УДК 519.717.7
ББК 22.181
В 58

Власенко А.В.
Кандидат технических наук, доцент кафедры компьютерных технологий и информационной безопасности института информационных технологий и безопасности, начальник управления аспирантуры и докторантуры Кубанского государственного технологического университета, Краснодар, e-mail: Vlasenko@kubstu.ru
Дзьобан П.И.
Аспирант кафедры компьютерных технологий и информационной безопасности института информационных технологий и безопасности Кубанского государственного технологического университета, Краснодар, e-mail: antiemoboy@mail.ru

Анализ уязвимостей и моделирование атак на данные трафика “https”

(Рецензирована)

Аннотация. *В современном информационном мире гарантией конфиденциальной передачи данных в сети является использование протокола “https”. Действительно, в 2017 году 75% Web-ресурсов доменной зоны Российской Федерации и стран СНГ используют шифрование диалога «клиент-сервер». Рассмотрена демонстрация возможных исходов атак злоумышленников для формализации выводов и использования их при построении процедур передачи, обработки и хранения данных в сети.*

Ключевые слова: *атаки, шифрование, конфиденциальность, Web-ресурс, аутентификация, https, SSL/TLS.*

Vlasenko A.V.
Candidate of Technical Sciences, Associate Professor of Computer Technologies and Information Security Department, Institute of Information Technologies and Security, Head of Department of Postgraduate and Doctoral Studies, Kuban State University of Technology, Krasnodar, e-mail: Vlasenko@kubstu.ru
Dzoban P.I.
Post-graduate student of Computer Technologies and Information Security Department, Institute of Information Technologies and Security, Kuban State University of Technology, Krasnodar, e-mail: antiemoboy@mail.ru

Vulnerability analysis and simulation of “https” data traffic attacks

Abstract. *In the modern information world, guaranty of confidential data transmission in the network is the use of the “https” protocol. Indeed, in the year of 2017, 75% of the web resources of the domain zone of the Russian Federation and the CIS countries use encryption of the “client-server” dialog. This paper is aimed at demonstrating possible outcomes of the attacks of malefactors to formalize conclusions and use them at creation of procedures of transfer, processing and storage of data in network for protection from payment of taxes and fees.*

Keywords: *attacks, encryption, confidentiality, web-resource, authentication, https, SSL/TLS.*

Анализируя рейтинг атак от OWASP Top 10, представленные в таблице 1, можно сформулировать несколько методов, следуя которым разработчики Web-приложений и специалисты, занимающиеся их развертыванием и интеграцией, смогут обеспечить безопасность своих Web-ресурсов [1].

В первую очередь разработчикам Web-приложений необходимо считать, что любой пользователь их приложений обладает вредоносным потенциалом. Следовательно, требуется проверять все идентификационные и аутентификационные данные, полученные от пользователя, до того, как передавать, хранить и использовать их. Такой подход позволит избежать инъекций программного кода и XSS [1, 2].

Проверка пользовательских запросов также защитит от негативной стороны прямых ссылок на объекты и непроверенных редиректа и форвардинга. Контролируя запрос, можно применять политики доступа к ресурсам, а при использовании прямых ссылок информация может стать доступной кому угодно.

Следующая по важности область Web-приложения – это механизмы управления учетными записями пользователей и контроля доступа к ресурсам. Проверку прав доступа к ресурсам следует проводить на уровне программных функций, запрашивающих ресурс для дальнейшей передачи пользователю Web-приложения. Не стоит полагаться только на гло-

бальные переменные пользовательской сессии или «cookies-файлы» – контроль доступа должен осуществляться именно на уровне функций.

Таблица 1

Сравнение 10-ти самых актуальных атак на Web-приложения злоумышленниками в мировой практике

Актуальные атаки 2013 года	Актуальные атаки 2016 года
1. Инъекция программного кода (Injection)	1. Инъекция программного кода (Injection)
2. Межсайтовый скриптинг (Cross Site Scripting)	2. Межсайтовый скриптинг (Cross Site Scripting)
3. Ошибки при аутентификации и управлении сессиями (Broken Authentication and Session Management)	3. Ошибки при аутентификации и управлении сессиями (Broken Authentication and Session Management)
4. Небезопасные прямые ссылки на объекты (Insecure Direct Object References)	4. Небезопасные прямые ссылки на объекты (Insecure Direct Object References)
5. Межсайтовая подделка запросов (Cross Site Request Forgery)	5. Небезопасные настройки ПО (Security Misconfiguration)
6. Небезопасные настройки ПО (Security Misconfiguration)	6. Публикация конфиденциальной информации (Sensitive Data Exposure)
7. Небезопасная криптография (Insecure Cryptographic Storage)	7. Отсутствие контроля доступа на уровне функций (Missing Function Level Access Control)
8. Отсутствие контроля доступа к URL-адресам (Failure to Restrict URL Access)	8. Межсайтовая подделка запросов (Cross Site Request Forgery)
9. Недостаточная защита транспортного уровня (Insufficient Transport Layer Protection)	9. Использование компонентов с известными уязвимостями (Using Known Vulnerable Components)
10. Непроверенные редирект и форвардинг (Unvalidated Redirects and Forwards)	10. Непроверенные редирект и форвардинг (Unvalidated Redirects and Forwards)

Специалистам по развертыванию и интеграции Web-приложений стоит обратить внимание на конкретные настройки программного окружения Web-приложений (ОС, библиотеки, Web-сервер и др.), значения которых остаются без изменений. Если оставлять значение по умолчанию, то это может не только привести к состоянию абсолютно скомпрометированной информационной или автоматизированной систем, но и даст злоумышленнику дополнительное знание о защищаемой системе [2]. Имея доступ к аналогичному программному обеспечению, также с настройками по умолчанию, злоумышленник может создать похожую модель атакуемого Web-приложения, благодаря чему легко найдет лазейку сквозь оборону.

Стоит отметить, что требуется регулярно устанавливать обновления серверного программного обеспечения, в которых были исправлены ошибки предыдущих версий. Конечно, не стоит понимать это буквально и устанавливать обновления сразу же на рабочую станцию. Желательно иметь тестовый стенд, где можно будет проверить, не нарушат ли обновления работу основного Web-приложения и всей системы в целом. Это скорее рекомендации к принципам реализации политики безопасности в целом.

Требование использования новых версий относится также и к протоколам, обеспечивающим безопасность соединения между серверной стороной и клиентским Web-браузером. Стандарт на протокол TLSv1.2 был опубликован еще в 2008 году, однако Интернет-компании и разработчики Web-приложений массово начинают использовать его только сейчас. Это обусловлено двумя причинами.

Во-первых, поддержка TLSv1.2 в популярных Web-браузерах появилась совсем недавно. Версии Web-браузеров клиентов, поддерживающие эту версию протокола, стали реализовывать в конце 2013 года:

– “Internet Explorer” с версии 8 для Win 7 – март 2009;
– “Google Chrome” с версии 29 – сентябрь 2013;
– “Opera”: с версии 10 (июнь 2009) по 12, 14–16 – нет (переход на Blink), с 17 версии – октябрь 2013;
– “Firefox” с версии 27 – январь 2014;
– “Safari” с версии 7 – октябрь 2013.

Современные версии одноименных Web-браузеров для мобильных устройств на базе ОС Android и iOS также поддерживают протокол TLSv1.2.

Во-вторых, в связи с участившимися публикациями в СМИ о тотальной слежке в Интернете различных государственных ведомств, пользователи Web-приложений стали больше интересоваться средствами защиты конфиденциальности своих данных [2, 3].

Чтобы не потерять клиентов, коммерческим организациям приходится переходить на использование современных средств защиты, таких как TLSv1.2. На текущий момент версию 1.2 протокола TLS используют в своих Web-приложениях компании Google, Facebook, Сбербанк России и другие. Достоинство протокола TLSv1.2 заключается в том, что помимо улучшений, затрудняющих выполнение атаки «человек посередине» (Man in the middle – MITM), эта версия поддерживает новые криптографические алгоритмы и режимы их использования: SHA-256, AES128-GCM, AES256-GCM, а также ряд криптографических алгоритмов на эллиптической кривой.

Следует отметить, что не стоит делать протокол TLSv1.2 единственным доступным средством защиты, отбросив SSL 3.0, TLS версий 1.0 и 1.1, так как не все пользователи имеют современные версии Web-браузеров и не все ресурсы функционируют на данной версии с аттестованными удостоверяющим центром сертификатами.

Большинство сервисов использует шифрование с помощью Transport Layer Security (TLS) для передачи важных данных, в том числе паролей. Основные свойства TLS с точки зрения информационной безопасности:

– аутентификация – сервер всегда аутентифицируется, в то время как клиент аутентифицируется в зависимости от согласованного в процедуре «рукопожатия» алгоритма шифрования данных;

– целостность – обмен сообщениями включает в себя проверку целостности, основываясь на стандартах FIPS 113 и FIPS 140-2;

– частность канала – ассимметричное шифрование используется для установки соединения клиента с сервером и согласования настроек и алгоритмов шифрования. После установления соединения используется симметричное шифрование для всех сообщений данного и будущих диалогов [1].

Вариантов для перехвата данных, передаваемых через TLS, немного. Пассивное наблюдение злоумышленника не принесет желаемого результата, так как передаваемые данные зашифрованы временным ключом. Ключ известен и клиенту, и серверу, но его невозможно вычислить, лишь наблюдая со стороны за трафиком. Примером того, как можно реализовать защиту от пассивного наблюдения, является алгоритм Диффи-Хеллмана [1, 3].

Если злоумышленник не может «влезть» в защищенное соединение, то почему бы не установить два разных соединения: одно между клиентом и атакующим (который в данном случае притворяется сервером), и второе между атакующим и сервером. От этого алгоритм Диффи-Хелмана не защищен. Но TLS, в отличие от других протоколов, требует аутентификацию сервера, и поэтому такой подход также обречен на неудачу. Клиент при установлении соединения ждет от сервера обязательной аутентификации TLS. При реализации атак, основанных на MITM, вместо Web-сервера пользователь будет соединен с атакующим, который не владеет цифровым сертификатом и соответствующим доменным именем.

Существует два способа решения проблемы отсутствия сертификата SSL/TLS.

Первый способ – создать необходимый сертификат самостоятельно, то есть реализовать «самоподписанный» сертификат. Однако система цифровых сертификатов требует возможности проверить состоятельность сертификата. Для этого используются «удостоверяющие центры» (Certification Authority, CA), которые могут подписывать сертификаты сайтов (и не только), а также подписывать сертификаты других удостоверяющих центров. Сертификаты корневых удостоверяющих центров – их немного – внесены в Web-браузер разработчиками. И браузер считает сертификат подлинным, только если он находится в конце цепочки сертификатов, начинающейся из одного удостоверяющего центра, где каждый следующий сертификат подписан предыдущим в цепочке. Таким образом, процедура шифрования будет

реализована, но пользователь будет получать предупреждения от браузера о сомнительном диалоге и рекомендациями его прекратить.

Второй способ – получить у удостоверяющего центра сертификат для стороннего доменного имени, но применить его для атаки. Такой способ также недействителен в отношении большинства ресурсов, так как в каждом сертификате сайта указан домен, для которого он выдан [2]. В случае несовпадения домена в адресной строке с доменом, указанным в сертификате, браузер не реализует SSL/TLS шифрование.

При реализации описанных методов браузер покажет пользователю предупреждение, пример которого представлен на рисунке 1.

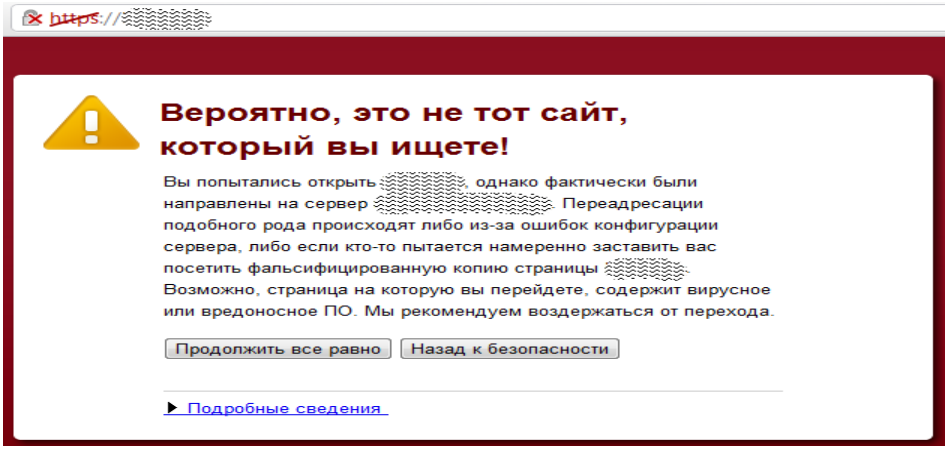

Рис. 1. Пример предупреждения пользователя о некорректной работе SSL/TLS сертификата

Ярко-красное предупреждение Web-браузера, показанное на рисунке 1, а также необходимость нажимать лишнюю кнопку содержания «Продолжить все равно» – это тот минимум мер, который побуждает пользователей избегать сайтов, где есть нарушения в работе TLS.

Пример шифрованного диалога пользователя с Web-ресурсом, особенно в моменты проведения финансовых транзакций или передачи аутентификационных данных, гарантируется использованием “https” и представлен на рисунке 2.

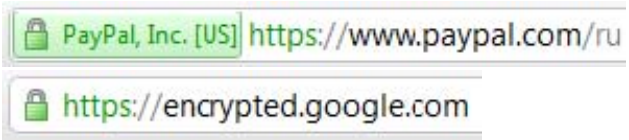

Рис. 2. Пример Web-сайтов, правильно использующих TLS с разным классом сертификатов

Если пользователь получит от браузера сообщение о невалидном сертификате, то он с достаточно большой вероятностью задумается, нужно ли ему на этот сомнительный Web-ресурс.

В случае, когда пользователь не получит от браузера подтверждение безопасности, то по сравнению с яркими предупреждениями об ошибках в TLS, которые он часто встречает на своем пути, отсутствие символа шифрования TLS пользователь заметит с куда меньшей вероятностью.

Пользователь видит, что соединение идет через “http”, но ошибок TLS нет, и, как было установлено, если браузер не выдает предупреждения о некорректной работе протокола, то пользователь, скорее всего, не заметит отсутствия шифрования, чем и воспользуется условный злоумышленник.

Для того чтобы принять меры по предотвращению дешифрования трафика, необходимо проанализировать различные атаки на “https”. При моделировании реализации атак использовался анализатор сетевого трафика “Wireshark”. Реализация атаки позволила дешифровать не только аутентификационные данные пользователей сети, но и весь трафик диалога «клиент-сервер».

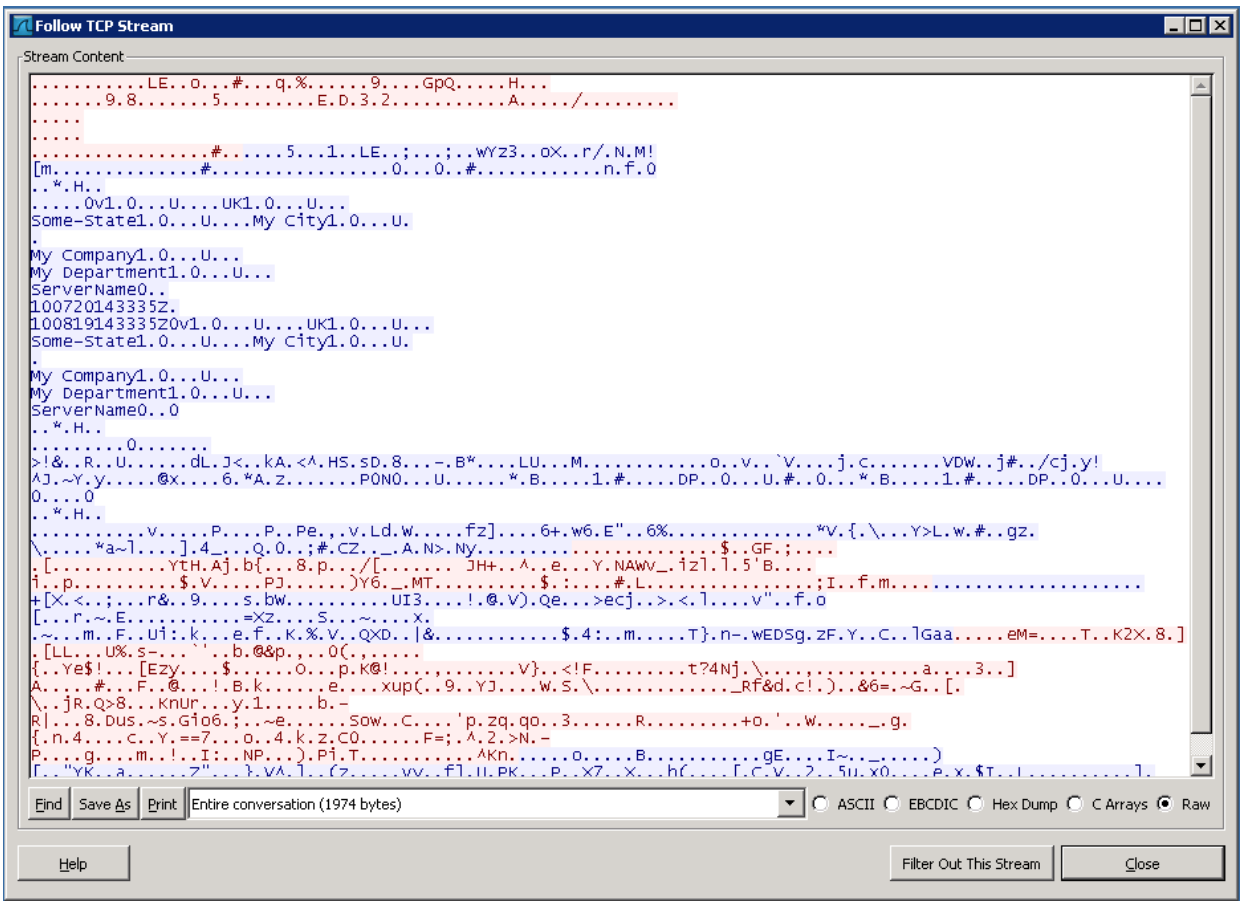

Рис. 3. Содержимое пакета при использовании SSL/TLS шифрования

Содержание пакета, в случае использования SSL/TLS шифрования не имеет никакой интеллектуальной нагрузки, символы представляют собой энтропийную последовательность, зашифрованную симметричным ключом [3]. Изначально “Wireshark” не имеет возможности проанализировать зашифрованный трафик SSL/TLS, как показано на рисунке выше. В случае согласования клиентом и Web-сервером симметричного шифрования RSA злоумышленник указывал в “Wireshark” похищенные приватные ключи (слепок), что позволяло расшифровывать трафик в режиме реального времени.

Рис. 4. Процедура «рукопожатия» при SSL/TLS шифровании

Данная уязвимость устраняется при реализации «прямой секретности» PFS (Perfect Forward Secrecy), и приватного ключа стало недостаточно, чтобы получить сессионный ключ, который используется для расшифровки данных.

Вторая проблема заключается в том, что приватный ключ не должен или не может быть выгружен с клиента, сервера или HSM (Hardware Security Module) [3]. В этом случае злоумышленник прибегает к сомнительным ухищрениям с расшифровкой трафика через “man-in-the-middle” с помощью “sslstrip”, что продемонстрировано на рисунке 5.

Рис. 5. Схема реализации атаки по хищению трафика аутентификации с помощью MITM “sslstrip”

На данный момент актуальным способом дешифрования SSL/TLS трафика является использование «.log-файлов» Web-браузера. Все современные браузеры поддерживают логирование, в том числе симметричных сессионных ключей, которые используются для зашифровки трафика, в файл. Злоумышленнику необходимо похитить данный файл и синтезировать его в “Wireshark”. Для этого необходимо добавить новую пользовательскую переменную “SSLKEYLOGFILE” и указать путь к файлу [2, 3].

 # export SSLKEYLOGFILE=/Users/username/sslkeylogs/output.log

 # open -a firefox

 # wireshark

Затем экспортировать похищенный “.log” в “Wireshark” и при анализе сетевого трафика с фильтром “ssl.handshake” вести аудит дешифрованного трафика, как показано на рисунке 6.

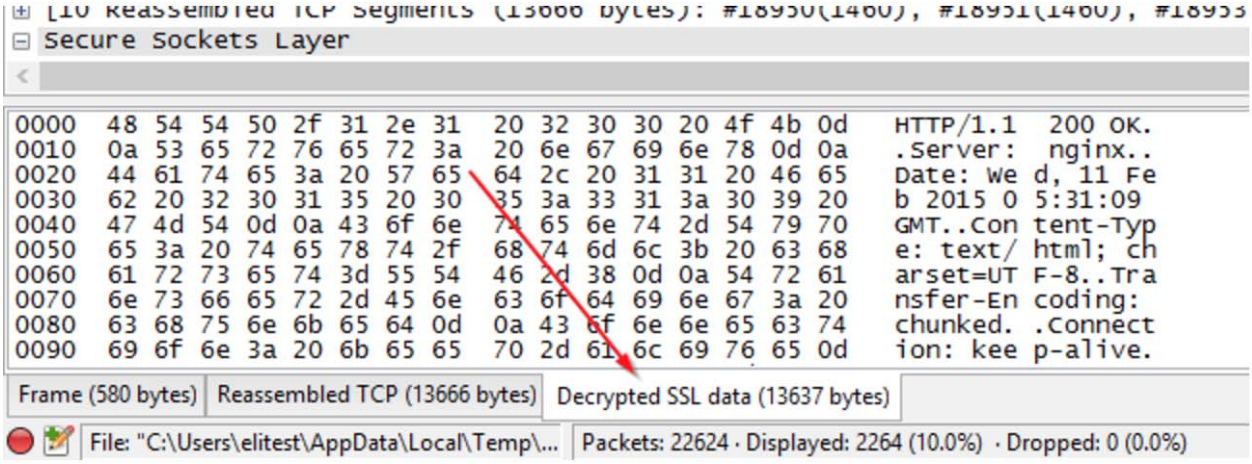

Рис. 6. Дешифрование SSL/TLS трафика в режиме реального времени

Таким образом, злоумышленнику необходимо иметь доступ к «.log-файлу» браузера «жертвы» для реализации дешифрования в реальном времени всего трафика, в том числе идентификационных и аутентификационных данных пользователя. Содержимое пакета (рис. 3) с учетом реализованного синтеза «.log-файла» представлено на рисунке 7 [2, 3].

Если сравнить рисунок 3 и рисунок 7, без сомнения можно говорить об информативности содержимого пакета. Таким образом, в статье были продемонстрированы уязвимости, реализация которых позволила условному злоумышленнику получить доступ к данным авторизации как в открытом виде, так и файлам “cookies”.

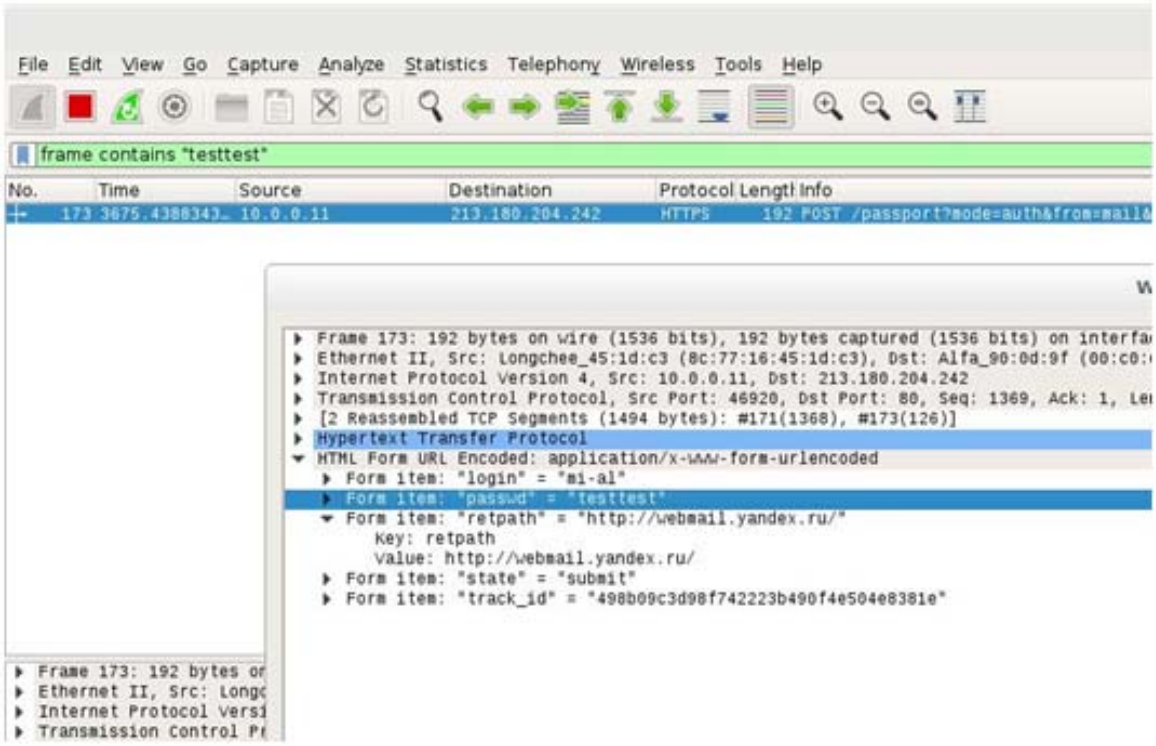

Рис. 7. Содержимое пакета после дешифрования трафика “https”

Примечания:

1. Власенко А.В., Дзьобан П.И. Разработка и системный анализ математической модели угроз, модели нарушителя, процедур защиты Web-приложений на всех этапах функционирования // Политематический сетевой электронный научный журнал КубГАУ = Scientific Journal of KubSAU. 2014. № 101. С. 2154–2164.
2. Власенко А.В., Дзьобан П.И. Разработка алгоритмов и программ выбора оптимального набора компонент нейтрализации актуальных угроз на основе описания модели и интеграции их в Web-приложение // Вестник Адыгейского государственного университета. Сер. Естественно-математические и технические науки. 2014. Вып. 3 (142). С. 189–193. URL: http://vestnik.adygnet.ru
3. Власенко А.В., Дзьобан П.И., Тимченко М.В. Разработка алгоритмов, инструментов и методов авторизации пользователей в Web-приложениях с использованием хеш-функций // Вестник Адыгейского государственного университета. Сер. Естественно-математические и технические науки. 2015. Вып. 4 (171). С. 144–150. URL: http://vestnik.adygnet.ru

References:

1. Vlasenko A.V., Dzoban P.I. Development and system analysis of the mathematical model of threats, the model of the intruder, procedures for protecting Web-applications at all stages of functioning // Polythematic Network Electronic Scientific Journal of KubSAU. 2014. No. 101. P. 2154–2164.
2. Vlasenko A.V., Dzoban P.I. Development of algorithms and programs to choose the optimal set of components of actual threat neutralization on the basis of model description and their integration in Web appendix // The Bulletin of the Adyghe State University. Ser. Natural-Mathematical and Technical Sciences. 2014. Iss. 3 (142). P. 189–193. URL: http://vestnik.adygnet.ru
3. Vlasenko A.V., Dzoban P.I., Timchenko M.V. Development of algorithms, tools and methods for user authentication in the Web-applications using hash functions // The Bulletin of the Adyghe State University. Ser. Natural-Mathematical and Technical Sciences. 2015. Iss. 4 (171). P. 144–150. URL: http://vestnik.adygnet.ru