**Универсальный метод защиты веб-приложений***А.В. Адамов, А.В. Бабич*
(Тюмень, ТюмГУ, Кафедра ИБ)

**Universal web-application protection method***A.V. Adamov, A.V. Babich*

**Введение**

На сегодняшний день порог вхождения в сферу создания веб-сайтов минимальный, благодаря упрощенным языкам веб-программирования и развивающимся технологиям, делающими процесс создания сайта простым, но эффективным с точки зрения достижения цели заказчиком. Не удивительно, что наряду с бурным развитием сферы Интернет, теми же темпами развиваются и угрозы безопасности работающих в ней сервисов.

 В тоже время, роль сайтов и их влияние на бизнес постоянно растет, а перенасытившийся рынок неквалифицированных веб-программистов, зачастую роль которых выполняют студенты, пагубно влияет на качество программных продуктов и сайтов в частности. Проблема безопасности у таких исполнителей, к сожалению, стоит далеко не на первом месте. Положение усугубляется и тем, что заказчик порой остается в неведении, что его сайт и все, располагающиеся там данные, являются слабо защищенными или не защищенными вообще.

По данным статистики WASC (Web Application Security Consortium)[1], более 13% сайтов могут быть скомпромитированы полностью автоматически, 80-96% из которых имеют высокую степень уязвимостей, 86% — среднюю степень уязвимостей, 37% — низкую.

В данной статье хотелось бы поговорить о части Интернета, являющейся его непосредственным лицом, и касающейся практически каждого пользователя — это веб сайты и их безопасность. Актуальность проблемы безопасности такого вида приложений растет с каждым днем, а большинство уязвимостей, существующих на данный момент в этой сфере, связаны с ошибками и недочетами, допущенными на этапе разработки сайта.

## Что делать?

Таким образом, мы подошли к вопросу о необходимости решения указанной проблемы. Возможных вариантов несколько. Первый — обучить всех веб-программистов основам безопасности при создании сайтов, но данный способ не масштабируем и трудно реализуем с учетом темпов роста количества программистов. Наиболее оптимальным вариантом компенсации человеческого фактора, — недочетов программиста в процессе создания веб-сайтов, на сегодняшний день является технология WAF (Web Application Firewall).

**WAF — Брандмауэр веб-приложений**

WAF — это межсетевой экран, накладывающий определенный набор правил на то, как происходит взаимодействие сервера и клиента, обрабатывая HTTP-пакеты. В основе лежит тот же принцип, что и в обычных фаерволах — контроль и анализ всех пакетов, поступающих от клиента. WAF опирается на набор правил, с помощью которого выявляется факт атаки по сигнатурам — признакам активности пользователя, которые могут означать нападение.

Брандмауэр WEB приложений еще называют третьей линией обороны. В такой парадигме первой линией обороны являются межсетевые экраны, второй — системы IPS, и, наконец третьей — WAF (рис. 1).

 

Рисунок 1

**Типы WAF**

Web Application Firewall разделяют на 2типа: аппаратный и программный. Наибольшее распространение получил второй ввиду более простой реализации.

*По принципу действия* WAF можно разделить на три типа:

1. Реализованные в виде обратного прокси-сервера;
2. Работающие в режиме маршрутизации/моста;
3. Встроенные в веб-приложения.

**Обратный прокси-сервер**

В данном типе WAF все данные сначала обрабатываются прокси-сервером, который уже решает пропускать пакеты или блокировать. В случае положительного результата данные перенаправляются к веб-серверу без изменения, либо с частичной правкой.

*К данному типу относятся: mod\_security [2], Barracuda [3], nevisProxy [4].*

**Режим маршрутизации/моста**

К этому типу чаще всего относят аппаратные WAF. У данного типа реализации есть как плюсы, так и минусы. К первым можно отнести прирост производительности, ко вторым более сложную и тонкую настройку.

*Примером данного типа WAF является Impreva SecureSphere [5].*

**Встроенные в веб-приложения**

Данный тип WAF встраивается непосредственно в веб-приложение в качестве дополнительного функционала и работает на программном уровне.

*Наиболее распространенным подобным WAF в России является брандмауэр, встроенный в CMS Битрикс[6].*

Обработка правил в WAF может осуществляться по принципу blacklist (производится сопоставление со списком недопустимых условий), whitelist (принимаются только разрешенные действия) или смешано.

**Типы угроз**

На сегодняшний день, почти все брандмауэры веб приложений призваны защитить от основных типов угроз свойственных веб сайтам.

* SQL инъекция
* Межсайтовый скриптинг (XSS)
* Межсайтовые подделки запросов (CSRF)
* Спам в комментариях
* Распределенный отказ в обслуживании (DDoS-атаки)
* Отсутствие таймаута сессии
* Обратный путь в директориях

**Недостатки WAF**

Основной проблемой, существующей на данный момент, являются ограниченные возможности существующей технологии WAF в обеспечении защиты от широкого спектра угроз. А также возможность обхода существующих на данный момент брандмауэров.

Каждый брандмауэр имеет отличительную особенность и оставляет за собой след. Для обнаружения того или иного firewall используется метод распознавания “fingerprint”, что в переводе означает “отпечаток пальца”. Например:

* специальные коды ответа при передаче особых данных или вызове ошибок;
* специальные переменные, хранимые в Cookie;
* изменение HTTP-заголовков, в частности данные, передающиеся в Server;
* незамедлительное завершение соединения при срабатывании недопустимого условия;
* встроенный набор базовых правил, поддающийся раскрытию.

После того как WAF обнаружен остается только найти его соответствующую уязвимость и использовать ее. 100% защиты не существует и WAF в этом не исключение. На сегодняшний день существует множество вариантов обхода WAF и этот список постоянно пополняется. В качестве примера можно взять исследование компании Positive Technologies, которые нашли более 30 возможностей обхода существующих WAF [7].

**Заключение**

Если говорить простым языком, WAF — это универсальный способ минимизировать угрозы, связанные с человеческим фактором, при создании веб-приложений. Как у любого универсального метода, WAF имеет ряд недостатков.

Основная проблема современных WAF кроется в их архитектуре, основанной на общем принципе. Все они используют сигнатурный анализ для определения типа угроз.

Недостаток такого подхода очевиден — его легкая обнаруживаемость и относительно легкий способ обхода.

Один из возможных вариантов решения этой проблемы я вижу в применении методов поведенческого анализа. Принцип такого подхода в корне отличается от сигнатурного. За основу берется нормальное поведение, скажем, в скрипте С чтение из таблицы А - нормально, если происходит чтение из таблицы Б, это считается аномальным. Данный подход в теории может закрыть уязвимости связанные с сигнатурным анализом. Данное направление я вижу наиболее перспективным в решении проблем безопасности веб-сайтов.

**Библиографический список**

1. Статистика уязвимостей Web-приложений за 2008 год – Режим доступа: http://ru.scribd.com/doc/21324267/WASC-Web-Application-Security-Statistics-2008-Russian

2. ModSecurity – Режим доступа: http://[modsecurity.org](http://modsecurity.org),

3. Barracuda Networks, Inc. (US) – Режим доступа: http://[barracudanetworks.com](http://barracudanetworks.com),

4. AdNovum Informatik AG – Режим доступа: http://[adnovum.ch](http://adnovum.ch),

5. A division of Virtual Graffiti, Inc. – Режим доступа: http://[impervaguard.com](http://impervaguard.com/),

6. Компания «1С-Битрикс» – Режим доступа: http://[1c-bitrix.ru](http://1c-bitrix.ru/),

7. Дмитрий Евтеев, Методы обхода Web Application Firewall – Режим доступа: <http://www.ptsecurity.ru/download/PT-devteev-CC-WAF.pdf>