Практична робота 8. Віддалене адміністрування сервера за допомогою сервера терміналів OpenSSH

Мета: вивчення протоколу SSH та способів його застосування.

Теоретичні відомості

Протокол SSH (Secure SHell – «безпечна оболонка») – мережевий протокол сеансового рівня, що дозволяє здійснювати віддалене керування операційною системою та тунелювання TCP-з'єднань (наприклад, для передачі файлів). Схожий по функціональності з протоколами Telnet і rlogin, але, на відміну від них, шифрує весь трафік, включаючи паролі, що передаються. SSH припускає вибір різних алгоритмів шифрування. SSH-клієнти та SSH-сервери доступні для більшості мережевих операційних систем.

Перша версія протоколу, SSH-1, була розроблена 1995 р. дослідником Тату Улененом з Технологічного університету Гельсінкі, Фінляндія. SSH-1 був написаний для забезпечення більшої конфіденційності, ніж протоколи rlogin, telnet та rsh. У 1996 р. була розроблена безпечніша версія протоколу, SSH-2, несумісна з SSH-1. Протокол набув ще більшої популярності, і до 2000 р. він мав близько двох мільйонів користувачів. В даний час під терміном SSH зазвичай мається на увазі саме SSH-2, т.к. перша версія протоколу через істотні недоліки зараз практично не застосовується. У 2006 р. протокол був затверджений робочою групою IETF як Інтернет-стандарт. Перш ніж аналізувати протоколи ssh докладніше, слід визначити поняття ключ хоста.

Кожен хост, що працює з ssh, на якому може виконуватися як клієнт, так і сервер, може мати не менше одного ключа, причому для шифрування допускаються різні криптографічні алгоритми. Декілька хостів можуть мати загальний ключ хоста. Однак кожен хост повинен мати хоча б один ключ, з яким працює кожен із необхідних алгоритмів роботи з відкритими ключами. У проекті стандарту необхідний алгоритм лише один – DSS (Digital Signature Standard).

Протокол SSH розроблявся для надання безпеки даних, що передаються шляхом реалізації стійкого алгоритму шифрування даних, надійної системи аутентифікації користувача і сервера, наданням системи контролю цілісності переданих даних, а також інкапсуляцією додатків працюючих на основі протоколу TCP для встановлення безпечних тунелів.

Короткий опис та призначення кожного з трьох протоколів, що формують протокол SSH-2:

1. Протокол транспортного рівня – надає можливість шифрування та стиснення даних, що передаються, а також реалізує систему контролю цілісністю даних.

2. Протокол з'єднання – дозволяє клієнтам встановлювати багатопотокове з'єднання через оригінальний SSH тунель, таким чином знижуючи навантаження, яке створюють клієнтські процеси.

3. Протокол аутентифікації – протокол аутентифікації відокремлений від протоколу транспортного рівня, так як не завжди буває необхідним використання системи аутентифікації. Якщо потрібна аутентифікація, процес захищається оригінальним безпечним каналом, встановленим через протокол транспортного рівня.

Сам собою протокол транспортного рівня є достатнім для встановлення захищеного з'єднання, він є основою протоколу SSH-2 і протоколи з'єднання та аутентифікації базуються на ньому. Протокол аутентифікації відокремлений від протоколу транспортного рівня, так як іноді виникає ситуація, коли використання аутентифікації як не обов'язково, а й навіть небажано. Наприклад, якась організація надає на своєму FTP сервері анонімний доступ до патчів безпеки для будь-якої людини (або системи), яка захоче їх завантажити. У цьому випадку автентифікація не вимагатиметься, тоді як шифрування, стиснення та контроль цілісності даних забезпечуватиметься протоколом транспортного рівня. Більш того, за наявності каналу високої пропускної спроможності клієнти зможуть організувати багатопотокове з'єднання через оригінальне SSH з'єднання, використовуючи протокол з'єднання. Розробники проекту протоколу

ssh особливо дбали про його довговічність. Протокол буде розширюваним; планується можливість доповнення криптографічних алгоритмів, які використовуються під час роботи ssh. З цією метою проектом передбачено, що між клієнтом та сервером відбуваються переговори, в результаті яких вибираються методи шифрування, формати відкритих ключів тощо, які будуть використані у даному сеансі. При цьому з метою забезпечення інтероперабельності має підтримуватись певний мінімальний набір національних криптографічних стандартів.

Окремої уваги заслуговують питання збільшення трафіку у зв'язку із застосуванням протоколів ssh. Зрозуміло, що при передачі в мережі великих пакетів додаткове навантаження, викликане передачею заголовків ssh, що управляють, невелика. Основну увагу слід звернути на програми, яким характерні короткі пакети, наприклад, telnet. Мінімальний розмір заголовка TCP/IP дорівнює 32 байти; мінімальний розмір пакета при використанні ssh збільшиться з 33 до (приблизно) 51 байту. Враховуючи, що в Ethernet мінімальна довжина поля даних пакета дорівнює 46 байт, додатковим навантаженням в 5 байт можна знехтувати. Найбільш істотним впливом ssh є, ймовірно, при використанні протоколу PPP на низькошвидкісних модемних з'єднаннях, оскільки PPP стискає заголовки TCP/IP. Однак суттєвий прогрес у швидкостях передачі дозволяє розраховувати, що додаткові затримки будуть вимірюватися кількома мілісекундами і залишаться непомітні людині.

Поради щодо безпеки використання SSH

1. Заборона віддаленого root-доступу.

2. Заборона підключення з порожнім паролем або вимкнення входу паролем.

3. Вибір нестандартного порту SSH-сервера.

4. Використання довгих SSH2 RSA-ключів (2048 біт і більше). Системи шифрування на основі RSA вважаються надійними, якщо довжина ключа щонайменше 1024 біт.

5. Обмеження списку IP-адрес, з яких дозволено доступ (наприклад, налаштування файлового блоку).

6. Заборона доступу з деяких потенційно небезпечних адрес.

7. Відмова від використання поширених або широко відомих системних логінів для доступу до SSH.

8. Регулярний перегляд повідомлень про помилки автентифікації.

9. Встановлення систем виявлення вторгнень (IDS-Intrusion Detection System).

10. Використання пасток, що підробляють SSH-сервіс (honeypots).

OpenSSH – це набір вільних програм, що надають шифрування сеансів зв'язку через комп'ютерні мережі з використанням протоколу SSH. Більшість користувачів telnet, rlogin, ftp та інших подібних програм не усвідомлюють, що їхні паролі пересилаються через інтернет у незашифрованому вигляді. OpenSSH шифрує весь трафік (включаючи паролі) для запобігання підслуховування, перехоплення з'єднань та інших видів мережевих атак. Крім того, OpenSSH надає різні способи створення тунелів, численні методи автентифікації, а також підтримує всі версії протоколу SSH.

Пакет OpenSSH містить програми ssh для заміни rlogin і telnet, scp для заміни rcp, і sftp як альтернативу для ftp. Пакет також включає демон sshd, та інші утиліти, такі як ssh-add, ssh-agent, ssh-keysign, ssh-keyscan, ssh-keygen і sftp-server.

Встановлення пакету OpenSSH

Коли відбувається підключення до інших комп'ютерів, OpenSSH запускає два процеси. Перший процес є привілейованим і керує видачею прав доступу у міру виникнення в них потреби. Другий процес взаємодіє із мережею. Для того, щоб мати правильно налаштоване середовище, необхідне додаткове налаштування, яке можна виконати в ролі користувача root за допомогою наступних команд:

```
install -v -m700 -d /var/lib/sshd &&
chown -v root:sys /var/lib/sshd &&
```

```
groupadd -g 50 sshd &&
useradd -c 'sshd PrivSep' -d /var/lib/sshd -g sshd \
-s /bin/false -u 50 sshd
```

Пакет OpenSSH дуже чутливий до змін у прикомпонованих бібліотеках OpenSSL. Після повторної компіляції пакет OpenSSH може не запуститись. В якості альтернативи використовуйте посилання на статичну бібліотеку OpenSSL. Щоб зробити посилання на статичну бібліотеку, потрібно виконати наступну команду:

```
sed -i 's@-lcrypto@/usr/lib/libcrypto.a -ldl@' configure
```

Встановлюється пакет OpenSSH за допомогою наступних команд:

```
sed -i.bak 's/ -ldes//' configure &&
./configure --prefix=/usr \
--sysconfdir=/etc/ssh \
--datadir=/usr/share/sshd \
--libexecdir=/usr/lib/openssh \
--with-md5-passwords \
--with-privsep-path=/var/lib/sshd &&
make
```

Якщо tcp_wrappers скомпоновано з використанням параметру --with-tcp-wrappers і якщо у є файл з обмеженнями /etc/hosts.deny, потрібно переконатися, що в рядок sshd файлу /etc/hosts.allow додано адресу 127.0.0.1. В іншому випадку набір тестів не пройде. Крім того, тестовий набір вимагає встановлення копії scp для того, щоб можна було завершити виконання тестів, які використовують мультиплексування. Щоб запустити набір тестів, спочатку в директорій /usr/bin потрібно скопіювати програму scp, попередньо переконавшись, що перед цим зроблено резервні копії всіх файлів, що містяться там.

Щоб запустити тестовий набір, потрібно виконати наступні команди:

```
make tests 2>&1 | tee check.log
grep FATAL check.log
```

Якщо вказана вище команда не видасть фатальної помилки ("FATAL"), то в ролі користувача root можна перейти до встановлення:

```
make install &&
install -v -m755 -d /usr/share/doc/openssh-5.6p1 &&
install -v -m644 INSTALL LICENCE OVERVIEW README* WARNING.RNG \
/usr/share/doc/openssh-5.6p1
```

Конфігурація пакету OpenSSH

Конфігураційні файли

~/.ssh/, /etc/ssh/ssh_config ma /etc/ssh/sshd_config*

У ці файли вносити зміни не потрібно. Проте, можна вивчити файли /etc/ssh/ і внести деякі зміни, що відповідають вимогам безпеки системи. Однією з рекомендованих змін є заборона користувачеві root входити до системи через ssh. Щоб вимкнути можливість користувача root входити в систему через ssh, потрібно виконати в ролі користувача root наступну команду:

```
echo "PermitRootLogin no" >> /etc/ssh/sshd_config
```

Додаткову інформацію про конфігурування можна знайти на сторінках man команд sshd, ssh та ssh-agent.

Завантажувальний скрипт

Щоб запускати сервер SSH під час завантаження системи, потрібно встановити завантажувальний скрипт /etc/rc.d/init.d/sshd, який знаходиться в blfs-bootscripts-20230101.

```
make install-sshd
```

Опис пакету

Встановлені програми: scp, sftp, sftp-server, slogin, ssh, sshd, ssh-add, ssh-agent, ssh-keygen, ssh-keyscan та ssh-keysign

Встановлені директорії: /etc/ssh, /var/lib/sshd, /usr/lib/openssh та /usr/share/doc/openssh-5.6p1

Короткий опис

scp – програма копіювання файлів, що діє як програма rcp, за винятком лише того, що вона використовує захищений протокол.

sftp – є програмою, схожою на FTP, яка працює поверх протоколів SSH1 та SSH2.

sftp-server – є підсистемою сервера SFTP. Ця програма зазвичай не викликається безпосередньо користувачем.

ssh – клієнтська програма, схожа на rlogin/rsh, за виключенням лише того, що вона використовує захищений протокол.

sshd – демон, який через ssh приймає запити на вхід до системи.

ssh-add – інструментальний засіб, за допомогою якого до ssh-agent додаються ключі.

ssh-agent – є агентом аутентифікації, який може зберігати закриті ключі.

ssh-keygen – є інструментальним засобом генерації ключів.

ssh-keyscan – утиліта збору відкритих ключів із ряду хостів.

ssh-keysign – використовується програмою ssh для доступу до ключів локальних хостів та генерації цифрового підпису, необхідної для аутентифікації протоколу SSH версії 2 і при використанні окремого хоста. Ця програма зазвичай не викликається безпосередньо користувачем.

Завдання

1. Створити дві віртуальні машини (клієнт та сервер), налаштувати з’єднання між цими комп’ютерами, встановити на них ОС Linux та встановити OpenSSH.

2. Перевірити чи працюють на комп'ютерах ssh-сервери (ps ax|grep ssh). Якщо не працюють, то запустити їх відповідною командою.

3. Спробувати підключитися з клієнта до сервера за допомогою протоколу SSH. Вивчити передані пакети за допомогою tcpdump. Переглянути які події були відмічені у файлі журналу /var/log/auth.log

4. Перезавантажити віддалений сервер, не підключаючись до нього.

5. Віддалено перезапустити демон ssh без підключення.

6. Підключитися до віддаленого сервера та перезапустити демон ssh. Звернути увагу на те, що сеанс не завершився.

7. Запустіть FTP-сервіс на сервері.

8. Виконати передачу файлу через FTP за допомогою утиліти wget. Проаналізувати пакети, що проходять, за допомогою утиліти tcpdump(-XX). Запам'ятати швидкість передачі.

9. Виконати передачу через ssh. Проаналізувати пакети, що проходять, за допомогою утиліти tcpdump(-XX). Запам'ятати швидкість передачі. Порівняти пакети, що передаються, і швидкості передачі даних.

10. Увімкнути стиснення ssh і повторити вимірювання швидкості. У кожному тесті аналізувати результати для файлу, що складається з нулів, і для файлу, що складається з випадкових послідовностей (ddif= /dev/urandomor=fileds=1Mcount=10), для текстового конфігураційного файлу або бінарного файлу.

Відобразіть у звіті скріншоти з результатами виконання завдання.