УДК 004.5

Белоусов А.В., Глаголев С.Н., Кошлич Ю.А., Быстров А.Б.
БГТУ им. В.Г. Шухова, г.Белгород
koshlich@yandex.ru

**Решение вопроса информационной безопасности в интерактивных обучающих системах с удаленным доступом**

В статье рассматриваются проблемы разработки и создания интерактивных обучающих систем с удаленным доступом к лабораторному оборудованию. Особое внимание уделяется вопросу безопасности передачи данных между сервером и клиентской станцией и организации доступа к ресурсам системы. Предлагается подход с HTTP аутентификацией пользователей на основе HTML-формы с шифрованием трафика посредством SSL/TLS.

Ключевые слова: интерактивные обучающие системы, удаленный доступ, HTTPS, HMAC.

**Approach to informational security concerns of remote interactive learning systems**

We discuss diffuculties, which usually arise during development of interactive e-learning systems, aimed at providing remote access to laboratory equipment. Particular attention is paid to securing client-server communication and organization of a foolproof remote access to equipment. We propose an approach, relying on secure form-based HTTPS authentication with signed-cookie session management, as robust and effective solution.

Keywords: interactive e-learning systems, remote access, HTTPS, HMAC.

Современные информационные технологии все шире проникают в образовательный процесс. Уже невозможно представить работу обучающих систем без использования мультимедийных технологий, которые обеспечивают представление текстового, иллюстративного и видеоматериала, поясняющего устройство сложных технологических объектов и их работу; анимационное представление иллюстративного материала, обеспечивающее возможность интерактивного взаимодействия обучаемого с изучаемым курсом; звуковое сопровождение изучаемого материала; тестирование знаний в режимах самообучения и экзамена.

Применение в составе интерактивных обучающих систем лабораторного оборудования позволяет значительно снизить расходы образовательных учреждений на приобретение, внедрение и обслуживание весьма дорогого лабораторного оборудования и построить крупномасштабный комплекс для практического изучения по различным направлениям науки и отраслям. Важным аспектом интерактивных обучающих

систем является предоставление удаленного доступа пользователям посредством сети интернет. С точки зрения методики создания, разработки и использования распределенных интерактивных лабораторий с удаленным доступом, можно выделить важные проблемы информационной безопасности, которые заключаются в организации защищенных каналов связи, разграничении прав доступа и защиты от несанкционированных вмешательств извне. Несанкционированное вмешательство может вывести из строя дорогостоящее оборудование.

Основным вопросом в решении проблемы информационной безопасности распределенных ресурсов интерактивных обучающих систем является выбор и реализация метода аутентификации web-пользователей.

С точки зрения устойчивости к сетевым атакам методы аутентификации web-пользователей условно разделяются на неустойчивые (базовая аутентификация (Basic Access Authentication), аутентификация на основе HTML-форм (HTTP+HTML аутентификация)) и устойчивые (дайджест-аутентификация (Digest Access Authentication), использование клиентского SSL-сертификата, а также с использованием протокола SRPP (Secure Remote Password Protocol)).

Дайджест аутентификация (DA) достаточно надёжна, однако неудобна для конечного пользователя и практически не поддается настройке со стороны разработчика и по этой причине мало распространена. Детали реализации работы с пользовательским SSL-сертификатом различных браузеров отличаются между собой и не в полной мере стандартизованы. Этот факт препятствует широкому распространению клиентских сертификатов в сфере веб-аутентификации, несмотря на их высокую надёжность.

Наиболее распространена аутентификация пользователей посредством передачи по протоколу HTTP значений полей HTML-формы. При этом логин и пароль пользователя передаются серверу открытым текстом без модификаций, что делает схему уязвимой даже для простейших видов прослушивания трафика. Возможное хеширование пароля с помощью клиентского сценария перед его передачей на сервер никак не сказывается на практическом уровне информационной безопасности, так как хеш, в случае перехвата, может быть использован злоумышленником для повторной аутентификации[1].

На практике эффективным способом борьбы с перехватом конфиденциальной информации является использование для её передачи защищённого канала на основе технологий криптографических протоколов SSL или TLS. Протоколы используют асимметричную криптографию для обмена ключами, а также симметричное шифрование данных для конфиденциальности и коды аутентичности для сохранения целостности сообщений.

За счёт того, что взаимодействие с сервером в процессе аутентификации происходит через защищённое соединение на основе TLS,

становится невозможным перехват конфиденциальной информации, поскольку:

1) на этапе обмена ключами пользовательский агент (веб-браузер) проверяет подлинность предоставленного сервером сертификата и удостоверяется в том, что форма аутентификации не является поддельной, препятствуя попыткам «фишинга» со стороны злоумышленника;

2) передаваемая пользовательским агентом информация шифруется с помощью временного ключа, созданного в процессе установления соединения, что исключает возможность дешифровки данных при прослушивании сетевого трафика злоумышленником. Описанное выше расширение протокола передачи HTTP за счёт SSL/TLS носит названия HTTPS.

После проверки полученных от пользовательского агента данных и его авторизации, в рамках дальнейшей коммуникации серверу необходимо идентифицировать запросы, поступающие от этого агента, в течение некоторого времени, то есть организовать временный виртуальный канал связи с агентом. Аутентификация на основе HTML-форм является общепринятым и распространённым подходом, но не стандартизована, то есть конкретный механизм идентификации запросов авторизованных пользователей не регламентируется и выбирается разработчиками веб-ресурсов на собственное усмотрение. Однако, в этой области существуют распространённые подходы и соглашения, являющиеся стандартами де-факто. [2]

Пары «запрос-ответ» протокола HTTP являются независимыми друг от друга самостоятельными транзакциями. Иными словами, протокол не отслеживает связи между отдельными запросами. Поэтому для идентификации запросов, поступающих от авторизованных агентов, в случае HTTP+HTML аутентификации используются виртуальные сессии на основе «cookie».

После проверки предоставленной информации, в случае успешной аутентификации, сервер создаёт виртуальную сессию и генерирует временный ключ (идентификатор), который передаётся пользовательскому агенту и хранится в виде «cookie» на стороне клиента. Данный ключ, в соответствии с механизмом работы «cookie», передаётся серверу в заголовке каждого HTTP-запроса, относящегося к веб-ресурсу, до момента истечения срока действия «cookie». По значению полученного временного ключа сервер определяет сессию, к которой относится запрос (рисунок 1).

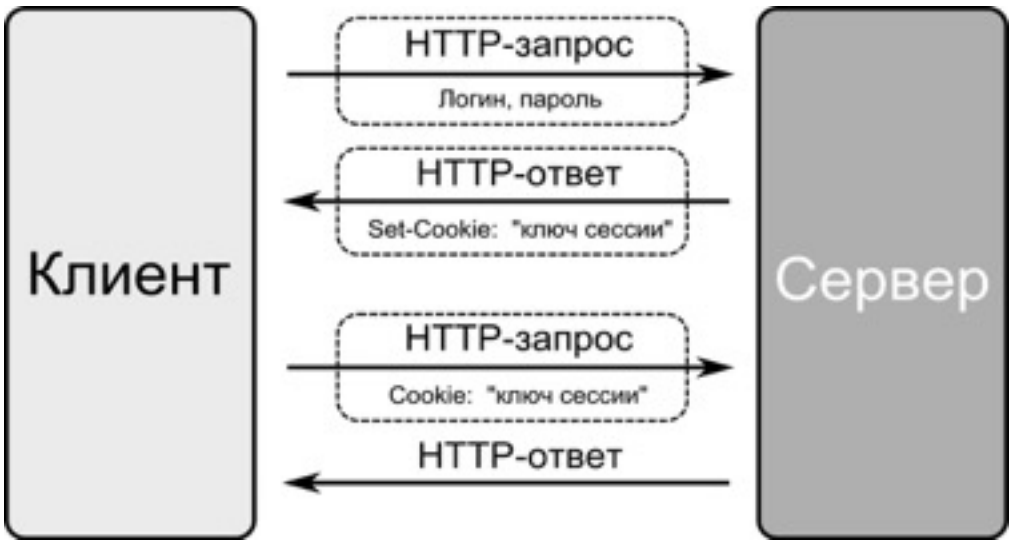


*Рисунок 1. Механизм идентификации запросов авторизованных агентов.*

Информация, позволяющая проверить подлинность полученного ключа, и определить, к какой сессии относится полученный запрос, хранится на сервере и во избежание её перехвата по сети не передаётся. Также, в целях безопасности для передачи временных ключей не используются «cookie» с неограниченных сроком действия.

Для того, чтобы можно было гарантировать невозможность подделки ключа сессии злоумышленником, временный ключ должен удовлетворять ряду критериев. Во-первых, временный идентификатор сессии должен быть устойчив к простому перебору значений. Во-вторых, значения идентификаторов должны быть криптографически случайны. В-третьих, должны существовать способы проверки подлинности ключа сервером и обнаружения попыток модификации ключа. [1]

Представленным критериям удовлетворяют значения, получаемые в результате хешированя текстовой строки и последующего шифрования результата с использованием секретного ключа. Исходная текстовая строка получается путём конкатенации некоторого подмножества известных серверу данных о конкретной сессии, - этим обеспечивается верифицируемость временного ключа.

Примером такого механизма является, HMAC (hash-based message authentication code - хеш-код аутентификации сообщений) на основе алгоритма криптографического хеширования SHA1. HMAC — общее наименование механизмов шифрования, которые используют криптографические хеш-функции в сочетании с секретным ключом.

С целью использования полученного идентификатора сессии в виде «cookie» и передачи по протоколу HTTP, его необходимо преобразовать в форму ASCII-строки. Для этого полученное на выходе HMAC число представляется в 64-разрядной системе счисления base64.

Пример 64-разрядного кода аутентификации сообщения, полученного методом HMAC-SHA-1 из строки «login:datetime» с секретным ключом «SECRET_KEY», приведён ниже:

owqliFxF0h0OBL2F6dMDb8gkvjo=

Строка «login» в данном примере представляет логин авторизованного пользователя, а «datetime» дату и время создания сессии. Для того чтобы надёжно ограничить временные рамки актуальности ключа, дата и время активации добавляются к исходной строке в виде целого числа секунд, прошедших с 1 января 1970 года, то есть величины «epoch» Unix-систем.

Полученные коды аутентификации сообщений используются в составе «подписанных cookie». Строка данных «подписанных cookie» состоит из некоторого достаточного идентификатора сессии и его «подписи». Подпись представляет собой код аутентификации сообщения, полученный из достаточного идентификатора.

Наиболее простым и надёжным способом получения «подписанного cookie» является использование в качестве достаточного идентификатора некритичной составляющей пользовательских данных, например логина, вместе с датой создания сессии, как это было сделано в предыдущем примере. Временный ключ сессии в виде «подписанного cookie» из предыдущего примера выглядит следующим образом:

login:datetime;owqliFxF0h0OBL2F6dMDb8gkvjo=

Исходная строка и её код аутентификации разделены в данном случае символом «;».

Применение HMAC-SHA-1 гарантирует невозможность генерации верного кода аутентификации, если отсутствует хотя бы один из необходимых компонентов: исходная строка или секретный ключ. Исходная строка содержится в «подписанном cookie» вместе с её кодом верификации, однако секретный ключ известен только серверу и не передаётся по сети. Таким образом, до тех пор пока секретный ключ остаётся известным только серверу, подделка такого идентификатора сессии невозможна. [1]

Реализация функции генерации кода аутентификации сообщения HMAC-SHA-1 для некоторой сессии на языке Python 2.7 приведена ниже.

```
import binascii
import hmac
from hashlib import sha1
from time import mktime

def session_key(secret, login, session_datetime):
    timestamp = mktime(session_datetime.timetuple())
    raw_string = u":".join(login, unicode(timestamp))
    hashed = hmac.new(secret, raw_string, sha1)

    return binascii.b2a_base64(hashed.digest())[:-1]
```

Функция session_key возвращает строку с 64-разрядным представлением кода аутентификации исходной строки. Использование «подписанного cookie» позволяет сократить число обращений к базе данных сессий сервера или вообще отказаться от её использования, так как информация необходимая для верификации временного ключа сессии и идентификации сессии, к которой относится данный запрос, содержится в самом ключе.

Таким образом, применение HTTP аутентификации пользователей на основе HTML-формы с шифрованием трафика посредством SSL/TLS позволяет организовать достаточно высокий уровень информационной безопасности ресурсов интерактивных обучающих систем с удаленным доступом.

## Список литературы

1. Fu K. и др. Dos and Don’ts of Client Authentication on the Web // Proceedings of the 10th USENIX Security Symposium. 2001. Т. 42.

2. Akhawe D. и др. Towards a formal foundation of web security // Computer Security Foundations Symposium (CSF), 2010 23rd IEEE. 2010. С. 290–304.