УДК 004.056.53 DOI: 10.26102/2310-6018/2020.28.1.033

Классификация и сравнительный анализ технологий многофакторной аутентификации в Веб-приложениях

Д.С. Богданов, С.Г. Клюев
Федеральное государственное казенное образовательное учреждение высшего образования «Краснодарский университет Министерства внутренних дел Российской Федерации»,
Краснодар, Российская Федерация

Резюме: Актуальность исследования обусловлена растущими темпами внедрения механизмов многофакторной аутентификации в веб-приложения, популяризацией веб-технологий, а также отсутствием в Российской Федерации конкретных стандартов, описывающих работу технологий многофакторной аутентификации и устанавливающих требования к веб-приложениям, которые используют данную технологию. В работе рассматриваются существующие технологии и протоколы аутентификации как в общем виде, так и в части касающейся аутентификации в веб-приложениях, рассмотрена последовательная работа протоколов стандарта HTTP 1.0, 1.1, отмечены их достоинства и недостатки. Были рассмотрены и соотнесены с существующими протоколами аутентификации комбинации факторов аутентификации, применяемых при разработке механизмов многофакторной аутентификации. На основе проведенного исследования была предложена классификация технологий многофакторной аутентификации в веб-приложениях. Цель данного исследования – общий анализ способов аутентификации, применяемых в веб-приложениях, сравнительный анализ протоколов аутентификации в веб-приложениях, классификация технологий многофакторной аутентификации с целью выделения наиболее значимых параметров для систем и протоколов аутентификации с последующим определением рациональности использования многофакторной аутентификации в том или ином веб-приложении. Материалы работы представляют теоретическую ценность для дальнейших исследований в данной области.

Ключевые слова: многофакторная аутентификация, веб-приложение, двухфакторная аутентификация, классификация, протоколы аутентификации.

Для цитирования: Богданов Д.С., Клюев С.Г. Классификация и сравнительный анализ технологий многофакторной аутентификации в Веб-приложениях. *Моделирование, оптимизация и информационные технологии.* 2020;8(1). Доступно по: https://moit.vivt.ru/wp-content/uploads/2020/02/BogdanovKluev_1_20_1.pdf DOI: 10.26102/2310-6018/2020.28.1.033

Classification and comparative analysis of technologies of multifactor authentication in Web applications

D.S. Bogdanov, S.G. Klyuev
Krasnodar University of the Ministry of Internal Affairs
Krasnodar, Russian Federation

Abstract: The relevance of the study is due to the growing pace of the introduction of multi-factor authentication mechanisms in web applications, the popularization of web technologies, as well as the lack of specific standards in the Russian Federation that describe the operation of multi-factor authentication technologies and establish requirements for web applications that use this technology. The work discusses existing authentication technologies and protocols both in a general form and in terms of authentication in web applications, considers the sequential operation of HTTP 1.0, 1.1 standard

protocols, and their advantages and disadvantages are noted. Combinations of authentication factors used in the development of multi-factor authentication mechanisms were considered and correlated with existing authentication protocols. Based on the study, a classification of multifactor authentication technologies in web applications was proposed. The purpose of this study is a general analysis of authentication methods used in web applications, a comparative analysis of authentication protocols in web applications, classification of multifactor authentication technologies in order to highlight the most significant parameters for authentication systems and protocols, and then determine the rationality of using multifactor authentication in one or another web application. The materials of the work are of theoretical value for further research in this area.

Keywords: multifactor authentication, web application, two-factor authentication, classification, authentication protocols.

For citation: Bogdanov D.S., Klyuev S.G. Classification and comparative analysis of technologies of multifactor authentication in Web applications. *Modeling, Optimization and Information Technology*. 2020;8(1). Available from: https://moit.vivt.ru/wp-content/uploads/2020/02/BogdanovKluev_1_20_1.pdf DOI: 10.26102/2310-6018/2020.28.1.033(In Russ).

Введение

Высокая популяризация сети Интернет, рост вычислительных мощностей персональных компьютеров, активное использование веб-технологий – это безусловно позитивные факторы, которые демонстрируют рост информационных технологий в целом, но, с другой стороны, подобные тенденции также приводят к возникновению новых, ранее не известных уязвимостей в информационных системах. Высокая популярность сети Интернет и все вытекающие из этого последствия, включая стабильную и бесперебойную работу важных для государства сервисов, которые были призваны заменить материальные носители информации и ускорить работу определенных организаций могут быть подвергнуты атакам злоумышленников, которые имеют достаточную для проведения подобного рода атак квалификацию. Рост вычислительных мощностей компьютеров так же имеет свою негативную сторону – использование мощностей процессора и видеокарты для проведения атак на хэш-суммы паролей, которые могли быть получены, например, вследствие недостаточно качественно организованного процесса аутентификации администраторами или программистами информационной системы. Внедрение веб-приложений в повседневную жизнь пользователей также несет в себе угрозу безопасности информации, учитывая значительное количество паттернов, библиотек и технологий, на основе которых происходит развертывание подобных приложений. Их количество и сложность реализации также несет в себе уязвимости, о которых могут не знать даже сами разработчики. Одной из ключевых особенностей современных веб-приложений выступает система разграничения доступа веб-приложения, которой всегда предшествует тот или иной механизм аутентификации, и учитывая современные потребности одного фактора аутентификации в некоторых ситуациях и при определенных условиях может быть недостаточно. Исходя из сложившихся условий в отдельные категории сервисов на данный момент интегрированы механизмы двухфакторной аутентификации, которые направлены на повышение надежности систем безопасности этих сервисов. В дан ной работе будут рассмотрены существующие способы и протоколы, аутентификации, которые применяются в веб-приложениях, варианты реализации механизмов многофакторной аутентификации, а также произведена их классификация с целью проведения дальнейших исследований для разработки научно-методического аппарата сравнения эффективности функционирования технологий многофакторной аутентификации в веб-приложениях.

Материалы и методы

Аутентификация в веб-приложении – процедура проверки сервером подлинности данных, предоставляемых клиентом. Процедура аутентификации является одним из составных компонентов процесса проверки аутентичности, предоставленных данных и делегирования соответствующих полномочий клиенту сервером. Логическая последовательность рассматриваемого процесса выглядит следующим образом:

1. Идентификация в веб-приложении – процедура предварительного заявления клиентом о его намерении получить соответствующие права на доступ к веб-приложению, посредством предоставления серверу проверки уникальных идентификаторов.
2. Аутентификация в веб-приложении – процедура проверки сервером данных, которые предоставил клиент. Как правило, осуществляется путем сопоставления данных, предварительно внесенных в базу сервера аутентификации, которые предоставил клиент.
3. Авторизация в веб-приложении – процедура присвоения сервером клиенту соответствующих прав, в случае успешной проверки аутентичности предоставленных данных.

Субъектами процедуры аутентификации в веб-приложениях могут выступать: пользователь, процесс и иные сведения, обладающие высокой степенью значимости для архитектуры функционирования веб-приложения, подлинность которых необходимо подтверждать.

На текущий момент существует большое количество механизмов аутентификации, которые повсеместно используются для проверки подлинности субъектов аутентификации при обмене данными в веб-приложениях. Практически все существующие механизмы аутентификации построены на проверке факторов, присущих субъекту аутентификации:

- фактор знания;
- фактор владения;
- фактор свойства.

Использование одного из вышеперечисленных факторов долгое время предоставляло администраторам веб-приложений достаточную гарантию того, что процедуру авторизации пройдет легитимный пользователь.

Сервер аутентификации – выделенный или специализированный компьютер с предустановленным специальным программным обеспечением, позволяющим принимать и обрабатывать аутентификационные запросы.

Протокол аутентификации – криптографический протокол, в ходе которого одна сторона удостоверяется в идентичности другой стороны, вовлеченной в протокол, а также убеждается в том, что вторая сторона активна во время или непосредственно перед моментом выполнения протокола [1].

Существует два основных способа организации работы протоколов аутентификации – односторонняя и двусторонняя. Односторонняя аутентификация предполагает проверку только одной стороны, например, аутентификация клиента сервером, тогда как протоколы двусторонней аутентификации позволяют производить проверку подлинности обеих сторон.

При описании утверждений для протоколов аутентификации будут использованы следующие обозначения для участников процедуры: претендент (**P**), проверяющий (**V**) и третья сторона (**M**). В общем виде требования к протоколу аутентификации выражаются следующими утверждениями [1]:

1. Если оба участника аутентификации P и V являются легитимными, то протокол будет успешно завершен, путем принятия претендента P проверяющей стороной V.
2. Проверяющая сторона V не имеет возможности повторного использования протокола с целью имперсонификации P, при проверке третьей стороны M.
3. Вероятность успешного завершения протокола аутентификации проверяющей стороной V при попытке подделать аутентификационные данные легитимного претендента P третьей стороной M пренебрежимо мала.
4. Предыдущие свойства остаются справедливыми, даже если между P и V совершено большое, но полиномиально ограниченное число сеансов рассматриваемого протокола аутентификации, а третья сторона M участвовала в предыдущих сеансах выполнения протокола, и некоторое количество сеансов могли выполнятся параллельно.

Классической формой построения протоколов аутентификации на основе фактора знания в веб-приложениях является использование логинов и паролей. Такая форма подразумевает предоставление клиентом пары логин/пароль для прохождения процедуры авторизации. При этом, на стороне сервера существует база данных, которая хранит записи о предварительно зарегистрированных пользователях, что позволяет серверу произвести проверку предоставленных клиентом данных. Простым примером протокола, основанным на факторе знания, используемом в веб-приложениях, является протокол «HTTP authentication», работа протокола описана стандартами HTTP 1.0, 1.1. Основная идея заключается в добавлении данных аутентификации в заголовок HTTP-запроса, которые сервер может идентифицировать. При первой попытке клиента обращения к защищенному веб-приложению веб-сервер отсылает сообщение «401 Unauthorized», и автоматически добавляет HTTP заголовок «WWW-Authenticate», содержащий в себе инструкции аутентификации. Клиенту для получения доступа к веб-странице необходимо подтвердить свою аутентичность, заполнив форму ввода логина и пароля, которую предварительно сгенерирует сервер, например, средствами языка программирования PHP. Стоит отметить, что в данном протоколе существует возможность использования как пары логин/пароль, так и одного пароля, без предоставления логина. После отправки запроса авторизации клиентом, сервер проверяет данные, указанные в форме, и принимает решение о предоставлении доступа к защищаемому ресурсу. При повторных запросах авторизованного клиента к веб-ресурсу аутентификация не требуется. На данный момент существует три основных схемы HTTP аутентификации [4]:

1. **Basic**. Является простейшей формой HTTP аутентификации, в данном случае значения *username* и *password* передаются в открытом виде, в кодировке «base64-encoded» (Рисунок 1). На первый взгляд может показаться, что использование такой схемы аутентификации является небезопасным решением, однако, исходя из того, что на текущий момент времени большая часть веб-приложений использует протокол защиты транспортного уровня TLS и передает весь трафик по защищенному протоколу HTTPS в зашифрованном виде, можно сделать вывод о возможности применения данной схемы в отдельных случаях, не предполагающих высоких требований к обеспечению безопасности информации.

Рисунок 1 - Поэтапная схема работы Basic HTTP аутентификации
Figure 1 - Basic HTTP authentication step-by-step scheme

2. **Digest**. Данная схема HTTP аутентификации является более усложненной версией Basic варианта. Веб-сервер передает клиенту параметр *nonce*, клиент, получив этот параметр применяет к нему функцию хеширования «md5» следующим образом: $responce = md5(username : nonce : password)$, где username и password – данные, вводимые пользователем, после чего клиент отправляет ответ серверу, который записывает полученный ответ в переменную *$_SERVER* (Рисунок 2). Преимуществом данной схемы относительно Basic является использование хеширования, вместо кодирования «base64-encoded», что повышает безопасность передаваемых данных аутентификации. Стоит отметить, что на сегодняшний день алгоритм хеширования «md5» является устаревшим и не отвечает современным требованиям безопасности.

3. **Forms authentication**. Протокол, который активно используется в системах аутентификации различного рода веб-приложений. Не имеет четкого описания как стандарт. Большинство форм реализаций данного протокола уникальны и специфичны для конкретных систем. Для работы протокола необходима HTML форма, сервер-обработчик и база данных. Клиентские идентификационные данные помещаются в HTML форму, после чего отправляются на сервер POST запросом. Сервер идентифицирует клиента, выполняет процедуру аутентификации, сверяя отправленный клиентом пароль с паролем, хранимым в базе данных, который соответствует предоставленному идентификатору клиента. В случае успешной аутентификации сервер присваивает клиенту уникальный *session id*, который сохраняет в свою временную базу данных. Клиент записывает полученный от сервера *session id* в *browser cookies*, что позволяет сохранить сессию на установленный сервером промежуток времени и не проходить процедуру аутентификации повторно. Стоит отметить, что сессия остается активной до тех пор, пока сервер хранит во временной базе данные *session id*. Необходимо учитывать, что разработчик может реализовать передачу формы методом *GET*, что подразумевает отображение в адресной строке всех передаваемых данных формы, включая пароль в открытом виде.

Все вышеперечисленные протоколы аутентификации, используемые в веб-приложениях стандарта HTTP основаны на факторе знания и представляют собой механизмы однофакторной аутентификации. Рассмотренные протоколы аутентификации в классическом исполнении с учетом современных тенденций информатизации являются недостаточно надежными на сегодняшний день. Главным достоинством использования данных протоколов является простота их реализации и минимальные требования как к программным, так и к аппаратным ресурсам. К недостаткам их применения можно отнести:

– применение кодировки base-64 в протоколе Basic;

– использование устаревшего алгоритма хеширования «md5» в протоколе Digest;

– отсутствие возможности двусторонней аутентификации субъектов в рассмотренных протоколах.

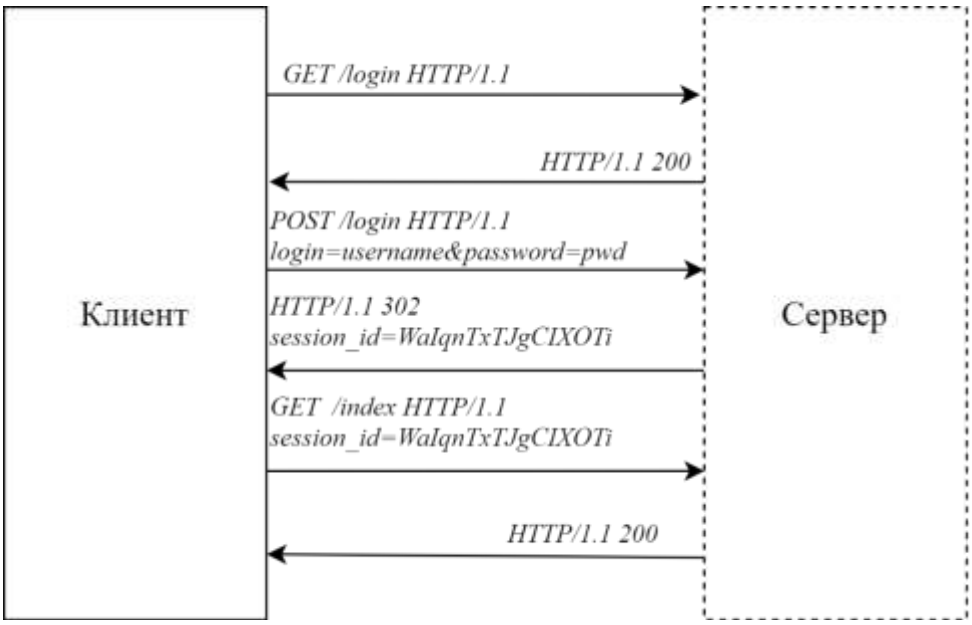

Рисунок 2 - Поэтапная схема работ Forms HTTP аутентификации
Figure 2 - Step-by-step forms HTTP authentication workflow

Использование протоколов многофакторной аутентификации было вызвано множеством причин, например, такими как повышение вычислительных мощностей среднестатистических компьютеров, разработка эффективных алгоритмов перебора паролей, сложность и несовершенство современных систем авторизации.

Механизмы многофакторной аутентификации – механизмы установления подлинности объектов и/или субъектов, в которых используется, по крайней мере, два независимых фактора аутентификации [2].

Примерами реализации механизмов многофакторной аутентификации могут выступать следующие веб-приложения и ресурсы:

1. **Яндекс. Паспорт**. Разработчиками из Яндекс была реализована удобная схема односторонней двухфакторной аутентификации, которая исключает из механизма фактор знания и максимально проста с точки зрения обычного пользователя (Рисунок 4). Рассматриваемая схема многофакторной аутентификации производит проверку двух факторов, которые предоставляет пользователь системы: фактор свойства и фактор владения.

Два эти фактора связаны между собой устройством пользователя (смартфоном), на который предварительно установлено приложение, которое генерирует и хранит временные пароли длинной 8 символов, срок действия которых составляет 30 секунд, при этом использование статического пароля, установленного пользователем при регистрации, для авторизации не предусмотрено. Большинство современных смартфонов поддерживают функцию считывания отпечатка пальца и возможность фото- и видеосъемки, что было использовано для реализации фактора свойства в данном механизме.

Фактором свойства является отпечаток пальца пользователя, который необходимо предоставить для получения доступа к приложению, а фактором владения выступает сам смартфон, который при помощи встроенной камеры считывает QR-код, который генерирует веб-страница. QR-код хранит в себе в себе номер сессии (Рисунок 3), в момент сканирования его приложением номер с выработанным одноразовым паролем и именем пользователя передается на сервер-обработчик. В структуре веб-страницы, на которой отображен QR-код работает JavaScript, который ожидает со стороны сервера-обработчика ответа на проверку аутентификационных данных, предоставленных мобильным приложением.

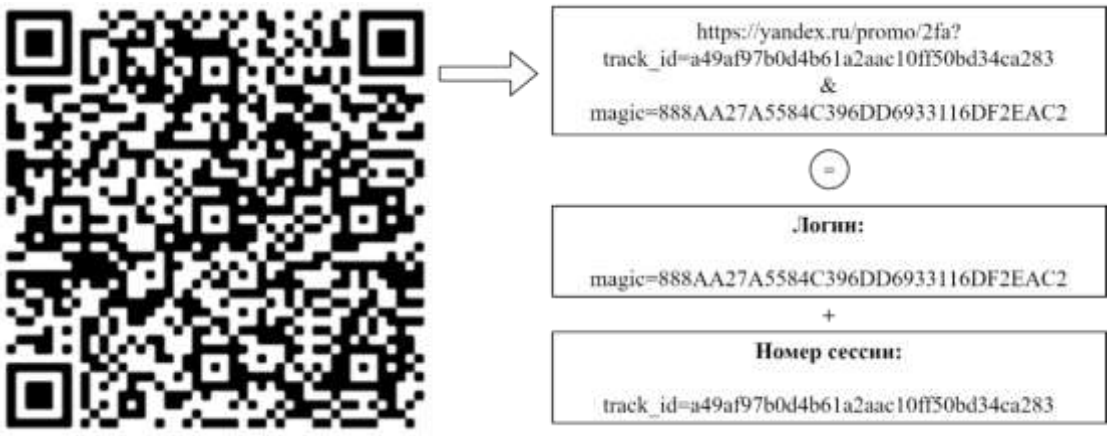

Рисунок 3 - Структура GET-запроса, содержащегося в QR-коде
Figure 3 - The structure of the GET request contained in the QR code

В случает успешной проверки полученных данных сервер-обработчик формирует ответ и устанавливает сессионные *cookie*, с этого момента пользователь считается аутентифицированным [3].

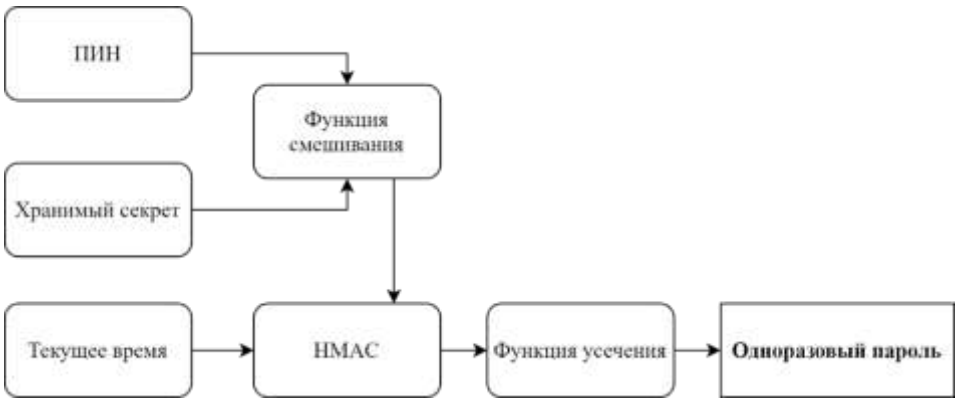

Рисунок 4 - Схема работы двухфакторной аутентификации Яндекс. Паспорт
Figure 4 - The two-factor authentication scheme of Yandex. Passport

Для пользователей, устройства которых не имеют возможности считывания отпечатка пальца, предусмотрена комбинация из факторов: знание и владение, где фактором владения является одноразовый пароль, генерируемый приложением, а фактором знания – вводимый пин-код для получения доступа к приложению.

2. **Google Authenticator**. Данный механизм односторонней аутентификации был разработан Google, изначально имел открытый исходный код на GitHub, но на данный момент является частной собственностью Google [5]. Для работы механизма используется мобильный телефон с предустановленным приложением и сервер-обработчик, необходимый для подтверждения аутентичности вводимых пользователем данных. Преимуществом данного механизма аутентификации является его доступность и возможность использования любыми сторонними веб-приложениями, чего нельзя сказать о двухфакторной аутентификации Яндекс, которая работает исключительно с сервисами самого Яндекс. Механизм работы основан на факторах знания и владения. Фактором знания является логин и пароль пользователя на любом из сервисов, который использует Google Authenticator, а фактором владения – устройство (смартфон), который генерирует одноразовые пароли длиной 6 символов со сроком действия 30 секунд. Google Authenticator имеет множество реализаций для серверов-обработчиков на разных языках программирования, которые размещены в открытом доступе [6-7].

Любой этап механизма многофакторной аутентификации образуется путем комбинации факторов аутентификации с учетом последовательности их проверки (Таблица 1).

Таблица 1 - Комбинации факторов аутентификации
Table 1 - Authentication factor combinations

Тип фактора аутентификации		**А**	**Б**	**В**
		Знание	Владение	Свойство
1	Знание	-	1Б	1В
2	Владение	2А	-	2В
3	Свойство	3А	3Б	-

Исходя из потребностей информационной системы разработчиком могут быть реализованы различные комбинации, последовательности и количество используемых факторов.

Результаты

Промежуточным результатом проведенного исследования является сравнительный анализ механизмов аутентификации в веб-приложениях (Таблица 2). За основу сравнительного анализа были взяты наиболее значимые свойства и механизмы технологий аутентификации в веб-приложениях.

Таблица 2 - Сравнение механизмов аутентификации в веб-приложениях
Table 2 - Comparison of authentication mechanisms in web applications

Свойство / Механизм	HTTP Basic	HTTP Digest	HTTP Form	Яндекс П.	Google А.
Комбинация факторов	1/А	1/А	1/А	3Б	1Б
Односторонняя аутентификация	+	+	+	+	+
Двусторонняя аутентификация	-	-	-	-	-
Кодирование аутентификационных данных	+	-	-	-	-

Хеширование аутентификационных данных	-	+	+	+	+
Сервер аутентификации на стороне сервера обработчика	+	+	+	-	-
Обособленный сервер аутентификации	-	-	+/-	+	+
Независимый сервер аутентификации	-	-	-	-	+/-
Последовательные многофакторные механизмы	-	-	-	+	+
Параллельные многофакторные механизмы	-	-	-	-	-
Комбинированные многофакторные механизмы	-	-	-	-	-

Проведя анализ существующих способов аутентификации в веб-приложениях, была предложена классификация технологий многофакторной аутентификации (Рисунок 5) по количеству субъектов аутентификации, по способу преобразования аутентификационных данных, по типу сервера аутентификации.

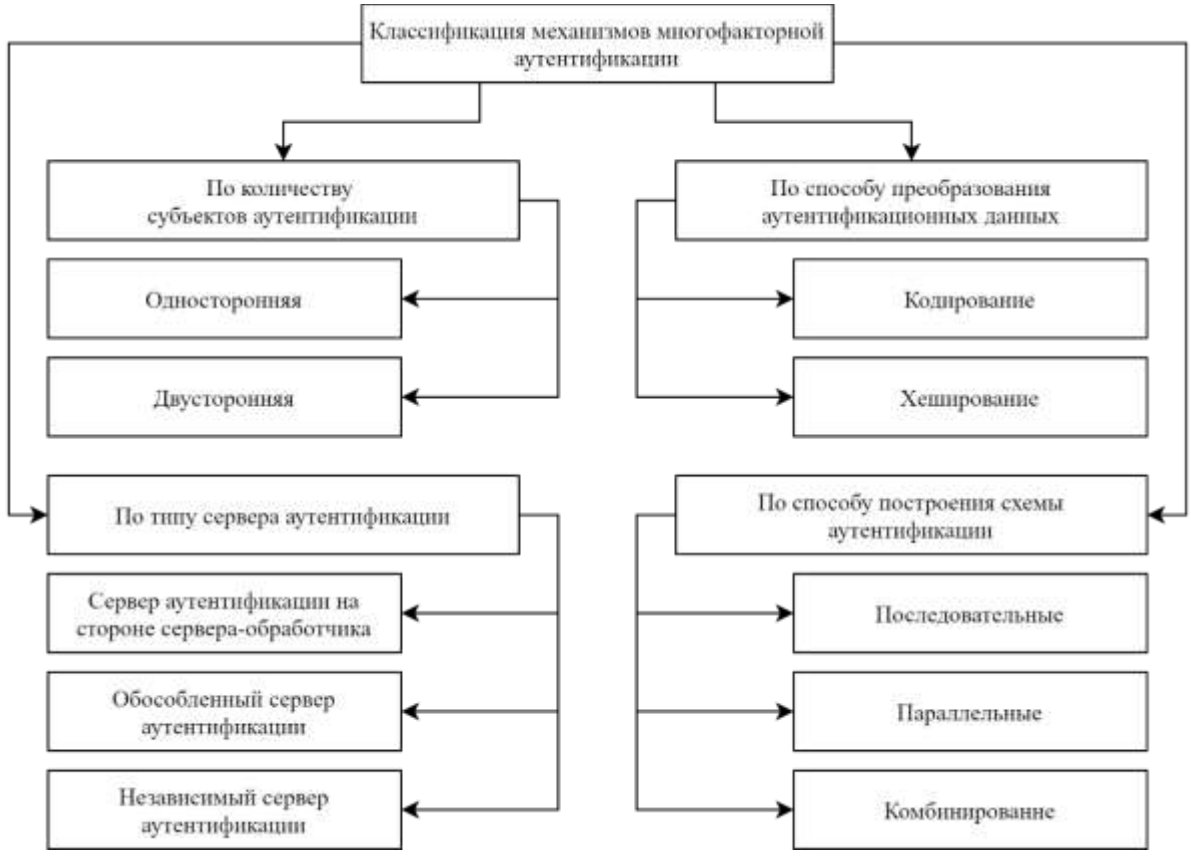

Рисунок 5 - Классификация технологий многофакторной аутентификации в веб-приложениях
Figure 5 - Classification of multifactor authentication technologies in web applications

1. По количеству субъектов аутентификации:
   – *Односторонняя*. Используется для проверки аутентичности клиента.

- *Двусторонняя*. Включает механизм проверки аутентичности клиента и сервера.

2. По способу преобразования аутентификационных данных:
   - *Кодирование*. Для сокрытия передаваемых данных используются кодировки, например, «base-64».
   - *Хеширование.* В рассмотренных протоколах Digest и Form аутентификации стандарта HTTP применятся алгоритм хеширования «md5».
3. По типу сервера аутентификации:
   - *Сервер аутентификации* на стороне сервера-обработчика. Случай, когда сервер аутентификации физически размещен на оборудовании веб-сервера.
   - *Обособленный сервер* аутентификации. Сервер аутентификации размещен на оборудовании не связанным с веб-сервером.
   - *Независимый сервер* аутентификации. Сервером аутентификации владеет третье лицо, никак не связанное с владельцем веб-приложения.
4. По способу построения схемы аутентификации:
   - *Последовательные*. Под последовательными схемами понимаются схемы, при котором предоставление и проверка аутентификационных данных выполняется с определенной очередью, например, изначально проверяется фактор знания, предположим, логина и пароля. В случае успешного прохождения пользователем первого фактора, необходимо подтвердить свою аутентичность еще одним, при этом такой порядок должен быть строго соблюден механизмом аутентификации.
   - *Параллельные*. В параллельных механизмах аутентификации пользователь должен предъявить единовременно сразу два или более факторов. При этом отправка аутентификационных данных на сервер аутентификации должна быть реализована одним сообщением.
   - *Комбинированные.* Механизмы аутентификации, предусматривающие комбинированное применение последовательных и параллельных схем.

Обсуждение

Стоит отметить, что были рассмотрены базовые протоколы HTTP аутентификации в веб-приложениях и наиболее качественно реализованные на сегодняшний день механизмы многофакторной аутентификации. В соответствии с проведенным анализом можно сделать следующие выводы:

1. Двусторонняя аутентификация не поддерживается существующими протоколами стандарта HTTP 1.0, 1.1.
2. Протокол HTTP Basic в его классической реализации не имеет возможности хеширования аутентификационных данных.
3. Из рассмотренных механизмов аутентификации только Google Authenticator имеет возможность подключения к механизму сторонних сервисов и соответственно, сервис, использующий этот механизм будет иметь частично независимый сервер аутентификации.
4. Рассмотренные механизмы многофакторной аутентификации имеют последовательную реализацию.

Выделенные классификацией параметры механизмов многофакторной аутентификации могут быть использованы в дальнейших исследованиях для определения критериев эффективности механизмов многофакторной аутентификации, построения эффективных механизмов и целесообразности их использования в том или ином веб-приложении.

Заключение

В работе проведен сравнительный анализ протоколов аутентификации в веб-приложениях стандартов HTTP 1.0, 1.1, отмечены положительные и отрицательные стороны использования данных протоколов. Были рассмотрены распространенные механизмы многофакторной аутентификации в веб-приложениях, на основе полученных сведений и результатов других исследований была проведена классификация механизмов многофакторной аутентификации.

Результаты проведенного анализа и предложенную классификацию возможно использовать для выделения наиболее значимых параметров для различного рода информационных систем при разработке механизмов многофакторной аутентификации, для определения рациональности использования многофакторных механизмов.

Требуется проведение дальнейших исследований в данной области в целях разработки научно-методического аппарата сравнения эффективности функционирования технологий многофакторной аутентификации в веб-приложениях.

ЛИТЕРАТУРА

1. Тарасов А.А., Казарин О.В., Запечников С.В. *Криптографические Методы Защиты Информации*. 1-е изд. Москва: Издательство Юрайт; 2019. (Высшее образование).
2. Горбенко Ю.И., Олешко И.В. Модели и методы оценки защищенности механизмов многофакторной аутентификации. *Восточно-Европейский Журнал Передовых Технологий*. 2013;6(2):4-10.
3. Двухфакторная аутентификация, которой удобно пользоваться [Интернет]. 2015. Доступно по адресу: https://habr.com/ru/company/yandex/blog/249547 (дата обращения 06.02.2020 г.).
4. Обзор способов и протоколов аутентификации в веб-приложениях [Интернет]. 2015. Доступно по адресу: https://habr.com/ru/company/dataart/blog/262817 (дата обращения 07.02.2020 г.).
5. Willis N. FreeOTP multi-factor authentication [LWN.net] [Интернет]. 2014. Доступно по адресу: https://lwn.net/Articles/581086/ (дата обращения 07.02.2020 г.).
6. Stocker C. GoogleAuthenticator for PHP [Интернет]. 2019. Доступно по адресу: https://github.com/chregu/GoogleAuthenticator.php (дата обращения 07.02.2020 г.).
7. Jaskowski T. GoogleAuthenticator for Python [Интернет]. 2019. Доступно по адресу: https://github.com/tadeck/onetimepass (дата обращения 07.02.2020 г.).

REFERENCES

1. Tarasov A.A., Kazarin O.V., Zapnechnikov S.V. *Cryptographic Information Protection Methods*. 1st ed. Moscow: Yurayt Publishing House; 2019. 309 p. (Higher education).
2. Gorbenko Y.I., Oleshko I.V. Models and methods for assessing the security of multi-factor authentication mechanisms. *East European Journal of Advanced Technology*. 2013;6(2):4-10.
3. Two-factor authentication, which is convenient to use [Internet]. 2015. Available at: https://habr.com/en/company/yandex/blog/249547(accessed 06.02.2020).
4. An overview of authentication methods and protocols in web applications [Internet]. 2015. Available at: https://habr.com/ru/company/dataart/blog/262817 (accessed 07.02.2020).
5. Willis N. FreeOTP multi-factor authentication [LWN.net] [Internet]. 2014. Available at: https://lwn.net/Articles/581086/ (accessed 07.02.2020).

6. Stocker C. Google Authenticator for PHP [Internet]. 2019. Available at: https://github.com/chregu/GoogleAuthenticator.php (accessed 07.02.2020).
7. Jaskowski T. Google Authenticator for Python [Internet]. 2019. Available at: https://github.com/tadeck/onetimepass (accessed 07.02.2020).

ИНОРМАЦИЯ ОБ АВТОРЕ / INFORMATION ABOUT THE AUTHOR

Богданов Дмитрий Сергеевич, преподаватель, кафедра информационной безопасности, ФГКОУ ВО "Краснодарский университет Министерства внутренних дел Российской Федерации", Краснодар, Российская Федерация.
e-mail: bdkrdu@yandex.ru
ORCID: 0000-0002-6523-7725

Dmitriy S. Bogdanov, Lecturer, Department of Information Security, Federal State-Funded Educational Institution of Higher Education "Krasnodar University of the Ministry of Internal Affairs of the Russian Federation", Krasnodar, Russian Federation.

Клюев Станислав Геннадьевич, кандидат технических наук, доцент, кафедра информационной безопасности, ФГКОУ ВО "Краснодарское высшее военное училище", Краснодар, Российская Федерация.
e-mail: s.g.klyuev@mail.ru

Stanislav G. Klyuev, Cand. Sci. (Technical), Associate Professor, Department of Information Security, Federal State-Funded Educational Institution of Higher Education "Krasnodar Higher Military School", Krasnodar, Russian Federation.