УДК 004.056

БЕЗОПАСНОСТЬ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ ПЛАТЕЖНЫХ СИСТЕМ

Поляков Н.Н., Пуртинов А.М., Легконогих А.Н., Никишина Т.Г., Черкесова Л.В.
Донской государственный технический университет, Ростов-на-Дону, e-mail: polyak_1995@mail.ru

В предлагаемой статье рассматриваются угрозы информационной безопасности, возникающие в современном мире, направленные на банковские и платёжные коммерческие системы, используемые во всём мире и в России в частности. Описаны схемы работы самых популярных российских банковских и платежных систем, использующих электронные деньги, и их возможные уязвимости, вызванные возрастанием денежных расчётов через интернет. Не все банковские устройства поддерживают биометрическое сканирование, поэтому возникает вопрос о том, каким образом огромное количество пользователей банковских платёжных систем может защитить свои персональные данные от несанкционированных действий (от взлома хакеров)? Для обеспечения безопасности платежей авторы предлагают использование дополнительной усложненной аутентификации – применение упрощенного криптографического токена, имеющего вид программного приложения и установленного на USB-устройство. Это позволит усилить безопасность на этапе аутентификации, так как программное приложение, генерирующее одноразовые пароли и проверяющее подлинность пользователя с помощью белого списка, способно устранить существующие уязвимости и бреши в информационной защите банковских, платёжных коммерческих систем. Цель исследований заключается в обеспечении безопасности аутентификации пользователей банковских, платёжных и коммерческих систем. Задача данной работы – проверка уязвимости этапа аутентификации в электронном кошельке пользователя и устранение найденных проблем с помощью создания криптографического токена на базе языка программирования Python и программного приложения Pinentry. Для решения такой задачи нужно рассмотреть: возможность кибератак на личные данные пользователя во время входа на сайт банковской или коммерческой платежной системы; создание дополнительной проверки с помощью случайно сгенерированного одноразового пароля на внешнем носителе; создание и формирование «белого списка» IP адресов пользователей.

Ключевые слова: платежные системы, атаки на платежные системы, аутентификация, Pinentry, криптографический токен, Python

SECURITY USER AUTHENTICATION PAYMENT SYSTEM

Polyakov N.N., Purtinov A.M., Legkonogikh A.N., Nikishina T.G., Cherkesova L.V.
Don State Technical University, Rostov-on-Don, e-mail: polyak_1995@mail.ru

In the offered article, the threats of informational safety, arising in the modern world, directed on the bank and commercial payment systems, using in all world and in Russia in particular, are considered. The circuits of operation of the most popular Russian banking and commercial payment systems, using electronic money (e–cash), and their possible vulnerability, called by increase of monetary calculations through the Internet, are described. Not all bank devices support biometric scanning; therefore there is question on how the huge amount of bank payment systems users can protect their personal data from not authorized operations (hackers breaking)? For support of payments safety, authors offer using of the additional complicated authentification – the application of simplified cryptography token which is looking like program application, and installed on USB – device. It will allow amplifying the safety at authentification stage as the program application generating of one – time keywords and checking (controlling) of user's authenticity with the helpness of white list, is capable to eliminate all existing vulnerabilities and gaps in the informational protection of banking and commercial payment systems. The purpose of investigations consists of providing authentification security of banking and commercial payment systems users. The problem of this work is vulnerabilities checking on the authentification stage in electronic purse of users. To eliminate the found problems is possible by means of cryptographic token creation on the base of programming language Python and program applications Pinentry. To solve this problem you need to consider: consideration of possible cyber attacks on the user's personal data during input on the site of banking or commercial payment system; creation of additional check control by means of casually generated one–time keyword on the external carrier; creation and formation of «white list» users IP addresses.

Keywords: payment systems, attacks on payment systems, authentication, Pinentry, cryptographic token, Python

Платежные системы – это звенья, обеспечивающие удобство проведения финансовых операций для клиентов банков.

В Российской Федерации на сегодняшний день самыми популярными являются:Visa International Service Association, Master Card International (WorldWide или Incorporated) и НСПК «МИР». *Visa* и *MasterCard* – ведущие международные платежные системы и обслуживаются по всему миру, «*МИР*» – это российская национальная платёжная система, которая является обязательной для всех работников государственных организаций [1].

В последние годы все большую популярность набирают расчеты через сеть Интернет. Они выполняются с помощью сайтов или клиентских приложений. Банки вырабатывают собственные системы безопасной аутентификации. Например, последняя разработка Сбербанка – Сбербанк ID.

Однако не все устройства поддерживают биометрическое сканирование, и возникает вопрос, каким образом огромной массе

пользователей можно защитить свои персональные данные. Для обеспечения безопасности платежей авторы предлагают использование дополнительной усложненной аутентификации – применение упрощенного криптографического токена, который будет иметь вид приложения и установлен на USB-устройство.

Цель исследования: обеспечение безопасности аутентификации пользователей банковских, платёжных и коммерческих систем.

Задача работы: проверка уязвимости этапа аутентификации в электронном кошельке и устранение найденных проблем с помощью создания криптографического токена на базе языка программирования Python и приложения Pinentry. Для решения такой задачи нужно рассмотреть возможные атаки на пользователя во время входа на сайт платежной системы, создать дополнительную проверку с помощью случайно сгенерированного одноразового пароля на внешнем носителе, а также сформировать «белый список» IP адресов.

Электронные платежные системы представляют собой сервисы, где можно завести личный кабинет с привязанным к нему электронным кошельком, в котором отслеживается движение электронных денег. Они позволяют проводить те же денежные операции, что и в случае с обычной банковской картой.

Популярные платежные сервисы в России [2]:

1. Яндекс.Деньги.
2. WebMoney.
3. QIWI (КИВИ).
4. PayPal.

Большинство систем являются не анонимными или частично анонимными. Оплату товаров через такие системы сегодня предлагает практически любой интернет-магазин. Для обеспечения безопасности своих клиентов платежные системы используют различные методы.

Например, в модели цифровых наличных, т.е. digitalcash, главной гарантией безопасности является стойкость криптографических протоколов, используемых при изготовлении цифровых денег, а также протоколы, регламентирующиеих использование. Цифровые деньги, как и наличные, содержат шаблонные сведения, обеспечивающие защиту от подделки. Такими данными могут быть: сведения о номинальной стоимости, эмитенте, серии, номере, а также элементы защиты от подделки путем заверения их цифровой подписью эмитента [3]. Но следует отметить, что создание и использование цифровых денег в нашем законодательстве пока не регламентируется, а значит – все операции с digitalcash осуществляются исключительно на договоренностях об использовании их как платежных средств. Это ставит под вопрос защиту клиентских данных от мошеннических атак, так как, если произойдет потеря данных – то доказать, что вы потеряли реальные деньги, будет очень сложно.

Нас интересует не юридическая сторона вопроса, а то, как клиент сможет защитить себя от перехватов данных и, следовательно, от потери собственных электронных денежных средств – сегодня этот вопрос актуален для всех.

Рассмотрим несколько наиболее популярных и масштабных платежных систем.

1. WebMoneyTransfer построен на основе использования электронных кошельков. Для проведения расчетов в системе используются учетные единицы, так называемые *титульные знаки*, которые являются для пользователя аналогом денег (WMR, WME, WMZ, WMU, WMY, WM-C и WM-D). WebMoney позволяет совершать переводы только между кошельками, имеющими одинаковую валюту [3].

2. Яндекс.Деньги. – основан на технологии платежной системы PayCash, использующей модель цифровых наличных. В отличие от других электронных кошельков разработчики использовали так называемую «платежную книжку». Ее главное достоинство – номинал, который подтверждается подписью банка. Это обеспечивает защиту от использование той же электронной валюты повторно. То есть при осуществлении оплаты банк проверяет и подтверждает, что приводимые виртуальные средства не использовались ранее.

3. PayCash – это система, которая позволяет хранить цифровую наличность непосредственно в электронных кошельках клиентов, совершенно независимо от сервера системы. С одной стороны, это обеспечивает более высокий уровень защиты электронных денег, а с другой стороны, если клиент теряет электронный носитель или на нем обнаруживаются технические неполадки, не будет возможности восстановить потерянные средства. Но с точки зрения безопасности она превосходит традиционные системы оплат, такие как платежные карты [3].

4. Система RUpay создана упростить систему переводов и мгновенно совершать сделки между пользователями не только внутри страны, но и на международном уровне. Еще один плюс данной системы – возможность интеграции с другими электронными кошельками, то есть с другими платежными системами. Клиент может оформить виртуальную карту Visa и распоряжаться ей че-

рез сервис PayPal (но его действие в России ограничено, так как он используется только для расчетов за покупки).

Проблемы безопасности платежных систем

В случае сетевых денег достаточно компрометации паролей и кодов безопасности, в результате атаки на компьютер пользователя, и деньги смогут украсть. Кроме того, неполадки с компьютером могут привести к потере полученных сертификатов безопасности, дающих доступ к электронному кошельку, либо самого электронного кошелька. В инструкциях по безопасности во всех платежных системах предлагаются определенные правила, соблюдение которых значительно снижает риски потерь электронной наличности.

Например, WebMoney предлагает дополнительную аутентификацию через мобильный телефон или электронную почту, а также использование одноразового ключа, который генерируется каждый раз при входе в личный кабинет. Но перехват данных позволяет злоумышленнику выполнять множество операций без дополнительного подтверждения.

Анонимность, которую обеспечивает эмитент, не привязывая свою подпись к конкретной цифровой купюре, становится уязвимостью для пользователя. Причиной является то, что этом случае пользователь может потерять деньги из-за атаки на устройство, с которого он осуществляет вход в платежную систему. Еще более опасны выходы через общественные сети.

Атаки на этапе аутентификации

Аутентификация – необходимая процедура для доступа к платежной системе. Сервис, которым пользуется клиент, берет на себя генерирование ключей и одноразовых паролей. Именно на данном этапе злоумышленник может начать получать необходимые ему данные для взлома электронного кошелька добросовестного пользователя [4].

Получить логин пользователя он сможет без проблем – перехват идентификатора сессии даст злоумышленнику все сведения о пользователе.

Когда пользователь заходит в свой электронный кошелек, ему присваивается номер (этот номер сохраняется в Cookie браузере и передаётся сайту при открытии каждой страницы) – для того, чтобы сохранить сведения о действиях в данной сессии. На сервере будет сохраняться файл, который отвечает за идентификацию пользователя. Если злоумышленник сумеет получить номер текущей сессии, то сможет выдать себя за добросовестного клиента и получить множество его данных (рис. 1).

Криптографический токен будет обеспечивать защиту от перехвата данной информации, когда злоумышленник будет пытаться стать посредником между клиентом и электронным кошельком. Примерно атака выглядит таким образом:

1. Расположившись на канале связи между пользователем и электронным кошельком, злоумышленник может перехватить запросы пользователя и отправить их на сайт – уже от своего имени.

```php
<?php
/*
протокол авторизации MAS
*/

session_start();
/*Праметры неавторизованной гостевой сессии*/
$_SESSION['nominee']='guest';            //индентификатор неаторизованной гостевой сессии (sid)
$_SESSION['ip']=getenv ("REMOTE_ADDR");    //определение и запись ip-адреса пользователя (ip1)
$_SESSION['timestamp']=time();           //временная метка начала протокола (t1)

require "config.php";           //подключение конфигурациоонного файла

/*вычисление хеша*/
$r = md5($k.$_SESSION['timestamp'].$_SESSION['ip']);

/*Вывод формы и передача скрипта*/
require "form.html";
echo $r;
echo " | ";
echo md5('user1_password');
echo " | ";
echo $REMOTE_ADDR;
?>
```

Рис. 1. Процесс получения файла сессии

```php
<?php
/*
протокол авторизации MAS
*/

/*Проверка старотвала ли гостевая сессия*/
if (! isset( $_REQUEST[session_name()] ))
{
echo "сессия не стартовала";
exit;
}

session_start();            //стартуем сессию
require "config.php";        //подключаем конфигурационный файл

/*Проверка получен ли ответ в допустимом временном интервале*/
if ( (time() - $_SESSION['timestamp']) > $access_interval)
{
echo "превышен временной интервал ответа";
exit;
}

/*Извлечение хеша пароля из БД*/
require "validate.php";        //подключаем файл с функцией для извленчения хеша пароля
if (validate($name) == "NULL")
{
echo "пользователь с таким именем не зарегистрирован";
exit;
}

if ( md5( md5($k.$_SESSION['timestamp'].$_SESSION['ip']).$name.validate($name)) == $r_type)
{
echo "Вы авторизированы";
exit;
}

echo "Пароль не тот";
```

Рис. 2. Процесс добычи логина

2. Злоумышленник может транслировать все действия пользователя, и сайт определяет его как своего клиента, так как он вводит необходимый пароль, который может знать только пользователь. В данном случае невозможно определить, что клиент не тот, за кого себя выдает, так как IP-источника идентичен на всех этапах протокола.

3. Для завершения атаки злоумышленнику достаточно выдать пользователю какую-нибудь ошибку HTTP и прервать связь. Клиент воспримет это как техническую неполадку или проблему с сервером (рис. 2).

Для того, чтобы обеспечить защиту от подобной атаки, необходимо изначально защитить себя от угрозы перехвата соединения. Решение, предлагаемое авторами – это создание приложения для клиентов, которое будет выступать посредником между кошельком и клиентом.

Шифрование PGP

В качестве меры безопасности при аутентификации используется пара ключей [5]. То, каким образом этот ключ защищен, и определяет эффективность его работы. Для аутентификации имеет перспективу шифрование PGP (рис. 3). Пользователь PGP создаёт ключевую пару: открытый и закрытый ключи. При генерации ключей задаются: их владелец (имя и адрес электронной почты), тип ключа, длина ключа и срок его действия. Открытый ключ используется для шифрования и проверки цифровой подписи. Закрытый ключ – для декодирования и создания цифровой подписи [6].

Приложение выступает в качестве протокола доверия, который подтверждает, что ключ, отправленный добросовестным пользователем, действительно принадлежит ему. То есть будет *недостаточно сведений* об *ID клиента*, правильной паре логин/пароль (которые, как мы показали выше, могут быть перехвачены злоумышленником). Приложение, предлагаемое нами, в данном случае является третьей стороной, которая свидетельствует и подтверждает, что не произошло подмены данных добросовестного пользователя.

Для этого приложение будет генерировать одноразовые пароли для доступа к закрытому ключу [7]. То есть невозможно будет создать цифровую подпись пользователя, не имея доступа к одноразовому паролю.

При каждом запросе парольной фразы необходимо будет ввести также и одноразовый пароль – его нужно дописать непосредственно к концу парольной фразы (не отделяя пробелом или какими-либо иными знаками). Введённый пароль тут же аннулируется: так, если пользователь

допустит опечатку в парольной фразе закрытого ключа, то при следующей попытке её ввода потребуется ввести уже следующий одноразовый пароль. Когда список паролей окажется пуст, какие-либо операции с закрытыми ключами станут недоступны, пока не будет сгенерирован новый список.

Схема работы криптографического токена

Листинг выглядит следующим образом (на базе Python) в простой форме вопрос/ответ:

<len_p> | <len_j> | JSON(<header>, <type>, <fields>, <files_meta>) | [binary]

len_p (8 bytes): Общая длина пакета.

len_j (8 bytes): Длина JSON-пакета.

header (list): Токен аутентификации auth (опционально) и версия приложения version.

type (str): Идентификатор пакета.

fields (list): Произвольный набор полей.

files_meta (dict): Отображение путь_файла->длина_файла.

binary (bytes): Конкатенированное содержимое файлов (опционально).

Удалённый ввод парольных фраз, по нашему мнению, должен быть реализован на основе специализированного приложения – прокладки Pinentry. Он выполняется по следующему протоколу:

server>gpg-agent: устанавливает в переменной окружения PINENTRY_USER_DATA (передаваемой через весь стек вызова gpg>gpg-agent>pinentry) сведения для установления IPC-канала, в том числе имя IPC-сокета и сеансовый ключ аутентификации.

gpg-agent>pinentry: вызывает pinentry, передавая переменную окружения PINENTRY_USER_DATA и инициализирует Assua–протокол.

server>pinentry: передаёт по IPC-каналу (UNIX-сокету) открытый сетевой сокет клиентского соединения непосредственно процессу pinentry.

pinentry>client: по предоставленному сетевому соединению посылает клиенту необходимые данные (текстовые строки и опции запуска, полученные от gpg-agent на этапе 2) для вызова стандартного приложения pinentry.

client: вызывает стандартное приложение pinentry и получает пользовательский ответ (в виде ответа Assuan-протокола).

client>pinentry: пересылает пользовательский ответ.

pinentry: выполняет «тревожные» команды, если введённая парольная фраза вызвала срабатывание каких–либо из них.

pinentry>gpg-agent: воспроизводит пользовательский ответ в начатом на этапе 2 Assuan-протоколе.

При вводе неверной парольной фразы цикл будет повторяться вновь и вновь, по требованию приложения.

Одноразовые пароли – это дополнительная мера безопасности для защиты закрытых ключей. После её активации, при использовании закрытого ключа, пользователь должен будет ввести короткий случайный пароль (из заранее сгенерированного списка) наряду с основной парольной фразой. Поскольку для каждой операции такой пароль уникален, то это препятствует использованию закрытого ключа со стороны злоумышленника, даже если он перехватит парольную фразу [8].

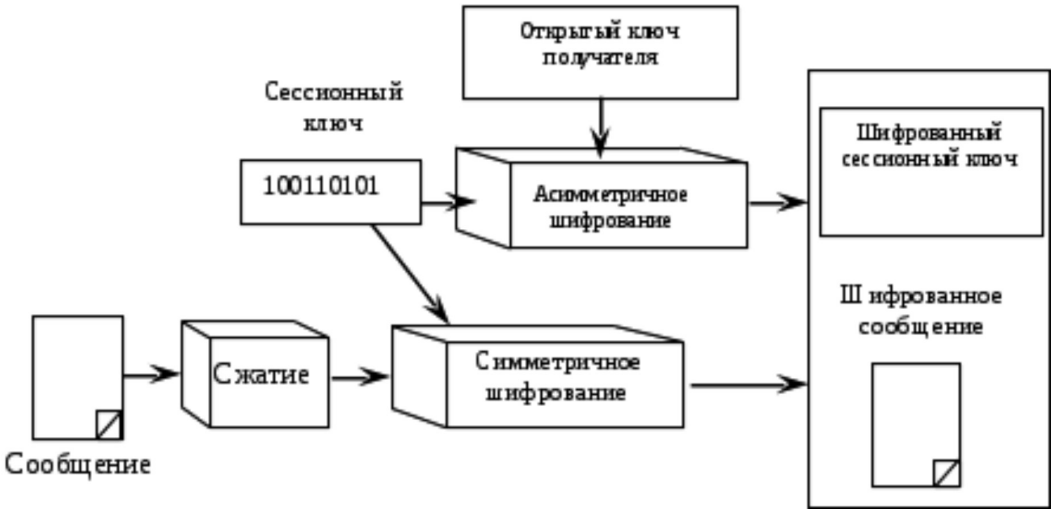

Рис. 3. Алгоритм шифрования PGP

Для использования данной функции, нужно активировать её в файле настроек сервера, или опцией --otp при запуске сервера. После этого нужно вызвать сервер с опцией --gen-otp и указать, какое число одноразовых паролей требуется сгенерировать. Чем длиннее будет список, тем реже его надо будет обновлять, но и тем больше операций сможет совершить наш противник (злоумышленник), если этот список окажется скомпрометирован. В любой момент можно сгенерировать новый список паролей; тогда все пароли, которые оставались в предыдущем списке, будут аннулированы; после генерации нового списка перезапуск сервера не требуется.

Если функция OTP включена, то при каждом запросе парольной фразы необходимо будет ввести также и одноразовый пароль – его нужно дописать непосредственно к концу парольной фразы (не отделяя пробелом или какими-либо иными знаками). Введённый пароль тут же будет аннулирован: так, если пользователь допустит опечатку в парольной фразе закрытого ключа, то при следующей попытке её ввода потребуется ввести уже следующий одноразовый пароль. Когда список паролей окажется пуст, какие-либо операции с закрытыми ключами станут недоступны, пока не будет сгенерирован новый список.

Белый список

Несмотря на то, что ввод дополнительного одноразового пароля увеличивает уровень безопасности соединения, тем не менее аутентификация проходит вне этого приложения. Для того чтобы исключить любое внешнее воздействие, необходимо создать «*белый список*» опций. Для него создается отдельный файл whitelist.conf, который должен располагаться в одном каталоге с конфигурационным файлом. Его формат достаточно прост и создается со следующими особенностями:

1. Строки, не начинающиеся со знака тире, игнорируются.

2. Каждая строка содержит один набор опций.

3. Набор может быть представлен одной опцией в короткой (с одним тире) или в длинной форме (с двумя тире), либо разделёнными пробелом несколькими опциями в короткой или длинной форме (в произвольном порядке).

4. Если набор содержит слова, не начинающиеся с тире, то они имеют следующее значение:

(a) Слово в квадратных скобках является произвольным параметром – опции в данном наборе считаются параметризуемыми параметрами.

(b) Слово без квадратных скобок является разрешённым значением параметра – опции в данном наборе считаются параметризуемыми параметрами и могут принимать параметр только в указанном значении. Если требуется разрешить несколько значений параметра, то их необходимо привести в этой же строке через пробел (поддерживаются кавычки и экранирование пробелов).

(c) Слово #NO_FILES в квадратных скобках устанавливает для данного набора опций флаг «без файлов» – в этом случае программа не будет рассматривать переданные аргументы командной строки в качестве имён файлов.

Заключение

На сегодняшний день ясно и понятно, что добросовестный пользователь банковских, платежных и коммерческих систем практически незащищен от возможных атак злоумышленников. Известна схема, по которой возможна утечка данных, что и происходит на практике.

Однако есть способ защитить добросовестных пользователей платёжных систем (банковских, коммерческих и др. систем, осуществляющих денежные операции) с помощью предложенного авторами программного приложения.

Предположим, что после работы добросовестный пользователь встречается с друзьями в кафе, подключается к общественной Wi-Fi сети и совершает покупки со своей банковской карты либо электронного кошелька. Спустя некоторое время с его счета исчезают все деньги. Но этого не произойдёт, если он будет использовать предложенное авторами приложение, так как в этом приложении обеспечивается несколько уровней безопасности: в начале сессии – ввод одноразового пароля, который хранится на внешнем устройстве и не сохраняется в кэшированных данных; затем – создание «*белого списка*», который обеспечит доступ к сессии только настоящих пользователей с проверенными адресами.

Список литературы

1. Федеральный закон от 27 июня 2011 г. № 161-ФЗ «О национальной платежной системе» (с изм. и доп. на 27.06.2018) [Электронный ресурс]. URL: http://www.consultant.ru/cons/cgi/online.cgi?req=doc&base=LAW&n=115625&dst=100004#07070985803449295 (дата обращения: 25.05.2019).

2. Порублева Е.С., Гареева Г.А., Григорьева Д.Р. Электронные деньги и электронные платежные сервисы в России XXI века // Символ науки. 2018. № 1–2. С. 89–92.

3. Мусалаева С. А. Электронные деньги и платежные системы // Практическая силовая электроника. 2010. № 4. С. 206–208.

4. Комаров А. Современные методы аутентификации: токен и это все о нем! // T-Comm – Телекоммуникации и транспорт. 2008. № 6. С. 24–28.

5. Matthew Daniel Green. Cryptography for Secure and Private Databases: Enabling Practical Data-base Access without Compromising Privacy: dissertation. Baltimore. 2009. [Electronic resource]. URL: https://isi.jhu.edu/~mgreen/MatthewGreenPhDThesis.pdf (date of access: 25.05.2019).

6. A bad couple of years for the cryptographic token industry. [Electronic resource]. URL: https://blog.cryptographyengineering.com/2012/06/21/ bad-couple-of-years-for-cryptographic (date of access: 25.05.2019).

7. Ateniese G., Fu K., Green M., Hohenberger S. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. ACM Transactions on Information and System Security (TISSEC). 2006. Vol. 9. Is. 1.

8. Standage T. The Victorian Internet: the Remarkable Story of the Telegraph and the Nineteenth Century's On – line Pioneers (Pbk. ed.). N.Y.: Walker, 2007. P. 119.