КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Сколько существует уровней эталонной модели взаимосвязи открытых систем?
2. Перечислите уровни эталонной модели BOC, кратко охарактеризуйте их?
3. Объясните принципы функционирования протокола TCP/IP.
4. Объясните принципы функционирования межсетевых экранов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дэвид В. Чепмен, мл., Энди Фокс. Брандмауэры Cisco Secure PIX = Cisco® Secure PIX® Firewalls. — М.: «Вильямс», 2003. — С. 384. — ISBN 1-58705-035-8.
2. Оглтри Т. "Firewalls. Практическое применение межсетевых экранов", Пресс, 2001 год.
3. Мэйволд, Э. Безопасность сетей. - М: Эком, 2005. – 528 с.
4. Максимов, В. Межсетевые экраны. Способы организации защиты// КомпьютерПресс. – 2003. - № 3

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
_____О.Г. Локтионова
«___» _____2017 г.

Настройка межсетевого экрана в Linux

Методические указания по выполнению лабораторной работы по дисциплине «Безопасность операционных систем» для студентов укрупненной группы специальностей 10.00.00

Курск 2017

	на наш хост к портам 3312,2234	334:1658	
По умолчанию	запретить	запретить	разрешить
Вариант 5	Входящие	Исходящие	Транзитные
Разрешить	• telnet • domain • ntp • smtp • icmp • ftp,ftp-data • 1024:65535 •	• telnet • domain • ntp • smtp • icmp • 1024:65535	
Запретить	• icmp с 192.168.1.15/30 • udp на наш хост к портам 1456,3333	• tcp 368:774 • udp 334:1658	
По умолчани	запретить	запретить	запре

Оглавление

ВВЕДЕНИЕ 4

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5

1.1 Сетевые пакеты 5

1.2 Классификация сетевых угроз 7

1.3 Защита от сетевых угроз 11

2. ПРАКТИЧЕСКАЯ ЧАСТЬ 15

2.1 LINUX IpTables 15

2.2 Методические указания 20

ЗАКЛЮЧЕНИЕ 22

СПИСОК ЛИТЕРАТУРЫ 23

	портов 20,21 192.168.1.2	334	
По умолчанию	запретить	разрешить	запретить
Вариант 2	Входящие	Исходящие	Транзитные
Разрешить	• http(80) • domain • smpt • ftp,ftp-data • icmp	• http(80) • domain • smpt • icmp	
Запретить	• icmp с 192.168.1.7 • tcp с портов 17:30 192.168.1.2	• tcp 3682 • udp 16825	
По умолчанию	запретить	разрешить	запретить
Вариант 3	Входящие	Исходящие	Транзитные
Разрешить	• http(80) • domain	• http(80) • domain	

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Сетевые пакеты

Сетевые пакеты - структурно отделенные друг от друга порции данных, которые используются для представления информации, передаваемой по сети с коммутацией пакетов. Наряду с термином "пакет" используются также термины "кадр", "фрейм" и "ячейка".

История компьютерных сетей берет свое начало в 1960-х годах, когда телефонные сети были основным средством связи в мире. В телефонных сетях использовался принцип коммутации каналов, при этом передача осуществляется с постоянной частотой. Стремительный рост потребности в вычислительных ресурсах, сочетающийся с высокой стоимостью ЭВМ, стал причиной объединения компьютеров в сети для обеспечения к ним удаленного совместного доступа пользователей. Сетевой трафик был неравномерным и характеризовался наличием периодов активности (например, когда один пользователь посылал команду удаленному компьютеру) и пассивности (ожидание результатов).

Три группы инженеров, находившиеся в разных частях света, независимо друг от друга начали разработку технологии коммутации пакетов, рассматривая ее как мощную и эффективную альтернативу технологии коммутации каналов. Первая научная работа на эту тему была опубликована ученым Леонардом Клейнроком, в то время еще студентом-старшекурсником. С помощью теории очередей работа Клейнрока наглядно продемонстрировала эффективность принципа коммутации пакетов в условиях неравномерной нагрузки.

Эти разработки заложили основу современного Интернета. Однако было бы неверно сводить зарождение Интернета только к разработке технологии коммутации пакетов. В начале 1960-х годов коллеги Клейнрока, ученые Ликлайдер и Роберте, стали участниками программы развития компьютерных технологий в агентстве DARPA (Defense Advanced Research Projects Agency - агентство по защите прогрессивных исследовательских проектов). Роберте разработал схему компьютерной сети ARPAnet, основанной на коммутации пакетов и являющейся прямым предком

8.Можно разрешить пользователям инициировать исходящие соединения и передавать по ним данные, но только для определенных протоколов. Именно здесь вы можете запретить применение FTP и других необязательных программ.

iptables -A FORWARD -m multiport -p tcp -o eth0 -d 0.0.0.0 --dport www --syn -j ACCEPT
iptables -A FORWARD -m multiport -p tcp -o eth0 -d 0.0.0.0 --dport smtp --syn -j ACCEPT

```
root@dhcppc4:~# iptables -A FORWARD -m multiport -p tcp -o eth0 -d 0.0.0.0 --dpo
rt smtp --syn -j ACCEPT
root@dhcppc4:~# iptables -A FORWARD -m multiport -p tcp -o eth0 -d 0.0.0.0 --dpo
rt www --syn -j ACCEPT
```

9.Необходимо пропускать некоторые входящие и исходящие пакеты UDP. UDP применяются для DNS, и, если эти пакеты заблокировать, то пользователи не смогут выполнять разрешение адресов. Так как, в отличие от TCP, UDP-пакеты не имеют состояния, нельзя полагаться на проверки флагов SYN или ACK. Вы хотите разрешить UDP только на порт 53, поэтому вы задаете domain (встроенную переменную для порта 53) как единственно допустимый порт. Это делается с помощью следующих инструкций:

iptables -A FORWARD -m multiport -p udp -i eth0 -d 192.168.0.0/24 --dports domain -j ACCEPT
iptables -A FORWARD -m multiport -p udp -i eth0 -s 192.168.0.0/24 --sports domain -j ACCEPT
iptables -A FORWARD -m multiport -p udp -i eth1 -d 0.0.0.0--dports domain -j ACCEPT
iptables -A FORWARD -m multiport -p udp -i eth0 -s 0.0.0.0 --sports domain -j ACCEPT

```
root@dhcppc4:~# iptables -A FORWARD -m multiport -p udp -i eth0 -d 192.168.0.0/24 --dport domain -j ACCEPT
root@dhcppc4:~# iptables -A FORWARD -m multiport -p udp -i eth0 -s 192.168.0.0/24 --dport domain -j ACCEPT
root@dhcppc4:~# iptables -A FORWARD -m multiport -p udp -i eth1 -d 0.0.0.0 --dport domain -j ACCEPT
root@dhcppc4:~# iptables -A FORWARD -m multiport -p udp -i eth0 -s 0.0.0.0 --dport domain -j ACCEPT
root@dhcppc4:~#
```

Первая из двух приведенных выше инструкций разрешает входящие дейтаграммы UDP, а вторая - исходящие.

10.Наконец, вы хотите установить протоколирование, чтобы, просматривая журнал, можно было увидеть, какие пакеты были отброшены. Журнал желательно периодически просматривать,

избыточного кода над содержимым пакета, чем проверка каждого символа с помощью бита чётности. Хвостовая часть пакета часто содержит данные проверки ошибок, возникших во время передачи пакета по сети.

Современные сети обычно соединяют между собой три или более хоста. В таких случаях заголовок пакета обычно содержит информацию, в которой записан фактический адрес хоста. В сложных сетях, построенных из нескольких узлов коммутации и маршрутизации, такие как ARPANET или современный интернет, ряд пакетов, отправленных с одного компьютера на другой, может следовать разными маршрутами. Эта технология называется пакетной коммутацией.

1.2 Классификация сетевых угроз

Сетевые угрозы столь же разнообразны, как и системы, против которых они направлены. Некоторые атаки отличаются большой сложностью. Другие может осуществить обычный оператор, даже не предполагающий, какие последствия может иметь его деятельность. Для оценки типов атак необходимо знать некоторые ограничения, изначально присущие протоколу TPC/IP. Сеть Интернет создавалась для связи между государственными учреждениями и университетами в помощь учебному процессу и научным исследованиям. Создатели этой сети не подозревали, насколько широко она распространится. Ниже будут представлены основные виды сетевых угроз.

Снифферы пакетов

Сниффер пакетов представляет собой прикладную программу, которая использует сетевую карту, работающую в режиме promiscuous mode (в этом режиме все пакеты, полученные по физическим каналам, сетевой адаптер отправляет приложению для обработки). При этом сниффер перехватывает все сетевые пакеты, которые передаются через определенный домен. В настоящее время снифферы работают в сетях на вполне законном основании. Они используются для диагностики неисправностей и анализа трафика. Однако ввиду того, что

iptables -P FORWARD DROP
iptables -A INPUT -i eth0 -j DROP

4.Решение о допуске фрагментированных пакетов в Iptables необходимо оформить явным образом:

iptables -A FORWARD -f -j ACCEPT

```
root@dhcppc4:~# iptables -P FORWARD DROP
root@dhcppc4:~# iptables -A INPUT -i eth0 -j DROP
root@dhcppc4:~# iptables -A FORWARD -f -j ACCEPT
```

5.Существует два типа распространенных атак, которые необходимо сразу заблокировать. Одна из них называется подделкой (подделываются заголовки IP-пакетов, чтобы казалось, будто внешний пакет имеет внутренний адрес). Делая это, злоумышленник может попасть в вашу сеть, даже если вы используете собственные IP-адреса. Другой тип атаки реализуется отправкой потока пакетов на широковещательный адрес сети, чтобы перегрузить ее. Это называется штормовой атакой. Атаки перечисленных типов можно блокировать с помощью двух простых инструкций:

iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -j DROP
iptables -A FORWARD -p icmp -i eth0 -d 192.168.0.255 -j DROP

```
root@dhcppc4:~# iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -j DROP
root@dhcppc4:~# iptables -A FORWARD -p icmp -i eth0 -d 192.168.0.255 -j DENY
iptables v1.4.14: Couldn't load target `DENY':No such file or directory

Try `iptables -h' or 'iptables --help' for more information.
root@dhcppc4:~# iptables -A FORWARD -p icmp -i eth0 -d 192.168.0.255 -j DROP
root@dhcppc4:~#
```

Первая инструкция предписывает отбрасывать все пакеты, приходящие из Интернет-интерфейса eth0 с внутренним адресом 192.168.0.0/24. По определению ни один пакет не должен приходить из недоверенного интерфейса с внутренним, собственным исходным адресом. Вторая инструкция отвергает все приходящие извне на адрес внутренней сети широковещательные пакеты протокола ICMP.

6.Вы, как правило, желаете принимать входящие потоки данных, поступающие по соединениям, инициированным изнутри (например, кто-то просматривает web-страницу). Пока соединение, инициированное изнутри, поддерживается - все, наверное, хорошо.

и сможет отвечать на них так, будто он является санкционированным пользователем.

Отказ в обслуживании (Denial of Service - DoS)

DoS, без всякого сомнения, является наиболее известной формой хакерских атак. Кроме того, против атак такого типа труднее всего создать стопроцентную защиту. Даже среди хакеров атаки DoS считаются тривиальными, а их применение вызывает презрительные усмешки, потому что для организации DoS требуется минимум знаний и умений. Тем не менее, именно простота реализации и огромный причиняемый вред привлекают к DoS пристальное внимание администраторов, отвечающих за сетевую безопасность.

Атаки DoS отличаются от атак других типов. Они не нацелены на получение доступа к вашей сети или на получение из этой сети какой-либо информации. Атака DoS делает вашу сеть недоступной для обычного использования за счет превышения допустимых пределов функционирования сети, операционной системы или приложения.

Атаки типа Man-in-the-Middle

Для атаки типа Man-in-the-Middle хакеру нужен доступ к пакетам, передаваемым по сети. Такой доступ ко всем пакетам, передаваемым от провайдера в любую другую сеть, может, к примеру, получить сотрудник этого провайдера. Для атак этого типа часто используются снифферы пакетов, транспортные протоколы и протоколы маршрутизации. Атаки проводятся с целью кражи информации, перехвата текущей сессии и получения доступа к частным сетевым ресурсам, для анализа трафика и получения информации о сети и ее пользователях, для проведения атак типа DoS, искажения передаваемых данных и ввода несанкционированной информации в сетевые сессии.

Эффективно бороться с атаками типа Man-in-the-Middle можно только с помощью криптографии. Если хакер перехватит данные зашифрованной сессии, у него на экране появится не перехваченное сообщение, а бессмысленный набор символов. Заметим, что если хакер получит информацию о

Данная строка сообщает, какой интерпретатор использовать для выполнения команд. Вы должны иметь его в своей ОС, а команды, помещенные в файл, должны быть его корректными командами. Приведенный пример задает маршрутное имя интерпретатора bash в Mandrake Linux. Можно использовать другой интерпретатор, например, Tcsh или Csh. Просто задайте в первой строке его маршрутное имя. Затем сохраните файл.

Сделайте файл исполнимым, чтобы интерпретатор мог выполнить его как программу. Это делается с помощью команды chmod. Введите

chmod 700 имя_командного_файла

Такой режим доступа делает файл читаемым, записываемым и исполнимым.

Чтобы выполнить командный файл, наберите его имя в командной строке. (В bash необходимо задать ./ перед именем файла, расположенного в текущем каталоге.) После нажатия клавиши ввода должны выполниться команды из файла.

Вы должны находиться в том каталоге, где размещен командный файл, или задать его маршрутное имя. Чтобы он выполнялся из любого места, можно добавить этот каталог в переменную окружения PATH или поместить файл в один из каталогов, фигурирующих в значении $PATH.

переадресации получает прямой доступ к защищенному хосту. Примером приложения, которое может предоставить такой доступ, является netcat.

1.3 Защита от сетевых угроз

Смягчить угрозу сниффинга пакетов можно с помощью следующих средств:

Аутентификация

Сильные средства аутентификации являются первым способом защиты от сниффинга пакетов. Под "сильным" мы понимаем такой метод аутентификации, который трудно обойти. Примером такой аутентификации являются однократные пароли (OTP - One-Time Passwords). OTP - это технология двухфакторной аутентификации, при которой происходит сочетание того, что у вас есть, с тем, что вы знаете. Типичным примером двухфакторной аутентификации является работа обычного банкомата, который опознает вас, во-первых, по вашей пластиковой карточке и, во-вторых, по вводимому вами ПИН-коду. Для аутентификации в системе OTP также требуется ПИН-код и ваша личная карточка. Под "карточкой" (token) понимается аппаратное или программное средство, генерирующее (по случайному принципу) уникальный одномоментный однократный пароль. Если хакер узнает этот пароль с помощью сниффера, эта информация будет бесполезной, потому что в этот момент пароль уже будет использован и выведен из употребления. Заметим, что этот способ борьбы со сниффингом эффективен только для борьбы с перехватом паролей. Снифферы, перехватывающие другую информацию (например, сообщения электронной почты), не теряют своей эффективности.

Коммутируемая инфраструктура

Еще одним способом борьбы со сниффингом пакетов в вашей сетевой среде является создание коммутируемой инфраструктуры. Если, к примеру, во всей организации используется коммутируемый Ethernet, хакеры могут получить доступ только к трафику, поступающему на тот порт, к которому они подключены. Коммутируемая инфраструктуры не ликвидирует угрозу сниффинга, но заметно снижает ее остроту.

Анти-снифферы

Специфи кация правила	Описание	
	LOG	Протоколирует пакет в файле.
	MARK	Помечает пакет для дальнейших действий.
	TOS	Изменяет поле TOS (тип обслуживания).
	MIRR OR	Меняет местами исходный и целевой адреса и посылает пакеты обратно, по сути "отражая" их назад отправителю.
	SNAT	Трансляция исходных сетевых адресов. Эта опция применяется при выполнении трансляции сетевых адресов. Исходный адрес преобразуется в другое статическое значение, определенное с помощью ключа - to-source.
	DNAT	Трансляция целевых сетевых адресов. Данная опция аналогична предыдущей, но применяется к целевым адресам.
	MASQ	Маскарад с помощью общедоступного IP-адреса.
	REDIR ECT	Перенаправля ет пакет.

Написание командных файлов

Часто требуется автоматизировать некоторый процесс или иметь одну команду для запуска нескольких инструкций. Применительно к межсетевому экрану обычно желательно,

Таблица 4 – команды IPTables

Команда	Описание
-A цепочка	Добавляет в конец указанной цепочки одно или несколько правил, заданных в инструкции вслед за командой
-I цепочка	Вставляет правила в позицию с

гражданином"). Для этого необходимо отбраковывать любой исходящий трафик, исходный адрес которого не является одним из IP-адресов вашей организации. Этот тип фильтрации, известный под названием "RFC 2827", может выполнять и ваш провайдер (ISP). В результате отбраковывается весь трафик, который не имеет исходного адреса, ожидаемого на определенном интерфейсе. К примеру, если ISP предоставляет соединение с IP-адресом 15.1.1.0/24, он может настроить фильтр таким образом, чтобы с данного интерфейса на маршрутизатор ISP допускался только трафик, поступающий с адреса 15.1.1.0/24. Заметим, что до тех пор, пока все провайдеры не внедрят этот тип фильтрации, его эффективность будет намного ниже возможной. Кроме того, чем дальше от фильтруемых устройств, тем труднее проводить точную фильтрацию. Так, например, фильтрация RFC 2827 на уровне маршрутизатора доступа требует пропуска всего трафика с главного сетевого адреса (10.0.0.0/8), тогда как на уровне распределения (в данной архитектуре) можно ограничить трафик более точно (адрес - 10.1.5.0/24).

Наиболее эффективный метод борьбы с IP-спуфингом тот же, что и в случае со сниффингом пакетов: необходимо сделать атаку абсолютно неэффективной. IP-спуфинг может функционировать только при условии, что аутентификация происходит на базе IP-адресов. Поэтому внедрение дополнительных методов аутентификации делает этот вид атак бесполезными. Лучшим видом дополнительной аутентификации является криптографическая. Если она невозможна, хорошие результаты может дать двухфакторная аутентификация с использованием одноразовых паролей.

Угроза атак типа DoS может снижаться тремя способами:

Функции анти-спуфинга

Правильная конфигурация функций анти-спуфинга на ваших маршрутизаторах и межсетевых экранах поможет снизить риск DoS. Эти функции, как минимум, должны включать фильтрацию RFC 2827. Если хакер не сможет замаскировать свою истинную личность, он вряд ли решится провести атаку.

Функции анти-DoS

определению TCP такой пакет — всего лишь элемент открытия сеанса.

Существует также понятие «связанных соединений». Например, когда в ответ на UDP-пакет с нашего хоста удаленный хост отвечает ICMP-пакетом icmp-port-unreachable, формально этот ответ является отдельным соединением, так как использует совсем другой протокол.

В некоторых случаях целесообразно отключить отслеживание состояния соединений. Например, если ваш сервер находится под (D)Dos-атакой типа флуд, и вам удалось локализовать ее источники, отслеживать соединения с атакующих хостов и тратить для этого ресурсы своей системы явно не имеет смысла.

Критерий состояния соединения

При помощи критерия conntrack вы можете классифицировать пакеты на основании их отношения к соединениям. В частности, состояние NEW позволяет выделять только пакеты, открывающие новые соединения, состояние ESTABLISHED — пакеты, принадлежащие к установленным соединениям, состоянию RELATED соответствуют пакеты, открывающие новые соединения, логически связанные с уже установленными (например, соединение данных в пассивном режиме FTP). Состояние INVALID означает, что принадлежность пакета к соединению установить не удалось.

Например, одним простым правилом

iptables -I INPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT

вы можете обеспечить корректное пропускание всех входящих пакетов, принадлежащих установленным соединениям, и сконцентрироваться только на фильтрации новых соединений.

Заменив в предыдущем правиле ESTABLISHED на ESTABLISHED,RELATED и подгрузив соответствующие модули ядра, вы автоматически обеспечите корректную фильтрацию протоколов, использующих связанные соединения — FTP, SIP, IRC, H.323 и других.

Использование

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 LINUX IpTables

IpTables сохраняет идеологию, ведущую начало от ipfwadm: функционирование фаервола определяется набором правил, каждое из которых состоит из критерия и действия, применяемого к пакетам, подпадающим под этот критерий. В ipchains появилась концепция цепочек — независимых списков правил. Были введены отдельные цепочки для фильтрации входящих (INPUT), исходящих (OUTPUT) и транзитных (FORWARD) пакетов. В продолжении этой идеи, в iptables появились таблицы — независимые группы цепочек. Каждая таблица решала свою задачу — цепочки таблицы filter отвечали за фильтрацию, цепочки таблицы nat — за преобразование сетевых адресов (NAT), к задачам таблицы mangle относились прочие модификации заголовков пакетов (например, изменение TTL или TOS). Кроме того, была слегка изменена логика работы цепочек: в ipchains все входящие пакеты, включая транзитные, проходили цепочку INPUT. В iptables через INPUT проходят только пакеты, адресованные самому хосту.

Такое разделение функциональности позволило iptables при обработке отдельных пакетов использовать информацию о соединениях в целом (ранее это было возможно только для NAT). В этом iptables значительно превосходит ipchains, так iptables может отслеживать состояние соединения и перенаправлять, изменять или отфильтровывать пакеты, основываясь не только на данных из их заголовков (источник, получатель) или содержимом пакетов, но и на основании данных о соединении. Такая возможность фаервола называется stateful-фильтрацией, в отличие от реализованной в ipchains примитивной stateless-фильтрации (подробнее о видах фильтрации см. статью о фаерволах). Можно сказать, что iptables анализирует не только передаваемые данные, но и контекст их передачи, в отличие от ipchains, и поэтому может принимать более обоснованные решения о судьбе каждого конкретного пакета.

Все пакеты пропускаются через определенные для них последовательности цепочек, представленных ниже. При прохождении пакетом цепочки, к нему последовательно

внутренние машины могут общаться с внешним миром и инициировать соединения, но внешним машинам не удастся открыть сеанс.

Межсетевые экраны

В Linux существует несколько встроенных экранирующих приложений: Iptables в версиях ядра 2.4x, Ipchains в 2.2x и Ipfwadm в ядре версии 2.0. Большинство межсетевых экранов на платформе Linux делают свое дело, используя одну из этих служебных программ уровня ядра.

Все три упомянутых приложения действуют аналогичным образом. У межсетевых экранов обычно имеется два или больше сетевых интерфейсов, и под Linux это достигается наличием в компьютере двух или большего количества сетевых плат. Один интерфейс обычно соединяется с внутренней ЛВС – доверенный; другой интерфейс предназначен для общедоступной стороны (ГВС). Все пакеты, проходящие через машину, подвергаются обработке определенными фильтрами.

Межсетевые экраны могут фильтровать пакеты на нескольких различных уровнях. Они могут анализировать IP-адреса и блокировать трафик, приходящий от определенных машин или сетей, проверять заголовок TCP и определять его состояние, и на более высоких уровнях анализировать приложение или номер порта TCP/UDP. Межсетевые экраны можно конфигурировать для отбрасывания целых категорий трафика, таких как ICMP. Пакеты типа ICMP, такие как ping, обычно отбрасываются межсетевыми экранами, поскольку они часто используются для исследования сети и атак на доступность.

Существует два метода настройки межсетевых экранов. Можно в качестве исходного принять положение "разрешить все" и затем задавать поведение, которое требуется блокировать, или же начать с положения "запретить все", после чего специфицировать то, что следует разрешить (допустимое поведение пользователей). Безусловно, предпочтительным является исходное положение "запретить все", поскольку при этом автоматически блокируются все потоки данных, если только они не разрешены явным образом. Используя подход "запретить все", вы автоматически блокируете все, что не считается добропорядочной активностью.

Основными компонентами системы IpTables являются:

netfilter

Компонент ядра Linux, обеспечивающий фильтрацию и модификацию трафика. Собственно, именно он и является фаерволом. В состав netfilter входят следующие модули:

- ip_tables — фаервол для протокола IPv4. Обеспечивает фильтрацию пакетов, модификацию их заголовков и трансляцию сетевых адресов.
- ip6_tables — фаервол для протокола IPv6. Обеспечивает фильтрацию пакетов и модификацию их заголовков.
- arp_tables — фаервол для протоколов ARP и RARP. Обеспечивает фильтрацию и модификацию пакетов.
- x_tables — бэкенд для ip_tables, ip6_tables и arp_tables. В этом модуле определены основные операции для работы с фаерволами «таблично-цепочечной» структуры и их компонентами.
- ebtables — Ethernet-фаервол (префикс eb от Ethernet Bridge). В отличие от трех перечисленных выше фаерволов, работающих с протоколами сетевого и более высоких уровней, ebtables работает на канальном уровне, выполняя фильтрацию и модификацию ethernet-кадров, проходящих через сетевые мосты, если таковые имеются на хосте.

В задачи ip_tables и ip6_tables входит :

- Классификация пакетов на основе различных критериев. В качестве критерия могут выступать, например, IP-адреса источника и/или назначения, состояние соединения (новое/уже установленное), порты источника и/или назначения (для протоколов транспортного уровня, имеющих порты), вспомогательные значения в заголовках пакетов (скажем, длина пакета, TOS, TTL) и т. п. Результаты этой классификации используются при решении других задач из данного списка.
- Фильтрация входящих (INPUT), исходящих (OUTPUT) и транзитных (FORWARD) пакетов, сводящуюся либо к их пропусканию (ACCEPT), либо к блокированию (DROP или REJECT). Также поддерживаются дополнительные возможности,

определенную информацию для следующего протокольного уровня. Пакеты помечаются 32-битными порядковыми номерами, чтобы даже в случае прихода в неправильном порядке передаваемые данные можно было собрать заново. Когда пакет пересекает различные части сети, каждый уровень открывается и интерпретируется, а затем оставшиеся данные передаются дальше согласно полученным инструкциям. Когда пакет данных прибывает в место назначения, реальные данные, или полезная нагрузка, доставляются приложению.

В табл. 2 показано, как некоторые сетевые протоколы инкапсулируют данные.

Таблица 2 – Пример пакета данных TCP/IP

Протокол	**Содержимое**	**Уровень модели ВОС**
Ethernet	MAC-адрес	Канальный
IP	IP-адрес	Сетевой
TCP	Заголовок TCP	Транспортный
HTTP	Заголовок HTTP	Прикладной
Прикладные данные	Web-страница	Данные

Можно видеть, что на внешнем "конверте" для наших данных написан адрес Ethernet. Он идентифицирует пакет в сети Ethernet. Внутри этого конверта находится сетевая информация, а именно, IP-адрес; еще глубже находится транспортный уровень, который устанавливает соединение и закрывает его. Затем располагается прикладной уровень с заголовком HTTP, сообщающим web-навигатору, как форматировать страницу. Наконец, мы доходим до реальной полезной нагрузки пакета - содержимого web-страницы. Этот пример иллюстрирует многоуровневую природу сетевых коммуникаций.

Также в рамках данного проекта разрабатываются два набора библиотек:

- libiptc — содержит функции, осуществляющие вышеперечисленные операции по управлению цепочками и правилами. Именно с этими функциями и работают утилиты iptables (библиотека libip4tc) и ip6tables (библиотека libip6tc). Приложения, использующие эти библиотеки, могут обращаться к netfilter напрямую, минуя вызов утилит iptables и ip6tables. В качестве примера такого приложения можно назвать Perl-модуль IPTables::libiptc, предоставляющий доступ к функциям этих библиотек из Perl-скриптов.
- libipq — содержит функции, позволяющие пользовательским приложениям принимать IP-пакеты на обработку. Отправка со стороны ядра выполняется через модуль ip_queue (ip6_queue для IPv6), что соответствует действию QUEUE. В настоящее время этот механизм объявлен устаревшим, и вместо него рекомендуется использовать действие NFQUEUE, работа которого реализуется через модуль ядра nfnetlink_queue и библиотеку libnetfilter_queue.

conntrack

Компонент netfilter, обеспечивающий отслеживание состояния соединений и классификацию пакетов с точки зрения принадлежности к соединениям, что позволяет netfilter осуществлять полноценную stateful-фильтрацию трафика. Как и netfilter, система conntrack является частью ядра Linux. К его задачам относятся:

- Отслеживание состояний отдельных соединений с тем, чтобы классифицировать каждый пакет либо как относящийся к уже установленному соединению, либо как открывающий новое соединение. При этом понятие "состояние соединения" искусственно вводится для протоколов, в которых оно изначально отсутствует (UDP,ICMP). При работе же с протоколами, поддерживающими состояния (например, TCP), conntrack активно

другим узлом. Примером адреса канального уровня служит MAC-адрес сетевой платы (Medium Access Control - управление доступом к среде передачи). MAC-адрес является числом, которое уникальным образом идентифицирует плату компьютера в сети. В 1970-80-х годах корпорации использовали много различных типов стандартов канального уровня, определенных по большей части их поставщиками оборудования. В наше время большинство организаций используют Ethernet, так как он широко распространен и недорог.

Сетевой уровень

Этот уровень является первой частью видимого взаимодействия с сетями TCP/IP. Сетевой уровень дает возможность взаимодействовать через различные физические сети с помощью вторичного уровня идентификации. В сетях TCP/IP для этого используется IP-адрес. IP-адрес на компьютере помогает осуществлять маршрутизацию данных при передаче из одного места в другое в сети и через Интернет. Этот адрес является уникальным числом для идентификации компьютера в IP-сети - ни одна другая машина в Интернете не может иметь такой адрес. Это справедливо для обычных открыто маршрутизируемых IP-адресов.

При возникновении сетевых проблем, для их решения могут использоваться MAC-адреса, а также сетевые анализаторы. С их помощью можно проследить источник проблемного сетевого трафика. MAC-адреса обычно регистрируются серверами DHCP в Windows или межсетевыми экранами, поэтому можно сопоставить MAC-адреса с определенным IP-адресом или именем машины.

Транспортный уровень

Этот уровень обеспечивает доставку пакета данных из точки А в точку В. На этом уровне располагаются протоколы TCP и UDP. TCP (Transmission Control Protocol - протокол управления передачей) по сути обеспечивает согласованность отсылки пакетов и их приема на другом конце. Он позволяет исправлять ошибки на уровне битов, повторно передавать потерянные сегменты и переупорядочивать фрагментированный трафик и пакеты. UDP (User Datagram Protocol - пользовательский дейтаграммный протокол) является менее тяжеловесной схемой, используемой для потоков мультимедийных данных и кратких взаимодействий с

утилитами, такими как IpTables. Имеется детально разобранный пример и варианты для выполнения лабораторной работы.

С методическими указаниями можно ознакомиться в приложении А.

Используя утилиту IPTables, встроенную в систему Linux с ядром версии 2.4 и выше, настроить межсетевой экран защищающий от наиболее распространенных атак.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть;
2. Произвести настройку межсетевого экрана в соответствии с методическими требованиями, подкрепляя каждый из этапов работы скриншотом;
3. Составить отчет;

СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист;
2. Теоретическая часть;
3. Текст командного файла и скриншоты;
4. Вывод;

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Основы архитектуры сетей

Прежде чем перейти к подлинному пониманию сетевой безопасности, необходимо понять архитектуру сетей. В данном разделе приведен краткий обзор сетевых концепций и терминов. Знакомство с ними поможет вам понять основы протокола TCP/IP.

Как вы, вероятно, знаете, конструкцию каждой сети можно разделить на семь логических частей, каждая из которых решает определенную часть коммуникационной задачи. Эта семиуровневая конструкция называется Эталонной моделью взаимосвязи открытых систем (BOC). Она была разработана Международной организацией по стандартизации (ISO) для представления логической модели описания сетевых коммуникаций, и она помогает поставщикам стандартизовать оборудование и

СПИСОК ЛИТЕРАТУРЫ

1. Дэвид В. Чепмен, мл., Энди Фокс. Брандмауэры Cisco Secure PIX = Cisco® Secure PIX® Firewalls. — М.: «Вильямс», 2003. — С. 384. — ISBN 1-58705-035-8.

2. Оглтри Т. "Firewalls. Практическое применение межсетевых экранов", Пресс, 2001 год.

3. Мэйволд, Э. Безопасность сетей. - М: Эком, 2005. – 528 с.

4. Максимов, В. Межсетевые экраны. Способы организации защиты// КомпьютерПресс. – 2003. - № 3.

5. Андрей Киселев, Iptables Tutorial 1.1.16, 2001-2002 by Oskar Andreasson.

6. Галатенко В.А. Основы информационной безопасности. - М.: ИНТУИТ.РУ "Интернет-Университет Информационных Технологий, 2005.

7. Емельянова Н.В., Партыка Т.Л., Попов И.И. Основы построения автоматизированных информационных систем: Учебник. – М.: ФОРУМ: ИНФРА-М, 2005.

8. Крис Касперский, Техника сетевых атак, 2001.

9. Блум Р., Бреснахэн К., Командная строка Linux и сценарии оболочки. Библия пользователя. 2-е изд., 2012. – 784 стр.

10. Немет Э., Снайдер Г., Хейн Т., Уэйли Б., Unix и Linux. Руководство системного администратора. 4-е изд., 2012. – 1312 стр.

ОРГАНИЗАЦИЯ КОНТРОЛЯ СЕТЕВЫХ ПАКЕТОВ СРЕДСТВАМИ ОС LINUX

Методические указания по выполнению лабораторной работы по дисциплине «Программно-аппаратные средства обеспечения информационной безопасности» для студентов специальностей ...

Курск 2015

УДК 004

Составители: М.О. Таныгин, А.А. Кретов, А.В. Захарченко

Рецензент
Кандидат технических наук, доцент кафедры
защиты информации и систем связи А.Г. Спеваков