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  Колесникова Е.А. 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

Колесниковой Екатерине Алексеевне

**Анализ уязвимостей в пакете программ openSSL**

Провести анализ уязвимостей в \_\_\_\_\_\_\_\_\_, 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 Колесникова Е.А.**

**Содержание**

[1. Введение 4](#_Toc120907323)

[2. Основные термины и понятия 5](#_Toc120907324)

[3. Обзор уязвимостей openSSL приложений. 12](#_Toc120907325)

[3.1. BDU:2019-00765: Уязвимость программного обеспечения криптографической библиотеки OpenSSL, связанная с некорректной работой механизма «error state», позволяющая нарушителю передавать незашифрованные конфиденциальные данные по сети. 12](#_Toc120907326)

[3.2. BDU:2016-00661: Уязвимость библиотеки OpenSSL, позволяющая нарушителю расшифровать передаваемые данные. 13](#_Toc120907327)

[3.3. BDU: 2015-00644: Уязвимость программного обеспечения OpenSSL, позволяющая удаленному злоумышленнику нарушить доступность защищаемой информации. 13](#_Toc120907328)

[3.4. BDU: 2015-09820: Уязвимости операционной системы Red Hat Enterprise Linux, позволяющие удаленному нарушителю нарушить конфиденциальность, целостность и доступность защищаемой информации 14](#_Toc120907329)

[3.5. CVE-2014-0160: OpenSSL TLS Heartbeat Extension - 'Heartbleed'. 15](#_Toc120907330)

[4. Практическое применение 16](#_Toc120907331)

[5. Заключение 22](#_Toc120907332)

[Список использованной источников 23](#_Toc120907333)

# **Введение**

Актуальность данной проблемы связана с широким использованием программных средств, с использованием стандартизированной технологии шифрования, которая включает в себя протоколы SSL (Secure Sockets Layer) и TLS (Transport Layer Security). Программа ОpenSSL — это инструмент командной строки для использования различных криптографических функций криптографической библиотеки OpenSSL в консоли.

SSL была представлена в 1995 году и обновлена до TLS в 1999-м — технология видоизменялась, реагируя на новые уязвимости. Протоколы постоянно обновляются и большинство их предыдущих версий считаются устаревшими. Важно следить за обновлениями и использовать самую актуальную версию SSL/TLS.

В настоящее время протокол часто используется как один из этапов атаки, на котором либо происходит перехват сертификата, либо его подделка или изменение.

В рамках курсовой работы проводится обзор протокола SSL, описание его работы и особенностей, классификация удаленных атак на OС Linux. Далее рассматриваются атаки, относящиеся непосредственно к программе на базе OpenSSL.

# **Основные термины и понятия**

**Что такое OpenSSL?**

Программная система OpenSSL обеспечивает широкие возможности использования различных криптографических алгоритмов в прикладных программах. Начиная с версии 1.0.0, OpenSSL поддерживает и российские криптографические алгоритмы (ГОСТ 28147-89, ГОСТ Р34.11-94, ГОСТ Р34.10-2001).

OpenSSL - это библиотека программного обеспечения для приложений, которые защищают связь по компьютерным сетям от подслушивания или необходимости идентификации абонента на другом конце. Он широко используется интернет -серверами, включая большинство HTTPS -сайтов.

OpenSSL содержит реализацию протоколов SSL и TLS с открытым исходным кодом. Основная библиотека, написанная на языке программирования C, реализует основные криптографические функции и предоставляет различные служебные функции. Доступны оболочки, позволяющие использовать библиотеку OpenSSL на различных компьютерных языках.

**Особенности протокола** **SSL**

SSL (secure sockets layer — уровень защищённых cокетов) представляет собой криптографический протокол для безопасной связи.

Протокол SSL предоставляет различные меры обеспечения безопасности, две из которых являются основополагающими в HTTPS:

* Взаимная аутентификация (peer authentication, также известная как mutual challenge): каждая сторона соединения аутентифицирует идентификационные данные другой стороны. Если Алиса и Боб собираются обмениваться сообщениями через SSL, то сначала каждый должен аутентифицировать идентификационные данные своего собеседника.
* Конфиденциальность: отправитель шифрует сообщения перед их отправкой по каналу связи. Затем получатель расшифровывает каждое полученное сообщение. Этот процесс защищает сетевое взаимодействие. Даже если злоумышленник Ева перехватит зашифрованное сообщение от Алисы к Бобу (атака через посредника), то она будет не в состоянии расшифровать это сообщение из-за необходимой для этого вычислительной сложности.

Ещё одной мерой безопасности SSL является поддержка целостности сообщений это гарантирует, что полученное сообщение совпадает с отправленным. Эта функция реализована на основе хеш-функций, которые также входят в набор инструментов OpenSSL.

**Принцип работы SSL**

Цель протокола — обеспечить защищенную передачу данных. При этом для аутентификации используются асимметричные алгоритмы шифрования (пара открытый — закрытый ключ), а для сохранения конфиденциальности — симметричные (секретный ключ). Первый тип шифрования более ресурсоемкий, поэтому его комбинация с симметричным алгоритмом помогает сохранить высокую скорость обработки данных.

Принцип работы SSL состоит из двух фаз: **фаза рукопожатия** и **фаза передачи данных**. Во время фазы рукопожатия клиент и сервер используют шифрование открытым ключом для того, чтобы определить параметры секретного ключа, используемого клиентом и сервером для шифрования во время фазы передачи данных.

Клиент инициирует рукопожатие посылая “*hello*”-сообщение серверу. Такое сообщение содержит список алгоритмов симметричного шифрования (cipher specs), поддерживаемых клиентом. Сервер отвечает похожим “*hello*”-сообщением, выбрав при этом наиболее подходящий алгоритм шифрования из полученного списка. Далее сервер отправляет сертификат, который содержит его **публичный ключ**.

**Сертификат** - это набор данных, который подтверждает подлинность. Подтвержденная третья сторона, известная как **центр сертификации** (CA), генерирует сертификат и проверяет его подлинность. Чтобы получить сертификат сервер должен использовать безопасные каналы для отправки своего публичного ключа в центр сертификации. Он генерирует сертификат, который содержит его собственный ID, ID сервера, публичный ключ сервера и другую информацию. А также центр сертификации создает отпечаток (digest) сертификата, который, по сути, является контрольной суммой. Далее центр сертификации создает подпись сертификата (certificate signature), которая формируется путем шифрования отпечатка сертификата приватным ключом центра сертификации.

Для проверки сертификата сервера клиент использует публичный ключ центра сертификации для расшифровки подписи. Затем клиент самостоятельно считает отпечаток сертификата сервера и сверяет с расшифрованным. Если они не совпадают, то сертификат был подделан. Естественно, для расшифровки подписи у клиента должен быть публичный ключ центра авторизации. Поэтому клиент хранит у себя список публичных ключей подтвержденных центров сертификации. По факту, многие браузерные приложения имеют подобный список, находящийся непосредственно в их коде. Когда клиент установил подлинность сервера (сервер также может запросить сертификат у клиента), сервер использует шифрование открытым ключом для определения **секретного ключа** для обмена информацией.

Фаза *рукопожатия* завершается отправкой “*finished*”-сообщений, как только обе стороны готовы начать использование секретного ключа. Начинается *фаза передачи данных*, в ходе которой каждая сторона разбивает исходящие сообщения на фрагменты и прикрепляет к ним коды авторизации сообщений **MAC** (message authentication code). **Код авторизации сообщения**- это зашифрованный отпечаток, вычисленный на основе содержимого сообщений. Из соображений безопасности, он не совпадает с секретным ключом и вычисляется вместе с секретным ключом на стадии рукопожатия. Для получения полноценного SSL пакета каждая из сторон объединяет данные фрагмента, код авторизации сообщения, заголовки сообщения и шифруют с использованием секретного. При получении пакета, каждая из сторон расшифровывает его и сверяет полученный код авторизации сообщения со своим. Если они не совпадают, то пакет был подделан.

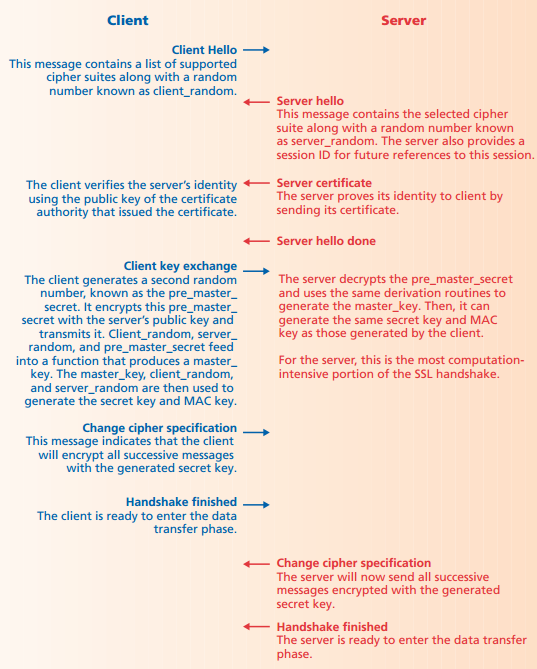


Рисунок 1 - Принцип работы SSL протокола

**Цифровые сертификаты**

Протокол SSL использует сертификаты для проверки соединения. Сертификаты расположены на безопасном сервере и используются для шифрования данных и идентификации Web-сайта.

Способы получения SSL-сертификата:

1. Использовать сертификат, выданный центром сертификации (Certification authority)
2. Использовать самоподписанный сертификат
3. Использовать «пустой» сертификат

Самоподписанный сертификат — сертификат, созданный самим пользователем — в этом случае издатель сертификата совпадает с владельцем сертификата. «Пустой» сертификат — сертификат, содержащий фиктивную информацию, используемую в качестве временной для настройки SSL и проверки его функциональности в данной среде.

**Шифрование**

Существует два основных способа шифрования данных: симметричный ключ (общий секретный ключ) и асимметричный ключ (открытый ключ).

1. Открытый ключ

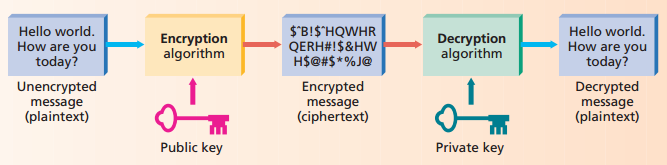


Рисунок 2 - Асимметричный ключ

Суть асимметричного шифрования заключается в том, что используется пара ключей. Один из них используется в качестве открытого (как правило, он публикуется в самом сертификате владельца), второй ключ называется секретным — он держится в тайне и никогда никому не открывается. Оба ключа работают в паре: один используется для запуска противоположных функций другого ключа. Если открытый ключ используется для того, чтобы зашифровать данные, то расшифровать их можно только секретным ключом и наоборот. Такая взаимосвязь позволяет делать две важные вещи.

Любой пользователь может получить открытый ключ по назначению и использовать его для шифрования данных, расшифровать которые может только пользователь, владеющий секретным ключом. (RSA)

Если заголовок шифрует данные, используя свой секретный ключ, каждый может расшифровать данные, используя соответствующий открытый ключ. Именно это является основой для цифровых подписей. (DSA)

**RSA** — самый распространенный алгоритм шифрования с использованием асимметричных ключей.

1. Секретный ключ

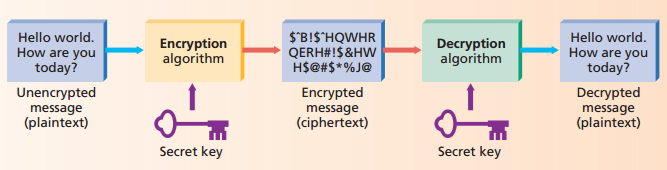


Рисунок 3 - Симметричный ключ

При шифровании секретным ключом используется один и тот же ключ для шифрованных данных. Если стороны хотят обменяться зашифрованными сообщениями в безопасном режиме, то у обеих сторон должны быть одинаковые симметричные ключи. Такой тип шифрования используется для большого объёма данных. Обычно используются алгоритмы DES, 3-DES, RC2, RC4 и AES

**Преимущества и недостатки SSL**

SSL-сертификаты обеспечивают защиту передачи данных и их конфиденциальность в Интернете. Высокая эффективность этого решения сделала его стандартом во всем мире, и теперь у каждого есть возможность видеть «зеленый замочек» в адресной строке браузера при использовании онлайн платежей.

Преимущества использования SSL-сертификатов:

1. Предоставление пользователям конфиденциальности, целостности, предотвращая утечку личных данных.
2. Гарантия подлинности и правильности загружаемых данных.
3. Ощущение комфорта при серфинге и удовлетворенность клиентов услугами на высоком уровне.
4. Сравнительно низкая стоимость и простота установки на любом хостинге.

Недостатки SSL-сертификатов:

1. Стоимость сертификата. Можно получить бесплатный сертификат SSL, но это не рекомендуется по многим причинам. Цена будет варьироваться в зависимости от типа сертификата, который вы покупаете. Однако, если учесть дополнительный уровень безопасности, стоимость для большинства веб-сайтов не является чрезмерной.
2. Смешанные режимы. Если ваша реализация SSL настроена неправильно, и у вас все еще есть файлы, обслуживаемые через HTTP, а не через HTTPS, посетители получат предупреждающее сообщение в своем браузере, сообщающее им, что некоторые данные не защищены. Это может сбивать с толку некоторых посетителей сайта.
3. Кэширование прокси. Еще одна возможная проблема - если на вашем веб-сервере настроена сложная система кеширования прокси. Зашифрованный контент нельзя будет кэшировать. Чтобы обойти это, вам нужно добавить сервер для обработки шифрования, прежде чем оно попадет на сервер кеширования. Это потребует дополнительных затрат, но это хороший способ убедиться, что вы шифруете данные посетителей, когда они заходят на ваш сайт.

# **Обзор уязвимостей openSSL приложений.**

# **BDU:2019-00765: Уязвимость программного обеспечения криптографической библиотеки OpenSSL, связанная с некорректной работой механизма «error state», позволяющая нарушителю передавать незашифрованные конфиденциальные данные по сети.**

Описание уязвимости: уязвимость программного обеспечения криптографической библиотеки OpenSSL связана с некорректной работой механизма «error state», в случае если функции SSL\_read() или SSL\_write() вызываются напрямую приложением. Эксплуатация уязвимости может позволить нарушителю передавать незашифрованные конфиденциальные данные по сети на уровне SSL/TLS.

Вендор: OpenSSL Software Foundation, Siemens AG.

Наименование ПО: OpenSSL, MindConnect IoT2040, SIMATIC ET 200SP Open Controller CPU 1515SP PC, MindConnect Nano (IPC227D), SIMATIC HMI WinCC Flexible, SIMATIC IPC DiagMonitor, SIMATIC WinCC OA, SIMATIC WinCC (TIA Portal), SIMATIC S7-1200, SIMATIC S7-1500, SIMATIC S7-1500 Software Controller, SIMATIC STEP 7 (TIA Portal), SINUMERIK Integrate Operate Client, SIMATIC IPC DiagBase, SINUMERIK Integrate Access MyMachine.

Тип ПО: программное средство защиты, ПО программно-аппаратного средства АСУ ТП, Программное средство АСУ ТП.

Тип ошибки: обработка ошибок.

Идентификатор типа ошибки: CWE-388.

Класс уязвимости: уязвимость кода.

Дата выявления- 07.08.2018.

Уровень опасности уязвимости: средний уровень опасности (базовая оценка CVSS 2.0 составляет 5,4).

Средний уровень опасности (базовая оценка CVSS 3.0 составляет 5,9).

Возможные меры по устранению уязвимости: обновление программного обеспечения.

# **BDU:2016-00661: Уязвимость библиотеки OpenSSL, позволяющая нарушителю расшифровать передаваемые данные.**

Описание уязвимости: уязвимость библиотеки OpenSSL связана с тем, что протокол SSLv2 требует отправки с сервера сообщения ServerVerify перед установкой соединения, вследствие чего клиент обладает частью открытого текста RSA. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, расшифровать данные TLS.

Вендор: OpenSSL Software Foundation.

Наименование ПО: OpenSSL.

Версия ПО: от 1.0.1s до 1.0.2g.

Тип ошибки: проблемы использования криптографии.

Идентификатор типа ошибки: CWE-310.

Класс уязвимости: уязвимость кода.

Дата выявления 01.03.2016.

Уровень опасности уязвимости - средний уровень опасности (базовая оценка CVSS 2.0 составляет 4,3).

Возможные меры по устранению уязвимости: обновление программного обеспечения до версии 1.0.2g и более поздней.

Статус уязвимости: подтверждена производителем.

# **BDU:** **2015-00644: Уязвимость программного обеспечения OpenSSL, позволяющая удаленному злоумышленнику нарушить доступность защищаемой информации.**

Описание уязвимости: двойное освобождение в d1\_both.c в реализации DTLS для OpenSSL позволяет злоумышленникам, действующим удаленно, вызвать отказ в обслуживании (аварийное завершение работы приложения) при помощи специально сформированных пакетов DTLS, вызывающих ошибку.

Вендор: OpenSSL Software Foundation

Наименование ПО: OpenSSL

Версия ПО: от 0.9.8 до 0.9.8zb; от 1.0.0 до 1.0.0n; от 1.0.1 до 1.0.1i

Дата выявления: 06.08.2014

Уровень опасности уязвимости: средний уровень опасности (базовая оценка CVSS 2.0 составляет 5)

Возможные меры по устранению уязвимости: для устранения уязвимости необходимо установить последнюю версию продукта, соответствующую используемой платформе.

Идентификаторы других систем описаний уязвимостей: CVE: CVE-2014-3505

# **BDU:** **2015-09820: Уязвимости операционной системы Red Hat Enterprise Linux, позволяющие удаленному нарушителю нарушить конфиденциальность, целостность и доступность защищаемой информации**

Описание уязвимости: множественные уязвимости пакета openssl операционной системы Red Hat Enterprise Linux, эксплуатация которых может привести к нарушению конфиденциальности, целостности и доступности защищаемой информации. Эксплуатация уязвимостей может быть осуществлена удалённо.

Вендор: Red Hat Inc., OpenSSL Software Foundation.

Наименование ПО: Red Hat Enterprise Linux, OpenSSL.

Версия ПО: Server 6 (Red Hat Enterprise Linux); HPC Node 6 (Red Hat Enterprise Linux); Desktop 6 (Red Hat Enterprise Linux); Workstation 7 (Red Hat Enterprise Linux); Server 7 (Red Hat Enterprise Linux); HPC Node 7 (Red Hat Enterprise Linux); Desktop 7 (Red Hat Enterprise Linux); Server EUS 6.6.z (Red Hat Enterprise Linux); Workstation 6 (Red Hat Enterprise Linux); до 0.9.8zf (OpenSSL); от 1.0.0 до 1.0.0r (OpenSSL); от 1.0.1 до 1.0.1m (OpenSSL); от 1.0.2 до 1.0.2a (OpenSSL).

Тип ошибки: уязвимость кода.

Идентификатор типа ошибки: CWE-17.

Класс уязвимости: уязвимость кода.

Дата выявления - 23.03.2015.

Уровень опасности уязвимости Высокий уровень опасности (базовая оценка CVSS 2.0 составляет 7,5).

Высокий уровень опасности (базовая оценка CVSS 3.0 составляет 7,3).

Возможные меры по устранению уязвимости: обновление пакета до более новой версии.

Способ устранения: обновление программного обеспечения.

# **CVE-2014-0160: OpenSSL TLS Heartbeat Extension - 'Heartbleed'.**

Ошибка Heartbleed является серьезной уязвимостью в популярной криптографической библиотеке программного обеспечения OpenSSL. Эта уязвимость позволяет украсть информацию, защищенную в обычных условиях шифрованием SSL / TLS, используемым для защиты Интернета. Этот баг позволяет любому пользователю Сети прочитать память на сервере, который защищен уязвимой версией библиотеки. В худшем случае этот небольшой блок памяти может содержать что-то важное — имена пользователей, пароли или даже секретный ключ, который используется сервером для шифровки соединений с клиентами. К тому же эксплуатация Heartbleed не оставляет следов, поэтому нет надежного способа определить, был ли сервер взломан и какие данные при этом украдены. Это ставит под угрозу секретные ключи, используемые для идентификации поставщиков услуг и шифрования трафика, имена и пароли пользователей и фактический контент. Это позволяет злоумышленникам подслушивать сообщения, красть данные непосредственно у служб и пользователей и выдавать себя за службы и пользователей.

# **Практическое применение**

**Обзор уязвимости CVE-2014-0160: OpenSSL TLS Heartbeat Extension - 'Heartbleed'**

Эту уязвимость называют кровоточащим сердцем потому, что когда она используется, это приводит к утечке содержимого памяти с сервера на клиент и с клиента на сервер



Рисунок 4 - Вырезка из документа

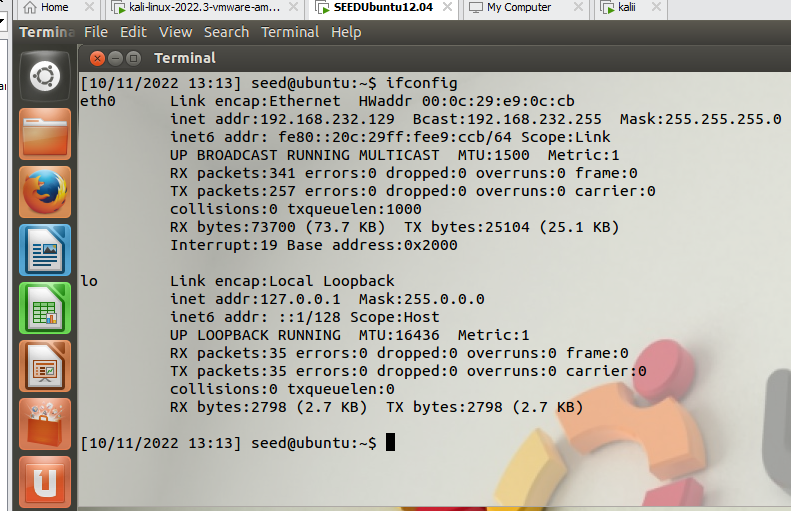
Статус разных версий:

* OpenSSL с 1.0.1 по 1.0.1f (включительно) уязвимы
* OpenSSL 1.0.1g НЕ уязвим
* Ветка OpenSSL 1.0.0 НЕ уязвима
* Ветка OpenSSL 0.9.8 НЕ уязвима

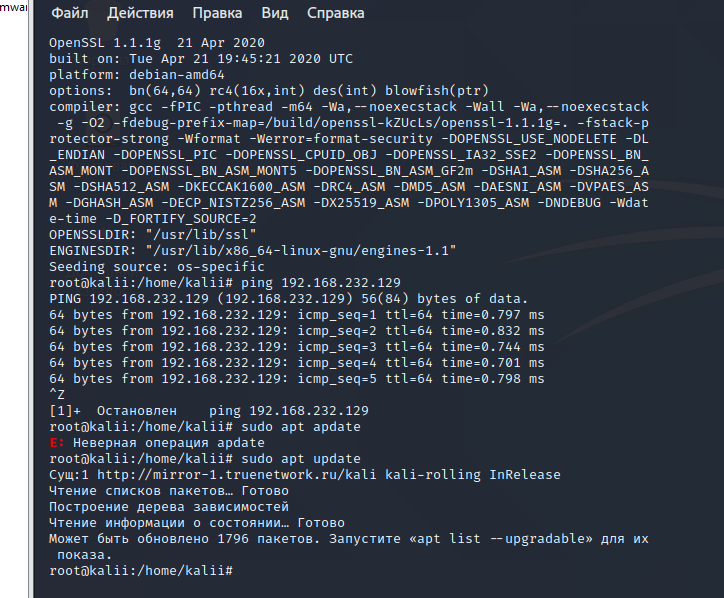
Ошибка была введена в OpenSSL в декабре 2011 года и была распространена с момента выпуска OpenSSL 1.0.1 14 марта 2012 года. OpenSSL 1.0.1g, выпущенный 7 апреля 2014 года, исправляет ошибку.

**Ход работы:**

Узнала IP жертвы:



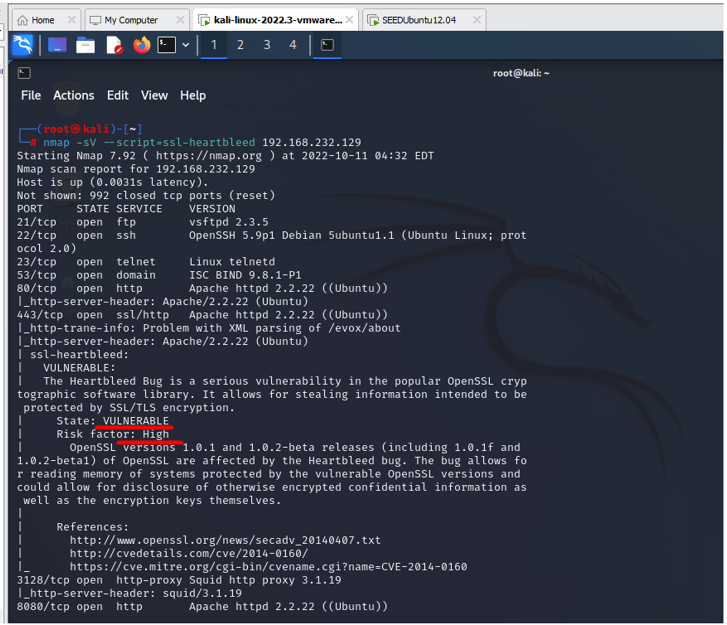
Убедилась, что стоит нужная версия OpenSSL, затем пропинговала:



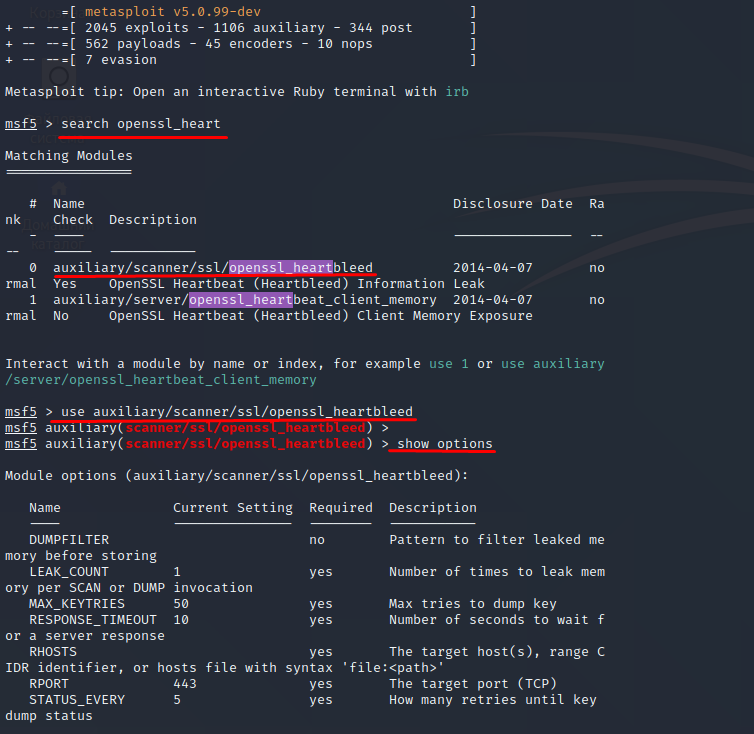
С kali проверила наличие уязвимости heartbleed командой:

*nmap –sV –script=ssl-heartbleed 192.168.232.129,*

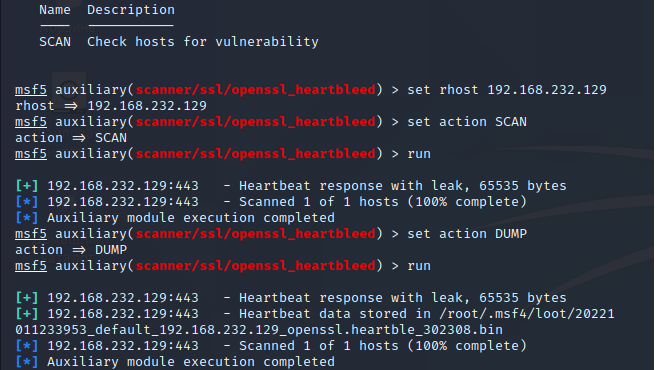
результат показывает, что данная уязвимость присутствует и риск её высок, так же видно описание уязвимости и открытые порты:



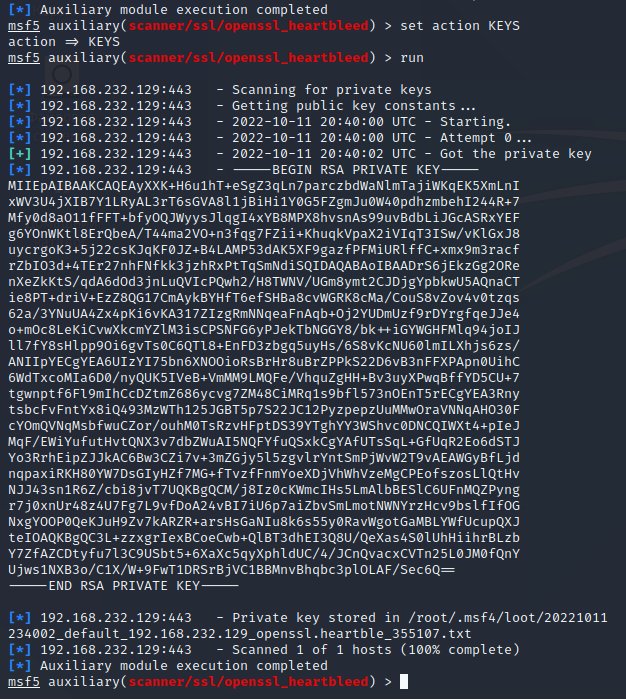
Производим запуск metasploit, ищем эксплойт по названию и смотрим его опции:



Указали IP-адресс жертвы, и первым делом просканировали, затем собрали трафик:



После чего достаём ключ шифрования:



Меры по устранению уязвимости: пока используется уязвимая версия OpenSSL, ею можно злоупотреблять. Исправная версия OpenSSL была выпущена, и должна быть установлена. Поставщики и дистрибьюторы операционных систем, поставщики устройств, независимые поставщики программного обеспечения должны принять исправление и уведомить своих пользователей. Поставщики услуг и пользователи должны устанавливать исправное ПО по мере его появления для операционных систем, сетевых устройств и программного обеспечения, которые они используют.

# **Заключение**

В рамках курсовой работы, были изучены основные термины и принцип работы протокола SSL. Рассмотрены некоторые уязвимости, связанные с данным протоколом, а также была реализована одна из уязвимостей. После всего этого можно сделать вывод, что в настоящее время все ценят безопасность и многие компании много делают в этом направлении, тема SSL-сертификатов заслуживает внимания. Они позволяют защититься от изменения страницы третьей стороной при передаче по сети Интернет. Таким образом, использование SSL сертификата в настоящее время можно считать необходимым для каждого веб-сайта. Есть несколько сервисов, которые предоставляют бесплатные сертификаты, которые позволяют стать веб-сайту безопаснее, например, Let’s Encrypt. Наличие на веб-ресурсе «зеленого замочка» в поисковой системе, естественно, побуждает воспользоваться различными услугами на данном сайте.

# **Список использованной источников**

1. Губенков А.А., Байбурин В.Б. Информационная безопасность: уч. пос. / А.А. Губенков, В.Б. Байбурин. – М.: ЗАО «Новый издательский дом», 2005. – 128 с.
2. SSL. [Электронный ресурс] – https://ru.wikipedia.org/wiki/SSL
3. Exploit database [Электронный ресурс]. – Режим доступа: https://www.exploit-db.com/?platform=linux&port=21
4. The Heartbleed Bug [Электронный ресурс]. – Режим доступа: https://heartbleed.com.
5. Форум xaker.ru [Электронный ресурс]. – Режим доступа: https://xaker.ru/2019/09/20/top-25-cwe/
6. Интернет-ресурс SequrityLab by Positive Technologies [Электронный ресурс]. – Режим доступа: https://www.sequritylab.ru/news/362606.php
7. Эриксон Дж. Хакинг. Искусство эксплойта. 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2010.-512 с.
8. Ховард М. 24 смертных греха компьютерной безопасности: как написать безопасный код / М. Ховард, Д. Лебланк, Дж. Вьега. – Пер. с англ. – СПб.: Питер, 2010. – 384 с.
9. ГОСТ Р 53114-2008 Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения. [Электронный ресурс]. –URL.: http://docs.cntd.ru/document/1200075565
10. ГОСТ Р 50922-2006 Защита информации. Основные термины и определения. [Электронный ресурс]. – URL.: http://docs.cntd.ru/document/1200058320
11. Доктрина информационной безопасности РФ. [Электронный ресурс]. – URL.: https://rg.ru/2016/12/06/doktrina-infobezopasnost-site-dok.html
12. Интернет-ресурс Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Kali\_Linux 16. Форум Hackware. – Режим доступа: https://hackware.ru/?p=2900#2

ЛИСТ ЗАМЕЧАНИЙ