Труды научно-технической конференции кластера пензенских предприятий,
обеспечивающих **БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ.**
Том 9, Пенза-2014, с. 58-63 (http://www.pniei.penza.ru/RV-conf/T9/C14)

БИОМЕТРИЧЕСКАЯ ПОДДЕРЖКА ДОВЕРЕННОГО ПРОГРАММНОГО АГЕНТА, КОНТРОЛИРУЮЩЕГО КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ ОБМЕНА МГНОВЕННЫМИ СООБЩЕНИЯМИ

Карпушин Д.А., Майоров А.В. (Пенза)

Интернет, в его современном виде, претерпел значительные изменения – из военной сети обмена данными высокой устойчивости, коей была ARPANET в самом начале своего существования, она эволюционировала во всемирную сеть передачи данных, состоящую из тысяч более мелких сетей – корпоративных и домашних.

Одними из первых протоколов обмена сообщениями в сети были почтовые протоколы, такие как POP и SMTP. Затем появился протокол IRC, который позволил пользователям обмениваться т.н. «мгновенными сообщениями» в реальном времени. В настоящее же время каждый год появляются новые протоколы прикладного (высшего, согласно модели OSI) уровня для обмена данными. Среди популярных (на момент написания статьи) сервисов, предоставляющих услуги обмена мгновенными сообщениями, стоит упомянуть: Skype, ICQ и XMPP (более известен как Jabber).

Все упомянутые выше протоколы, несомненно, предоставляют своим пользователям определённый уровень безопасности и конфиденциальности: присутствует как шифрование передаваемых по каналу связи данных, так и аутентификация пользователей. Все эти меры, естественно, реализуются в интересах конечного пользователя и направлены на защиту передаваемых им данных. Для образования защищённого канала связи чаще всего применяется криптографический протокол SSL/TLS[1]. Использование этого протокола позволяет зашифровать все передаваемые данные, что делает бессмысленным перехват этих самых данных – атакующая сторона не сможет прочитать перехваченные сообщения. Для аутентификации пользователей стандартом де-факто является аутентификация по связке логин/пароль – это самое простое решение с точки зрения технической реализации с одной стороны, и достаточно комфортное с точки зрения повседневного использования конечным пользователем с другой стороны.

Схему взаимодействия клиента и сервера в условиях парольной аутентификации можно увидеть на рисунке 1. Пользователь вводит логин/пароль, клиент хэширует пароль «с солью», отправляет серверу и ждёт подтверждения правильности ввода. При подтверждении правильности ввода клиент с сервером начинают работать в штатном режиме.

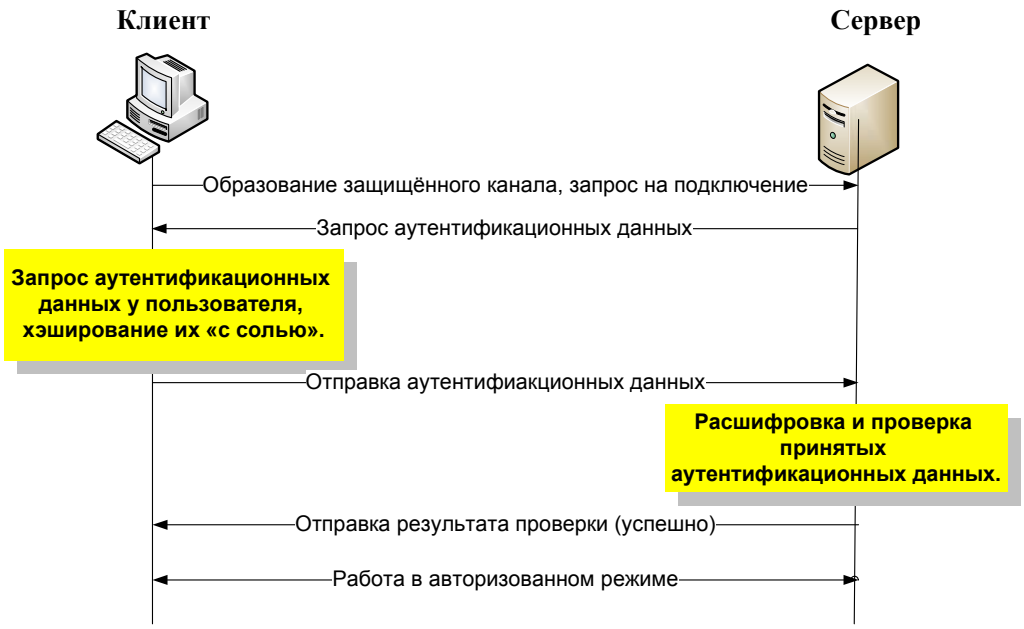


Рис.1 Схема взаимодействия клиента и сервера (парольная аутентификация)

В рассмотренном нами случае гипотетические проблемы с безопасностью начинаются ещё на этапе образования защищённого канала связи. Как было сказано ранее, для образования оного обычно используется протокол SSL/TLS, а он, в свою очередь, подвержен атаке типа “Man in the middle” (человек посередине) [1]. Суть данного типа атак заключается в том, что злоумышленник может для каждого соединения создать две независимые сессии: одну с клиентом, представившись ему как сервер, а другую – с сервером, представившись ему как клиент. Клиент устанавливает соединение с атакующим, а тот, в свою очередь, создает соединение с сервером. Даже если клиентское приложение заподозрит что-то неладное (например, провалившуюся проверку подлинности сертификата) и предупредит об этом пользователя, то пользователь, как это часто бывает, просто проигнорирует данное предупреждение. К тому же, у злоумышленника может оказаться сертификат, подписанный центром сертификации. Из всего этого следует, что при успешной атаке злоумышленник может как минимум следить за перепиской, а как максимум – искажать информацию, делая её недостоверной.

Ещё одним слабым местом данной схемы является осуществление авторизации по логину/паролю, что не является достаточно надёжным видом аутентификации. Существует огромное множество способов скомпрометировать логин и пароль: подсмотреть, допросить оператора «с пристрастием» и иже с ними. Не стоит забывать и про человеческий фактор – оператор уснул на рабочем месте или отошёл от рабочего места, не выйдя из системы; записал аутентификационные данные на каком-либо носителе информации, доступ к которому неограничен и т.п. Данные оплошности позволяют третьим лицам получить несанкционированный доступ к системе в лучшем случае. В худшем – сменить пароль и получить полный контроль над системой.

Проанализировав сложившуюся ситуацию можно придти к выводу, что необходимо разработать программный агент с повышенным уровнем доверия.

Добиться этого можно с помощью биометрической аутентификации, которая, на наш взгляд, является гораздо более удобным и безопасным способом аутентификации.

Биометрическая аутентификация – это аутентификация пользователя, осуществляемая путём предъявления им своего биометрического образа [2]. Под биометрическим образом, в данном контексте, подразумевается какая-либо отличительная черта, характеристика человека: отпечаток пальца, ладони, сетчатки глаза и т.п. В основе метода биометрической аутентификации лежит т.н. «преобразователь биометрия-код» (ПБК). На вход преобразователя подаётся биометрический контейнер – это, по сути, способы и методы преобразования биометрических данных пользователя в т.н. «ключ». Ключ – это некоторые данные, по виду которых можно судить об успешности преобразования.

Например, при создании биометрического контейнера мы используем аутентификацию по отпечатку одного пальца, а в качестве ключа - некоторую фразу, например, «Мама мыла раму». При подаче на вход ПБК этого контейнера потребуется предоставить ещё и образ своего отпечатка пальца. Считав образ, преобразователь, основываясь на методах, хранящихся в контейнере, преобразует этот образ в символьное представление. Т.о. если пользователь действительно тот, за кого себя выдаёт (если отпечаток пальца совпал с тем, что был введён при создании контейнера), то на выходе получится исходная фраза «Мама мыла раму». Если же наш образ низкого качества, например, имеется значительный шрам на пальце, то на выходе мы можем получить что-то типа «м4ма м@ла ра%у». Если же мы предоставили образ совершенно другого человека (или другой палец, например), то на выходе получится набор несвязных букв и цифр, например, «ndubl*&ykghl^&%$77».

Преимущества биометрической аутентификации над парольной очевидны: не нужно ничего запоминать – достаточно приложить палец или сетчатку глаза к сканеру; процесс аутентификации занимает меньше времени; чтобы скомпрометировать ваши данные необходимо сделать очень дорогую и высококачественную подделку (не без вашего непосредственного участия и против вашей воли). К возможным минусам биометрической аутентификации можно отнести необходимость приобретения (или создания собственных) аппаратно-программных комплексов считывания, обработки и преобразования биометрических данных в ключ, а так же хрупкость человеческого тела относительно различных внешних воздействий окружающей среды.

В рамках исследования методов биометрической аутентификации, как одного из возможных способов аутентификации, была создана система «БиоПароль». В ней используются самые актуальные наработки ЛБНТ ОАО «ПНИЭИ» в виде набора библиотек для биометрической аутентификации.

Сама по себе, система представляет собой программный агент с повышенным уровнем доверия, аналог множества имеющихся в данных момент в интернете служб обмена мгновенными сообщениями с возможностью передачи файлов. Для образования защищённого канала связи используется библиотека OpenSSL, в качестве основного языка программирования был использован C++, а

в качестве вспомогательного фрэймворка для графического интерфейса пользователя, клиент-серверной части и прочего – Qt.

Функции системы:

- Установка и закрытие соединения
- Отправка файлов и сообщений
- Запрос на подтверждение личности не только при непосредственном подключении, но и в любой момент времени
- Удаление данных аутентификации локального пользователя или своих у удалённого пользователя
- Настройки программы (например, указание порта)

Система поддерживает аутентификацию либо по связке логин/пароль, либо биометрическую аутентификацию (по образу отпечатка пальца). Схема работы системы в случае биометрической аутентификации представлена на рисунке 2.

Рис.2 Схема клиент-серверного взаимодействия системы при подключении нового клиента (проверка клиента сервером)

Самым первым шагом является генерация своих аутентификационных данных. Генерирование состоит в создании файла с аутентификационными данными. В случае парольной аутентификации в файле будет расположен хэш от введённого пароля, а логин пользователя – это имя файла. В случае биометрической аутентификации вызывается мастер, создающий биометрический контейнер: будет необходимо, во-первых, обучить нейронную сеть путём ввода

как минимум 12 образов одного и того же пальца и, во-вторых, ввести «ключ». В нашем случае ключом будет являться, опять таки, хэш от пароля. После создания файла с аутентификационными данными его необходимо перенести на внешнем носителе на компьютер потенциального собеседника. Данная мера хоть и понижает удобство работы для конечного пользователя, но полностью исключает атаку типа «человек посередине».

После обмена файлами с аутентификационными данными можно уже непосредственно приступать к установлению связи: защищённое соединение устанавливается с помощью протокола SSL/TLS. После этого на машине вашего абонента проверяется, есть ли такой пользователь в базе. И если такой пользователь имеется – то проверяется, какой тип аутентификации должен использоваться и отправляется соответствующий запрос вашему клиенту. В запросе, помимо непосредственно запроса, находится и биометрический контейнер, хранящийся у сервера. Клиент, получив контейнер, «извлекает» из него ключ – на вход контейнеру подаётся образ вашего отпечатка, а на выходе мы получаем код (рисунок 3).

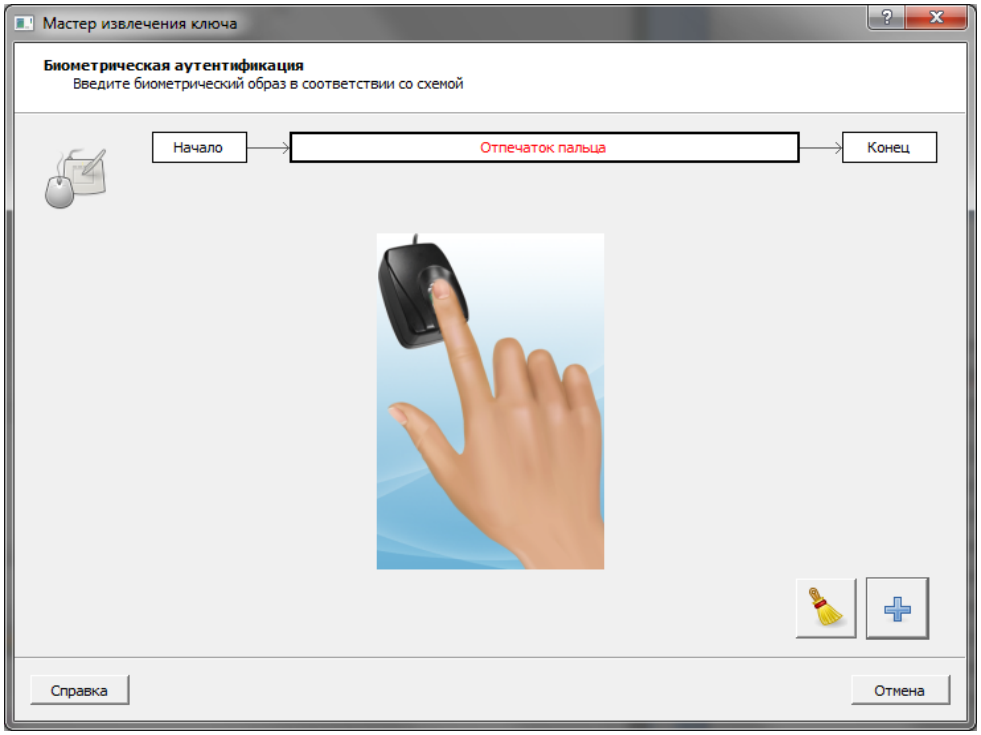


Рис.3 Запрос биометрической аутентификации у клиента

```
STORED HASH   :"da4b9237bacccdf19c0760cab7aec4a8359010b0"

INCOMING HASH: "di4b9237bacccdf19c0760cab7aec4a8359010b0"
```

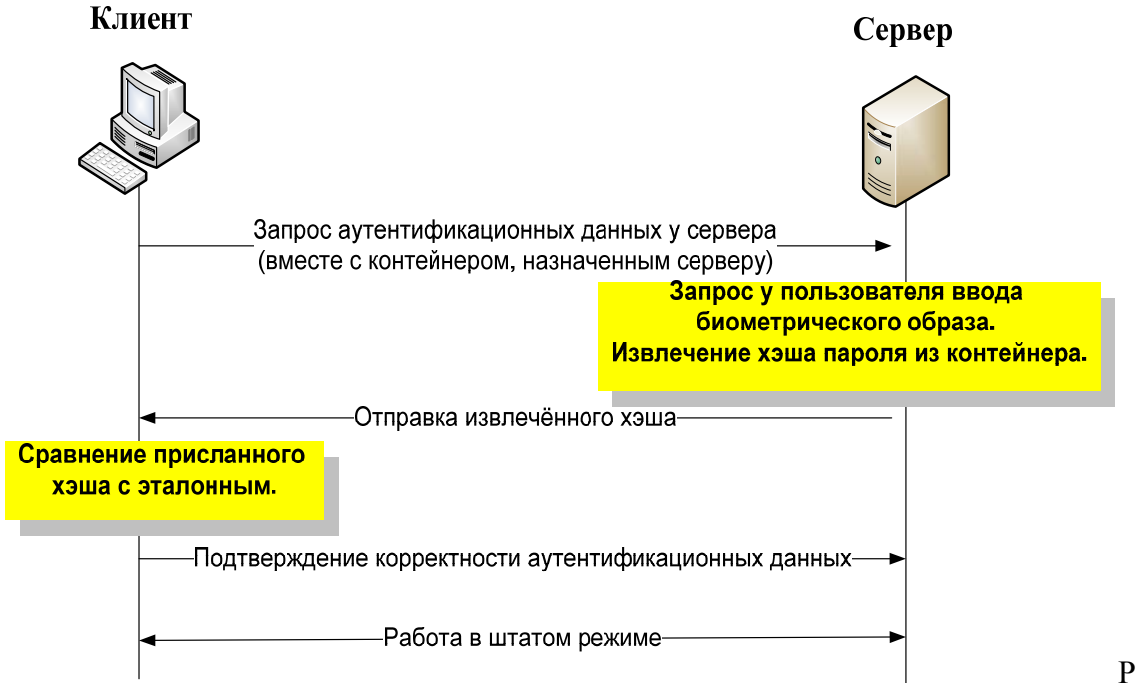
Рис.4 Различия в имеющемся на сервере и полученном от клиента (искажённом) хэшах

Как уже было упомянуто, в случае предоставления системе искажённого образа аутентификация будет неуспешной. На рисунке 4 продемонстрированы 2

хэша: хранящийся на сервере (эталонный) и полученный от клиента (на основе незначительно искажённого образа).

Как видите, разница между хэшами всего в одном символе. Но и этого достаточно, чтобы провалить аутентификацию.

После успешной аутентификации, клиент может проверить аутентификационные данные сервера, нажав кнопку «Подтвердить личность». Данный вариант использования системы вы можете увидеть на рисунке 5.



Р

ис.5 Проверка клиентом аутентификационных данных сервера

Как вы могли убедиться, система «БиоПароль» успешно выполнила возложенную на неё функцию по повышению достоверности обмена сообщениями путём устранения уязвимости типа «человек посередине» и минимизации угрозы компрометации аутентификационных данных. Добиться этого удалось применением методов биометрической аутентификации.

Идеальных систем аутентификации на данном этапе развития человечества в целом и информационных технологий в частности ещё не существует, да и принципиальная возможность их создания, по нашему мнению, невозможна. Но биометрическая аутентификация, в её текущем состоянии, находится ближе всего к тому, чтобы называть её лучшей из имеющихся.

Литература

1. Венбо Мао Современная криптография: теория и практика. М. Изд-во: Издательский дом «Вильямс», 2005 г.
2. ГОСТ Р 52633.0 Требования к средствам высоконадёжной биометрической аутентификации.

Материал поступил 13.10.14, опубликован по положительной рецензии д.т.н. Малягина А.Ю.