О некоторых методах повышения безопасности веб-серверов и веб-сайтов

Артур Петросян
Институт проблем информатики и автоматизации НАН РА
Ереван, Армения
Эл.почта: arthur@sci.am

Гурген Петросян
Институт проблем информатики и автоматизации НАН РА
Ереван, Армения
Эл.почта: gurgen@sci.am

АННОТАЦИЯ
Современные тенденции в организации нападений на веб-сайты и веб-серверы требуют постоянного мониторинга средств нападений, для своевременной разработки и внедрению собственной или уже существующей и максимальной защиты от них. В статье представлены как авторские, так и некоторые нынешние передовые методы по повышению безопасности как самих веб-сайтов написанных на PHP/MySQL, так и веб-серверов Apache, где они установлены. Особое внимание уделено защите веб серверов в среде виртуального разделяемого хостинга.

Ключевые слова
Защита веб-сервера, защита веб-сайтов, разделяемый хостинг, анализ безопасности.

1. ВВЕДЕНИЕ
В современном мире сетевых коммуникаций услуга веб сервиса является одной из ключевых, поэтому защита веб серверов и веб сайтов является актуальной задачей. В связи с различными формами атак на веб сайты, становится очень важным внедрение эффективных методов защиты веб серверов. Рассмотрим некоторые шаги, благодаря которым возможно реализовать повышение безопасности веб-серверов и веб-сайтов.

2. ТРЕБОВАНИЯ К КОНФИГУРАЦИИ ВЕБ-СЕРВЕРА
Рассмотрим один из самых распространенных примеров конфигурации виртуального разделяемого хостинга веб-серверов **Linux**/**Apache-mpm-itk**/**MySQL**/**PHP**. Apache-mpm-itk это мультипроцессный модуль Apache, который позволяет запускать каждый виртуальный хост в рамках отдельного UID и GID настройка которого описана в статье [1]. Использование данного модуля крайне рекомендовано для виртуального разделяемого хостинга, т.к. при использовании **Apache-mpm-itk** скрипты и файлы конфигурации для одного виртуального хоста не будут доступны даже для чтения для всех других пользователей виртуальных хостов на том же веб-сервере, поскольку для каждого виртуального хоста отмечается свой пользователь и группа, от имени которого будет запускаться процесс apache, если к этому виртуальному хосту обратятся извне. Это заметно увеличит защиту, не давая возможность вредоносным скриптам одного хостинга проникать на другие. Также и на уровне системы пользователи не должны иметь доступ к каталогам друг друга, что можно организовать например с помощью механизма **SSH Chroot**. Помимо этого установка директивы настройки **PHP open_basedir** с помощью **php_admin_value**, для каждого отдельно взятого виртуального хоста позволит изолировать друг от друга пользователей и их виртуальные хосты также и на уровне **PHP** [1].

После установки данных пакетов необходимо отключить все неиспользуемые сервисы, которые могут слушать определенные порты, давая возможность атаковать их. Крайне желательно заменить стандартные порты на нестандартные для сетевых служб, которые необходимо оставить включенными (SSH, MySQL и т.д.), что снизит количество прямых атак на них. После установки пакетов веб-сервера, желательно настроить его как можно оптимально, в частности уменьшить выдаваемую информацию о веб-сервере изменив такие директивы Apache как **ServerSignature** и **ServerTokens** на **Off** и **Prod,** соответственно. Настройки PHP зависят от используемой версии, некоторые представлены ниже. Желательно выключать директивы **register_globals** (актуально до версии 5.4), **expose_php**, **allow_url_fopen**, определить глобальный **open_basedir** причем также и для **cli** версии, выставить **max_execution_time**, **post_max_size**, **upload_max_filesize** в соответствии с требованиям хостинга, определить список запрещенных функций, особенно тех, которые разрешают запуск системных команд (**system, exec** и т.д.), т.к. большинство вредоносных кодов (например бекдоров) используют их.

Наличие на веб-сервере установленного пакета **Fail2ban** [3] крайне желательно. Этот пакет в онлайн режиме анализирует лог файлы сетевых служб и в соответствии с критериями частоты, специфичности запросов и ответных отказов отправляет в бан лист те IP адреса, которые превышают заранее определенный лимит в зараннее определенный период времени. Бан лист представляет из себя отдельные цепочки сетевого фильтра **iptables,** куда на длительный период времени попадает провинившийся IP адрес. Довольно простой метод построения фильтров на основе регулярных выражений позволяет администратору при необходимости самому подстроить под себя каждый фильтр. На уровне сетевых служб желательно, чтобы **Fail2ban** следил за всеми сетевыми службами, слушающие открытые порты.

Также можно создать и использовать небольшой дополнительный скрипт для предотвращения небольших DoS атак. Его суть сводится к тому, чтобы с помощью команды **netstat** собирать информацию о количестве подключений по тому или иному протоколу с определенного IP адреса и если количество превышает заранее установленное значение, то IP адрес вносится в бан лист с помощью **iptables**.

3. ПОВЫШЕНИЕ БЕЗОПАСНОСТИ ВЕБ-САЙТОВ

Для реализации повышения безопасности веб-сайтов рекомендуется реализация следующих шагов:

- Фильтрация или экранирование абсолютно всех переменных передаваемых скриптам извне ($_GET, $_POST …) на наличие символов: " ' / \ & | > < = и т.д.
- Пользователь баз данных, который используется в front end части сайта, должен иметь минимальные права доступа к базе данных, в наилучшем варианте права только на чтения, если же необходимо иметь также и права записи, то желательно предоставлять такие права на конкретные таблицы, а не на всю базу данных, при этом используя подключения с широкими правами только в тех файлах сайта, где необходимы такие права.
- Административный интерфейс сайта необходимо выносить из виртуального хоста front end части сайта в другой, несуществующий, при этом сменив стандартный порт 80 на другой.
- Административный интерфейс и утилита для администрирования базы данных должны быть защищены как минимум методом .htaccess/.htpasswd, с ограничением доступа с конкретных IP адресов.
- И сайт, и Административный интерфейс, и интерфейс для администрирования базы данных должны работать только через протокол HTTPS.

Известно, что для того, чтобы через скрипты сайта возможно было загружать файлы в определенную папку на сервер, необходимо, чтобы пользователь, от имени которого запускается процесс веб-сервера, имел права записи на эту папку. При использовании **apache-mpm-itk** и специальных настроек такие права выставлять можно только для группы [1]. При возможности следует выносить такие папки из зоны видимости веб-сервера, то есть выше папки виртуального хоста. Также необходимо реализовать самим скриптом загрузки файлов на сервер, проверку на разрешенные для загрузки типы файлов.Однако этого недостаточно - желательно глобально следить за потенциально уязвимыми папками сайтов, которые становятся целью нападающих.С этой целью авторами статьи была разработан специальный механизм [2]. Его суть сводится к следующему:
Специальный скрипт анализирует содержимое таких папок, проверяя, чтобы в них не находились файлы запрещенных типов (заранее определенные, например любые php файлы). Если таковые находятся, они перемещаются в другую папку для дальнейшего анализа, о данном инциденте делается запись в лог файле, а также по эл. почте оповещается администратор.

В папках такого типа также всегда должен находиться **.htaccess** файл, который на уровне веб сервера запрещает выполнение любых скриптов. Разработанный механизм включает в себя также скрипт проверки на наличие такого .**htaccess** файла и если его нет или его содержимое не соответствует заранее определенному, то производится создание/корректировка файла. Скрипт запускается в постоянном режиме и делает проверку каждые 1-3 секунды (период можно менять). Таким образом, даже если в сайте будет уязвимость и злоумышленник сумеет каким-либо способом загрузить в папку вредоносный файл (бекдор) - он будет тут же оттуда перемещен, информация об инциденте запишется в лог файл, а администратор сервера будет об этом оповещен.

В дополнение к вышеуказанному крайне рекомендовано использование модуля **mod_security** [4] для веб-серверов **Apache**, т.к. львиную долю нападений, поисков уязвимостей, сканеров сайтов можно блокировать грамотно настроив этот модуль. Проектом **OWASP** (Открытый проект обеспечения безопасности веб-приложений) [5] был разработан основной пакет правил для модуля **mod_security** (OWASP ModSecurity Core Rule Set) [6]. Использование этих правил дает возможность определить и записать в лог попытки следующих атак на сайт: **SQL Injection** (SQLi), **Cross Site Scripting** (XSS), **Local File Inclusion** (LFI), **Remote File Inclusion** (RFI), **Remote Code Execution** (RCE), **PHP Code Injection**, **HTTP Protocol Violations**, **Shellshock** и т.д. Данные правила необходимо правильно использовать, не включать все подряд, из-за этого могут перестать работать некоторые функции сайтов. Правила надо включать поэтапно, при этом каждый раз анализируя логи.
После того, как **mod_security** запишет в лог попытку атаки, предотвращением атаки будет заниматься все тот же **Fail2ban,** у которого есть специальный фильтр для отслеживания срабатываний правил mod_secruity в логе, то есть OWASP правила модуля **mod_secruity** обнаруживают попытку атаки, записывают ее в лог, а затем **Fail2ban** анализирует этот лог и в соответствии с заранее определенными критериями заносит в бан лист нежелательные IP адреса. Получается, что симбиозом двух мощных инструментов можно эффективно предотвращать некоторые виды атак.

4. ТЕСТИРОВАНИЕ

Существует несколько инструментов тестирования и анализа безопасности веб-серверов и сайтов, наиболее распространенные это **Acunetix Web Vulnerability Scanner** [7], **sqlmap** [8], **OWASP ZAP**, **Nikto Web Scanner**. Остановимся на двух из них: первый инструмент позволяет анализировать такие уязвимости, как SQL Injection, XSS и т.д., а также уязвимости в самом веб-сервере.
Утилита **sqlmap** является мощным инструментом для поиска и эксплуатации уязвимостей типа SQL Injection. В утилите **sqlmap** предусмотрена возможность посылать запросы поиска уязвимостей каждый раз с разных IP адресов, используя сеть Tor. Данная проблема уже является распределенной атакой с использованием анонимных IP адресов. Частичное решение данной проблемы можно осуществить запретив все подключения с известных на данный момент и опубликованных IP адресов узлов Tor сети, но такая защита не является полной, т.к. списки IP адресов могут меняться и не являются полными.
Необходимо отметить, что при применении методов защиты, описанные в этой статье, как сканер **Acunetix** так и утилита **sqlmap** перестают работать после нескольких попыток поиска уязвимостей. IP, с которого производится тестирование попадает в бан лист, если же **sqlmap** работает через сеть Tor, то львиную долю атаки можно остановить, запретив подключения из Tor сети. Предотвращение попыток атак с использованием анонимной Tor сети является темой для дальнейших исследований авторов статьи.

5. ЗАКЛЮЧЕНИЕ

В данной статье, представлены методы по повышению безопасности виртуального разделяемого хостинга. Применение этих методов, адаптация их к конкретному случаю конфигурации, а также грамотное создание сайтов могут резко сократить поток атак, попыток сканирования, направленных на веб-сервер и веб-сайт.

ССЫЛКИ

[1] A. Petrosyan, G. Petrosyan - "Research and Deployment of Improved Web Server Protection Methods", Mathematical Problems of Computer Science 42, 81-84, 2014

[2] A. Petrosyan, G. Petrosyan – "Development and Implementation of Some Advanced Web Server Protection Methods", Mathematical Problems of Computer Science 46, 66-71, 2016.

[3] Fail2ban - http://www.fail2ban.org

[4] ModSecurity - https://modsecurity.org/

[5] OWASP Project - https://www.owasp.org

[6] OWASP ModSecurity Core Rule Set - https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

[7] Web Vulnerability Scanner - https://www.acunetix.com/

[8] sqlmap - http://sqlmap.org/