

Отчет по лабораторной работе №7 :  
Сервис тестирования корректности настройки  
SSL на сервере Qualys SSL Labs – SSL Server  
Test

Дедков Сергей

2015

## Содержание

<b>1</b>	<b>Цель работы</b>	<b>2</b>
<b>2</b>	<b>Ход работы</b>	<b>2</b>
2.1	Изучить лучшие практики по развертыванию SSL/TLS . .	2
2.2	Изучить основные уязвимости и атаки на SSL последнего времени – POODLE, HeartBleed . . . . .	2
2.3	Обзор домена из списка Recent Best . . . . .	4
2.4	Обзор домена из списка Recent Worst . . . . .	5
2.5	Обзор известного сайта по выбору . . . . .	7
2.5.1	Интерпретировать результаты в разделе Summary .	7
2.5.2	Расшифровать все аббревиатуры шифров в разделе Configuration . . . . .	10
2.5.3	Прокомментировать большинство позиций в разделе Protocol Details . . . . .	12
2.5.4	Сделать итоговый вывод о реализации SSL на заданном домене . . . . .	13
<b>3</b>	<b>Вывод</b>	<b>13</b>

## 1 Цель работы

## 2 Ход работы

### 2.1 Изучить лучшие практики по развертыванию SSL/TLS

- Приватный ключ и сертификат

Качество защиты, обеспечиваемой TLS полностью зависит от секретного ключа, закладываемого основу безопасности, и сертификата, который сообщает о подлинности сервера для его посетителей

- Используйте 2048-битные закрытые ключи

Используйте 2048-битный RSA или 256-битные ECDSA закрытые ключи для всех ваших серверов. Ключи такой крепости безопасны и будут оставаться безопасными в течение значительного периода времени. Если у вас есть 1024-битные RSA ключи, то следует заменить их более сильными ключами как можно скорее.

- Защитите закрытый ключ

Относитесь к закрытым ключам как к важным активам, предоставляя доступ к как можно меньшей группе сотрудников. Рекомендуемые меры:

- Генерируйте закрытые ключи и запросы на сертификат (CSRs) на доверенном компьютере. Некоторые CA предлагают генерацию ключей и CSRs для вас, но это нецелесообразно.

- Используйте парольную защиту закрытых ключей, чтобы предотвратить их компрометацию в тех случаях, когда они хранятся в резервных системах. Парольная защита закрытых ключей не помогает на промышленном сервере, потому что злоумышленник может получить ключи из процесса памяти. Есть аппаратные устройства, которые могут защитить секретные ключи даже в случае компрометации сервера, но они стоят дорого и, таким образом, оправданы только в организациях с высокими требованиями безопасности.

- После компрометации отзывайте старые сертификаты и генерируйте новые ключи.

- Обновляйте сертификаты каждый год и всегда с новыми закрытыми ключами.

- Обеспечьте охват всех используемых доменных имен

Убедитесь, что ваши сертификаты охватывают все доменные имена, которые вы хотите использовать на сайте. Например, у вас есть главный домен `www.example.com`, но вы также используете домен

www.example.net. Ваша цель — избежать предупреждения о недействительности сертификата, которое будет путать ваших пользователей и ослаблять их доверие.

Даже тогда, когда на сервере настроено только одно доменное имя, нужно иметь в виду, что вы не можете контролировать, как пользователи приходят к вам на сайт или какие ссылки на него указывают. В большинстве случаев, вы должны убедиться, что сертификат работает с и без WWW (например, как для example.com и www.example.com). Безопасный веб-сервер должен иметь сертификат, действительный для каждого настроенного доменного имени. Сертификаты на весь домен (Wildcard) имеют свое преимущество, но следует избегать их, если их использование означает предоставление закрытого ключа большой группе людей, например, системным администраторам разных организаций. Кроме того, имейте в виду, что Wildcard сертификаты могут быть использованы злоумышленниками для передачи уязвимости от одного веб-сайта на все другие сайты, которые используют один и тот же сертификат.

- Приобретайте сертификаты у надежного удостоверяющего центра. Выбирайте надежный удостоверяющий центр (CA), который заботится о своем бизнесе и безопасности. Рассмотрим следующие критерии при выборе CA:

#### Отношение к безопасности

Все CA проходят регулярный аудит (иначе они не имели бы право работать как CA), но некоторые из них более серьезно относятся к безопасности, чем другие. Выяснить, какие из них лучше в этом отношении нелегко, но один способ заключается в изучении их истории инцидентов безопасности и выявлении того, как они реагировали на компрометации и инциденты безопасности и учились ли они на своих ошибках. Основное направление деятельности

CA, у которых выпуск сертификатов является основным направлением деятельности, потеряют бизнес, если они сделают что-то ужасно неправильно, и они, вероятно, не будут пренебрегать разделением сертификатов, преследуя потенциально более прибыльные возможности в других местах.

#### Предлагаемые услуги

Как минимум, выбранный CA должен обеспечивать поддержку списка отозванных сертификатов (CRL) и протокола OCSP.

#### Инструменты управления сертификатами

Если вам нужно большое количество сертификатов, то выберите центр сертификации, который даст вам хорошие инструменты для управления ими.

## Поддержка

Выберите центр сертификации, который предоставляет хорошую поддержку, когда это необходимо.

- Используйте надежные алгоритмы подписи сертификата

Безопасность сертификата зависит от длины закрытого ключа и прочности используемой функции хеширования. Сегодня большинство сертификатов используют алгоритм SHA1, который считается слабым.

Вам нужно немедленно заменить все ваши сертификаты, использующие алгоритм SHA1, если они истекают после 2015 года.

- Используйте безопасные протоколы

Существует пять версий протоколов в SSL/TLS семейства: SSL v2, SSL v3, TLS v1.0, TLS v1.1 и TLS v1.2. Из них: • SSL v2 является небезопасным и не должен быть использован. • SSL v3 является небезопасным при использовании с HTTP и слабым при использовании с другими протоколами. Эта версия также устарела, поэтому она не должна использоваться. • TLS v1.0 до сих пор является безопасным протоколом. При использовании с HTTP этот протокол обеспечивает безопасность, но только при тщательной конфигурации. • TLS v1.1 и v1.2 не имеют известных проблем безопасности.

TLS v1.2 должен быть вашим основным протоколом. Эта версия лучше, потому что она поддерживает важные функции, которые недоступны в более ранних версиях. Если ваш сервер (или любое промежуточное устройство) не поддерживает TLS v1.2, то планируйте его модернизацию в ускоренном режиме. Если ваши поставщики услуг не поддерживают TLS v1.2, требуйте, чтобы они модернизировали свою систему.

Для поддержки более старых клиентов вы должны продолжать поддерживать TLS v1.0 и TLS v1.1 еще некоторое время. С некоторыми обходными путями эти протоколы еще можно считать достаточно безопасными для большинства веб-сайтов.

- Используйте безопасные алгоритмы шифрования

Для безопасного обмена данными вы должны сначала убедиться, что вы общаетесь непосредственно с нужным абонентом (и не через кого-то, кто будет подслушивать). В SSL и TLS алгоритмы шифрования используются для определения, насколько безопасно происходит обмен данными. Они состоят из различных строительных блоков. Если в одном из строительных блоков наблюдается слабая безопасность, то вы должны быть в состоянии переключиться

на другой. Ваша цель — использовать только те алгоритмы шифрования, которые обеспечивают аутентификацию и шифрование в 128 бит или более. Всего остального следует избегать: • Наборы со слабыми алгоритмами шифрования (как правило, от 40 до 56 бит) могут быть легко взломаны • RC4 также сейчас считается слабым. Вы должны убрать поддержку этого алгоритма как можно раньше, но только после проверки потенциального негативного воздействия на совместимость. • 3DES обеспечивает около 112 бит безопасности. Это ниже рекомендованного минимума 128 бит, но это все еще достаточно сильный алгоритм. Большая практическая проблема в том, что 3DES гораздо медленнее, чем альтернативные варианты. Таким образом, мы не рекомендуем его для повышения производительности.

- Контроль за выбором алгоритма шифрования

В SSL версии 3 и более поздних версиях протокола, клиенты отправляют список алгоритмов шифрования, которые они поддерживают, и сервер выбирает один из них для организации безопасного канала связи. Не все сервера могут делать это хорошо, так как некоторые выбирают первый поддерживаемый алгоритм из списка. Таким образом, выбор правильного алгоритма шифрования является критически важным для безопасности.

- Поддержка Forward Secrecy.

Forward Secrecy — это особенность протокола, который обеспечивает безопасный обмен данными, он не зависит от закрытого ключа сервера. С алгоритмами шифрования, которые не поддерживают Forward Secrecy, возможно расшифровать ранее зашифрованные разговоры с помощью закрытого ключа сервера. Нужно поддерживать и предпочитать ECDHE (аббревиатура ECDHE расшифровывается как «эфемерный алгоритм Диффи-Хеллмана с использованием эллиптических кривых») алгоритмы шифрования. Для поддержки более широкого круга клиентов, вы должны также использовать DHE, как запасной вариант после ECDHE.

- Отключите Renegotiation по инициативе клиента

В SSL / TLS renegotiation позволяет сторонам остановить обмен данными, с тем чтобы повторно инициировать его для обеспечения безопасности. Есть некоторые случаи, в которых renegotiation должен быть инициирован сервером, но нет никакой известной необходимости позволять инициировать renegotiation клиентом. Кроме того это может облегчить организацию DDoS-атаки на ваши сервера.

- Отключите TLS compression

В 2012 году CRIME attack показал, как TLS сжатие может быть использовано злоумышленниками для выявления деталей конфиденциальных данных (например, сессионные куки). Очень немногие клиенты поддерживали TLS сжатие тогда (и в настоящее время), так что маловероятно, что вы будете испытывать какие-либо проблемы с производительностью после отключения TLS сжатия на серверах.

- Отключите RC4

Алгоритм RC4 является небезопасным и должен быть отключен. В настоящее время мы знаем, что для взлома RC4 требуются миллионы запросов, много пропускной способности и времени. Таким образом, риск все еще относительно невелик, но вполне возможно, что атаки будут масштабнее в будущем. Перед снятием RC4 проверьте, будут ли ваши существующие пользователи затронуты; другими словами, проверить, если у вас есть клиенты, которые поддерживают только RC4.

- Будьте в курсе атаки BEAST

Успешная атака BEAST похожа на взлом сессии. К сожалению, для смягчения угрозы со стороны сервера требуется RC4, который больше не рекомендуется. Из-за этого, а также из-за того что атака BEAST теперь в значительной степени уменьшается на стороне клиента, мы больше не рекомендуем смягчения на сервере путем использования RC4. В некоторых ситуациях, когда есть большое количество старых клиентов, уязвимых для атаки BEAST, более безопасно использовать RC4 с TLS 1.0 и более ранние версии протокола. Принимать это решение следует осторожно и только после полного понимания окружающей среды и модели ее угроз.

- Отключить SSL v3

SSL v3 уязвим против POODLE атаки, которая была обнаружена в октябре 2014. Лучший способ устранения уязвимости POODLE атаки — это отключить SSL v3, который в большинстве сайтов можно сделать безопасно.

## **2.2 Изучить основные уязвимости и атаки на SSL последнего времени – POODLE, HeartBleed**

- POODLE

Специалист по безопасности Бодо Мёллер (Bodo Möller) с коллегами из компании Google опубликовал подробности об уязвимо-

сти в дизайне протокола SSL 3.0. Уязвимость под кодовым названием POODLE («пудель», Padding Oracle On Downgraded Legacy Encryption, CVE-2014-3566) позволяет расшифровать содержимое защищённого канала коммуникации. В общем, на всех системах нужно срочно блокировать использование SSL 3.0, потому что работающего способа обойти эксплоит не существует.

SSL 3.0 (RFC6101), использующий шифр RC4, устарел примерно на 15 лет. На замену ему создали TLS 1.0, TLS 1.1 и TLS 1.2, но он до сих пор широко используется в браузерах, веб-серверах и т.д. И многие реализации TLS обратно совместимы с SSL 3.0.

Злоумышленник может умышленно принудить клиента подключиться именно по SSL 3.0, эмулируя разрывы связи, и после этого эксплуатировать уязвимость.

Google пишет, что для защиты достаточно отключить поддержку SSL 3.0 или шифрования в режиме сцепления блоков (CBC mode). Однако, в этом случае возникнут серьёзные проблемы с совместимостью. Поэтому рекомендуемый способ обхода — поддержка механизма TLS\_FALLBACK\_SCSV, который не позволяет злоумышленнику снизить защиту канала до SSL 3.0. Механизм также предотвращает снижение защиты с TLS 1.2 до 1.1 или 1.0, что может помочь в предотвращении будущих атак.

Браузер Chrome и серверы Google поддерживают TLS\_FALLBACK\_SCSV с февраля, так что есть достаточно доказательств, что метод действительно эффективен.

Дополнительно, для Google Chrome также представили патч, который не позволяет соединяться по SSL 3.0. Аналогичная опция появится в Firefox 34, который ожидается к выходу 25 ноября: там поддержку SSL 3.0 отключат по умолчанию.

В SSLv3 обнаружена возможность Padding Oracle атаки, которая позволяет злоумышленнику, имеющему какую-либо возможность отправлять свои данные на сервер по SSLv3 от имени жертвы, расшифровывать по 1 байту за 256 запросов. Происходит это из-за того, что в SSLv3 padding не учитывается в MAC.

Теоретически, реализовать атаку можно на любой сервис, где есть возможность влиять на отправляемые данные со стороны атакующего. Проще всего это реализовать, например, если злоумышленнику необходимо получить Cookie на HTTPS-странице, добавляя свой код на HTTP-страницы, который делает подконтрольные запросы на HTTPS-страницы, и подменяя зашифрованные блоки.

Главной проблемой RC4 является наличие смещений: чем больше соединений и потоков шифрования используется для отправки

одних и тех же данных (например, пароля или HTTP-куки), тем больше можно извлечь из трафика информации, которая помогает осуществить дешифрование. Ниже будет показано, как можно совместить эффективную атаку на CBC-шифрование при использовании SSL 3.0 (при условии, что злоумышленник может модифицировать сетевой обмен между клиентом и сервером). При этом, в отличие от уязвимостей BEAST и Lucky 13, здесь нет каких-то обходных решений. У нас есть только небезопасный протокол SSL 3.0, и чтобы обеспечить надёжное шифрование, нужно избегать его использования.

Самая серьёзная проблема CBC-шифрования в SSL 3.0 заключается в том, что дополнение блоков (паддинг) может быть произвольным (за исключением последнего байта), на него не распространяется MAC (Message Authentication Code). Целостность дополнения не может быть полностью подтверждена в ходе дешифрования, поскольку в SSL 3.0 сообщение сначала подписывается с помощью MAC, затем дополняется паддингом, и уже после — шифруется блочным шифром. Паддинг от 1 до L байт (где L — размер блока в байтах) используется для получения целого числа блоков перед шифрованием. Легче всего пробить защиту, если есть целый блок паддинга, который (до шифрования) состоит из L-1 произвольных байт, за которыми следует одиночный байт со значением L-1.

- HeartBleed

Ошибка (переполнение буфера) в криптографическом программном обеспечении OpenSSL, позволяющая несанкционированно читать память на сервере или на клиенте, в том числе для извлечения закрытого ключа сервера. Информация об уязвимости была опубликована в апреле 2014 года, ошибка существовала с конца 2011 года.

Heartbeat-пакет состоит из данных, которые сервер должен вернуть в неизменном виде (это гарантирует, что сервер расшифровывал пакет), и случайных заполняющих байтов. OpenSSL не проверял корректность этого пакета: возможен, например, пакет длиной 16 байт, в котором написано, что длина данных 64 килобайта (поле размера двухбайтовое). Подверженные ошибке версии OpenSSL выходили за пределы буфера и передавали клиенту столько памяти, сколько он запросил, позволяя атакующему получать не предназначенные для этого данные. RFC предписывает не отвечать на такие «отравленные» пакеты.

## 2.3 Обзор домена из списка Recent Best

Выбранный домен: `encoredentalplan.com` (198.39.202.30)



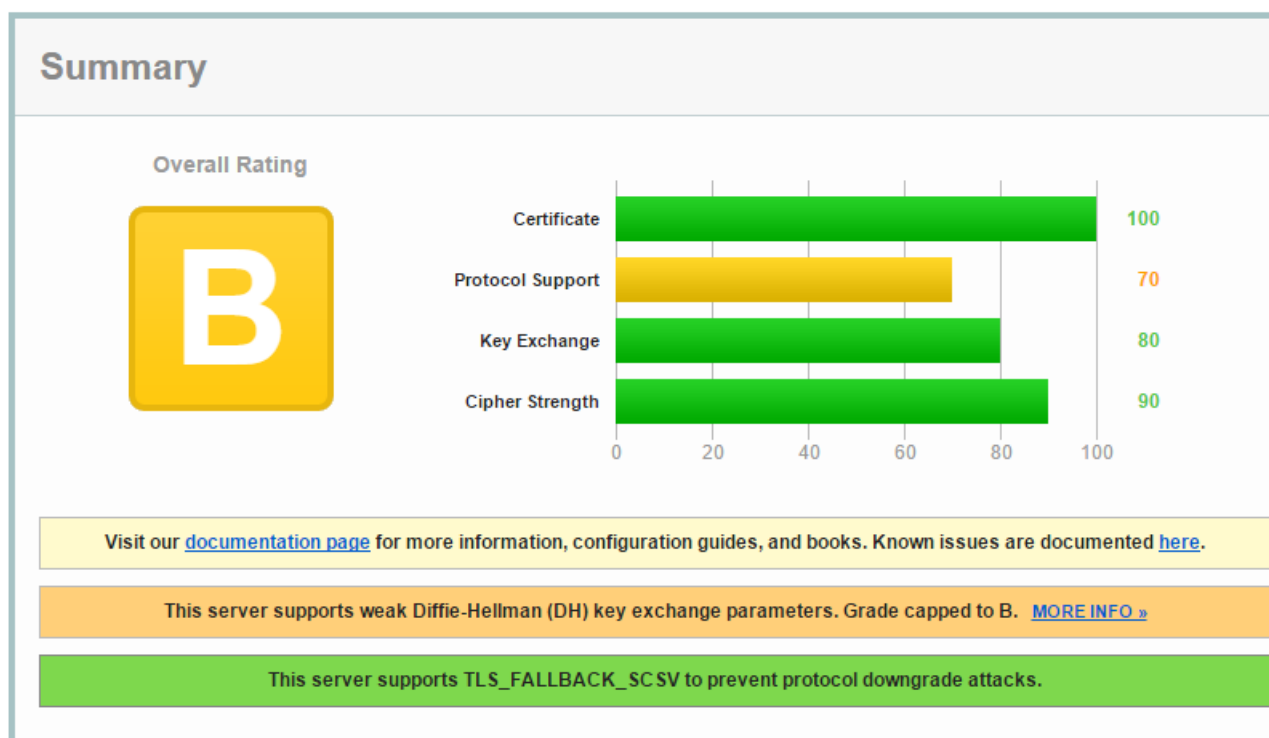


Рис. 1: Summary домена из списка Recent Best

Общий рейтинг обозначается буквами A-F. В данном случае оценка домену выставлена B. В разделе summary можно увидеть выставленные оценки по 4 параметрам: Сертификат, Поддержка протокола, Обмен ключами, Стойкость шифра. (см. рисунок 1)

В пояснительной информации ниже указано почему оценка была снижена до B - слабые параметры протокола Диффи — Хеллмана. Значит может быть проведена атака Logjam Attack.

Так же указано, что сервер поддерживает TLS\_FALLBACK\_SCSV, который не позволяет злоумышленнику снизить защиту канала до SSL 3.0. Механизм также предотвращает снижение защиты с TLS 1.2 до 1.1 или 1.0, что может помочь в предотвращении будущих атак.

Из прочей информации можно отметить следующее(см. рисунок 2):

- Присутствуют слабые наборы шифров.
- На некоторых платформах обнаружены несоответствия

## 2.4 Обзор домена из списка Recent Worst

Выбранный домен: receiver.tvc.org (67.51.200.102)



	<b>Cipher Suites (SSL 3+ suites in server-preferred order; deprecated and SSL 2 suites always at the end)</b>			
	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f)	DH 1024 bits (p: 128, g: 1, Ys: 128)	FS <b>WEAK</b>	256
	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e)	DH 1024 bits (p: 128, g: 1, Ys: 128)	FS <b>WEAK</b>	128
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39)	DH 1024 bits (p: 128, g: 1, Ys: 128)	FS <b>WEAK</b>	256
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33)	DH 1024 bits (p: 128, g: 1, Ys: 128)	FS <b>WEAK</b>	128
	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x16)	DH 1024 bits (p: 128, g: 1, Ys: 128)	FS <b>WEAK</b>	112
	<b>Handshake Simulation</b>			
	<a href="#">Android 2.3.7</a>	No SNI <sup>2</sup>	Protocol or cipher suite mismatch	Fail <sup>3</sup>
	<a href="#">Android 4.0.4</a>		Protocol or cipher suite mismatch	Fail <sup>3</sup>
	<a href="#">Android 4.1.1</a>		Protocol or cipher suite mismatch	Fail <sup>3</sup>
	<a href="#">Android 4.2.2</a>		Protocol or cipher suite mismatch	Fail <sup>3</sup>
	<a href="#">Android 4.3</a>		Protocol or cipher suite mismatch	Fail <sup>3</sup>

Рис. 2: Прочая информация домена из списка Recent Best

Аналогично предыдущему примеру можно увидеть оценки по 4м параметрам. (см. рисунок 3)

Из пояснительной информации:

- The server does not support Forward Secrecy with the reference browsers.

Домен не поддерживает прямую секретность, следовательно не гарантирует, что сессионные ключи, полученные при помощи набора ключей долговременного пользования, не будут скомпрометированы при компрометации одного из долговременных ключей. С алгоритмами шифрования, которые не поддерживают Forward Secrecy, возможно расшифровать ранее зашифрованные разговоры с помощью закрытого ключа сервера. Нужно поддерживать и предпочитать ECDHE (аббревиатура ECDHE расшифровывается как «эфемерный алгоритм Диффи-Хеллмана с использованием эллиптических кривых») алгоритмы шифрования. Для поддержки более широкого круга клиентов, необходимо также использовать DHE, как запасной вариант после ECDHE.

- This server uses RC4 with modern browsers. Grade capped to C.

Алгоритм RC4 является небезопасным и должен быть отключен. В настоящее время известно, что для взлома RC4 требуются миллионы запросов, много пропускной способности и времени. Таким образом, риск все еще относительно невелик, но вполне возможно, что атаки будут масштабнее в будущем. Если снимать RC4 нужно заранее проверить, будут ли существующие пользователи за-

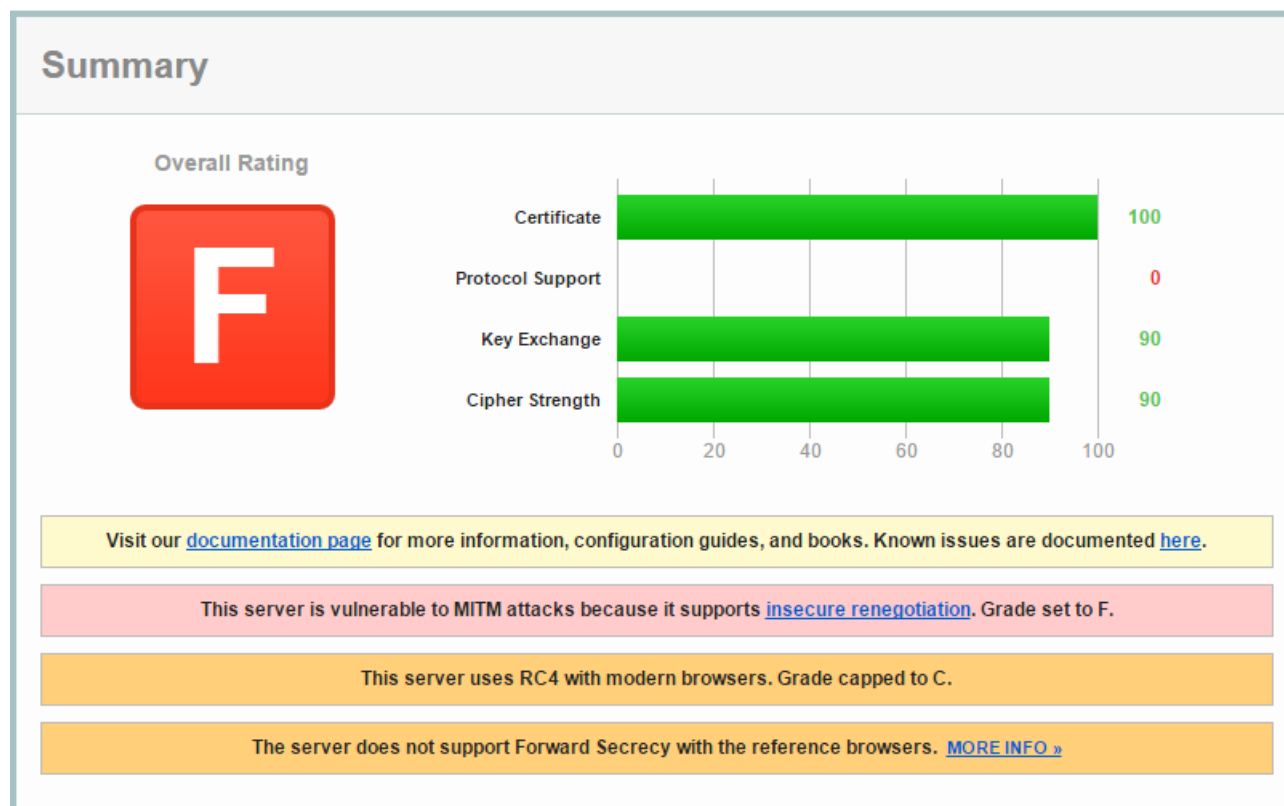


Рис. 3: Summary домена из списка Recent Worst

тронуты; другими словами, проверить, есть ли клиенты, которые поддерживают только RC4.

- This server is vulnerable to MITM attacks because it supports insecure renegotiation. Grade set to F.

Сервер поддерживает небезопасное пересогласование ключей, отчего возможны такие типы атак, как Cross-Site Request Forgery и Cross-Site Scripting.


Прочую информацию можно увидеть на рисунке 4.

## 2.5 Обзор известного сайта по выбору

Выбранный домен: rzd.ru (217.175.140.90)

### 2.5.1 Интерпретировать результаты в разделе Summary

На рисунке 5 можно увидеть результаты и оценки проверки.



Protocol Details	
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	Supported DoS DANGER ( <a href="#">more info</a> )
Insecure Client-Initiated Renegotiation	Supported INSECURE ( <a href="#">more info</a> )
BEAST attack	Mitigated server-side ( <a href="#">more info</a> ) TLS 1.0: 0x4
POODLE (SSLv3)	No, SSL 3 not supported ( <a href="#">more info</a> )
POODLE (TLS)	No ( <a href="#">more info</a> )
Downgrade attack prevention	No, TLS_FALLBACK_SCSV not supported ( <a href="#">more info</a> )
TLS compression	No
RC4	Yes WEAK ( <a href="#">more info</a> )
Heartbeat (extension)	No
Heartbleed (vulnerability)	No ( <a href="#">more info</a> )
OpenSSL CCS vuln. (CVE-2014-0224)	No ( <a href="#">more info</a> )
Forward Secrecy	No WEAK ( <a href="#">more info</a> )

Рис. 4: Прочая информация домена из списка Recent Worst

Из пояснительной информации(описание раньше не встречавшихся замечаний):

- The server supports only older protocols, but not the current best TLS 1.2. Grade capped to C.

На рисунке 6 можно увидеть, что на сервере не поддерживается TLS 1.2 и TLS 1.1, зато поддерживается TLS 1.0, SSL 3, SSL 2. При этом SSL 3 и SSL 2 считаются незащищенными.

- Certificate uses a weak signature. When renewing, ensure you upgrade to SHA2.

Используемый алгоритм подписи на сервере: SHA1. Это можно увидеть в разделе Authentication. (см. рисунок 7)

Т.к. SHA1 считается небезопасным алгоритмом шифрования, предлагается обновить его на более современный и криптостойкий SHA2.

- This server is vulnerable to the POODLE attack. If possible, disable SSL 3 to mitigate. Grade capped to C. This server is vulnerable to the POODLE attack against TLS servers. Patching required. Grade set to F.

В виду того, что используется SSL шифрование на сервер можно осуществить атаку POODLE(см. выше). Так же возможно осуществлять атаку на сервер, где используется TLS(для некоторых реализаций), как в данном случае.

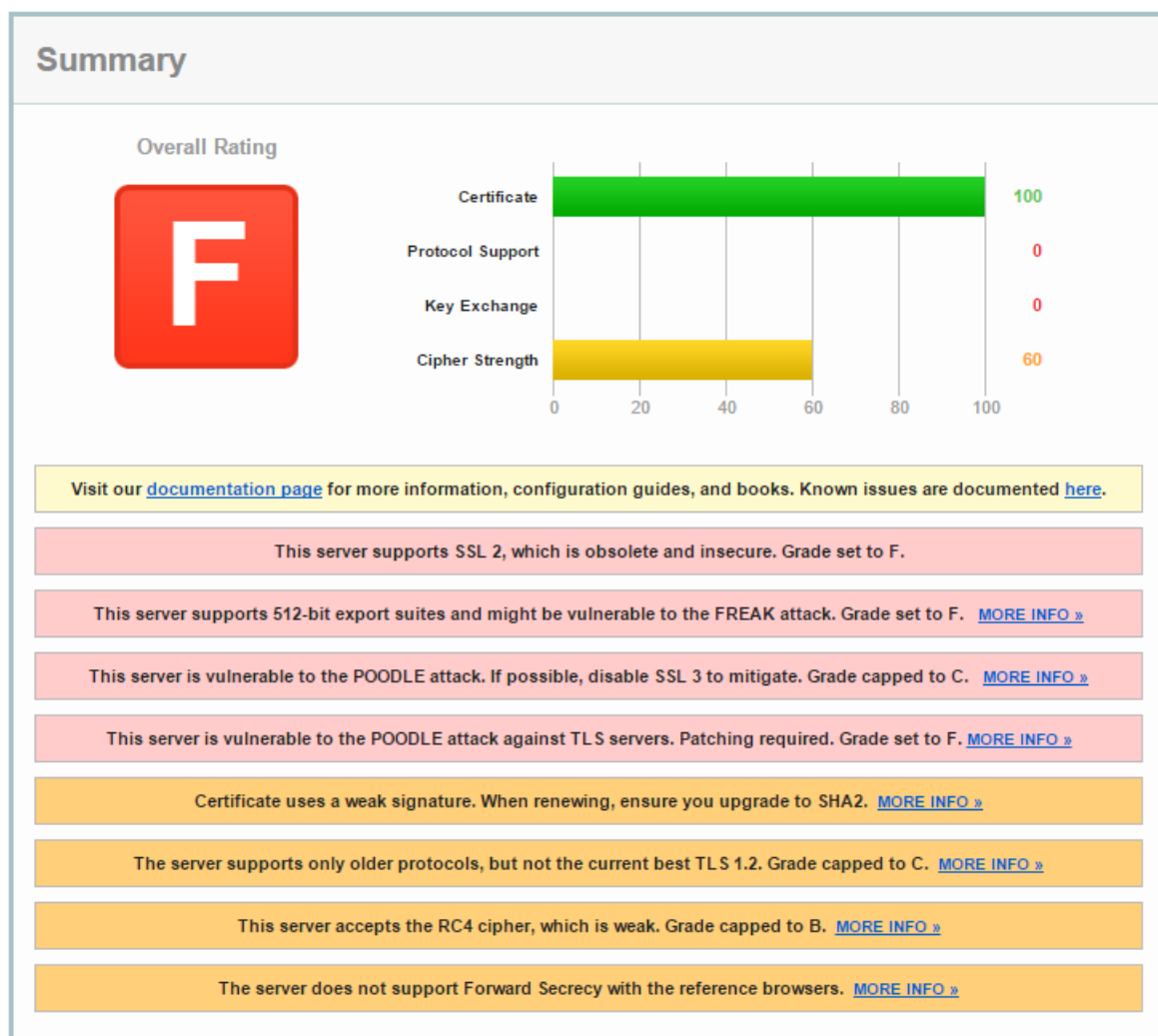


Рис. 5: Summary rzd.ru

	Protocols	
	TLS 1.2	No
	TLS 1.1	No
	TLS 1.0	Yes
	SSL 3 INSECURE	Yes
	SSL 2 INSECURE	Yes

Рис. 6: Protocols rzd.ru

- This server supports 512-bit export suites and might be vulnerable to the FREAK attack. Grade set to F.

Рис. 7: Алгоритм подписи rzd.ru

Возможность FREAK атаки. название уязвимости «атака FREAK» происходит от фразы «Factoring attack on RSA-EXPORT Keys», означающей способ подбора открытых ключей к «экспортному» шифрованию RSA. Суть уязвимости заключается в том, что злоумышленники могут заставить браузеры использовать более слабое шифрование, чем принято обычно. Тогда они смогут взломать его за считанные часы, получив не только доступ к чужим личным данным, но и возможность управлять содержимым страниц в браузере вплоть до кнопки лайка Фейсбука.

Причина появления FREAK лежит в старом требовании властей США, принятом ещё в 1990-е годы. После внедрения шифрования в браузер от Netscape государство требовало от технологических компаний оставлять в своих алгоритмах шифрования «лазейку» для спецслужб при экспортировании своих продуктов за рубеж.

Агентства вроде АНБ и ФБР опасались, что не смогут расшифровать слишком стойкий шифр, если понадобится вести слежку за пользователями в других странах. Несмотря на то, что требование не применять сильную криптозащиту в «экспортных» продуктах было снято в конце 1990-х, более слабое шифрование было интегрировано в множество программ и оставалось незамеченным публикой до недавнего времени.

«Экспортное» шифрование использовало ключи безопасности длиной в 512 бит, а не 1024, как было принято обычно. По данным исследователей, такой ключ можно было подобрать в течение семи часов при помощи мощности 75 обычных компьютеров или аренды аналогичной мощности за 100 долларов в «облачном» сервисе вроде Amazon Web Services. Демонстрацию подбора 512-битного ключа впервые публично провели в 1999 году.


- This server supports SSL 2, which is obsolete and insecure. Grade set to F.

Поддержка SSL2, который считается устаревшим и не безопасным.

### 2.5.2 Расшифровать все аббревиатуры шифров в разделе Configuration

На рисунке 8 приведены используемые шифры. Ниже расшифровка аббревиатур:

- TLS (англ. Transport Layer Security — безопасность транспортного уровня), как и его предшественник SSL (англ. Secure Sockets Layer



**Cipher Suites (SSL 3+ suites in server-preferred order; deprecated and SSL 2 suites always at the end)**

TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_RC4_128_SHA (0x5) <b>WEAK</b>	128
TLS_RSA_WITH_RC4_128_MD5 (0x4) <b>WEAK</b>	128
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	112
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x62) <b>WEAK</b>	56
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x64) <b>WEAK</b>	56
TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x3) <b>INSECURE</b>	40
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x6) <b>INSECURE</b>	40
TLS_RSA_WITH_DES_CBC_SHA (0x9) <b>WEAK</b>	56
SSL CK_DES_192_EDE3_CBC_WITH_MD5 (0x700c0) <b>INSECURE</b>	112

Рис. 8: Ипользуемые шифры rzd.ru

— уровень защищённых сокетов) — криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет[1]. TLS и SSL используют асимметричную криптографию для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

- SSL (англ. secure sockets layer — уровень защищённых сокетов) — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.
- RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и других.
- Advanced Encryption Standard (AES), также известный как Rijndael — симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES.
- Режим сцепления блоков шифротекста (англ. Cipher Block Chaining, CBC) — один из режимов шифрования для симметричного блоч-

ного шифра с использованием механизма обратной связи. Каждый блок открытого текста (кроме первого) побитово складывается по модулю 2 (операция XOR) с предыдущим результатом шифрования.


- Secure Hash Algorithm 1 — алгоритм криптографического хеширования. Описан в RFC 3174. Для входного сообщения произвольной длины (максимум  $2^{64} - 1$  бит, что примерно равно 2 эксабайта) алгоритм генерирует 160-битное хеш-значение, называемое также дайджестом сообщения. Используется во многих криптографических приложениях и протоколах.
- RC4 (англ. Rivest cipher 4 или англ. Ron's code, также известен как ARCFOUR или ARC4 (англ. alleged RC4)) — потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и TLS, алгоритмах обеспечения безопасности беспроводных сетей WEP и WPA).
- MD5 (англ. Message Digest 5) — 128-битный алгоритм хеширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института (Massachusetts Institute of Technology, MIT) в 1991 году. Предназначен для создания «отпечатков» или дайджестов сообщения произвольной длины и последующей проверки их подлинности.
- Triple DES (3DES) — симметричный блочный шифр, созданный Уитфилдом Диффи, Мартином Хеллманом и Уолтом Тачманном в 1978 году на основе алгоритма DES, с целью устранения главного недостатка последнего — малой длины ключа (56 бит), который может быть взломан методом полного перебора ключа.

### 2.5.3 Прокомментировать большинство позиций в разделе Protocol Details

Прокомментируем позиции в разделе Protocol Details. (см. рисунок 9)

- Secure Renegotiation - Supported.  
Сервер поддерживает безопасное пересогласование ключей.
- POODLE (SSLv3) Vulnerable INSECURE (more info), POODLE (TLS) Vulnerable INSECURE (more info), Downgrade attack prevention No, TLS\_FALLBACK\_SCSV not supported (more info)  
Существует поддержка SSL3, и при этом не установлен механизм TLS\_FALLBACK\_SCSV. Таким образом клиент может соединить-





Protocol Details	
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Not mitigated server-side ( <a href="#">more info</a> ) SSL 3: 0x35, TLS 1.0: 0x35
POODLE (SSLv3)	Vulnerable INSECURE ( <a href="#">more info</a> )
POODLE (TLS)	Vulnerable INSECURE ( <a href="#">more info</a> )
Downgrade attack prevention	No, TLS_FALLBACK_SCSV not supported ( <a href="#">more info</a> )
TLS compression	No
RC4	Yes WEAK ( <a href="#">more info</a> )
Heartbeat (extension)	No
Heartbleed (vulnerability)	No ( <a href="#">more info</a> )
Open SSL CCS vuln. (CVE-2014-0224)	No ( <a href="#">more info</a> )
Forward Secrecy	No WEAK ( <a href="#">more info</a> )
Next Protocol Negotiation (NPN)	No
Session resumption (caching)	No (IDs assigned but not accepted)

Рис. 9: Описани протокола rzd.ru

ся с сервером используя SSL3, в этом случае может быть использована атака POODLE.

Прочие моменты комментировались выше.

#### 2.5.4 Сделать итоговый вывод о реализации SSL на заданном домене

Как видно из оценки и прочих данных реализация SSL на сайте rzd.ru не очень хороша. Существует ряд уязвимостей и атак, которыми могут воспользоваться злоумышленники.

### 3 Вывод

В результате выполнения работы были изучены лучшие практики по развертыванию SSL, возможные уязвимости. Получен опыт использования инструмента тестирования SSL на сервере (SSL Server Test). Помимо этого в ходе обнаружения недостатков серверов были изучены какие типы атак могут быть совершены на них, причины этого и каким образом можно предотвращать эти атаки.