Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт электронной техники и приборостроения

Кафедра Информационная безопасность автоматизированных систем

Специальность 10.05.03 Информационная безопасность автоматизированных систем

**КУРСОВАЯ РАБОТА**

по дисциплине «Безопасность сетей ЭВМ»

**Тема работы**

«Анализ уязвимостей в пакете программ OpenSSL»

|  |  |
| --- | --- |
|  | Выполнила: студент 4 курса  учебной группы с-ИБС42  очной формы обучения  Солодилов В.В. Руководитель работы:  доцент каф. ИБС Губенков А.А.  Комиссия по защите:  доцент каф. ИБС Губенков А.А.  доцент каф. ИБС Хороводова Н.Ю. |

Курсовая работа защищена на оценку \_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_2022 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись члена комиссии)

«\_\_\_»\_\_\_\_\_\_2022 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись члена комиссии)

Саратов 2022

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Кафедра Информационная безопасность автоматизированных систем

**Задание**

**на курсовую работу**

студенту 4 курса учебной группы с-ИБС42

Института электронной техники и приборостроения

Солодилову Владимиру Владимировичу

**Анализ уязвимостей в пакете программ OpenSSL**

Провести анализ уязвимостей в OpenSSL, воспользовавшись информацией из открытых источников (bdu.fstec.ru, web.nvd.nist.gov, cwe.mitre.org), сделать обзор уязвимостей. В обзор включать только уязвимости новее 2005 года.

Реализовать угрозу безопасности на примере конкретной уязвимости с помощью эксплойтов. Поиск эксплойтов осуществлять на сайтах metasploit.com, vulners.com, exploit-db.com. Оформить отчет по выполненной работе, процесс эксплуатации уязвимости продемонстрировать с помощью снимков экрана. В заключении дать рекомендации по защите систем от угроз безопасности информации.

Сроки выполнения 14.09.2022 г. - 23.12.2022 г.

Дата защиты 26.12.2022 г.

**Руководитель работы Губенков А.А.**

**Задание принял к исполнению Солодилов В.В.**

Содержание

**Введение**

В современном Интернете большинство соединений с сайтами устанавливается с использованием различных средств шифрования, включающих в себя протоколы SSL и TLS. Данные протоколы входят в OpenSSL – пакет программ, используемый для работы с различными криптографическими функциями библиотеки OpenSSL в консоли.

Протокол SSL был выпущен в 1995 году и был обновлен до TLS в 1999 году. С тех пор происходит активное развитие данного протокола. С появлением новых версий данного протокола большинство предыдущих его версий становятся устаревшими ввиду устаревания методов шифрования, в них используемых, а также появления различных уязвимостей, в том числе критических, которые ставят под угрозу возможное использование данного протокола. В связи с этим важно следить за выходом обновлений для протоколов SSL/TLS и использовать его актуальную версию.

В рамках курсовой работы проводится обзор протокола SSL/TLS, его особенностей, преимуществ и недостатков, а также пакета программ OpenSSL, использующий данный протокол. В дальнейшем рассматриваются выявленные уязвимости для OpenSSL, а также реализация угрозы с использованием уязвимости Heartbleed для OpenSSL 1.0 (дописать) на ОС Ubuntu Linux 12.04.4 LTS.

1. Основные термины и определения
   1. Понятие SSL

Протокол SSL является протоколом прикладного уровня, выпущенным компанией Netscape в 1995 году для защиты информации передаваемых по Интернету данных кредитных карт. В настоящий момент данный протокол располагается поверх протокола HTTP, что обеспечивает соединение клиента с сервером посредством уже более безопасного протокола HTTPS.

Основополагающими средствами обеспечения безопасности передачи информации в HTTPS с использованием протокола SSL являются:

* Взаимная аутентификация (известна как mutual challenge) – заключается в аутентификации идентификационных данных другой стороны. При передачи данных между двумя собеседниками первоначально каждый из них должен аутентифицировать идентификационные данные друг друга, после чего становится возможным непосредственная передача самих сообщений.
* Конфиденциальность – перед отправкой сообщения происходит его шифрование для дальнейшей передачи по каналу связи. После чего получатель расшифровывает исходное сообщение. При перехвате зашифрованного сообщения злоумышленником понадобится большая вычислительная мощность для его дешифрования, в то время как исходная информация может устареть.
* Целостность – гарантия совпадения полученного сообщения с отправленным. Реализуется на основе использования хэш-функций, также входящих в состав OpenSSL.
  1. Основные принципы работы SSL

Принцип работы протокола SSL заключается в реализации двух фаз: **фазы рукопожатия** и **фазы передачи данных**. При данных действиях активно используются асимметричные и симметричные алгоритмы шифрования, первых из которых предназначен для аутентификации сторон, второй – для сохранения конфиденциальности передаваемой по каналу связи информации.

При установке соединения между клиентом и сервером происходит фаза рукопожатия, при которой выполняются следующие действия:

* Клиент генерирует hello-сообщение и отправляет его серверу.
* В ответном сообщении сервер отправляет hello-сообщение клиенту, а также свой сертификат, который содержит публичный ключ.
* Клиент публичным ключом производит расшифровку сертификата сервера и проверяет полученные данные с отпечатком (digest) сертификата, который выдаётся центром сертификации. Если данные не совпадают, значит сертификат сервера был подделан, вследствие чего дальнейшее соединение с сервером прекращается.
* После проверки подлинности сервера, сервер производит шифрование открытым ключом для определения секретного ключа, используемого для обмена информацией.
* Производится взаимная отправка finished-сообщений с целью подтверждения согласия сторон на начало передачи данных.

Во время фазы передачи данных каждая сторона разбивает исходящие сообщения на фрагменты и прикрепляет к ним коды авторизации сообщений MAC (message authentication code). Для получения полноценного SSL пакета каждая из сторон объединяет данные фрагмента, код авторизации сообщения, заголовки сообщения и шифрует их с использованием секретного ключа. При получении пакета, каждая из сторон расшифровывает его и сверяет полученный код авторизации сообщения со своим. Если полученные в итоге проверки коды не совпадают, то можно сделать вывод, что пакет был подделан. Такой пакет отбрасывается.

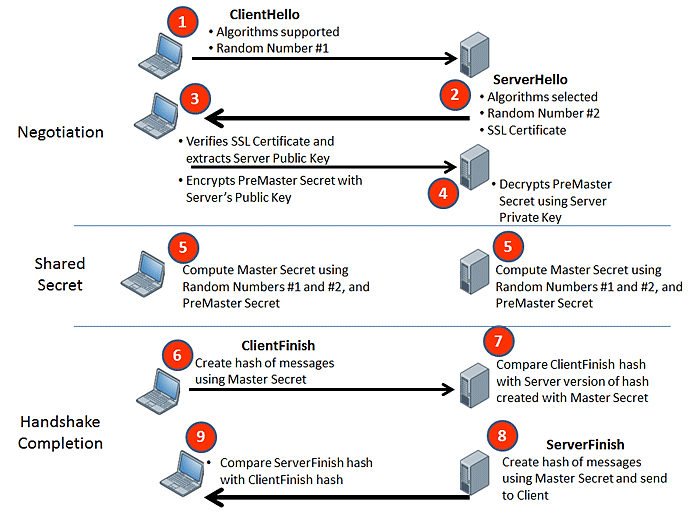


Рисунок 1 – Принцип работы SSL протокола

* 1. Цифровые сертификаты

В качестве сертификата владелец сайта также может использовать:

* Сертификат, выданный Центром сертификации (CA)
* Самоподписанный сертификат – сертификат, который создан самим пользователем. В данном случае издатель сертификата и его владелец совпадают.
* «Пустой» сертификат – сертификат, содержащий фиктивную информацию, используемую в качестве временной для настройки SSL и проверки его функциональности в данной среде.

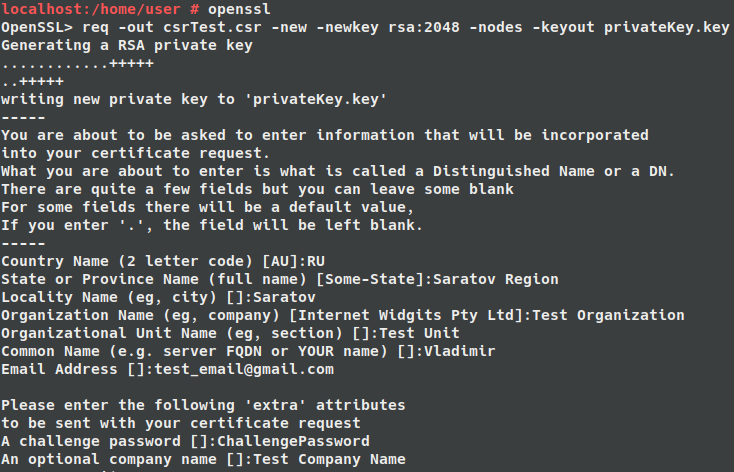
Для использования протокола SSL владельцу сервера необходимо получить соответствующий сертификат, который проверяется и выдаются **Центром сертификации (CA - Certificate Authorities)**. При подаче заявки на получение сертификата генерируется **CSR (Certificate Signing Request - запрос на получение сертификата)** с парой ключей на сервере владельца.

При получении сертификата сервера владельцу необходимо совершить следующие действия:

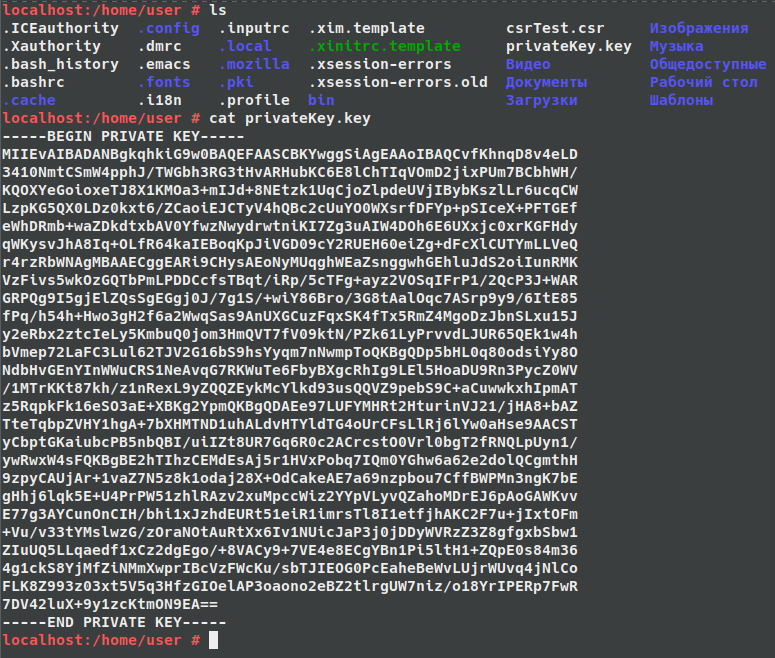
1. Сгенерировать CSR и пару ключей локально на вашем сервере. Пара ключей состоит из открытого (**public key**) и закрытого (**private key**) ключей.
2. Отправьте CSR и открытый ключ в центр сертификации, который проверит вашу личность, а также владеете ли вы доменом, указанным в заявке. Центр сертификации проверяет вашу организацию и проверяет, зарегистрирована ли организация в расположении, указанном в CSR, и существует ли домен.
3. После проверки организация получает копию своего сертификата SSL, включающего бизнес данные, а также открытый ключ. Теперь организация может установить сертификат на своем сервере.
4. Когда центр сертификации выдает сертификат, он связывается с сертификатом «доверенного корня» (trusted root) центра сертификации. Корневые сертификаты встроены в каждый браузер и связаны с индивидуально выданными сертификатами для установления HTTPS-соединения.
   1. Генерирование CSR в OpenSSL

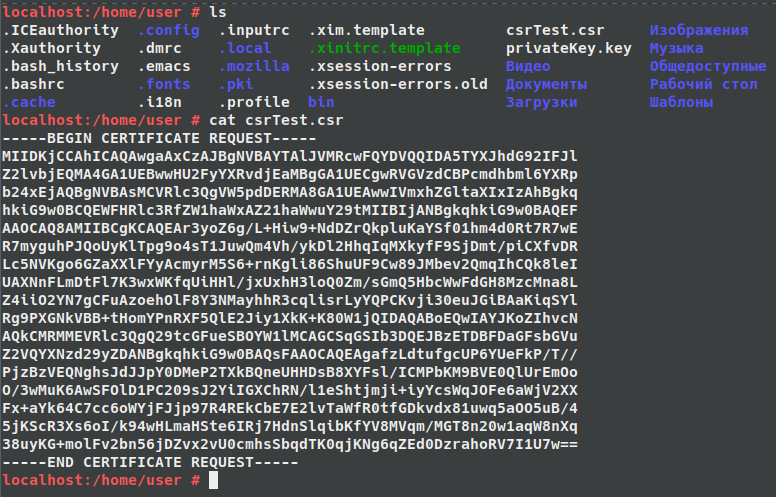
Сгенерируем CSR для получения сертификата. Для этого будем использовать OpenSUSE Leap 15.4 с предустановленным OpenSSL 1.1.1l.

Запустим OpenSSL, после чего сгенерируем запрос на создание приватного ключа и CSR, необходимых для получения сертификата. В процессе создания CSR-формы заполним её тестовыми данными:

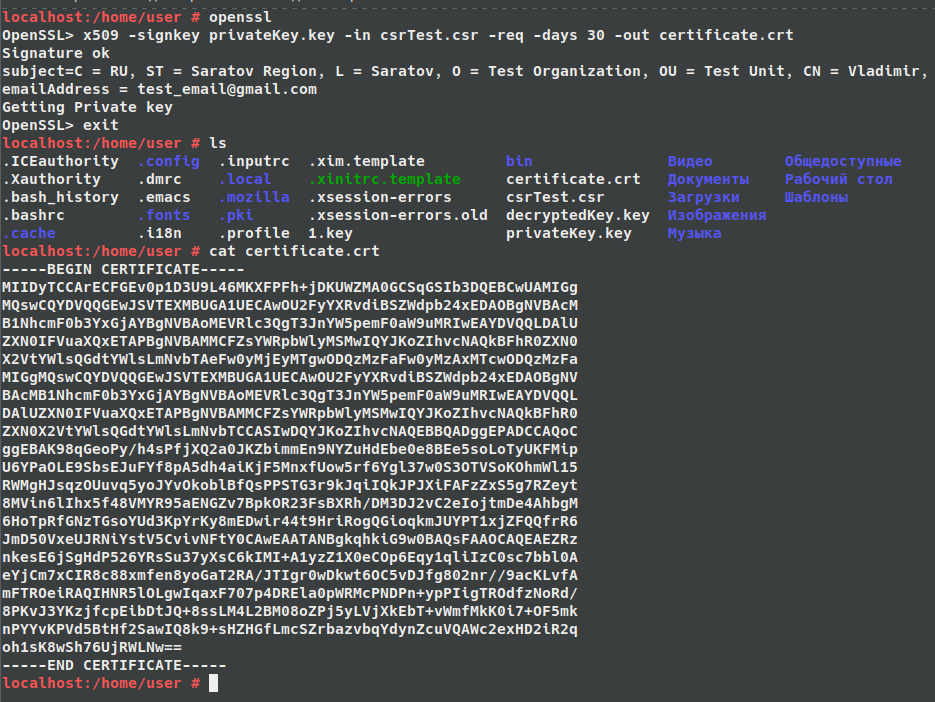


Результатом заполнения являются файлы с приватным ключом и запросом на получение сертификата:





В случае с владельцем сервера полученные данные можно передавать в Центр сертификации для получения необходимого сертификата. Для нашего случая можно получить самоподписанный сертификат, используя OpenSSL. Получим данный сертификат, после чего выведем его содержимое в консоль:



Данный сертификат можно использовать для проведения тестирования работоспособности SSL. Однако для непосредственной деятельности необходимо обратиться в Центр сертификации для получения необходимого сертификата.

Заключение

Список используемой литературы

ЛИСТ ЗАМЕЧАНИЙ