Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра кібербезпеки

ПРИКЛАДНА КРИПТОЛОГІЯ ТА СТЕГАНОГРАФІЯ

Методичні вказівки до лабораторних робіт
для здобувачів першого (бакалаврського) рівня вищої освіти
галузі знань F Інформаційні технології
спеціальності F5 Кібербезпека та захист інформації
денної та заочної форм навчання

Луцьк 2026

УДК 004:415.3(07)
У П-75

Рекомендовано до видання вченою радою факультету КІТ ЛНТУ,
протокол № _____ від «__ » _____ 20 26 року.

Голова вченої ради факультету КІТ _____ Інна КОНДІУС

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ
Директор бібліотеки _____ Наталія ПОЛІЩУК

Розглянуто і схвалено на засіданні кафедри кібербезпеки
ЛНТУ, протокол № _____ від « _____ » _____ 20 26 року.

Завідувач кафедри КБ _____ Сергій КОСТЮЧКО

Укладачі: _____ Микола ПОЛІЩУК, кандидат технічних наук, доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ

Рецензент: _____ Сергій ГРИНЮК, кандидат технічних наук, доцент кафедри кібербезпеки ЛНТУ

Відповідальний за випуск: _____ Сергій КОСТЮЧКО, кандидат технічних наук, доцент кафедри кібербезпеки ЛНТУ

У П-75 Прикладна криптологія та стеганографія: методичні вказівки до лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань F Інформаційні технології спеціальності F5 Кібербезпека та захист інформації денної та заочної форм навчання / уклад. М. М. Поліщук: ЛНТУ, 2026. 44 с.

Методичні вказівки до лабораторних робіт «Прикладна криптологія та стеганографія»: складене відповідно до діючої програми курсу.

Призначене для здобувачів вищої освіти всіх спеціальностей галузі знань F Інформаційні технології

М. М. Поліщук 2026

ЗМІСТ

ЛАБОРАТОРНА РОБОТА №1 ШИФРУВАННЯ 4
ЛАБОРАТОРНА РОБОТА №2 ПОНЯТТЯ ТЕОРІЇ ЧИСЕЛ 5
ЛАБОРАТОРНА РОБОТА № 3 СТВОРЕННЯ КОДІВ 9
ЛАБОРАТОРНА РОБОТА 4 ШИФРУВАННЯ І РОЗШИФРУВАННЯ ДАНИХ З ДОПОМОГОЮ OPENSSL 12
ЛАБОРАТОРНА РОБОТА №5 ШИФРУВАННЯ ТА РОЗШИФРОВКА ДАНИХ З ПІДТРИМКОЮ ХАКЕРСЬКИХ ІНСТРУМЕНТІВ 16
ЛАБОРАТОРНА РОБОТА 6 ВИВЧЕННЯ СЕАНСІВ ЗВ'ЯЗКУ ЗА ПРОТОКОЛАМИ TELNET І SSH З ДОПОМОГОЮ ПРОГРАМИ WIRESHARK 24
ЛАБОРАТОРНА РОБОТА №7 ХЕШУВАННЯ 29
ЛАБОРАТОРНА РОБОТА №8 СХОВИЩА ЦЕНТРІВ СЕРТИФІКАЦІЇ 33

ЛАБОРАТОРНА РОБОТА №1
ШИФРУВАННЯ

Мета роботи: вивчити алгоритмів шифрування та навчитись шифрувати дані.

Задача 1. При шифруванні слова МАЙДАН підстановкою з ключем воно змінюється на слово ТУРКУБ. Якщо слово за допомогою цієї підстановки зашифрувати ще раз, то отримаємо слово ҐФЙЦФП, а після третього циклу шифрування – ЖАРЩАЮ. На якому циклі шифрування знову побачимо слово МАЙДАН, якщо кількість виконаних циклів шифрування не обмежити?

Р о з в’ я з а н н я. Кількість різних слів, що можна дістати із початкового слова МАЙДАН, збігається з найменшим номером циклу шифрування, на якому знову виникне початкове слово. Буква М повторюється в кожному циклі, номер якого є кратним 5, буква А – у кожному циклі з номером, що є кратним 3, а букви Й, Д, Н – у циклах з номерами, які є кратними 2; 12; 11 відповідно. Якщо підстановку розкласти в добуток незалежних циклів, то порядок підстановки дорівнюватиме найменшому спільному кратному довжин циклів [1-3]. Тоді слово МАЙДАН вперше з’явиться в циклі під номером, який дорівнює найменшому спільному кратному НСК(2,3,5,12,11) = 660.

Таблиця 1.1 – шифрування алфавіту

А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
У	П	Ь	Х	Ж	К	Г	И	Ш	Я	С	Н	Д	Р	Ц	О	Т	Б	Е	Ю	Й	Ї	Ґ	Ф	А	Ч	Щ	І	М	З	Є	Л	В

Завдання. Зашифрувати своє прізвище згідно таблиці 1.1 та визначити в якому циклі знову з’явиться ваше прізвище.

ЛАБОРАТОРНА РОБОТА №2
ПОНЯТТЯ ТЕОРІЇ ЧИСЕЛ

Мета роботи: Ознайомитися з основними поняттями теорії чисел.

Використовуване програмне забезпечення: пакети математичних обчислень MathCad і Maple.

1.1 Завдання на лабораторну роботу

1. Представити власне прізвище записане латиницею у вигляді десяткового числа (див. Теоретичні відомості).

2. Перетворити десяткове число T на текст.

3. Виписати в конспект всі прості числа від 2 до 99.

4. Виписати в конспект таблицю відповідності шістнадцятеричних цифр десятковим числам та двійковим тетрадам (наприклад, 2 1016 1000 88 , 2 1016 1100 12C).

5. Знайти серед чисел a, b, c, d пари взаємно простих чисел

6. Знайти) ,(b aНСД ,) ,,(c baНСД ,) ,,,(d cbaНСД .

7. Знайти серед чисел a,b,c,d пари чисел, порівняних за модулем m.

8. Знайти суму і добуток чисел a, b, c, d за модулем m.

9. Обчислити значення функції Ейлера для кожного з чисел a,b,c,d .

Значення параметрів T,a,b,c,d ,m візьміть із таблиці 2.1 згідно з номером варіанту N.

1.2 Зміст звіту

1. Титульний лист, тема і мета роботи.

2. Результати обчислень.

3. Висновки по роботі.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Перетворення між текстом та числами

Криптографія вивчає методи та засоби перетворення інформації з відкритого вигляду у закритий (зрозумілий лише тим, хто володіє ключем). Інформація, як правило, подається в текстовому вигляді. Тим не менш, більшість

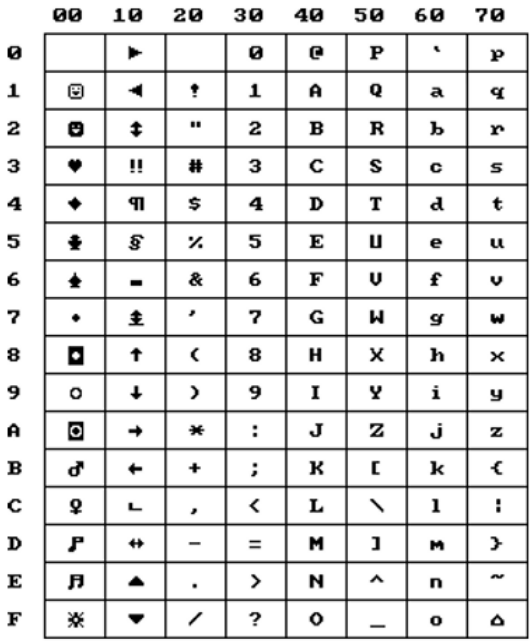
криптографічних алгоритмів оперують числовими величинами [2-3]. Таким чином потребується попередня обробка тексту для представлення його у вигляді числа – кодування.

Таблиця 2.1 – Варіанти завдань

N	*T*	*a*	*b*	*c*	*d*	*m*
1	60093118633877632121	45	37	27	11	17
2	88772934743480459988660078	76	24	19	33	9
3	22066599077835371	41	26	15	69	11
4	4859222852730381682	42	36	70	53	14
5	22643820939338093	71	52	39	87	15
6	5291294591267267956	18	67	35	49	4
7	24604245826172787	90	45	28	73	14
8	12068496994072772896621	35	23	92	15	8
9	5792036358531605868	64	79	39	22	6
10	91712171958636	77	42	59	60	17
11	1483985444626882588530	34	63	45	29	11
12	5071443942490206059	30	78	61	55	12
13	23775192852751730	51	91	43	27	8
14	3606962798869438854232225	25	31	46	50	7
15	5075397850727345260	47	82	28	39	9

При кодуванні кожному символу тексту надається певний унікальний цифровий код. В сучасних інформаційних технологіях використовуються багато способів кодування: ASCII (American Standard Code for Information Interchange), Unicode тощо. Для перетворення тексту на число кожна літера або знак замінюється на запис у цифрах, згідно з таблицею кодування. Стандартна таблиця кодування ASCII наведена в таблиці 2.1.

Таблиця 2.2 – Таблиця ASCII-кодів



Приклади:

Перетворимо на десяткове число повідомлення «Student» (без лапок). Для цього замінимо кожну літеру відповідним кодом з таблиці А.1. і переведемо отримане шістнадцятеричне число у десяткове згідно з правилами перетворення між системами числення:

$$\text{Student}=53747564656E74_{16}=23490470611349108_{10}$$

Для зворотного перетворення необхідно перевести десяткове число у шістнадцятеричне і замінити кожну пару цифр на відповідний символ з таблиці кодування.

Знайдемо, яке повідомлення записується десятковим числом 1515082837.

$1515082837_{10}=5A4E5455_{16}=ZNTU$

Допоміжні функції пакету MAPLE

convert (r,her) – перетворює десяткове число r в 16-ве

$r := 30; a := convert\ (r, her); r := 30\ a := 1E$

convert (S,decimal,hex) – перетворює 16-ве число S в десяткове

$r := convert\ (\text{"1E"}\ , decimal, hex); r := 30$

$r := convert\ (\text{"1E2A"}\ , decimal, hex); r := 7722$

convert (z,binary) – перетворює десяткове число z в двійкове

$z := 19; z2 := convert\ (z, binary);\ z := 19 \quad z2 := 10011$

convert (z, base, 2) – перетворює десяткове число z в двійковий масив

$zar := convert\ (z, base, 2);\ zar := [1,1,0,0,1]$

ЛАБОРАТОРНА РОБОТА № 3
СТВОРЕННЯ КОДІВ

МЕТА: створювати і шифрувати повідомлення з допомогою інтерактивних засобів.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Загальні відомості і сценарій. Існує кілька алгоритмів шифрування, які можна використовувати для шифрування і розшифровки повідомлень. Віртуальні приватні мережі (VPN) широко використовуються для автоматизації процесу шифрування і розшифровки. У цій лабораторній роботі ви і ваш партнер по лабораторної роботі будете використовувати інтерактивний інструмент для шифрування і розшифровки повідомлень [4].

Необхідні ресурси: ПК з доступом до Інтернету.

ХІД РОБОТИ

1. Пошук інтерактивного засобу шифрування і дешифрування. У сучасних мережах використовується багато алгоритмів шифрування різних типів. Одним з найбільш безпечним є симетричний алгоритм блочного шифрування (AES). Потрібно використовувати цей алгоритм в нашій демонстрації.

2. У веб-браузері виконайте пошук " encrypt decrypt AES online ". У результатах пошуку будуть перераховані різні інструменти.

3. Вивчіть різні посилання і виберіть інструмент. У нашому прикладі ми використовували засіб, яке можна отримати за наступним адресою: http://aesencryption.net/

4. Зашифруйте повідомлення і відправте його по електронній пошті вашим одногрупникам. На цьому етапі кожен зашифрує повідомлення і відправляє зашифрований текст іншому одногрупнику.

Примітка. Незашифровані повідомлення називаються простим текстом, а зашифровані повідомлення називаються криптограмами.

5. Введіть будь-який незашифрований текст в текстовому полі. Повідомлення може бути дуже коротким або довгим. Переконайтеся, що одногрупник НЕ бачить відкритий текст повідомлення.

Зазвичай для шифрування повідомлень потрібно секретний ключ (наприклад, пароль). Секретний ключ використовується разом з алгоритмом шифрування для шифрування повідомлень. Тільки користувач, який знає секретний ключ, зможе розшифрувати повідомлення.

6. Введіть секретний ключ. Деякі інструменти можуть попросити підтвердити пароль. У нашому прикладі ми використовували секретний ключ cyberops (рис. 3.1).

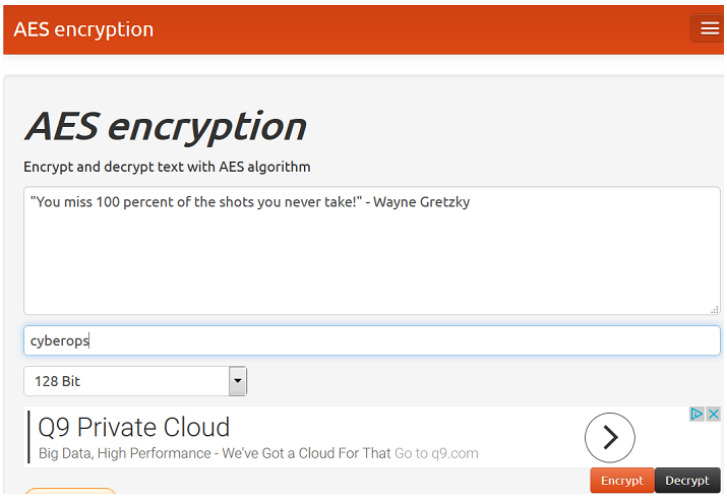


Рисунок 3.1 – Засіб шифрування

7. Потім натисніть Encrypt (Шифрувати). У вікні Result of encryption in base64 (Результат шифрування в base64 представлено на рисунку 3.2) відображається довільний текст. Це зашифроване повідомлення .



Рисунок 3.2 – Результат шифрування в base64

8. Скопіюйте або завантажте те, що вийшло в результат повідомлення.

9. Надішліть зашифроване повідомлення по електронній пошті одногрупникам.

10. Розшифруйте криптограму. Алгоритм AES представляє собою симетричний алгоритм шифрування. Це означає, що обидві сторони, що обмінюються зашифрованими повідомленнями, повинні заздалегідь мати секретний ключ.

11. Відкрийте електронне повідомлення від свого одногрупника.

12. Скопіюйте криптограму і вставте її в текстове поле.

13. Введіть вже наявний у вас загальний секретний ключ як представдено на рисунку 3.3.

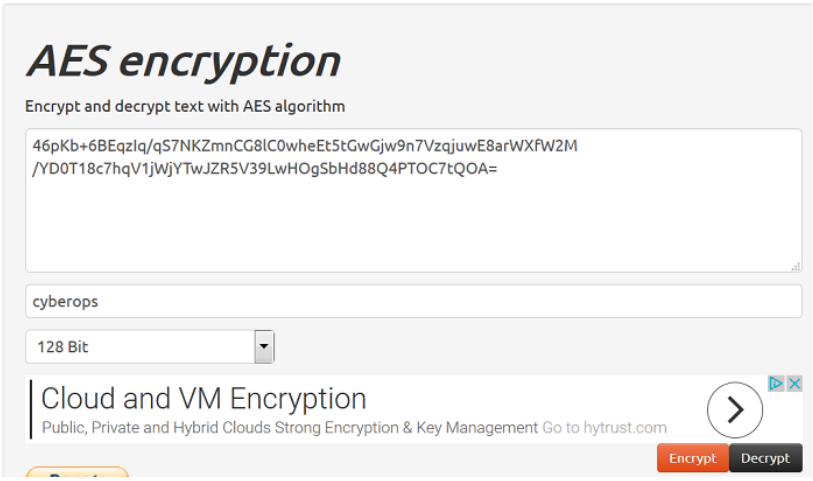


Рисунок 3.3 – Введення вже наявного загального секретного ключа

14. Клацніть Decrypt (Розшифрувати); має відображатися вихідне повідомлення відкритим текстом (рис. 3.4).

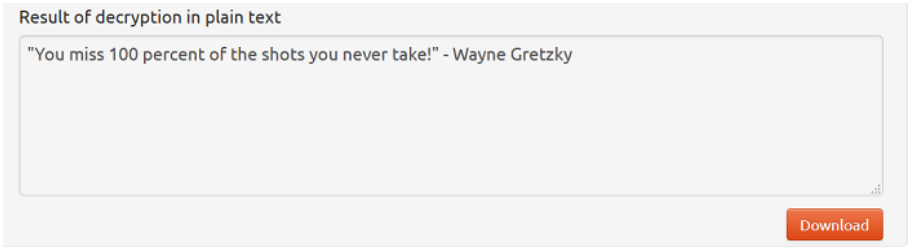


Рисунок 3.4 – Вихідне повідомлення відкритим текстом

ЛАБОРАТОРНА РОБОТА 4
ШИФРУВАННЯ І РОЗШИФРУВАННЯ ДАНИХ З ДОПОМОГОЮ OpenSSL

Мета: навчитись шифрування та розшифрування повідомлень з допомогою OpenSSL

ТЕОРЕТИЧНІ ВІДОМОСТІ

OpenSSL - проект з відкритим вихідним кодом, який забезпечує надійний і повнофункціональний набір комерційних інструменті для протоколів TLS і SSL. Це також криптографічний бібліотека загального призначення. У цій лабораторній роботі ви будете використовувати OpenSSL для шифрування і розшифровки тексту повідомлень [5-6].

Примітка. Хоча в даний час OpenSSL є фактично стандартною криптографічного бібліотекою , приклад використання , представлений в цієї лабораторної роботи , що не рекомендується , якщо потрібно надійна захист. Нижче представлені дві проблеми безпеки , властиві цій лабораторній роботі.

1) У методі, описаному в цій лабораторній роботі , використовується слабка функція формування ключа. Єдине засіб забезпечення безпеки – це дуже надійний пароль.

2) Метод, описаний в цій лабораторній роботі, що не гарантує цілісності текстового файлу. Цю лабораторну роботу слід використовувати тільки в навчальних цілях. *Методи , представлені тут , НЕ повинні використовуватися для захисту дійсно конфіденційних даних* .

Необхідні ресурси:

- Віртуальна машина робочої станції CyberOps
- Доступ до Інтернету

Завдання 1

Шифрування повідомлень з допомогою OpenSSL

OpenSSL можна використовувати як автономне засіб для шифрування . Можна використовувати безліч алгоритмів шифрування , але в цій лабораторній роботи обраний AES. Для того щоб використовувати AES для шифрування

текстового файлу безпосередньо з командного рядка з допомогою OpenSSL , виконайте наступні дії:

1. Шифрування текстового файлу.

1.1. Виконайте вхід в віртуальну машину робочої станції CyberOps.

1.2. Відкрийте вікно терміналу.

1.3. Текстовий файл, який шифрується, знаходиться в каталозі */ home / analyst / lab.support.files /,* тому перейдіть в цей каталог.

[Analyst @ secOps ~] $ cd ./ lab.support.files /

[analyst @ secOps lab.support.files] $

1.4. Введіть наведену нижче команду для відображення вмісту зашифрованого текстового файлу letter_to_grandma.txt на екрані.

[Analyst @ secOps lab.support.files] $ cat letter_to_grandma.txt

Hi Grandma , (Привіт , бабуся !)

I am writing this letter to thank you for the chocolate chip cookies you sent me. (Я пишу це лист, щоб подякувати тебе за шоколадне печиво , яке ти мені прислала.) I got them this morning and I have already eaten half of the box ! (Я отримав їх вранці і вже з'їв половину коробки!) They are absolutely delicious ! (Вони дуже смачні !)

I wish you all the best . (Бажаю тобі всього найкращого .) Love , (С любов'ю ,) Your cookie-eater grandchild . (Твій онук, який дуже любить печиво .)

[Analyst @ secOps lab.support.files] $

1.5. З того ж вікна терміналу виконайте наведену нижче команду для шифрування текстового файлу. В команді буде використаний алгоритм AES-256 для шифрування текстового файлу , зашифрована версія зберігається в файлі message.enc. OpenSSL запросить пароль і підтвердження пароля. Введіть викликана пароль і обов'язково запам'ятайте пароль.

[Analyst @ secOps lab.support.files] $ openssl aes-256-cbc - in letter_to_grandma.txt - out message.enc

Введіть пароль шифрування: aes-256-cbc.

Перевірка - введіть пароль шифрування: aes-256-cbc.

[Analyst @ secOps lab.support.files] $

Збережіть пароль.

1.6. Після завершення процесу , щоб відобразити вміст файлу message.enc, знову використовуйте команду cat .

[Analyst @ secOps lab.support.files] $ cat message.enc

Питання: Правильно чи відображається вміст файлу message.enc ? Як воно виглядає ? Дайте пояснення .

1.7. Для того щоб зробити файл придатним для читання , виконайте команду OpenSSL повторно, але на цей раз додайте параметр -a . Параметр -a повідомляє OpenSSL , що потрібно закодувати зашифроване повідомлення з допомогою іншого способу кодування Base64 перед збереженням результатів у файлі.

Примітка. Base64 представляє собою групу схожих схем перетворення в текст довічних даних , використовуваних для подання двійкових даних в форматі рядків ASCII.

[Analyst @ secOps lab.support.files] $ openssl aes-256-cbc -a - in letter_to_grandma.txt - out message.enc

Введіть пароль шифрування aes-256-cbc.

Перевірка - введіть пароль шифрування aes-256-cbc.

1.8. Знову використовуйте команду cat для відображення вмісту знову створеного файлу message.enc .

Примітка . Вміст message.enc буде різним .

[Analyst @ secOps lab.support.files] $ cat message.enc

U2FsdGVkX19ApWyrn8RD5zNp0RPCuMGZ98wDc26u / vmj1zyDXobGQhm / dDRZasG7
rfnth5Q8NHValEw8vipKGM66dNFyyr9 / hJUzCoqhFpRHgNn + Xs5 + TOtz / QCPN1bi
08LGTSzOpfkg76XDCk8uPy1hl / + Ng92sM5rgMzLXfEXtaYe5UgwOD42U / U6q73pj
a1ksQrTWsv5mtN7y6mh02Wobo3A1ooHrM7niOwK1a3YKrSp + ZhYzVTrtksWDl6Ci
XMufkv + FOGn + SoEEuh7l4fk0LIPEfGsExVFB4TGdTiZQApRw74rTAZaE / dopaJn0
sJmR3 + 3C + dmgzZIKEHWsJ2pgLvj2Sme79J / XxwQVNpw =

[Analyst @ secOps lab.support.files] $

Питання: Відображається чи message.enc правильно тепер? Дайте пояснення. Які переваги кодування файлу message.enc з використанням Base64 ви можете назвати ?

2. Розшифровка повідомлень з допомогою OpenSSL. З допомогою схожої команди OpenSSL можна розшифрувати файл message.enc.

2.1. Для розшифровки message.enc використовуйте наведену нижче команду.

[Analyst @ secOps lab.support.files] $ openssl aes-256-cbc -a -d - in message.enc - out decrypted_letter.txt

2.2. OpenSSL запросить пароль, який використовувався для шифрування. Введіть пароль ще раз.

2.3 Після розшифровки з допомогою OpenSSL файлу message.enc розшифроване повідомлення зберігається в текстовому файлі з ім’ям decrypted_letter.txt. Використовуйте cat для показу вмісту decrypted_letter.txt.

[Analyst @ secOps lab.support.files] $ cat decrypted_letter.txt

Питання: Правильно чи розшифровано лист? Команда, яка використовується для розшифровки , також містить параметр -a. Поясніть чому .

ЛАБОРАТОРНА РОБОТА №5
ШИФРУВАННЯ ТА РОЗШИФРОВКА ДАНИХ З ПІДТРИМКОЮ ХАКЕРСЬКИХ ІНСТРУМЕНТІВ

Мета: навчитись створювати і шифрувати файли; навчитись відновлювати пароль зашифрованого ZIP-файлу.

ЗАГАЛЬНІ ВІДОМОСТІ ТА СЦЕНАРІЇ

Що ви робите, якщо ви працюєте в крупній корпорації, в якій діє корпоративна політика відносно з'ємних носіїв? В загальному, вказано, що тільки зашифровані ZIP-документи можна скопіювати на портативних USB-носіїв.

У цьому сценарії фінансовий директор виїхав з міста у відрядження і зв'язався з вами, надіславши запит невідкладну допомогу. Будучи у відрядженні поза містом, він спробував розпакувати важливі документи з зашифрованого ZIP-файлу на USB-накопичувачі. Однак пароль, наданий для відкриття ZIP-файл, недійсний. Фінансовий директор зв'язався з вами, щоб з'ясувати, можете. Чи ви щось зробити в цій ситуації [7].

Примітка. Наданий сценарій простий і служить тільки як приклад.

Іноді доступні інструменти для відновлення втрачених паролів. Це особливо вірно в ситуаціях, аналогічних описаній, коли аналітик з кібербезпеки може отримати корисні відомості від фінансового директора, наприклад про довжину пароля і уявлення про те, яким він може бути. Знання відповідної інформації значно полегшує спроби відновлення паролів.

Серед програм для відновлення паролів - hashcat, John the Ripper, Lophtcrack і інші. У нашому випадку ми будемо використовувати fcrackzip - це проста утиліта Linux для відновлення паролів зашифрованих ZIP-файлів.

Пам'ятайте, що ці ж інструменти можуть бути використані кіберзлочинцями для виявлення невідомих паролів. Хоча у них немає доступу до деяких корисним відомостями, витративши якийсь час, можна виявити паролі для відкриття зашифрованих ZIP-файлів. Кількість затраченого часу залежить

від складності та довжини пароля. Довші і більш складні паролі (поєднання різних типів символів) більш надійні.

У цій лабораторній роботі вам запропоновано:

- створити та зашифрувати образів текстових файлів;
- розшифрувати зашифрованний ZIP-файл.

Примітка. Цю лабораторну роботу слід використовувати тільки в навчальних цілях. методи, представлені тут, НЕ повинні використовуватися для захисту дійсно конфіденційних даних.

Необхідні ресурси:

- Віртуальна машина робочої станції CyberOps;
- Доступ до Інтернету.

СТВОРЕННЯ ТА ШИФРУВАННЯ ФАЙЛІВ

У цій частині лабораторної роботи ви повинні створити кілька текстових файлів, які будуть використовуються для створення зашифрованих ZIP-файлів на наступних кроках:

1. Створення текстових файлів:

1.1 Завантажте віртуальну машину робочих станцій CyberOps.

1.2 Відкрийте вікно терміналу. Переконайтеся, що ви перебуваєте в головному каталозі analyst. В іншому випадку введіть cd ~ в командному рядку терміналу.

1.3 Створіть новую папку з іменем Zip-Files, використовуючи команду mkdir Zip-Files.

1.4 Перейдіть у цей каталог, використовуючи команду CD Zip-Files.

1.5 Введіть наступні команди для створення трьох текстових файлів, як на рисунку 5.1:

[analyst @ secOps Zip-Files] $ echo Це зразок текстового файлу> sample-1.txt
[analyst @ secOps Zip-Files] $ echo Це зразок текстового файлу> sample-2.txt
[analyst @ secOps Zip-Files] $ echo Це зразок текстового файлу> sample-3.txt

1.6 Збережіть, що створили файли, за допомогою команд ls.

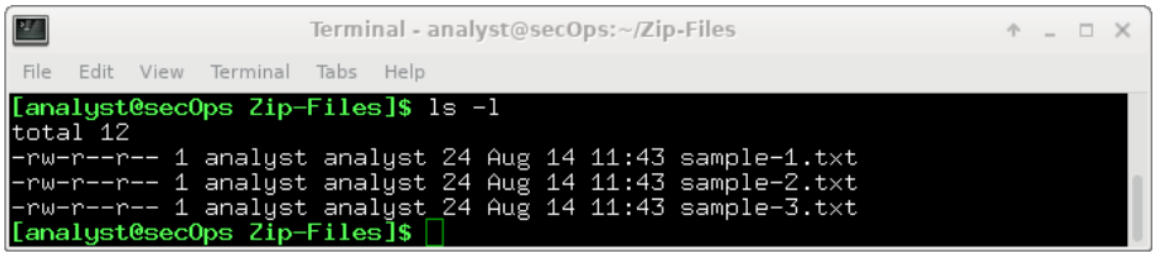


Рисунок 5.1 – Команди для створення трьох текстових файлів

2. Запакуйте в архіві та зашифруйте текстові файли.

Потім ми створимо кілька зашифрованих ZIP-файлів з використанням паролів різної довжини. Для цього всі три текстові файли будуть зашифровані за допомогою службової програми zip.

2.1 Створіть зашифрований ZIP-файл з ім’ям file-1.zip, що містить три текстові файли за допомогою наступної команди, як на рисунку 5.2:

*[analyst @ secOps Zip-Files] $zip –e файл-1.zip зразок **

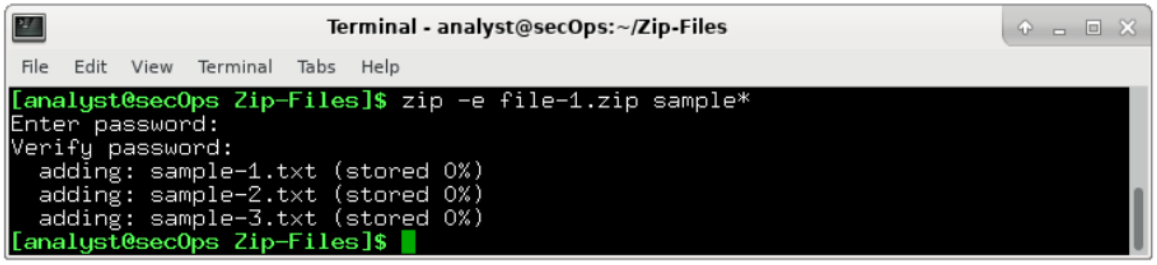


Рисунок 5.2 – Зашифрований ZIP-файл, що містить три текстові файли

2.2 У відповідь на запит пароля введіть «Односимвольний пароль» на свій вибір. У цьому прикладі введена буква B. Введіть ту ж букву коли з’явиться відповідний запит підтвердження.

2.3 Повторити процедуру створення 4 інших файлів:

– file-2.zip з двосимвольним паролем по вашому вибору. У нашому прикладі використовується R2;
– file-3.zip з трехсимвольним паролем по вашому вибору. У нашому прикладі використовується 0B1;

– file-4.zip з 4-символьним паролем по вашому вибору. У нашому прикладі використовується Y0Da;
– file-5.zip з 5-символьним паролем по вашому вибору. У нашому прикладі використовується C-3P0.

2.4 Виконайте, що всі ZIP-файли були створені за допомогою команд ls –lf * (рис. 5.3).

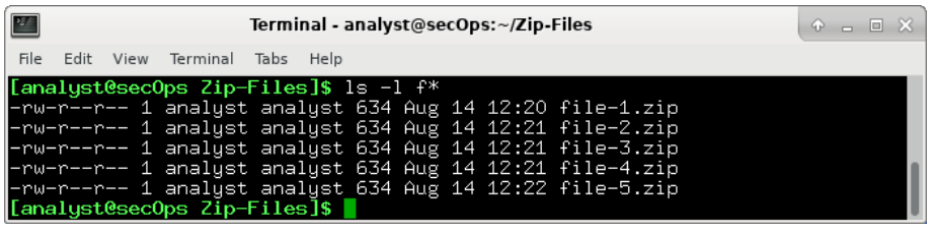


Рисунок 5.3 – Створення ZIP-файлів

2.5 Спробуйте відкрити zip-файл із підтримкою неправильного пароля, як показано на рисунку 5.4 та за допомогою команди:

[analyst @ secOps Zip-Files] $ unzip file-1.zip

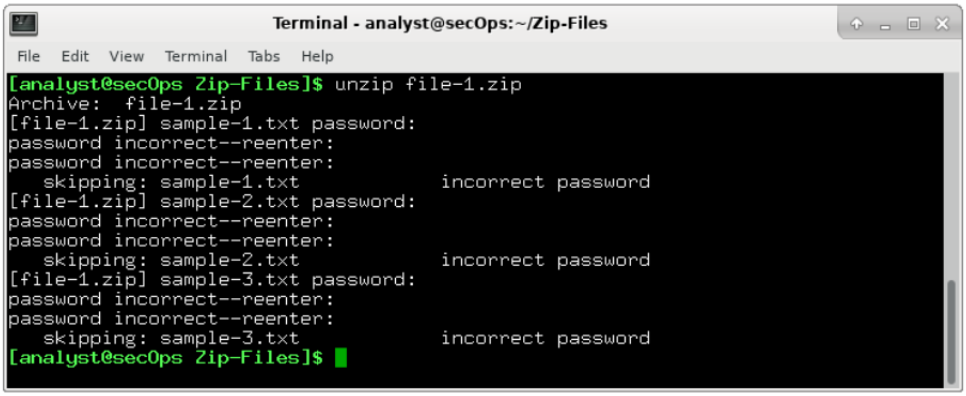


Рисунок 5.4 – Відкриття zip-файлів

ВІДНОВЛЕННЯ ПАРОЛІВ ЗАШИФРОВАНИХ ZIP-ФАЙЛІВ

У цій частині лабораторної роботи втрачені паролі з зашифрованих ZIP-файлів будуть відновлені за допомогою службової програми fcrackzip. Fcrackzip виконує пошук в кожному ZIPфайле, наданому для зашифрованих файлів, і намагається вгадати пароль методом повного перебору.

ZIP-файли з паролями різної довжини були створені, щоб з'ясувати, як довжина пароля впливає на час, необхідний, щоб розкрити пароль:

1. Введення в fcrackzip

1.1 У вікні терміналу введіть команду fcrackzip -h для перегляду відповідних параметрів команди (рис. 5.5).

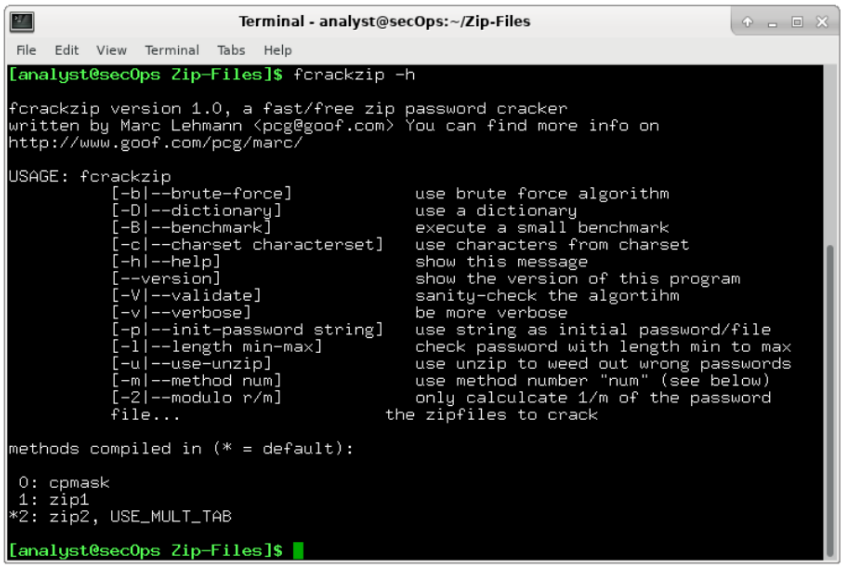


Рисунок 5.5 – Введення команди fcrackzip у вікно терміналу

У наших прикладах будуть використані параметри команд —v-u і -l. Параметр - l буду вказані останнім, то що він задає довгу паролю. Також можна по експериментувати з іншими власними.

2. Встановлення пароля з підтримкою fcrackzip

2.1 Тепер спробуйте відновити пароль файлу file-1.zip. Нагадуємо, що він був зашифрований з використанням пароля довжиною в один символ. Тому використовуйте наступну команду fcrackzip (рис. 5.6):

[analyst @ secOps Zip-Files] $ fcrackzip -vul 1-4 file-1.zip

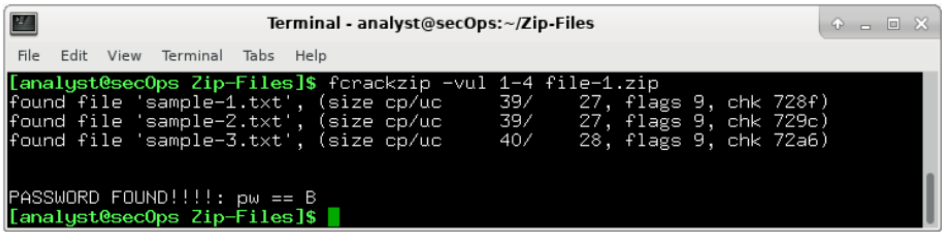


Рисунок 5.6 – Використання команди fcrackzip

Примітка. Можливо, була задана довжина пароля менш 1-4 символів. Скільки часу потрібно для розкриття пароля?

2.2 Тепер потрібно спробувати відновити пароль файлу file-2.zip. Нагадуємо, що для шифрування використовувався однорядкове резюме пароль. Тому використовується наступна команда fcrackzip (рис. 5.7):

[analyst @ secOps Zip-Files] $ fcrackzip –vul 1-4 file-2.zip

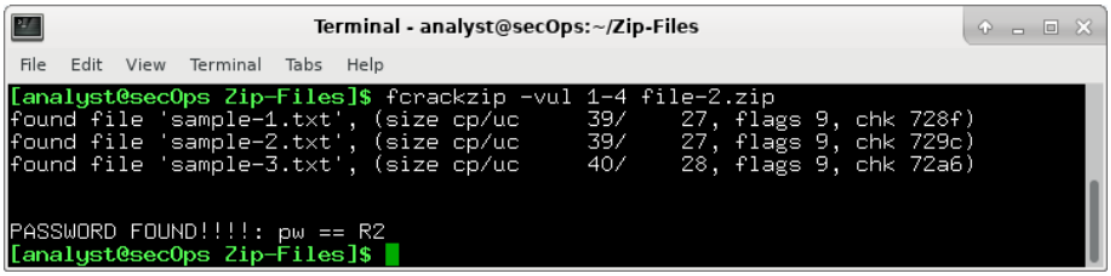


Рисунок 5.7 – Відновлення пароля файлу file-2.zip

2.3. Повторіть процедуру і відновіть пароль файлу file-3.zip. Нагадуємо, що для шифрування використовувався трьохсимвольний пароль. Виміряйте час, щоб з’ясувати, як довго відбувається розкриття трьохсимвольного пароля. Використовуйте наступну команду fcrackzip:

[analyst @ secOps Zip-Files] $ fcrackzip –vul 1-4 file-3.zip

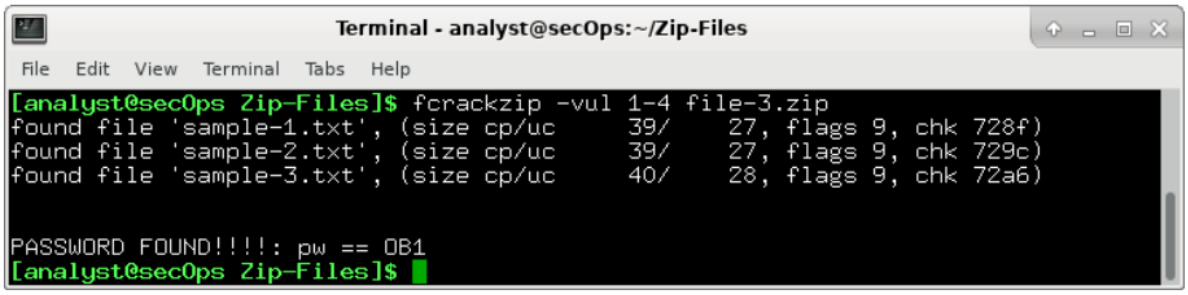


Рисунок 5.8 – Повторіть процедурт відновлення паролю файлу file-3.zip

Скільки часу потрібно для розкриття пароля?

2.4. Скільки часу потрібно для злому паролів з чотирьох символів? Повторіть процедуру і відновите пароль файлу file-4.zip. Виміряйте час, щоб

з’ясувати, як довго відбувається розкриття пароля за допомогою наступної команди fcrackzip (рис. 5.8):

[analyst @ secOps Zip-Files] $ fcrackzip –vul 1-4 file-4.zip

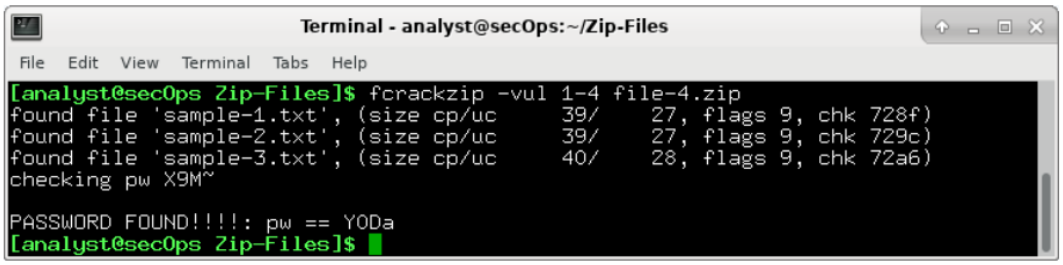


Рисунок 5.8 – Розкриття пароля файлу file-4.zip за допомогою команди fcrackzip

Скільки часу потрібно для розкриття пароля?

2.5. Скільки часу потрібно для злому паролів з п’яти символів? Повторіть процедуру і відновіть пароль файлу file-5.zip. Довжина пароля становить 5 символів, тому необхідно призначити параметру команди -l значення 1-5. Ще раз дізнаємося, скільки часу займе виявлення пароля за допомогою наступної команди fcrackzip (рис. 5.9):

[analyst @ secOps Zip-Files] $ fcrackzip –vul 1-5 file-5.zip

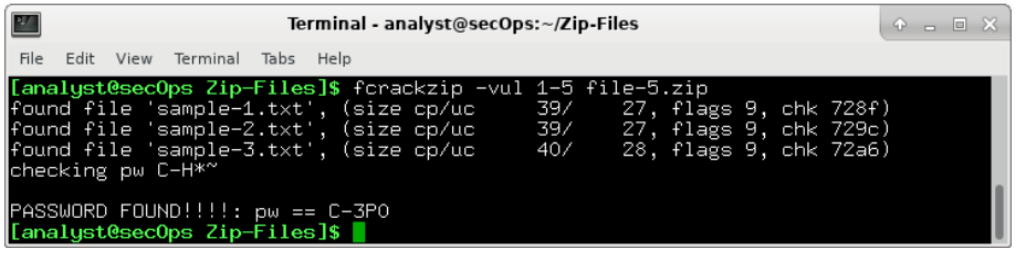


Рисунок 5.9 – Розкриття пароля файлу file-5.zip за допомогою команди fcrackzip

Скільки часу потрібно для розкриття пароля?

2.6 Відновіть 6-символьний пароль за допомогою команди fcrackzip. Схоже, для розкриття більш довгих паролів потрібно більше часу, тому вони виявляються більш безпечними. Однак 6-символьний пароль не відлякає кіберзлочинця [8-9].

Як ви вважаєте, скільки часу буде потрібно для розкриття 6-символьного пароля за допомогою fcrackzip?

Для того щоб відповісти на це питання, створіть файл з ім’ям file-6.zip з використанням 6 символьного пароля за своїм вибором. У нашому прикладі використаний JarJar:

*[analyst @ secOps Zip-Files] $ zip –e файл-6.zip зразок **

Повторіть процедуру, щоб відновити пароль файлу file-.zip з використанням наступної команди fcrackzip:

[analyst @ secOps Zip-Files] $ fcrackzip –vul 1-6 file-6.zip

Скільки часу потрібно для розкриття пароля за допомогою fcrackzip?

Проста істина полягає в тому, що більш довгі паролі більш надійні, так як потрібно більше часу, щоб їх розкрити.

Яку довжину ви рекомендуєте для надійного пароля?

ЛАБОРАТОРНА РОБОТА 6
ВИВЧЕННЯ СЕАНСІВ ЗВ'ЯЗКУ ЗА ПРОТОКОЛАМИ TELNET І SSH З ДОПОМОГОЮ ПРОГРАМИ WIRESHARK

Мета: вивчення сеансу Telnet з допомогою програми Wireshark; вивчення сеансу SSH допомогою програми Wireshark

ЗАГАЛЬНІ ВІДОМОСТІ І СЦЕНАРІЙ

В ході цієї лабораторної роботи вам належить налаштувати маршрутизатор для підключень по протоколу SSH і використовувати програму Wireshark для перехоплення і перегляду даних, що передаються у час сеансів Telnet і SSH. Ви побачите, наскільки важливо шифрування даних з допомогою SSH [10-11].

Необхідні ресурси: Віртуальна машина робочої станції CyberOps.

Вивчення сеансу Telnet з допомогою програми Wireshark

Вам потрібно скористатися програмою Wireshark для перехоплення і перегляду даних, що передаються у час сеансу зв'язку з маршрутизатором по протоколу Telnet.

1. Виконайте перехоплення даних .

1.1. Запустіть і виконайте вхід в віртуальну машину робочої станції CyberOps (ім'я користувача: analyst / пароль: cyberops).

1.2. Відкрийте вікно терміналу і запустіть програму Wireshark. Натисніть кнопку ОК для продовження після прочитання застережливого повідомлення.

[Analyst @ secOps analyst] ***$ sudo wireshark-gtk***

[sudo] password for analyst: ***cyberops***

** (wireshark-gtk: 950): WARNING **: Could not connect to accessibility bus: Failed to connect to socket / tmp / dbus -REDRWOHelr: Connection refused

(ПОПЕРЕДЖЕННЯ **. Чи не вдається підключитися до шині доступності. Неможливо підключитися до сокету / tmp / dbus-REDRWOHelr. В підключенні відмовлено.)

Gtk-Message: GtkDialog mapped without a transient parent (GtkDialog зіставлений без проміжного батька) Це НЕ рекомендується.\

1.3 Запустіть перехоплення даних програмою Wireshark на інтерфейсі *Loopback : lo*

1.4 Відкрийте інше вікно терміналу. Почніть сеанс підключення до localhost по протоколу Telnet. За запитом введіть ім’я користувача analyst і пароль cyberops. Зверніть увагу , що може пройти кілька хвилин, перш ніж буде виконано підключення до localhost і з’явиться вікно входу в систему:

[Analyst @ secOps ~] ***$ telnet localhost***

Trying :: 1 ...

Connected to localhost.

Escape character is ^]’.

Linux 4.10.10-1-ARCH (unallocated.barefruit.co.uk) (pts / 12)

secOps login: analyst

Пароль: Last login: Fri Apr 28 10:50:52 from localhost.localdomain

[analyst @ secOps ~] $

1.6. Зупиніть перехоплення даних програмою Wireshark після того, як будуть введені облікові дані користувача.

Вивчіть сеанс Telnet:

1.7. Застосуйте фільтр, який показує тільки трафік по протоколу Telnet . Введіть T eln et в поле фільтра і натисніть Apply (Застосувати).

1.8. Клацніть правою кнопкою миші одну з рядків T elnet в розділі Packet list (Список пакетів) програми Wireshark і виберіть в спадному списку пункт Follow TCP Stream (рис. 6.1).

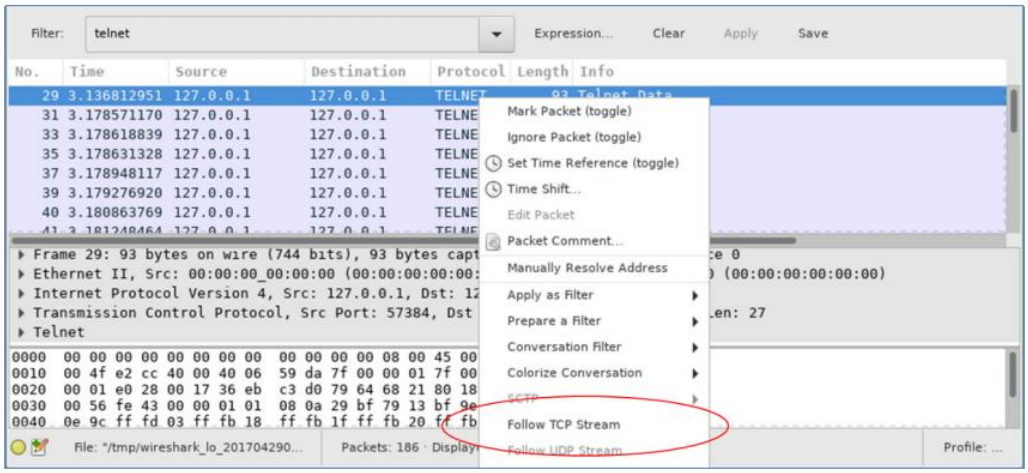


Рисунок 6.1 – Вибір Follow TCP Stream у програмі Wireshark

1.8. У вікні Follow TCP Stream (Стежити за TCP-потоком) відображаються дані про поточному сеансі підключення до віртуальної машині робочої станції CyberOps по протоколу Telnet. Весь сеанс зв'язку (в тому числі пароль) відображається у вигляді відкритого тексту . Зверніть увагу на те, що введене ім'я користувача відображається з повторюваними символами. Це пов'язано з налаштуванням відображення («відлуння») в Telnet, яка дозволяє виводити на екран символи, що набираються на клавіатурі (рис. 6.2).

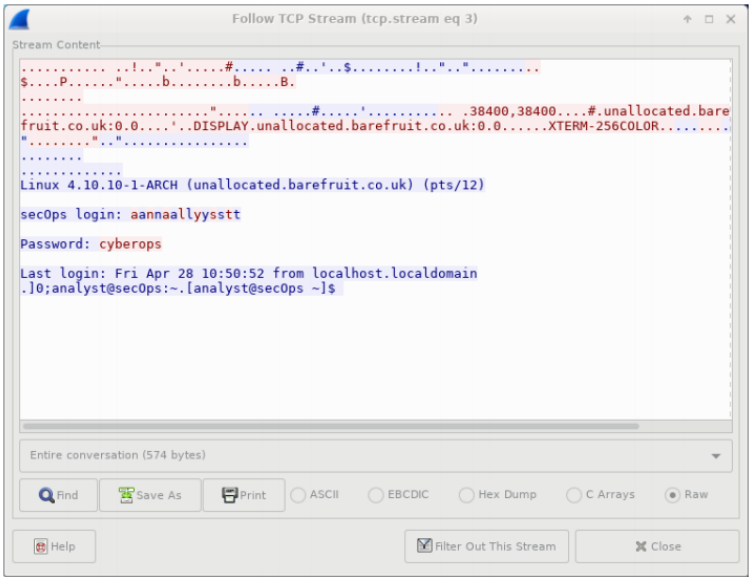


Рисунок 6.2 – Виведення символів на екран в Telnet

1.9. Закінчивши перегляд сеансу Telnet в вікні Follow TCP Stream, натисніть кнопку Close (Закрити).

1.10. Введіть exit в терміналі, щоб завершити сеанс Telnet. *[Analyst @ secOps ~] $* exit

2. Вивчення сеансу SSH з допомогою програми Wireshark

У частині 2 потрібно створити сеанс SSH з локальним хост-комп'ютером. Для перехоплення і перегляду даних цього сеансу SSH буде використовуватися програма Wireshark.

2.1 Почніть ще один перехоплення даних програмою Wireshark.

2.2 Потрібно створити сеанс SSH з локальним хост-комп'ютером. У командному рядку терміналу введіть ssh localhost. Для продовження введіть yes (та). Введіть cyberops при появі відповідного запиту.

[Analyst @ secOps ~] $ ssh localhost

Справжність хоста localhost (:: 1) Не може бути встановлена.

Відбиток ключа ECDSA - SHA256: uLDhKZflmvsR8Et8jer1NuD91cGDS1mUl / p7VI3u6kI.

Are you sure you want to continue connecting (yes / no)? (Ви дійсно хочете продовжити підключення (так / ні)?) Yes (та)

Попередження. У список відомих хостів постійно доданий localhost (ECDSA).

пароль analyst @ localhost :

Last login: Sat Apr 29 00:04:21 2026 from localhost.localdomain

2.3. Зупиніть перехоплення даних програмою Wireshark.

2.4. Застосуйте один з фільтрів SSH для даних, що збираються програмою Wireshark. Введіть ssh в поле фільтра і натисніть кнопку Apply (Застосувати).

2.5. Натисніть правою кнопкою миші на одну з рядків SSHv2 в розділі Packet list (Список пакетів) програми Wireshark і виберіть в спадному списку пункт Follow TCP Stream .

2.6. Вивчіть вікно Follow TCP Stream сеансу SSH. Дані зашифровані і недоступні для читання. Порівняйте дані сеансу SSH з даними сеансу Telnet (рис. 6.3).

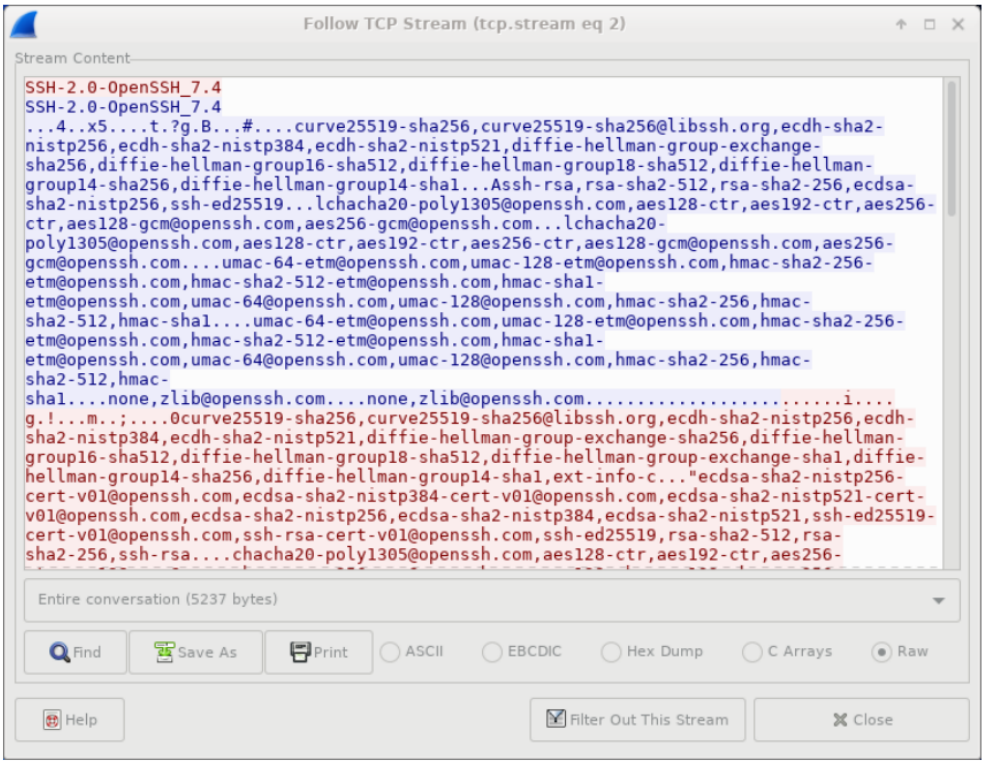


Рисунок 6.3 – Порівняння даних сеансу SSH з даними сеансу Telnet

2.7. Після вивчення сеансу SSH натисніть Close (Закрити).

2.8. Закрийте програму Wireshark.

Контрольні питання:

Чому для віддалених підключень протокол SSH є кращим, ніж протокол Telnet?

ЛАБОРАТОРНА РОБОТА №7
ХЕШУВАННЯ

МЕТА: створення хешів з допомогою OpenSSL; перевірка хешів

ЗАГАЛЬНІ ВІДОМОСТІ І СЦЕНАРІЙ

Хеш-функції представляють собою математичні алгоритми , призначені для прийому вхідних даних і формування унікальної рядка символів фіксованого розміру, званого хешем. Хеш-функції спроектовані з розрахунком на швидкодію, і для них дуже складно виконати зворотний розрахунок; дуже важко відновити дані, використані при створенні будь-якого конкретного хешу, маючи один лише хеш. Інша важлива властивість хеш функцій - це те, що навіть саме мале зміна вхідних даних призводить до отримання абсолютно іншого хешу [11].

OpenSSL можна використовувати для створення і порівняння хешів, але є і інші інструменти. Деякі з цих коштів також розглядаються в цій лабораторній роботі.

Необхідні ресурси:

- Віртуальна машина робочої станції CyberOps
- Доступ до Інтернету

1. Створення хешів з допомогою OpenSSL

OpenSSL можна використовувати в якості автономного засобу для хешування . Для того щоб створити хеш текстового файлу , виконайте такі дії.

Хешування текстового файлу:

1.1 У віртуальній машині робочої станції CyberOps відкрийте вікно терміналу.

1.2 Хеш текстового файлу знаходиться в каталозі / home / analyst / lab.support.files /, тому перейдіть в цей каталог. [Analyst @ secOps ~] $ cd / home / analyst / lab.support.files /

1.3 Введіть наведену нижче команду для відображення вмісту текстового файлу letter_to_grandma.txt на екрані:

[Analyst @ secOps lab.support.files] $ cat letter_to_grandma.txt

Hi Grandma, (Привіт , бабуся !) I am writing this letter to thank you for the chocolate chip cookies you sent me . (Я пишу це лист, щоб подякувати тебе за шоколадне печиво , яке ти мені прислала.) I got them this morning and I have already eaten half of the box ! (Я отримав їх вранці і вже з'їв половину коробки!) They are absolutely delicious ! (Вони дуже смачні !)

I wish you all the best . (Бажаю тобі всього найкращого.) Love , (С любов'ю ,) Your cookie-eater grandchild . (Твій онук, який дуже любить печиво).

1.4. По- , як і раніше з вікна терміналу введіть наведену нижче команду для хеширования текстового файлу . Команда буде використовувати MD5 в якості алгоритму хешування для створення хешу текстового файлу. Розрахований з допомогою OpenSSL хеш буде виведений на екран.

[Analyst @ secOps lab.support.files] $ openssl md5 letter_to_grandma.txt

MD5 (letter_to_grandma.txt) = 8a82289f681041f5e44fa8fbeeb3afb6

Зверніть увагу на формат вихідних даних . OpenSSL показує використаний алгоритм хешування MD5, а потім ім'я файлу , використовуваного в якості вхідних даних . Власне хеш MD5 відображається після знака одно (=).

1.5. Хеш функції корисні для перевірки цілісності даних , будь то зображення, пісня або просто текстовий файл. Саме мале зміна призводить до абсолютно іншому хешу. Хеш можна розрахувати до і після передачі , а потім порівняти. Якщо хеш-кодування НЕ збігаються, дані були змінені під час передачі. Давайте змінимо текстовий файл letter_to_grandma.txt і перерахуємо хеш MD5. Виконайте наведену нижче команду, щоб відкрити nano , текстовий редактор, що запускається з командного рядка.

[Analyst @ secOps lab.support.files] $ nano letter_to_grandma.txt

З допомогою nano змініть перше пропозицію з Hi Grandma на Hi Grandpa. Зверніть увагу , що змінюється тільки один символ: m на p . Після внесення зміни натисніть клавіші <CONTROL + X> , щоб зберегти змінений файл. Натисніть Y , щоб підтвердити ім'я і зберегти файл. Натисніть клавішу < Enter > , щоб вийти з nano і перейти до наступного кроку.

1.6 Тепер , коли файл змінений і збережений , запустіть ту ж команда повторно, щоб створити хеш MD5 файлу.

[Analyst @ secOps lab.support.files] $ openssl md5 letter_to_grandma.txt

MD5 (letter_to_grandma.txt) = dca1cf6470f0363afb7a65a4148fb442

Відрізняється чи новий хеш від розрахованого в пункті (1.6)? Наскільки відрізняється ?

1.7 Хеші MD5 вважаються слабкими і вразливими для атак. До більш надійним алгоритмам хешировання відносяться SHA-1 і SHA-2. Для створення хешу SHA-1 файлу letter_to_grandma.txt використовуйте наведену нижче команду.

[Analyst @ secOps lab.support.files] $ openssl sha1 letter_to_grandma.txt
SHA1 (letter_to_grandma.txt) = 08a835c7bcd21ff57d1236726510c79a0867e861
[analyst @ secOps lab.support . files] $

Примітка. Існують і інші засоби для створення хешів. Зокрема , md5sum, sha1sum і sha256sum можуть бути використані для отримання хешей MD5, SHA-1 і SHA-2-256 відповідно.

1.8. Використовуйте md5sum і sha1sum для створення хешу MD5 і SHA-1 файлу hash letter_to_grandma.txt.

[Analyst @ secOps lab.support.files] $ md5sum letter_to_grandma.txt

dca1cf6470f0363afb7a65a4148fb442 letter_to_grandma.txt

[analyst @ secOps lab.support.files] $ sha1sum letter_to_grandma.txt

08a835c7bcd21ff57d1236726510c79a0867e861 letter_to_grandma.txt

[analyst @ secOps lab.support.files] $

Збігаються чи хеші , створені з допомогою md5sum і sha1sum , з образами, створеними в пунктах (1.7) та (1.8) відповідно ? Дайте пояснення.

Примітка . SHA-1 поки практично невразливий , але комп’ютери стають все більш і більш потужними . Передбачається , що в результаті природного розвитку зловмисникам незабаром часу вдасться зламати SHA-1. В якості запобіжного заходи в даний час рекомендується по замовчуванням використовувати для хешування SHA-2. Слід також зазначити , що SHA-2 в

дійсності представляє собою сімейство алгоритмів хешування . Сімейство SHA-2 складається з шести хеш функцій , а саме : SHA-224, SHA-256, SHA-384, SHA-512, SHA- 512/224, SHA-512/256. Ці функції створюють хеш значення довжиною 224, 256, 384 і 512 біт відповідно .

Примітка. ВМ CyberOPS підтримує тільки SHA-2-224, SHA-2-256 і SHA-2-512 (sha224sum, sha256sum і sha512sum відповідно).

2. Перевірка хешів

Як було відзначено , зазвичай хеші використовуються для перевірки цілісності файлів . Виконайте наступні дії , використовуючи хеш-кодування SHA-2-256, щоб перевірити цілісність файлу sample.img , завантаженого з Інтернету [11].

2.1 Разом з sample.img також був завантажений файл sample.img_SHA256.sig. Файл sample.img_SHA256.sig містить SHA-2-256, обчислений на веб-сайті. По- перше , використовуйте команду cat для показу вмісту файлу sample.img_SHA256.sig.

[Analyst @ secOps lab.support.files] $ cat sample.img_SHA256.sig

c56c4724c26eb0157963c0d62b76422116be31804a39c82fd44ddf0ca5013e6a

2.2 Використовуйте SHA256sum для розрахунку хешу SHA-2-256 файлу sample.img.

[Analyst @ secOps lab.support.files] $ sha256sum sample.img

c56c4724c26eb0157963c0d62b76422116be31804a39c82fd44ddf0ca5013e6a sample.img

Файл *sample.img* завантажений правильно? Дайте пояснення.

Примітка. Порівняння хеш-сум – це відносно надійний метод виявлення помилок при передачі даних, але існують кращі способи переконатися, що вміст файлу не спотворено. Такі засоби, як gpg, пропонують набагато більш ефективний метод перевірки того, що завантажений файл не був змінений третіми сторонами і що він дійсно є файлом, призначеним видавцем для публікації.

ЛАБОРАТОРНА РОБОТА №8
СХОВИЩА ЦЕНТРІВ СЕРТИФІКАЦІЇ

Частина 1. Сертифікати, яким довіряє ваш браузер

Частина 2. Виявлення атаки через посередника

ЗАГАЛЬНІ ВІДОМОСТІ І СЦЕНАРІЙ

У міру розвитку Інтернету зростає необхідність забезпечення безпеки. Протокол HTTPS (де S означає security, тобто «безпека») поряд з концепцією центрів сертифікації був запропонований компанією Netscape в 1994 році і раніше використовується в даний час.

У цій лабораторній роботі потрібно:

– перерахувати всі сертифікати, яким довіряє ваш браузер (виконується на вашому комп'ютері);

– використовувати хеш-кодування, щоб з'ясувати, перехоплюється чи ваше інтернет-підключення (виконується на віртуальній машині CyberOps).

Необхідні ресурси:

– віртуальна машина робочої станції CyberOps;

– доступ до Інтернету

Частина 1: Сертифікати, яким довіряє ваш браузер

Для перевірки HTTPS використовують сторонні кошти. Стороннє засіб, відоме як центр сертифікації (ЦС), перевіряє, що доменне ім'я дійсно належить організації, яка стверджує, що є його власником. Якщо перевірка завершується успішно, ЦС створює підписаний сертифікат, що містить відомості про організацію, в тому числі його загальному ключі.

Вся система ґрунтується на тому, що веб-оглядачі та операційні системи поставляються з списком довірених ЦС. Будь-сертифікат, підписаний якимось ЦС в списку, буде розглядатися веб-браузером як законний і автоматично заслуговує на довіру. Для того щоб зробити систему більш безпечною і масштабується, ЦС часто розподіляють завдання створення і підписування сертифікатів між багатьма дочірніми ЦС. Батьківський ЦС називається

кореневим центром сертифікації. Якщо веб-браузер довіряє кореневого центру сертифікації, він також довіряє всім його дочірнім ЦС.

Примітка. Хоча сховища сертифікатів веб-браузерів аналогічні, в даній лабораторній роботі розглядаються Chrome 56 і Firefox 59. Меню і графіки можуть відрізнятися в залежності від версії веб-браузера. Виконайте дії, щоб відкрити сховище ЦС в браузері.

Крок 1: Відкрийте кореневі сертифікати в Chrome.

Цей крок можна виконати на локальному комп'ютері або використовувати FireFox на віртуальній машині робочої станції CyberOps. Якщо ви використовуєте Firefox, перейдіть до кроку 2. Якщо ви використовуєте веб-браузер, відмінний від Chrome або Firefox, виконайте пошук в Інтернеті, щоб знайти опис дій, необхідних для відображення своїх кореневих сертифікатів.

Примітка. Меню і графіки можуть відрізнятися в залежності від версії браузера Chrome:

1. Відкрийте веб-оглядач Chrome на своєму ПК.

2. Натисніть піктограму з трьома крапками в дальньому правому кінці адресного рядка, щоб відобразити параметри Chrome (рис. 8.1).

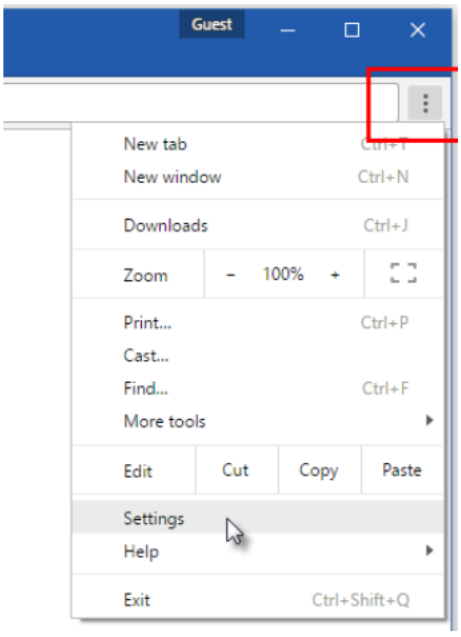


Рисунок 8.1 – Відображення параметри Chrome

3. Натисніть Settings (Настройки), а потім натисніть Show Advanced Settings (Показати додаткові параметри).

4. Прокрутіть сторінку вниз і натисніть кнопку Manage certificates ... (Налаштувати сертифікати ...) в розділі HTTPS / SSL.

5. У вікні Certificates (Сертифікати) виберіть вкладку Trusted Root Certification\Authorities (Довірені кореневі центри сертифікації). Відкриється вікно, в якому показані всі сертифікати і ЦС, яким довіряє Chrome (рис. 8.2).

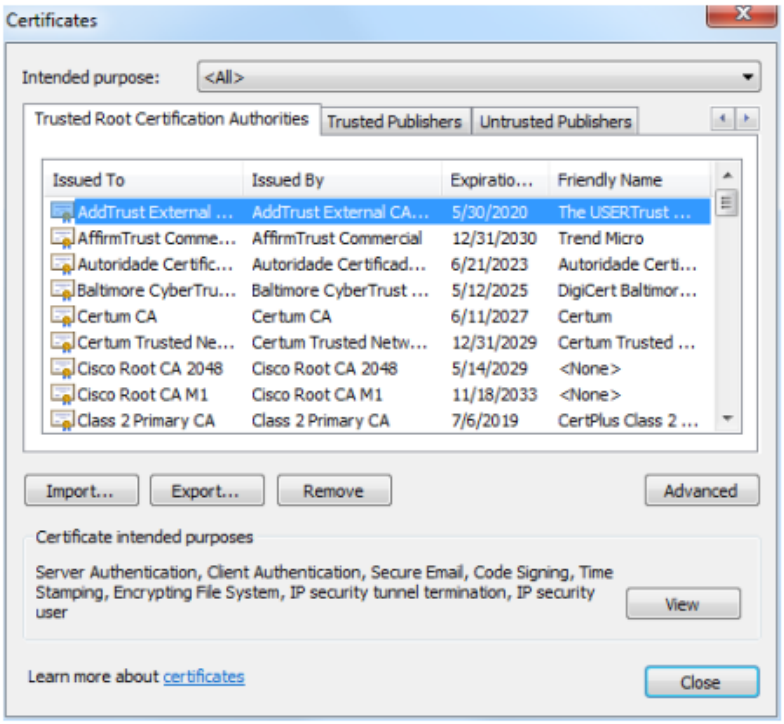


Рисунок 8.2 – Сертифікати і ЦС, яким довіряє Chrome

Крок 2: Відображення сертифікатів в сховище ЦС в Firefox.

Примітка. Меню і графіки можуть відрізнятися в залежності від версії браузера Firefox і операційної системи. Firefox 59 на віртуальній машині робочої станції CyberOps показаний на цьому етапі.

1. Відкрийте Firefox і клацніть значок меню. Значок Menu (Меню) знаходиться в крайній правій частині вікна Firefox поруч з адресним рядком (рис. 8.3).

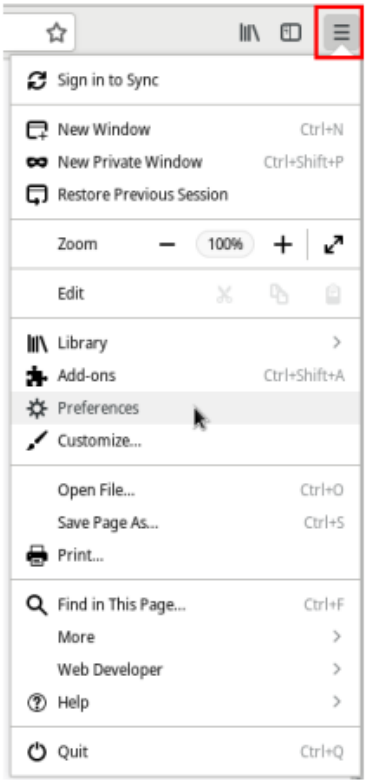


Рисунок 8.3 – Меню браузера Firefox

2. Натисніть Preferences> Privacy & Security (Параметри> Безпека і конфіденційність).

3. Прокрутіть вниз до розділу безпеки і натисніть View Certificates (Переглянути сертифікати).

4. Відкриється вікно, в якому показані сертифікати і ЦС, яким довіряє Firefox (рис. 8.4).

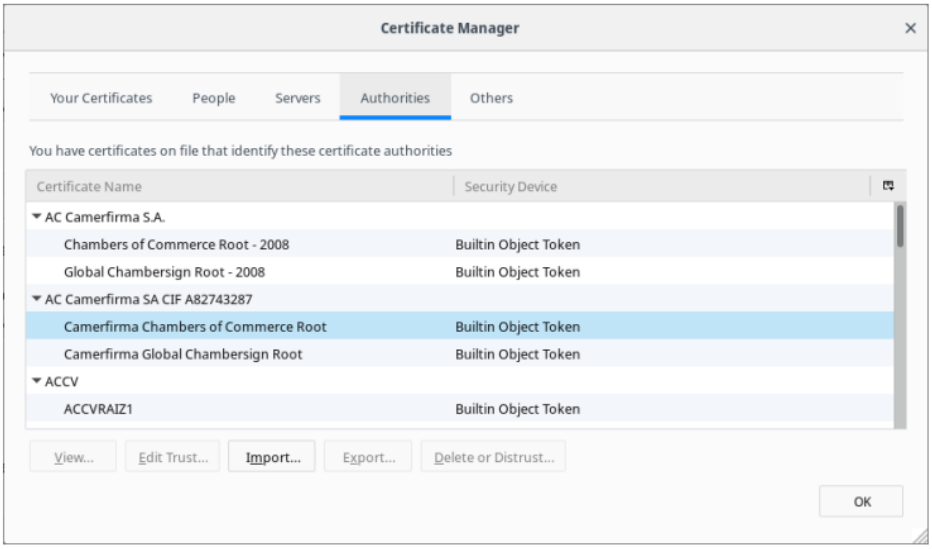


Рисунок 8.4 – Сертифікати і ЦС, яким довіряє Firefox

Частина 2: Виявлення атаки через посередника

Ця частина виконується з використанням робочої станції віртуальної машини CyberOps.

Хеші широко використовуються для перевірки цілісності даних, але вони також можуть використовуватися для виявлення атак через посередника HTTPS.

Для захисту даних користувача все більше веб-сайтів переключаються на зашифрований трафік. Відповідно до підходом, відомим як HTTPS, сайти використовують такі протоколи, як TLS / SSL, для шифрування призначеного для користувача трафіку від початку до кінця. Після того як трафік надійно зашифрований, дуже складно кому б то не було, крім користувача і даного сайту, переглянути вміст зашифрованого повідомлення. Це корисно для користувачів, але створює проблеми організаціям, які хочуть заглянути в трафік. Підприємства та молодіжні організації часто прагнуть переглядати трафік своїх співробітників з метою контролю. Їм необхідно мати можливість переглядати TLS / SSL-зашифрований трафік. Це робиться за допомогою проксі-сервера HTTPS.

Веб-оглядачі довіряють посвідченню відвідуваного веб-сайту, якщо сертифікат, представлений цим веб-сайтом, підписаний одним з ЦС, встановлених в сховище сертифікатів браузера. Для того щоб мати можливість переглядати TLS / SSL-зашифрований трафік користувачів, компанія або організація просто додає ще один ЦС в список встановлених ЦС браузера користувача.

Розглянемо наступний сценарій: компанія X наймає нового співробітника і надає йому ноутбук. Перед передачею ноутбука IT-відділ компанії встановлює все необхідне для роботи програмне забезпечення. Поряд з встановленими програмним забезпеченням і процесами IT-відділ включає в список довірених ЦС один додатковий ЦС. Цей додатковий ЦС вказує на керований компанією комп’ютер, відомий як проксі-сервер HTTPS. Оскільки компанія контролює трафік, проксі-сервер HTTPS можна помістити в середину будь-якого підключення.

Принцип дії наступний:

1. Користувач намагається встановити захищене з'єднання з використовують протокол HTTPS веб-сайтом H, розміщеним в Інтернеті. H може бути будь-який HTTPS-сайт: банк, інтернет-магазин, сервер електронної пошти і т.д.

2. Оскільки компанія управляє трафіком, вона робить це таким чином, щоб весь призначений для користувача трафік проходив через проксі-сервер HTTPS. Потім проксі-сервер HTTPS *уособлює* веб-сайт H і являє самоподпісанний сертифікат, щоб довести, що це H. По суті, проксі-сервер HTTPS говорить: «Привіт, я є HTTPS-сайт H. Ось мій сертифікат. Він підписаний ... мною ».

3. Оскільки представлений сертифікат підписаний одним з ЦС, включених в сховище ЦС ноутбука (слід пам'ятати, що він був доданий IT-відділом), веб-оглядач помилково вважає, що дійсно обмінюється даними з H. Зверніть увагу, що, якщо б додатковий ЦС не додано до сховище ЦС, ноутбук не довіряв би сертифікату і негайно розпізнав, що хтось інший намагається *уособлювати* H.

4. Ноутбук довіряє підключенню і встановлює безпечний канал зв'язку з проксі-сервером HTTPS, помилково вважаючи, що безпечно обмінюється данними з H.

5. Тепер проксі-сервер HTTPS встановлює другий безпечне підключення з H, веб-сайтом, до якого користувач намагався отримати доступ з самого початку.

6. Проксі-сервер HTTPS тепер є кінцевою точкою двох окремих безпечних підключень: одного, встановленого з користувачем, і іншого, встановленого з H. Оскільки HTTPS - кінцева точка обох підключень, можна розшифрувати трафік обох підключень.

7. Тепер проксі-сервер HTTPS може отримати TLS / SSL-зашифрований трафік користувача, призначений для H, розшифрувати його, проінспектувати, повторно зашифрувати за допомогою TLS / SSL і відправити H. Коли H відповідає, проксі-сервер HTTPS виконує зворотний процес перед пересиланням трафіку користувачеві.

Зверніть увагу, що процес в основному прозорий для користувача, який вважає протоколу TLS / SSL-зашифрованими (зелені значки в браузері).

З’єднання безпечне (TLS / SSL-зашифроване), але встановлено через помилковий веб-сайт.

Хоча його присутність зазвичай є прозорим для користувача, TLS-проксі можна легко виявити за допомогою хешів. У наведеному вище прикладі, оскільки проксі-сервер HTTPS НЕ має доступу до приватних ключами сайту H, сертифікат, представлений користувачеві, відрізняється від сертифіката, поданого сайтом H. У кожен сертифікат входить значення, відоме як *відбиток.* По суті, хеш розраховується і підписується видавцем сертифіката, а відбиток виступає в якості унікальної зведення всього вмісту сертифіката. Якщо хоча б одну буква сертифіката змінена, обчислений відбиток матиме зовсім інше значення. Через це властивості відбитки використовуються для швидкого порівняння сертифікатів. Повертаючись до наведеного вище прикладу, користувач може запросити сертифікат H і порівняти включений в нього відбиток з відбитком, наданим, коли було встановлено підключення до веб-сайту H. Якщо відбитки збігаються, з’єднання дійсно встановлено з H. Якщо відбитки не збігаються, значить, підключення встановлено з якоїсь іншої кінцевою точкою. Виконайте наступні дії, щоб з’ясувати, чи є проксі-сервер HTTPS в вашому з’єднанні.

Крок 1: Збір правильних і незмінених відбитків сертифіката.

a. Перший крок – зібрати відбитки кількох сайтів. Це важливо, оскільки вони будуть використовуватися для порівняння пізніше. У наведеній нижче таблиці міститься кілька відбитків сертифікатів популярних сайтів.

Примітка. Відбитки SHA-1, показані в таблиці 8.1, можливо, більше не дійсні, так як організації регулярно оновлюють свої сертифікати. Відбиток (fingerprint) також називається thumbprint в комп’ютерах Windows.

Таблиця 1 – Пошукові системи та їх відбитки сертифіката SHA-1

сайт	Домени, на які поширюється сертифікат	Відбиток сертифіката SHA-1 (станом на квітень 2018 г.)
www.cisco.com	www.cisco.com	64: 19: CA: 40: E2: 1B: 3F: 92: 29: 21: A9: CE: 60: 7D: C9: 0C: 39: B5: 71: 3E
www.facebook.com	* .facebook.com	BD: 25: 8C: 1F: 62: A4: A6: D9: CF: 7D: 98: 12: D2: 2E: 2F: F5: 7E: 84: FB: 36
www.wikipedia.org	* .wikipedia.org	4B: 3E: D6: B6: A2: C7: 55: E8: 56: 84: BE: B1: 42: 6B: B0: 34: A6: FB: AC: 24
twitter.com	twitter.com	26: 5C: 85: F6: 5B: 04: 4D: C8: 30: 64: 5C: 6F: B9: CF: A7: D2: 8F: 28: BC: 1B
www.linkedin.com	www.linkedin.com	3A: 60: 39: E8: CE: E4: FB: 58: 87: B8: 53: 97: 89: 8F: 04: 98: 20: BF: E3: 91

Що таке відбитки? Чому вони важливі?_____

Хто обчислює відбитки? Як знайти їх?_____

Крок 2: Отримайте відбиток сертифіката, який використовується ВМ робочої станції CyberOps.

Тепер, коли у нас є справжні відбитки, настав час отримати відбитки з локального хоста і порівняти значення. Якщо відбитки не збігаються, використовуваний сертифікат НЕ належить до перевіряється сайту HTTPS. Це означає, що є проксі-сервер HTTPS між хост-комп'ютером і перевіряється HTTPS-сайтом. Збіг відбитків означає, що проксісервер HTTPS відсутня.

1) Використовуйте три наведені нижче конвеєрні команди, щоб отримати відбиток для Cisco.com. У показаної нижче рядку OpenSSL використовується для підключення до cisco.com через порт 443 (HTTPS), запиту сертифіката і його збереження в текстовому файлі з ім'ям cisco.pem. Для контексту також показаний результат.

[analyst @ secOps ~] $ echo -n | openssl s_client -connect cisco.com:443 | sed -ne '/ -BEGIN CERTIFICATE - /, / - END CERTIFICATE- / p'> ./cisco.pem

depth = 2 C = BM, O = QuoVadis Limited, CN = QuoVadis Root CA 2

verify return: 1

depth = 1 C = US, O = HydrantID (Avalanche Cloud Corporation), CN = HydrantID SSL ICA G2

verify return: 1

depth = 0 C = US, ST = CA, L = San Jose, O = "Cisco Systems, Inc.", CN = www.cisco.com

verify return: 1

ГОТОВИЙ

2) При необхідності використовуйте команду cat для відображення вмісту сертифіката, отриманого і збереженого в текстовому файлі cisco.pem .

[analyst @ secOps ~] $ cat cisco.pem

--- ПОЧАТОК СЕРТИФІКАТУ ---

MIIG1zCCBL + gAwIBAgIUKBO9xTQoMemc9zFHNkdMW + SgFO4wDQYJKoZIhvcNAQEL
BQAwXjELMAkGA1UEBhMCVVMxMDAuBgNVBAoTJ0h5ZHJhbnRJRCAoQXZhbGFuY2hl
IENsb3VkIENvcnBvcmF0aW9uKTEdMBsGA1UEAxMUSHlkcmFudElEIFNTTCBJQ0Eg
RzIwHhcNMTcxMjA3MjIxODU1WhcNMTkxMjA3MjIyODAwWjBjMQswCQYDVQQGEwJV
UzELMAkGA1UECAwCQ0ExETAPBgNVBAcMCFNhbiBKb3NlMRwwGgYDVQQKDBNDaXNj
byBTeXN0ZW1zLCBJbmMuMRYwFAYDVQQDDA13d3cuY2lzY28uY29tMIIBIjANBgkq
yvo6dWpJdSircYy8HG0nz4 + 936 + 2waIVf1BBQXZUjNVuws74Z / eLIpl2c6tANmE0
q1i7fiWgItjDQ8rfjeX0oto6rvp8AXPjPY6X7PT1ulfhkLYnxqXHPETRwr8l5COO
MDEh95cRxATXNAlWAwLcBT7lDmrGron6rW6hDtuUPPG / rjZeZbNww5p / nT3EXX2L
Rh + m0R4j / tuvy / 77YRWyp / VZhmSLrvZEYiVjM2MgCXBvqR + aQ9zWJkw + CAm5Z414
Eiv5RLctegYuBUMGTH1al9r5cuzfwEg2mNkxl4I / mtDro2kDAv7bcTm8T1LsZAO /
1bWvudsrTA8jksw + 1WGAEd9bHi3ZpJPYedlL

--- КІНЕЦЬ СЕРТИФІКАТУ ---

[analyst @ secOps ~] $

3) Тепер, коли сертифікат збережений у текстовому файлі cisco.pem , використовуйте наведену нижче команду для його вилучення і відображення його відбитків.

[analyst @ secOps ~] $ openssl x509 -noout -in cisco.pem -fingerprint -sha1

SHA1 Fingerprint = 64: 19: CA: 40: E2: 1B: 3F: 92: 29: 21: A9: CE: 60: 7D : C9: 0C: 39: B5: 71: 3E

[analyst @ secOps ~] $

Примітка. Існує дві причини, за якими відбитки можуть мати різні значення. По-перше, ваша операційна система може відрізнятися від ОС на ВМ робочої станції CyberOps.

По-друге, сертифікати регулярно оновлюються, а значення відбитків змінюються.

Який алгоритм хешування використовувався OpenSSL для розрахунку відбитка?_____

Чому було обрано саме цей алгоритм? Це важливо?_____

Крок 3: Порівняйте відбитки

Використовуйте таблицю 8.1 для порівняння відбитка сертифіката, отриманого на HTTPS-сайті Cisco, з відбитком, отриманим всередині своєї мережі. Нагадуємо, що відбитки можуть змінюватися з часом.

Відбитки збігаються?_____

Що це означає?_____

Чи забезпечує цей метод повну надійність?_____

Частина 3: Завдання

a. Перевірте відбитки для сайтів, показаних в таблиці 1, але скористайтеся графічним інтерфейсом свого веб-оглядача.

Поради. найдіть спосіб показати відбиток в графічному інтерфейсі браузера.

Нагадування.

У цьому завданні корисно використовувати Google, а в Windows відбиток (fingerprint) часто іменується thumbprint.

b. Перевірте всі відбитки, перераховані в таблиці 1, за допомогою OpenSSL (частина 2, кроки 1-3).

Питання для повторення

Які умови необхідні для роботи проксі-сервера HTTPS?_____

СПИСОК ЛІТЕРАТУРИ

1. Білан, С. В., Федоренко, І. О. Сучасні методи шифрування інформації та їх програмна реалізація. Львів : ЛНУ ім. І. Франка, 2022. 184 с.

2. Мельник, А. П., Шевчук, В. П. Основи стеганографії та цифрового водяного маркування : навч. посіб. Харків : ХНУРЕ, 2021. 168 с.

3. Гончар, О. В., Кузьменко, С. І. Прикладна криптологія : теорія та практика. Київ : КПІ ім. І. Сікорського, 2023. 240 с.

4. Литвин В., Костючко С., Поліщук М., Гринюк С. Криптографічна обробка облікових даних користувача в мережевому обладнанні. Міжнародна науково-практична конференція «Сучасні інформаційні технології: від теорії до практики (СІнТех-2026)», м. Луцьк, 22-23 травня 2026 р. Луцьк : ЛНТУ. С. 671-673.

5. Kostiuchko, S., Polishchuk, M., Zabolotnyi, O., Tkachuk, A., Twarog, B. The Auxiliary Parametric Sensitivity Method as a Means of Improving Project Management Analysis and Synthesis of Executive Elements. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, 2021. P. 174–184.

6. Поліщук, М., Семенюк, О., Поліщук, Л., Ломакін, М. Можливості авторизації та захисту даних користувача під час розробки хмарних веб-додатків для IoT. Комп’ютерно-інтегровані технології: освіта, наука, виробництво. Луцьк, 2023. Вип.52. С. 94-103.

7. Костючко С., Поліщук М., Конкевич Л., Абрамович В. Аналітика поведінки користувачів і об'єктів. Технічні вісті. Львів, 2022. №1(55), 2(56). С. 34-35.

8. Пархоменко, О. М. Сучасні методи криптоаналізу симетричних шифрів. Дніпро : ДНУ, 2022. 153 с.

9. Шевченко, О. Г. Алгоритми шифрування та захисту даних у комп’ютерних мережах: монографія. Київ : НАУ, 2023. 230 с.

10. Cisco. Networking Academy. URL: https://www.netacad.com (дата звернення: 23.10.2026)

У П-75 Прикладна криптологія та стеганографія: методичні вказівки до лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань F Інформаційні технології спеціальності F5 Кібербезпека та захист інформації денної та заочної форм навчання / уклад. М. М. Поліщук: ЛНТУ, 2026. 44 с.

Комп’ютерний набір: М. М. Поліщук

Редактор: М. М. Поліщук

Підп. до друку «___» _____ 2026р.
Формат 60х84/16. Папір офс. Гарнітура Таймс.
Ум. друк. арк. ____. Тираж 10 прим. Зам. ____

Відділ іміджу та промоції
Луцького національного технічного університету
43018, м. Луцьк, вул. Львівська, 75
Друк – ВІП ЛНТУ