ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ WAF ДЛЯ ЗАЩИТЫ ВЕБ-ПРИЛОЖЕНИЙ – ЧАСТЬ 1

Майкл Койфман (Michael Koyfman), главный разработчик глобальных решений
Netskope, США

На протяжении многих лет я неустанно просвещал клиентов и партнеров по поводу того, насколько важно принимать серьезные и эффективные меры по обеспечению безопасности приложений. Одной из моих основных рекомендаций всегда был и остается совет о развертывании Web Application Firewall (WAF). А поскольку F5 предлагает отличный передовой WAF под названием BIG-IP Application Security Manager (ASM), я тратил немало времени на разъяснение всех преимуществ, которые он дает.

Безопасность приложений – это непростая тема. И вот один из самых частых вопросов: «Сколько времени нужно для развертывания политики безопасности в режиме блокирования, чтобы эффективно защитить свое приложение?». Универсального ответа на него нет, но недавняя беседа с одним из заказчиков, задававших его, побудила меня взглянуть на ситуацию с точки зрения клиента. Я решил развернуть ASM перед общедоступным приложением и поделиться своими долгосрочными наблюдениями за решением реальных задач, а также достигнутыми результатами.

Я развернул уязвимое современное приложение Hackazon от Rapid7. Однако, прежде чем предоставить доступ к нему всем и каждому, мне нужно было обеспечить ему надежную защиту от взлома. Поэтому я развернул перед приложением BIG-IP ASM, а затем настроил политику безопасности в режиме блокирования. Как вы думаете, сколько времени мне на это понадобилось? Аж 15 минут!

Тут можно, конечно, сказать, что это вранье или так нечестно, ведь у меня многолетний опыт работы с продуктами F5 и деятельности в сфере обеспечения безопасности приложений. И это действительно так. Но за все эти годы команде разработки F5 удалось сделать развертывание базовой политики безопасности действительно очень простым и эффективным процессом

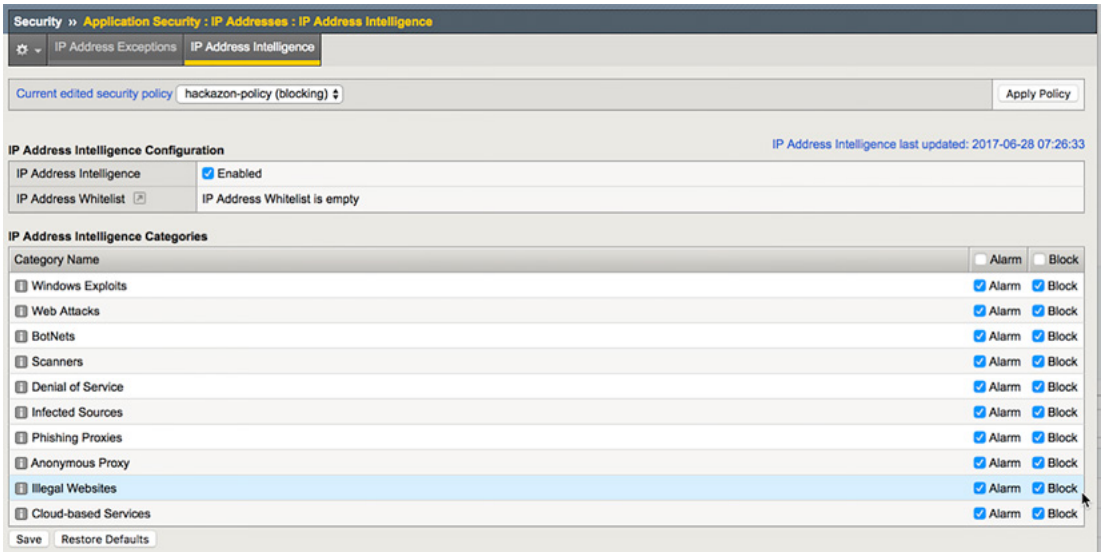
(спасибо им за это!). И, поскольку моя задача заключалась в том, чтобы поставить себя на ваше место, я просто воспользовался самым коротким и результативным способом, хорошо зарекомендовавшим себя на протяжении уже многих лет. То, что я сделал, – это отличный старт, поэтому рекомендую вам начать именно с этих шагов.

Во-первых, если уж совсем по-честному, в упомянутые мной 15 минут не вошли настройка и инсталляция физического или виртуального устройства BIG-IP «с нуля». Я допустил, что BIG-IP Local Traffic Manager уже запущен и работает: ведь сетевые администраторы скорее всего уже это сделали, а главная задача заключается в том, чтобы запустить и развернуть WAF. Когда BIG-IP LTM был настроен я создал политику безопасности на основе шаблона Rapid Deployment Policy, которым мы пользуемся для запуска большинства систем безопасности приложений вот уже много лет.

По умолчанию политика безопасности Rapid Deployment предусматривает:

- выполнение проверок на соответствие требованиям HTTP;
- выполнение проверок на наличие обязательных заголовков HTTP;
- остановка утечки информации;
- предотвращение использования в запросе недопустимых методов HTTP;
- проверку кодов ответа;
- обеспечение соответствия файлов cookie требованиям RFC;
- применение сигнатур атак к запросам (и ответам, если это предусмотрено);
- выявление использования методов обхода;
- предотвращение доступа с запрещенных геолокаций;
- предотвращение доступа для заблокированных пользователей, сессий и IP-адресов;
- проверку длины запроса на соответствие установленному размеру буфера;
- выявление запрещенного содержимого в загружаемых файлах;
- выполнение проверки на символы, которые не удалось конвертировать;
- отслеживание запросов с модифицированными ASM файлами cookies.

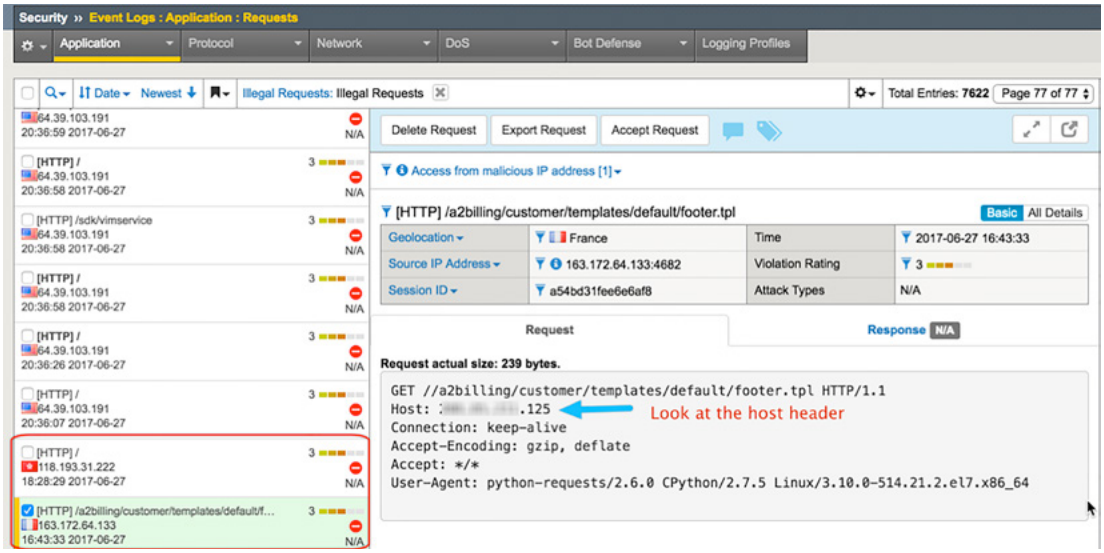
Несмотря на то, что все здесь выглядит превосходно, я воспользовался своими знаниями и опытом, и внес в политику безопасности два небольших изменения. Во-первых, я разрешил блокировку IP на основании собранных данных (intelligence-based).



Это значит, что при получении любого запроса на подключение к данному приложению с IP-адреса, отнесенного к любой из категорий вредоносных источников, политика безопасности просто его заблокирует – и точка! Данное решение может остановить любой вид трафика: от явных попыток взлома до, казалось бы, безвредных попыток зондирования со стороны известных угроз. Именно на них приходится значительная доля попыток взлома, поэтому блокирование таких запросов при помощи этого простого дополнения является весьма полезным шагом.

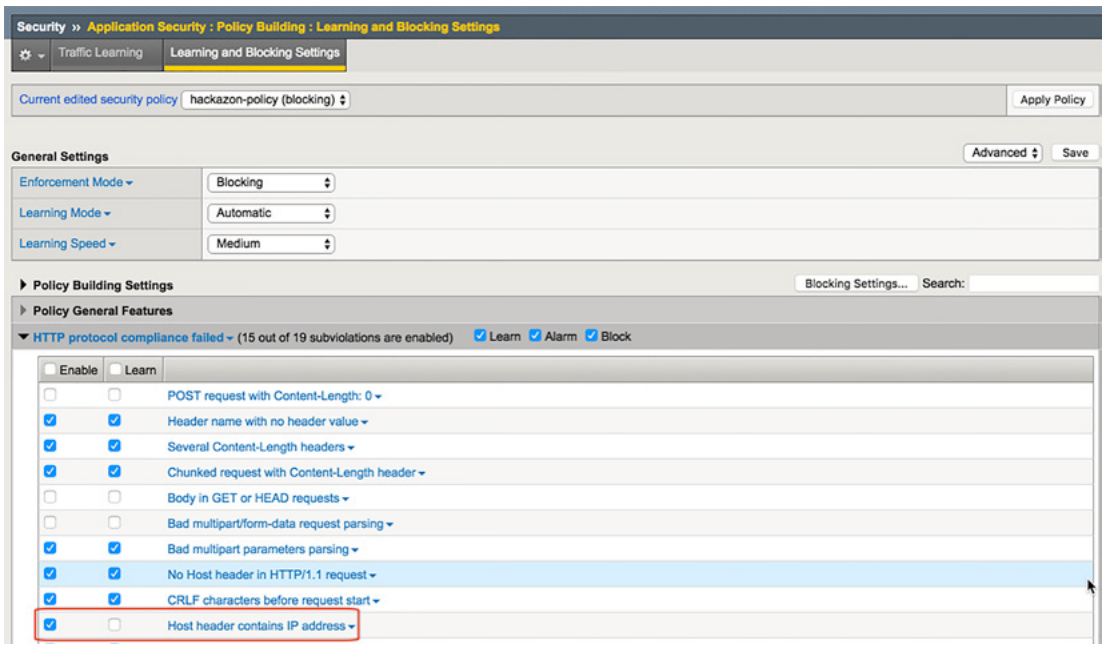
Внеся данное изменение, я перевел политику в режим блокировки и открыл доступ к приложению из Интернета (тоже в режиме блокировки).

Всего за пару часов произошло несколько несанкционированных запросов:



Как видно из первого запроса снизу, некто сканировал мою систему, чтобы узнать, не используется ли в качестве бэк-энда решение Asterisk A2Billing Softswitch. Моя политика заблокировала этот запрос (который сам по себе вредоносным не был) – вот и отлично! Следующий запрос тоже был заблокирован из-за того, что поступил с вредоносного IP-адреса. Я обрадовался. Однако, взглянув на первое нарушение, я вспомнил о том, что в политику нужно внести еще одно изменение. Возле стрелки, указывающей на заголовок имени хоста на скриншоте выше видно, что IP-адрес использовался в запросе в качестве заголовка хоста. Использование IP-адресов в качестве заголовков хоста не является обычным способом обращения к веб-приложениям, особенно при повсеместном распространении SSL, потому что выдать SSL-сертификат на IP-адрес невозможно. Можно сказать, что любой запрос, обращенный к приложению с поддержкой SSL, в котором в заголовке хоста используется IP-адрес, с большой вероятностью окажется вредоносным. Зловреды не знают полностью URL или доменное имя, используемое приложением, поэтому им приходится ставить IP-адрес пункта назначения в заголовок хоста HTTP для подключения, а это уже неопровержимая улика.

Именно это и побудило меня внести в политику второе изменение. Я включил блокировку за указание IP-адреса в заголовке хоста:



С этого момента любой запрос (неважно, с какого IP-адреса он поступил) будет заблокирован ASM, если заголовок хоста содержит IP-адрес.

Почему это так важно? Представьте себе обнаружение уязвимости нулевого дня, влияющего на используемую вами инфраструктуру бэк-энд сервера. Мое приложение работает на

ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ WAF ДЛЯ ЗАЩИТЫ ВЕБ-ПРИЛОЖЕНИЙ – ЧАСТЬ 1

Apache, поэтому здесь в качестве примера можно было бы обсудить уязвимость Apache Struts. Сразу же после того, как не устраненная уязвимость нулевого дня будет выявлена и начнет распространяться, все усилия взломщиков будут нацелены на IP-адреса, поскольку большинство атакующих либо не нацелены на имя хоста приложения, либо просто его не знают. Таким образом любой запрос, в котором нет заголовка хоста или имеется такой заголовок с IP-адресом, на ваш сервер не попадет. Это обеспечит вашу защищенность от атак нулевого дня, за исключением тех случаев, когда хакеры взламывают конкретно вас. Разве это не простая и беспроблемная защита от атак нулевого дня?

В целом, примерно за 15 минут я развернул надежную и эффективную политику безопасности перед своим приложением и могу теперь просматривать журналы, чтобы убедиться в ее пользе. Это лишь начало целой серии статей, в которых я расскажу о своих наблюдениях за результативностью такой защиты в долгосрочной перспективе и рассмотрю простые дополнительные настройки и доработки политик, повышающие эффективность обеспечения безопасности защищаемого или приложения.

Продолжение статьи читайте в части 2.

F5 Networks, Inc.
401 Elliott Avenue West, Seattle, WA 98119
888-882-4447 f5.com

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com