УДК 004.056.53
DOI 10.21685/2587-7704-2019-4-1-11

Open Access RESEARCH ARTICLE

Информационная безопасность интернет-сайтов

А. С. Круглов
Пензенский государственный университет, Россия, 440026 г. Пенза, ул. Красная, 40

М. Ю. Лупанов
Пензенский государственный университет, Россия, 440026 г. Пенза, ул. Красная, 40

Аннотация. В данной статье рассмотрены проблемы безопасности интернет-сайтов с динамически формируемым содержимым. Проанализированы основные угрозы информационной безопасности интернет-сайтов как комплексных информационных систем, рассмотрены существующие основные методики защиты. Выявлена и обоснована необходимость создания новых способов разграничения прав доступа к интернет-сайтам ввиду слабого развития данного направления в веб-разработке. На основе проведенного исследования автором предлагаются примеры новых способов разграничения прав доступа к интернет-сайтам, в которых отсутствуют основные уязвимости устаревших способов.

Ключевые слова: информация, безопасность, информационная безопасность, интернет-сайт, веб-сайт, веб-приложение, угрозы, уязвимости, разграничение прав доступа.

Information security of Internet sites

A. S. Kruglov
Penza State University, 40 Krasnaya Street, 440026, Penza, Russia

M. Yu. Lupanov
Penza State University, 40 Krasnaya Street, 440026, Penza, Russia

Abstract. This article considers problems of security of Internet sites with dynamically generated content. The main threats for information security of Internet sites as integrated information systems are analyzed, and the existing basic protection techniques are considered. The need to create new methods for access rights differentiation to Internet sites due to the weak development of this direction in web development is revealed and justified. Based on the study, the author proposes examples of new methods for access rights differentiation to Internet sites that do not contain the main vulnerabilities of obsolete methods.

Keywords: information, security, information security, Internet site, website, web application, threats, vulnerabilities, access rights differentiation.

Большинство веб-сайтов построены по трехуровневой архитектуре, представленной на рис. 1. 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. Основная часть систем управления базами данных работает на языке структурированных запросов SQL, который предназначен для управления данными в реляционных базах данных. После обработки запроса от сервера из базы данных поступают необходимые данные по запросу.

Уровень веб-сервера содержит в себе программное обеспечение, которое обеспечивает работу веб-сайта. Это сам веб-сервер, промежуточный код и его обработчик, протоколы передачи информации, а также некоторые компоненты СУБД. Атаки на данный уровень в настоящее время практически не производятся ввиду отсутствия простых для использования уязвимостей [1], так как основным способом реализации являлось внедрение в незащищенное сетевое соединение по протоколу HTTP

между клиентом и веб-приложением. Это позволяет злоумышленнику перехватить cookie пользователей, что, в свою очередь, давало нарушителю возможность авторизоваться в аккаунт пользователя, чье соединение было атаковано. Но в последнее время большинство веб-приложений перешло на защищенное HTTPS-соединение, не позволяющее выполнять нарушителям перехват данных.

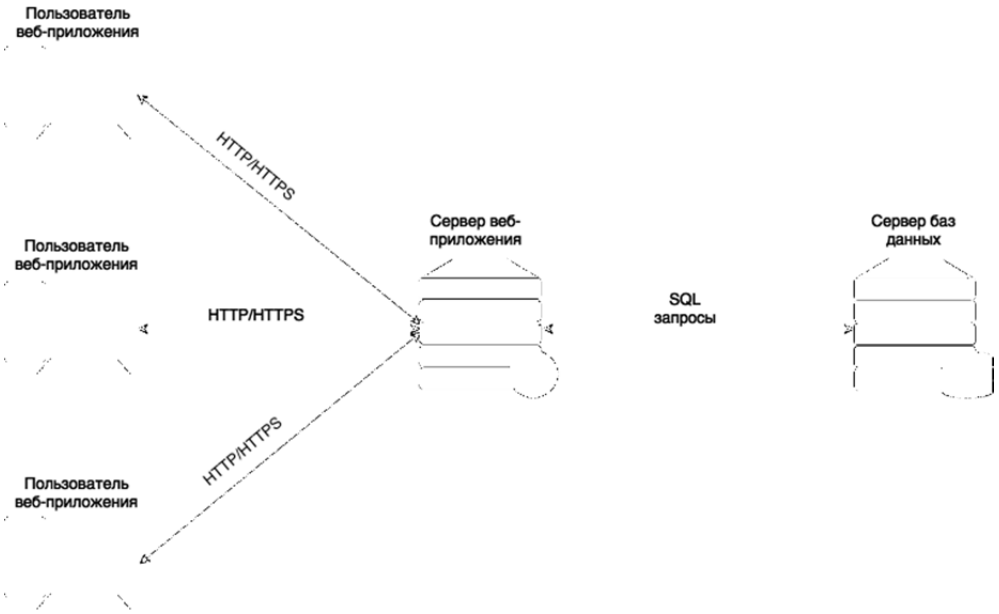

Рис. 1. Трехуровневая архитектура построения веб-приложений

Уровень приложения отвечает за обработку данных веб-приложения, обработку данных, вводимых пользователем, а также за генерацию ответов пользователю по запросу. На данном уровне широко распространены атаки, получившие название «SQL-инъекции». Уязвимость, позволяющая реализовать данную атаку, согласно исследованиям [1], присутствует в 25 % исследованных веб-приложений. Распространение данные атаки получили в силу того, что для их осуществления злоумышленнику нужно иметь только доступ к пользовательскому интерфейсу, т.е. иметь возможность отправлять запросы к базе данных через пользовательский интерфейс.

Суть SQL-инъекций заключается во внедрении в передаваемый запрос произвольного SQL-кода, получении возможности чтения и (или) записи локальных файлов и выполнения произвольных команд на атакуемом сервере и изучении реакции веб-сайта на данный запрос. В случае отсутствия защиты от данного типа атак, злоумышленник имеет возможность выполнять любые действия с базой данных, обработку различных локальных файлов, хранящихся на сервере, а также использовать иные команды на атакуемом сервере.

Клиентский уровень отвечает за интерфейс пользователя, т.е. позволяет пользователю отправлять запросы веб-серверу и получать отображаемые данные. За данный уровень отвечает программное обеспечение – браузер. Данный уровень может быть использован для реализации угроз за счет уязвимостей веб-браузера, подбора паролей пользователей методом «грубой силы», атаки DDoS и т.д.

Уровень представления является структурированной статической либо динамической информацией, полученной от веб-сервера и представленной пользователю в веб-браузере. Угрозу для этого уровня представляют атаки типа Cross-SiteScripting (XSS). Уязвимости, позволяющие реализовать XSS, обычно возникают из-за недостаточной обработки пользовательского ввода, который затем отображается на веб-странице. Они позволяют атакующему внедрить код, исполняемый на стороне клиента в веб-странице, которую просматривает пользователь. Вредоносный код, полученный пользователем, может похищать cookie, перенаправлять пользователя на другие веб-ресурсы, а также выполнять иные вредоносные действия. Данный тип уязвимости является преобладающим в настоящее время [1], он был обнаружен в 80 % исследованных сайтов.

Для большинства из описанных выше угроз и уязвимостей постоянно предлагаются новые способы защиты и противодействия. Но проблема защиты архитектуры «клиент – промежуточный код – СУБД» затрагивается нечасто и решается в основном методами фильтрации, которые требуют постоянного мониторинга и обновления. Основная часть разработчиков в вопросе доступа к СУБД использует следующее решение: заполняется и хранится один единственный конфигурационный файл, в

котором записаны данные для авторизации одного пользователя с правами администратора с полным доступом. Абсолютно все пользователи при работе с веб-сайтом авторизуются в СУБД этим администратором. Права каждого из них хранят в отдельной таблице ролей пользователей. При проверке наличия доступа к тому или иному функционалу у пользователя проверяется именно наличие необходимой роли у этого пользователя в таблице ролей.

Таким образом, все реквизиты доступа к системе управления базой данных хранятся в открытом виде в промежуточном коде веб-сайта. То есть, получив доступ к промежуточному коду, злоумышленник имеет возможность получить доступ к реквизитам доступа к СУБД, а следовательно, и к самой СУБД.

Так как от любого пользователя веб-сайта аутентификация в СУБД, при необходимости обращения к ней, осуществляется от имени администратора с полным доступом, т.е. никак не ограничивается, нарушитель при отсутствии защиты от SQL-инъекций имеет возможность отправлять к СУБД любые SQL-команды без ограничений.

Основным в силу простоты и доступности для реализации способом атак на веб-сайты является эксплуатация SQL-инъекций. Защита от SQL-инъекций в настоящее время заключается лишь в правильном составлении промежуточного кода, обрабатывающего запросы пользователей. Для защиты от SQL-инъекций на уровне промежуточного кода существуют различные фильтры, так называемые «белые» списки идентификаторов (имен полей и таблиц) и ключевых слов, которые жестко прописывают все варианты выбора для баз данных, и в запрос к базе данных после фильтрации должны поступать только они. Также для защиты возможно использование так называемых плейсхолдеров, т.е. пользователю предоставляется уже подготовленное выражение – шаблон запроса к базе данных, в который ему необходимо лишь подставить нужный параметр – плейсхолдер. При этом параметр, вводимый пользователем, и сам подготовленный запрос поступают на сервер отдельно, что исключает возможность злонамеренных действий со стороны пользователя.

Межсайтовый скриптинг позволяет злоумышленнику включать свой HTML-код в веб-станицу. Наиболее уязвимы для такого вида атак являются гостевые книги и форумы, где происходит динамическое формирование страниц. Возможности кода, который злоумышленник может вставить в код сайта, практически неограниченны. Суть атаки – выйти за пределы HTML-тега через специальные символы и далее внедрять свой код. Защита от этого вида атак сводится к фильтрованию данных, отосланных пользователем. То есть по аналогии с SQL-инъекциями защита от XSS заключается в грамотном построении промежуточного кода веб-проекта. Из-за популярности данной атаки разрабатываются специальные функции для большинства языков, используемых в веб-разработке, позволяющие блокировать данный вид атак. Большинство из этих функций имеют схожий алгоритм работы: обработка переданных аргументов (скриптов и т.д.), включающая в себя анализ и преобразование в HTML-сущности потенциально небезопасных составляющих аргумента, исключение из аргумента строк, помеченных пользователем как небезопасные, либо пропуск только разрешенных пользователем частей кода.

Как было сказано ранее, 25 % всех атак на веб-сайты проводится посредством эксплуатации SQL-инъекций. Таким образом, внесением новых способов разграничения доступа пользователей к содержимому веб-сайта можно добиться существенного уменьшения количества атак посредством SQL-инъекций либо аналогичных атак, эксплуатирующих уязвимости существующей системы разграничения прав доступа.

В качестве альтернативы существующей системы могут быть предложены представленные ниже способы построения архитектуры веб-сайта.

Создание на одном сервере двух виртуальных хостов. В результате данного разделения становится возможным сделать раздельные конфигурационные файлы для каждого из хостов. Один хост создается для входа администратора, другой – для входа пользователей. Таким образом, можно создать два различных конфигурационных файла, в которых будут содержаться различные реквизиты доступа к базе данных. В результате этого обычные пользователи и администратор при работе с сайтом будут обращаться к базе данных от разных пользователей СУБД с различными правами. При этом становится достаточно затруднительным использование SQL-инъекций при входе на сайт с правами пользователя, так как права пользователя СУБД попросту не позволят выполнять неуполномоченные запросы. Также возможно при таком подходе создание двух СУБД для разных категорий пользователей сайта.

Аутентификация пользователей непосредственно средствами СУБД. Возможен вариант, при котором пользователи аутентифицируются не через таблицу в БД, а непосредственно средствами са-

мой СУБД. При этом отпадает необходимость разграничения прав доступа на уровне промежуточного кода либо разработки отдельной для этого таблицы с ролями пользователей.

Также возможна передача аутентификации внешнему сервису, использующему не язык SQL (например, через протокол LDAP).

Введение сессии на уровне СУБД. В базе данных создается процедура авторизации, которая проверяет логин и пароль пользователя и в случае успеха устанавливает значение некоторой сессионной переменной, которая была бы доступна для чтения до конца текущей сессии.

Библиографический список

1. Уязвимости веб-приложений: под ударом пользователи // Habr.com : информ.-справочный портал. – URL: https://habr.com/company/pt/blog/306622/ (свободный).

Образец цитирования:

Круглов, А. С. Информационная безопасность интернет-сайтов / А. С. Круглов, М. Ю. Лупанов // Инжиниринг и технологии. – 2019. – Vol. 4(1). – С. 1–4. – DOI 10.21685/2587-7704-2019-4-1-11.