УДК 004.056.53

Р.Х. Хамдамов, К.Ф. Керимов, Дж.О. Ибрагимов

МЕТОДИКА РАЗРАБОТКИ БРАНДМАУЭРА ВЕБ-ПРИЛОЖЕНИЙ

Ключевые слова: веб-приложение; угроза безопасности; брандмауэр веб-приложений; введение в PHP.

Введение

Из-за быстрого развития интернет-технологий веб-приложения становятся неотъемлемой частью повседневной жизни людей. В мире насчитывается огромное количество веб-приложений, работающих под управлением языка программирования PHP. Однако высокий темп развития веб-приложений создает угрозу для них. Угрозы информационной безопасности настолько разнообразны, что традиционные средства защиты не всегда эффективны.

Необходима разработка брандмауэра веб-приложений, который будет защищать веб-приложение от угроз информационной безопасности.

Брандмауэр веб-приложения значительно упрощает управление безопасностью и дает возможность администраторам защищать приложение от угроз вторжения.

Основная часть

Веб-сайт может содержать несколько компонентов, таких как веб-клиент, веб-приложения, веб-серверы и базы данных и т.д. Любой компонент может быть серьезной проблемой безопасности.

Рассмотрим общие угрозы безопасности и дадим рекомендации по защите.

SQL-инъекция: злоумышленник использует недостаточную фильтрацию входящих данных программистами. Воспользовавшись данной уязвимостью, атакующий может создать и отправить специальный код, содержащий инструкцию SQL для несанкционированного доступа на сервер, что может привести к серьезным проблемам.

Необходимые меры по обеспечению защиты:

а) проверять вводные данные, фильтровать или преобразовывать специальные символы в строке (varchar, текст и т.д.), например одинарные кавычки и символ комментария и т.д.

б) при возникновении ошибки во время выполнения сценария sql, рекомендуется не отображать полное сообщение об ошибке для пользователя;

в) по возможности использовать сохраняемые процедуры при создании запроса.

Межсайтовый скриптинг (XSS): XSS — это особый способ атаки на динамические страницы. Злоумышленник внедряет вредоносный код на веб-страницы. Любой пользователь, который обращается к этой странице через браузер, загрузит и выполнит внедренный код.

Необходимые меры защиты:

а) запретить браузеру автоматически загружать сценарий на языке JavaScript, запретить мета-тег <META REFREASH> и тег <IFRAME>;

б) отфильтровать специальные символы в коде динамического сценария, например JavaScript-скрипт <>;

в) ограничить длину ввода;

г) использовать ограничения для загрузки файлов, в связи с тем что XSS-атака часто происходит с Flash- и другими файлами.

Отказ в обслуживании (DoS): цель данной угрозы — заставить сервер прекратить свое обычное функционирование. Злоумышленник использует уязвимости в сетевом протоколе (протокол TCP/IP), может произвести атаку на такие ресурсы сервера, как память, процессор, пропускная способность сети и т.д., с помощью различных методов.

Необходимые меры по защите:

а) обеспечить защиту операционной системы, т.е. установить файрволы и антивирусы и периодически просматривать журнал безопасности системы;

б) ограничить пропускную способность сети определенным протоколом во избежание нерационального использования системных ресурсов;

в) настроить брандмауэр, чтобы разрешить только необходимые процессы. Блокировать неиспользуемые порты и нежелательные IP-адреса.

Brute Force: метод взлома различных учетных записей путем подбора логина и пароля.

Необходимые меры по защите:

а) установить надежный пароль;

б) заблокировать учетную запись после нескольких неудачных попыток авторизации, т.е. если нельзя войти в систему определенное количество раз, заблокируйте учетную запись. Этот способ имеет недостаток. Если блокировка слишком чувствительна, злоумышленник может попытаться заблокировать всех пользователей. Следовательно, блокировка должна быть временной, например на пять минут. Таким образом, атаки могут быть эффективно предотвращены.

Path Traversal: злоумышленник может получить доступ к файлам веб-узла удаленно через определенный URL-запрос.

Необходимые меры по защите:

а) своевременно тестировать веб-приложения на уязвимость;

б) поместить динамические страницы, такие как *. cgi, *. asp, * .php и т.д., в защищенный каталог и запретить пользователям напрямую обращаться к файлам этого каталога.

Методика разработки брандмауэра веб-приложений

Брандмауэр веб-приложений (WAF) предназначен для защиты веб-приложений. В отличие от обычного брандмауэра, WAF может анализировать данные прикладного уровня и фильтровать данные уровня приложения. Что касается защиты веб-приложений, он обладает неоспоримыми преимуществами.

При исследовании существующих брандмауэров веб-приложений выявлено, что они работают на основе трех методов:

1) распознавание функций. Каждая атака имеет свои особенности. Этот метод распознавания признаков угроз обычно используют для обнаружения вирусов и червей. Но есть тысячи атак, имеющих похожие характеристики, поэтому число ложных срабатываний довольно высокое.

2) распознавание алгоритмов. Этот метод лежит в основе распознавания признаков. Классифицируются разновидности атак, которые имеют одни и те же характеристики в разных запросах. Тем самым на основе признаков выявляется угроза информационной безопасности.

3) согласование шаблонов. Составляются сигнатуры угроз информационной безопасности. Далее используются регулярные выражения для определения каждой угрозы.

Среди этих трех методов наиболее часто используется метод согласования шаблонов, поскольку уровень обнаружения в данном случае превышает уровень обнаружения в двух остальных методах.

Разработан брандмауэр веб-приложений PHP-WAF на основе технологии согласования шаблонов.

Структура PHP-WAF показана на рис. 1 — на одном компьютере находятся WAF, сервер PHP и веб-сервер.

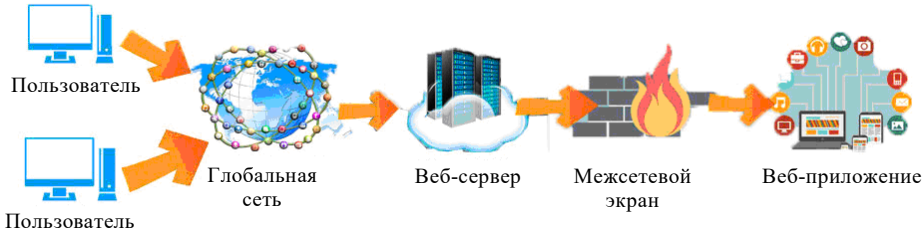

Рис. 1

Разработаны четыре основные функциональные подсистемы брандмауэра веб-приложений PHP-WAF:

- подсистема обнаружения угроз;
- подсистема аудита;
- подсистема конфигурации;
- интерактивная подсистема.

Общая архитектура показана на рис. 2.

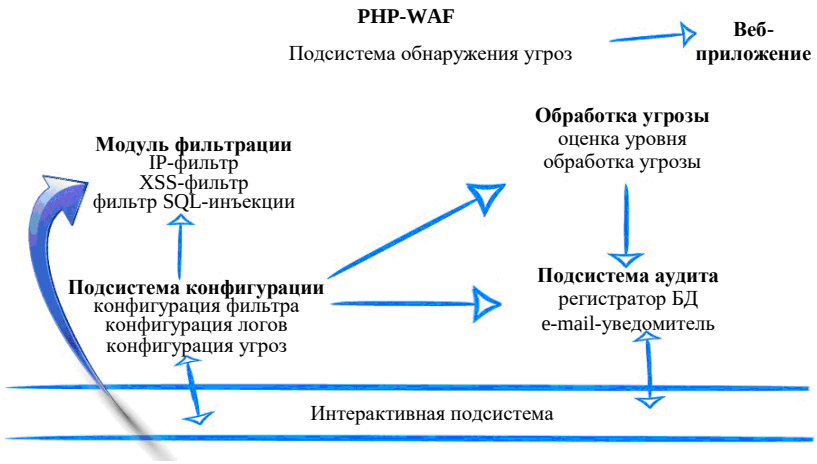

Рис. 2

Подсистема обнаружения угроз состоит из двух модулей:

- модулей фильтров;
- модуля обработки угроз.

Веб-запросы проходят фильтрацию данных через модуль фильтрации. Данный модуль фильтрует веб-запросы с помощью IP-фильтра, фильтра SQL-инъекций, фильтра XSS, фильтра идентификаторов и фильтра загрузки файлов. Модуль обработки оценивает уровень обнаруженной угрозы, сравнивает значение оценки и установленный порог в соответствии с результатом для осуществления блокировки.

IP-фильтр включается с самого начала и выполняет задачу блокировки запросов с IP-адресов, которые находятся в списке запрещенных. Добавление IP-адресов в список происходит автоматически при выявлении атаки с определенного IP-адреса.

После прохождения фильтра IP-адресов веб-запрос будет передан в фильтр SQL. Фильтр внедрения SQL анализирует запросы в соответствии с правилами фильтрации, а затем передает результаты в модуль обработки угроз.

Правила фильтрации состоят из таких выражений:

; \ W * (drop \ delete \ union \ grant \ select \ group \ update \ insert \ / \ * \ ex ec \ sp_ \ xp_ \ create \ alter \ truncate \ declare \ cast \ waitfor \ shutdown (соответствует ключевым словам SQL-инъекции);

/ \ *. **/a также . + (- \ ##) \ s * $ (соответствует символу комментария и т.д.).

После прохождения этих двух фильтров веб-запрос будет передан соответственно в фильтры XSS, идентификаторов и загрузки файлов. Архитектура этих фильтров похожа на фильтр SQL, в основе архитектуры которого лежит построение правил фильтрации.

Подсистема аудита содержит журнал угроз, который хранится в базе данных, и направляет уведомления по почте при обнаружении атак. В журнале базы данных записывается вся информация о веб-запросе, уровень угрозы которого превышает допустимый порог. Состояние безопасности веб-приложения отображается в виде диаграмм. Система уведомлений по электронной почте отправляет мгновенно уведомление при атаке на веб-приложение.

Администратор брандмауэра может настроить брандмауэр в подсистеме конфигурации. Обычные пользователи могут просматривать информацию о конфигурации. Модуль фильтра может быть настроен так, чтобы устанавливать порядок фильтрации, черный список IP-адресов, а также различные фильтры для включения и отключения. Журнал базы данных можно настроить для включения или отключения функции журнала и функции уведомления по электронной почте. Модуль обработки угроз может быть настроен для установки порога угрозы для ведения журнала.

Интерактивная подсистема в основном используется для взаимодействия с брандмауэром. Подсистема включает в себя две функции: просмотр журнала и управление учетными записями. После аутентификации администратор может просматривать журналы брандмауэра, перенастраивать брандмауэр, изменять информацию учетной записи администратора, добавлять пользователей и изменять информацию о пользователе и разрешении на просмотр. Обычные пользователи могут просматривать журнал брандмауэра и изменять информацию об учетной записи. Интерактивная подсистема использует большое количество диаграмм и статистических таблиц, которые представляют журнал брандмауэра пользователю в виде веб-страницы. По этим диаграммам и таблицам пользователи могут интуитивно понять состояние безопасности сервера веб-приложений.

Заключение

После всестороннего анализа основных угроз безопасности и соответствующих мер по защите [1–7] был разработан брандмауэр для веб-приложений PHP-WAF.

По результатам тестирования PHP-WAF может эффективно перехватывать различные веб-атаки и защищать веб-приложения, написанные на любом веб-ориентированном языке программирования.

Р.Х. Хамдамов, К.Ф. Керімов, Дж.О. Ібрагімов

МЕТОДИКА РОЗРОБКИ БРАНДМАУЕРА ВЕБ-ДОДАТКІВ

На сьогодні розвиток веб-ресурсів вказує на те, що відсутні єдині стандарти розробки захищених веб-додатків, що може призвести до помилок і вразливостей у веб-додатках. Вразливий веб-додаток може бути легко зламано без спеціалізованих засобів, тільки за допомогою браузера. У світі налічується величезна кількість веб-додатків, що працюють під керуванням PHP. Загрози інформаційної безпеки насті-

льки різноманітні, що традиційні засоби захисту не завжди ефективні. На основі всебічного аналізу загроз безпеки для веб-додатків запропоновано брандмауер веб-додатків на базі PHP. Розроблений брандмауер веб-додатків складається з чотирьох підсистем: підсистема виявлення загроз, аудиту, конфігурації та інтерактивна підсистема. Кожна з них виконує певні функції щодо захисту веб-додатку від загроз інформаційної безпеки. Запропоноване рішення працює як проксі-сервер і перевіряє весь вхідний трафік до веб-додатку, що дозволяє повністю контролювати всі вхідні запити. При виявленні деструктивних запитів відбувається їх блокування, а також оповіщення адміністратора про поточну атаку на веб-додаток. Результати тестування показують, що брандмауер може ефективно блокувати різні шкідливі атаки на рівні додатків, такі як SQL Injection — sql-ін'єкції, Remote Code Execution (RCE) — віддалене виконання коду, Cross Site Scripting (CSS) — міжсайтовий скриптинг, Cross Site Request Forgery (CSRF) — міжсайтова підробка запитів; Remote File Inclusion (RFI) — віддалений інклуд; Local File Inclusion (LFI) — локальний інклуд; Auth Bypass — обхід авторизації, Bruteforce — підбір паролів і т.д., а також комплексно захищати веб-додатки.

Ключові слова: веб-додаток; загроза безпеки; брандмауер веб-додатків; введення в PHP.

R.Kh. Khamdamov, K.F. Kerimov, J.O. Ibrahimov

METHOD OF DEVELOPING A WEB-APPLICATION BRANDMAUER

The development of web-resources indicate that there are no uniform standards for the development of secure web-applications, which can lead to errors and the appearance of vulnerabilities in web-applications. A vulnerable web-application can be easily hacked without using specialized tools, only using a browser. In the world there are a huge number of web-applications running PHP. Information security threats are so diverse that traditional remedies are not always effective. Based on a comprehensive analysis of security threats for web-applications, a web-application firewall based on the PHP language has been proposed. The developed web-application firewall consists of 4 subsystems: a threat detection subsystem, an audit subsystem, a configuration subsystem, and an interactive subsystem. Each subsystem performs certain functions to protect the web-application from information security threats. The proposed solution works as a proxy server and checks all incoming traffic to the web-application, which allows ont to control fully all incoming requests. If destructive requests are detected, they are blocked, and the administrator is notified of the current attack on the web-application. Test results show that the firewall can effectively block various malicious attacks at the application level, such as SQL Injection-sql injection, Remote Code Execution (RCE) — remote code execution, Cross Site Scripting (CSS) — cross-site scripting, Cross Site Request Forgery (CSRF) — intersite request forgery; Remote File Inclusion (RFI) — remote inclusion; Local File Inclusion (LFI) — local inclusion; Auth Bypass — bypass authorization, Bruteforce — selection of passwords. etc., as well as comprehensively protect web-applications.

Keywords: web-application; security threat; web-application firewall; introduction to PHP.

1. Пазизин С.В. Основы защиты информации в компьютерных системах. М. : ТВП-ОпиПМ, 2003. 73 с.
2. Петренко С.А., Петренко А.А. Аудит безопасности Intranet. М. : ДМК Пресс, 2002. 416 с.
3. Ржавский К.В. Информационная безопасность: практическая защита информационных технологий и телекоммуникационных систем: Учебное пособие. Волгоград : ВолГУ, 2002. 122 с.
4. Семкин С.Н., Беляков Э.В., Гребенев С.В. и др. Основы организационного обеспечения информационной безопасности объектов информатизации. М. : «Гелиос АРВ», 2005. 186 с.
5. Хорев П.Б. Методы и средства защиты информации в компьютерных системах. М. : Гелиос, 2006. 62 с.
6. Кондрашова Н.В. Согласование внешнего критерия и способа разбиения выборки при решении задачи структурно-параметрической идентификации методом группового учета аргументов. *«Международный научно-технический журнал «Проблемы управления и информатики»*. 2015. № 5. С. 20–33.
7. Opanasenko V.N., Kryvyi S.L. Synthesis of adaptive logical networks on the basis of Zhegalkin polynomials. *Cybernetics and Systems Analysis.* 2015. **51**, 6. P. 969–977. DOI: 10.1007/s10559-015-9790-1.

Получено 17.12.2018