Инструмент Maltego, Recon-ng, FOCA, Утилиты командной стоки Windows. HTTrack и Metasploit

**Лабораторная работа №10. DDoS (Distributed Denial of Service) - основные особенности их организации и защиты от них.**

**Цель:** Получить теоретические и практические навыки о принципах осуществления DDoS атак и защиты от них.

**Основные теоретические сведения**

**DoS/DDoS**

DoS (от англ. Denial of Service — отказ в обслуживании) — хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т.д.). Но чаще это мера экономического давления: простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.

Если атака выполняется одновременно с большого числа компьютеров, говорят о DDoS-атаке (от англ. Distributed Denial of Service, распределённая атака типа «отказ в обслуживании»).

Для обнаружения распределённых сетевых атак типа «отказ в обслуживании» и защите от них необходимо классифицировать их и знать принципы их работы.

В данной работе критерием для классификации рассмотрим объект, на который нацелена атака. В таком случае, получается четыре основных класса атак, соответствующих уровням модели ISO OSI.

**Канальный уровень (L2)** — атаки направлены на исчерпание ёмкости сетевого канала. В следствие этого лишается доступ сервера к внешней сети. Для реализации используются объёмные потоки трафика. На данным момент измеряются в Гб/с. Во время этой атаки обрабатывать трафик необходимо на стороне провайдера, дата-центра. С помощью BGP Flow Spec фильтруется часть атак по сигнатурам пакета. Amplification атаки отсекаются по порту.

**Сетевой уровень (L3)** — атаки направлены на нарушение работы элементов сетевой инфраструктуры. Необходим ручной анализ сетевой инфраструктуры. Если своей автономной системы нет, то борьба с атаками данного класса ведется провайдером или дата-центром. Желательно сотрудничество с ними.

**Транспортный уровень (L4)** — атаки направлены на эксплуатацию слабых мест TCP-стека. В TCP протоколе используется таблица открытых соединений. Атаки именно на неё составляют этот класс. Необходим постоянный анализ поведения TCP-стека, TCP-клиентов, TCP-пакетов. Эвристический анализ.

**Прикладной уровень (L7)** — атаки направленны на нарушение работы Web-приложения. Атаки этого класса характеризуются большим разнообразием. Исчерпывают ресурсы сервера. Необходим поведенческий и корреляционный анализ, мониторинг ресурсов сервера. Необходима оптимальная настройка сервера под решаемые им задачи. Полностью автоматизировать борьбу с данным классом атак почти невозможно.

**Медленная атака. SlowLoris**

**Уровень атаки: транспортный уровень (L4).**

**Описание и принцип работы:**

Атака Slowloris устанавливает много открытых соединений на сервере с помощью постоянной отправки незавершенных HTTP-запросов. В определенные моменты времени Slowloris отправляет следующие HTTP заголовки для каждого запроса, но не завершает соединение. Если запросы посылаются с оптимальной периодичностью, сервер начинает ожидать завершения открытых соединений. В данном случае ресурсы сервера остаются относительно свободными, но сам сервер перестаёт обслуживать новые подключения.

Дело в том, что веб-серверы Apache 1.x, Apache 2.x, dhttpd, GoAhead WebServer и Squid поддерживают ограниченное число одновременно открытых подключений. Но Slowloris не представляет угрозы для серверов IIS, lighttpd, NGINX. Они имеют эффективные механизмы распределения нагрузки и используют worker pool — «пулы рабочих потоков», которые позволяют удерживать любое количество открытых соединений при наличии свободных ресурсов.

**Медленная атака. Slow HTTP POST/GET**

**Уровень атаки: транспортный уровень (L4).**

**Описание и принцип работы:**

Атака основана на уязвимости в протоколе HTTP. Slow HTTP POST атака отправляет POST заголовок с полем «Content-Length». Веб-сервер понимает, какой объём данных он должен получить. После этого с очень низкой скоростью передаётся тело POST сообщения. Это позволяет задействовать ресурсы сервера длительное время, и в последствии помешать обработке других запросов. Атака опасна для веб-серверов Microsoft IIS и Apache и NGINX со стандартными настройками в рамках протоколов HTTP, HTTPS, подключений SSL, VPN. Также атака может быть настроена для работы с SMTP и DNS-серверами.

Данный тип атаки на отказ в обслуживании, можно организовать через proxy. Трафик данной атаки схож с легитимным трафиком.

**Медленная атака. Sockstress**

**Уровень атаки: транспортный уровень (L4).**

**Описание и принцип работы:**

Атака заключается в следующем. Если на веб-сервере есть объект, размер которого больше send buffer, выделенного ядром для соединения. То можно заставить ядро не принимать данные, а сервер будет пробовать отправить кусок данных, занимая стек соединений, ресурсы процессора и память. При большом количество подобных соединений TCP-стек заполниться и не будет открывать новые соединения.

**Примеры:**

Отправить в пакете размер окна равный нулю, то есть нет места для получения данных. Скрипт Sockstress отправляет такие пакеты и считает время, когда их отправлять, чтобы не загрузить persist timer.

Создать сокет с малым объёмом receive buffer на клиенте. Отправлять HTTP-запрос на объект сайта больший по размеру, чем буфер. И периодически считываем пару байт из receive buffer. Сервер будет пытаться отправлять данные и занимать ресурсы.

**Атака произвольными пакетами. HTTP-Flood**

**Уровень атаки: канальный уровень (L2), прикладной уровень (L7).**

**Описание и принцип работы:**

В основе этой атаки лежит механизм отправления максимального числа HTTP запросов на 80-й порт веб-сервера. Целью атаки может быть корень сервера или ресурсоёмкий элемент. В результате данной атаки возможно прекращение предоставления услуг по HTTP, и затруднен доступ легитимных пользователей к сайту. Распознать атаку можно с помощью выявления быстрого роста количества запросов к некоторым элементам веб-сервера и логов сервера.

**Атака произвольными пакетами. UDP-Flood**

**Уровень атаки: канальный уровень (L2), прикладной уровень (L7).**

**Описание и принцип работы:**

UDP flood атака основана на отправке большого количества UDP-пакетов на некоторые порты сервера. Он должен определить приложение для каждого полученного пакета, удостоверится в его неактивности и отправить в ответ ICPM-сообщение «недоступен». В итоге вырастут затрачиваемые ресурсы атакуемого сервера и полоса пропускания заполнится UDP-пакетами. В UDP протоколе нет механизма проверки отправителя пакетов, тем самым злоумышленник может подменить IP-адреса и обеспечить анонимность.

**Атака произвольными пакетами. SYN-Flood**

**Уровень атаки: канальный уровень (L2), транспортный уровень (L4).**

**Описание и принцип работы:**

SYN Flood атака использует механизм рукопожатия в протоколе TCP. Работает следующим образом. Посылается пакет с флагом SYN на атакуемый сервер. Он вынужден отправить в ответ пакеты с флагами SYN+ACK. Злоумышленник игнорирует SYN+ACK пакеты сервера и не высылает в ответ пакет ACK. Либо подделывает IP-адрес SYN пакета, чтобы ответный SYN+ACK отправляется на некорректный адрес. Цель данной атаки — заполнение TCP стека множеством полуоткрытых соединений, в следствие чего сервер перестают устанавливать соединения с новыми клиентами. Запросы на соединение полученные сервером хранятся в стеке с определенным размером, который зависит от операционной системы. Они находятся в стеке, пока сервер не получит информацию об установленном соединении от клиента.

**Атака произвольными пакетами. ICMP-Flood**

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

Данный тип флуда направлен на сетевое оборудование. Принцип данной атаки заключается в том, что ICMP-пакет при небольшом размере самого запроса требует от устройств значительно большего объёма работы. То есть, при отправлении сравнительно небольшого объёма ICMP запросов возникает перегрузка сетевого оборудования и значительная часть легитимных запросов теряется. Злоумышленник, меняет IP-адрес источника, отправляет ICMP Echo Request пакет к определённым компьютерам, входящим в бот-нет. Они отвечают ICMP Echo Reply пакетом, посылая его на изменённый IP-адрес. Для увеличения мощности атаки используют локальные сети (LAN) с включенной опцией направленной широковещательной рассылки (directed broadcast).

**Атака с помощью SSL**

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Secure Sockets Layer (SSL) — это протокол безопасности для защиты целостности сети связи и передачи данных. SSL может зашифровать подключение к сети на транспортном уровне. Процессы шифрования протокола SSL, расшифровки и обмена ключами потребляют огромное количество системных ресурсов. Существует два типа атака основанных на протоколе SSL.

Первый тип эксплуатирует механизм рукопожатия, который исчерпывает ресурсы атакуемого сервера. Злоумышленник посылает некорректные SSL данные на сервер, на обработку которых затрачивается большое количество вычислительной мощности.

Второй тип использует функцию повторного подтверждения соединения — SSL Renegotiation.

Установка безопасного соединения и повторное подтверждение SSL затрачивают в разы больше вычислительной мощности на сервере, чем на стороне клиента. Благодаря этому возможно осуществление атаки и истощение ресурсов атакуемого сервера. Как правило, HTTPS расшифровывается глубоко внутри организационной сети, где серверы и модули более уязвимы к вредоносному трафику. Также злоумышленники используют этот протокол для обхода механизмов безопасности. Таким образом, возможно туннелирование других атак.

**Атака почтового сервера. SMTP-Flood**

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

В атаках этого типа злоумышленник пытается установить соединение с почтовым сервером и отправляет произвольные письма на сгенерированные случайным образом адреса, либо бездействует до истечения тайм-аута, удерживая соединение открытым. Каждое SMTP соединение утилизирует часть ресурсов сервера, тем самым атакующий пытается вызвать отказ в обслуживании.

Технология заключается в том, что когда мы пишем e-mail получателя несуществующий, то SMTP server оповещает нас с помощью e-mail(отправителя), что письмо не дойдёт. Мы посылаем на SMTP server очень много запросов (пишем письма) и указываем все e-mail не валидные, а отправителя(объект атаки). И к нему будут приходить уведомления, что письмо не дошло. С нескольких SMTP server будет отправляться трафик на объект атаки.

**Некорректные пакеты/фрагменты. UDP fragment flood**

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Данный тип атаки основан на отправки UDP датаграмм, которые случайным образом ссылаются на датаграммы отсутствующие в потоке. Это приводит к увеличению потребления памяти на атакуемом сервере. При атаке UDP Fragment Flood, злоумышленники посылают UDP пакеты большого размера, для истощения пропускной способности канала.

**Некорректные IP-фрагменты**

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Данный тип атак эксплуатирует уязвимости в поддержке фрагментации пакетов протокола IP. Одна из атак этого типа — это пересечение IP-фрагментов. Она реализуется с помощью уязвимости операционной системы, которая заключается в сборке фрагментированных IP- пакетов. В процессе сборки образуется цикл по принятым фрагментам. Затем из них копируется информативная часть и передаётся на IP уровень. Разработчики предусмотрели проверку на чрезмерный объем копируемой информации, но не ввели проверку на копирование фрагмента отрицательной длины. Копирование блока информации отрицательной длины равносильно копированию очень большого блока информации. Это приводит к затиранию большого участка памяти и к нарушению работы ЭВМ (электронная вычислительная машина). Существует две программы с небольшими отличиями в константах механизма, который осуществляет пересечение IP-фрагментов: newtear и teardrop. Они отправляют пакеты с заданного IP-адреса на любой порт, независимо, открыт он или закрыт. Еще один вариант данной атаки - bonk. После сборки фрагментов в пакете остаются пустые места. Это приводит к сбою ядра операционной системы и нарушению работы электронных вычислительных средств. Данные уязвимости присутствуют в старых версиях ОС(операционных систем) Windows и Linux. На сегодняшний день большинство сетевых ОС защищены от сбоев в работе, вызванных данной атакой.

**Неверные значения в заголовках пакетов**

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Данные атаки нацелены на определённые приложения и операционные системы, которые неправильно обрабатывают некорректные значения в заголовках пакетов. Пример такой атаки — Land attack. В ходе неё злоумышленник устанавливает в пакете одинаковый IP-адрес для источника и приёмника. Это приводит к зацикливанию установки соединения сервера с самим собой.

**Amplification атаки**

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

В основе данных атак лежит отсутствие проверки отправителя в UDP протоколе. Ответ посылается адресату, указанному в заголовках пакета. Злоумышленник может подменить свой IP-адрес на IP-адрес атакуемого сервера в заголовках отправляемых пакетов. Также суть атаки заключается в многократном превышении объёма ответа по сравнению с запросом. Таким образом, злоумышленник может анонимно организовывать атаки с огромным объёмом трафика. Службы, работающие по UDP протоколу: DNS, NTP, SNMP, rsyslog и многие другие, могут использоваться для реализации атаки. Дело в том, что сетевые устройства с этими службами встречаются в сети повсеместно. Службы включены по умолчанию и часто некорректно настроены.

В таблице 1 представлены типы amplification атак проведённых в ходе исследования. В ней отображается по какому протоколу осуществляются атаки, коэффициент их усиления и уязвимая команда, используемая для реализации атаки.

Таблица 1 — Amplification атаки

|  |  |  |
| --- | --- | --- |
| **Протокол** | **Коэффициент усиления** | **Уязвимая команда** |
| DNS | x28-x92 | DNS server request |
| NTP | x994 | Monlist request |
| SNMPv2 | x29 | GetBulk request |
| CharGEN | x350 | Character generation request |
| BitTorrent | x4 | File search |
| RIPv1 | x131 | Malformed request |
| SSDP | x31 | SEARCH request |
| NetBIOS | x4 | Name resolution |
| Quake Network Protocol | x64 | Server info exchange |
| Steam Protocol | x5.5 | Server info exchange |

￼￼￼￼￼￼￼￼￼￼￼￼￼￼ DNS, NTP, SNMPv2 — протоколы для получения информации о доменах, синхронизации времени и сетевого управления. Часто встречаются в сети. NetBIOS и SSDP — протоколы в Windows. CharGEN — старый тестовый сервис, но его до сих пор можно встретить на различных системах. BitTorrent — протокол для обмена файлами. Quake Network Protocol и Steam Protocol — протоколы компьютерных игр.

**NTP amplification атака**

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

Злоумышленник отправляет запрос monlist с IP-адресом атакуемого сервера к NTP-серверу. Ответ monlist включает в себя список 600 последних клиентов ntpd. Сущность амплификации заключается в том, что нарушитель отправляет небольшой запрос к уязвимому серверу и с него на атакуемый сервер отправляется большой поток UDP трафика. Уязвимый NTP-сервер является невольным промежуточным звеном атаки. Ntpd до версии 4.2.7p26 подвержены атаке.

**DNS Amplification атака**

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

Атака основана на том, что нарушитель отправляет запрос уязвимому DNS-серверу с IP-адресом атакуемого сервера. DNS-сервер отправляет ответ, размер которого во много раз превышает запрос, жертве. Таким образом, исчерпывается канальная ёмкость атакуемого сервера.

Можно выделить ключевые моменты атаки:

1. Эффект отражения: подмена IP-адреса позволяет перенаправить ответы от всех DNS-серверов на атакуемый сервер.
2. Коэффициент усиления атаки: (amplification factor): он может принимать значения от 28 до 92. То есть на 1 байт запроса — совокупность DNS-серверов отправит 28-92 байт ответа. Это обеспечивает кратное увеличение объёма трафика.
3. Проблема «open resolver»: это неправильно настроенный или старой версии DNS-сервер. Он разрешает получать запросы из сторонних сетей, выполняя рекурсивные запросы для них, и отправлять ответы без необходимых предварительных проверок.

**HTTP flood с помощью сервисов**

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

WordPress сайт с включенным Pingback, можно использовать для проведения HTTP flood атаки на другие сайты. Они отправляют множество запросов к атакуемому сайту со случайными параметрами («?a=a» и др.), с помощью которых обходится кэширование страницы. Эта операция быстро расходует ресурсы атакуемого сервера и нарушает его работу. Злоумышленник может использовать большое количество обычных WordPress сайтов для DDoS атаки и не бояться быть обнаруженным с помощью Pingback запросу к файлу XML-RPC. Google использует FeedFetcher для кэширования любого контента в Google Spreadsheet, вставленного через формулу =image(«link»). Если в клетку таблицы вставить формулу =image(«<http://target/file.pdf>»), то Google отправит FeedFetcher скачать этот PDF файл и закэшировать для дальнейшего отображения в таблице. Но если добавлять случайный параметр к URL картинке (от «?r=1» до «?r=1000»), FeedFetcher будет скачивать её каждый раз заново. Это приведёт к исчерпанию лимита трафика атакуемого сервера. Злоумышленник может запустить массированную HTTP GET flood атаку на веб-сервер, используя браузер с одной открытой вкладкой.

**Практическая часть.**

Для эффективной защиты от распределённых сетевых атак типа «отказ в обслуживании» требуется своевременное обнаружение начала атаки. Чтобы обнаружить DDoS-атаку необходимо накапливать статистические данные о трафике сервера в сети, работающем в штатном режиме. При знании среднестатистических значений характеристик сервера, можно отследить появление аномалии трафика в сети.

Для выполнения данной лабораторной работы необходимо создать две виртуальные машины или объединить в локальную сеть два компьютера. Одна машина будет жертвой, другая злоумышленником.

На машине жертве необходимо установить:

* Web-сервер (Apache, NGINX);
* Средства мониторинга и анализа ресурсов сервера и сетевого трафика (iptraf, ksysguard или другие);

**Примечание**

Дополнительно ознакомьтесь со следующими ресурсами [1](https://nginx.org/ru/) , [2](https://httpd.apache.org/) , [3](http://help.ubuntu.ru/wiki/%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) , [4](http://proubuntu.com.ua/2011/11/17/system-monitoring-ubuntu.html) .

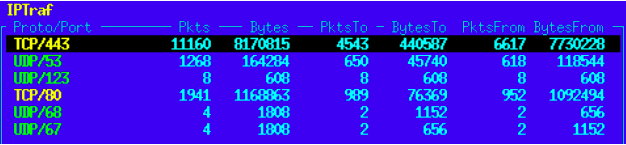
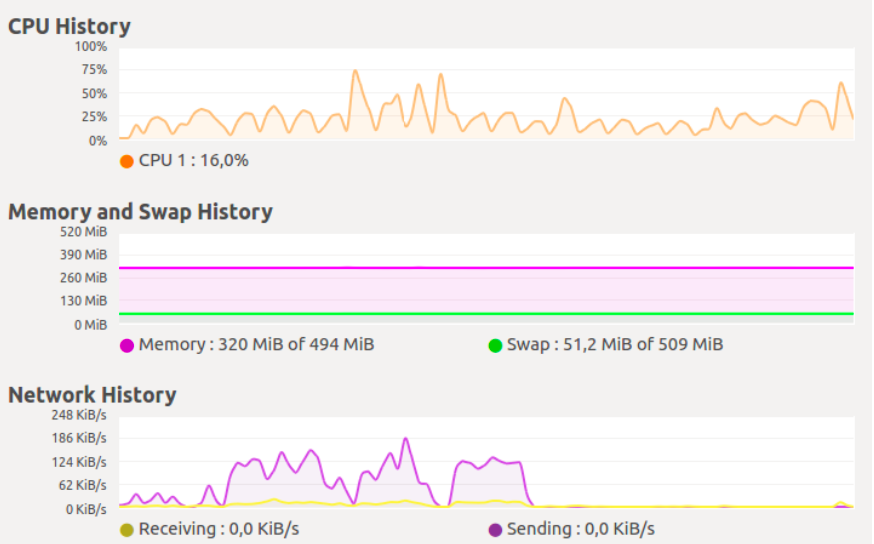
На машине злоумышленника необходимо установить:

* Программное обеспечение и скрипты для проведения DDoS атак;
* Уязвимые сервисы для усиления DDoS атак (dns, ntp, chargen);

**Примечание**

Чтобы сделать сервис уязвимым, ознакомьтесь со следующими ресурсами [1](http://blogerator.ru/page/dns-amplification-ddos-otrazhenie-bind-dns-reflect-attack-chast-2) , [2](https://habrahabr.ru/post/209438/) .

На машине-жертве фиксируются системные характеристики, такие как: загрузки процессора, объем оперативной памяти и характеристики сетевого трафика, такие как: количество пакетов, открытые соединения.



**Скриншоты мониторинга и трафика**

**Примечание**

Изменение указанных характеристик не обязательно обозначает атаку на сервер, но показывает отклонение от среднестатистических показателей работы в штатном режиме. Для большинства DDoS-атак можно выделить аномалии и признаки, соответствующие только им.

**В ходе данной лабораторной работы необходимо провести каждый из представленных типов DDoS атак, отследить аномалии в использовании ресурсов сервера и сетевом трафике, проверить признаки DDoS атаки и, если они подтвердились, защитить сервер от атаки с помощью настройки сервера, добавление правил для межсетевого экрана, фильтрации пакетов по сигнатурам.**

**Пошаговый алгоритм методики:**

1. Обнаружение аномалии в наблюдаемых системных характеристиках и характеристиках сетевого трафика.
2. Проверить присутствие признаков атаки в трафике.
3. Внести изменения в настройки сервера под конкретную атаку и внести правила для фильтрации.

Для каждого типа атак в сети интернет необходимо найти по несколько программ и скриптов, которые реализуют данный вид атаки.

* Для атаки HTTP flood: goldeneye, ddosim, DAVOSET, HULK, LOIC, BBHH.
* Для медленных атак: r-u-dead-yet, slowhttptest, pyloris, sockstress, torshammer. Для UDP flood: LOIC, fudp, hping3.
* Для SYN flood и атак на TCP-стек: sprut, sitekiller, mummy, hping3.
* Для ICMP flood: hping3.
* Для land атаки: fudp, hping3.
* Для атак с использованием SSL: thc-ssl-dos.
* Для amplification атак: saddam, chargen\_amp.

**Примечание**

Скачать набор программного обеспечения можно по этой [ссылке](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/_static/DDoS.rar) .

**Предупреждение**

Ознакомьтесь со справочной документацией используемых Вами программ и скриптов.

**Дополнительная информация**

**Медленные атаки**

**Аномалия:** Кратковременный скачок нагрузки процессора, повышение исходящего трафика и используемой оперативной памяти.

**Признак:** В TCP-стеке большое количество соединений со статусом ESTABLISHED.

**Защита:** Если установлен apache, то поставить перед ним кэширующий сервер NGINX. Установить и настроить балансировщик нагрузки. На apache можно установить mod\_security — firewall с готовыми правилами от OWASP и mod\_reqtimeout — установка таймаутов и минимальной скорости передачи данных для получения запросов.

Добавить правила межсетевого экрана:

* ограничение количества соединений с одного IP-адреса;

*# iptables -A INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 30 -j DROP*

* заблокировать IP после 10 подключений к порту 80 в течение 30 секунд;

*# iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent — set*

*# iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 30 --hitcount 10 -j DROP*

Также NGINX имеет функцию кэширования клиентского запроса перед отправкой на бэкэнд — client\_body\_buffer\_size. Бэкэнд получит запрос, только когда он полностью загрузится.

**Атака произвольными пакетами.**

**Аномалия:** Резкое увеличение нагрузки процессора, входящего и исходящего-HTTP трафика.

**Признак:** На сервер приходит много однотипных пакетов, резко выросло количество обращений к ресурсоёмкому элементу сайта.

**Защита:** Борьба с HTTP flood ведётся с помощью усовершенствования работы веб-сервера и баз данных. Также меры включают в себя использование обработки подключений методом epoll, увеличение количества соединений, отключение тайм-аута на закрытие подключений keep-alive.

worker\_processes 2; worker\_rlimit\_nofile 8000; events {

worker\_connections 4000;

use epoll; }

Установить на NGINX модуль ngx\_http\_limit\_req\_module и блокировать IP-адреса, которые стали получать ответ «Service unavailable».

http {

limit\_req\_zone $binary\_remote\_addr zone=zne:15m rate=3r/s; server {

location / {

limit\_req zone=zne burst=5;

}

**Атака произвольными пакетами. UDP-Flood**

**Аномалия:** Резкое увеличение нагрузки процессора, входящего UDP-трафика.

**Признак:** Со всех открытых UDP-портов идёт исходящий трафик.

**Защита:** Необходимо выставить ограничение на количество подключений к открытым портам и закрыть неиспользуемые порты с помощью межсетевых экранов.

Добавить правила межсетевого экрана:

* ограничить количество подключений;

*# iptables -I INPUT -p udp --dport 53 -j DROP -m iplimit --iplimit-above 1 • разрешить подключение только доверенным IP-адресам;*

*# iptables -A OUTPUT -p udp --dport 53 -d 8.8.4.4 -j ACCEPT*

* блокировать все остальные порты.

*# iptables -A OUTPUT -p udp -j DROP*

**Атака произвольными пакетами. SYN-Flood**

**Аномалия:** Кратковременный скачок используемых ресурсов процессора и многократное увеличение входящего и исходящего трафика, и их количество равно.

**Признак:** В TCP-стеке появляется большое количество полуоткрытых соединений со статусом SYN\_RECV.

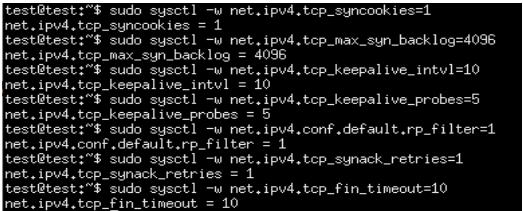
**Защита:** Современные реализации TCP протокола и некоторые межсетевые экраны имеют механизм защиты от SYN flood атак.

Принцип его действия заключается в следующем:

1. Серверу отправляется запрос на установление соединения, механизм регистрирует его в таблице.
2. После ответа сервера о подтверждении запроса на соединение механизм отправляет пакет серверу с подтверждением соединения и отправляет ответный пакет клиенту. В этот момент в механизме запускается таймер, отсчитывающий время на ответ от клиента.
3. При получении пакета с подтверждением соединения механизм считает запрос валидным, остановит таймер и отправит его на сервер.
4. Если пакет не пришёл в установленное время, механизм отправляет серверу запрос на удаление информации о соединении.

Принцип действие механизма основан на контроле времени задержки ответа клиента. При слишком коротком тайм-ауте будут обрываться легитимные пользователи, при длинном будут накапливаться соединения, что может привести к серьёзным последствиям.

Также можно использовать SYN cookie, или ограничить количество запросов на новые подключения от конкретного пользователя за конкретный период времени. На Рисунке 1 представлен пример настройки ядра операционной системы для защиты от SYN-Flood с помощью команды sysctl.



**Атака произвольными пакетами. ICMP-Flood**

**Аномалия:** Очень сильная загрузка процессора и сильное увеличение входящего и исходящего ICMP трафика, равного по значениям.

\*\* Признак:\*\* Множество пакетов передаётся по протоколу ICMP.

**Защита:**

Для противодействия атаке возможны следующие меры:

* отключить ответы на ICMP-запросы;

*#sysctl net.ipv4.icmp\_echo\_ignore\_all=1*

* понизить приоритет обработки ICMP-сообщений;
* отбросить или фильтровать ICMP-трафик межсетевым экраном;

*# iptables -A INPUT -p icmp -j DROP --icmp-type 8*

* увеличить очередь обрабатываемых подключений.

**Атака с помощью SSL**

**Аномалия:** Резкое повышение используемых системных ресурсов.

**Признак:** Появляется HTTPS трафик. Большое количество обращений к SSL серверу.

**Защита:** Можно установить правила разрыва соединения с пользователем, выполняющего функцию повторного подтверждения больше заданного количества раз в определённый период времени. Можно использовать контроллер доставки приложений (ADC), чтобы выгрузить SSL с сервера и использовать Web Application Firewall (WAF), для просмотра трафика на наличие атак. Но грамотно спланированные атаки могут превысить количество подключений к этим устройствам и нарушить их работу. Необходимо проводить поведенческий анализ. Целью этих методов является снижение объема трафика атаки пока ресурсы сервера не смогут эффективно бороться с ним. Данный тип атак не отличается поведением от легитимного трафика на сетевом уровне. Из-за этого борьба не всегда эффективна и механизмы ослабления склонны к ложным срабатываниям.

**Amplification атаки**

**Аномалия:** Огромное увеличение входящего трафика, которое может достигать до 600 Гб/с.

**Признак:** Увеличение входящих UDP-пакетов на порты, используемые уязвимыми сервисами. Например, NTP 123 порт, DNS 53 порт, СharGEN 19 порт.

**Защита:** Такие атаки фильтруются по порту источника и сигнатурам пакетов, так как для каждой amplification атаки используется определённая уязвимая команда сервиса, сигнатура которой известна.

В межсетевом экране необходимо добавить правила:

* закрыть неиспользуемые UDP порты;

*# iptables -A INPUT -p udp -j DROP*

* разрешить подключение к Х UDP порту только с IP-адреса используемого сервиса;

*#iptables -A INPUT -p udp --dport Х -d x.x.x.x -j ACCEPT*

* пропускать только пакеты, которые имеют определённую сигнатуру 0x00=0x00000000.

*# iptables -A INPUT -p udp --dport Х -m u32 --u32 «0x00=0x00000000» -j ACCEPT*

**Задания к лабораторной работе:**

1. Установить необходимое ПО и настроить сервера.
2. Сделать скриншоты мониторинга ресурсов сервера и сетевого трафика в штатном режиме.
3. Провести несколько атак из списка.
4. Сделать скриншоты мониторинга для каждой проведённой атаки.
5. Написать какие аномалии вы наблюдали в используемых ресурсах сервера, характеристиках сервера, и что является признаком атаки.
6. Защитить сервер от атак и сделать скриншоты результатов.
7. Оформить отчёт по проделанной работе.

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Вопросы к лабораторной работе**

1. Принцип работы каждой из сетевых атак типа «отказ в обслуживании» ?
2. Какие аномалии в сетевом трафике и ресурсах сервера вы заметили в каждой̆ атаке ?
3. Что является признаком конкретной атаки ?
4. В чем отличие DDoS от DoS?
5. Что такое UDP-флуд?
6. Как защититься от различных атак DoS/DDoS?
7. Что такое SYN-флуд? Укажите решение защиты от SYN-флуда с помощью iptables.
8. Как определить, что осуществляется атака SYN-флуд?
9. Как осуществить защиту от HTTP-флуда?
10. Как работают атаки типа SlowLoris?
11. Как совершить DDoS-атаку на почтовый сервер и в дальнейшем реализовать его защиту?
12. Как работают атаки типа SSL? Как определить, что осуществляется атака?
13. Перечислите Amplification атаки с коэффициентами их усиления.
14. Опишите теоретически возможность расследования DoS/DDoS атак, установление личности атакующего, определение источников атак.

**Лабораторная работа №11. Iptables, WEB APLICATION FIREWALL**

**Основные теоретические сведения**

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

**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.

| *Основные команды Iptables* | | |
| --- | --- | --- |
| **Параметр** | **Описание** | **Пример** |
| –append (-A) | Позволяет добавить в указанную цепочку и таблицу заданное правило, помещаемое в КОНЕЦ списка | iptables -A FORWARD критерии -j действие |
| –delete (-D) | Позволяет удалить заданное номером или каким-либо правилом правило. В первом примере удаляются все правила с номерами 10,12 во всех цепочках, в таблицах filter. | iptables -D 10,12 iptables -t mangle -D PREROUTING критерии -j действие |
| –rename-chain (-E) | Изменить имя цепочки. | iptables -E OLD\_CHAIN NEW\_CHAIN |
| –flush (-F) | Очищает все правила текущей таблицы. Ко всем пакетам, относящимся к уже установленным соединениям, применяется терминальное действие ACCEPT — пропустить | iptables -F |
| –insert (-I) | Добавляет заданное правило в соответствии с номером. | iptables -I FORWARD 5 критерии -j действие |
| –list (-L) | Позволяет просматривать существующие правила (без явного указания таблицы - отображается таблица filter всех цепочек). | iptables -L |
| –policy (-P) | Позволяет устанавливать стандартную политику для заданной цепочки. | iptables -t mangle -P PREROUTING DROP |
| –replace (-R) | Заменяет заданное номером правило на заданное в критериях. | iptables -R POSROUTING 7 | критерии -j действие |
| –delete-chain (-X) | Удалить ВСЕ созданные вручную цепочки (оставить только стандартные INPUT, OUTPUT…) | iptables -X |
| –zero (-Z) | Обнуляет счетчики переданных данных в цепочке. | iptables -Z INPUT |
| –line-numbers | Указывать номера правил при выводе (может использоваться совместно с -L). | iptables -L –line-numbers |
| –help (-h) | Помощь | Iptables –help |
| -t таблица | Задает название таблицы, над которой необходимо совершить действие. В примере сбрасывается таблица nat во всех цепочках. | iptables -t nat -F |
| –verbose (-v) | Детальный вывод. | iptables -L -v |
|  | **Основные правила отбора пакетов** |  |
| –protocol(сокр. -p) | Определяет протокол транспортного уровня. Опции tcp, udp, icmp, all или любой другой протокол определенный в /etc/protocols | iptables -A INPUT -p tcp |
| –source(-s, –src) | IP адрес источника пакета. Может быть определен несколькими путями:Одиночный хост: host.domain.tld, или IP адрес: 10.10.10.3 Пул-адресов (подсеть): 10.10.10.3/24 или 10.10.10.3/255.255.255.0 | iptables -A INPUT -s 10.10.10.3 |
| –destination(-d) | IP адрес назначения пакета. Может быть определен несколькими путями (см. –source). | iptables -A INPUT –destination 192.168.1.0/24 |
| –in-interface (-i) | Определяет интерфейс, на который прибыл пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках INPUT,FORWARD и PREROUTING. Возможно использование знака +, тогда подразумевается использование всех интерфейсов, начинающихся на имя+ (например eth+ - все интерфейсы eth). | iptables -t nat -A PREROUTING –in-interface eth0 |
| –out-interface(-o) | Определяет интерфейс, с которого уйдет пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках OUTPUT, FORWARD и POSTROUTING. Возможно использование знака +. | iptables -t nat -A POSTROUTING –in-interface eth1 |
|  | **Неявные (необщие) параметры** |  |
| -p proto -h | Вывод справки по неявным параметрам протокола proto. | iptables -p icmp -h |
| –source-port(–sport) | Порт источник, возможно только для протоколов –protocol tcp, или –protocol udp | iptables -A INPUT –protocol tcp –source-port 25 |
| –destination-port(–dport) | Порт назначения, возможно только для протоколов –protocol tcp, или –protemocol udp | iptables -A INPUT –protocol udp –destination-port 67 |
|  | **Явные параметры** |  |
| -m state –state (устарел) он же -m conntrack –ctstate | Состояние соединения. Доступные опции: NEW (Все пакеты устанавливающие новое соединение) ESTABLISHED (Все пакеты, принадлежащие установленному соединению) RELATED (Пакеты, не принадлежащие установленному соединению, но связанные с ним. Например - FTP в активном режиме использует разные соединения для передачи данных. Эти соединения связаны.) INVALID (Пакеты, которые не могут быть по тем или иным причинам идентифицированы). | iptables -A INPUT -m state –state NEW, ESTABLISHED iptables -A INPUT -m conntrack –ctstate NEW, ESTABLISHED |
| -m mac –mac-source | Задает MAC адрес сетевого узла, передавшего пакет. MAC адрес должен указываться в форме XX:XX:XX:XX:XX:XX. | -m mac –mac-source 00:00:00:00:00:0 |
|  | **Дополнительные параметры** |  |
|  | DNAT (Destination Network Address Translation) |  |
| –to-destination | Указывает, какой IP адрес должен быть подставлен в качестве адреса места назначения. В примере во всех пакетах протокола tcp, пришедших на адрес 1.2.3.4, данный адрес будет заменен на 4.3.2.1. | iptables -t nat -A PREROUTING -p tcp -d 1.2.3.4 -j DNAT –to-destination 4.3.2.1 |
|  | **LOG** |  |
| –log-level | Используется для задания уровня журналирования (log level). В примере установлен максимальный уровень логирования для всех tcp пакетов в таблице filter цепочки FORWARD. | iptables -A FORWARD -p tcp -j LOG –log-level debug |
| –log-prefix | Задает текст (префикс), которым будут предваряться все сообщения iptables. Префикс может содержать до 29 символов, включая и пробелы. В примере отправляются в syslog все tcp пакеты в таблице filter цепочки INPUT с префиксом INRUT-filter. | iptables -A INPUT -p tcp -j LOG –log-prefix INRUT-filter |
| –log-ip-options | Позволяет заносить в системный журнал различные сведения из заголовка IP пакета. | iptables -A FORWARD -p tcp -j LOG –log-ipoptions |

в скобках – сокращенный вариант записи

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.

**Задания к лабораторной работе**

**Часть 1**

* Установите web-сервер <sudo apt-get install apache2>
* Просмотрите список текущих правил iptables таблицы filter

sudo iptables -L

* Вы увидите, что список содержит три цепочки по умолчанию (INPUT, OUTPUT и FORWARD), в каждой из которых установлена политика по умолчанию (на данный момент это ACCEPT).
* С поможью команды <sudo iptables -S> данный список можно просмотреть в другом формате, который отражает команды, необходимые для активации правил и политик.
* Чтобы сбросить текущие правила (если таковые есть), наберите:

sudo iptables -F

* Цепочка INPUT отвечает за входящий траффик.
* Чтобы внести локальный интерфейс выполните:

sudo iptables -A INPUT -i lo -j ACCEPT

* Чтобы заблокировать весь исходящий трафик, кроме портов для SSH и веб-сервера, нужно сначала разрешить подключения к этим портам. В цепочку ACCEPT добавьте 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

* В данной работе мы не используем SSH. Так что удалим ненужное правило. Для этого:

sudo iptables -D INPUT -p tcp -m tcp –dport 22 -j ACCEPT

* Нужно добавить еще одно правило, которое позволит устанавливать исходящие соединения (т.е. использовать ping или запускать обновления программного обеспечения):

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

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

sudo iptables -P OUTPUT ACCEPT

sudo iptables -P INPUT DROP

* Просмотрите список правил

sudo iptables -L

* 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>.
* Следующее правило отражает атаки syn-flood <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.
* Далее нужно защитить сервер от разведывательных пакетов XMAS <sudo iptables -A INPUT -p tcp –tcp-flags ALL ALL -j DROP>. Теперь сервер защищен от некоторых общих атак, которые ищут его уязвимости.
* Со второй виртуальной машиины, на которую установите nmap, проведите XMAS сканирование <sudo nmap -sX>.
* По умолчанию все не сохраненные правила действуют до следующей перезагрузки сервера; сразу же после перезагрузки не сохраненные правила будут потеряны. Самый простой способ загрузить пакет iptables-persistent <sudo apt-get install iptables-persistent>. 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.

**Часть 2**

* Для начала понадобится LAMP( Apache, MySQL, PHP). В лабораторной работе № 8, уже было показано, как установить его, используя tasksel.
* Установите mod\_security <sudo apt-get install libapache2-mod-security2>
* Выполните команду <sudo apachectl -M | grep –color security2>. Если на экране появился модуль по имени security2\_module (shared), значит, все прошло успешно.
* В каталоге логов Apache можно найти новый лог-файл для mod\_security. /var/log/apache2/modsec\_audit.log
* Установка ModSecurity включает в себя конфигурационный файл, который нужно переименовать: <sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf>.
* Затем перезапустите Apache <sudo service apache2 reload>.
* Стандартный конфигурационный файл настроен на DetectionOnly, то есть, фаервол только отслеживает логи, при этом ничего не блокируя. Чтобы изменить это поведение, отредактируйте файл modsecurity.conf: <sudo nano /etc/modsecurity/modsecurity.conf>
* Найдите в файле строку: «SecRuleEngine DetectionOnly». И измените ее так: «SecRuleEngine On».
* Найдите «SecResponseBodyAccess On» и замените на «SecResponseBodyAccess Off». Эта директива отвечает за буферизацию тела ответа; ее рекомендуется включать, только если требуется обнаружение и предохранение от утечки данных. Включенная директива (SecResponseBodyAccess On) не только будет использовать больше ресурсов сервера, но и увеличит размер лог-файла, следовательно, ее желательно отключить.
* По умолчанию mod\_security поставляется с базовым набором правил CRS (Core Rule Set), которые находятся в /usr/share/modsecurity-crs/
* Чтобы подгрузить эти готовые правила, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого отредактируйте файл mod-security.conf:

nano /etc/apache2/mods-enabled/mod-security.conf

* Между <IfModule security2\_module> </IfModule> внесите следующие параметры:

Include "/usr/share/modsecurity-crs/\*.conf"

Include "/usr/share/modsecurity-crs/activated\_rules/\*.conf"

* Директория activated\_rules аналогична директории Apache mods-enabled. Правила доступны в каталогах: /usr/share/modsecurity-crs/base\_rules ; /usr/share/modsecurity-crs/optional\_rules ; /usr/share/modsecurity-crs/experimental\_rules
* Чтобы активировать правила, нужно создавать символические ссылки в каталоге activated\_rules. <cd /usr/share/modsecurity-crs/activated\_rules/>
* Добавьте несколько правил, например <sudo ln -s /usr/share/modsecurity-crs/base\_rules/modsecurity\_crs\_30\_http\_policy.conf> ; <sudo ln -s /usr/share/modsecurity-crs/base\_rules/modsecurity\_crs\_49\_generic\_attacks.conf>
* Чтобы новые правила вступили в исполнение, нужно перезапустить Apache <sudo service apache2 reload>

**Вопросы к лабораторной работе**

1. Что такое межсетевой экран?
2. Для чего используется межсетвой экран?
3. Принцип работы Netfilter.
4. Таблицы межсетевого экрана Netfilter. Для чего они используются?
5. Что такое правила межсетевого экрана?
6. Как создавать правила для межсетевого экрана утилитой Iptables?
7. Как сохранить правила для последующей автозагрузки?
8. Что такое Web Application Firewall?
9. Как настроить правила в WAF mod\_security?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа NIPS/NIDS: Snort**

**Основные теоретические сведения**

**Цель:** Получить сведения о том, как осуществляется защита с помощью систем обнаружения и предотвращения вторжений. Научиться использовать SNORT.

Система обнаружения вторжений (IDS) — программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет.

Сетевая система обнаружения вторжений (англ. network intrusion detection system, NIDS) — система обнаружения вторжений, которая отслеживает такие виды вредоносной деятельности, как DoS атаки, сканирование портов или даже попытки проникновения в сеть.

В пассивной IDS при обнаружении нарушения безопасности, информация о нарушении записывается в лог приложения, а также сигналы опасности отправляются на консоль и/или администратору системы по определенному каналу связи. В активной системе, также известной как Система Предотвращения Вторжений (IPS — Intrusion Prevention system (англ.)), IDS ведет ответные действия на нарушение, сбрасывая соединение или перенастраивая межсетевой экран для блокирования трафика от злоумышленника. Ответные действия могут проводиться автоматически либо по команде оператора.

Обнаружение проникновения позволяет организациям защищать свои системы от угроз, которые связаны с возрастанием сетевой активности и важностью информационных систем. При понимании уровня и природы современных угроз сетевой безопасности, вопрос не в том, следует ли использовать системы обнаружения проникновений, а в том, какие возможности и особенности систем обнаружения проникновений следует использовать.

Snort — свободная сетевая система предотвращения вторжений (IPS) и обнаружения вторжений (IDS) с открытым исходным кодом, способная выполнять регистрацию пакетов и в реальном времени осуществлять анализ трафика в IP-сетях.

Выполняет протоколирование, анализ, поиск по содержимому, а также широко используется для активного блокирования или пассивного обнаружения целого ряда нападений и зондирований, таких как попытки атак на переполнение буфера, скрытое сканирование портов, атаки на веб-приложения, SMB-зондирование и попытки определения операционной системы. Программное обеспечение в основном используется для предотвращения проникновения, блокирования атак, если они имеют место.

Snort использует правила, написанные простым , но в то же время гибким и достаточно мощным языком. Существует ряд общих принципов написания, запомнить которые достаточно просто.

Большая часть правил Snort умещается в 1 строку. Это следствие того, что до версии 1.8 нельзя было использовать многострочные записи. В более поздних версиях правила можно растягивать на несколько строк, вставляя в конец строки символ “” (без кавычек).

Правила Snort состоят из двух частей: заголовка правила и параметров правила. Заголовок содержит описание действия, протокол передачи данных, IP-адреса, сетевые маски и порты источника и назначения. Параметры правила хранят предупреждающее сообщение, а также информацию о том, какую часть обнаруженного пакета нужно обработать в случае срабатывания правила.

**Задания к лабораторной работе**

* Узнайте свой ip адрес командой ifconfig
* Установите SNORT <sudo apt-get install snort>
* При установке будет нужно указать защищаемую сеть. ВВедите *.*.\*.0/24 (Где *.*.\* - первые три числа вашего ip-адреса, например эот будет 192.168.1.0/24, если вы используете VirtualBox и у вас в настройках сети стоит сетевой мост)
* Запустите SNORT <sudo service snort start>
* Настройка правил
* Перейдите в каталог /etc/snort/rules < cd /etc/snort/rules)
* Создайте файл с правилами <nano test.rules>

alert tcp any any -> any any (content:»<https://www.google.ru/>» ; msg:»Someone open Google website» ; sid: 12312313;)

* Перейдите в каталог /etc/snort <cd /etc/snort)
* Теперь нужно изменить содержимое конфигурационного файла Snort < sudo nano snort.conf>
* Найдите строчки с правилами (они начинаются с include $RULE\_PATH, это в части Step 7) и добавьте файл с нашими правилами

include $RULE\_PATH/test.tules

* В файле snort.conf так же укажите домашнюю сеть. В Step 1 измените строчку «ipvar HOME\_NET any» , на

ipvar HOME\_NET 192.168.1.0/24

* Запустите snort <sudo snort -A console -i eth0 -c snort.conf>
* Зайдите на <https://www.google.ru/> и проверьте в терминале, как работает правило.
* Теперь нам понадобиться еще одна виртуальная машина, на ней должен быть установлен nmap.
* Со второй ВМ используйте ping, посмотрите, как реагирует SNORT
* Используйте различные методы сканирования nmap( используйте -sS, -sT, -sN, -sU, -sX, -sF и посмотрите, как реагирует SNORT;
* В файл test.rules добавьте правило обнаружения сканирования nmap -sN (NULL Scan)

alert tcp any any -> any any (msg:»NULL Scan»; flags: 0; sid:322222;)

* Запустите snort <sudo snort -A console -i eth0 -c snort.conf>
* Со второй виртуальной машины произведите NULL сканирование <sudo nmap -sN>, проверьте, как работает правило.
* Можно загрузить обновленные правила SNORT, для этого:
* Зарегистрируйтесь на сайте <https://www.snort.org/> и скачайте последнюю версию правил
* Разархивируйте скачанный архив и скопируйте каталоги rules, so\_rules и preproc\_rules в /etc/snort :

sudo cp -R ./rules/ /etc/snort/

sudo cp -R ./so\_rules/ /etc/snort/

sudo cp -R ./preproc\_rules/ /etc/snort/

**Вопросы к лабораторной работе**

1. Что такое IDS?
2. Что такое сетевая система обнаружения вторжений?
3. Чем отличаются пассивные и активные IDS?
4. Что такое SNORT?
5. Какие задачи выполняет SNORT?
6. Как работают правила SNORT?
7. Как писать правила для SNORT?
8. Зачем писать собственные правила SNORT?
9. Зачем загружать обновление правил SNORT?
10. Как в SNORT создавать логи?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа SIEM**

**Основные теоретические сведения**

**Цель:** Получение теоритических и практических навыков работы с SIEM

SIEM (Security information and event management) – объединение двух терминов, обозначающих область применения ПО: SIM - Security information management - управление информационной безопасностью и SEM - Security event management - управление событиями безопасности. Технология SIEM обеспечивает анализ в реальном времени событий (тревог) безопасности, исходящих от сетевых устройств и приложений. SIEM представлено приложениями, приборами или услугами, и используется также для журналирования данных и генерации отчетов в целях совместимости (с прочими бизнес-данными).

Перед системой SIEM ставятся следующие задачи.

* Агрегация данных: управление журналами данных; данные собираются из различных источников сетевые устройства и сервисы, датчики систем безопасности, серверы, базы данных, приложения; обеспечивается консолидация данных с целью критических событий.
* Корреляция: поиск общих атрибутов, связывание событий в значимые кластеры. Технология обеспечивает применение различных технических приемов для интеграции данных из различных источников для превращения исходных данных в значащую информацию. Корреляция является типичной функцией подмножества Security Event Management.
* Оповещение: автоматизированный анализ коррелирующих событий и генерация оповещений (тревог) о текущих проблемах. Оповещение может выводиться на «приборную» панель самого приложения, так и быть направлено в прочие сторонние каналы: e-mail, GSM-шлюз итп.
* Средства отображения (информационные панели): отображение диаграмм помогающих идентифицировать паттерны отличные от стандартного поведения.
* Совместимость (трансформируемость): применение приложений для автоматизации сбора данных, формированию отчетности для адаптации агрегируемых данных к существующим процессам управления информационной безопасностью и аудита.
* Хранение данных: применение долговременного хранилища данных в историческом порядке для корреляции данных по времени и для обеспечения трансформируемости. Долговременное хранение данных критично для проведения компьютерно-технических экспертиз, поскольку расследование сетевого инцидента вряд ли будет проводиться в сам момент нарушения.
* Экспертный анализ: возможность поиска по множеству журналов на различных узлах; может выполняться в рамках программно-технической экспертизы.

SIEM способна выявлять:

* сетевые атаки во внутреннем и внешнем периметрах;
* вирусные эпидемии или отдельные вирусные заражения, неудаленные вирусы, бэкдоры и трояны;
* попытки несанкционированного доступа к конфиденциальной информации;
* фрод и мошенничество;
* ошибки и сбои в работе информационных систем;
* уязвимости;
* ошибки конфигураций в средствах защиты и информационных системах.

Splunk Enterprise - платформа для операционной аналитики. Способна осуществлять мониторинг и анализ всех действий, от посещений веб-сайтов и транзакций до сетевых операций и зарегистрированных вызовов.

Splunk – это мощный инструмент операционной аналитики, отслеживающий логи любых систем и собирающий их в единую базу.

Особенности системы:

* Сбор данных из удалённых источников
* Корреляция сложных событий, охватывающих множество разнородных источников данных в среде.
* Масштабирование для сбора и индексации сотен терабайтов данных в день
* Возможность комбинирования данных из традиционных реляционных БД и Hadoop для последующего анализа.
* Ролевая модель доступа к данным.
* Возможность создавать собственные приложения. Можно создавать панели (dashboard’ы), из которых формировать свое собственное Splunk-приложение. У Splunk есть магазин приложений (хотя большинство из них бесплатны), где есть море уже готовых конфигураций для анализа популярных систем, например, UNIX syslog, логи Apache, Microsoft Exchange и т.д.

**Задания к лабораторной работе**

* Загрузите Splunk Enterprise с <http://www.splunk.com/ru_ru/download/splunk-enterprise.html> Выберите вышу систему, после чего нужно будет зарегистрироваться.
* После загрузки дистрибутива, его необходимо установить. Установка deb пакета выполняется командой <dpkg -i splunk\_package\_name.deb>. О других типах установки можно прочитать [здесь](http://docs.splunk.com/Documentation/Splunk/4.3.2/Installation/InstallonLinux)
* Для запуска Splunk выполните </opt/splunk/bin/splunk start>
* Запустите web-интерфейс, при запуске splunk будет указано, как полдключиться к нему (Что-то похожее на [https://sit-VirtualBox:8000](https://sit-virtualbox:8000/) ), чтобы начать использовать систему.
* Учётные данные по умолчанию – admin – changeme. При первом входе Вам будет предложено их изменить.
* В левой части окна будут перечислены приложения, установленные в Splunk и доступные для работы. Приложение это своего рода среда или интерфейс, в котором пользователь работает с событиями, которые собирает Splunk. По умолчанию доступно приложение Search and Reporting. У Splunk есть несколько основных типов расширения функциональности – приложения (Apps) и дополнения (Add-on).
* В центральной части экрана будет пустое окно, на котором предполагается размещение главного дашборда. В правой верхней части расположено меню для управления системой Splunk, в том числе всеми источниками данных.
* Подключим источник событий. Добавим журнал событий Linux, для мониторинга. В правой верхней части экрана выбирайте меню Settings и переходите в Data Inputs
* Перейдите в меню Settings – Data Input - Files & directories. Тип Files & directories позволяет получать события из локальных файлов и директорий.
* Нажмите на кнопку «New», введите путь к файлу auth.log (var/log/.auth.log) и выберите continuously monitor.
* Нажмите «Next». Выберите тип данных (sourcetype – operating system) из списка, а именно «linux\_audit». В открывшемся окне можно ничего не менять. Если всё прошло успешно, то после нажатия на «Start searching» вы увидите перечень событий из журнала аудита.
* Перейдите в меню Settings – Data Input - Files & directories. Добавьте домашнюю директорию, в ней создайте и удалите несколько файлов, Просмотрите журнал событий в Splunk.
* Добавьте еще несколько файлов, директорий и логов, через меню Settings – Data Input - Files & directories.
* Перейдите в приложение «Search and Reporting». Вы попадете на вкладку Search.
* Найдите события, которые относятся к файлу var/log/.auth.log , для этого введите «source=var/log/.auth.log». Здесь так же можно выбрать записи который относятся к Sourcetype (sourcetupe=operating system) – это имя типа данных, куда предполагается относить все данные определённого типа, или Host (host=splunk) – это идентификатор источника, от которого приходят события в какой-либо sourcetype (обычно доменное имя или ip-адрес). Можно фильтровать данные, введя в строку поиска определенные параметры, вы получите записи, только с этими параметрами. Можно делать составные запросы. Один запрос может состоять из множества подзапросов разделенных между собой pipe (|), и справа налево каждый следующий запрос оперирует данными полученными в результате выполнения предыдущего.
* Сбор логов – это далеко не всё, что необходимо для безопасности. Для SIEM нужно, чтобы система не только собирала логи, но и находила события, связанные с нарушениями безопасности. При слежении за логами, можно автоматически обнаруживать любые угрозы безопасности. Splunk можно использовать вместе с IDS.
* В лабораторной работе №14, вы уже познакомились IDS Snort. Так что, установите и настройте Snort, так же как в лабораторной работе №14. Запустите Snort с ведением логов <sudo snort -A console -i eth0 -c snort.conf -l /var/log/snort/>. Произведите различные типы сканирования nmap, и проверку правил Snort. И добавьте логи Snort в Splunk. Вы так же можете загрузить приложение Snort для Splunk <https://splunkbase.splunk.com/app/340/> . Вместо Snort можно так же использовать OSSEC, для OSSEC тоже есть приложение в Splunk.

**Вопросы к лабораторной работе**

1. Что такое SIEM?
2. Для чего используют SIEM?
3. В чем отличие SIEM от IDS?
4. Где используют SIEM?
5. В чем преимущества SIEM?
6. В чем недостатки SIEM?
7. Какие существуют альтернативы использованию SIEM?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа Антиспам (ASSP)**

**Основные теоретические сведения**

**Цель:** Изучить работу почтового сервера, получить практические навыки работы по защите от спама.

Почтовый сервер, сервер электронной почты, мейл-сервер — в системе пересылки электронной почты так обычно называют агент пересылки сообщений (англ. mail transfer agent, MTA). Это компьютерная программа, которая передаёт сообщения от одного компьютера к другому. Обычно почтовый сервер работает «за кулисами», а пользователи имеют дело с другой программой — клиентом электронной почты (англ. mail user agent, MUA).

Спам (англ. spam) — рассылка коммерческой и иной рекламы или подобных коммерческих видов сообщений лицам, не выражавшим желания их получать. Также, название распространяемых материалов. Распространителей спама называют спамерами.

В общепринятом значении термин «спам» в русском языке впервые стал употребляться применительно к рассылке электронных писем. Незапрошенные сообщения в системах мгновенного обмена сообщениями (например, ICQ) носят название SPIM (англ.)русск. (англ. Spam over IM).

Доля спама в мировом почтовом трафике составляет от 60% (2006) до 80% (2011). Самый большой поток спама распространяется через электронную почту. В настоящее время доля вирусов и спама в общем трафике электронной почты составляет по разным оценкам от 70 до 95 процентов

Anti-Spam SMTP Proxy (ASSP) - это ПО с открытым исходным текстом, платформонезависимый SMTP прокси сервер, в котором реализованы белые списки и фильтрация на основании теоремы Байеса.

Антиспамовый сервер собирает письма со спамом и нормальную почту, потом на основании вероятности встречи слов из анализируемого письма в каждой из коллекций (спам или не-спам) сервер делает вывод о том, является ли письмо спамом или нет.

Другие возможности ASSP:

* настройка через веб-интерфейс в браузере,
* автоматическое ведение белого списка,
* наличие необрабатываемых адресов и доменов,
* адреса для сбора спама,
* поддержка дополнительных регулярных выражений для идентификации спама и не-спама,
* обнаружение спама, кодированого MIME,
* автоматическое ведение баз спама и нормальной почты,
* защита от пересылки почты третьми лицами,
* простейший контроль вирусов,
* наличие почтового интерфейса для управления и пополнения коллекций спама и нормальной почты,
* проверка отправителя по RBL и SPF.

**Задания к лабораторной работе**

* На две виртуальные машины установить почтовый сервер Zimbra Collaboration Server. ([На странице загрузки](https://www.zimbra.com/downloads/zimbra-collaboration-open-source), можете посмотреть, какие системы поддерживаются)
* Установите зависимости <sudo apt-get install libgmp10 libperl5.18 unzip pax sysstat sqlite3 dnsmasq wget libaio1>
* Откройте <nano /etc/hostname>. Измените имя хоста на «mail.sit.local»
* Узнайте ip-адрес <ifconfig>.
* Откройте <sudo nano /etc/hosts>. Добавьте строку «192.168.1.113 mail.sit.local mail» , где 192.168.1.113 ваш ip-адрес.
* Откройте <sudo nano /etc/dnsmasq.conf> И добавьте туда:

server=192.168.1.113

domain=sit.local

mx-host=sit.local, mail.sit.local, 5

mx-host=mail.sit.local, mail.sit.local, 5

listen-address=127.0.0.1

* Перезагрузите виртуальную машину <sudo reboot>
* Скачайте Zimbra для своей системы <https://www.zimbra.com/downloads/zimbra-collaboration-open-source>
* Извлеките архив и перейдите в паку, которую извлекли из архива.
* Запустите установку. <sudo ./install.sh>
* Согласитесь с лицензионным соглашением. Y.
* Устанавливайте все пакеты ( Выбираете Y), кроме zimbra-dnscache (Выбирайте N, так как уже используем dnsmask).
* Когда появится меню введите **6** и нажмите Enter.
* Введите **4** и нажмите Enter, введите пароль администратора, минимум 6 символов.
* Введите **r** и нажмите Enter, для возврата в главное меню, затем введите **a** и нажмите Enter, чтобы принять изменения.
* На запросы «Save configuration data to a file» и «The system will be modified - continue?» введите y.
* Ждите пока Zimbra не установится
* Чтобы проверить работу, можете ввести <su - zimbra> <zmcontrol status>.
* Подключитесь к Zimbra в браузере <https://192.168.1.113/> или к странице администратора <https://192.168.1.113:7071/>
* На одну из виртуальных машин, для защиты от спама, установите ASSP (Anti-Spam SMTP Proxy Server). <http://sourceforge.net/projects/assp/> . Скачайте и разархивируйте ASSP.

sudo apt-get install build-essential pmtools libterm-readline-perl-perl libterm-readline-gnu-perl libyaml-perl libtext-glob-perl libnumber-compare-perl libio-compress-perl libemail-mime-perl libemail-send-perl libemail-valid-perl libfile-readbackwards-perl libwww-perl libmime-types-perl libmail-dkim-perl libmail-spf-perl libmail-srs-perl libnet-cidr-lite-perl libnet-dns-perl libnet-ldap-perl libnet-smtp-server-perl libthreads-perl libthread-queue-any-perl libtie-dbi-perl libschedule-cron-perl libio-socket-ssl-perl libdbd-anydata-perl libdbd-csv-perl libdbd-ldap-perl libdbd-mock-perl libdbd-odbc-perl libdbd-mysql-perl libfile-find-rule-perl libfile-slurp-perl libfile-which-perl libfile-chmod-perl liblinux-usermod-perl libcrypt-rc4-perl libtext-pdf-perl libsmart-comments-perl libcam-pdf-perl libpdf-api2-perl imagemagick perlmagick poppler-utils xpdf libauthen-sasl-perl libnet-snmp-perl libsnmp-base libsnmp-dev libsnmp-perl snmp libsnmp-\*-perl libsnmpkit-dev libregexp-optimizer-perl libnet-smtp-tls-perl liblingua-stem-snowball-perl liblingua-identify-perl unzip libberkeleydb-perl

sudo apt-get install tesseract-ocr tesseract-ocr-\*

sudo apt-get install libmodule-signature-perl libtest-pod-perl libtest-pod-coverage-perl libarchive-zip-perl

sudo apt-get install libssl-dev

**sudo cpan**

[…]

Would you like to configure as much as possible automatically? [yes]

[…]

Would you like me to automatically choose some CPAN mirror

sites for you? (This means connecting to the Internet) [yes]

cpan> install Test::Perl::Critic

cpan> install CPAN

cpan> reload cpan

cpan> force install Mail::SPF::Query

cpan> install Net::IP::Match::Regexp Net::SenderBase Net::Syslog Thread::State Sys::MemInfo Crypt::CBC Crypt::OpenSSL::AES DBD::Log DBD::MVS\_FTPSQL DBD::Multiplex DBD::Ovrimos DBD::PgPP DBD::Sprite DBD::Template DBD::mysqlPP DBIx::AnyDBD LEOCHARRE::DEBUG LEOCHARRE::CLI PDF::Burst Image::OCR::Tesseract PDF::GetImages PDF::OCR PDF::OCR2 Mail::DKIM::Verifier Convert::Scalar Unicode::GCString Sys::CpuAffinity

cpan> exit

sudo apt-get install clamav clamav-daemon

sudo freshclam

sudo /etc/init.d/clamav-daemon start

sudo apt-get install libfile-scan-perl

sudo cpan

cpan[1]> test File::Scan::ClamAV

cpan[1]> look File::Scan::ClamAV

/.cpan/build/File-Scan-ClamAV-1.91-Ik8fWD# make install

/.cpan/build/File-Scan-ClamAV-1.91-Ik8fWD# exit

cpan[1]> exit

**Скачайте ASSP и настройте его**:

http://sourceforge.net/projects/assp/

unzip ASSP\_2.3.3\_13137\_install.zip

sudo mkdir -p /usr/share/assp

sudo mv -f assp/\* /usr/share/assp

rm -rf assp ASSP\_2.3.3\_13137\_install.zip Install.txt MacOSX-launchd.txt quickstart.txt Win32-quickstart-guide.txt

sudo chown -R nobody:nogroup /usr/share/assp

sudo chmod 755 /usr/share/assp/assp.pl

sudo nano /etc/init.d/assp

Contents:

===

#!/bin/sh -e

# Start or stop ASSP

#

# original version by Ivo Schaap <ivo@lineau.nl> had issues on Debian4. Modified by atramos.

#

### BEGIN INIT INFO

# Provides: ASSP (Anti-Spam SMTP Proxy)

# Required-Start: $syslog, $local\_fs

# Required-Stop: $syslog, $local\_fs

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Start ASSP

# Description: Enable service provided by daemon.

### END INIT INFO

PATH=/bin:/usr/bin:/sbin:/usr/sbin

case “$1” in

start)

echo -n “Starting the Anti-Spam SMTP Proxy”

cd /usr/share/assp

perl assp.pl 2>&1 > /dev/null &

;;

stop)

echo -n “Stopping the Anti-Spam SMTP Proxy”

kill -9 ps ax | grep "perl assp.pl" | grep -v grep | awk '{ print $1 }'

;;

restart)

$0 stop || true

$0 start

;;

``\*)``

echo “Usage: /etc/init.d/assp {start|stop|restart}”

exit 1

;;

esac

exit 0

===

* sudo chmod 755 /etc/init.d/assp
* sudo /usr/share/assp/assp.pl
* Нажмите Ctrl+C
* sudo update-rc.d assp defaults
* sudo /etc/init.d/assp start
* Переходим на [http://antispam\_host:55555](http://antispam_host:55555/)
* login:root
* pass:nospam4me
* Через машину без ASSP, отправляйте спам на машину с ASSP. Скриптов для генерации спама в интернете полно, но нужен такой, где можно использовать свой почтовый сервер (Например, можете использовать Social Engineer Toolkit). Посмотрите, как блокируется спам и как система обучается.

**Вопросы к лабораторной работе**

1. Что такое почтовый сервер?
2. Принцип работы почтового сервера.
3. Что такое спам?
4. Что такое ASSP?
5. Как работает ASSP?
6. Можно ли полностью защитить почтовый сервер от спама?
7. Какие еще угрозы почтовому серверу существуют?
8. Какие методы защиты почтового сервера существуют?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа Sandbox**

**Основные теоретические сведения**

**Цель:** Получение теоритических и практических навыков работы с песочницами и файловыми антивирусами.

Вредоносная программа (англ. malware, malicious software — «злонамеренное программное обеспечение») — любое программное обеспечение, предназначенное для получения несанкционированного доступа к вычислительным ресурсам самой ЭВМ или к информации, хранимой на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ или причинения вреда (нанесения ущерба) владельцу информации, и/или владельцу ЭВМ, и/или владельцу сети ЭВМ, путем копирования, искажения, удаления или подмены информации. Многие антивирусы считают крэки, кейгены и прочие программы для взлома вредоносными программами, или потенциально опасными.

Песочница (англ. Sandbox) - это ограниченная среда в вашей системе для исполнения гостевых программ без доступа к главной операционной системе. Это закрытое для доступа извне виртуальное пространство, в котором можно работать с программным обеспечением без изменения системных файлов.

Песочница (от англ. Sandbox, схожие понятия — англ. honeypot, англ. fishbowl) - в компьютерной безопасности, механизм для безопасного исполнения программ. Песочницы часто используют для запуска не протестированного кода, непроверенного кода из неизвестных источников, а также для запуска и обнаружения вирусов.

Песочница — в компьютерной безопасности специально выделенная среда для безопасного исполнения компьютерных программ.

Песочница обычно представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность сообщаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. Песочницы представляют собой пример виртуализации. Повышенная безопасность исполнения кода в песочнице зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды песочниц используют только для не отлаженного или подозрительного кода.

Cuckoo Sandbox — система для автоматического исследования вредоносного ПО, эксплоитов, вредоносных скриптов, документов, архивов и ссылок. Система способна проверять документы pdf, doc, xls, rtf, скрипты Python, JS, DLL библиотеки, бинарники, jar и многое другое.

Файловый Антивирус — компонент модуля Защита компьютера, контролирующий файловую систему компьютера. Он запускается при старте операционной системы, постоянно находится в оперативной памяти компьютера и проверяет все открываемые, сохраняемые и запускаемые файлы. Каждый файл, к которому вы обратитесь, будет перехвачен Файловым Антивирусом и проверен на присутствие известных вирусов.

Clam AntiVirus — пакет антивирусного ПО, работающий во многих операционных системах, включая Unix-подобные ОС, OpenVMS, Microsoft Windows и Apple Mac OS X.

Главная цель Clam AntiVirus — интеграция с серверами электронной почты для проверки файлов, прикреплённых к сообщениям. В пакет входит масштабируемый многопоточный демон clamd, управляемый из командной строки сканер clamscan, а также модуль обновления сигнатур по Интернету freshclam.

**Задания к лабораторной работе**

**Установите Cukoo Sandbox.**

**Примечание**

Если будут проблемы с установкой, воспользуйтесь документацией Cukoo sandbox <http://docs.cuckoosandbox.org/en/latest/installation/>

\* Установка зависимостей

cd /tmp

apt-get update

apt-get install git automake mongodb mingw32 dkms unzip wget python python-sqlalchemy python-bson python-pip python-dpkt python-jinja2 python-magic python-mysqldb python-gridfs python-libvirt python-bottle python-pefile python-chardet -y

apt-get install python-dev libxml2-dev libxslt1-dev libevent-dev libpcre3 libpcre3-dev zlib1g-dev libtool libpcre++-dev –y

apt-get install mariadb-server -y

pip install lxml

pip install cybox==2.0.1.4

pip install maec==4.0.1.0

pip install django

pip install py3compat

pip install pymongo

apt-get install ssdeep python-pyrex subversion libfuzzy-dev -y

svn checkout http://pyssdeep.googlecode.com/svn/trunk/ pyssdeep

cd pyssdeep

python setup.py build

python setup.py install

pip install pydeep

cd /tmp

wget https://github.com/plusvic/yara/archive/v2.1.0.tar.gz

tar xzf v2.1.0.tar.gz

cd yara-2.1.0

chmod +x build.sh

./build.sh

make install

cd yara-python

python setup.py build

python setup.py install

cd /tmp

wget http://distorm.googlecode.com/files/distorm3.zip

unzip distorm3.zip

cd distorm3

python setup.py build

python setup.py install

add-apt-repository ppa:pi-rho/security

apt-get update

apt-get install volatility

**Установка и настройка Virtualbox**:

wget -q http://download.virtualbox.org/virtualbox/debian/oracle\_vbox.asc -O- | sudo apt-key add –

sh -c 'echo "deb http://download.virtualbox.org/virtualbox/debian trusty contrib" >> /etc/apt/sources.list.d/virtualbox.list'

apt-get update

apt-get install virtualbox-4.3

cd /tmp

VBOX\_LATEST\_VERSION=$(curl http://download.virtualbox.org/virtualbox/LATEST.TXT)

wget http://download.virtualbox.org/virtualbox/${VBOX\_LATEST\_VERSION}/Oracle\_VM\_VirtualBox\_Extension\_Pack-{VBOX\_LATEST\_VERSION}.vbox-extpack

vboxmanage extpack install /tmp/Oracle\_VM\_VirtualBox\_Extension\_Pack-${VBOX\_LATEST\_VERSION}.vbox-extpack

cd /opt

wget http://dlc.sun.com.edgesuite.net/virtualbox/${VBOX\_LATEST\_VERSION}/VBoxGuestAdditions\_${VBOX\_LATEST\_VERSION}.iso

**Установка Cuckoo Sandbox**:

useradd cuckoo

usermod -a -G vboxusers cuckoo

id cuckoo

cd /opt

wget http://downloads.cuckoosandbox.org/1.1/cuckoo\_1.1.tar.gz

tar xzf cuckoo\_1.1.tar.gz

**Настройка Cuckoo Sandbox**:

cd /opt/cuckoo

./utils/community.py --signatures --force

**Настройка БД**:

mysql -u root -p

> create database cuckoo;

> grant all privileges on cuckoo.\* to cuckoo**@localhost** identified by 'cuck00pass' ;

> flush privileges;

> quit;

**Настраиваем cuckoo:**:

\* Файл /opt/cuckoo/conf/cuckoo.conf

Включаем запись дампа памяти:

memory\_dump = on

Настраиваем подключение к бд:

connection = mysql://cuckoo:cuck00pass\**@localhost**/cuckoo

Увеличиваем временные лимиты:

default = 240

critical = 1200

vm\_state = 600

\* Файл /opt/cuckoo/conf/memory.conf

Отключаем сохранение дампов памяти:

delete\_memdump = yes

\* Файл /opt/cuckoo/conf/processing.conf

Включаем анализ оперативной памяти:

memory = yes

\* nano /opt/cuckoo/conf/virtualbox.conf

Меняем режим работы Virtualbox:

mode = headless

Меняем названия виртуальной машины с cuckoo1 на WindowsXP:

machines = WindowsXP

[WindowsXP]

label = WindowsXP

\* Файл /opt/cuckoo/conf/reporting.conf

Включим импорт отчётов в MongoDB для работы веб интерфейса

[mongodb]

enabled = yes

**Примечание**

* На этом настройка Cuckoo закончена, теперь приступим к Virtualbox и гостевой ОС.

**Загрузка виртуальной ОС с сайта:**

wget https://az412801.vo.msecnd.net/vhd/VMBuild\_20131127/VirtualBox/IE6\_WinXP/Linux/IE6.WinXP.For.LinuxVirtualBox.sfx

chmod +x IE6.WinXP.For.LinuxVirtualBox.sfx

./IE6.WinXP.For.LinuxVirtualBox.sfx

vboxmanage **import** **IE6**\ -\ WinXP.ova --vsys 0 --unit 10 --disk=/root/VirtualBox\ VMs/WindowsXP/WindowsXP.vmdk --memory 1024 --vmname WindowsXP

**Настраиваем сеть**:

vboxmanage hostonlyif create

vboxmanage modifyvm "WindowsXP" --nic1 hostonly --hostonlyadapter1 vboxnet0 --nicpromisc1 allow-all --hwvirtex off --vtxvpid off

**Настраиваем общие папки**:

mkdir -p /opt/cuckoo/shares/setup

mkdir -p /opt/cuckoo/shares/WindowsXP

vboxmanage sharedfolder add "WindowsXP" --name "WindowsXP" --hostpath /opt/cuckoo/shares/WindowsXP --automount

vboxmanage sharedfolder add "WindowsXP" --name setup --hostpath /opt/cuckoo/shares/setup --automount --readonly

vboxmanage modifyvm "WindowsXP" --nictrace1 on --nictracefile1 /opt/cuckoo/shares/WindowsXP/dump.pcap

**Включаем доступ по RDP, порт можете указать любой**:

vboxmanage modifyvm "WindowsXP" --vrdeport 5000 --vrde on

**Примечание**

**На этом конфигурация виртуальных контейнеров полностью закончена, осталось настроить iptables, tcpdump.**:

iptables -A FORWARD -o eth0 -i vboxnet0 -s 192.168.56.0/24 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -A POSTROUTING -t nat -j MASQUERADE

sysctl -w net.ipv4.ip\_forward=1

setcap cap\_net\_raw,cap\_net\_admin=eip /usr/sbin/tcpdump

getcap /usr/sbin/tcpdump

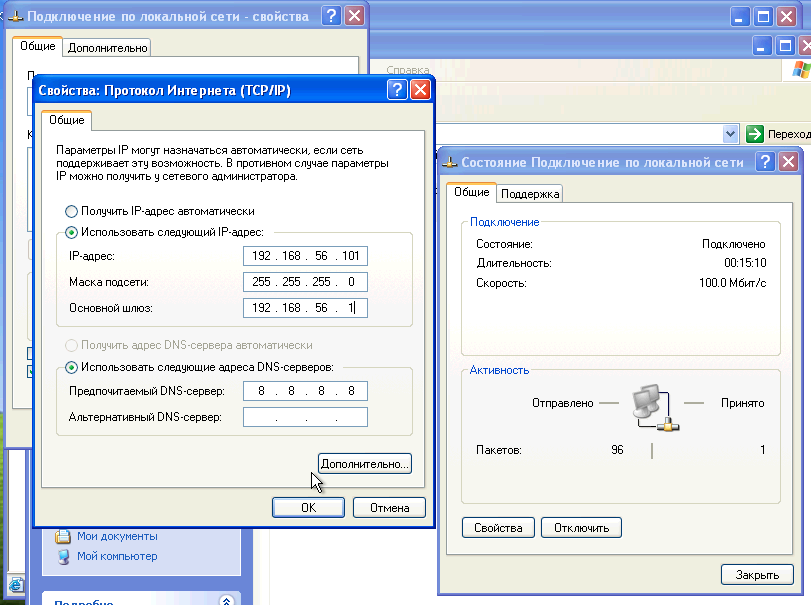
ifconfig vboxnet0 192.168.56.1

* После установки и настройки Windows переходим непосредственно на саму гостевую ОС.

**Предупреждение**

(Если возникли проблемы с установкой виртуальной машины, можете посмотреть, как это сделано в этой статье <http://gwallgofi.com/cuckoo-sandbox-part-2-installing/> )

* Следующим образом настроим подключение к сети (dns можете указать любой):

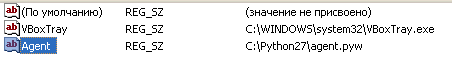


* Установим VboxTools с диска, который подключен к системе.
* Устанавливаем Pyton 2.7: <http://python.org/download/>
* Устанавливаем <http://www.activestate.com/activepython>
* Устанавливаем PIL Python модуль, для создания скриншотов: <http://www.pythonware.com/products/pil/>
* Отключаем автоматическое обновление Windows.
* Отключаем брандмауэер.
* Копируем агент из сетевой папки setup в папку C:Python27, Ставим агент на автозагрузку, для этого добавляем в ветку реестра(пуск->выполнить->regedit) HKLMSOFTWAREMicrosoftWindowsCurrentVersionRun строковый параметр

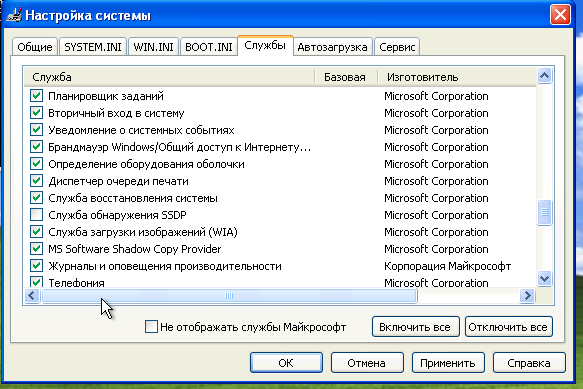
Имя:“Agent“

Тип:“REG\_SZ“

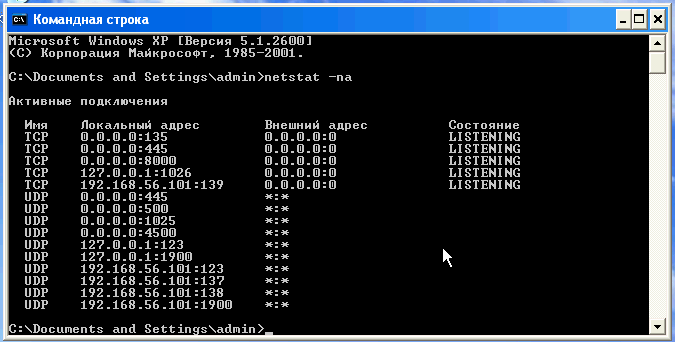
Содержание:«C:Python27agent.pyw»



* Включаем IE, в настройках ставим домашней страницей пустую вкладку, по желанию в свойствах обозревателя выключаем все защитные механизмы.
* Отключаем SSDP: пуск->выполнить->msconfig и в разделе службы отключаем «Служба обнаружения SSDP», чтобы в отчётах не фигурировали сетевые запросы этой службы.



* Перезагружаемся и в появившемся при загрузке окне выбираем «При перезагрузке не выводить это сообщение» и ОК.
* После перезагрузки гостевой ОС, пуск->выполнить->cmd и в консоли набираем netstat –na и смотрим есть ли агент на 8000-ом порту



* По желанию устанавливаем различное уязвимое ПО старых версий (браузеры, Flash player, Java, Acrobat Reader…)
* На этом установка гостевой ОС закончена.Делаем снапшот (не выключая гостевую ОС) <vboxmanage snapshot «WindowsXP» take «WindowsXPSnap01» –pause>
* И выключаем: <vboxmanage controlvm «WindowsXP» poweroff>
* Запуск Cukoo Sandbox <python cuckoo.py>
* Загрузите вредоносное ПО, для проверки работы Cukoo Sandbox. Здесь есть небольшой список ресурсов с образцами вредоносного ПО <https://zeltser.com/malware-sample-sources/> . Например, Можете загрузить с <http://malshare.com/> или с <https://malwr.com/> .
* Чтобы отправить файл в Cukoo на анализ используйте команду submit: <python submit.py /path/to/binary>
* Запустите web-интерфейс cukoo и просмотрите результаты <python web.py>

**Вопросы к лабораторной работе**

1. Что такое песочница?
2. Принцип работы песочниц.
3. Где используют песочницы?
4. Преимущества и недостатки песочниц.
5. Альтернативы использованию песочниц.
6. Что такое эмуляция?
7. Что такое эвристический анализ? В чем отличия от сигнатурного анализа?
8. Для чего нужен файловый антивирус?
9. Что такое вредоносное ПО?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа Iptables, WEB APLICATION FIREWALL**

**Основные теоретические сведения**

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

**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.

| *Основные команды Iptables* | | |
| --- | --- | --- |
| **Параметр** | **Описание** | **Пример** |
| –append (-A) | Позволяет добавить в указанную цепочку и таблицу заданное правило, помещаемое в КОНЕЦ списка | iptables -A FORWARD критерии -j действие |
| –delete (-D) | Позволяет удалить заданное номером или каким-либо правилом правило. В первом примере удаляются все правила с номерами 10,12 во всех цепочках, в таблицах filter. | iptables -D 10,12 iptables -t mangle -D PREROUTING критерии -j действие |
| –rename-chain (-E) | Изменить имя цепочки. | iptables -E OLD\_CHAIN NEW\_CHAIN |
| –flush (-F) | Очищает все правила текущей таблицы. Ко всем пакетам, относящимся к уже установленным соединениям, применяется терминальное действие ACCEPT — пропустить | iptables -F |
| –insert (-I) | Добавляет заданное правило в соответствии с номером. | iptables -I FORWARD 5 критерии -j действие |
| –list (-L) | Позволяет просматривать существующие правила (без явного указания таблицы - отображается таблица filter всех цепочек). | iptables -L |
| –policy (-P) | Позволяет устанавливать стандартную политику для заданной цепочки. | iptables -t mangle -P PREROUTING DROP |
| –replace (-R) | Заменяет заданное номером правило на заданное в критериях. | iptables -R POSROUTING 7 | критерии -j действие |
| –delete-chain (-X) | Удалить ВСЕ созданные вручную цепочки (оставить только стандартные INPUT, OUTPUT…) | iptables -X |
| –zero (-Z) | Обнуляет счетчики переданных данных в цепочке. | iptables -Z INPUT |
| –line-numbers | Указывать номера правил при выводе (может использоваться совместно с -L). | iptables -L –line-numbers |
| –help (-h) | Помощь | Iptables –help |
| -t таблица | Задает название таблицы, над которой необходимо совершить действие. В примере сбрасывается таблица nat во всех цепочках. | iptables -t nat -F |
| –verbose (-v) | Детальный вывод. | iptables -L -v |
|  | **Основные правила отбора пакетов** |  |
| –protocol(сокр. -p) | Определяет протокол транспортного уровня. Опции tcp, udp, icmp, all или любой другой протокол определенный в /etc/protocols | iptables -A INPUT -p tcp |
| –source(-s, –src) | IP адрес источника пакета. Может быть определен несколькими путями:Одиночный хост: host.domain.tld, или IP адрес: 10.10.10.3 Пул-адресов (подсеть): 10.10.10.3/24 или 10.10.10.3/255.255.255.0 | iptables -A INPUT -s 10.10.10.3 |
| –destination(-d) | IP адрес назначения пакета. Может быть определен несколькими путями (см. –source). | iptables -A INPUT –destination 192.168.1.0/24 |
| –in-interface (-i) | Определяет интерфейс, на который прибыл пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках INPUT,FORWARD и PREROUTING. Возможно использование знака +, тогда подразумевается использование всех интерфейсов, начинающихся на имя+ (например eth+ - все интерфейсы eth). | iptables -t nat -A PREROUTING –in-interface eth0 |
| –out-interface(-o) | Определяет интерфейс, с которого уйдет пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках OUTPUT, FORWARD и POSTROUTING. Возможно использование знака +. | iptables -t nat -A POSTROUTING –in-interface eth1 |
|  | **Неявные (необщие) параметры** |  |
| -p proto -h | Вывод справки по неявным параметрам протокола proto. | iptables -p icmp -h |
| –source-port(–sport) | Порт источник, возможно только для протоколов –protocol tcp, или –protocol udp | iptables -A INPUT –protocol tcp –source-port 25 |
| –destination-port(–dport) | Порт назначения, возможно только для протоколов –protocol tcp, или –protemocol udp | iptables -A INPUT –protocol udp –destination-port 67 |
|  | **Явные параметры** |  |
| -m state –state (устарел) он же -m conntrack –ctstate | Состояние соединения. Доступные опции: NEW (Все пакеты устанавливающие новое соединение) ESTABLISHED (Все пакеты, принадлежащие установленному соединению) RELATED (Пакеты, не принадлежащие установленному соединению, но связанные с ним. Например - FTP в активном режиме использует разные соединения для передачи данных. Эти соединения связаны.) INVALID (Пакеты, которые не могут быть по тем или иным причинам идентифицированы). | iptables -A INPUT -m state –state NEW, ESTABLISHED iptables -A INPUT -m conntrack –ctstate NEW, ESTABLISHED |
| -m mac –mac-source | Задает MAC адрес сетевого узла, передавшего пакет. MAC адрес должен указываться в форме XX:XX:XX:XX:XX:XX. | -m mac –mac-source 00:00:00:00:00:0 |
|  | **Дополнительные параметры** |  |
|  | DNAT (Destination Network Address Translation) |  |
| –to-destination | Указывает, какой IP адрес должен быть подставлен в качестве адреса места назначения. В примере во всех пакетах протокола tcp, пришедших на адрес 1.2.3.4, данный адрес будет заменен на 4.3.2.1. | iptables -t nat -A PREROUTING -p tcp -d 1.2.3.4 -j DNAT –to-destination 4.3.2.1 |
|  | **LOG** |  |
| –log-level | Используется для задания уровня журналирования (log level). В примере установлен максимальный уровень логирования для всех tcp пакетов в таблице filter цепочки FORWARD. | iptables -A FORWARD -p tcp -j LOG –log-level debug |
| –log-prefix | Задает текст (префикс), которым будут предваряться все сообщения iptables. Префикс может содержать до 29 символов, включая и пробелы. В примере отправляются в syslog все tcp пакеты в таблице filter цепочки INPUT с префиксом INRUT-filter. | iptables -A INPUT -p tcp -j LOG –log-prefix INRUT-filter |
| –log-ip-options | Позволяет заносить в системный журнал различные сведения из заголовка IP пакета. | iptables -A FORWARD -p tcp -j LOG –log-ipoptions |

в скобках – сокращенный вариант записи

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.

**Задания к лабораторной работе**

**Часть 1**

* Установите web-сервер <sudo apt-get install apache2>
* Просмотрите список текущих правил iptables таблицы filter

sudo iptables -L

* Вы увидите, что список содержит три цепочки по умолчанию (INPUT, OUTPUT и FORWARD), в каждой из которых установлена политика по умолчанию (на данный момент это ACCEPT).
* С поможью команды <sudo iptables -S> данный список можно просмотреть в другом формате, который отражает команды, необходимые для активации правил и политик.
* Чтобы сбросить текущие правила (если таковые есть), наберите:

sudo iptables -F

* Цепочка INPUT отвечает за входящий траффик.
* Чтобы внести локальный интерфейс выполните:

sudo iptables -A INPUT -i lo -j ACCEPT

* Чтобы заблокировать весь исходящий трафик, кроме портов для SSH и веб-сервера, нужно сначала разрешить подключения к этим портам. В цепочку ACCEPT добавьте 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

* В данной работе мы не используем SSH. Так что удалим ненужное правило. Для этого:

sudo iptables -D INPUT -p tcp -m tcp –dport 22 -j ACCEPT

* Нужно добавить еще одно правило, которое позволит устанавливать исходящие соединения (т.е. использовать ping или запускать обновления программного обеспечения):

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

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

sudo iptables -P OUTPUT ACCEPT

sudo iptables -P INPUT DROP

* Просмотрите список правил

sudo iptables -L

* 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>.
* Следующее правило отражает атаки syn-flood <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.
* Далее нужно защитить сервер от разведывательных пакетов XMAS <sudo iptables -A INPUT -p tcp –tcp-flags ALL ALL -j DROP>. Теперь сервер защищен от некоторых общих атак, которые ищут его уязвимости.
* Со второй виртуальной машиины, на которую установите nmap, проведите XMAS сканирование <sudo nmap -sX>.
* По умолчанию все не сохраненные правила действуют до следующей перезагрузки сервера; сразу же после перезагрузки не сохраненные правила будут потеряны. Самый простой способ загрузить пакет iptables-persistent <sudo apt-get install iptables-persistent>. 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.

**Часть 2**

* Для начала понадобится LAMP( Apache, MySQL, PHP). В лабораторной работе № 8, уже было показано, как установить его, используя tasksel.
* Установите mod\_security <sudo apt-get install libapache2-mod-security2>
* Выполните команду <sudo apachectl -M | grep –color security2>. Если на экране появился модуль по имени security2\_module (shared), значит, все прошло успешно.
* В каталоге логов Apache можно найти новый лог-файл для mod\_security. /var/log/apache2/modsec\_audit.log
* Установка ModSecurity включает в себя конфигурационный файл, который нужно переименовать: <sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf>.
* Затем перезапустите Apache <sudo service apache2 reload>.
* Стандартный конфигурационный файл настроен на DetectionOnly, то есть, фаервол только отслеживает логи, при этом ничего не блокируя. Чтобы изменить это поведение, отредактируйте файл modsecurity.conf: <sudo nano /etc/modsecurity/modsecurity.conf>
* Найдите в файле строку: «SecRuleEngine DetectionOnly». И измените ее так: «SecRuleEngine On».
* Найдите «SecResponseBodyAccess On» и замените на «SecResponseBodyAccess Off». Эта директива отвечает за буферизацию тела ответа; ее рекомендуется включать, только если требуется обнаружение и предохранение от утечки данных. Включенная директива (SecResponseBodyAccess On) не только будет использовать больше ресурсов сервера, но и увеличит размер лог-файла, следовательно, ее желательно отключить.
* По умолчанию mod\_security поставляется с базовым набором правил CRS (Core Rule Set), которые находятся в /usr/share/modsecurity-crs/
* Чтобы подгрузить эти готовые правила, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого отредактируйте файл mod-security.conf:

nano /etc/apache2/mods-enabled/mod-security.conf

* Между <IfModule security2\_module> </IfModule> внесите следующие параметры:

Include "/usr/share/modsecurity-crs/\*.conf"

Include "/usr/share/modsecurity-crs/activated\_rules/\*.conf"

* Директория activated\_rules аналогична директории Apache mods-enabled. Правила доступны в каталогах: /usr/share/modsecurity-crs/base\_rules ; /usr/share/modsecurity-crs/optional\_rules ; /usr/share/modsecurity-crs/experimental\_rules
* Чтобы активировать правила, нужно создавать символические ссылки в каталоге activated\_rules. <cd /usr/share/modsecurity-crs/activated\_rules/>
* Добавьте несколько правил, например <sudo ln -s /usr/share/modsecurity-crs/base\_rules/modsecurity\_crs\_30\_http\_policy.conf> ; <sudo ln -s /usr/share/modsecurity-crs/base\_rules/modsecurity\_crs\_49\_generic\_attacks.conf>
* Чтобы новые правила вступили в исполнение, нужно перезапустить Apache <sudo service apache2 reload>

**Вопросы к лабораторной работе**

1. Что такое межсетевой экран?
2. Для чего используется межсетвой экран?
3. Принцип работы Netfilter.
4. Таблицы межсетевого экрана Netfilter. Для чего они используются?
5. Что такое правила межсетевого экрана?
6. Как создавать правила для межсетевого экрана утилитой Iptables?
7. Как сохранить правила для последующей автозагрузки?
8. Что такое Web Application Firewall?
9. Как настроить правила в WAF mod\_security?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы**