DOI 10.37882/2223–2966.2023.01.28

ИНФРАСТРУКТУРА JSON WEB TOKEN. ИНФРАСТРУКТУРА ЗАЩИТЫ

JSON WEB TOKEN INFRASTRUCTURE. SECURITY INFRASTRUCTURE

M. Monakhov
A. Uymin

Summary. The article considers the security issues of JSON Web Token-based authorization. There is a possible risk of an attacker realizing the threat. This article also considers organization of authentication based on JWT and organization of token storage. When used as additional means of protection, the authors stipulate the organization of a secure encrypted channel to the server infrastructure in order to eliminate the possibility of interception of open (unencrypted) traffic, as well as session tracking.

Keywords: token, jwt, user, web, key.

Монахов Михаил Юрьевич
Доктор технических наук, профессор, Владимирский государственный университет им. А.Г. и Н.Г. Столетовых, кафедра информатики и защиты информации; заведующий кафедрой, г. Владимир
mmonakhov@vlsu.ru
Уймин Антон Григорьевич
Старший преподаватель, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва
au-mail@ya.ru

Аннотация. В статье рассмотрены вопросы безопасности авторизации на основе JSON Web Token. Возможен риск реализации угрозы злоумышленником. Также в этой статье рассмотрена организация аутентификации на базе JWT и организация хранения маркеров. При использовании в качестве дополнительных средств защиты, авторы предусматривают организацию защищенного шифрованного канала до инфраструктуры сервера с целью исключения возможности перехвата открытого (не зашифрованного) трафика, а также отслеживания сессии.

Ключевые слова: токен, jwt, пользователь, веб, ключ.

Введение

Авторизация, одна из самых сложных задач при удаленной работе пользователя [1]. Наиболее распространенный функционал в использовании JSON Web Token (JWT) [2]. Как только клиент входит на сайт, каждый следующий шаг включает JWT и предоставляет возможность клиенту получить доступ к инфраструктуре управления активами или ресурсам в системе посредством переданного токена. В настоящее время JWT широко используется в различных областях благодаря его небольшим расходам (техническим и инфраструктурным) и возможности удобного использования в различных сферах (разработанные API и библиотеки) [3].

Целью исследования является описание принципов работы JSON Web Token. Реализация системы защиты процедуры Аутентификации посредством инструментов JSON Web Token, на примере проекта Remoute Topology

Применяя JSON в качестве основы веб-токена, можно передавать данные без риска их потери. Поэтому, если JWT могут быть подтверждены — например с помощью наборов открытых / закрытых ключей, можно с высокой долей уверенности утверждать, что отправитель является тем человеком, который легитимно осуществляет передачу. В то же время, поскольку метка определяется с использованием заголовков и полезной нагрузки вы также можете подтвердить факт изменения структуры и данных, что позволит определить атаки на целостность информации [4].

Каждая секция веб-токена JSON разделена на три части, каждая из которых полностью независима [5].

- Текст заголовка
- Полезная нагрузка
- Подпись

Заголовок обычно содержит две части. Тип токена (JWT) и используемую технологию подписания, такую как HMACSHA256 или RSA;

Example:
{
“alg”: “HS256”,
“typ”: “JWT”
}

Уникальный идентификатор пользователя, предоставляет токен с датой окончания срока его действия

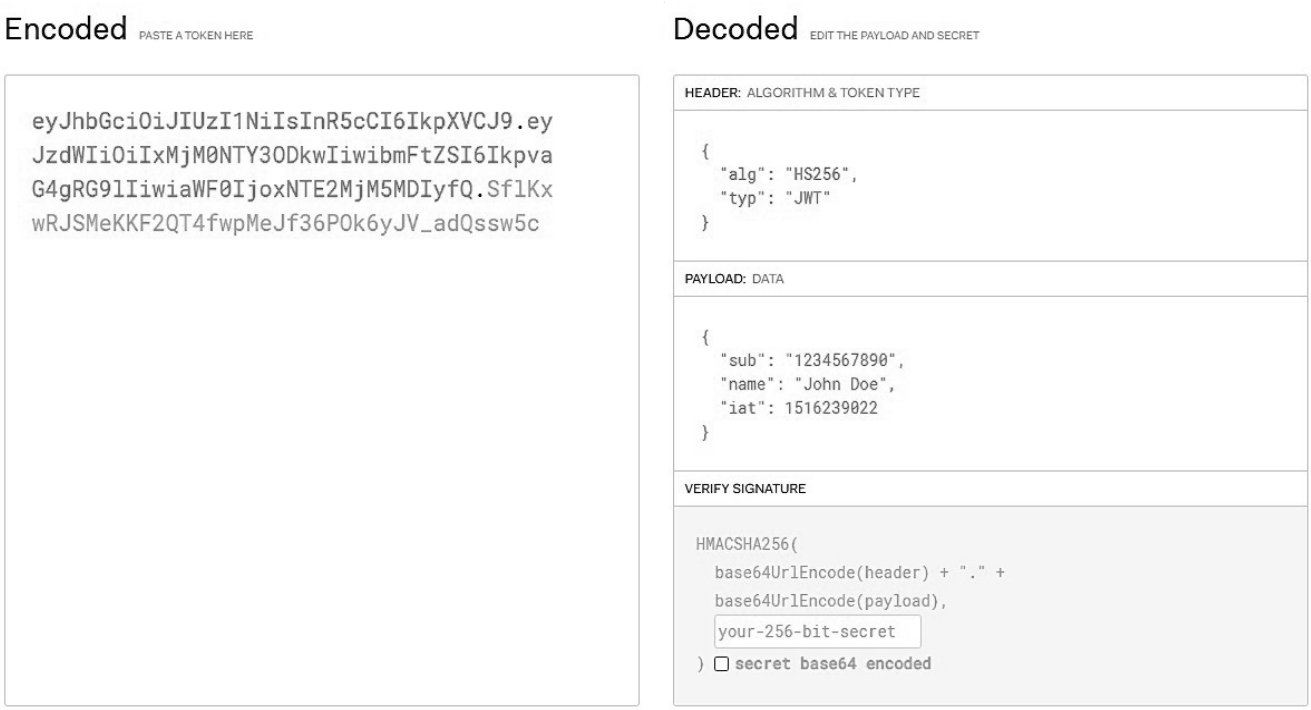

Рис. 1. Расшифровка веб-токена JSON

и правом доступа к нему. В JSON для полезной нагрузки используются следующие:

{
«sub»:»0987654301»,
«name»: «sangi»,
«iat»: 8550022720
}

Он объединяет заголовки, помимо этого, кодирует их с использованием вывода base64, затем он использует набор правил HMAC256 для кодирования данных [6].

Код, создающий метку, выглядит следующим образом

HMACSHA256
(base64Url,
Encode(header) + “.”
+base64Url,
Encode(payload),
)
secret base64 encoded

Поэтому в качестве идентификатора конвейеров используется субъективная строка, которая может использоваться для получения авторизации. Также возможно использование токена JWT, который является подтверждением для использования в качестве токена конвейера. Как и в случае с другими системами идентификации личности (ID), токен конвейер — это просто субъективный текст, который используется для авторизации. При этом токен должен быть применен только в том случае, когда авторизация осуществляется с помощью JWT.

Симметричная подпись

В рамках обмена, подписи должны быть симметричными, то есть они должны быть основаны на одном и том же ключевом слове для подтверждения и создания подписей с использованием графического изображения HMAC [7]. В большинстве случаев симметричные подписи правильно расположены и могут быть реализованы в отдельном программном обеспечении. Не секретная (публичная) подпись используется для подтверждения, а личная подпись (секретная) — для маркировки.

Заголовок + полезная нагрузка + подписи (см. Рисунок 1)

Псевдокод JWT

Token=f(Base64Encode) Σ (header. payload. Signature)

Этап 1: кодирование типа токена (jwt) и вычисление, используемое в качестве кода для возраста JWT (здесь hmac SHA 256) с кодом base64. Он формирует сегмент,

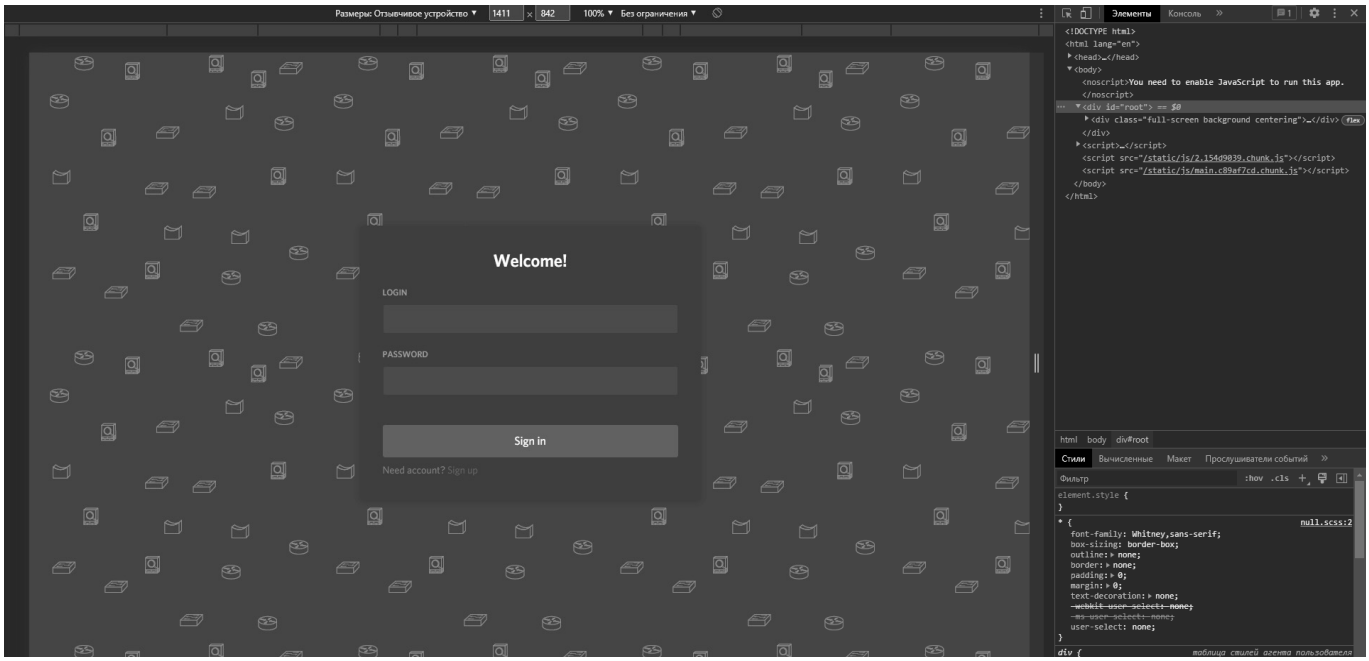

Рис. 2. Изображение проекта RT

в котором происходит формирование начального сегмента токена.

Этап 2: с помощью Base64encode можно кодировать полезную нагрузку. В состав полезной нагрузки входят особенности клиента. Он структурирует вторую часть токена.

Этап 3: создается хэш заголовка и полезной нагрузки. Для этого используют расчет HMAC.

Этап 4: в настоящее время полученный хэш снова шифруется посредством SHA256, а также скрытый ключ для формирования последней части подписи, закодированной с помощью base64. Формирует третью часть токена.

Этап 5: все три зашифрованные части соединены между собой и изолированы точками. На основе этой базы создается база JWT, которая имеет длину 304 байта.

В действительности токены JWT не зашифрованы, они просто записаны в базу Base64encode. JWT — это совокупность всех «индивидуальных особенностей» клиента, используемых для проверки и подтверждений. Аналогично, у нас нет предпочтений в том, чтобы максимально эффективно раскрыть все «индивидуальных особенностей» внутри токена для каждого пользователя. Поэтому в них добавлена часть encode с целью их открепить, от пользователя. У человека, имеющего тайный ключ, есть возможность контролировать все нюансы передачи. JWT имеет три основных раздела: заголовок, полезную нагрузку и подпись. Это необходимо для того чтобы понять каким образом можно использовать эти вычисления при разработке и возможных атаках.

Данный код содержит различные вычисления шифрования, которые используются в blends [8]. Их использование заключается в применении хэш-мощностей или вычислений хэша, симметричных вычислений и неправильных вычислений — все это используется совместно для повышения защищенности системы. Используют две вычислительные операции, которые основаны на использовании одного вычисления, как правило, вычисление хэш-кода или ключа шифрования, для информации о заголовке и полезной нагрузке, сохраненной отдельно от информации, изменяющейся позднее вместе с ключом шифрования, который является секретным ключом, эти хэшированные части. В соответствии с этим JWT использует два вычисления в blend, хэш + симметричный алгоритм, например: SHA-256, который использует вычисление хэша HMAC для хэширования заголовка и части полезной нагрузки, позже он использует SHA-256 для хэширования последней группы с секретным ключом, используемым для цифровой подписи. При использовании этого метода, ключ используется для скремблирования и создания цифровой подписи [9]. При этом, если клиент хочет получить доступ к системе, он должен пройти регистрацию или передать запрос на вход в систему обработчику. Секретный ключ в качестве закрытого ключа использу-

ется для подписания токена, после чего создается JWT и отправляется клиенту. После изготовления JWT его нужно будет отправить куда-нибудь, чтобы клиент мог добавить запрос с токеном при каждом входе в систему. В результате, мощность (степень секретности хранения) переходит от хранения договора к соседней мощности. Предположим, что веб-маркер JWT хранится в соседнем хранилище. При необходимости получить определенную информацию из информационной базы с помощью любого HTTP-запроса или, по сути, интерфейса программирования клиент должен прикрепить токен как маркер носителя и отослать его обработчику. Если авторизация была успешно пройдена, он сообщает об этом клиенту.

При регистрации на сайте пользователь вводит всю информацию: имя пользователя, электронная почта и так далее. В проекте RT предложен следующий интерфейс [10] (см. Рисунок 2)

Позже будет введен пароль. Как только пользователь входит на сайт, все данные пользователя аутентифицируются и проверяются. После проверки генерируется токен JWT и передается пользователю. Для того чтобы воспользоваться любыми функциями, пользователь должен отправить JWT в заголовок HTTP и отправить его на сервер программного обеспечения. На сервере проверяется код, который был получен от пользователя. В случае подтверждения он может получить право доступа к запрашиваемым фактам, в результате чего статистика возвращается пользователю.

JWT.io — это сайт, который обеспечивает интерпретацию токена JWT. Он осуществляет проверку заголовка и часть полезной нагрузки токена JWT. В то время как маркированная часть JWT-токена не раскрывается. Чтобы узнать метку и изменить суть JWT-токена, необходим секретный ключ. Сайт JWT.io помогает нам проверить, является ли полученный токен JWT или нет, а также проверить, является ли токен просроченным или действующим. JWT.io анализирует токен: разделяет его на заголовок, полезную нагрузку и подпись, считая полные остановки разделенной частью. Затем, каждый момент, он берет отдельный из всех ротделенных элементов и разворачивает с помощью декодера Base64, и показывает конечные результаты в правой половине веб-страницы, каждый в отдельности. Компонент подписи остается в стороне и запрашивает секретный ключ, который является основным элементом защиты токена JWT [11].

В настоящее время разработчики прилагают усилия для создания приятных для человека приложений и повышают эргономику их использования. Для достижения этой цели используются стеки для создания программ в ограниченные временные интервалы. Одним из таких стеков является стек JavaScript, называемый стеком MERN [12]. MERN помогает разработчикам создавать эффективные сетевые программы в короткие сроки, используя только знание JavaScript. Большим преимуществом стека JavaScript является легкая интеграция и эффективное тестирование. С ростом потребностей в Интернете спрос на веб-приложения значительно увеличился. Веб-сайты, которые раньше представляли собой комбинацию HTML, CSS, PHP или сложного JavaScript, теперь уже не отвечают современным требованиям. Веб-сайты переходят в разряд веб-приложений, в которых задействованы высоко динамичные данные. С ростом спроса на динамические данные вырос и спрос на высокий уровень пользовательского опыта. Таким образом, разработчики стремятся сделать новые веб-приложения более интеллектуальными, быстрыми и эффективными, что негативно сказывается на защищенности решений. Чтобы обеспечить высокую эффективность и масштабируемость веб-приложений, разработчики сегодня используют набор технологий, позволяющих сделать все возможное. Этот набор технологий в совокупности называется стеком. Исходя из текущих потребностей пользователей, инженеры-программисты используют стековую веб-разработку, при которой они разрабатывают веб-приложения на основе уже существующих фреймворков (например, JavaScript framework). Два популярных фреймворка развились из JavaScript и являются наиболее востребованными — MEAN (Components include Mongo DB, Angular JS, Express, and Node) и MERN (Components include Mongo DB, React JS, Express, and Node). Оба стека созданы из компонентов с открытым исходным кодом и предоставляют сквозную основу для создания динамических комплексных веб-приложений, которые позволяют браузерам подключаться к базе данных. Два важных преимущества использования стека:

- Путаницы в кодировании можно избежать, просто кодируя на одном языке.
- Гибкость может быть развита во всех веб-приложениях, разработанных с использованием стека.

При реализации проекта Remoute Topology, были применены следующие компоненты системы:

- PostgreSQL — это объектно-реляционная система управления базами данных (ОРСУБД, ORDBMS), основанная на POSTGRES,
- Axios — это HTTP-клиент с открытым исходным кодом, который основан на Promise для node js и браузера. Он изоморфный (= он может работать в браузере и node.js с той же базой кодов). На сервере он использует нативный node.js http-модуль, а на стороне клиента (браузер) он используется XMLHttpRequests.
- React появился в системе Facebook, в рекламном агентстве Facebook. Изначально разработчики

в facebook использовали стандартную клиентскую модель MVC, но в ней были все данные как для шаблонов, так и для представлений. Представления — это те, которые реагируют на изменения в моделях, просто изменяя их размер. По мере увеличения сложности, в приложение вносились новые изменения. Поскольку в коде обновления будут тонкие различия в зависимости от причины обновления, каскадные обновления сложно поддерживать.

- Node.js — это событийно-управляемая реализация JavaScript, предназначенная для реализации масштабируемых сетевых приложений. Node.js использует событийно-асинхронную модель, которая не исключает модель I / O, что делает ее эффективной, простой и высокопроизводительной.

Браузер отображает учетные данные для входа в систему во время регистрации посредством HTTP-запроса к веб-серверу.

На стороне сервера генерируется секретный ключ, используя хэш пароль и строку ноль или ASCII значение «48».

На сервере генерируется JSON веб-токен с учетом секретного ключа.

Сгенерированный токен отправляется в браузер для последующих запросов.

Теперь вместе с каждым HTTP-запросом в HTTP-заголовке отправляется JSON web-токен для авторизации и аутентификации пользователя браузером.

Сервер рассматривает маркер и проверяет, разрешен ли пользователю доступ к запрашиваемым данным.

Если аутентификация и доступ разрешены, браузер получает нужные данные в качестве ответа по протоколу HTTP.

Таким образом, можно отметить, что реализация JSON Web Token в рамках проекта Remoute Topology является стандартной для данного класса программного обеспечения, и предположить, что она подвержена основным типам уязвимостей (атак) рассматриваемых для JWT.

Использование инструмента wireshark [13] позволяет захватить и анализировать сетевой трафик, который содержит момент авторизации пользователя. При изучении запроса «http» обнаруживаем в нем cookie, а внутри «tokenAccess». На сегодняшний день веб-токены JSON (JWT) используются для обеспечения безопасности и аутентификации пользователей на сайтах, API. Далее, после копирования и проверки полученного JWT (см. Рисунок 1) определяем, что используется симметричное шифрование HS256. Также расшифровка дает нам Username человека, который находится в payload.

Примеры атак на данный тип шифрования. можно найти в базе уязвимостей. Согласно базе CVE в настоящее время к актуальным угрозам можно отнести следующие: CVE-2015–2951, CVE-2015–9235, CVE-2016–5431, CVE-2016–10555, CVE-2018–0114, CVE-2018–1000531, CVE-2019–20933, CVE-2019–7644, CVE-2020–28042

Реализация которых возможна, как из-за ошибок разработчиков, так и в виду примирения специализированных эксплойтов, входящих в большинство стандартных баз. Данные проблемы особенно остро встают при рассмотрении вопросов разработки отечественных программных продуктов, на базе международных проектов, с учетом возможностей атак на программный код отдельной страны и сложности верификации программного кода сот международного сообщества [14]. Все данные уязвимости можно определить как инфраструктурные, т.е. при корректной разработке и сопровождении программного обеспечения атака на данные уязвимости не принесет успеха. Сложность при работе состоит в подходах к моделям угроз и применении парадигм инженерного мышления и формальной логики [15]

Киберпреступность сегодня отдельный вектор бизнеса. Постоянное усовершенствование методов защиты становится необходимым сопровождением ИТ инфраструктуры. Атаки «грубой силы», фишинг, перехват трафика и т.д. показывают, что применение только базового инструмента, основанного на аутентификации — недостаточно. Аутентификация, на основе маркеров, в сочетании с дополнительными методами аутентификации создает только базовую защиту. Системы авторизации на базе маркеров, считаются достаточно безопасными, но не смотря на все их преимущества, всегда существует риск реализации угрозы злоумышленником. Поэтому в данной статье рассмотрен вопрос организации аутентификации на базе JWT, организации хранения и передачи маркера. Сделан акцент на возможности перехвата трафика, что повышает риски потери информации и получения доступа злоумышленником. В качестве дополнительных средств защиты, авторы предусматривают организацию защищенного шифрованного канала до инфраструктуры сервера, с целью исключения возможности перехвата открытого (не зашифрованного), трафика, а также отслеживания сессий с возможностью дополнительной идентификации пользователей.

ЛИТЕРАТУРА

1. Setiawan E.R. Implementasi Authentication & Authorization Berbasis JWT Pada Sistem Pengelolaan Perkuliahan Menggunakan Algoritma HMAC: дис.— Universitas Muhammadiyah Ponorogo, 2022.
2. Jones M., Campbell B., Mortimore C. Json web token (jwt) profile for oauth 2.0 client authentication and authorization grants.— 2015.— № . rfc7523.
3. Dalimunthe S., Reza J., Marzuki A. The Model for Storing Tokens in Local Storage (Cookies) Using JSON Web Token (JWT) with HMAC (Hash-based Message Authentication Code) in E-Learning Systems //Journal of Applied Engineering and Technological Science (JAETS).— 2022.— Т. 3.— № . 2.— С. 149–155.
4. Bulgakova O. et al. Risk of Information Loss Using JWT Token.— 2021.
5. Dalimunthe S., Reza J., Marzuki A. The Model for Storing Tokens in Local Storage (Cookies) Using JSON Web Token (JWT) with HMAC (Hash-based Message Authentication Code) in E-Learning Systems //Journal of Applied Engineering and Technological Science (JAETS).— 2022.— Т. 3.— № . 2.— С. 149–155.
6. Alkhulaifi A., El-Alfy E. S.M. Exploring Lattice-based Post-Quantum Signature for JWT Authentication: Review and Case Study //2020 IEEE91st Vehicular Technology Conference (VTC2020-Spring).— IEEE, 2020.— С. 1–5.
7. Allali H. Authentication Model Based on JWT and Local PKI for Communication Security in Multi-agent Systems //Innovation in Information Systems and Technologies to Support Learning Research: Proceedings of EMENA-ISTL 2019.— 2019.— Т. 7.— С. 469.
8. Wu Y. et al. One-pot Synthesis of Multifunctional KGM/PDA/PVDF Composite Membrane for Efficient Treatment of Oil–water Emulsion and Dye //Nano.— 2021.— Т. 16.— № . 03.— С. 2150025.
9. Sabir B.E. et al. Authentication Model Based on JWT and Local PKI for Communication Security in Multi-agent Systems //International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning.— Springer, Cham, 2019.— С. 469–479.
10. Свидетельство о государственной регистрации программы для ЭВМ № 2021619990 Российская Федерация. RemoteTopology-модуль авторизации: № 2021613424: заявл. 09.03.2021: опубл. 21.06.2021 / А.Г. Уймин, С.В. Любкин.— EDN KEDGKG.
11. Курьянович Д.Ю. Аутентификация при помощи JsonWebToken (JWT). Структура JWT.— 2020.
12. Dhivya K.D.R., Sangeetha N. Json Web Token Used in MERN Stack for Makinge-Commerce Web-Application //Journal homepage: www. ijrpr. com ISSN.— Т. 2582.— С. 7421.
13. Pramukantoro E.S., Bakhtiar F.A. Cloud-based Middleware for Syntactical Interoperability in Internet of Things //Journal of Information Technology and Computer Science.— 2020.— Т. 5.— № . 1.— С. 32–37.
14. Уймин, А.Г. Инструментальные средства обучения компьютерным сетям. Развёртывание на базе российского программного обеспечения / А.Г. Уймин, Г.И. Токарев // Системы управления и информационные технологии.— 2022.— № 4(90).— С. 88–92.— DOI 10.36622/VSTU.2022.90.4.019.
15. Уймина, О.И. Стереотипы как границы креативного инженерного мышления / О.И. Уймина // Инженерное мышление: социальные перспективы: материалы международной междисциплинарной конференции, Екатеринбург, 12–13 февраля 2020 года / Уральский федеральный университет имени первого Президента России Б.Н. Ельцина.— Екатеринбург: Общество с ограниченной ответственностью «Издательство «Деловая книга», 2020.— С. 197–202

Журнал «Современная наука: актуальные проблемы теории и практики»