Защита сервера при помощи Port Knocking

Port knocking — это сетевой защитный механизм, действие которого основано на следующем принципе: сетевой порт является по-умолчанию закрытым, но до тех пор, пока на него не поступит заранее определённая последовательность пакетов данных, которая «заставит» порт открыться. Например, вы можете сделать «невидимым» для внешнего мира порт SSH, и открытым только для тех, кто знает нужную последовательность.

Настройка Сервера

Наиболее популярным ПО для организации port knocking является knockd. Работая в режиме демона совместно с iptables, knockd прослушивает сетевой интерфейс, ожидая корректной последовательности запросов на подключение. Как только knockd отлавливает корректную последовательность, он выполняет команду, определённую в конфигурационном файле knockd для данной последовательности — как правило, это вызов iptables, разрешающий соединение на определённый сетевой порт.

Например, у вас запущен демон SSH, ожидающий входящих подключений на 22 порту. Однако, правилом iptables входящие соединения на 22-й порт запрещены. Knockd, прослушивающий интерфейс eth0, ожидает последовательности из TCP SYN-пакетов на порты 9000, 6501 и 1234. Как только эта последовательность соединений будет обнаружена, knockd при помощи вызова iptables, изменит правило сетевого фильтра таким образом, чтобы разрешить подключение извне к 22-му TCP-порту, на котором уже ожидает SSH-демон.

Вы можете собрать knockd из исходных кодов, полученных на сайте (также, на сайте имеются ссылки на готовые пакеты для различных систем) или же установить, используя менеджер пакетов вашей системы. После установки knockd, его конфигурационный файл можно найти в `/etc/knockd.conf`.

В начале конфигурационного файла находится раздел `[options]`, содержащий глобальные настройки демона. Например, строкой:

```
Interface = eth1
```

вы можете определить, какой интерфейс прослушивать в случае, если в вашей системе он не является единственным.

Далее, после раздела `[options]`, идут описания последовательностей. По-умолчанию их две:

```
[openSSH]
sequence    = 7000,8000,9000
seq_timeout = 5
command     = /sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags    = syn

[closeSSH]
sequence    = 9000,8000,7000
seq_timeout = 5
command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags    = syn
```

Значение параметра `sequence` определяет последовательность. Числа являются номерами TCP-портов. Также, вы можете явно указать, TCP или UDP порт использовать, при помощи суффиксов `:tcp` и `:udp`. Например:

```
sequence = 3333:tcp,9999:udp,1010:udp,8675:tcp
```

Значение параметра `seq_timeout` задаёт максимальное время в секундах, которое отводится на совершение клиентом последовательности подключений. Если клиент не укладывается в это время — подключение будет отклонено.

Значение параметра `command` определяет путь и параметры вызываемой программы в случае обнаружения корректной последовательности.

Параметром `tcpflags` вы можете определять, какие флаги должны иметь пакеты, участвующие в последовательности. Несколько флагов необходимо разделять запятой:

```
tcpflags = syn,ack,urg
```

А для явного исключения отельных флагов нужно использовать восклицательный знак:

```
tcpflags = syn,!ack,urg
```

Другим интересным вариантом конфигурации knockd является использование параметров `start_command`, `cmd_timeout` и `stop_command`:

```
[opencloseSMTP]
one_time_sequences = /etc/knockd/smtp_sequences
seq_timeout        = 15
tcpflags           = fin,!ack
start_command      = /usr/sbin/iptables -A INPUT -s %IP% -p tcp --dport 25 -j ACCEP
cmd_timeout        = 5
stop_command       = /usr/sbin/iptables -D INPUT -s %IP% -p tcp --dport 25 -j ACCEP
```

Параметр `start_command` по смыслу идентичен параметру `command`. Значение параметра `cmd_timeout` определяет временной интервал в секундах, по истечении которого запустится команда, определённая значением параметра `stop_command`. Таким образом, вы можете открывать определённый порт лишь на некоторый промежуток времени.

Как правило, чтобы демон knockd запускался автоматически при старте системы, достаточно в файле `/etc/default/knockd` установить:

```
START_KNOCKD=1
```

Настройка Клиента

Протестировать соединение с вашим сервером вы можете обычным telnet-клиентом, последовательно подключившись к заданным портам.

При повседневном же использовании, естественно, не очень удобно запускать последовательные серии telnet-соединений перед тем, как вам необходимо подключиться к вашему SSH-серверу. В комплекте с демоном knockd поставляется утилита `knock`, которая и призвана осуществлять необходимые серии подключений. Например:

```
knock 192.168.1.100 3333:tcp 9999:udp 1010:udp 8675:tcp
```

Обратите внимание на формат описания номеров и типов портов. Он такой же, как и в файле `/etc/knockd.conf`.

Если придуманные вами последовательности основаны, в том числе и на TCP-флагах пакетов, то вам понадобятся более продвинутые средства, нежели knock. Например, SendIP или packit, умеющие формировать сетевые пакеты с произвольным содержимым.

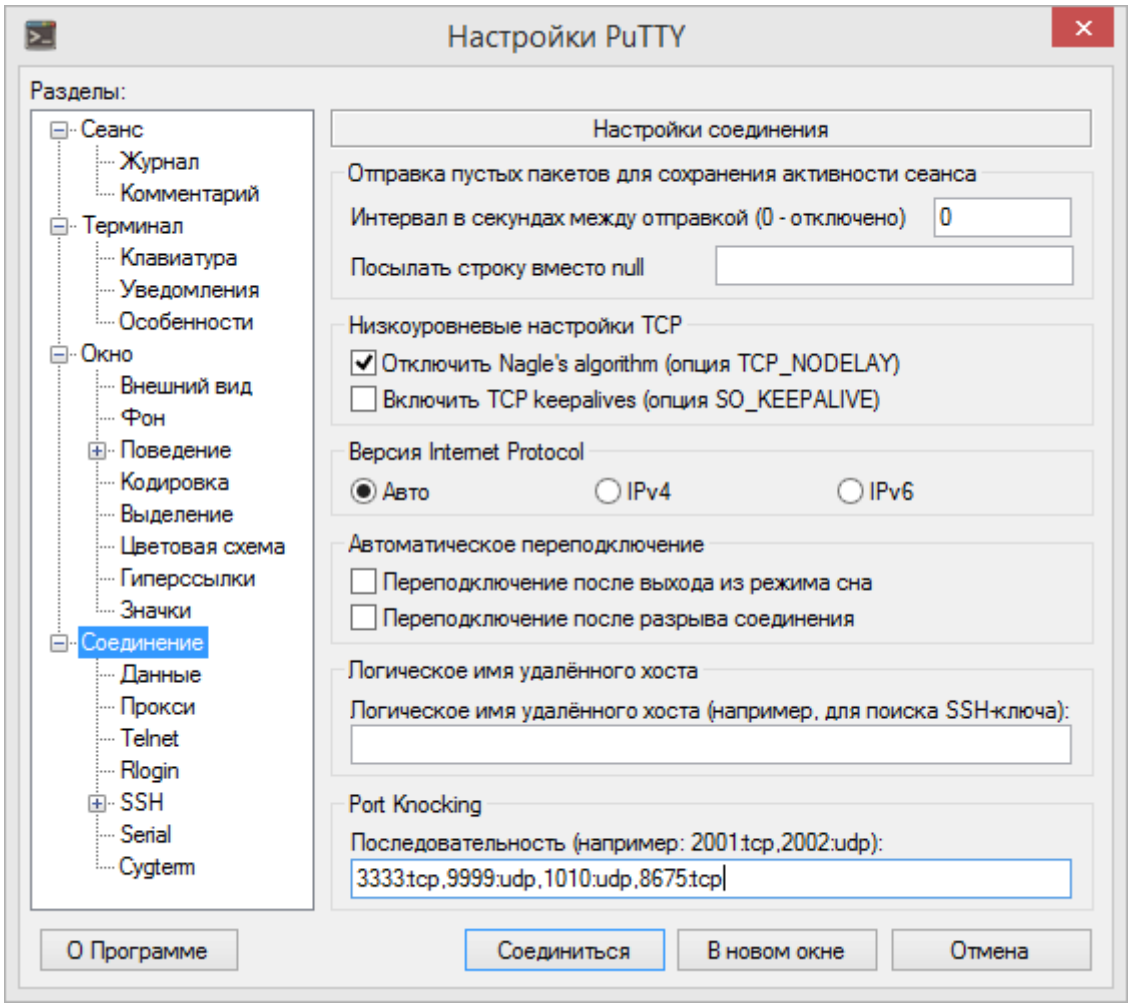
Если вы используете port knocking повсеместно, то вам, вероятно, могут понадобиться knock-клиенты для мобильных платформ.

Для iPhone вы можете попробовать пару knockd-совместимых клиента — Port Knock Lite и KnockOnD, для Android — knock-android.

Port Knocking в PuTTY

Под Windows вы можете использовать русскую сборку PuTTY. Начиная с версии 0.63-RU-15, в ней появилась возможность осуществлять последовательность подключений к закрытым портам по

технологии Port Knocking:



Практические Моменты

Одна из причин, по которой не стоит использовать knock-клиенты, запоминающих секретную последовательность и воспроизводящих её без вашего участия — это очень «похожесть» на сохранение где-то пароля к вашей учётной записи в открытом виде. Секретная последовательность подключений к портам должна находиться лично у вас в памяти и больше нигде. Тогда метод port knocking'а будет на самом деле эффективным.

Также, port knocking критикуют за то, что всегда существует вероятность того, что ваш трафик перехватят и смогут вычленить из него последовательность, и, чем чаще вы используете port knocking, тем выше эта вероятность. Конечно, вы можете дополнять вашу последовательность псевдопоследовательностями для того, чтобы запутать злоумышленников, однако в первую очередь всегда заботьтесь о безопасности самого сервиса, открытый порт которого вы маскируете и рассматривайте port knocking как дополнение к безопасности сервиса, нежели как панацею.

Более сложный метод защиты от прослушивания port knocking состоит в использовании одноразовых секретных последовательностей. Для этого в knockd.conf нужно использовать параметр `one_time_sequences`, значением которого должен быть путь к файлу с определением последовательностей, по одной на строку. После использования каждой последовательности knockd комментирует строку с использованной последовательностью и переключается на следующую.

В том числе имейте ввиду некоторые недостатки использования knockd. Во-первых, если процесс knockd внезапно «упадёт», вы не сможете получить удалённый доступ к маскируемой службе. Во-вторых, помните, что IP-пакеты могут доставлять к вашему серверу разными путями и приходить вовсе не в той последовательности, в которой были отправлены. Именно поэтому не стоит использовать слишком маленькое значение параметра seq_timeout.

Само-собой разумеется, что knockd можно использовать не только для управления правилами сетевого фильтра посредством запуска iptables. Вы можете запускать всё, что вам угодно, начиная от запуска

процесса резервного копирования, заканчивая удалением важных данных с сервера и отправкой заявления об увольнении вашему начальнику.

Продвинутый Port Knocking

Можно установить knockd на серверах, находящихся за «основным» сервером, создав целую цепочку knockd-серверов.

Некоторые проекты, реализующие port knocking, решают проблему «replay»-атак за счёт использования криптоалгоритмов аутентификации. К таким проектам относятся, например, cryptknock, COK, и tariq.

Конечно, более продвинутые механизмы port knocking'а не оставят вам возможности «просто помнить» последовательность номеров портов, однако присутствие массы полезных «фишек» с лихвой перекроет этот «недостаток».

Всем интересующимся технологией port knocking'а рекомендуется посетить страницу portknocking.org.

Итоги

Когда специалист по безопасности скажет вам что весь этот ваш port knocking — лишь ещё один тонкий слой безопасности — он будет прав. Всегда ответственно подходите к защите непосредственно сервисов, как таковых, и не рассматривайте port-knocking как панацею. Port knocking будет полезен, чтобы скрыть запущенные службы от «случайных прохожих» и, если вы регулярно меняете секретные последовательности вашего knock-демона, то, вероятно, и предотвратит bruteforce-атаки по словарю учётных записей SSH.