Защита Web приложений с помощью Apache и mod_security

Авторы: Иван Ристик, перевод Александр Антипов

Источник: http://www.securitylab.ru/analytics/216322.php

Вступление:

В последнее время резко увеличилось количество нападений через HTTP протокол, поэтому все чаще возникает потребность в использовании дополнительных средств защиты Web приложений. Большинство существующих инструментов работает на TPC/IP уровне и не способны контролировать нападения, специфические для HTTP протокола. Для увеличения безопасности Web приложения лучше всего использовать прикладные шлюзы – утилиты, которые являются обратными прокси к Web приложению, способные выполнять анализ протокола. В этой статье мы покажем, как быстро можно развернуть ваш собственный прикладной шлюз, используя общедоступный бесплатный инструмент mod_security.

Обратный прокси сервер

Наша задача состоит в том, чтобы защитить один или более Web серверов, постоянно находящихся во внутренней сети и обеспечивающие услуги внешним клиентам. В этой статье мы предполагаем, что внутренние клиенты, типа служащих, постоянно находящихся во внутренней сети, тоже являются внешними клиентами для защищаемых нами Web серверов. Также мы предполагаем, что нам требуется защитить два или более Web серверов, сервер базы данных и, возможно, другие внутренние сервера. Чем больше серверов, тем более оправдано использование обратного прокси сервера.

Прокси, по определению, является устройством, которое расположено между двумя объектами, участвующими в сеансе связи. Чаще всего прокси сервер используется как прямой прокси – устройство, которое расположено между клиентом и сервером. Обратный прокси-сервер делает точно наоборот:

он расположен между сервером и всеми его клиентами. В более широком смысле, один обратный прокси-сервер будет использоваться для всех внутренних Web серверов.

Основное преимущество использования обратного прокси сервера – единая централизация. Как только мы заставляем весь трафик проходить через единую точку, мы можем выгодно использовать другие инструментальные средства для его анализа. Кратко рассмотрим преимущества и недостатки использования обратного прокси сервера:

Преимущества:

Единая точка доступа. Используя единую точку доступа, вы можете централизованно управлять доступом для всех ваших Web серверов. Здесь, например, лучше всего осуществлять ограничение доступа, основанное на IP адресах. Также вы можете централизованно регистрировать все запросы, что значительно упрощает их дальнейшую обработку.

Firewall на HTTP уровне. Так как прокси сервер обычно создает новый запрос, основанный на первоначальном запросе, то вы можете использовать прикладные межсетевые экраны для выполнения более широкого набора проверок, контроля трафика и реагирования на возможные нападения в реальном масштабе времени.

Увеличение производительности. Поскольку обратный прокси-сервер установлен на отдельной машине, это означает, что вы можете использовать дополнительные ресурсы центрального процессора. Вы можете осуществить кэширование как статического, так и динамического содержания. SSL трафик может использоваться только между прокси сервером и клиентом, освобождая фактический Web сервер только на ответ входящим запросам. Наконец, исходящий трафик может быть сжат, понижая тем самым требования к пропускной способности.

Сетевая изоляция. В этом случае обратный прокси сервер использует другой уровень межсетевой защиты, вместо того, чтобы защищать

множественные Web сервера и операционные системы, вы скрываете их позади единственного прокси.

Топология сети скрыта от внешнего мира. Это хорошо как минимум по двум причинам. Изначально мы даем атакующему намного меньше информации. Во вторых, вы отделяете структуру сети от ее интерфейса. Любые изменения в топологии сети будут выполнены незаметно для внешнего мира.

Недостатки:

Увеличенная сложность. Увеличивая защиту, вы увеличиваете сложность сети.

Единственная точка отказа. Для критических операций недопустимо наличие единственной точки отказа. Эта проблема может быть решена наличием двух обратных прокси в кластере, однако такой подход еще сильнее увеличивает сложность нашей сети.

Описание mod_security

ModSecurity – модуль Apache, добавляющий возможности обнаружения и предотвращения вторжения на Web сервер. Модуль подобен IDS системе, которую вы бы использовали для анализа сетевого трафика, за исключением того, что mod_security работает только на HTTP уровне. Модуль позволяет вам анализировать действия, обычные с точки зрения HTTP протокола, но трудные для анализа классическими IDS системами.

В дополнение к обнаружению, mod_security также поддерживает предотвращение нападение. Поскольку модуль расположен между клиентом и сервером, то если он найдет, что запрос содержит злонамеренные данные, он может отклонить запрос, выполняя любое множество встроенных действий.

Поскольку mod_security такой же модуль, как и множество других, вы можете использовать его как часть любой инсталляции Apache. Вот несколько из основных возможностей, предоставляемых модулем mod_security (более

подробно о работе модуля можно узнать из описания, доступного на Web сайте mod_security):

Анализ запроса. Это основная функция данного модуля, особенно когда вы имеете дело с POST запросами, в которых получение тела запроса может быть затруднено.

Выполнение канонизации и функции антиуклонения. Выполнение ряда преобразований, для преобразования входных данных в форму, подходящую для анализа. Этот шаг применяется для борьбы с различными методами уклонения.

Выполнение специальных встроенных проверок. В этом месте выполняются более сложные проверки правильности, такие как проверка правильности URL кодирования и проверка правильности Unicode кодирования. Вы можете также контролировать некоторые значения байтов в запросе для борьбы с shellcode.

Запуск входных правил. В этом месте запускаются правила, созданные вами. Они позволяют вам анализировать каждый аспект запроса, используя регулярные выражения. Также здесь могут быть объединены несколько правил для более сложного анализа. Затем запрос достигает обработчика. После запроса:

Запуск правил вывода. Правила вывода применяются к телу ответа. Они очень полезны для предотвращения утечек информации.

Регистрация запроса. Регистрируется окончательный запрос, состоящий из тела запроса и заголовков ввода и вывода. Чтобы предотвращать чрезмерную регистрацию, mod_security может регистрировать запросы по выбору, например запросы, которые получили ответ от mod_security.

Заключение

В данной статье освящена только малая часть очень сложной проблемы. Я предлагаю вам посмотреть ссылки на ряд сайтов, инструментальных средств, для того, чтобы ознакомиться с другими аспектами, которые мы не охватили в данной статье, например, балансирование загрузки обратного прокси и объединенная или прозрачная конфигурация обратного прокси. В данной ситуации можно действовать и другим способом, загрузите руководство по mod_security и узнайте его особенности. Однако данный модуль содержит также другие особенности, не упомянутые в руководстве. Поэтому, если вам это будет необходимо, свяжитесь со мной, чтобы узнать о новых особенностях mod_security.