скриншот треба у звіт**

Лістинг коду target2.c **- Скриншот треба у звіт**

Дослідити представлені функції та команди, виявити, що в цільовій програмі допущено помилку на одинцю.

3. Розробити експлойт

Каталог sploits/ вміщує шаблони кодів для екплойтів. Файли програмних розробок названо у відповідності до цілей: sploit1.c, sploit3.c sploit5.c, у заголовок кожного включено файл shellcode.h.

Індикація експлойтів - **Скриншот треба у звіт**

Шаблон експлойту для цільової програми target2 - **Скриншот треба у звіт**

Далі потрібно з’ясувати адресу комірки початку змінної buf (представлена в коді вразливої програми), і там будемо виконувати shellcode. Для такої ситуації виконали команду *x buf*. Результат команди – 0xbffffcc8, це значення буде використовуватися для розробки експлойту.

Наступним кроком треба запустити команду info frame для виводу інформації про стековий фрейм функції bar() – адресу повернення з функції. І значення, що відповідає за початок EBP, є 0xbffffd90.

Ідентифікована адреса комірки **- Скриншот треба у звіт**

Наповнити експлойт sploit2.c. Спробуйте перевизначити молодший байт вже наявного EBP. Змінити вказівник стекового кадру, аби в результати він опинився за адресою, яку займає рядок шелкоду.

За допомогою нововведень ми мусимо змінити LSB так, щоб збережений EBP вказував на адресу всередині нашого буфера, де знаходиться шкідливий EIP (адреса шеллкоду).

Першим кроком треба заповнити рядок експлуатації початковою адресою buf в bar. У процесі виконання в певний момент одразу перед вказівкою «ret» покажчик стека буде вказувати на середину buf, таким чином, інструкція «ret» поверне 0xBFFFFC90 (початкова адреса buf) у eip. Це запустить шеллкод, і ми отримаємо доступ до root-прав.

Цикл for містить перевірку «i <= len», яка дасть змогу перезаписати LSB збереженого EBP для кадру стека foo (зі значенням 0xBFFFFD68). Це відрізняється лише в LSB з 0xBFFFFD50 (адреса всередині buf).

Після перезапису LSB файлу збереженого EBP для кадру стека foo до 0x50 і, як наслідок, виходу, інструкція для виклику функції панелі відновить EBP до 0xBFFFFD50.

Інструкція «leave» для функції foo встановить esp на 0xBFFFFD50, перейде на 4 байти, щоб відновити непотрібну інформацію EBP і залишити ESP рівним 0xBFFFFD50. Адреса 0xBFFFFD50 знаходиться всередині buf, воно містить значення 0xBFFFFC90 (початок буфера), а отже, виклик «ret» встановить EIP значення 0xBFFFFC90.

Також payload – це 201 байт рядка експлуатації. Це той самий байт, який буде перезаписувати LSB збереженої ebp для стекового фрейму foo.

Лістинг експлойту sploit2.c - **Скриншот треба у звіт**

Отримайте повністю новий код. Для його компіляції використайте утиліту **gcc**, і запустить команду: **gcc -Wall –o sploit2 sploit2.c**. Дана команда згенерує виконуваний файл elf (Executable and Linkable Format). Запускаємо скомпільований експлойт: **./sploit2**.

Компіляція і запуск експлойту **Скриншот треба у звіт**

Показати , що досягнуто прав-root.
Перевірити за допомогою команди «whoami».
Зробить **висновок.**

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке помилка на одиницю?
2. Що таке експлойт?
3. Як зловмисники використовують «помилку на одиницю»?
4. Хто вразливий до цього виду атак?
5 Як захиститися від цього шкідливого коду?

6. Які ви знаєте сучасні найпоширеніші антивірусні програми?
7. Охарактеризуйте можливості шкідливого програмного коду.

Лабораторна робота №8

Тема: Криптографічний вид захисту інформації. Поняття шифрування.

Мета: Ознайомитися з поняттям криптографії, способами шифрування файлів, папок, повідомлень та криптографічними методами захисту інформації, розглянути основні засоби здійснення криптографічного захисту інформації.

Час проведення: - 2 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Криптографія — наука про методи перетворення (шифрування) інформації з метою її захисту від зловмисників.

Криптографічні методи захисту інформації - це методи захисту даних із використанням шифрування.

Шифрування — процес застосування шифру і інформації, що захищається, тобто перетворення інформації, що захищається, в шифроване повідомлення за допомогою певних правил, що містяться в шифрі.

Всі відомі способи шифрування розбиті на п'ять груп: підстановка (заміна), перестановка, аналітичне перетворення, гамування і комбіноване шифрування. Кожен з цих способів може мати декілька різновидів.

Під ***кодуванням*** розуміється такий вид криптографічного закриття, коли деякі елементи даних (не обов'язково окремі символи), що захищаються, замінюються заздалегідь вибраними кодами (цифровими, літерними, літерно-цифровими поєднаннями і так далі). Цей метод має два різновиди: *смислове* і *символьне* кодування.

При *смисловому кодуванні* кодовані елементи мають цілком певний сенс (слова, пропозиції, групи пропозицій). При символьному кодуванні кодується кожен символ тексту, що захищається.

Символьне кодування по суті співпадає з підстановлювальним шифруванням.

До окремих видів криптографії відносяться методи розтину і стиснення даних. Розтин полягає в тому, що масив даних, що захищаються, ділиться (розтинається) на такі елементи, кожен з яких окремо не дозволяє розкрити зміст інформації, що захищається. Виділені таким чином елементи даних розносяться по різних зонах пам'яті або розташовуються на різних носіях. Стиснення даних є заміною однакових рядків даних або послідовностей

однакових символів, що часто зустрічаються, деякими заздалегідь вибраними символами.

Розтин шифру — процес отримання інформації (відкритого тексту), що захищається, з шифрованого повідомлення (шифртекста) без знання застосованого шифру.

Дешифрування — процес, зворотний шифруванню, що полягає в перетворенні шифрованого повідомлення в інформацію, що захищається, за допомогою певних правил, що містяться в шифрі.

Під ***ключем*** в криптографії розуміють змінний елемент шифру, який застосовують для шифрування конкретних повідомлень.

Одне з центральних місць в понятійному апарату криптографії займає таке поняття, як стійкість шифру. Під ***стійкістю шифру*** розуміють здатність шифру протистояти всіляким методам розтину.

Для того, щоб криптографічні методи перетворення забезпечили ефективний захист інформації, вони повинні задовольняти ряду вимог. У стислому вигляді їх можна сформулювати таким чином:

- складність і стійкість криптографічного захисту повинні вибиратися залежно від об'єму і ступеня секретності даних;

- надійність захисту повинна бути такою, щоб секретність не порушувалася у тому випадку, коли зловмисникові стає відомий метод захисту;

- метод захисту, набір використовуваних ключів і механізм їх розподілу не можуть бути дуже складними;

- виконання процедур прямого і зворотного перетворень повинне бути формалізованим. Ці процедури не повинні залежати від довжини повідомлень;

- помилки, що виникають в процесі виконання перетворення, не повинні розповсюджуватися на текст повною мірою і по системі;

- надмірність, що вноситься процедурами захисту повинна бути мінімальною.

Головна мета шифрування (кодування) інформації - її захист від несанкціонованого читання.

Системи криптографічного захисту (системи шифрування інформації) можна поділити за різними ознаками:

• за принципами використання криптографічного захисту (вбудований у систему або додатковий механізм, що може бути відключений);

• за способом реалізації (апаратний, програмний, програмно-аларатний);

• за криптографічними алгоритмами, які використовуються (загальні, спеціальні);

• за цілями захисту (забезпечення конфіденційності інформації (шифрування) та захисту повідомлень і даних від модифікації, регулювання доступу та привілеїв користувачи);

• за методом розподілу криптографічних ключів (базових/сеансових ключів, відкритих ключів) тощо.

Додаткові механізми криптозахисту - це додаткові програмні або апаратні засоби, які не входять до складу системи. Така реалізація механізмів криптозахисту має значну гнучкість і можливість швидкої заміни. Для більшої ефективності доцільно використовувати комбінацію додаткових і вбудованих механізмів криптографічного захисту.

За способом реалізації криптографічний захист можна здійснювати різними способами: *апаратним, програмним або програмно-апаратним.*

Апаратна реалізація криптографічного захисту - найбільш надійний спосіб, але й найдорожчий. Інформація для апаратних засобів передається в електронній формі через порт обчислювальної машини всередину апаратури, де виконується шифрування інформації. Перехоплення та підробка інформації під час її передачі в апаратуру може бути виконана за допомогою спеціально розроблених програм типу "вірус"[5].

Програмна реалізація криптографічного захисту значно дешевша та гнучкіша в реалізації. Але виникають питання щодо захисту криптографічних ключів від перехоплення під час роботи програми та після її завершення. Тому, крім захисту від "вірусних" атак, потрібно вжити заходів для забезпечення повного звільнення пам'яті від криптографічних ключів, що використовувались під час роботи програм "збирання сміття".

Крім того, можна використовувати комбінацію апаратних і програмних механізмів криптографічного захисту. Найчастіше використовують програмну реалізацію криптоалгоритмів з апаратним зберіганням ключів. Такий спосіб криптозахисту є досить надійним і не надто дорогим. Але, вибираючи апаратні засоби для зберігання криптографічних ключів, треба пам'ятати про забезпечення захисту від перехоплення ключів під час їх зчитування з носія та використання в програмі.

В основу шифрування покладено два елементи: *криптографічний алгоритм і ключ.*

Криптографічний алгоритм - це математична функція, яка комбінує відповідний текст або іншу зрозумілу інформацію з ланцюжком чисел (ключем) з метою отримання незв’язаного (шифрованого) тексту.

Усі криптографічні алгоритми можна поділити на дві групи: *загальні і спеціальні.*

Спеціальні криптоалгоритми мають таємний алгоритм шифрування, а ***загальні*** криптоалгоритми характерні повністю відкритим алгоритмом, і їх криптостійкість визначається ключами шифрування. Спеціальні алгоритми найчастіше використовують в апаратних засобах криптозахисту.

Загальні криптографічні алгоритми часто стають стандартами шифрування, якщо їхня висока криптостійкість доведена. Ці алгоритми оприлюднюють для обговорення, при цьому навіть визначається премію за успішну спробу його "злому". Криптостійкість загальних алгоритмів визначається ключем шифрування, який генерується методом випадкових чисел і не може бути повторений протягом певного часу. Криптостійкість таких алгоритмів буде вищою відповідно до збільшення довжини ключа.

Є дві великі групи загальних криптоа лгоритмі в***: симетричні і асиметричні***. До ***симетричних криптографічних алгоритмів*** належать такі алгоритми, для яких шифрування і розшифрування виконується однаковим ключем, тобто і відправник, і отримувач повідомлення мають користуватися тим самим ключем. Такі алгоритми мають досить велику швидкість обробки як для апаратної, так і для програмної реалізації. *Основним їх недоліком* є труднощі, пов'язані з дотриманням безпечного розподілу ключів між абонентами системи. ***Асиметричні криптоалгоритми шифрування і розшифрування*** виконують за допомогою різних ключів, тобто, маючи один із ключів, не можна визначити парний для нього ключ. Такі алгоритми часто потребують значно довшого часу для обчислення, але не створюють труднощів під час розподілу ключів, оскільки відкритий розподіл одного з ключів не зменшує криптостійкості алгоритму і не дає можливості відновлення парного йому ключа.

Усі криптографічні алгоритми можна використовувати з різними цілями, зокрема:

• для шифрування інформації, тобто приховування змісту повідомлень і даних;

• для забезпечення захисту даних і повідомлень від модифікації.

З найпоширеніших методів шифрування можна виділити американський алгоритм шифрування DES (Data Encryption Standart, розроблений фахівцями фірми IBM і затверджений урядом США 1977 році) із довжиною ключа, що може змінюватися, та алгоритм ГОСТ 28147-89, який був розроблений та набув

широкого застосування в колишньому СРСР і має ключ постійної довжини. Ці алгоритми належать до симетричних алгоритмів шифрування.

Алгоритм Потрійний DES був запропонований як альтернатива DES і призначений для триразового шифрування даних трьома різними закритими ключами для підвищення ступеня захисту.

RC2, RC4, RC5 - шифри зі змінною довжиною ключа для дуже швидкого шифрування великих обсягів інформації. Здатні підвищувати ступінь захисту через вибір довшого ключа.

IDEA (International Data Encryption Algoritm) призначений для швидкої роботи в програмній реалізації.

Для приховування інформації можна використовувати деякі асиметричні алгоритми, наприклад, алгоритм RSA. Алгоритм підтримує змінну довжину ключа та змінний розмір блоку тексту, що шифрується.

Алгоритм RSA дозволяє виконувати шифрування в різних режимах:

• за допомогою таємного ключа відправника. Тоді всі, хто має його відкритий ключ, можуть розшифрувати це повідомлення;

• за допомогою відкритого ключа отримувача, тоді тільки власник таємного ключа, який є парним до цього відкритого, може розшифрувати таке повідомлення;

• за допомогою таємного ключа відправника і відкритого ключа отримувача повідомлення. Тоді тільки цей отримувач може розшифрувати таке повідомлення.

Але не всі асиметричні алгоритми дозволяють виконувати шифрування даних у таких режимах. Це визначається математичними функціями, які закладені в основу алгоритмів.

Другою метою використання криптографічних методів є захист інформації від модифікації, викривлення або підробки. Цього можна досягнути без шифрування повідомлень, тобто повідомлення залишається відкритим, незашифрованим, але до нього додається інформація, перевірка якої за допомогою спеціальних алгоритмів може однозначно довести, що ця інформація не була змінена.

Для симетричних алгоритмів шифрування така додаткова інформація - це код автентифікації, який формується за наявності ключа шифрування за допомогою криптографічних алгоритмів.

Для асиметричних криптографічних алгоритмів формують додаткову інформацію, яка має назву електронний цифровий підпис. Формуючи електронний цифровий підпис, виконують такі операції:

• за допомогою односторонньої хеш-функції обчислюють прообраз цифрового підпису, аналог контрольної суми повідомлення;

• отримане значення хеш-функції шифрується: а) таємним або відкритим; 6) таємним і відкритим ключами відправника і отримувача повідомлення - для алгоритму RSA;

• використовуючи значення хеш-функції і таємного ключа, за допомогою спеціального алгоритму обчислюють значення цифрового підпису.

Для того, щоб перевірити цифровий підпис, потрібно:

• виходячи із значення цифрового підпису та використовуючи відповідні ключі, обчислити значення хеш-функції;

• обчислити хеш-функцію з тексту повідомлення;

• порівняти ці значення. Якщо вони збігаються, то повідомлення не було модифікованим і відправлене саме цим відправником.

Ефективність захисту систем за допомогою будь-яких криптографічних алгоритмів значною мірою залежить від безпечного розподілу ключів. Тут можна виділити такі основні методи розподілу ключів між учасниками системи.

1. Метод базових/сеансових ключів. Такий метод описаний у стандарті ISO 8532 і використовується для розподілу ключів симетричних алгоритмів шифрування. Для розподілу ключів вводиться ієрархія ключів: головний ключ (так званий майстер-ключ, або ключ шифрування ключів) і ключ шифрування даних (тобто сеансовий ключ). Ієрархія може бути і дворівневою: ключ шифрування ключів/ключ шифрування даних. Старший ключ у цій ієрархії треба розповсюджувати неелектронним способом, який виключає можливість його компрометації. Використання такої схеми розподілу ключів потребує значного часу і значних затрат.

2. Метод відкритих ключів. Такий метод описаний у стандарті ISO 11166 і може бути використаний для розподілу ключів як для симетричного, так і для асиметричного шифрування. За його допомогою можна також забезпечити надійне функціонування центрів сертифікації ключів для електронного цифрового підпису на базі асиметричних алгоритмів та розподіл сертифікатів відкритих ключів учасників інформаційних систем. Крім того, використання методу відкритих ключів дає можливість кожне повідомлення шифрувати окремим ключем симетричного алгоритму та передавати цей ключ із самим повідомленням у зашифрованій асиметричним алгоритмом формі.

Вибір того чи іншого методу залежить від структури системи і технології обробки даних. Жоден із цих методів не забезпечує "абсолютного" захисту

інформації, але гарантує, що вартість "злому" у кілька разів перевищує вартість зашифрованої інформації.

Щоб використовувати систему криптографії з відкритим ключем, потрібно генерувати відкритий і особистий ключі. Після генерування ключової пари слід розповсюдити відкритий ключ респондентам. Найнадійніший спосіб розповсюдження відкритих ключів - через сертифікаційні центри, що призначені для зберігання цифрових сертифікатів.

Цифровий сертифікат - це електронний ідентифікатор, що підтверджує справжність особи користувача, містить певну інформацію про нього, слугує електронним підтвердженням відкритих ключів.

Сертифікаційні центри несуть відповідальність за перевірку особистості користувача, надання цифрових сертифікатів та перевірку їхньої справжності.

Розкриття зашифрованих текстів (в першу чергу знаходження ключа) здійснюється за допомогою методів криптоаналізу.

Основними методами криптоаналізу є:

- ***статистичні***, при яких знаючи статистичні властивості відкритого тексту намагаються досліджувати статистичні закономірності шифротексту і на підставі виявлених закономірностей розкрити текст;

- метод вірогідних слів, в якому при зіставленні деякої невеликої частини шифротекста з відомим фрагментом відкритого тексту намагаються знайти ключ і з його допомогою розшифрувати весь текст.

Необхідний фрагмент відкритого тексту можна знайти за допомогою статистичних методів або просто вгадати, виходячи з передбачуваного змісту або структури відкритого тексту.

Оскільки криптографічні методи застосовуються давно, то вже сформульовані основні вимоги до них.

1. Метод повинен бути надійним, тобто відновлення відкритого тексту при володінні тільки шифротекстом, але не ключем повинно бути практично нездійсненним завданням

2. Через труднощі запам'ятовування об'єм ключа не повинен бути великим.

3. Через труднощі, пов'язані з складними перетвореннями, процеси шифрування повинні бути простими.

4. Через можливості появи помилок передачі дешифровка шифротексту, що містить окремі помилки, не повинна привести до нескінченного збільшення помилок в отриманому передбачуваному відкритому тексті.

5. Через труднощі передачі об'єм шифротексту не повинен бути значно більше відкритого тексту.

Криптографічні перетворення забезпечують вирішення двох головних проблем захисту інформації: *проблеми секретності* (позбавлення супротивника можливості витягувати інформацію з каналу зв'язку) і проблеми *імітостійкості* (позбавлення супротивника можливості ввести помилкову інформацію.

Сучасний криптографічний захист інформації здійснюється за допомогою спеціалізованого програмного забезпечення, що дає змогу використовувати складні методи шифрування з мінімальною затратою часу. Ще одним позитивним моментом такого застосування є доступність можливостей зашифрувати важливу інформацію, в тому числі повідомлення, окремі файли і папки звичайним користувачам, адже більшість таких програм доступні для розповсюдження та мають нескладний інтерфейс, звільняють користувача від необхідності знання способів та методів шифрування. Звичайно, якщо йде мова про криптографічний захист банківських систем, інформації стратегічного чи державного значення, то використовують спеціальні криптографічні системи, що складаються з програмних засобів, доступних вузькому колу користувачів, а частіше всього, оригінальними, спеціально створеними засобами.

Програмний комплекс криптографічного захисту інформації «Криптосервер»

Комплекс є сукупністю засобів криптографічного захисту інформації з функціями шифрування інформації, формування та зберігання ключової інформації, а також надання послуг встановлення автентичності даних, які надходять, зберігаються та обробляються в комп’ютеризованих системах оброблення інформації.

Комплекс реалізує наступні функції:

Захист даних - забезпечує захист інформації, яка передається по загальнодоступним мережам, від несанкціонованого ознайомлення й/або модифікації.

Керування - забезпечує конфігурування та налаштування параметрів компонентів Комплексу, необхідних для їх функціонування.

Аудит - дозволяє проводити аналіз записів у файлах протоколів.

Ідентифікація й автентифікація - блокує доступ до можливостей керування Комплексом осіб, що не мають відповідних повноважень.

Захист функціонування - підтримує функціонування Комплексу при спробах порушити цілісність програмного забезпечення або конфігураційної інформації.

Комплекс складається з наступних компонентів:

Центр генерації ключів (ЦГК) - програмний модуль, призначений для генерації закритих і відкритих ключів, а також для запису ключових носіїв для всіх компонентів Комплексу. Центр генерації ключів встановлюється на комп'ютері, що не має мережевих з'єднань.

Центр розподілу ключів (ЦРК) - програмний модуль, призначений для зберігання та видачі мережевими каналами довіреним компонентам Комплексу сертифікатів відкритих ключів, інформації про контур безпеки, що визначає учасників захищеної мережі, та іншої службової інформації.

Модуль шифрування (МШ) - програмний модуль, призначений для побудови захищеної мережі (шлюз ЗМ), який встановлюється на границі ЗМ або границі сегмента ЗМ, що функціонує в інтересах одного, декількох або всіх суб'єктів (об'єктів) даної ЗМ (сегмента ЗМ), що забезпечує створення захищених з'єднань із іншими довіреними модулями шифрування.

Модуль керування (МК) - програмний модуль, призначений для дистанційного керування компонентами Комплексу, такими як ЦРК, МШ.

Компоненти Комплексу обмінюються ідентифікаційною інформацією й виконують процедури автентифікації з використанням сертифікатів відкритих ключів, які запитуються у ЦРК. Захищене з'єднання може бути встановлено тільки після виконання процедур взаємної автентифікації компонентів Комплексу.

На сьогодні існує велика кількість алгоритмів і протоколів шифрування. Серед алгоритмів симетричного криптографії, яких безліч, можна згадати RC4, RC5, CAST, DES, AES і т.д. Оптимальна довжина ключів шифрування для цих алгоритмів - 128 розрядів. Що стосується асиметричного шифрування, то тут в основному використовуються алгоритми RSA, Diffie-Hellman і El-Gamal, при цьому довжина ключів шифрування звичайно становить 2048 розрядів. Найбільш широко для криптографічного захисту переданих по каналах зв'язку даних, включаючи листи електронної пошти, застосовується протокол SSL, у якому для шифрування даних використовуються ключі RSA.

Популярним пакетом програм для шифрування листування по електронній пошті і будь-яких даних, що зберігаються на жорсткому диску є PGP (Pretty Good Privacy).

В безкоштовному варіанті PGP Desktop Email 9.6 (призначений тільки для приватного некомерційного використання) включені функції шифрування файлів і папок, в платному варіанті опцій значно більше - від шифрованого листування (включаючи через інтернет-пейджери) і створення зашифрованих дисків на локальному комп'ютері до розгортання захищеної локальної мережі.

Переваги:

• Простота і зручність у використанні.

• Вичерпна безпека електронної кореспонденції на шляху від відправника до одержувачу -100% захист від несанкціонованого доступу і зміни даних.

• Автоматичний пошук відкритих ключів одержувача в інтернет-каталозі PGP Global Directory.

• Загальна інфраструктура ключів для шифрування електронної пошти, миттєвих повідомлень, файлів.

• Створення зашифрованих архівів PGP Zip на одну дію.

• Політики захисту інформації для автоматичного шифрування/дешифрування та цифрового підпису електронних листів з урахуванням адреси одержувача і відправника, а також вмісту і теми листа.

• Шифрування миттєвих повідомлень і пересилаються файлів.

• Перевірені технології.

• Криптозахист даних з використанням перевірених часів технологій, які отримали широке галузеве визнання.

• Підтримка галузевих стандартів і 100% сумісність з рішеннями OpenPGP і S/MIME.

• Програма PGP Desktop Email можна захищати за допомогою ключа PGP або сертифікату

X.509. Воно також підтримує існуючі інфраструктури з ключами.

• Підтримка смарт-карт/маркеров забезпечує багатофакторну автентифікацію адміністраторів і користувачів.

• Інтеграція з популярними клієнтами електронної пошти, включаючи MS Outlook, Outlook Express, Eudora, Entourage і Apple Mail.

• Оновлення ПЗ PGP Desktop Email 9.6.

Програма STCLite 3.3 призначена для забезпечення захищеного і шифрованого пересилання поштових повідомлень по мережі Інтернет, а також для шифрованого та безпечного зберігання інформації на знімних носіях і жорстких дисках. Легко та витончено можна зашифрувати вміст листа, переконатися в цьому на власні очі (тому що шифрування виконується не на льоту, коли вже нічого змінити неможливо, а до відправки повідомлення) і

спокійно відправити його своєму кореспонденту. При цьому ви можете бути впевнені, що жодних слідів не залишилося у вашому комп'ютері, тому що вся обробка повідомлення ведеться тільки в оперативної пам'яті комп'ютера, без запису інформації на жорсткий диск.

Програма STCLite 3.3 складається з модуля шифрування тексту, модуля шифрування файлів і папок, модуля стеганографії.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Шифр Цезаря — симетричний алгоритм шифрування підстановками. Використовувався римським імператором Юлієм Цезарем для приватного листування. Принцип дії полягає в тому, щоб циклічно зсунути алфавіт, а ключ — це кількість літер, на які робиться зсув. Користуючись алфавітом АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ та використовуючи в якості ключа власний номер в журналі зашифрувати повідомлення та записати даний шифр в звіт **«Шифр Цезаря має замало ключів — на одиницю менше, ніж літер в абетці. Тому перебрати усі ключі не складає особливої роботи. Дешифрування з одним з ключів дасть нам вірний відкритий текст»**.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Дайте визначення поняттям «криптографія», «криптографічні методи захисту інформації», «шифрування».
2. Назвіть основні групи шифрування.
3. Що таке кодування? Які типи кодування ви знаєте?
4. В чому суть методів розтину та стиснення даних?
5. Що розуміють під стійкістю шрифту?
6. Сформулюйте основні вимоги до методів криптографічного перетворення.
7. Яка головна мета шифрування (кодування) інформації?
8. Охарактеризуйте способи реалізації криптографічного захисту.
9. Що таке криптографічні алгоритми? На які групи вони поділяються, охарактеризуйте їх.
10. Сформулюйте вимоги до криптографічних методів.

Лабораторна робота №9

Тема: Відновлення даних з різних носіїв інформації

Мета: Придбання теоретичних знань та практичних навичок з відновлювання даних за допомогою спеціалізованих програмних засобів.

Час проведення: - 8 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Попри постійне підвищення надійності запам'ятовувальних пристроїв, втрата цифрової інформації залишається досить поширеним явищем. До основних причин втрати файлів належать помилки користувачів, проблеми із програмним забезпеченням (наприклад, комп'ютерні віруси), перебої у подачі живлення, а також апаратні збої. На щастя, уся інформація, що зберігається на цифровому носії, майже завжди підлягає відновленню.

Відновлення даних можна описати як процес отримання інформації, що знаходиться на запам'ятовувальному пристрої, доступ до якої неможливий стандартними засобами через її попереднє видалення або пошкодження цифрового носія. Різні підходи використовуються для відновлення втрачених файлів, але тільки за умови, що їх *вміст присутній десь у сховищі*. Наприклад, відновлення даних не охоплює ситуації, коли файл ніколи не було записано до постійної пам’яті, наприклад документи, які були створені, але зрештою не збереглись на жорсткий диск через збій у живленні. Крім того, жоден з існуючих методів не здатен впоратися з випадками повного стирання, яке відбувається, коли інша інформація займає той самий простір у сховищі – за таких обставин втрачені файли можна дістати тільки із зовнішньої резервної копії.

В цілому методи відновлення даних поділяються на два типи: програмні й такі, що передбачають ремонт або заміну пошкоджених апаратних компонентів у лабораторних умовах. У даній роботі ми будемо розглядати програмний метод відновлення даних. Програмно-орієнтований підхід застосовується в більшості випадків та базується на використанні спеціалізованих утиліт, здатних інтерпретувати логічну структуру проблемного накопичувача, зчитувати необхідний вміст та надавати його користувачеві у вигляді, зручному для подальшого копіювання.

Кожна файлова система по-різному здійснює видалення файлу. Наприклад, у **Windows** файлова система **FAT** позначає запис про файл у каталозі як "невикористаний" і знищує інформацію щодо розміщення файлу (за винятком початку файлу), в **NTFS** тільки *запис про файл* позначається як *"невикористаний"*, елемент видаляється з каталогу, а дисковий простір також позначається як *"невикористаний"*; більшість файлових систем **Linux/Unix** *знищують файловий дескриптор* (інформацію про місце розташування файлу, тип, розмір і т. д.) та позначають дисковий простір як *"вільний"* [5,15,19].

Основна мета видалення файлу – звільнити місце у сховищі, яке займає файл, для зберігання нового файлу. Дисковий простір не спустошується одразу з міркувань продуктивності, тому фактичний вміст файлу залишається на диску до тих пір, поки цей простір не буде використаний повторно для збереження нового файлу.

Інформацію, що залишилася на **непошкодженому сховищі**, зазвичай можна відновити без допомоги професіоналів, користуючись спеціалізованим програмним забезпеченням. Однак слід мати на увазі, що *жодну інформацію не можна повернути після її перезапису*. З огляду на це, не слід нічого записувати до сховища до тих пір, поки не буде врятовано останній файл.

Більшість утиліт для відновлення даних працюють із використанням алгоритмів **аналізу метаданих**, методу **RAW-відновлення** на основі відомого вмісту файлів або комбінації цих двох підходів.

Метадані – це прихована службова інформація, що міститься у файловій системі. Її аналіз дозволяє програмі знайти основні структури у сховищі, що ведуть облік розміщення вмісту файлів, їх властивостей та ієрархії каталогів. Після цього інформація опрацьовується та використовується для відтворення пошкодженої файлової системи. Цей метод є більш оптимальним, ніж RAW-відновлення, через те, що дозволяє отримати файли з їх оригінальними іменами, теками, відмітками дати та часу. Якщо метадані не були серйозно пошкоджені, іноді вдається відновити всю структуру каталогів, залежно від специфіки механізмів, які використовує файлова система, позбавляючись від "непотрібних" елементів. Втім, такий аналіз не вдасться успішно виконати за відсутності значущих частин метаданих. Саме тому вкрай важливо утримуватися від використання засобів для виправлення помилок у файловій системі або ініціювання операцій, які можуть привести до її модифікації, до тих пір, поки дані не буде відновлено.

Флеш-носії, такі як USB флеш-накопичувачі, карти пам'яті та твердотільні накопичувачі (SSD), широко використовуються як окремі портативні носії даних, так і в якості розширення пам'яті для фотоапарата, мобільного телефону або ПК.

Незважаючи на велику різноманітність цих носіїв, у них є дещо спільне – файли, що зберігаються на них, не будуть негайно видалятися (єдиним винятком є SSD з командою TRIM). Після видалення простір, який займається файлом, не очищується, а позначається як доступний для подальших записів. Тому фото- та відеофайли залишаються незмінними на карті пам'яті, доки їх не буде перезаписано іншими файлами. З цієї причини настійно рекомендується припинити створення будь-яких нових файлів на носії та відкласти його для подальшого відновлення даних.

Як правило, якщо бажаний результат не досягнуто за допомогою аналізу метаданих, виконується пошук файлів за їх відомими вмістом. У цьому випадку під "відомим вмістом" мається на увазі не вміст всього файлу, а тільки окремі послідовності неопрацьованих даних, які характерні для файлів даного формату та можуть вказувати на початок або кінець файлу. Ці послідовності називаються "сигнатурами файлів" та можуть використовуватися для визначення того, чи належить фрагмент даних у сховищі файлу заданого типу. Файли, відновлені за допомогою цього методу, отримують розширення на основі знайденої сигнатури, нові імена та розміщуються у нових папках, що зазвичай створюються окремо для файлів різних типів. Основне обмеження цього підходу полягає в тому, що деякі файли можуть не мати упізнаваних сигнатур або мати тільки сигнатуру, що позначає початок файлу, що ускладнює визначення його кінця, особливо коли частини файлу не зберігаються послідовно.

Щоб повернути втрачені файли з максимальною ефективністю, програмне забезпечення для відновлення даних може використовувати описані методи одночасно під час однієї процедури сканування, запущеної для сховища.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Завантажити ПЗ Recuva та встановимо його. Сайт, де можливо скачати R.Saver. https://www.softportal.com/get-21645-rsaver.html
2. Встановив Recuva та запустив його

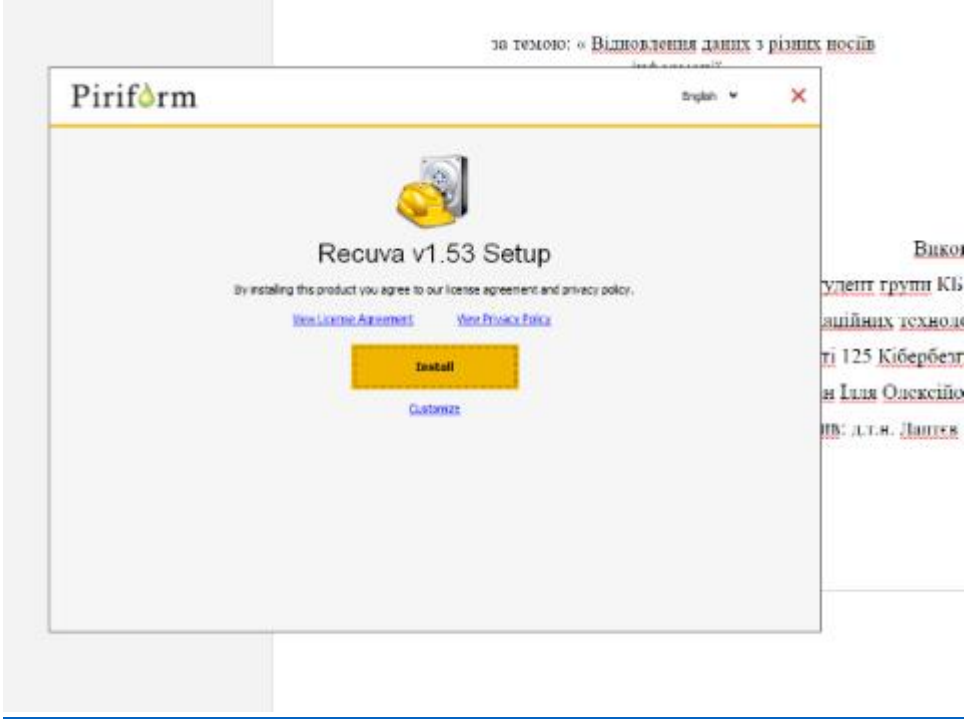

Рисунок 9.1 - Встановлення Recuva

3. Перевірти, щоб флеш-носій був пустий.
4. Створимо файл файл MS Word, який будемо використовувати для відновлення. Додамо туди довільний текст (рис. 8.2).

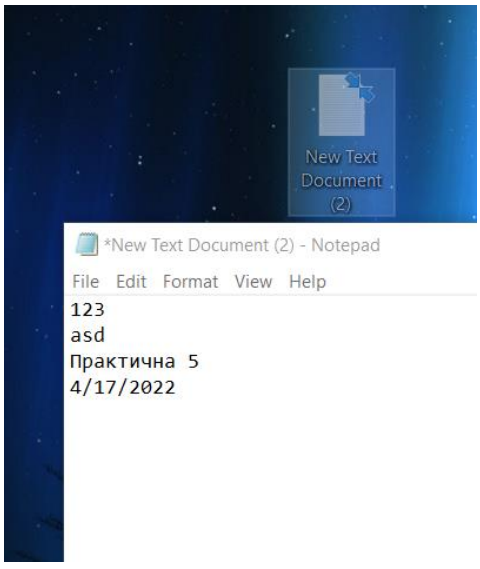

Рисунок 9.2 - Тестовий документ

5. Занотуйте параметри документа, дату його створення та модифікування, розмір та розміщення .

На наступному скриншоті видно параметри текстового файлу, дата його створення та розміп.

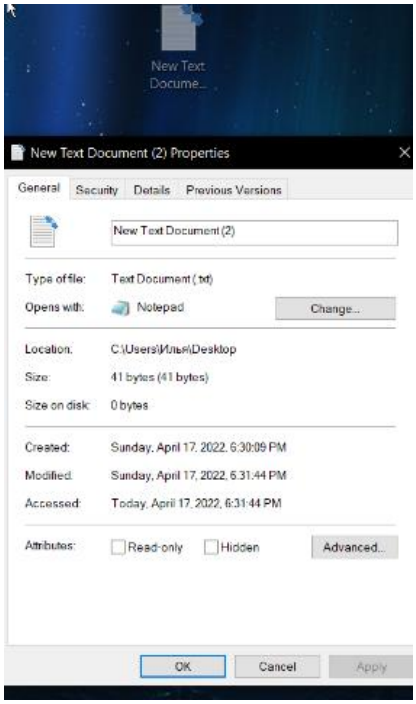

Рисунок 9.3 - Параметри файлу

6. Видалити файл та запустимо програму відновлення даних

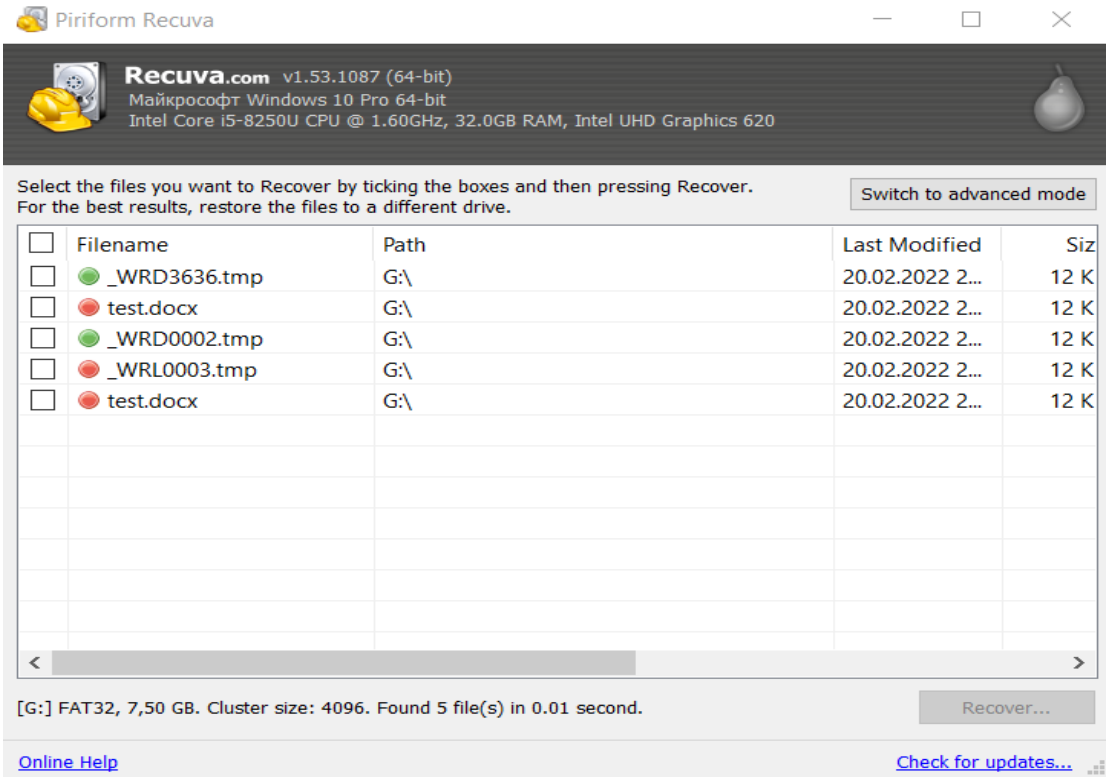

Рисунок 9.4 - Результат відновлення

Здійснити відновлення даного файлу та перевіримо його дані (рис. 9.4 – 9.5).

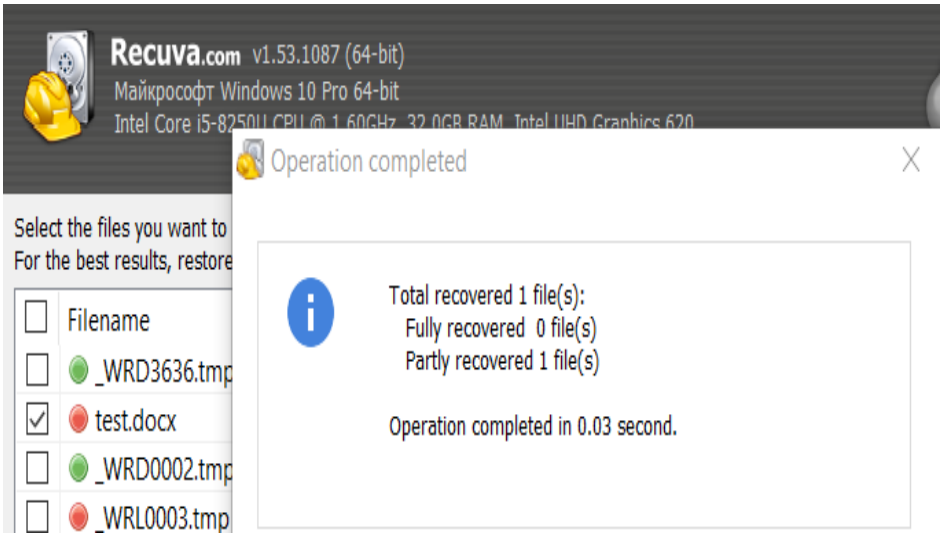

Рисунок 9.5 – Процес відновлення файлу

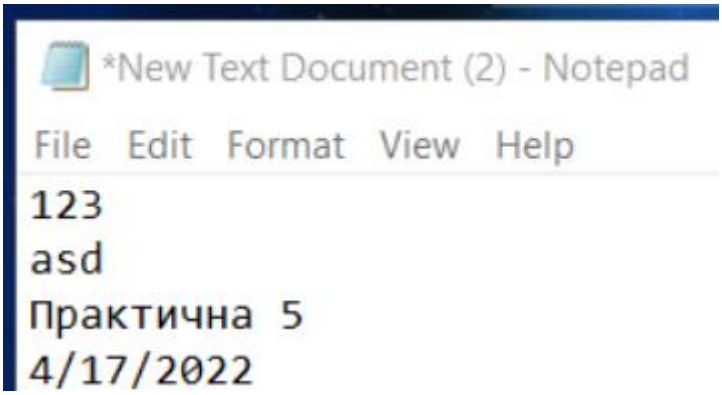

Рисунок 9.6 – Відновлений файл

7. Видалити файл у корзину та очистимо її, відформатуємо флеш-носій швидким форматуванням.

8. Спробуйте знову відновити фай

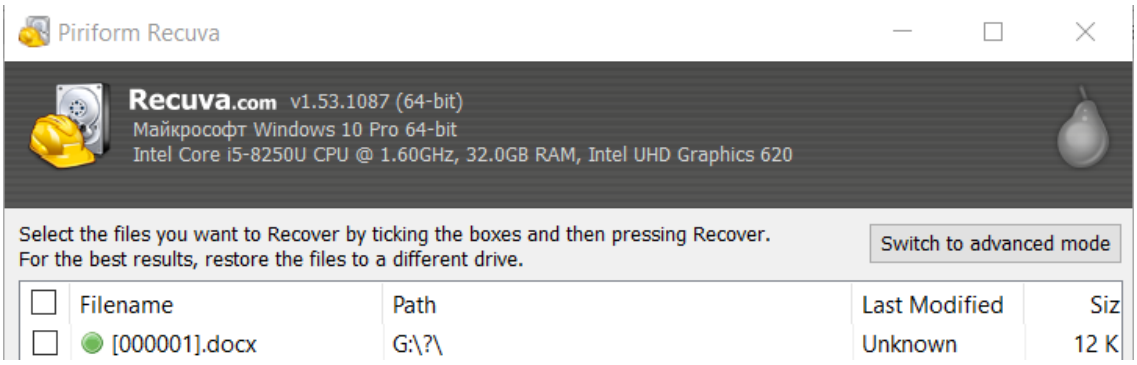

Рисунок 9.7. - Результат відновлення

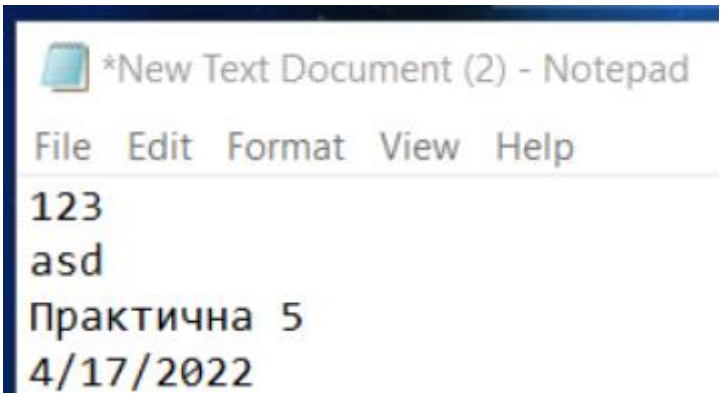

Рисунок 9.8. Відновлений файл

9. Зробити висновки.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Можуть програми поновляти стерти або видалені файли. Відразу або після перевантаження комп’ютера?

2. Які типи файлів поновлюють наведені у завданні програми?

3. Форматування диска може забезпечити неможливість поновлення?

4. Як видалити файл без можливості поновлення?

5. Особливості роботи програми якою Ви користувались?

6. Суть команди інтерфейсу ATA TRIM, чому її не використовують для HDD комп’ютеров.

7. Особливості відновлення даних після використання спеціалізованого ПЗ типу BitLocker To Go.

Лабораторна робота №10

Тема: Захист інформації на мобільних телефонах. Огляд найпоширеніших мобільних вірусів та засобів боротьби з ними.

Мета: Ознайомитися з причинами та способами попередження втрати інформації на мобільних телефонах, поняттями мобільного віруса та антивіруса, ознаками прослуховування розмов по мобільному телефоні та способами боротьби з ними.

Час проведення: - 4 години.

Місце проведення: - за розкладом занять.

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Невід'ємною частиною нашого життя є мобільні телефони. На сьогоднішній день вони стали не тільки засобом зв'язку, а й отримали ряд додаткових функцій, просто незамінних для сучасної людини в повсякденному житті, що робить їх не тільки корисними, а й вразливими до атак різного роду для отримання інформації. Мобільні загрози поділяються **на 3 типи:** рівня пристрою, рівня мережі та рівня додатків. Кожен вид має свою специфіку і способи попередження.

Загрози рівня пристрою існують через недосконалість операційних систем і драйверів. У кожного телефону є базовий заводський захист і хакери шукають способи його зламати. Для цього хакери залучають експлойти - спеціальні програми, які використовують уразливі місця в ПЗ смартфона.

Загрози рівня мережі використовують контроль над Wi-Fi, Bluetooth, USB-кабелем, SMS-повідомленнями, голосовими дзвінками. Наприклад, зловмисники можуть використовувати вразливі бездротові точки доступу, стаючи посередником між пристроєм співробітника і сервером.

Загрози рівня додатків несуть у собі використання шкідливого програмного забезпечення. Магазини додатків iOS і Android щодня блокують сотні підозрілих додатків для мобільних пристроїв. Крім шкідливого програмного забезпечення, існує ще й так зване сіре ПЗ, яке теж може бути небезпечним для чутливих даних компанії.

Тому мобільна безпека - це не встановлення однієї програми. Рішення повинно покривати всі рівні, проводячи моніторинг і аналітику пристроїв, своєчасно усувати загрози та попереджати користувачів про потенційно шкідливі об'єкти, веб-ресурси та небезпечні дії.

Особливості роботи з сучасними мобільними телефонами:

1. Вхід в систему "Банк-Клієнт тепер здійснюється не тільки з офісного комп'ютера, але і з домашніх ПК і з мобільних пристроїв (найчастіше під ОС Android), на яких, як правило, взагалі немає ніякого антивіруса, а якщо і є - то безкоштовна, сильно обмежена в функціоналі версія.

2. Мобільні телефони можуть величезному ризику втрати / крадіжки. Інформація (включаючи паролі і логіни доступу до корпоративних ресурсів) може потрапити в не завжди доброзичливі руки.

3. Кількість загроз для ОС Android зростає катастрофічними темпами разом із зростанням кількості використовуваних пристроїв.

4. Більш особистих пристроїв, приблизно 60%, не мають ніякого захисту. А значить, люди ніяк не захищені від атак хакерів, використовувані ними додатки можуть мати уразливості.

В більшості сучасних мобільних пристроїв для зберігання інформації (контактів, SMS, нотаток, календарів та інші) використовується Flash пам’ять та носії на її основі – карти пам'яті та/або SSD накопичувачі. Тип носія, формати запису та організації даних в пам'яті не стандартизовано, тому кожний виробник реалізує в своїх гаджетах власні формати та алгоритми збереження інформації. Тому у випадку втрати інформації на мобільному пристрої відновлення даних представляє собою складну технічну задачу.

Причини втрати інформації на мобільних пристроях:

1. Механічні пошкодження.

2. Випадкове чи навмисне видалення інформації.

3. Форматування зовнішньої пам’яті.

4. Неправильна синхронізація даних .

5. Випадкова втрата чи навмисне встановлення паролю доступу до пристрою.

6. Вплив вірусів та використання злоякісного ПЗ.

Мобільні віруси - це невеликі програми, призначені для втручання в роботу мобільного телефону, смартфона, комунікатора, які записують, пошкоджують або видаляють дані і поширюються на інші пристрої через SMS та Інтернет. Вперше про мобільних вірусах заговорили ще в 2000 році. Вірусами назвати їх було важко, так як це був набір команд, виконуваний телефоном, який передавався через SMS. Такі повідомлення забивали відповідні комірки пам'яті і при видаленні блокували роботу телефону. Найбільшого поширення набули команди для таких телефонів, як Siemens і Nokia. У недалекому минулому основною платформою, для якої писали віруси, була Symbian. Однак ситуація на ринку мобільних телефонів складається таким

чином, що смартфони на Symbian, хоча і вельми популярні, поступаються чималі частини ринку апаратам інших виробників. Наприклад, пристроям на Windows Mobile, це iPhone та інші платформи.

Тенденція така, що чим функціональніший телефон, тим до більшої кількості загроз він схильний. Будь-які команди, функції та можливості, що дозволяють створювати програми і програми для мобільних телефонів, можуть стати інструментом для створення вірусів. Найбільш перспективною платформою для написання вірусів є Java 2ME, так як переважна більшість сучасних телефонів підтримує дану платформу. Основною метою мобільних вірусів, як і у випадку з комп'ютерними вірусами, є отримання персональної інформації, яку можна продати або використати в особистих потребах. До такої інформації можуть відноситися особисті дані власника телефону, дані самого пристрою, приватні повідомлення, інколи номери кредитних карт. Цікава історія віруса Ikee, який був розроблений в кінці 2009 року, та лише поширювався та міняв заставку на телефоні. Автора хробака, австралійця Ешлі Таунс, навіть запросили на роботу в Apple після виходу вірусу. Але на основі Ikee були створені модифікації, які, наприклад, були пристосовані для крадіжки інформації про банківський рахунок власника апарату. У березні 2011 року власники смартфонів могли безкоштовно закачати з кількох інтернет - серверів таку невеличку гру як «3D Anti-terrorist». Так як же постраждали жертви даної іграшки? Справжні розробники даної гри безкоштовно надавали тільки демо-версію. Однак автор вірусу її зламав і додав код власної програми, потім розмістив на різних серверах мобільних ігор. Ті люди, які вирішили випробувати гру нічого поганого спочатку не відчували, поки не отримали рахунок за телефон від мобільного оператора. Виявилося, що їх улюблений телефон без участі власника сам дзвонив на платні закордонні номери. Після проведеного аналізу було констатовано, що програма цю операцію проробляла з тимчасовим інтервалом у 50 секунд в темний час доби, коли власники смартфонів скоріше за все відпочивали. Причому дзвінки виконувалися на шість платних номерів. Ще одним прикладом вірусу в стільниковому телефоні може стати, так званий, «MMS Bomber». Даною програмою інфіковані вже багато мільйонів телефонів у всьому світі. Суть даного вірусу полягає в тому, що в телефон потрапляє мила додаткова програмка, після інсталяції, якої починають висилатися мультимедійні повідомлення на всі номери, збережені в записній книжці зараженого телефону. Зміст повідомлення - це посилання на сайт, натиснувши на яку в телефон завантажується новий «MMS Bomber». Причому шкідник може відключити всю систему телефону, що робить деінсталяцію програми абсолютно неможливим. Існує декілька різних схем і

напрямків розвитку вірусів, за якими діють автори мобільних вірусів: Крадіжка персональної інформації. В даному випадку віруси збирають різні відомості, наявні в телефоні, наприклад, контакти власника телефону, паролі від програм, параметри облікових записів, таких, як Google Play або AppStore. Вся інформація, отримана вірусом, відправляється на сервер зловмисників, де використовується на їх розсуд. Один з найсерйозніших вірусів такого плану - Android.Geinimi. Потрапляючи в систему, він визначає місце розташування смартфона, завантажує файли з Інтернету, зчитує і записує закладки браузера, отримує доступ до контактів, здійснює дзвінки, відправляє, читає і редагує SMS-повідомлення. Відправка платних SMS-повідомлень, дзвінки на «партнерський номер» без відома власника. У даному випадку за відправку повідомлення або за дзвінок списується серйозна сума коштів з особового рахунку власника телефону. Зрозуміло, гроші потрапляють до рук зловмисників. З найвідоміших подібних загроз можна назвати Android SmsSend, а також давно відомі RedBrowser і Webster для Java-платформи. Вони маскуються під різні корисні програми, викликаючи тим самим довіру у користувача. Також існують віруси і для інших платформ, наприклад Symbian OS, Windows Mobile і інших.

Шахрайство за допомогою використання систем інтернет-банкінгу. У даному випадку вірус відкриває доступ до мобільного додатком для роботи з банком або відповідному веб-сайту, або перехоплює SMS-повідомлення, що передаються користувачеві від систем інтернет-банкінгу. Небезпека даного типу може підстерігати власників мобільних телефонів, що працюють на різних платформах. Відомий троян Trojan-Spy.Symb OS. Zbot, що працює в сукупності з популярним вірусом Zbot для звичайних ПК.

Боротьба з мобільними вірусами. Для того щоб запобігти зараженню мобільним вірусом, слід пам'ятати про заходи безпеки. Способи розповсюдження вірусів досить різні. Наприклад, вірус можна «підчепити» по Bluetooth. А це означає, що Bluetooth не слід тримати постійно включеним, а, беручи що-небудь через «блакитний зуб», потрібно точно знати, що саме Ви приймаєте. Вірус чи інша шкідлива програма повинна бути встановлена на стільниковому телефоні перш ніж вона зможе працювати. Творці шкідливого ПЗ широко використовують засоби соціальної інженерії для розповсюдження своїх програм, часом - досить примітивних, які не вміють розмножуватися самостійно, але вміють робити щось інше. Так, наприклад, зловмисник може написати програму, яка таємно відправляє платні SMS з зараженого телефону. Цю програму можуть безкоштовно пропонувати для скачування з Інтернету та встановлення на мобільний телефон. Як правило, подібні програми мають

вельми привабливі назви. У результаті, встановивши своїми руками на свій же телефон шкідливу програму, власник апарату може позбутися засобів на рахунку, а до усього іншого, втрачати їх регулярно. Уважно ставтеся до всього, що Ви встановлюєте на телефон. Не завантажуйте підозрілих програм. Пам'ятайте - підозріла (чи занадто вже привабливо названа) програма може бути небезпечною. Шкідливий код, на жаль, може проникнути на Ваш пристрій абсолютно без Вашого відома. І тут вам може допомогти лише спеціалізоване захисне ПЗ - мобільний антивірус. Антивірусні компанії вже почали випускати версії своїх програм для захисту мобільних телефонів: Symantec Client Security для смартфонів Nokia, Avira Mobile Security, TotalAV Antivirus & VPN, Panda Dome, WinMobile, Avast Mobile Security для Android і багато інших. Avira Security Pro призначений для всебічного захисту смартфона Android – під час перегляду веб-сайтів, скачування файлів, здійснення покупок або спілкування. Крім того, у випадку втрати або крадіжки смартфону Avira Security Pro допоможе захистити особисті дані і визначити місцезнаходження пристрою - навіть якщо в ньому замінили SIM-карту.

Основні можливості програм-антівірусів:

1. виявлення вірусів, шпигунських програм та інших загроз;
2. блокування переходів по шкідливим і фішингових посиланнях;
3. фільтрація небажаних дзвінків та SMS;
4. приховування особистих контактів, дзвінків та SMS-повідомлень;
5. можливість віддалено заблокувати смартфон, стерти особисті дані та визначити місцезнаходження пристрою в разі втрати або крадіжки;
6. мінімальне використання ресурсів батареї: Dr.Web Mobile Security Suite підтримується най сучасними операційними системами: Android OS, S60, Symbian, Windows Mobile.

Продукт забезпечує антивірусний захист комунікаторів і КПК від вірусів і інших інтернет-загроз, створених спеціально для інфікування мобільних пристроїв. Сумісний з будь-якими КПК і комунікаторами. Avira AntiVir Mobile надає надійний захист для кишенькових комп'ютерів і смартфонів на базі операційної системи Windows Mobile. Avira AntiVir Mobile може виконати сканування всього пристрою на предмет наявності шкідливого ПЗ, в тому числі сканування карт пам'яті і мережевих папок. Оновлення антивірусних баз не викличе складнощів, оскільки його можна виконати декількома шляхами, а сам файл оновлень дуже малий за розміром.

Ключові особливості та функції програми Avira AntiVir Mobile:

1. просте завантаження програми шляхом підключення телефону до комп'ютера або через WLAN, LAN, GPRS;

2. легке ручне оновлення через LAN, WLAN або Dial-Up GPRS. Підтримується також оновлення через комп'ютер або проксі-сервера;

3. програма оновлюється через один файл, який дуже малий за розміром. Відкат у разі незавершеного з якихось причин поновлення;

4. сканування карт пам'яті і мережевих папок;

5. автоматична або інтерактивна процедура видалення вірусів;

6. можливість видалення заражених файлів;

7. надання специфічної інформації по кожному знайденому вірусу;

8. повноцінне управління програмою кнопками телефону.

Втрата інформації в результаті прослуховування мобільного телефону. Прослуховування мобільного телефону - не така вже і дорога справа, як це може здатися. Для цього досить початкових знань в радіоелектроніці і вміння працювати своїми руками. Інтернет сьогодні сповнений описів самих різних приладів, за допомогою яких можна не тільки прослуховувати мобільні розмови, але і перепрограмувати мобільні телефони так, щоб отримати доступ до записника або навіть безперешкодно прослуховувати розмови, які ведуться в безпосередній близькості від включеного мобільного телефону. Свою майстерність не втомлюються вдосконалювати і спецслужби і шахраї. Система мобільного зв'язку в тому вигляді, в якому вона склалася і діє сьогодні, - це три основні складові. Перша - це власне ваш мобільний телефон, який, по суті, є портативним радіопередавачем. Друга - це система базових станцій, "сот", від потужності і розташування яких залежить рівень прийому вашого телефону. І основна складова - комунікаційний центр, який керує роботою всієї системи. У кожного мобільного телефону є персональний електронний серійний номер (MIN). Він кодується виробником в мікрочіпі стільникового телефону. Іноді виробник вказує цей персональний серійний номер в керівництві для користувача, щоб ви могли ідентифікувати свій телефон. Ну, скажімо, у випадку, якщо у вас вкрали ваш стільниковий телефон. Коли ваш апарат підключається до системи стільникового зв'язку, то його мікрочіп зчитує ще й ваш мобільний номер (ESN), який зашифрований в SIM-картці. Тому, якщо ви міняєте, з міркувань безпеки, або будь-яким іншим причинам, вашу SIM-карту, то разом з нею краще викиньте і телефон. Тому що він "повідомить" вашій наступній Sim-карті, який номер на ньому перш стояв. Так само точно "настукає" на вас і ваша нібито секретна Sim-карта, вставлена в інший апарат, тобто по ній можна буде визначити, в якому стільниковому телефоні вона раніше стояла. Цим азам електронної безпеки вчать, при підготовці своїх фахівців всі спецслужби світу. Телефонний зв'язок здійснюється через найближчу до вас "соту", тобто станцію радіопередачі. Вона пов'язана з

базовою станцією, яка вільна приймати і передавати сигнали на великій кількості радіочастот. Ця станція також підключена до звичайної провідної телефонної мережі. Інакше ви б не змогли дзвонити з стільникового телефону на звичайні міські телефони. Для цього базова станція оснащена апаратурою перетворення високочастотного сигналу стільниковому телефону в низькочастотний провідного телефону, і навпаки. Базова станція періодично випромінює службовий сигнал. Пам'ятайте, перші моделі стільникових телефонів моргали нам зеленим вогником: говорячи нам, що зв'язок є. Коли вогник ставав червоним, значить, станція або сигнал від неї загубився. Прийнявши цей сигнал, ваш стільниковий телефон навантажує його своїми даними - електронним номером апарату, мобільним номером, місцем розташування "соти" (територіальне знаходження абонента стільникового зв'язку) - і відправляє назад, щоб базова станція змогла ідентифікувати телефон і перевірити стан рахунку власника. Після цього базова станція прив'язує ваш стільниковий телефон до певної зоні, де ви в даний момент знаходитеся. Це потрібно для того, щоб у разі вхідного або вихідного дзвінка негайно виділити нам одну з вільних частот цієї зони. При цьому в комунікаційному центрі, комп'ютер якого управляє всім цим господарством і всіма з'єднаннями в мережі, в базі даних відкладається вся інформація як про місцезнаходження всіх клієнтів, так і про їхні зв'язки з іншими ж мережами. Це необхідно для ідентифікації абонентів та перевірки їх права на доступ в мережу. У цієї системи як мінімум дві дуже слабких ланки. Перша - сигнал, який йде від вашої трубки до "соти", і другий - передача від "соти" до базової станції. Більшість систем стільникового зв'язку відповідає одному із стандартів аналогового зв'язку. Таких, як AMPS, TAGS, NTS або цифровий - DAMPS, NTT, GSM і т.п. Найбільш поширені стільникові системи використовують діапазони 450, 800 і 900 МГц. Кожен з цих стандартів докладно описаний у відповідній літературі. А якщо ви знаєте діапазон частот, то завжди зможете підібрати апаратуру з потрібними технічними параметрами. Телефонні переговори з стільникових телефонів ловили навіть старі автомобільні магнітоли. Як стверджують фахівці, розмова, що ведеться з стільникового телефону, може бути прослухана за допомогою програмованих сканерів з полосою прийому 30 КГц, здатних здійснювати пошук в діапазоні 860-890 МГц. Для цієї ж мети можна використовувати і звичайні сканери після їх невеликої модифікації, яка детально описана в Інтернеті. І що ще цікаво! Перехопити розмову по мобільному телефону можна навіть шляхом перебудови тюнера в телевізорах старих моделей. Перепрограмування однієї мікросхеми в телефоні дозволяє отримати доступ комутаційного обладнання телефонних компаній. І тоді

рахунки за розмови нікому пред'явити. А клонування ідентифікаційних номерів, яке останнім часом набуває все більшого поширення! Для цього всього потрібен так званий стільниковий кеш - бокс, який являє собою комбінацію сканера, комп'ютера та стільникового телефону. Він виявляє номера MIN і ESN і автоматично перепрограмує себе на них. Причому після разового використання цього поєднання він стирає їх в пам'яті, і вибирає іншу пару. Що, як ви розумієте, робить виявлення шахрайства практично неможливим. При всьому при тому прослуховування мобільного зв'язку - справа аж ніяк не така дорога, як це нам намагаються представити. Так, уже сьогодні одночасна «прослушка» 10 тис. мобільних телефонів може бути здійснена за допомогою одного пристрою вартістю 50 тис. доларів. Будемо вважати, що в місті приблизно 5 млн. мобільних. Це дані експертів. Але розмова одночасно ведеться не більше ніж з 100 тис. з них. Тобто виходить, що для контролю всіх розмов користувачів місцевого зв'язку достатньо 10 таких пристроїв. Це близько півмільйона доларів. Додайте сюди накладні витрати, тобто обробку та зберігання записів, створення алгоритмів пошуку і формування баз даних плюс зарплату фахівців, які всім цим будуть займатися. По самій верхньої цінової планки це не більше 500 тис. на рік. Таким чином, для прослуховування мобільних телефонів великого міста протягом року досить одного мільйона доларів. Вражає? І не думайте, що вас врятує один з смартфонів, які використовують набагато складніший тип шифрування розмови, ніж той, що надається операторами стільникового зв'язку. Так, ще кілька років тому пристрій, здатний "розколювати" подібні розмови, коштував понад півмільйона доларів. Крім того, він був здатний відстежувати одночасно не більше 16 абонентів. Але часи змінюються, і можливостям розуму немає меж. За допомогою мобільного телефону можна легко і безперешкодно прослуховувати розмови, які ведуться в безпосередній близькості від нього. Для цього досить, щоб телефон був зареєстрований в мережі і батарея в нього була не сильно розряджена. Як повідомили деякі ЗМІ, ФБР знайшло спосіб включати практично будь-який мобільний телефон в режим прослуховуючого пристрою без відома його господаря. Здається, що подібною технологією давно вже володіють і наші спецслужби, однак рекламувати це своє досягнення вони якось не квапляться. Чи потрібно говорити про те, що для цього потрібно повне сприяння оператора стільникового зв'язку. Але отримати таку згоду для спецслужб, як ви розумієте, не проблема. На думку фахівців, одним зі слабких місць сучасних наворочених телефонів є процедура обробки службових sms. Саме вона дозволяє стороннім отримати контроль над вашим телефоном. Ці sms потрібні оператору для того, щоб, наприклад, модернізувати програмне

забезпечення вашого мобільного телефону, подаючи команди за допомогою службових sms повідомлень. Це отримало дуже велике поширення за кордоном, але все більше використовується і у нас. Телефон зовні ні як не реагує на ці смс і не перевіряє відправника. Цим можна скористатися для того, щоб перепрограмувати його непомітно для власника. Зокрема, таким чином можна отримати доступ до адресної книги власника або прослуховувати розмови завдяки таємній активації конференц-зв'язку.

Для захисту від прослуховування мобільного телефону фахівці рекомендують:

1. тримати документи з ESN-номером телефону в надійному місці;

2. щомісяця і ретельно перевіряти рахунки на користування мобільним зв'язком;

3. у випадку крадіжки або пропажі стільникового телефону відразу попередити оператора, який надає вам послуги зв'язку;

4. тримати телефон вимкненим до того моменту, поки ви не вирішили ним скористатися. Цей спосіб найлегший і найдешевший, але слід пам'ятати, що досить одного виходу на зв'язок, щоб виявити MIN/ESN номера апарату;

5. регулярно міняти через вашу компанію, що надає вам послуги стільникового зв'язку, MINномер вашого апарату;

6. встановити додатковий чотиризначний PIN-код, що набирається перед розмовою. Цей код ускладнює діяльність шахраїв, так як вони зазвичай перехоплюють тільки MIN і ESN номера, але невелика модифікація апаратури перехоплення дозволяє виявити і його;

7. під час важливих переговорів, на яких ви обговорюєте конфіденційну інформацію, вимкніть телефон, витягніть з нього батарею і, ще краще, приберіть його подалі від себе;

Можливі ознаки «прослуховування» стільникових телефонів Температура батареї. Один із ймовірних індикаторів наявності «прослушки» - це батарея. Якщо ваш телефон у той час, коли ви його не використовуєте буде теплим або навіть гарячим, це означає, що він все ще використовується. Врахуйте, що тепло буває перш за все від надмірного використання. Акумулятор може бути гарячим тільки в тому випадку, якщо телефон використовувався деякий час тому. Телефон розряджається дуже швидко. Заряджаєте свій мобільний телефон частіше, ніж зазвичай - отримаєте ще один знак потенційної загрози. Якщо ви не використовували гаджет більше, ніж зазвичай, цілком можливо, що ваш телефон використовується кимось без вашого відома. Коли мобільний телефон прослуховують, він втрачає заряд акумулятора набагато швидше. Прослуховуваний мобільний телефон постійно записує розмови в кімнаті,

навіть якщо виглядає так, начебто він лежить без діла. Можна використовувати додаток BatteryLife LX, щоб відслідковувати швидкість розрядження батареї. Примітка: мобільні телефони мають тенденцію втрачати максимальний рівень заряду батареї з плином часу. Якщо вашому телефону більше року, то ємність батареї буде неухильно знижуватися в залежності від інтенсивності використання. Затримка при вимиканні. Коли ви вимикаєте свій телефон і спостерігаєте велику затримку, підсвічування, палаючу протягом тривалого часу або просто відмову телефону від виключення, то цілком можливо, що ви на гачку. Завжди помічайте нетипову поведінка телефону. Хоча, описані проблеми можуть бути викликані технічними недоліками в апаратному або програмному забезпеченні телефону. Дивна активність. Коли ваш телефон працює, чи буває так, що у нього раптово спалахує підсвічування, самі по собі встановлюються якісь програми, відбувається мимовільне вимикання? Дивна поведінка може бути сигналом до того, що хтось віддалено керує цим пристроєм. До речі, це також може відбуватися через перешкоди при передачі даних. Фоновий шум. Коли ви розмовляєте, телефон «на прослуховуванні» може створювати перешкоди. Щось на зразок луни, розрядів електрики, клацань - ці звуки можуть бути викликані навколишнім середовищем, перешкодами при з'єднанні... або тим, що хтось вас прослуховує. Якщо чуєте пульсуючий шум зі свого телефону, коли ви його не використовуєте - це може бути серйозною проблемою. Якщо ви використовуєте свій телефон в безпосередній близькості до інших електронних пристроїв (на кшталт телевізора) і він створює перешкоди на них, то це може бути наслідком наявності сторонніх пристроїв в корпусі мобільного. У більшості випадків перешкоди - це нормально, але якщо це відбувається в ті моменти, коли ви не використовуєте телефон, то це цілком може значить те, що ви «під ковпаком».

Як забезпечити безпеку мобільного пристрою?

Мобільні технології невпинно розвиваються, а разом з ними - і шкідливе програмне забезпечення. Щоб убезпечити корпоративні дані, співробітникам компанії варто дотримуватися правил кібербезпеки.

- Здійснюйте **регулярне оновлення** операційної системи та іншого програмного забезпечення для зменшення кількості уразливостей, через які зловмисники можуть інфікувати пристрої.
- Використовуйте **складні та унікальні паролі**, а також **двофакторну аутентифікацію** для захисту облікових записів Інтернет-банкінгу, пошти та соціальних мереж від несанкціонованого доступу.

• Завантажуйте **додатки перевірених розробників**. В Інтернеті Ви можете знайти детальнішу інформацію про розробника чи окремий додаток, відшукати веб-сайт чи контактні дані.

• Будьте обізнаними з можливими функціями вашого пристрою. Неуважне використання **функції Touch ID** може призвести до викрадення Ваших особистих даних або навіть коштів з банківської карти.

• Не відкривайте підозрілі файли та посилання у листах Вашої електронної пошти від невідомих користувачів. Зловмисники часто використовують інфіковані електронні листи або SMS-повідомлення для здійснення **фішингових атак**. Особливо велика кількість різних листів та повідомлень надходить у сезон знижок та акцій.

• Будьте уважними під час здійснення онлайн-покупок. Остерігайтесь занадто спокусливих пропозицій здійснити покупку за гарячою ціною — це поширена схема викрадення коштів або інфікування Вашого смартфону. За статистикою, майже кожен третій відчував тиск від пропозицій «встигни придбати тільки сьогодні».

•

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустіть браузер Інтернет (Це може бути Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera чи будь-який інший, що встановлений на вашому комп’ютері)

2. Користуючись однією з пошукових систем (Yahoo!, Google, чи будь-якою іншою) знайдіть антивірус, що може бути встановлений на ваш мобільний телефон. Запишіть до звіту назву, короткий опис, можливості та особливості даного програмного продукту.

3. Користуючись однією з пошукових систем знайдіть та запишіть до звіту резонансні випадки втрати інформації через мобільний телефон (не менше 2).

4. Користуючись однією з пошукових систем знайдіть прикладі вірусів для мобільних телефонів, запишіть до звіту їх назви, особливості, опишіть шкоду, яку вони можуть завдати пристрою.

Контрольні запитання

1. Назвіть причини втрати інформації на мобільних телефонах.
2. Що називається мобільним вірусом?
3. Що є основною метою створення та розповсюдження мобільних вірусів?
4. Назвіть та опишіть відомі вам схеми роботи мобільних вірусів.

5. Які мобільні антивіруси ви знаєте? Опишіть їх основні можливості.

6. Як здійснюється прослуховування мобільного телефону?

7. Назвіть заходи захисту мобільного телефону від прослуховування.

8. Які ви знаєте ознаки прослуховування розмов по мобільному телефоні?

9. Який антивірус може бути встановлений на ваш мобільний телефон? Які його основні властивості?

10. Які вам відомі вірусні програми, що спрямовані на ураження мобільного телефону?

ДОДАТОК. Приклад оформлення звіту.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Інститут Комп'ютерних наук та інформаційних технологій

Кафедра кібербезпеки

ЗВІТ

ПРО ЛАБОРАТОРНУ РОБОТУ №__
з дисципліни **«Комплексні системи захисту інформації»**
за темою: «Налагодження віддаленого доступу до комп’ютера»

Роботу виконав студент групи: *№групи*

_____

Викладач: _____

_____

Роботу захищено
«___» _____ 20__ р.
_____
(підпис викладача)

ХАРКІВ – 20_____

Мета роботи: навчитися встановлювати віддалене з'єднання з ПК за допомогою програми TeamViewer.

Для проведення практичної роботи використовується наступне забезпечення: два персональних комп'ютера підключених до комп'ютерної мережі, ОС Windows, програма TeamViewer.

Хід роботи

1. Інсталюємо програму TeamViewer на комп'ютер.

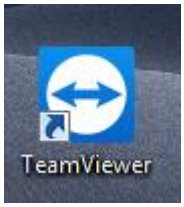

Рисунок.1.1- значок ПЗ

2. Запускаємо TeamViewer на комп'ютері, до якого будемо приєднуватися.

3. Записуємо дані, які з'являються у вікні TeamViewer на віддаленому комп'ютері в полях «Ваш ID» і «Пароль»

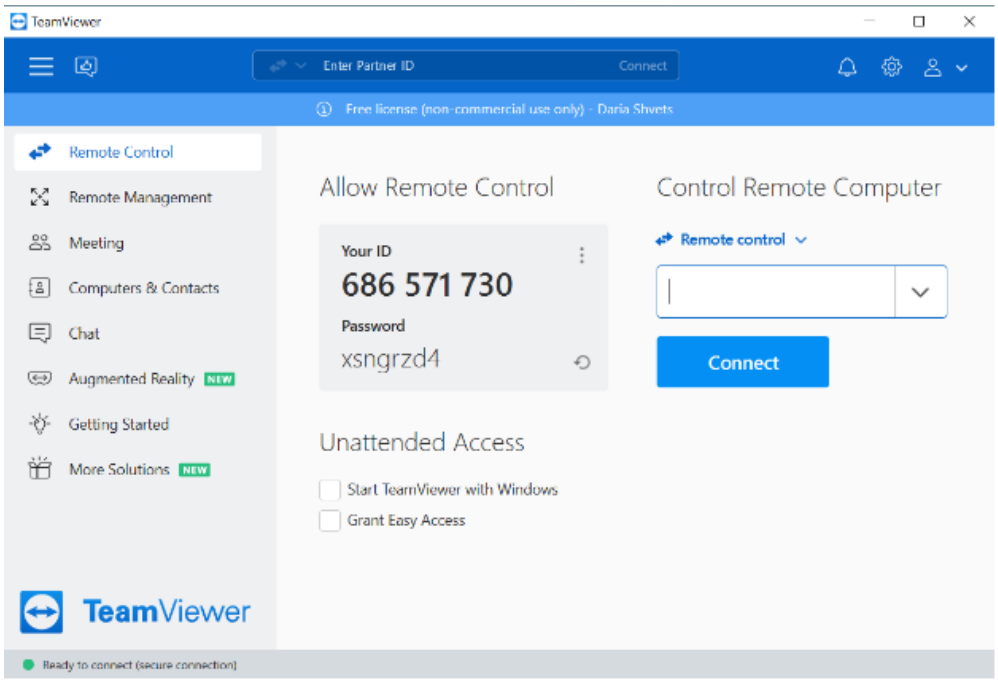

Рис.1.2.- Скріншот вікна ПЗ

4. Скріншот даного вікна поміщаємо у звіт.

5. Активував TeamViewer на своєму комп'ютері.

6. В поле ID партнера ввожу той код, який відображається в полі «Ваш ID» на віддаленому ПК. При тому має бути активна кнопка «Віддалене керування».

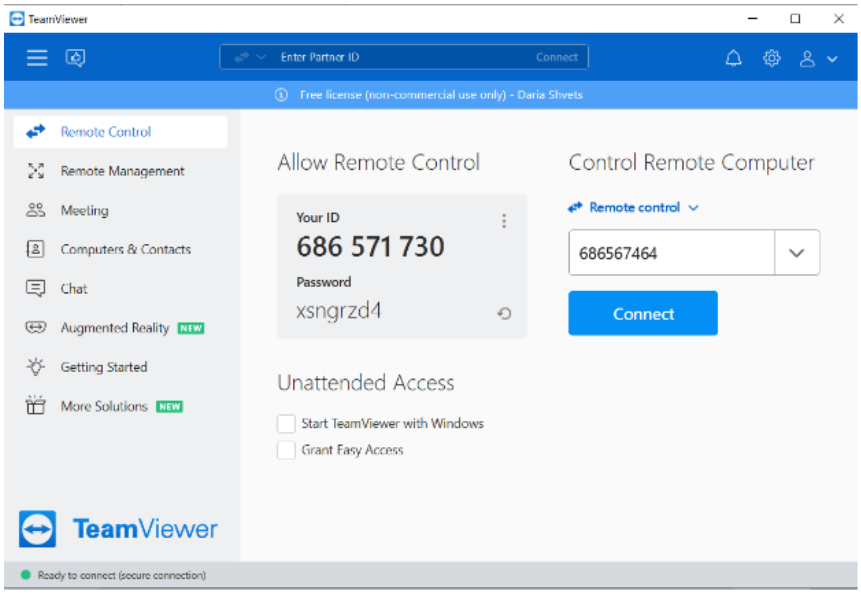

Рисунок1.3. - Активна кнопка «Віддалене керування»

8. Натисніть кнопку «Приєднатися до партнера».

9. У вікні, введіть пароль з віддаленого комп'ютера (даний код відображався в поле «Пароль» на віддаленому комп'ютере).

10. Скріншот даного вікна помістити у звіт

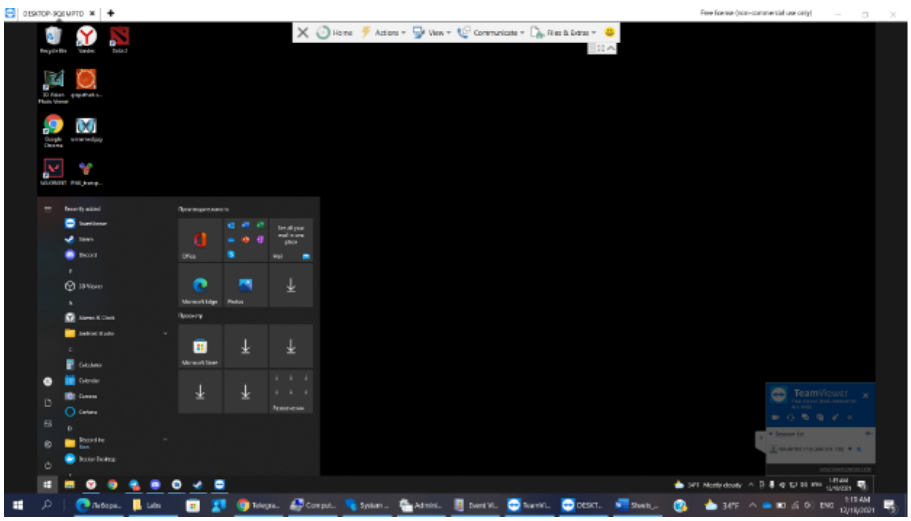

Рисунок1.4.- Скріншот вікна ПК

11. Після введення вказаного значення в єдине поле віконця натисніть кнопку «Вхід в систему». «Робочий стіл» вибраного комп'ютера зобразиться в окремому віконці на даному ПК.

12. Скріншот даного вікна помістити у звіт.

13. На робочому столі віддаленого комп'ютера натисніть «Пуск» → «Комп'ютер».

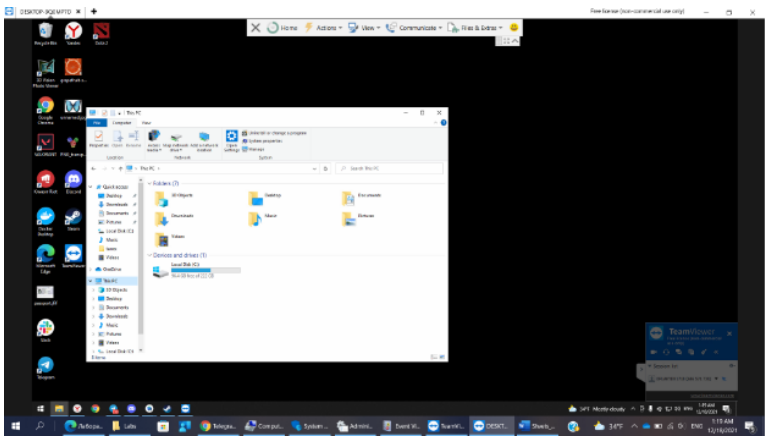

Рисунок 1.5 - Скріншот віддаленого комп'ютера

14. Скріншот даного вікна помістити у звіт

15. Деінсталюйте програму TeamViewer на комп'ютерах-по бажанню.

16. Висновки

Висновок: В цій роботі я навчився встановлювати, та доєднуватися, до комп’ютера партнера за допомгою програми TeamWiever.

Контрольні запитання: (треба дати кратку відповідь-письмово)

1.Програма TeamViewer: призначення, можливості.
2. Як в TeamViewer організована передача і копіювання файлів і папок?
3. Охарактеризуйте TeamViewer Web Connector.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O.Laptiev, V.Savchenko, G.Shuklin, O.Stefurak. Detection and blocking of means of illegal obtaining of information at objects of information activity.Kyiv. SUT. 2020. – 125 p.
2. S. Yevseiev, V. Ponomarenko, O. Laptiev, O. Milov and others. Synergy of building cybersecurity systems: monograph / Edited by– Kharkiv: PC TECHNOLOGY CENTER, 2021. – 188 p. http://monograph.com.ua/pctc/catalog/book/64
3. Богуш В. М., Кривуца В. Г., Кудін А. М., «Інформаційна безпека: Термінологічний навчальний довідник» За ред. Кривуці В. Г. — Київ. 2004. — 508 с.
4. Богуш В. М., Юдін О. К. «Інформаційна безпека держави». — К.: «МК-Прес», 2005. — 432с.
5. Вєртузаев М.С., Юрченко О.М. Захист інформації в комп'ютерних системах від несанкціонованого доступу: Навч. посібник/За ред. С.Г. Лаптєва. — К.: Видавництво Європейського університету, 2001. — 201 с.
6. ДСТУ ISO/IEC 27001: 2015 Методи захисту. Системи управління інформаційною безпекою. Вимоги (ISO/IEC 27001: 2013, IDT).
7. ДСТУ ISO/IEC 27002: 2015 Інформаційні технології. Методи захисту. Звід практик щодо заходів інформаційної безпеки (ISO/IEC 27003: 2013, IDT).
8. ДСТУ ISO/IEC 27005: 2019 Інформаційні технології. Методи захисту. Управління ризиками інформаційної безпеки (ISO/IEC 27005: 2018, IDT).
9. Закон України "Про захист інформаційно- телекомунікаційних системах". https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80
10. Закон України "Про інформацію". https://zakon.rada.gov.ua/laws/main/2657-12
11. Закон України "Про основні засади забезпечення кібербезпеки України" https://zakon.rada.gov.ua/laws/main/2163-19
12. Захист інформаційних ресурсів: навчально-методичний посібник до курсу “Захист інформаційних ресурсів” укл. С. О. Троян. – Умань : 2012.-120 с.
13. Кормич Б. А. Організаційно-правові основи політики інформаційної безпеки України: Автореф. дис. д-ра юрид. наук: 12.00.07. — Х.: НХУ України, 2004.
14. Кормич Б.А. Інформаційна безпека: організаційно-правові основи: Навч. посібник. - К.: Кондор, 2004. - 384 с.

15. Лаптєв О.А. Методологічні основи автоматизованого пошуку цифрових засобів негласного отримання інформації. – К. Міленіум. 2020 – 326 с. УДК 004.056.53. ISBN 987-966-8063-79-3.
16. Лєнков С. В., ПерегудовД. А., ХорошкоВ. А.Методи та засоби захисту інформації (в 2-ох томах). – К: Арий, 2008.
17. НД ТЗІ 1.1-003-99 Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу, введений в дію Наказом ДСТСЗІ від 28.04.1999 р. № 22
18. Організаційно-правові основи політики інформаційної безпеки України: Автореф. дис. д-ра юрид. наук: 12.00.07. — Х.: НХУ України, 2004.
19. Росоловський В.М., Анкудович Г.Г., Катерноза К.О., Шевченко М.Ю. Основи інформаційної безпеки автоматизованої інформаційної системи державної податкової служби України: Навч. посібник/За ред. М.Я. Азарова. — Ірпінь: Академія ДПС України, 2003. — 466 с.
20. Сідак В. С., Артемов В. Ю. Забезпечення інформаційної безпеки в країнах НАТО та ЄС: Навчальний посібник. — К.: КНТ, 2007.
21. Сідак В.С. Забезпечення інформаційної безпеки в країнах НАТО та ЄС: Навчальний посібник / В.С. Сідак, В.Ю. Артемов. – К. : Вид-во КНТ, 2007. Харченко В.С. Інформаційна безпека : глосарій / В.С. Харченко. – К. : Вид-во КНТ, 2005. – 13-18 с.