Министерство образования и науки Российской Федерации

Московский Государственный Технический Университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

Сурков Л.В.

Методические указания

к лабораторной работе по дисциплине

Корпоративные сети

Раздел

Удаленное управление и мониторинг сети

Построение защищенных SSL – каналов на базе цифровых сертификатов

2011

Теоретическая часть

Важно! *Перед выполнением практической части работы рекомендуется внимательно прочитать Раздел «Основные принципы и понятия» Лабораторной работы «Развертывание служб сертификации корпоративной сети».*

Существует два основных класса криптографических алгоритмов - симметричные и ассиметричные. В симметричных для шифрования и дешифрования сообщения используется один и тот же ключ. В ассиметричных разные. Тот которым расшифровывают сообщения называется закрытым ключем, ключ для шифрования называется открытым. С помощью открытых и закрытых ключей можно подписывать документы. Для этого рядом с ключем сохраняют данные о владельце ключа и полученный файл называется сертификатом. Вся система взаимотношений между владельцами сертификатов построена на доверии к определенным пользователям. Их подписи общеизвестны, и они подписывают сертификаты других членов, подтверждая тем самым достоверность открытого ключа и хранящихся вместе с ним данных о владельце. Для того, что бы система не была скомпрометирована, пользователи принимают только сертификаты с подписями доверенных членов. Приведем более строгую терминологию:

Инфраструктура открытого ключа (PKI) является системой цифровых сертификатов, центров сертификации (ЦС), которая производит проверку и подтверждение подлинности каждой из сторон, участвующих в электронной операции, с помощью криптографии открытых ключей.

Сертификат открытого ключа, обычно называемый просто сертификатом - это документ с цифровой подписью, связывающий значение открытого ключа с удостоверением пользователя, устройства или службы, которым принадлежит соответствующий закрытый ключ.

Центр Сертификации (Certification Authority, CA) является пакетом программного обеспечения, принимающим и обрабатывающим запросы на выдачу сертификатов, издающим сертификаты и управляющим выданными сертификатами.

Корневой сертификат - сертификат принадлежащий Центру Сертификации, с помощью которого проверяется достоверность других выданных центром сертификатов.

Список отозванных сертификатов - список скомпрометированных или недействительных по какой-либо другой причине сертификатов.

Отличительное имя (Distinguished Name, DN) - данные о владельце сертификата. Включают CN (Common Name), OU (Organization Unit), O (Organization), L (Locality), ST (State or province), C (Country name).

Электронная цифровая подпись (ЭЦП)- реквизит электронного документа, предназначенный для удостоверения источника данных и защиты данного электронного документа от подделки.

Схема электронной подписи обычно включает в себя:

- алгоритм генерации ключей пользователя;
- функцию вычисления подписи;
- функцию проверки подписи.

Функция вычисления подписи на основе документа и секретного ключа пользователя вычисляет собственно подпись. Функция проверки подписи проверяет, соответствует ли данная подпись данному документу и открытому ключу пользователя. Открытый ключ пользователя доступен всем, так что любой может проверить подпись под данным документом.

Для того что бы подписать документ нужно зашифровать с помощью закрытого ключа значение хеш-функции от содержимого документа. Чтобы проверить подпись, нужно расшифровать с помощью открытого ключа значение подписи и убедиться, что оно равно хешу подписанного документа. Таким образом цифровая подпись, это зашифрованный хеш документа.

Ключ - это набор параметров (чисел). Он может храниться в файле. В теории клиенты должны сами генерировать свои закрытые и открытые ключи, создавать запрос на подпись открытого ключа и отправлять его в центр. На практике большинство клиентов не умеют генерировать ключи, поэтому в нашем случае Центр осуществляет данную работу. Центр может отзывать сертификат, выданный клиенту, помещая его в черный список, который регулярно передается пользователям центра. С помощью корневого сертификата, который публично доступен, пользователи могут проверять сертификаты друг друга.

Итого у центра сертификации имеется:

- закрытый ключ
- корневой сертификат (хранящий в себе открытый ключ);
- список отозванных (скомпрометированных) сертификатов

У клиентов:

- закрытый ключ;
- сертификат, подписанный корневым;
- корневой сертификат для проверки того, что сертификаты других пользователей выданы его доверенным центром;
- список отозванных (скомпрометированных) сертификатов.

Создание корневого сертификата включает:

1. Генерацию закрытого ключа.
2. Генерацию открытого ключа и его подпись с помощью закрытого.

Создание обычного сертификата включает:

1. Генерацию закрытого ключа.
2. Создание запроса на подпись сертификата.
3. Подпись запроса в центре сертификации о получение сертификата.

Общепринятый формат информации, содержащейся в сертификате, называется X509. Сертификаты и ключи могут храниться в разных типах файлов.

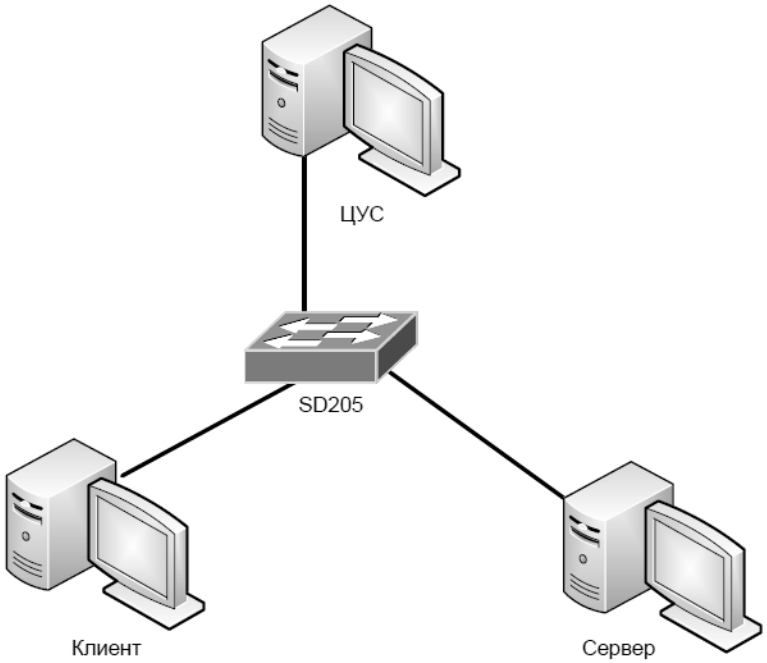
OpenSSL — криптографический пакет с открытым исходным кодом для работы с SSL/TLS. Позволяет создавать ключи RSA, DH, DSA и сертификаты X.509, подписывать их, формировать CSR и CRT. Также имеется возможность шифрования данных и тестирования SSL/TLS соединений

.

Практическая часть

Цель работы:

Изучение технологии цифровых сертификатов и получение навыков построения зашифрованной сети на базе этой технологии и ОС«Linux».



Рисунок

Порядок выполнения работы:

1. Соберите топологию сети, представленную на рисунке.

2. Запустите веб-сервис на сервере.

3. Инициируйте соединение клиента с веб-сервером и с помощью утилиты tcpdump убедитесь, что данные передаются без шифрования.

4. Создайте центр управления сертификатами (ЦУС) средствами *openssl*.

5. Создайте сертификат для веб-сервера. Настройте веб-сервер на работу с сертификатом.

6. Инициируйте соединение клиента с веб-сервером по протоколу HTTPS. Средствами браузера изучите сертификат. С помощью *tcpdump* изучите передаваемые пакеты и убедитесь, что передаваемая информация шифруется.

Литература, источники

1. James Boney, Cisco IOS in a Nutshell, 2-nd Ed., O`Reilly, 2008
2. Wendell Odom, Interconnecting Networking Devices Cisco, Part 2, Cisco Press, 2008
3. НПП «Учтех-Профи», ОС Arch Linux, Лабораторный практикум, Челябинск, 2010
4. OpenSSL - http://ru.wikipedia.org/wiki/OpenSSL
5. http://www.opennet.ru/base/sec/openssl.txt.html

Контрольные вопросы

1. Что подразумевается под понятием «сертификат»?
2. Для чего требуются сертификаты?
3. Чем отличаются закрытый и открытый ключи?
4. Какие существуют недостатки при использовании сертификатов?

Примечания

Краткое описание файлов:

bundleclient.sh	скрипт для упаковки необходимых для отправки клиенту файлов
bundleserver.sh	скрипт для упаковки необходимых для отправки серверу файлов
ca.conf	файл с параметрами корневого сертификата
ca.crt	корневой сертификат
ca.db.certs	Каталог, где хранятся выданные сертификаты
ca.key	закрытый ключ сертификата
ca.pfx	корневой сертификат в специальном формате
carevoke.config	файл с параметрами отзыва сертификатов
createca.sh	скрипт создания корневого сертификата
createclient.sh	скрипт создания клиентского сертфиката
createcrl.sh	скрипт создания списка отозваных сертификатов
createserver.sh	скрипт создания серверного сертификата
crl.pem	список отозваных сертификатов
revoke.sh	скрипт отзыва сертификата
sign1.sh	Скрипт, подписывающий серверные сертификаты
signclient.sh	Скрипт, подписывающий клиентские сертификаты

Пример создания сертификата:

Создание ключа и запроса:

#openssl req -new > new.cert.csr

Удаление пароля из ключа:
#openssl rsa -in privkey.pem -out new.cert.key

Конвертирование запроса в подписанный сертификат:

#openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey new.cert.key -days 365

Для клиентских сертификатов:

#openssl x509 -req -in client.cert.csr -out client.cert.cert -signkey my.CA.key -CA my.CA.cert -CAkey my.CA.key -CAcreateserial -days 365

****

Запуск веб-сервера осуществляется следующей командой:

 # /etc/rc.d/httpd start

Для добавление поддержки протокола SSL в файле *conf/httpd.conf*

раскомментируйте строку:

Include conf/extra/httpd-ssl.conf

Также требуется проверить конфигурационный файл *conf/extra/httpd-ssl.conf*

для сервера на наличие:

SSLCertificateFile /path/to/certs/new.cert.cert

SSLCertificateKeyFile /path/to/certs/new.cert.key

И для клиента:

SSLCACertificateFile /path/to/certs/my.CA.cert

SSLVerifyClient 2