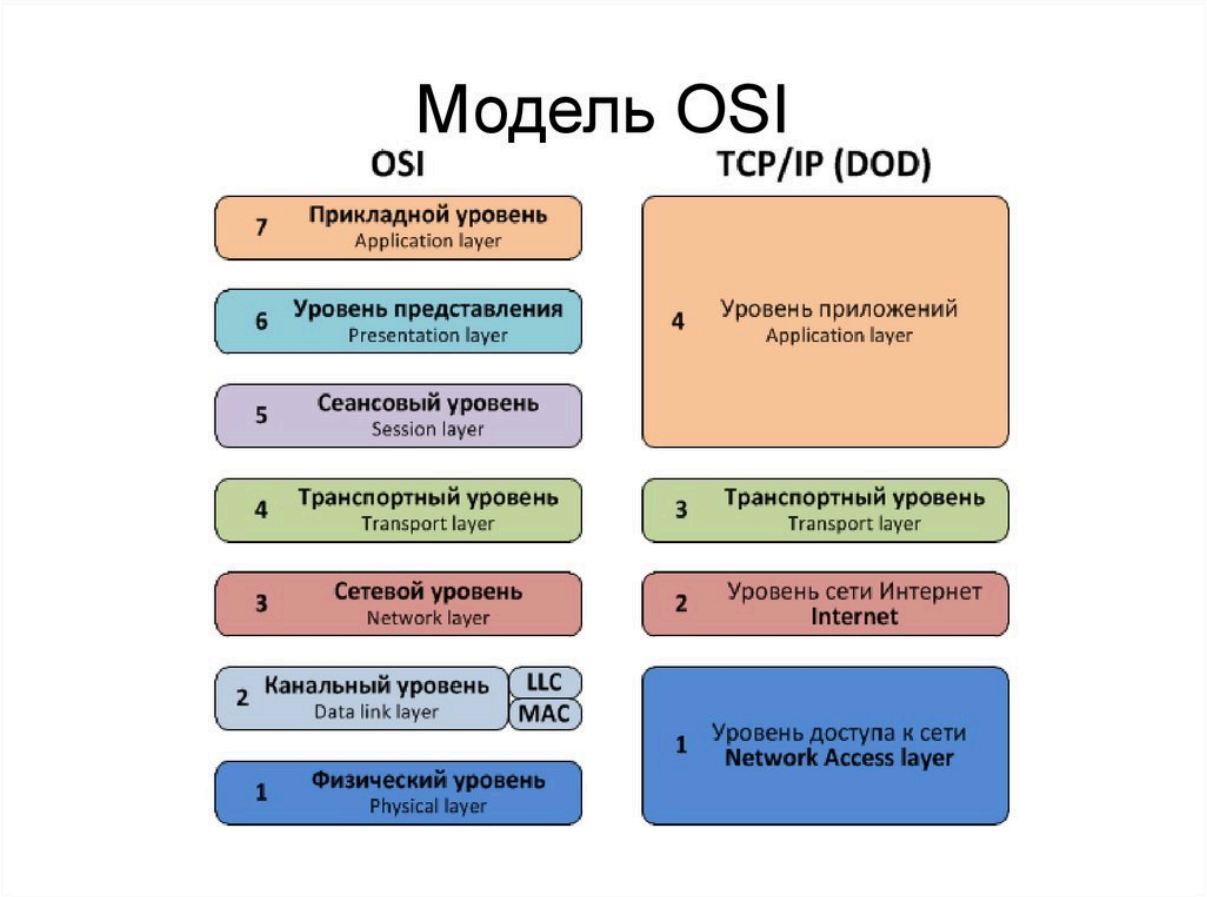
Лабораторна робота № 1
Виконали студенти групи ФІ-32мн: **Ємець Єлизавета**
Карловський Володимир
Коваленко Дар'я

1. На якому рівні проходить ініціалізація та функціонування протоколів SSL та TLS згідно зі стеком протоколів TCP/IP та мережевої моделі OSI?

Ініціалізація та функціонування протоколів SSL (Secure Sockets Layer) та TLS (Transport Layer Security) проходить на рівні транспортного шару у мережевій моделі OSI та рівні прикладних протоколів у стеку TCP/IP.

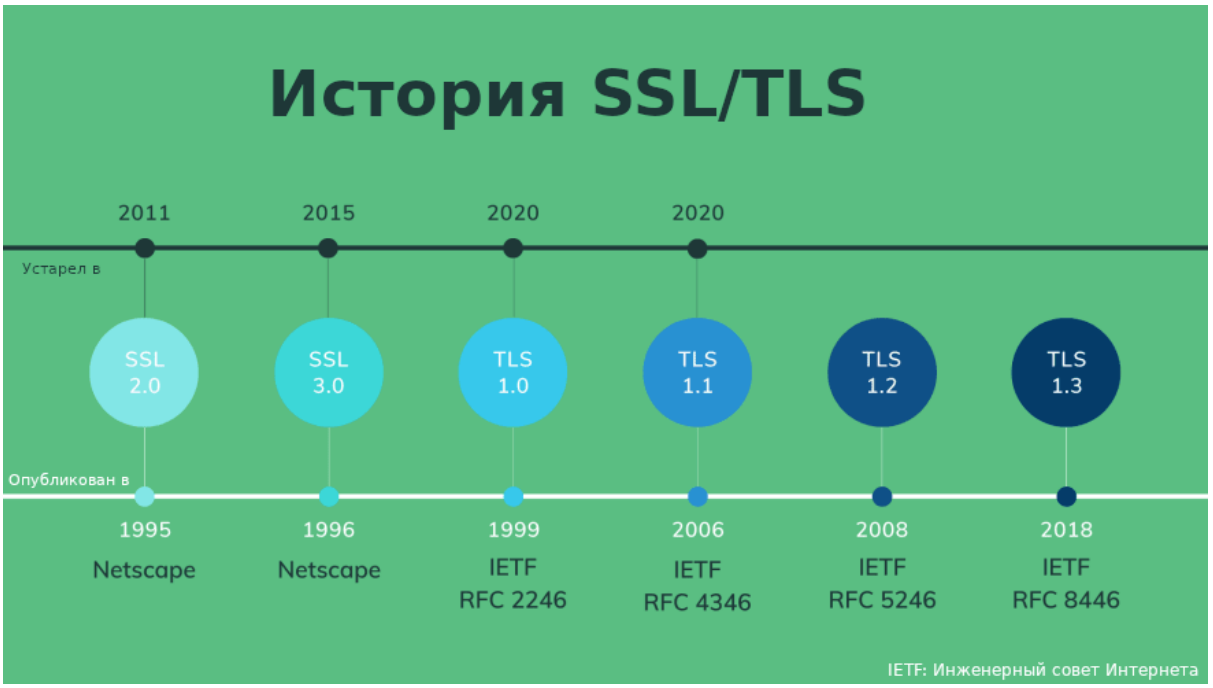


1. Модель OSI:
SSL/TLS функціонує на транспортному рівні.
Транспортний рівень - 4-й рівень моделі мережі OSI, розроблений для доставки даних. У той же час, не має значення, які дані передаються, де і де, тобто, він забезпечує сам механізм передачі. Він ділить блоки даних на фрагменти, розміри яких залежать від протоколу: короткі об'єднують в

одне і проривається довгим. Протоколи цього рівня призначені для взаємодії типу точки.
Цей рівень забезпечує безпечну передачу даних між додатками, використовуючи шифрування та аутентифікацію для захисту переданої інформації.
Сертифікати SSL/TLS діють як цифрові сертифікати ідентифікації для захисту мережевих з'єднань та встановлення справжності веб -сайтів в Інтернеті, а також ресурсів у приватних мережах

2. Модель TCP/IP:
У стеку TCP/IP SSL/TLS реалізується на рівні прикладних протоколів. SSL/TLS захищає передачу даних для прикладних протоколів, таких як HTTPS, SMTPS тощо, забезпечуючи безпеку з’єднання для протоколів, що передають конфіденційні дані через Інтернет.
Тому можна сказати, що SSL/TLS інтегрується з протоколами вищих рівнів і працює для забезпечення захищеного транспорту в межах транспортного рівня (OSI) або на рівні прикладних протоколів (TCP/IP).

2. Назвіть основні відмінності між версіями SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2. та TLS 1.3.



Версія	Рік випуску	Основні	Основні

		проблеми	**поліпшення**
SSL 1.0	1993	Не випущений через серйозні проблеми	-
SSL 2.0	1995	Низька безпека, відсутність перевірки цілісності	Початок SSL, але багато вразливостей
SSL 3.0	1996	Вразливість POODLE	Перевірка цілісності, кращі алгоритми
TLS 1.0	1999	Уразливий до BEAST	Нові шифри, захист від man-in-the-middle
TLS 1.1	2006	Застарілі методи шифрування	Захист від BEAST, покращена цілісність
TLS 1.2	2008	Складність handshake	Нові алгоритми AES, SHA-256, AEAD
TLS 1.3	2018	-	Швидкий handshake, сучасні шифри, краща безпека

3. Який стандарт цифрових сертифікатів використовується протоколами SSL та TLS? Назвіть основні елементи структури сертифікатів.

Протоколи SSL та TLS використовують цифрові сертифікати за стандартом X.509. Цей стандарт є основним для забезпечення автентифікації в мережах, де використовується криптографія з відкритим ключем.

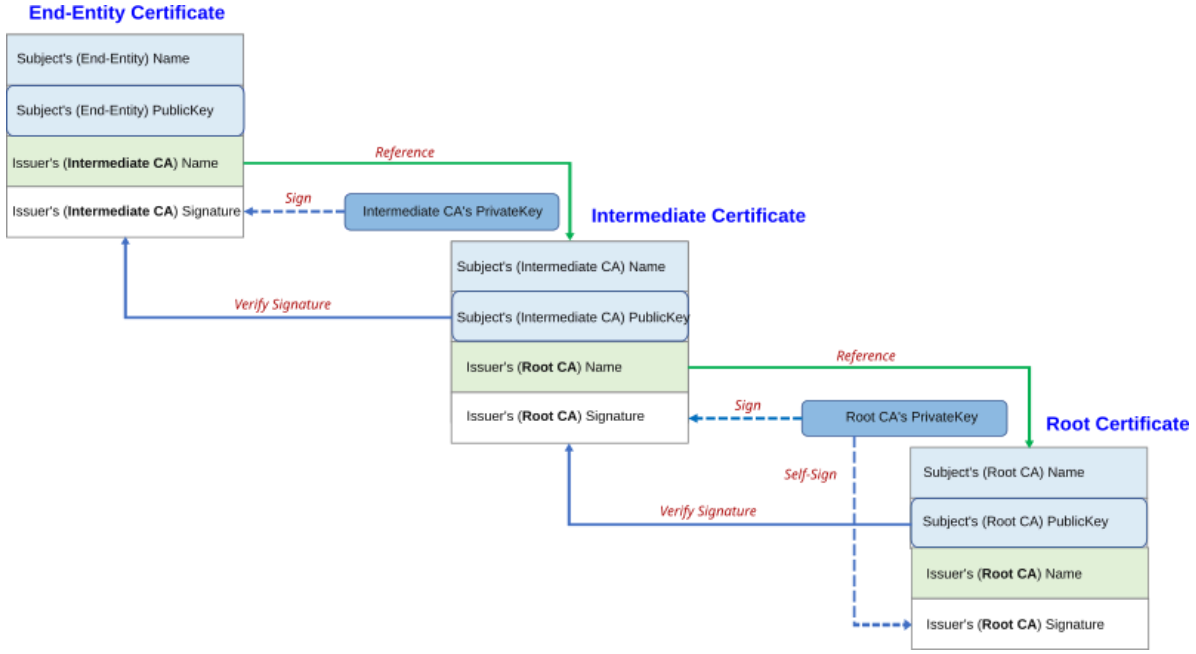
```
Certificate ::= SEQ {(кодується як послідовність, яка складається з полів)
                     tbsCertificate TBSCertificate(що треба підписувати)
                     signAlg AlgorithmIdent(алгоритм підпису)
                     Sign BIT STRING(підпис)
                     }

TBS Certificate ::= SEQ {
       version [0]INTEGER - v3
       serialNumber INTEGER(SIZE(0…20)) - {0, … 2^160 -1} - унікальний в межах ЦСК
       signAlg AlgId - унікальний ідентифікатор алгоритму підпису
       issuer Name - унікальний ідентифікатор довільного сертифікату
       validity SEQ{
              notBefore Time - початок терміну дії
              notAfter Time  - кінець терміну дії
              }
       subject Name - кому належить цей ключ
       subjectPublicKey SEQ{(який ключ тут лежить)
              keyAlg AlgId - алгоритм для якого цей ключ призначено
              Key BIT STRING
              }
       - - - - - - - - - - - - - - - v1
       issuerUniqueId [1]IMPL.OCTET STRING OPT - унікальний ідентифікатор видавця
       subjectUniqueId [2]IMPL.OCTET STRING OPT - унікальний ідентифікатор користувача
        - - -  - - - - - - - - - - - - - v2
       extensions [3]Extensions(дод інформація для довільних задач)
       }
```

Структура сертифіката X.509 забезпечує ідентифікацію, автентифікацію та можливість шифрування, що є ключовим для безпечної роботи протоколів SSL/TLS.

4. Що таке кореневі сертифікати, акредитовані центри сертифікації ключів (CA), сертифікат VeriSign та їх роль?
Кореневий сертифікат - це цифровий сертифікат найвищого рівня в ієрархії сертифікатів відкритого ключа (PKI).

Він є самопідписаним і ідентифікує кореневий центр сертифікації **(CA)**. За допомогою його сертифікуються сертифіката проміжного рівня. З усієї ієрархії має найдовший термін життя. (10-20р)



Центр сертифікації (CA) - це організація, яка зберігає, підписує та видає цифрові сертифікати. Цифровий сертифікат засвідчує право власності на відкритий ключ названого суб’єкта сертифіката. Це дозволяє іншим (довіряючим сторонам) покладатися на підписи або на твердження щодо закритого ключа, який відповідає сертифікованому відкритому ключу. Центр сертифікації діє як довірена третя сторона, якій довіряє як суб’єкт (власник) сертифіката, так і сторона, яка покладається на сертифікат.

Issuer	Market Share
Let's Encrypt	56.3%
GlobalSign	14.0%

IdenTrust	12.4%
Comodo Cybersecurity	7.3%
DigiCert Group	5.3%
GoDaddy Group	4.4%

Verisign розробила протокол SSL далі, який став TLS. Цей протокол важливий тим, Що завдяки ньому ми отримали більш безпечний інтернет

5. Де і як зберігаються відкриті ключі браузерів (Chrome, IE, Opera, Firefox) та як забезпечується їх цілісність?

Chrome/Opera (Opera is chromium based)

Windows: Використовує системне сховище Windows

Linux: ~/.pki/nssdb

macOS: Системний Keychain

Firefox

Власне сховище

IE

Використовує системне сховище Windows

Safari

Системний Keychain

Цілісність забезпечується за допомогою:

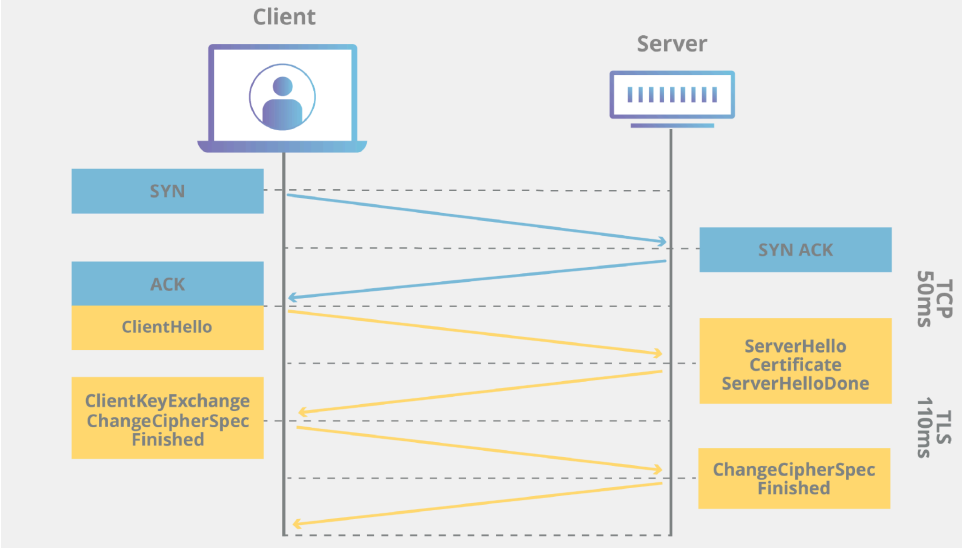
1. Хешування
2. Регулярна перевірка сертифікату через взаємодію з сервером
3. Взагалі є дуже багато механізмів оновлення і підтримки сертифікатів, але вони не є релевантними до цілісності

6. Який використовується метод формування спільного особистого майстер-ключа?

TLS-рукостискання відбувається щоразу, коли користувач переходить на веб-сайт за протоколом HTTPS, і браузер спочатку починає запитувати сервер-джерело сайту.

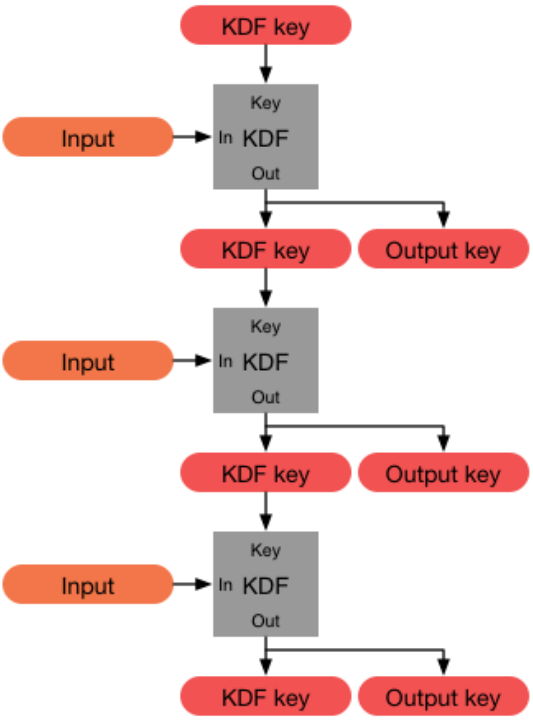
Під час TLS-рукостискання клієнт та сервер разом виконують такі дії:

1. Вказують, яку версію TLS (TLS 1.0, 1.2, 1.3 тощо) вони використовуватимуть
2. Вирішують, які набори шифрів (див. нижче) використовуватимуть
3. Перевіряють справжність сервера за допомогою відкритого ключа сервера та цифрового підпису центру сертифікації SSL.
4. Генерують сеансові ключі, щоб використовувати симетричне шифрування після завершення рукостискання



Генерація відбувається за допомогою DH або ElipticDH

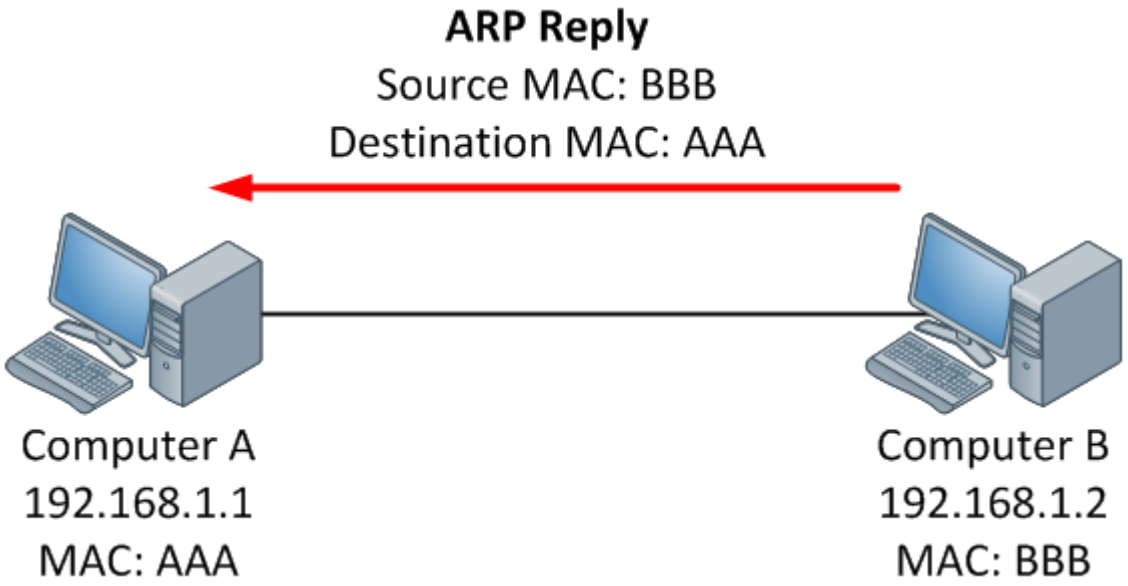
KDF - Key derivation function - це криптографічний алгоритм, який отримує один або кілька секретних ключів із секретного значення, такого як master-ключ.


KDF key
Key
Input
In KDF
Out
KDF key
Output key
Key
Input
In KDF
Out
KDF key
Output key
Key
Input
In KDF
Out
KDF key
Output key

7. Опишіть метод ARP спуфінгу та його застосування для атаки sslstrip.

Address Resolution Protocol (ARP) – це протокол мережевого рівня, який зіставляє IP-адреси з MAC-адресами в локальній мережі. При нормальній роботі:

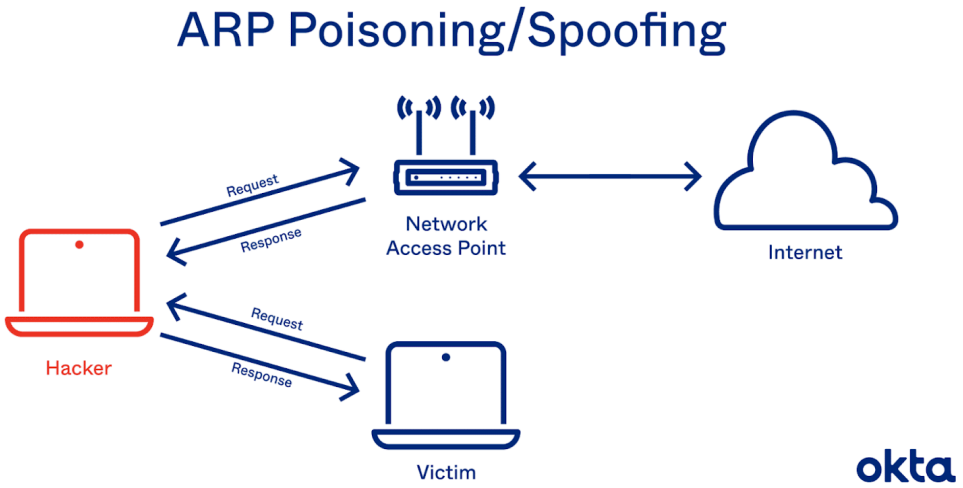
- Пристрій A запитує MAC-адресу пристрою B через broadcast ARP-запит
- Пристрій B відповідає своєю MAC-адресою
- Обидва пристрої зберігають цю інформацію в ARP-кеші



Механізм ARP спуфінгу

ARP спуфінг (також відомий як ARP poisoning) – це техніка атаки, при якій зловмисник надсилає підроблені ARP-повідомлення в локальну мережу. Мета – пов'язати MAC-адресу атакуючого з IP-адресою легітимного пристрою (зазвичай шлюзу за замовчуванням).

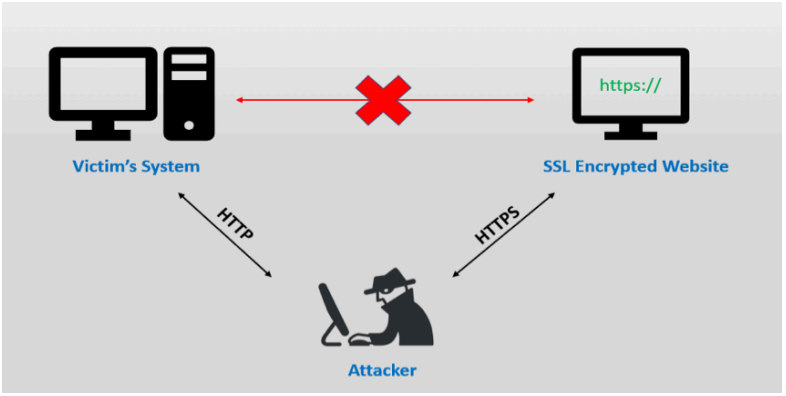
Процес атаки:



Принцип роботи SSL Strip

SSL Strip – це метод атаки типу "людина посередині" (MITM), розроблений Мокси Марлінспайком. Основна мета – понизити HTTPS з'єднання до HTTP, що дозволяє перехоплювати незашифрований трафік.

Етапи роботи:

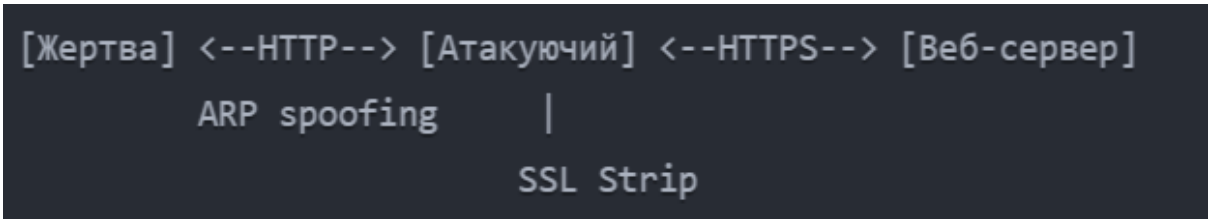


Інтеграція ARP спуфінгу та SSL Strip

Успішна реалізація комбінованої атаки з використанням ARP спуфінгу та SSL Strip вимагає ретельної підготовки середовища та розуміння мережевої архітектури. Ключовим аспектом є правильне налаштування системи атакуючого для перехоплення та модифікації мережевого трафіку. Розглянемо детально кожен етап підготовки та реалізації.

Першим кроком є налаштування системи атакуючого для коректної обробки перехопленого трафіку. У Linux-системах необхідно активувати IP-forwarding, що дозволяє системі виступати в ролі маршрутизатора. Це досягається модифікацією системних параметрів ядра. Активація IP-forwarding забезпечує безперервність мережевого з'єднання для жертви, приховуючи факт перехоплення трафіку.

Наступним важливим етапом є налаштування перенаправлення портів за допомогою iptables. Це необхідно для перехоплення HTTP-трафіку та його обробки утилітою sslstrip. Стандартний веб-трафік перенаправляється з порту 80 на порт 8080, де працює sslstrip.



Реалізація атаки складається з декількох паралельних процесів:

1. ARP спуфінг
2. Перехоплення та модифікація трафіку: SSL Strip прослуховує перенаправлений HTTP-трафік
3. Моніторинг та аналіз

Технічні особливості реалізації

При реалізації комбінованої атаки необхідно враховувати ряд технічних аспектів:

1. Стабільність мережевого з'єднання:
    - Необхідно забезпечити достатню пропускну здатність
    - Мінімізувати затримки при обробці пакетів
    - Підтримувати стабільність ARP-спуфінгу
2. Обробка SSL/TLS:
    - Коректна обробка SSL/TLS-сертифікатів
    - Управління сесіями для різних доменів
    - Обробка HSTS та інших механізмів безпеки

3. Масштабованість:
    - Можливість обробки множинних з'єднань
    - Ефективне управління ресурсами системи
    - Логування та аналіз даних

Комплексні методи захисту від атак

Захист від комбінованих атак з використанням ARP спуфінгу та SSL Strip вимагає комплексного підходу, що охоплює різні рівні мережевої інфраструктури. Розглянемо детально кожен аспект захисту.

Мережевий рівень захисту

На мережевому рівні необхідно впровадити наступні механізми захисту:

1. Використання статичних ARP-записів
2. Впровадження VLAN та мікросегментації
3. Системи виявлення вторгнень (IDS)

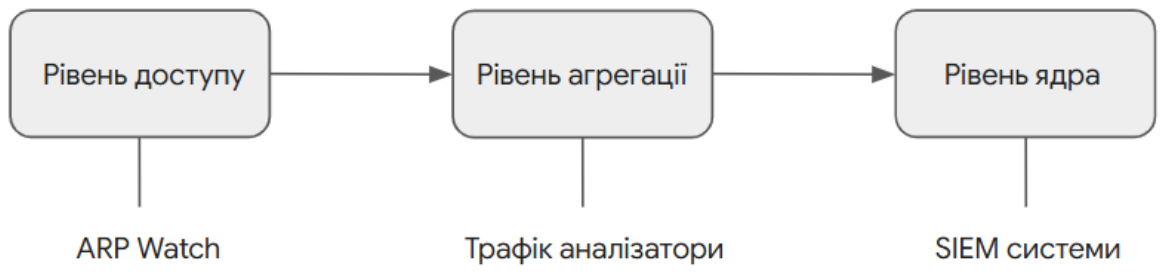
Захист веб-додатків

Для захисту від SSL Strip критично важливим є правильне налаштування веб-серверів та додатків:

1. Впровадження HSTS
2. Управління сертифікатами

Для ефективного виявлення та запобігання атакам необхідно впровадити системи моніторингу:

1. Мережевий моніторинг



2. Аналіз трафіку

- Використання спеціалізованих інструментів
- Збір та аналіз мережевої телеметрії
- Виявлення аномалій у режимі реального часу
- Автоматизоване реагування на інциденти

3. Централізований збір логів

Впровадження системи збору та аналізу логів, що забезпечує:

- Повну видимість мережевої активності
- Кореляцію подій безпеки
- Довгострокове зберігання даних
- Можливість forensic-аналізу

8. Наведіть рекомендації використання протоколу TLS для уникнення відомих вразливостей.

1) Використовуйте лише сучасні версії TLS - 1.2 та 1.3. Категорично відмовтеся від використання SSL 3.0 та TLS 1.0/1.1.
2) Використовуйте тільки сильні криптографічні набори шифрів, надаючи перевагу алгоритмам з Perfect Forward Secrecy (PFS).
3) Використовуйте сертифікати з належною довжиною ключа (мінімум RSA 2048 біт або ECC 256 біт)
4) Регулярно оновлюйте сертифікати.
5) Впровадьте HTTP Strict Transport Security (HSTS) з достатнім max-age.
6) Вимкніть стиснення на рівні TLS для запобігання CRIME атакам
7) Налаштуйте OCSP Stapling з must-staple
8) Регулярно перевіряйте конфігурацію на наявність вразливостей
9) Налаштуйте моніторинг закінчення терміну дії сертифікатів