Отчет по лабораторной работе 4: SSL WebGoat

Семён Мартынов
<semen.martynov@gmail.com>

5 іюня 2015 г.

Содержание

1 Тестирование корректности настройки SSL **2**
1.1 Цель работы . . . 2
1.2 Ход работы . . . 2
1.2.1 Интерпретировать результаты в разделе Summary . . . 3
1.2.2 Расшифровать все аббревиатуры шифров в разделе Configuration . . 4
1.2.3 Прокомментировать большинство позиций в разделе Protocol Details 5
1.3 Выводы . . . 6

2 Проект OWASP WebGoat **7**
2.1 Цель работы . . . 7
2.2 Ход работы . . . 7
2.2.1 Недостатки контроля доступа . . . 7
2.2.2 Безопасность AJAX . . . 7
2.2.3 Недостатки аутентификации . . . 7
2.2.4 Переполнение буфера . . . 7
2.2.5 Качество кода . . . 8
2.2.6 Многопоточность . . . 8
2.2.7 Межсайтовое выполнение сценариев . . . 8
2.2.8 Неправильная обработка ошибок . . . 8
2.2.9 Недостатки приводящие к осуществлению инъекций (SQL и прочее) . 8
2.2.10 Отказ в обслуживании . . . 8
2.2.11 Небезопасное сетевое взаимодействие . . . 8
2.2.12 Небезопасная конфигурация . . . 9

2.2.13 Небезопасное хранилище 9
2.2.14 Исполнение злонамеренного кода 9
2.2.15 Подделка параметров 9
2.2.16 Недостатки управление сессией 9
2.2.17 Безопасность веб-сервисов 9
2.3 Выводы 9

1 Тестирование корректности настройки SSL

1.1 Цель работы

Изучить сервис тестирования корректности настройки SSL на сервере Qualys SSL Labs – SSL Server Test

1.2 Ход работы

Для изучения, рассмотрим сайт кафедры (kspt.icc.spbstu.ru) и его сертификат.

Первое, что обращает на себя внимание - это самоподписанный сертификат. И сервер выдаёт стандартную страницу Apache 2. Звёздочка на месте favicon даёт основание полагать, что сервер работает на Mandriva Linux (компания Mandriva на днях объявила о своём закрытии). Заголовок HTTP подтверждает эти догадки – Apache/2.2.22 (Mandriva Linux/PREFORK-0.1mdv2010.2)

Дальнейшее изучение сертификата (см. рис. 2) показало неправильно выставленное значение поля CN и давно истекший срок валидности. В качестве алгоритма шифрования используется SHA-1, который браузер помечает как слабый, и предупреждает о прекращении поддержки таких сертификатов с 2017-го года.

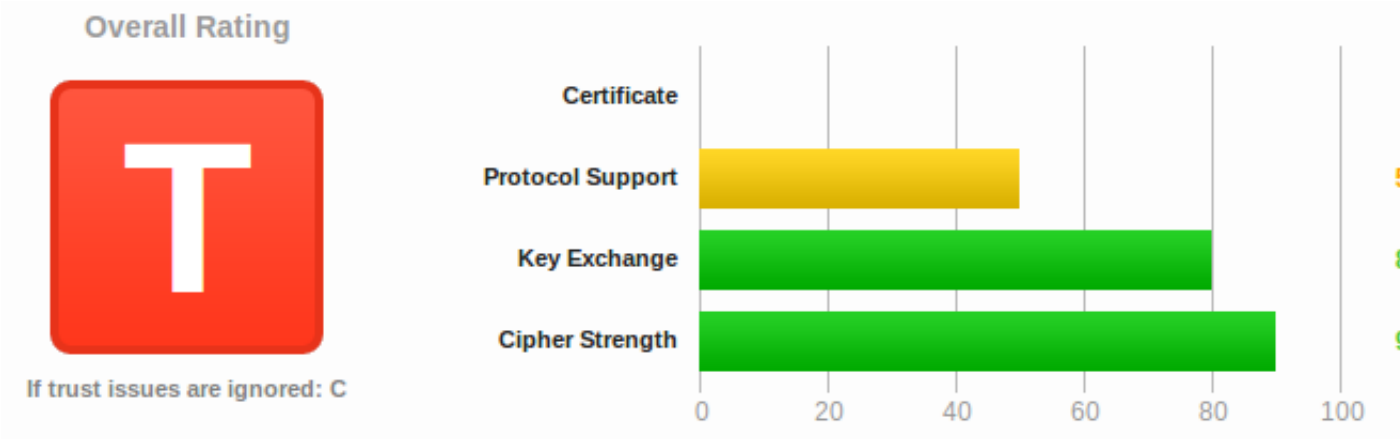

Рис. 1: Анализ сертификата

На ошибку в CN обращает внимание и ssllabs.com. Но если проигнорировать все проблемы, связанные с тем, что это самоподписанный сертификат, результат достаточно не плохой – ssllabs.com присвоил сертификату рейтинг C (см. рис. 1))

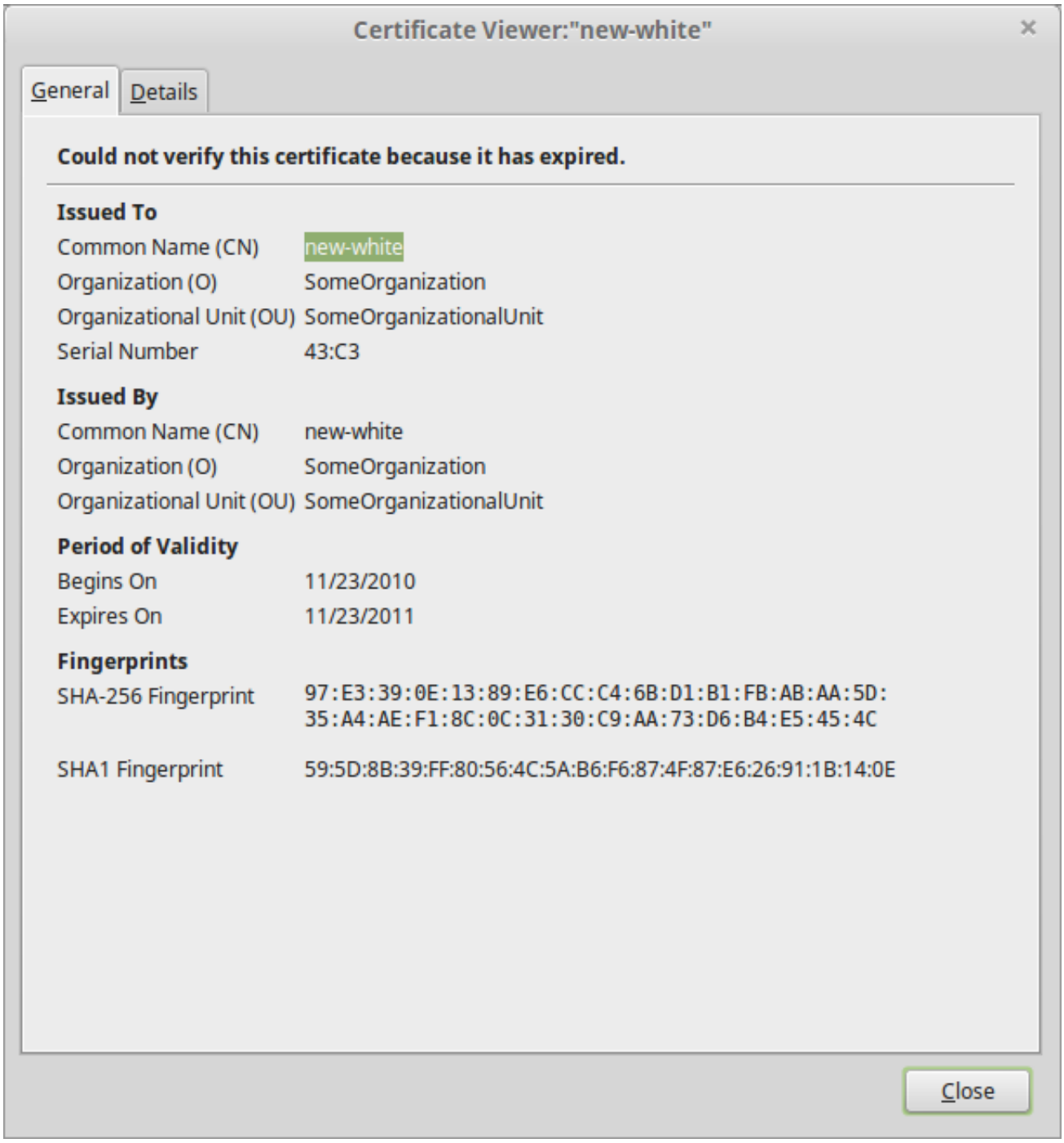

Рис. 2: Сертификат kspt.icc.spbstu.ru

1.2.1 Интерпретировать результаты в разделе Summary

1.2.1.1 This server supports weak Diffie-Hellman (DH) key exchange parameters.

Это показатель уязвимости под названием Logjam (обнаружена в различных реализациях протокола TLS). Уязвимость относится к типу *downgrade* и позволяет клиенту понизить стойкость шифрования до 512 бит DH при условии поддержки сервером шифра DHE_EXPORT, который задается в процессе рукопожатия (*handshake*) между клиентом

и сервером. Т. о. атакующему значительно проще организовать атаку типа Man-in-the-Middle (MitM). Уязвимость затрагивает как серверное ПО, использующее OpenSSL, и веб-браузеры.

1.2.1.2 This server is vulnerable to the POODLE attack. Знаменитая POODLE атака, вогнавшая последний гвоздь в SSL3. По причине того, что в SSLv3 padding не учитывается в MAC, злоумышленник может расшифровывать по 1 байту за 256 запросов, отправляя свои данные на сервер по SSLv3 от имени жертвы. Лечится полным отключением SSL3.

1.2.1.3 Certificate uses a weak signature. When renewing, ensure you upgrade to SHA2. Как говорилось выше, поддержка SHA1 постепенно завершается. Здесь содержится рекомендация использовать SHA2 при следующем обновлении сертификата.

1.2.1.4 The server supports only older protocols, but not the current best TLS 1.2. Наиболее актуальной версией TLS на данный момент является версия 1.2, которая не поддерживается сервером.

1.2.1.5 The server private key is not strong enough. В 2010-м году была представлена аккуратная реализация атаки на RSA1024 на основе ошибок железа.

1.2.1.6 This server accepts the RC4 cipher, which is weak. В 2013-м году была представлена техника взлома Transport Layer Security и Secure Sockets Layer, если в них используется шифр RC4. Уязвимость RC4 связана с недостаточной случайностью потока битов, которым скремблируется сообщение. Прогоняя через него одно сообщение много раз, удалось выявить достаточное количество повторяющихся паттернов для восстановления исходного сообщения.

1.2.1.7 The server does not support Forward Secrecy with the reference browsers. Сервер должен обеспечить гарантии того, что сессионные ключи, полученные при помощи набора ключей долговременного пользования, не будут скомпрометированы при компрометации одного из долговременных ключей.

1.2.2 Расшифровать все аббревиатуры шифров в разделе Configuration

1.2.2.1 TLS – Transport Layer Security.

1.2.2.2 SSL – Secure Sockets Layer.

1.2.2.3 RSA – аббревиатура от фамилий Rivest, Shamir и Adleman.

1.2.2.4 RC4 – **R**ivest **c**ipher 4 или **R**on's **c**ode 4.

1.2.2.5 SHA – Secure Hash Algorithm.

1.2.2.6 AES – Advanced Encryption Standard.

1.2.2.7 CBC – Cipher Block Chaining.

1.2.2.8 3DES – Triple Data Encryption Standard.

1.2.2.9 SNI – Server Name Indication

1.2.2.10 NPN – Next Protocol Negotiation.

1.2.2.11 HSTS – HTTP Strict Transport Security.

1.2.2.12 HPKP – HTTP Public Key Pinning.

1.2.2.13 HTTP – HyperText Transfer Protocol.

1.2.3 Прокомментировать большинство позиций в разделе Protocol Details

1.2.3.1 Secure Renegotiation – возобновление подключения TLS.

1.2.3.2 BEAST attack – атака утилитой BEAST (Browser Exploit Against SSL/TLS).

1.2.3.3 POODLE – уязвимость, позволяющая расшифровать содержимое защищённого канала коммуникации.

1.2.3.4 Downgrade attack – атака, при которой пользователя вынуждают использовать менее безопасные протоколы, которые всё ещё поддерживаются из соображений совместимости.

1.2.3.5 TLS compression – В 2012 году CRIME attack показал, как TLS сжатие может быть использовано злоумышленниками для выявления деталей конфиденциальных данных (например, сессионные куки).

1.2.3.6 Heartbleed – ошибка в OpenSSL, позволяющая несанкционированно читать память на сервере до 64 килобайт за один запрос. Атаку можно производить бесконечное количество раз.

1.2.3.7 Forward Secrecy – 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.

1.2.3.8 Next Protocol Negotiation – клиент сообщает серверу по каким протоколам он бы хотел общаться и сервер может ответить наиболее предпочтительным из тех, которые он знает.

1.2.3.9 Strict Transport Security – механизм, активирующий форсированное защищённое соединение по HTTPS. Данная политика безопасности позволяет сразу же устанавливать безопасное соединение, вместо использования HTTP. Механизм использует особый заголовок HTTP Strict-Transport-Security, для переключения пользователя, зашедшего по HTTP, на HTTPS-сервер.

1.3 Выводы

На заданном сервере использовались механизмы генерации, заданные по умолчанию, т.е. системный администратор ничего сам не конфигурировал. Данный сертификат защищает пустую страницу, т.е. реальной опасности нет, но при использовании этого сертификата на продакшен-сервере наибольшую опасность представлял бы пудель.

2 Проект OWASP WebGoat

2.1 Цель работы

Изучить наиболее распространенные веб-уязвимости и познакомиться с OWASP WebGoat.

2.2 Ход работы

Для выполнения этих работ достаточно воспользоваться встроенными средствами Firefox.

2.2.1 Недостатки контроля доступа

В данном разделе демонстрируется группа проблем, связанных с отсутствием достаточных проверок на стороне сервера. Модифицируя поля запроса, можно выполнять операции, доступ к которым пользователя закрыт (по крайней мере нет возможности легального выполнения через интерфейс).

2.2.2 Безопасность AJAX

Снова проблемы фильтрации параметров, получаемых от клиента, которые встречаются в разных местах. В реальной жизни такие ошибки встречаются достаточно редко т.к. на данный момент у разработчиков достаточно автоматизированных систем тестирования для выявления подобных уязвимостей.

2.2.3 Недостатки аутентификации

Раздел демонстрирует очевидный факт: чем сложнее пароль, тем выше (теоретически) время его обнаружения. Фактически для взлома легче использовать уязвимости в архитектуре системы и недостаточную фильтрацию параметров, чем перебор паролей в лоб.

2.2.4 Переполнение буфера

Не правильная обработка сервером больших пакетов может позволить злоумышленнику получить доступ к защищаемым данным, таким как приватные поля запроса.

2.2.5 Качество кода

Иногда разработчики оставляют для себя рабочие комментарии, которые, попав на продакшен, способны сыграть на руку взломщику.

2.2.6 Многопоточность

Отладка многопоточных приложений всегда является сложной задачей. В данном разделе демонстрируется проблема с блокировками и параллельным выполнением двух запросов, которая реально имела место до некоторого времени при обработке подарочных карт в сети кофеен Starbucks.

2.2.7 Межсайтовое выполнение сценариев

Идея атаки во внедрении в выдаваемую веб-системой страницу вредоносного кода. Этот код можно, к примеру, встроить в адрес... Это тоже является примером отсутствия достаточной фильтрации.

2.2.8 Неправильная обработка ошибок

Возникновение ошибки (к примеру, по причине отсутствия пароля) должно трактоваться системой как плохая ситуация (т.е доступ предоставляться не должен), а не наоборот.

2.2.9 Недостатки приводящие к осуществлению инъекций (SQL и прочее)

Недостаточная фильтрация! Параметр из адресной строки подставляется прямо в SQL-запрос, т.о. запрос можно отредактировать прямо из адресной строки.

2.2.10 Отказ в обслуживании

Ресурсы сервера конечны, и можно искусственно добиться их исчерпания, к примеру переполнением дискового накопителя лог-файлами.

2.2.11 Небезопасное сетевое взаимодействие

Передача паролей в открытом виде является плохой практикой.

2.2.12 Небезопасная конфигурация

Часто информацию о точке входа в интерфейс администратора можно найти в документации проекта. Там же может содержаться стандартный пользователь и пароль, который, теоретически, должен быть изменён при конфигурировании.

2.2.13 Небезопасное хранилище

По соли можно определить используемую хэш-функцию.

2.2.14 Исполнение злонамеренного кода

Если имеется возможность заливки своего файла на сервер (при этом опять же имеется не достаточная фильтрация данных), то можно легко залить свой веб-шелл.

2.2.15 Подделка параметров

Снова не достаточная фильтрация параметров. Пакеты можно перехватить и подменить параметры.

2.2.16 Недостатки управление сессией

Зная алгоритм формирования ключа в куки-файле, можно перехватить параметры и сформировать этот файл самостоятельно. Или просто его украсть.

2.2.17 Безопасность веб-сервисов

Снова SQL-инъекции и недостаточная фильтрация.

2.3 Выводы

Многие из представленных проблем хорошо известны и встречаются не часто, т.к. в данный момент имеется достаточно эффективные средства тестирования.