ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК 004.056.5 https://doi.org/10.51301/vest.su.2021.i4.06

Е.Г. Совет*, М.М. Коккоз
Карагандинский Технический Университет, Караганда, Казахстан
*e-mail: s_erkemai@mail.ru

ЗАЩИТА ДАННЫХ ВЕБ-ПРИЛОЖЕНИЙ ОТ ВНУТРЕННИХ УГРОЗ

Аннотация. С быстрым развитием Интернета веб-приложения становятся все более популярными. Многие люди, группы, организации или правительства используют веб-приложения как средство для обмена информацией или поддержки бизнес-задач. В связи с ростом числа угроз и атак на веб-приложения организациям требуется более эффективная концепция безопасности веб-приложений.

Статья посвящена безопасности веб-приложений и методам обеспечения безопасности на трехуровневой архитектуре защиты данных. Трёхуровневая архитектура - архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений и сервера баз данных. Безопасность веб-приложений - это защита конфиденциальности, целостности и доступности веб-ресурсов организации, а также ее репутации. Она также включает в себя политику, процедуры, законы, людей и практику. Безопасность, как и другие компоненты веб-приложения, лучше всего управлять, если она запланирована на начальном этапе разработки приложения.

Также, в статье были описаны методы обеспечения безопасности. Эти методы помогут специалистам по безопасности разработать политику безопасности, провести оценку рисков и устранить потенциальные риски экономически эффективным способом.

Ключевые слова: веб-приложение, угрозы, уязвимости, безопасность веб-приложений, конфиденциальность, трехуровневая архитектура, веб-сайт.

Введение. В целом безопасность касается конфиденциальности, целостности и доступности систем и данных [1]. Конфиденциальность - это способность гарантировать, что информация является частной для уполномоченных сторон и защищена от несанкционированного раскрытия. Целостность отражает точность информации и требует технологии и процессов, которые предотвращают несанкционированные стороны от ненадлежащего изменения информации. Доступность означает способность обеспечить своевременный доступ к информации для ее конечных пользователей в целях удовлетворения потребностей миссии. В контексте веб-приложения безопасность - это защита конфиденциальности, целостности и доступности веб-ресурсов организации (например, веб-страниц и баз данных клиентов). В частности, это процесс обеспечения того, чтобы, во-первых, данные (например, системная информация и данные клиентов) являются конфиденциальными для уполномоченных сторон, когда они хранятся на хостах или находятся в пути, во-вторых, данные защищены от случайного или злонамеренного изменения, когда они отображаются на веб-сайте или передаются через Интернет, и в-третьих, веб-сайт продолжает функционировать для законных пользователей в целях удовлетворения требований миссии.

Большинство веб-сайтов построены по трехуровневой архитектуре, представленной на рисунке. В данной архитектуре пользователь интернет-сайта обращается к серверу интернет-сайта, который в свою очередь обрабатывает запрос пользователя и при необходимости

посылает запрос в базу данных [4]. Основная часть систем управления базами данных работает на языкеструктурированных запросов SQL (Structured Query Language), который предназначен для управления данными в реляционных базах данных [3]. После обработки запроса от сервера из базы данных поступают необходимые данные по запросу.

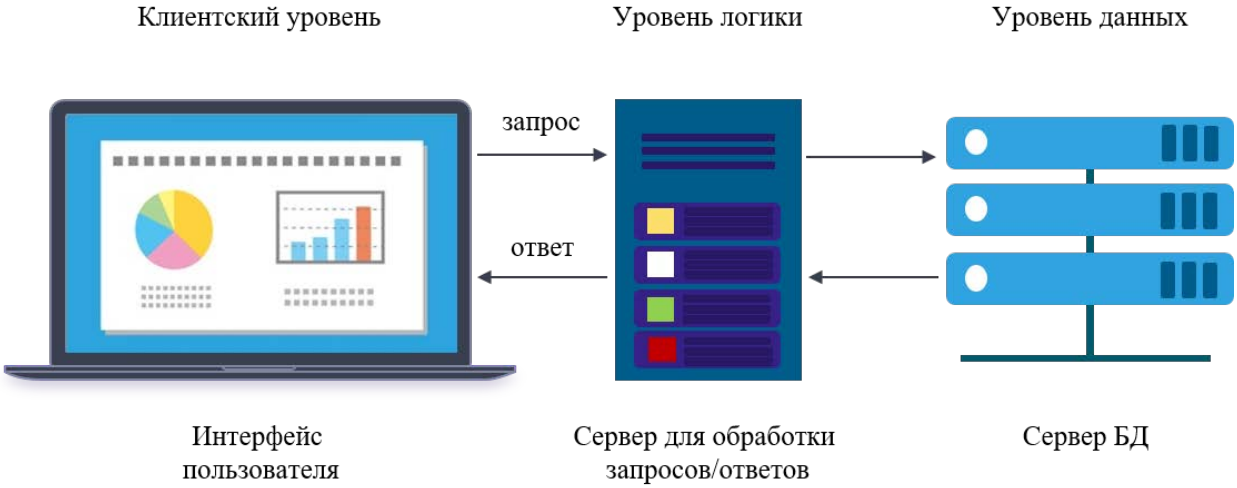

Рисунок. Трехуровневая архитектура построения веб-приложений

Нижний уровень (слой) в трехуровневой архитектуре хранилища данных состоит из репозитория данных. Репозиторий данных - это пространство для хранения данных, извлеченных из различных источников данных, которое подвергается ряду действий в рамках процесса ETL (Extract, Transform, Load — дословно «извлечение, преобразование, загрузка).

Средний уровень - логический уровень извлекается из уровня представления и, как собственный уровень, управляет функциональностью приложения, выполняя подробную обработку.

Верхний уровень - это интерфейсный уровень, то есть пользовательский интерфейс, который позволяет пользователю подключаться к системам баз данных. Этот пользовательский интерфейс обычно представляет собой инструмент или вызов API, который используется для получения необходимых данных для целей отчетности, анализа и интеллектуального анализа данных.

Веб-приложения должны быть защищены на нескольких уровнях архитектуры приложения.

Методы. Существует множество практических проблем безопасности, возникающих при проектировании трехуровневой системы. К ним относятся обеспечение аутентификации пользователей, контроль доступа пользователей, аудит действий пользователей, защита безопасности данных между уровнями, ограничение привилегий среднего уровня, управление удостоверениями между уровнями и создание масштабируемых систем. Для решения этих проблем ниже рассмотрим методы обоспечения безопасности в трехуровневых системах.

Аутентификация пользователя [2]. Трехуровневые архитектуры повышают сложность аутентификации пользователей и контроля доступа. В двухуровневой архитектуре клиент-сервер, где клиентские пользователи подключаются непосредственно к серверу, база данных может аутентифицировать пользователей во время подключения. База данных может связывать любые данные, запрос или транзакцию, которые передаются по соединению пользователя с этим пользователем, может предоставлять или отказывать пользователю в доступе к конфиденциальным ресурсам базы данных и соответственно проверять действия пользователя.

В типичной трехуровневой архитектуре средний уровень отвечает за аутентификацию идентификаторов пользователей. Более того, данные, которые пользователь отправляет на

средний уровень, обрабатываются средним уровнем до того, как они будут отправлены в серверную базу данных в виде запроса, обновления или аналогичной транзакции, и могут иметь мало общего с данными, которые пользователь первоначально отправил на средний уровень. Кроме того, несколько соединений клиент-средний уровень могут быть мультиплексированы через одно соединение средний уровень-база данных. По этим причинам база данных должна делегировать некоторую ответственность за аутентификацию пользователей среднему уровню. База данных также должна полагаться на средний уровень, чтобы правильно связать идентификаторы пользователей с данными, отправленными в базу данных от имени этих пользователей. Очевидно, что разработка стратегии безопасной аутентификации в трехуровневой системе намного сложнее, чем в двухуровневой.

Защита пользовательских данных. Данные, которыми обмениваются уровни, должны быть защищены от непреднамеренного раскрытия или изменения. Шифрование является стандартным механизмом для этой цели. Серверы среднего уровня должны реализовывать протокол шифрования, который поддерживается клиентами, взаимодействующими со средним уровнем; в случае клиентов веб-браузера этот протокол является SSL (Secure Sockets Layer). Средние уровни также должны поддерживать протокол шифрования для связи с базой данных. Помимо защиты данных в сети при обмене между уровнями, протоколы шифрования, такие как SSL, могут обеспечивать криптографическую аутентификацию пользователей.

Обратите внимание, что в трехуровневой архитектуре обычно нежелательно шифровать данные от клиента к базе данных. Сквозное шифрование подразумевает, что данные не расшифровываются на среднем уровне, что не позволяет среднему уровню выполнять значительную обработку данных. Это исключило бы большую часть ценности трехуровневой архитектуры. Однако если данные действительно расшифровываются на среднем уровне, то шифрование клиента аутентифицирует только клиента на среднем уровне, а не базу данных.

Отслеживание действий пользователя. Подотчетность посредством аудита является основным принципом информационной безопасности. Аудит дополняет контроль доступа и помогает гарантировать, что привилегированные пользователи не злоупотребляют своими правами доступа. Трехуровневые системы повышают сложность отслеживания и корреляции действий пользователей, которые могут быть чувствительны к безопасности, и, таким образом, затрудняют аудит безопасности. Даже принятие решения о том, какие события следует проверять, может быть сложной задачей в трехуровневой системе.

В трехуровневой системе пользователи обычно не имеют прямого доступа к базе данных. Вместо этого пользователь выполняет какое-то действие на среднем уровне, и это действие может привести к тому, что средний уровень запросит, чтобы база данных выполнила какое-то соответствующее действие от имени пользователя. Если база данных не регистрирует личность пользователя, от имени которого средний уровень выполнил какое-либо действие в базе данных, системному аудитору может быть трудно соотнести связанные действия, выполненные на среднем уровне и уровне базы данных, и определить, какой пользователь был ответственным.

Ограничение привилегий среднего уровня. Еще одна проблема при разработке безопасной трехуровневой системы заключается в делегировании базы данных соответствующей степени доверия среднему уровню. В некоторых трехуровневых архитектурах база данных позволяет среднему уровню подключаться как любому пользователю и принимать любую роль или привилегию, которые этот пользователь имел бы, если бы он был подключен непосредственно к базе данных. Этот подход может быть приемлемым в некоторых ситуациях, например, когда в базе данных мало или вообще нет конфиденциальных данных или когда пользователи не сильно отличаются по привилегиям. Однако в большинстве случаев желательно, чтобы база данных имела возможность ограничить привилегии среднего уровня, чтобы база данных могла ограничить, может ли конкретный средний уровень действовать от имени конкретного пользователя и может ли он

принимать определенные привилегии пользователя. Серверы среднего уровня, которые не реализуют строгие механизмы аутентификации или которые относительно уязвимы для атак (например, потому что они находятся за пределами корпоративного брандмауэра и подвергаются взлому пользователями, входящими через Интернет), могут, таким образом, иметь меньше привилегий, чем серверы среднего уровня, которые реализуют строгую аутентификацию или находятся внутри корпоративного брандмауэра.

Управление удостоверениями и учетными записями на разных уровнях. Существенной проблемой при построении трехуровневых систем является связывание идентификаторов пользователей среднего уровня с идентификаторами пользователей базы данных. Часто бывает так, что способ представления личности пользователя будет отличаться в различных компонентах трехуровневой системы. Например, пользователи, получающие доступ к системе через веб-сервер, могут аутентифицировать себя, обмениваясь сертификатами X.509 с веб-сервером через протокол Secure Sockets Layer (SSL). Сертификаты X.509 содержат личность пользователя в формате X.500 Distincted Name (DN); например, сертификаты X.509 содержат личность пользователя в формате X. 500 Distincted Name (DN).:

C=«US», O = «Oracle», OU= «Server Technologies», CN=«Mary Ann Smith».

Этот формат совсем не похож на типичное имя пользователя в базе данных, которым может быть MASMITH . Переход от идентификации пользователя среднего уровня к идентификации пользователя базы данных может представлять значительную проблему для проектирования архитектуры безопасности системы.

Связанная с этим проблема заключается в управлении удостоверениями пользователей и учетными записями в трехуровневой системе. При добавлении нового пользователя система, удостоверения, учетные записи и привилегии для этого пользователя должны быть созданы как на среднем уровне, так и в базе данных. Если для аутентификации клиента используются сертификаты X.509, то каждый клиентский пользователь должен запросить и установить сертификат X.509 у центра сертификации (ЦС), признанного средним уровнем. Если идентификаторы представлены по-разному на каждом уровне, то они должны быть каким-то образом коррелированы, чтобы (например) средний уровень мог правильно связать идентификатор X.509 пользователя клиента с идентификатором, связанным с учетной записью этого пользователя на среднем уровне. Средний уровень также должен быть в состоянии перевести идентификатор пользователя в соответствующий идентификатор базы данных, запрашивая выполнение некоторых действий в базе данных от имени пользователя.

Масштабирование до сообществ пользователей интернета. Хотя по соображениям безопасности желательно иметь возможность отслеживать и контролировать действия каждого пользователя как на среднем уровне, так и в базе данных, нежелательно, чтобы каждый пользователь имел отдельную учетную запись и схему пользователя в базе данных. Возможно, практично создавать и управлять многими сотнями или даже тысячами учетных записей пользователей в базе данных, но есть клиенты, которые хотят создавать трехуровневые интернет-приложения, поддерживающие миллионы пользователей. Нецелесообразно создавать и управлять таким количеством традиционных учетных записей пользователей в базе данных. Более того, во многих приложениях информация каждого пользователя состоит из одной строки в таблице, и создание отдельной учетной записи пользователя и схемы для каждого из этих пользователей является чрезмерным подслушиванием. Чтобы поддерживать сообщества пользователей в масштабе Интернета, база данных должна применять некоторые средства для управления доступом пользователей и отслеживания активности пользователей, что не требует от каждого пользователя наличия учетной записи и схемы в базе данных.

Результат. Безопасность веб – приложений - это не разовое усилие. Это должен быть непрерывный процесс, интегрированный в жизненный цикл разработки приложений. Для активной защиты веб-приложений безопасность должна приниматься во внимание на

начальном этапе жизненного цикла приложения. Технология - это не серебряные пули безопасности веб-приложений. Для достижения безопасности веб-приложений в организации должны разработать согласованную корпоративную политику и процедуры с учетом защищаемых веб-активов, характера потенциальных рисков, технологий и процессов, необходимых для минимизации рисков, а также бюджетов. Кроме того, правительство должно обеспечить соблюдение законов, чтобы свести к минимуму случаи преступного использования безопасности.

С технологической точки зрения комплексный технический контроль должен осуществляться во всей инфраструктуре приложения, включая сети, хосты и само приложение. Помимо размещения в защищенных сетях и хостах, веб-приложения должны разрабатываться и разрабатываться с учетом безопасности, включая проверку входных данных, управление сеансами и управление исключениями. Основные механизмы безопасности, такие как аутентификация, авторизация, контроль доступа, аудит и логирование, должны быть реализованы во всей инфраструктуре приложения.

Обсуждение. В ходе проведенного исследования была представлена трехуровневая защита данных в веб-приложениях в котором пользователь через браузер обращается не напрямую к серверу баз данных, а к серверу приложений, от которого уже идет запрос данных к серверу баз данных. Были рассмотрены методы которые решают проблемы связанные с обеспечением аутентификации пользователей, контролью доступа пользователей, аудитом действий пользователей, защитой безопасности данных между уровнями, ограничением привилегий среднего уровня и с управлением удостоверениями между уровнями и создание масштабируемых систем.

ЛИТЕРАТУРА

[1] Бабаш, А.В. Информационная безопасность: Лабораторный практикум / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. - М.: КноРус, 2019. - 7 с.

[2] Бабенко А. А., Безбабнов Д. В., Витенбург Е. А.. Исследование систем управления информационным наполнением web-ресурсов // Актуальные вопросы информационной безопасности регионов в условиях глобализации информационного пространства: материалы IV Всерос. науч.-практ. конф., г. Волгоград, 23–24 апр. 2015 г.; Федер. гос. авт. образоват. уч-реждение высш. проф. образования «Волгогр. гос. ун-т» — Волгоград: Изд-во ВолГУ, 2015 — С. 92–96.

[3] Оладько В. С. Механизмы защиты web-приложений от внедрения вредоносного кода// Новый университет. Серия: Технические науки. 2015. № 3–4 (37–38). С. 64–68

[4] Тузовский, А. Ф. Проектирование и разработка web-приложений:
учебное пособие для академического бакалавриата / А. Ф. Тузовский.М. : Издательство Юрайт, 2018. — 218 с.

[5] Атчисон Ли «Масштабирование приложений. Выращивание сложных систем» Питер, 2018 год, 256 стр.

REFERENCES

[1] Babash, A.V., Informacionnaya bezopasnost': Laboratornyj praktikum [Information security: Laboratory practice] / A.V. Babash, E. K. Baranova, Yu. N. Melnikov. - M.: KnoRus, 2019. - 7 p.

[2] Babenko A. A., Bezbabnov D. V., Vitenburg E. A. Issledovanie sistem upravleniya informacionnym napolneniem web-resursov [Research of information content management systems of web resources] / / Aktual'nye voprosy informacionnoj bezopasnosti regionov v usloviyah globalizacii informacionnogo prostranstva: materialy IV Vseros [Actual issues of information security of regions in the conditions of globalization of the information space: materials of the IV All-Russian Scientific Journal.] - Practical conference, Volgograd, April 23-24, 2015; Federal State Educational Institution. higher education Prof. education "Volgogr. state un-t" - Volgograd: Publishing House of the Volga State University, 2015-p. 92-96.

[3] Oladko V. S. Mekhanizmy zashchity web-prilozhenij ot vnedreniya vredonosnogo koda [Mechanisms for protecting web applications from the introduction of malicious code]// Novyj universitet. Seriya:Tekhnicheskie nauki. [A new university. Series:Technical sciences.] 2015. №. 3-4 (37-38). pp. 64-68.

[4] Tuzovskij, A. F. Proektirovanie i razrabotka web-prilozhenij [Web application design and development]: uchebnoe posobie dlya akademicheskogo bakalavriata / A. F. Tuzovskij.M. : Izdatel'stvo YUrajt, 2018. — 218 p.

[5] Atchison Lee «Masshtabirovanie prilozhenij. Vyrashchivanie slozhnyh sistem» [Scaling applications. Growing complex systems] Peter, 2018, 256 p.

Е.Ғ. Совет*, М.М. Көккөз
Қарағанды техникалық университеті, Қарағанды, Қазақстан
*e-mail: s_erkemai@mail.ru

ВЕБ-ҚОСЫМШАЛАРДЫҢ ДЕРЕКТЕРІН ІШКІ ҚАУІПТЕРДЕН ҚОРҒАУ

Аңдатпа. Интернеттің қарқынды дамуымен веб-қосымшалар танымал бола бастады. Көптеген адамдар, топтар, ұйымдар немесе үкіметтер веб-қосымшаларды ақпарат алмасу немесе бизнес міндеттерін қолдау құралы ретінде пайдаланады. Веб-қосымшаларға қауіптер мен шабуылдардың көбеюіне байланысты ұйымдар веб-қосымшалардың қауіпсіздігі туралы тиімді тұжырымдаманы қажет етеді.

Мақала үш деңгейлі деректерді қорғау архитектурасындағы веб-қосымшалардың қауіпсіздігі мен қауіпсіздік әдістеріне арналған. Үш деңгейлі сәулет-бағдарламалық кешеннің архитектуралық моделі, онда үш компонент бар: клиент, қосымшалар сервері және мәліметтер базасының сервері. Веб-қосымшалардың қауіпсіздігі - бұл ұйымның веб-ресурстарының құпиялығын, тұтастығын және қол жетімділігін, сондай-ақ оның беделін қорғау. Оған саясат, рәсімдер, заңдар, адамдар және практика кіреді. Қауіпсіздік, веб-қосымшаның басқа компоненттері сияқты, егер ол бағдарламаны әзірлеудің бастапқы кезеңінде жоспарланған болса, жақсы басқарылады.

Сондай-ақ, мақалада қауіпсіздікті қамтамасыз ету әдістері сипатталған. Бұл әдістер қауіпсіздік мамандарына қауіпсіздік саясатын жасауға, тәуекелдерді бағалауға және ықтимал тәуекелдерді экономикалық тиімді түрде жоюға көмектеседі.

Негізгі сөздер: веб-қосымша, қауіптер, осалдықтар, веб-қосымшалардың қауіпсіздігі, құпиялылық, үш деңгейлі архитектура, веб-сайт.

E.G. Sovet*, M.M. Kokkoz
Karaganda Technical University, Karaganda, Kazakhstan
*e-mail: s_erkemai@mail.ru

PROTECTING WEB APPLICATION DATA FROM INTERNAL THREATS

Abstract. With the rapid development of the Internet, web applications are becoming more and more popular. Many people, groups, organizations, or governments use web applications as a means to share information or support business tasks. With the growing number of threats and attacks on web applications, organizations need a more effective concept of web application security.

The article is devoted to the security of web applications and methods of ensuring security on a three-level data protection architecture. A three-level architecture is an architectural model of a software package that assumes that it has three components: a client, an application server, and a database server. Web application security is about protecting the privacy, integrity, and availability of an organization's web resources, as well as its reputation. It also includes policies, procedures, laws, people, and practices. Security, like other components of a web application, is best managed if it is planned at the initial stage of application development.

The article also describes the methods of ensuring security. These methods will help security professionals develop security policies, conduct risk assessments, and eliminate potential risks in a cost-effective way.

Key words: web application, threats, vulnerabilities, web application security, privacy, three-level architecture, website.