PortSentry - средство противодействия сканированию портов

Материал из Ай да Linux Wiki
Перейти к навигации Перейти к поиску

Содержание

Введение

Сканирование портов является одним из самых распространенных и простых способов узнать, какая операционная система установлена на компьютере, какие службы запущены в данный момент и получить другую информацию о компьютере, подключенном к Internet , которая может быть использована для взлома и проникновения. Существует много программ для обнаружения сканирования портов. Но обнаружения не достаточно. Должна последовать адекватная реакция. "Адекватная реакция" может заключаться в отправке в сторону сканирующего тебя человека неправильного фрагментированного пакета, ответного сканирования портов, установки на него firewall'a и т.д. Также желательно, чтоб он получил недостоверную информацию об открытых портах на твоем компьютере. Все это и многое другое позволяет делать прекрасная программа Psionic Software Portsentry.

Когда сканирование обнаружено могут последовать следующие ответы:

- занесение информации об инциденте в системный журнал через syslog().
- Компьютер замеченный в сканировании автоматически заносится в файл /etc/host.deny для TCP Wrappers.
- Локальный компьютер автоматически перенастраивается, чтобы направлять весь трафик от атакующего на несуществующий компьютер.
- Локальный компьютер автоматически перенастраивается, чтобы блокировать все пакеты от атакующего пакетным фильтром.

Цель этой программы – дать администратору информацию о том, что их сервер исследуется.

Установка

Предупреждение: в версии portsentry 1.2 было найдено несколько багов, и поэтому, я рекомендую использовать версию 1.2-r1. Хотя и там ситуация не изменилась. Проблема заключалась в некорректной остановке демона. Ниже будет приведено решение данной проблемы.

Для этого необходимо:

```
echo net-analyzer/portsentry ~x86 >> /etc/portage/package.keywords
```

Устанавливаем программу:

```
emerge -av net-analyzer/portsentry
```

Настройка

Предупреждение: Если у вас проблемы с остановкой или перезапуском демона, то необходимо поправить файл **`/etc/init.d/portsentry`**, а точнее разделы *stop* и *restart*.

```
stop() {
      ebegin "Stopping portsentry"
      start-stop-daemon --stop --quiet --exec /usr/bin/portsentry
      eend $?
}

restart() {
        $stop
        $start
}
```

Режим работы

Portsentry возможно запускать в трех режимах для каждого протокола. Одновременно можно использовать только один режим работы на одном протоколе.

Режим работы задается в файле:

`/etc/conf.d/portsentry`

Достаточно раскомментировать необходимую стройку.

Classic

Работая в данном режиме Portsentry открывает порты, указанные в TCP_PORTS или UDP_PORTS и находится в состоянии ожидания соединения. При попытке подключения к перечисленному порту происходит блокирование удаленного хоста. В этом режиме Portsentry не реагирует на Stealth-сканирование. Данный режим работы задается опциями командной строки: -tcp и -udp , для TCP и UDP-портов соответственно.

Enhanced Stealth Scan Detection

Данный режим используется для проверки перечисленных в TCP_PORTS или UDP_PORTS портов на предмет подключения или сканирования. Отличительная черта, то что палит практически все типы Stealth-сканирования, а не ограничивается только сканирование подключением. Порты, в отличие от предыдущего режима открытыми не держит, посему атакующий получает достоверную информацию об открытых портах. Задается ключиками командной строки: -stcp и -sudp , для TCP и UDP-портов соответственно.

Advanced Stealth Scan Detection

Данный режим используется для проверки всех портов входящих в пул от 1 до ADVANCED_PORT_TCP (для TCP) или ADVANCED_PORT_UDP (для UDP). Порты, открытые работающими на хосте программами и перечисленные в ADVANCED_EXLUDE_TCP(для TCP) или ADVANCED_EXCLUDE_UDP(для UDP) не проверяются. Любой хост, попытавшийся подключится к порту из этого промежутка, мгновенно блокируется. Наиболее удобный для использования метод, т.к. реакция на сканирование или попытку подключения у данного метода самая быстрая, а также в этом режиме используется гораздо меньше процессорного времени, чем в остальных. Задается ключами из коммандной строки: -atcp и -audp , TCP и UDP-портов соответственно.

Конфигурация

Приступаем к правке основный конфигурационных файлов. Будем отталкиваться от доступных файлов примеров:

```
cp /etc/portsentry/portsentry.conf.sample /etc/portsentry/portsentry.conf
cp /etc/portsentry/portsentry.ignore.sample /etc/portsentry/portsentry.ignore
```

Думаю, достаточно привести пример конфигурационного файла, а разобраться совсем просто.

```
#######################
 # Конфигурации портов #
 #######################
 #
 #
 # Несколько примеров настроены для классического и основного Stealth
 # режимов
 #
 # Я люблю всегда сохранить некоторые порты в нижнем конце диапазона.
 # Это позволит быстро обнаружить последовательное сканирование портов
 # и обычно эти порты не используются (например, tcpmux порт 1)
 #
 # ** Пользователи X-Windows **: Если вы запускаете X на вашем сервере, вам
 # нужно быть уверенным, что PortSentry не привязан к порту 6000 (или порту
 # 2000 для пользователей OpenWindows).
 # Сделав это вы обеспечите правильный старт X-клиента.
 #
 # Эти привязанные порты игнорируются для режима Advanced Stealth Scan
 # Detection Mode.
 #
 # Раскомментируйте это для чрезвычайного анализа:

TCP_PORTS="1,7,9,11,15,70,79,80,109,110,111,119,138,139,143,512,513,514,515,5
40,635,1080,1524,2000,2001,4000,4001,5742,6000,6001,6667,12345,12
346,20034,27665,30303,32771,32772,32773,32774,31337,40421,40425,49724,54320"
```

```
UDP_PORTS="1,7,9,66,67,68,69,111,137,138,161,162,474,513,517,518,635,640,641,
666,700,2049,31335,27444,34555,32770,32771,32772,32773,32774,3133 7,54321"
 #
 # Используйте их, если вы только хотите знать:

#TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,123
46,20034,27665,31337,32771,32772,32773,32774,40421,49724,54320"

#UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,37444,34555,31335,32770,3277
1,32772,32773,32774,31337,54321"
 #
 # Используйте это только для bare-bones

#TCP_PORTS="1,11,15,110,111,143,540,635,1080,1524,2000,12345,12346,20034,3277
1,32772,32773,32774,49724,54320"

#UDP_PORTS="1,7,9,69,161,162,513,640,700,32770,32771,32772,32773,32774,31337,
54321"
 ##########################################
 # Опции Advanced Stealth Scan Detection #
 ##########################################
 #
 # Это номера портов, которые PortSentry должен контролировать в
 # Advanced mode.
 # Любые порты "ниже" этого числа будут контролироваться. Оставьте это
 # для контроля всего ниже 1023.
 #
 # На многих Linux системах нельзя привязать порты выше 61000. Это
 # потому, что эти порты используются как часть IP маскарадинга. Я не
 # рекомендую вам привязываться к этим номерам портов. Реальность: Я не
 # рекомендую использовать порты за 1023, так как это будет приводить к
 # ошибочным предупреждениям. Вы были предупреждены!
 # Не пишите мне если у вас возникли проблемы, потому что я просто
 # высказываю вам свое мнение. Не используйте выше 1023 порта.
 #
 #
 ADVANCED_PORTS_TCP="1024"
 ADVANCED_PORTS_UDP="1024"
 #
 # Это поле говорит PortSentry, что порты (за исключением прослушиваемых
 # демонами) игнорируются. Это полезно для услуг, вызываемых ident,
 # например FTP, SMTP и wrappers, которые могут остаться не
 # запущенными.
 #
 # По установленным здесь портам PortSentry будет просто не отвечать
 # на входящие запросы. Фактически, PortSentry будет их обрабатывать как
 # будто они привязаны к демонам. Заданные по умолчанию порты могут
 # выступать в отчетах, как возможно ложные сигналы тревоги и
 # вероятно должны быть оставлены для всез кроме особо изолированных систем
 #
 # TCP ident и NetBIOS сервисы
 ADVANCED_EXCLUDE_TCP="113,139"
 # Default UDP route (RIP), NetBIOS, bootp broadcasts.
 ADVANCED_EXCLUDE_UDP="520,138,137,67"

  ######################
 # Конфигурационные файлы #
 ######################
 #
 # Игнорируемые хосты
 IGNORE_FILE="/etc/portsentry/portsentry.ignore"
 # Хосты, которым было отказано в доступе (из истории работы)
 HISTORY_FILE="/var/log/portsentry/portsentry.history"
```

```
# Компьютеры доступ которым заблокирован только в этой сессии
# (времмено до следующей перезагрузки)
BLOCKED_FILE="/var/log/portsentry/portsentry.blocked"
##############################
# Различные конфигурационные опции #
##############################
#
# Определять ли "имя" атакующего хоста используя DNS Name resolution
# 1 - определять
# 0 - не определять
RESOLVE_HOST = "0"
###################
# Опции ответов #
###################
# Опции ликвидации атакующего. Каждое из этих действий будет
# выполняться если будет обнаружена атака. Если вы не хотите отдельную
# опцию, то комментируйте ее и она будет пропущена.
#
# Переменная $TARGET$ будет замещена целью атаки, когда атака обнаружена.
# Переменная $PORT$ будет заменятьс портом, который был сканирован.
#
##################
# Опции игнорирования #
##################
# Эти опции позволяют вам допустить автоматический параметры ответа для
# UDP/TCP. Это бывает полезно, если вы хотите получить предупреждение о
# соединениях, но не хоти реагировать на определенный протокол.
# Для предотвращения возможных Denial of service атак через UDP и
# определение stealth сканирования для TCP, вы можете пожелать отключить
# блокирование, но оставить предупреждение.
# Лично я предпочитаю ждать начала возникновения проблем до того как
# что-нибудь предпринять, так как большинство атакующих ничего не делают.
# Третья опция позволяет вам запускать внешнюю команду в случае
# сканирования. Это может быть полезно, например, для администраторов,
# которые хотят блокировать TCP, но для UDP будет высылаться
# предупреждения и т.д.
#
#
# 0 = Не блокировать UDP/TCP сканирование.
# 1 = Блокировать UDP/TCP сканирование.
# 2 = Запуск внешней команды (KILL_RUN_CMD)
BLOCK_UDP="1"
BLOCK_TCP="1"
###################
# Сброс маршрутов #
###################
# Эти команды используются для удаления маршрута или
# хоста в локальную таблицу фильтрации.
#
# Шлюз (333.444.555.666) идеально должен быть неработающий хост
# в локальной подсети. На некоторых хостах вместо него используется
# localhost (127.0.0.1), что дает тот же эффект. ЗАМЕТИМ ЧТО
# 333.444.555.66 НЕ БУДЕТ РАБОТАТЬ, ИЗМЕНИТЕ ЭТО!!
#
# ВСЕ ОПЦИИ ОТКЛЮЧЕНИЯ МАРШРУТОВ ИЗНАЧАЛЬНО
# ЗАКОММЕНТИРОВАНЫ. Убедитесь, что вы раскомменировали
# правильные строки для вашей ОС. Если вашей ОС нет в списке
# и вы точно знаете команды сбрасывающие маршруты, то,
# пожалуйста, пришлите их мне. ТОЛЬКО ОДНА KILL_ROUTE ОПЦИЯ
# МОЖЕТ БЫТЬ ИСПОЛЬЗОАНА ОДНОВРЕМЕННО, ТАК ЧТО
# НЕ РАСКОММЕНТИРУЙТЕ БОЛЬШЕ ОДНОЙ СТРОКИ.
#
# ЗАМЕЧАНИЕ: route команды это наименее оптимальная дорога блокирования
# и она не предоставляет полной защиты от UDP атак и
```

```
 # будет спокойно создавать предупреждения для UDP и stealth сканирований.
 # Я всегда рекомендую вам использовать пакетный фильтр, потому что это
 # соответствует замыслу.
 #
 # Общий
 #KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"
 # Общий для Linux
 #KILL_ROUTE="/sbin/route add -host $TARGET$ gw 333.444.555.666"
 # Новые версии Linux поддерживают сейчас флаг reject. Это лучше, чем
 # вышестоящая опция
 #KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
 # Общие для BSD (BSDI, OpenBSD, NetBSD, FreeBSD)
 #KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"
 # Общий для Sun
 #KILL_ROUTE="/usr/sbin/route add $TARGET$ 333.444.555.666 1"
 # NEXTSTEP
 #KILL_ROUTE="/usr/etc/route add $TARGET$ 127.0.0.1 1"
 # FreeBSD
 #KILL_ROUTE="route add -net $TARGET$ -netmask 255.255.255.255 127.0.0.1 -
blackhole"
 # Digital UNIX 4.0D (OSF/1 / Compaq Tru64 UNIX)
 #KILL_ROUTE="/sbin/route add -host -blackhole $TARGET$ 127.0.0.1"
 # Общие для HP-UX
 #KILL_ROUTE="/usr/sbin/route add net $TARGET$ netmask 255.255.255.0
127.0.0.1"
 ##
 # Использование пакетного фильтра более предпочтительный метод. Ниже
 # перечисленные строки подходят для многих ОС. Помните, вы можете
 # раскомментировать только одну строку.
 # Опции KILL_ROUTE.
 ##
 # Linux с поддержкой ipfwadm
 #KILL_ROUTE="/sbin/ipfwadm -I -i deny -S $TARGET$ -o"
 #
 #Linux с поддержкой ipfwadm (без логгирования)
 #KILL_ROUTE="/sbin/ipfwadm -I -i deny -S $TARGET$"
 #
 #Linux с поддержкой ipchain
 #KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"
 #
 #Linux с поддержкой ipchain (без логгирования)
 #KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY"
 #
 #Linux с поддержкой iptables
 KILL_ROUTE="/sbin/iptables -I portscan_deny -s $TARGET$ -j DROP"
 #
 #Linux с поддержкой iptables support с лимитированием и логгирование
 # лимитирование пакетов обеспечит защиту от DOS атак
 # KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP && /sbin/iptables
-I INPUT -s $TARGET$ -m limit –limit 3/minute –limit-burst 5 -j LOG –log-
level DEBUG –log-prefix 'Portsentry: dropping: '"
 #
 # For those of you running FreeBSD (and compatible) you can
 # use their built in firewalling as well.
 #
 #KILL_ROUTE="/sbin/ipfw add 1 deny all from $TARGET$:255.255.255.255 to any"
 #
 #
 # For those running ipfilt (OpenBSD, etc.)
 # NOTE THAT YOU NEED TO CHANGE external_interface TO A VALID INTERFACE!!
 #
 #KILL_ROUTE="/bin/echo 'block in log on external_interface from $TARGET$/32
to any' | /sbin/ipf -f -"
```

```
 ###############
 # TCP Wrappers#
 ###############
 # Этот текст описывает внесение в файл hosts.deny для использования
wrappers.
 # Зжесь приводятся два формата TCP wrappers:
 #
 # Формат 1: Старый стиль - по умолчанию, когда хост не допускает обработки
 # параметров.
 #
 #KILL_HOSTS_DENY="ALL: $TARGET$"
 # Формат 2: Новый стиль √ включены расширенной обработки.
 # Вы можете просмотреть опции расширенной обработки, чтобы
 # быть уверенными, что все перед символами "%" стоит символ "\"
 # (например, \%c \%h )
 #
 #KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"
 ###################
 # Внешние команды #
 ###################
 # Эта команда выполняется когда хост подключен, это может быть то, что вам
 # нужно (пэйджер и т.д.). Эта команда выполняется перед очисткой маршрута.
 #
 # Я НЕ РЕКОМЕНДУЮ ПОМЕЩАТЬ КАКИЕ-ЛИБО КАРАТЕЛЬНЫЕ
 # ДЕЙСТВИЯ ПРОТИВ ХОСТА СКАНИРУЮЩЕГО ВАС.
 #
 #TCP/IP это протокол без подтверждения подлинности и люди могут организовать
 # сканирование из ниоткуда. Единственное, что можно безопасно запустить
 # это скрипт обратной проверки, который использует классический -tcp режим.
 # Этот режим требует полного соединения и очень труден для обмана (spoof).
 #
 # Переменная KILL_RUN_CMD_FIRST должна быть выставлена в "1" для запуска
внешней команды
 # ДО блокирования. Установите в "0" для выполнения команды ПОСЛЕ
блокирования
 #
 #KILL_RUN_CMD_FIRST = "0"
 #
 #
 KILL_RUN_CMD="/usr/home/script/work/scan_port_mail.sh $TARGET$ $PORT$
$MODE$"
 # for examples see /usr/share/doc/portsentry/examples/

 ####################
 # Значение триггеров сканирования #
 ####################
 # Введите число соединений к портам, когда вам будет дано предупреждение.
 # По умолчанию значение равно 0 - незамедлительная реакция.
 # Значения 1 или 2 будут уменьшать количество ложных срабатываний. В более
 # высоком значение нет необходимости. Это значение должно быть определено,
 # но как правило можно оставить 0.
 #
 # ЗАМЕЧАНИЕ: Если вы используете продвинутые опции определения, вам
 # нужно быть внимательным, чтобы не создать ситуацию "спускового крючка,
 # требующего легкого нажатия". Поскольку расширенный режим будет
 # реагировать на любой удаленный компьютер соединяющийся с интервалом
 # ниже определенного здесь, вы при определенных обстоятельствах
 # действительно разорвете что-нибудь. (например, кто-то невинно пытается
 # соединиться с вами через SSL [TCP порт 443] и вы сразу блокируете его).
 # Так что будьте внимательны.
 #
 SCAN_TRIGGER="0"
 ####################
 # Секция заголовка (banner) порта #
```

```
 ######################
 # Введите здесь текст, который вы хотите показать человеку отключаемого
 # PortSentry. Я не рекомендую насмехаться над человек, так как это может его
 # разозлить. Оставьте эти строки закомментированными, чтобы отключить эту
 # возможность.
 #
 # Режим определения Stealth сканирования не использует эту возможность.
 #
 #PORT_BANNER="** UNAUTHORIZED ACCESS PROHIBITED *** YOUR CONNECTION ATTEMPT
HAS BEEN LOGGED. GO AWAY."
 # EOF
```

В файле portsentry.ignore мы упоминаем IP-адреса компьютеров, которые не должны быть блокированы при подключении к отслеживаемому порту. По умолчанию, в файле заданы два IP-адреса: 127.0.0.1 и 0.0.0.0. туда следует добавить внутреннюю сеть и свои сервера.

После этого создаем указанный в конфиге внешний скрипт для отправки почты:

```
mkdir -p /usr/home/script/work/
 cd /usr/home/script/work/
 touch scan_port_mail.sh
 chmod +x scan_port_mail.sh

  ###################### /usr/home/script/work/scan_port_mail.sh
######################
 #!/bin/bash
 # оповещение по почте о попытках сканирования хостов

 # вводим переменные:
 main_e_mail=”root@mail.ru”
 attak_date=”`date +%Y-%m-%d`”
 attak_time=”`date +%H:%M:%S`”
 local_mashine=”`uname -n`”

 # достаём хост с которого сканили
 hacker_IP=$1
 scanned_port=$2

 # определяем DNS-атакующего
 hacker_DNS=`host ${hacker_IP} | awk ‘{print $5}’`

 # ваяем тревожную мессагу
 echo ” обнаружена попытка сканирования .
 Имя машины:             ${local_mashine}
 Отсканенные порты:       ${scanned_port}

 Прикрепляем данные атакующего:
 IP:     ${hacker_IP}
 DNS:    ${hacker_DNS}
 =======
 Атака заблокирована.
 ” | mail -s port_scanned_on_${local_mashine} ${main_e_mail}

 ############################
```

На заметку: Для работы скрипта необходимы: *mail-client/mailx* и *net-dns/bind-tools*

В случае фаервола естественно необходимо открыть прослушиваемые порты, для более успешной рыбалки.

См. также

odminblog.ru
lissyara.su

Источник — https://aidalinux.ru/wiki/index.php?title=PortSentry_-_средство_противодействия_сканированию_портов&oldid=415
Категории:

- Страницы, использующие устаревший тег source
- Руководства
- Net
- Безопасность

Навигация

Персональные инструменты

- Создать учётную запись
- Войти

Пространства имён

- Статья
- Обсуждение

Варианты

Просмотры

- Читать
- Просмотр кода
- История

Ещё

Поиск

Найти Перейти

Навигация

- Заглавная страница
- Main aidalinux.ru

- Все категории
- Все статьи
- Справка

Популярное

- Статьи
- Категории
- Свежие правки
- Случайная статья

Категории

- Man
- Сеть
- Gentoo
- Руководства
- Безопасность

Инструменты

- Ссылки сюда
- Связанные правки
- Служебные страницы
- Версия для печати
- Постоянная ссылка
- Сведения о странице

- Эта страница в последний раз была отредактирована 16 марта 2012 в 22:04.

- Политика конфиденциальности
- О Ай да Linux Wiki
- Отказ от ответственности

-