Лабораторна робота 7

Тема: Тунелювання з’єднань з використанням протоколу SSL

Мета: Вивчити принципи безпечного обміну інформацією з використанням протоколу SSL та ОС Linux

1. Теоретичні відомості

HTTPS, SSL, TLS – протоколи шифрування даних. HTTPS — розширення протоколу HTTP, що підтримує шифрування. Дані, що передаються за протоколом HTTP, «упаковуються» в криптографічний протокол SSL або TLS, тим самим забезпечується захист цих даних. На відміну від HTTP, для HTTPS за замовчуванням використовується TCP порт 443. Цю систему було розроблено компанією Netscape Communications Corporation, щоб забезпечити аутентифікацію та захищене з'єднання. HTTPS широко використовується у світі для веб додатків, у яких важливою є безпека з'єднання, наприклад, у платіжних системах.

Рис. 1 – Послідовність перетворення протоколу HTTP до HTTPS

SSL (англ. Secure Sockets Layer — протокол захищених сокетів) - криптографічний протокол, що забезпечує безпечну передачу даних через мережу Інтернет. За допомогою його створюється захищене з'єднання між клієнтом і сервером. SSL розроблявся компанією Netscape Communications і зараз прийнятий IETF у якості стандарту.

Використовує шифрування з відкритим ключем для підтвердження справжності передавача та одержувача. Підтримує надійність передачі за рахунок використання коригувальних кодів і безпечних хеш-функцій.

SSL складається із двох рівнів. На нижньому рівні багаторівневого транспортного протоколу (наприклад TCP) він є протоколом запису і використовується для інкапсуляції (тобто формування пакета) різних протоколів. Для кожного інкапсульованого протоколу він забезпечує умови, за яких сервер і клієнт можуть підтверджувати один одному свою справжність, виконувати алгоритми шифрування і обмін криптографічними ключами, перш ніж протокол прикладної програми почне передавати і отримувати дані.

Для доступу до сторінок, захищених протоколом SSL, в URL замість звичайного префікса http застосовується префікс https (порт 443), що вказує на те, що буде використовуватися SSL-з'єднання.

Для роботи SSL потрібно, щоб на сервері був SSL-сертифікат.

Протокол SSL розробили фірмою Netscape. Його завдання:

1. Обов'язковість підтвердження автентичності сервером.
2. Опціональна перевірка автентичності клієнта.
3. Спільна генерація випадкового сеансового ключа.
4. Підтримка різних симетричних алгоритмів для шифрування даних.
5. Підтримка різних алгоритмів хешування для реалізації перевірки цілісності через MAC.

TLS (англ. Transport Layer Security) - криптографічний протокол, що забезпечує захищену передачу даних між вузлами в мережі Інтернет.

TLS-протокол заснований на Netscape SSL-протоколі версії 3.0 і складається з двох частин - TLS Record Protocol та TLS Handshake Protocol. Різниця між SSL 3.0 та TLS 1.0 є незначною. TLS Working Group, яку засновано у 1996 році, продовжує працювати над протоколом.

Алгоритм процедури встановлення з'єднання протоколу TLS handshake наступний. Клієнт і сервер, що працюють за TLS встановлюють з'єднання, використовуючи процедуру handshake (з англ. handshake - рукостискання). Протягом цього handshake, клієнт та сервер приймають угоду щодо параметрів, що використовуються для встановлення захищеного з'єднання.

Послідовність дій під час встановлення TLS з'єднання наступна.

Клієнт посилає ClientHello повідомлення, вказуючи останню версію протоколу, що підтримується TLS, випадкове число і список підтримуваних методів шифрування і стискання, придатних для роботи з TLS.

Сервер відповідає ServerHello повідомленням, що містить: вибрану сервером версію протоколу, випадкове число, надіслане клієнтом, відповідний алгоритм шифрування та стиснення зі списку наданого клієнтом.

Сервер посилає Certificate повідомлення, яке містить цифровий сертифікат сервера. Залежно від алгоритму шифрування, цей етап може бути пропущено. Сервер може запросити сертифікат у клієнта, у такому разі з'єднання буде взаємно автентифіковано.

Сервер надсилає ServerHelloDone повідомлення, що ідентифікує закінчення handshake.

Клієнт відповідає ClientKeyExchange повідомленням, яке містить PreMasterSecret відкритий ключ, або нічого, залежно від алгоритму шифрування.

Клієнт та сервер, використовуючи PreMasterSecret ключ та випадково згенеровані числа, обчислюють загальний секретний ключ. Решта інформації про ключ буде отримано із загального секретного ключа і згенерованих клієнтом та сервером випадкових значень.

Клієнт надсилає ChangeCipherSpec повідомлення, яке вказує на те, що всю наступну інформацію буде зашифровано встановленим у процесі handshake алгоритмом, використовуючи загальний секретний ключ.

Клієнт надсилає повідомлення Finished, яке містить хеш і MAC (код автентифікації повідомлення), що згенеровані на основі попередніх повідомлень handshake.

Сервер намагається розшифрувати Finished-повідомлення клієнта та перевірити хеш та MAC. Якщо процес розшифрування або перевірки не вдається, handshake вважається невдалим і з'єднання буде обірвано.

Сервер надсилає ChangeCipherSpec та зашифроване Finished повідомлення та у свою чергу клієнт теж виконує розшифрування та перевірку.

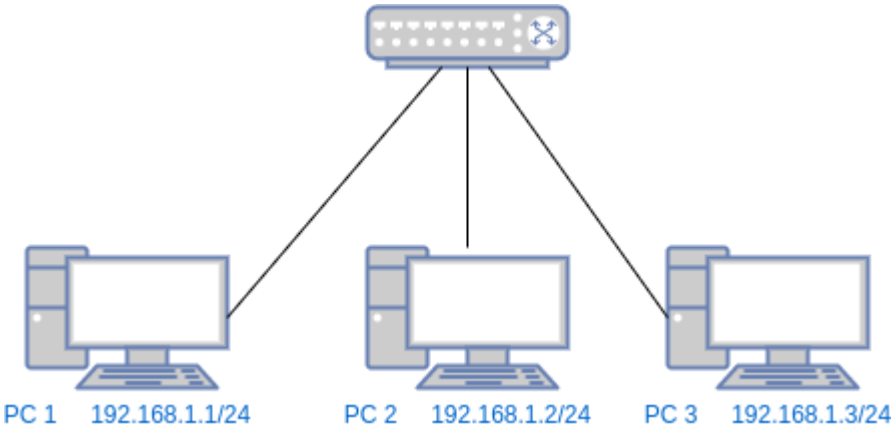
З цього моменту handshake вважається завершеним та встановленим протоколом. Увесь наступний вміст пакетів буде зашифровано.

Шифратор TCP-з’єднання Stunnel. Для шифрування з'єднання можна використовувати утиліту Stunnel. Це безкоштовна програма для шифрування довільних з'єднань TCP всередині SSL (Secure Sockets Layer), яка забезпечує безпеку не-SSL сервісів та протоколів (наприклад, POP, IMAP, LDAP, тощо) і не вимагає жодних змін у коді демону. Утиліта Stunnel не містить криптографічного коду, а використовує зовнішні бібліотеки SSL. Вона працює як із OpenSSL, так і з SSLeay. За допомогою програми Stunnel можна перевірити сертифікати, перетворити адреси, захиститися від перехоплення даних, захиститися від маніпулювання даними. Програма допоможе захиститися від IP маршрутизації до джерела та DNS спуфінгу.

Принцип роботи демона Stunnel наступний: зашифрований пакет, що надходить для сервісу, розшифровується Stunnel і передається незахищеному сервісу. Передавання відповіді від сервісу відбувається у зворотному порядку. До переваг Stunnel можна віднести також простоту встановлення та налаштування.

Завдання до роботи

1. Реалізуйте мережу, що представлено на схемі



   На комп’ютерах встановіть або запустіть ОС Linux. PC 1 виконує роль серверу, PC 2 — клієнта, PC 3 - зловмисник.
2. У налаштуваннях комутатору (switch) встановити режим “Mirroring Port” і впевнитись, що пакети каналу сервер-клієнт можна захопити на PC 3.
3. Налаштувати на PC 1 e-mail сервер. Автоматизувати авторизацію.
4. На PC 3 запустити Wireshark або tcpdump. Відправити пошту з комп’ютера клієнта. Визначити пакети з паролем на пошту.
5. На сервері встановіть (sudo apt install stunnel4) та запустить утиліту Stunnel на 995 порту для запуску поштового серверу.
6. Включить використання SSL/TLS на клієнті. Повторить етап 4.
7. Порівняти перехоплені пакети у двох випадках.
8. Підготуйте звіт.

Контрольні питання

1. Чим відрізняються протоколи SSL та TLS ?
2. Які принципи роботи протоколу SSL?
3. Які задачі виконує протокол SSL ?
4. Які принципи роботи протоколу TLS?
5. Які задачі виконує протокол TLS ?
6. Які принципи роботи протоколу HTTPS?
7. Для чого призначено утиліту Stunnel ?
8. Як використовується утиліта Stunnel ?