**Цель**: Изучение межсетевых экранов. Приобретение навыков работы с Iptables и WAF.

**1 Основные теоретические сведения**

**1.1 Межсетевой экран**

Скорее всего, ранее вы уже сталкивались с таким понятием как межсетевой экран. В ядро Linux встроен свой межсетевой экран, называемый Netfilter. Управление им осуществляется с помощью утилиты Iptables.

Межсетевой экран, сетевой экран, файервол, брандмауэр — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами. Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача — не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.

Рассмотрим принцип работы Netfilter. Когда сетевые пакеты попадают в сетевой интерфейс, они после ряда проверок ядром проходят последовательность так называемых цепочек. Пакет обязательно проходит через цепочку PREROUTING, после чего определяется, кому он, собственно, был адресован. Если пакет не адресован локальной системе (в нашем случае серверу), он попадает в цепочка FORWARD, а иначе — в цепочку INPUT, после прохождения которой отдается локальным демонам или процессам. После этого при необходимости формируется ответ, который направляется в цепочку OUTPUT. После цепочек OUTPUT или FORWARD пакет в очередной раз встречается с правилами маршрутизации и направляется в цепочку POSTROUTING. В результате прохождения пакетом цепочек фильтрации несколько раз, проверка его принадлежности определенным критериям осуществляется несколько раз. В соответствии с этими проверками к пакету применяется определенное действие:

● ACCEPT — пакет «принимается» и передается в следующую цепочку.

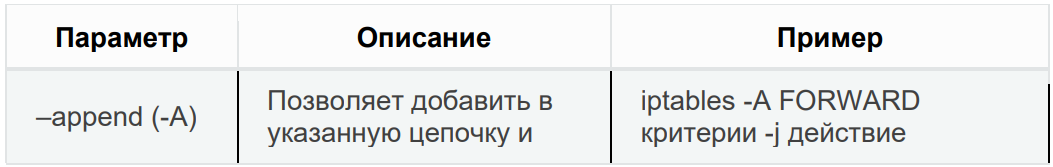
● DROP — удовлетворяющий условию пакет отбрасывается и не передается в другие таблицы или цепочки.

● REJECT — пакет отбрасывается, но при этом отправителю отправляется ICMP-сообщение, сообщающее об отказе.

● RETURN — пакет возвращается в предыдущую цепочку и продолжает её прохождение начиная со следующего правила

● SNAT — применить трансляцию источника в пакете. Используется только в цепочках POSTROUTING и OUTPUT таблицы nat.

● DNAT — применить трансляцию адреса назначения в пакете. Используется в цепочках PREROUTING и (очень редко) OUTPUT в таблице nat.

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Изображение выглядит как текст, снимок экрана, документ, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.  
Изображение выглядит как текст, снимок экрана, документ, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

**Основные цепочки межсетевого экрана Netfilter:**

● PREROUTING — изначальная обработка входящих пакетов

● INPUT — для входящих пакетов, адресованных непосредственно локальному компьютеру

● FORWARD — для маршрутизируемых пакетов

● OUTPUT — для пакетов, исходящих с локального компьютера

● POSTROUTING — для окончательной обработки исходящих пакетов

**Таблицы межсетевого экрана Netfilter:**

● raw - используется для маркировки пакетов, которые не должны обрабатываться системой определения состояний. Содержится в цепочках PREROUTING и OUTPUT.

● mangle — содержит правила модификации IP-пакетов.

● nat - предназначена для подмены адреса отправителя или получателя. Данную таблицу проходят только первые пакеты из потока - трансляция адресов или маскировка (подмена адреса отправителя или получателя) применяются ко всем последующим пакетам в потоке автоматически. Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержится в цепочках PREROUTING, OUTPUT, и POSTROUTING.

● filter — основная таблица, используется по умолчанию если название таблицы не указано. Используется для фильтрации пакетов. Содержится в цепочках INPUT, FORWARD, и OUTPUT.

**Пример создания правила для межсетевого экрана:**

Рассмотрим две цепочки, задающие два основных правила Iptables — PREROUTING и FORWARD.

● iptables -t nat -A PREROUTING -i eth0 -j DNAT —to-destination 192.168.57.102

● iptables -A FORWARD -d 192.168.57.102 -j ACCEPT

Первая из них определяет первоначальную обработку всех пакетов, приходящих на адаптер eth0:

● -t определяет подключаемую таблицу, в данном случае — nat — для подмены адреса отправителя или получателя

● -А — выбор цепочки

● -i — входящий интерфейс

● -j — действие с пакетами, удовлетворяющими условию — в данном случае DNAT — подмена адреса получателя

● –to-destination — выбор адреса, на который перенаправляются пакеты

● Вторая определяет проброс пакетов через сервер:

● -A — выбор цепочки

● -d — выбор адресата

● -j — выбор действия

**Web Application Firewall**

WAF (Web Application Firewall) - это межсетевые экраны, работающие на прикладном уровне и осуществляющие фильтрацию трафика Web-приложений. Эти средства не требуют изменений в исходном коде Web-приложения и, как правило, защищают Web-сервисы гораздо лучше обычных межсетевых экранов и средств обнаружения вторжений.

**Основные преимущества:**

● Анализ поведения пользователя в используемом приложении;

● Позволяет осуществлять мониторинг HTTP трафика и проводить анализ событий в реальном режиме времени;

● Предотвращение вредоносных запросов;

● Распознавание большинства опасных угроз;

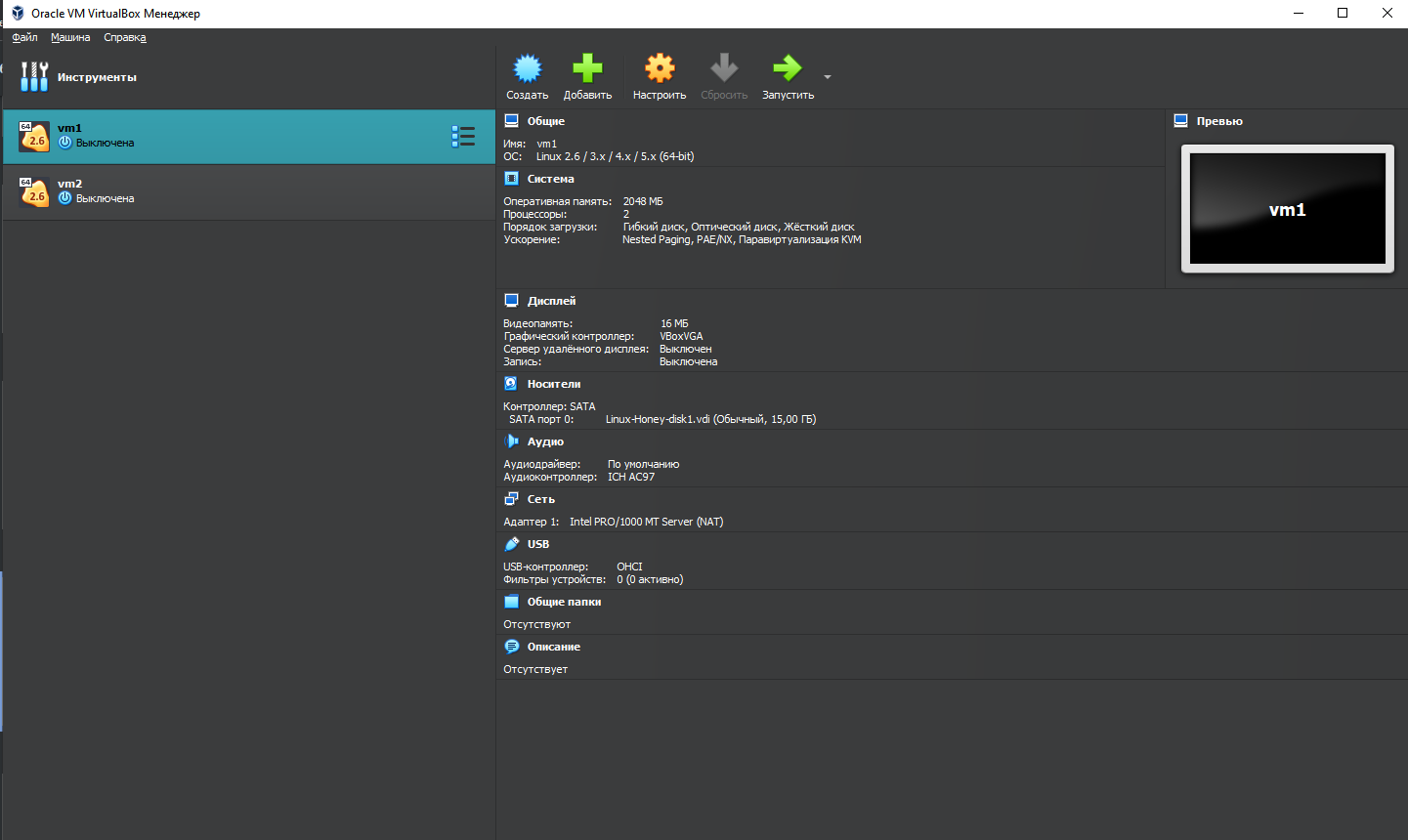
● Дополнение сетевых средств безопасности;

● Просматривать детальные отчеты об атаках и попытках взлома.

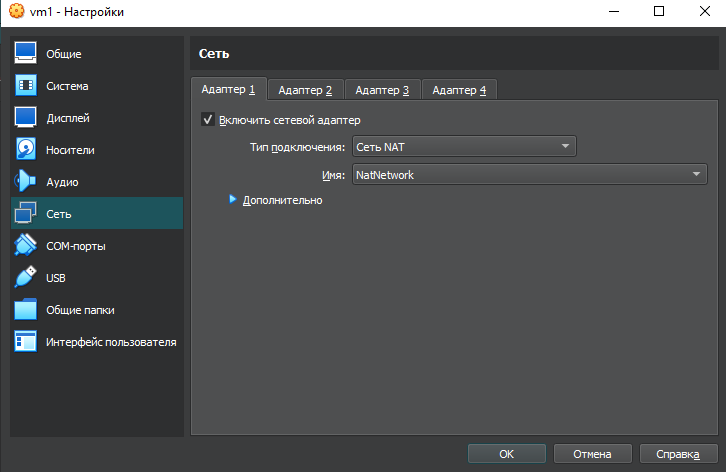
Практическая часть №3.

Часть 1. Iptables.

1. Создаём 2 виртуальные машины Linux по готовому шаблону, данному в задании:

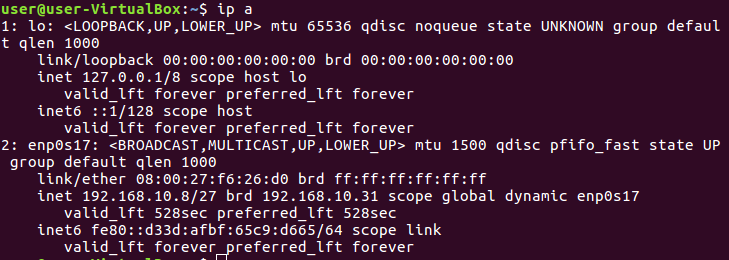


1. Настраиваем сетевые адаптеры данных машин для работы в режиме NAT. Для этого возьмём значение изолированной сети “Nat” из прошлой лабораторной работы. Затем выбираем тип подключения «Сеть NAT» на обеих машинах:



1. Запускаем машины. Пусть первая машина “vm1” будет Атакующей, “vm2” – Атакуемой. Теперь определяем IP и MAC адреса виртуальных машин используя команду “ip a”.

Первая машинa – Атакующий (X). IP адрес – 192.168.10.8/27, MAC адрес – 08:00:27:f6:26:d0.



Вторая машина – Атакуемый (N). IP адрес – 192.168.10.9/27, MAC адрес – 08:00:27:bf:76:1f.

Изображение выглядит как текст

Автоматически созданное описание

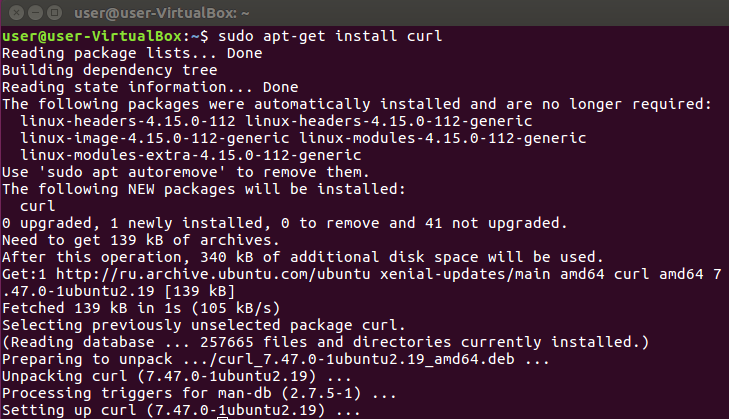
1. Далее на атакующую машину необходимо установить дополнительные пакеты. Выполним в терминале следующие команды:

“sudo apt-get update”

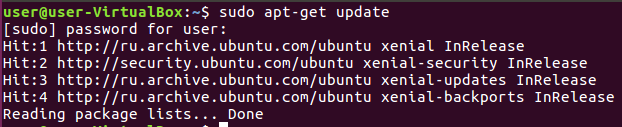
“sudo apt-get install curl”

Изображение выглядит как текст

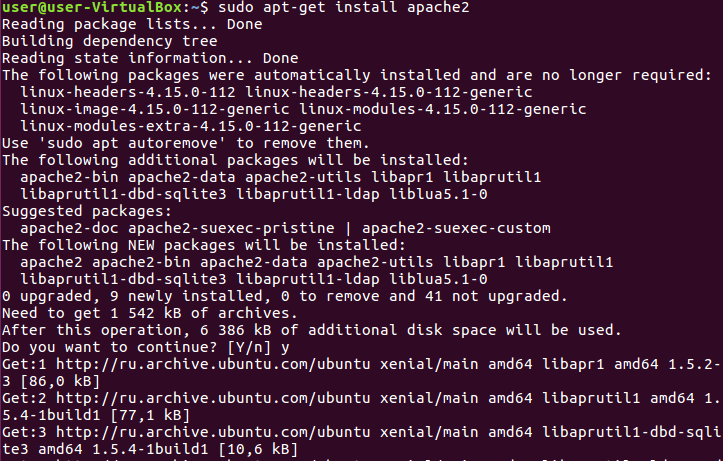
Автоматически созданное описание



1. Далее на атакуемой машине необходимо установить дополнительные пакеты. Выполним в терминале следующие команды:
2. “sudo apt-get update”



1. “sudo apt-get install apache2” (устанавливаем веб-сервер)

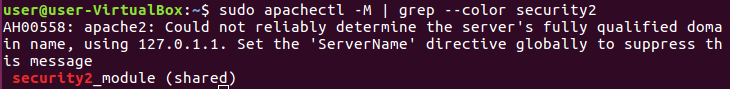


1. “sudo apt-get install libapache2-mod-security2” (устанавливаем модуль детектирования атак)

Изображение выглядит как текст

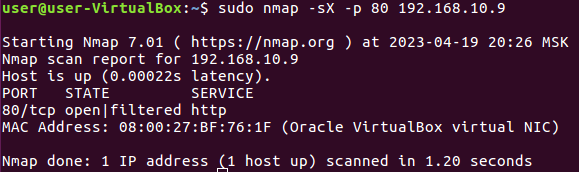
Автоматически созданное описание

1. “sudo apachectl -M | grep --color security2”



Видим, что на экране появился модуль по имени “security2\_module (shared)”, значит, все прошло успешно.

1. Далее проводим утилитой nmap Xmas сканирование атакуемой машины на порту 80. Воспользуемся командой “sudo nmap -sX -p 80 192.168.10.9”:



1. Далее настроим защиту атакуемой машины:
2. Просмотрим список текущих правил iptables таблицы filter, используя команду “sudo iptables -L”

Изображение выглядит как текст

Автоматически созданное описание

1. Видим, что список содержит три цепочки по умолчанию (INPUT, OUTPUT и

FORWARD), в каждой из которых установлена политика по умолчанию (на данный

момент это ACCEPT).

1. С помощью команды “sudo iptables -S” данный список можно просмотреть в

другом формате, который отражает команды, необходимые для активации правил

и политик.

Изображение выглядит как текст

Автоматически созданное описание

1. Чтобы сбросить текущие правила (если таковые есть), используем команду “sudo iptables -F”.



Цепочка INPUT отвечает за входящий траффик

1. Чтобы внести локальный интерфейс в разрешенные выполним команду “sudo iptables -A INPUT -i lo -j ACCEPT”.



1. Чтобы заблокировать весь исходящий трафик, кроме портов для SSH и веб-сервера, нужно сначала разрешить подключения к этим портам. В цепочку ACCEPT добавляем два порта (порт SSH 22 и порт http 80), что разрешит трафик на эти порты:

“sudo iptables -A INPUT -p tcp -m tcp –dport 22 -j ACCEPT”

“sudo iptables -A INPUT -p tcp -m tcp –dport 80 -j ACCEPT”



1. В данной работе мы не используем SSH. Так что удалим ненужное правило. Для этого выполним команду “sudo iptables -D INPUT -p tcp -m tcp –dport 22 -j ACCEPT”.



1. Нужно добавить еще одно правило, которое позволит устанавливать исходящие соединения (т. е. использовать “ping” или запускать обновления программного обеспечения). Для этого введём:

“sudo iptables -I INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT”



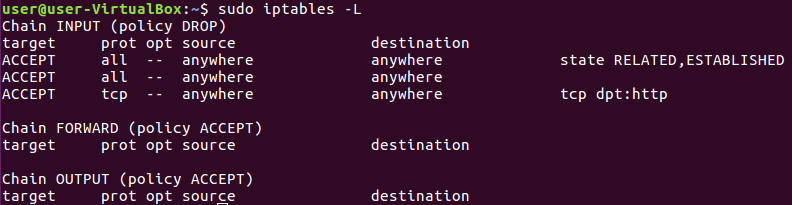
1. Создав все эти правила, можно заблокировать все остальное и разрешить все исходящие соединения.

“sudo iptables -P OUTPUT ACCEPT”

“sudo iptables -P INPUT DROP”



1. Просмотрим список правил, используя команду “sudo iptables -L”



1. Добавим еще несколько правил для блокировки наиболее распространенных атак. Для начала нужно заблокировать нулевые пакеты:

“sudo iptables -A INPUT -p tcp –tcp-flags ALL NONE -j DROP”



1. Следующее правило отражает атаки syn-пакетами без состояния NEW:

“sudo iptables -A INPUT -p tcp ! –syn -m state –state NEW -j DROP”



Теперь фаервол не будет принимать входящих пакетов с tcp-флагами. Нулевые пакеты, по сути, разведывательные. они используются, чтобы выяснить настройки сервера и определить его слабые места.

1. Далее нужно защитить сервер от разведывательных пакетов XMAS.

“sudo iptables -A INPUT -p tcp –tcp-flags ALL ALL -j DROP”

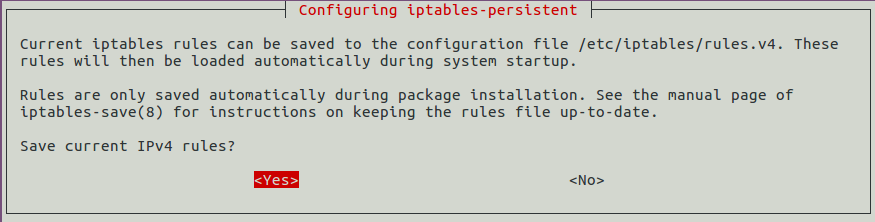


По умолчанию все не сохраненные правила действуют до следующей перезагрузки сервера, сразу же после перезагрузки несохраненные правила будут потеряны. Самый простой способ – загрузить пакет iptables-persistent “sudo apt-get install iptables-persistent”.

Изображение выглядит как текст

Автоматически созданное описание

Во время инсталляции пакет уточнит, нужно ли сохранить текущие правила для дальнейшей автоматической загрузки, если текущие правила были протестированы и соответствуют всем требованиям, их можно сохранить.



Теперь сервер защищен от некоторых общих атак, которые ищут его уязвимости.

1. Со второй виртуальной машины проводим XMAS сканирование и зафиксируем результат:

Изображение выглядит как текст

Автоматически созданное описание

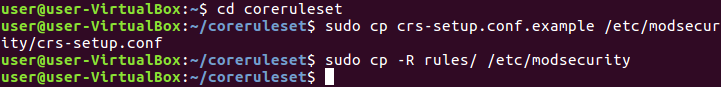
Часть 2. WAF

1. Для более серьезной защиты загрузим готовый пакет правил OWASP на атакуемую машину:

Изображение выглядит как текст

Автоматически созданное описание

И устанавливаем их в модуль защиты:



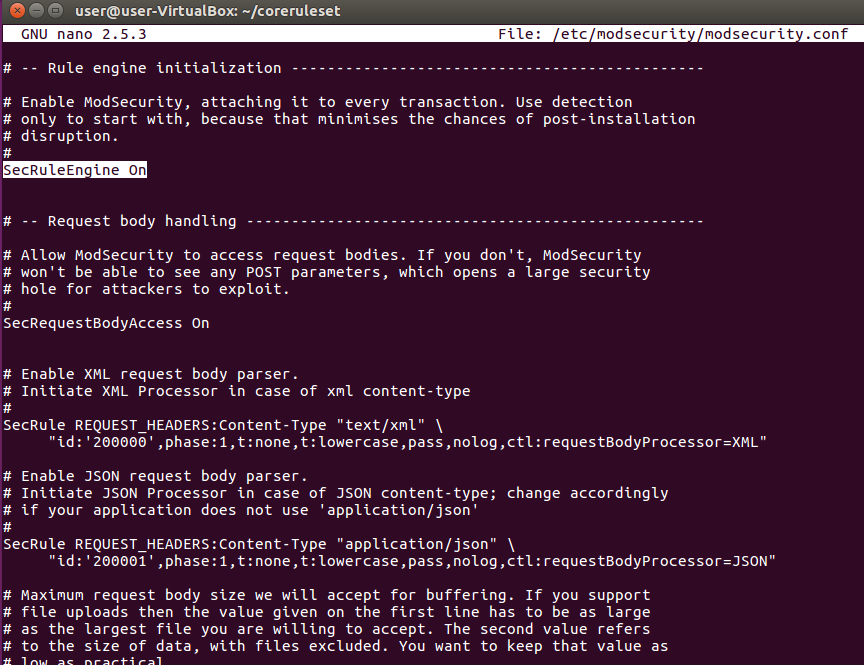
1. Установка ModSecurity включает в себя конфигурационный файл, который нужно переименовать:



1. Стандартный конфигурационный файл настроен на DetectionOnly, то есть, фаервол только отслеживает логи, при этом ничего не блокируя. Чтобы изменить это поведение, отредактируем файл “modsecurity.conf”:

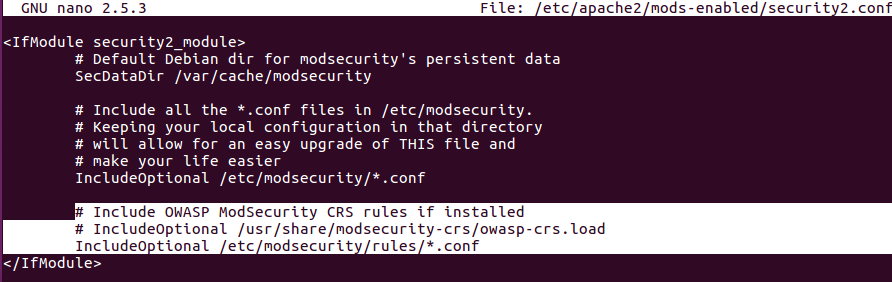
“sudo nano /etc/modsecurity/modsecurity.conf”

Находим в файле строку: «SecRuleEngine DetectionOnly». И изменяем её так: «SecRuleEngine On».



1. Чтобы подгрузить эти готовые правила и правила OWASP, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого отредактируем файл security2.conf:

“sudo nano /etc/apache2/mods-enabled/security2.conf”



1. Чтобы новые правила вступили в исполнение, нужно перезапустить Apache:

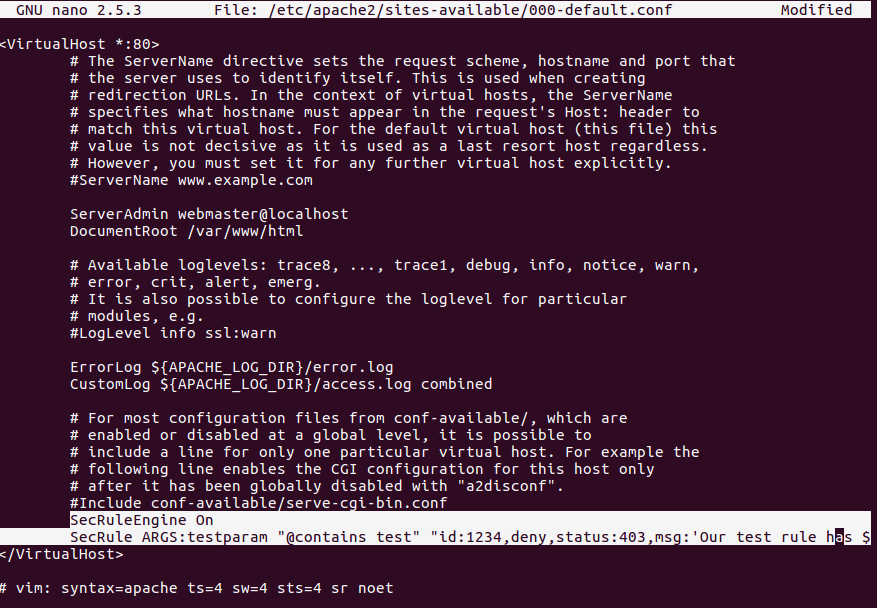


Тестирование модуля защиты WAF.

1. Открываем для редактирования дефолтный файл конфигурации сайта 000-default.conf

“sudo nano /etc/apache2/sites-available/000-default.conf”

Перед строкой “/VirtualHost” добавляем следующие 2 строки:



Последняя строка, это правило определяющее, что при попытке запроса с параметром

«testparam» запрос будет отклонен с кодом 403.

1. На атакующей машине выполняем следующий запрос:

Изображение выглядит как текст

Автоматически созданное описание

Видим, что все правильно, запрос вернул результат «403 Forbidden»

1. Для закрепления результата выполняем сканирование утилитой nmap с опцией

детектирования WAF. Выполните на атакующей машине:

“nmap –p 80 -sV --script=http-waf-fingerprint IP\_атакуемой\_машины”

Изображение выглядит как текст

Автоматически созданное описание

1. На атакуемой машине в лог-файле /var/log/apache2/modsec\_audit.log данное сканирование будет полностью зафиксировано.

“sudo nano /var/log/apache2/modsec\_audit.log”

Изображение выглядит как текст

Автоматически созданное описание

Контрольные вопросы:

1. Межсетевой экран — это программно-аппаратный комплекс, который предназначен для защиты компьютерных сетей от несанкционированного доступа и атак извне.
2. Межсетевой экран используется для обеспечения безопасности компьютерных сетей путем контроля трафика, проходящего через границы сети. Он может блокировать или разрешать доступ к определенным ресурсам, фильтровать сетевой трафик, а также проводить множество других операций, связанных с обеспечением безопасности сети.
3. Netfilter — это система фильтрации пакетов в ядре Linux, которая позволяет проводить межсетевой экран и другие операции с пакетами. Она работает на уровне ядра операционной системы и использует набор таблиц для принятия решений о том, как обрабатывать входящие и исходящие пакеты.
4. Таблицы межсетевого экрана Netfilter — это структуры данных, которые используются для принятия решений о том, как обрабатывать пакеты. В Netfilter есть пять таблиц: raw, filter, mangle, nat и security. Каждая из таблиц используется для определенных целей, таких как фильтрация, изменение заголовков пакетов, перенаправление пакетов на другой адрес и т.д.
5. Правила межсетевого экрана — это наборы инструкций, которые определяют, как обрабатывать входящие и исходящие пакеты. Они могут разрешать или запрещать доступ к определенным ресурсам, изменять заголовки пакетов, перенаправлять пакеты на другой адрес и т. д.
6. Правила для межсетевого экрана можно создавать с помощью утилиты Iptables. Для создания правил необходимо использовать команды Iptables в командной строке Linux. Например, для разрешения доступа к порту 80 можно использовать следующую команду: "iptables -A INPUT -p tcp --dport 80 -j ACCEPT". Эта команда добавит правило в таблицу INPUT, которое разрешит доступ к порту 80.
7. Для сохранения правил межсетевого экрана (Firewall) в Linux для последующей автозагрузки можно воспользоваться утилитой "iptables-persistent".
8. WAF — это межсетевые экраны, работающие на прикладном уровне и осуществляющие фильтрацию трафика Web-приложений. Эти средства не требуют изменений в исходном коде Web-приложения и, как правило, защищают Web-сервисы гораздо лучше обычных межсетевых экранов и средств обнаружения вторжений.
9. Для настройки правил в WAF mod\_security можно использовать файл конфигурации "modsecurity.conf". В этом файле можно определять правила фильтрации для различных типов запросов и ответов, а также определять дополнительные параметры настройки WAF.