найти и уничтожить — методы обнаружения (ре)трансляторов сетевых адресов

крис касперски, ака мыщъх, a.k.a. nezumi. a.k.a. souriz, a.k.a. elraton, no-email

внедрение безлимитных тарифов привело к появлению целой армии "партизанок", скрывающихся за NAT/Proxy-серверами и злостно уклоняющимися от исполнения своего воинского долга, тьфу, от абонементной платы. за одним IP-адресом могут прятаться десятки "уклоненцев" и, чтобы их прищемить, администратор должен выявить присутствие трансляторов сетевых адресов и/или proxy-серверов на клиентских узлах, используя доступное программное обеспечение, что всегда под рукой.

введение или против кого мы будет дружить

Прежде, чем бороться, необходимо отчетливо себе представлять с кем (и с чем!) мы, собственно говоря, боремся, а то так недолго и всех клиентов распугать. Компьютер уже давно не роскошь и большинство пользователей имеет по две-три машины, а то и больше (для себя, для сына, для жены, ноутбук или даже DVD/CD/MP3 проигрыватель с Ethernet портом, которому выход в Интернет нужен не только для скачивания файлов, но и считывания названия песен на CD из он-лайновой базы, про DRM тоже не стоит забывать — некоторые устройства требуют проверки аутентичности копии, и без Инетнета не живут).

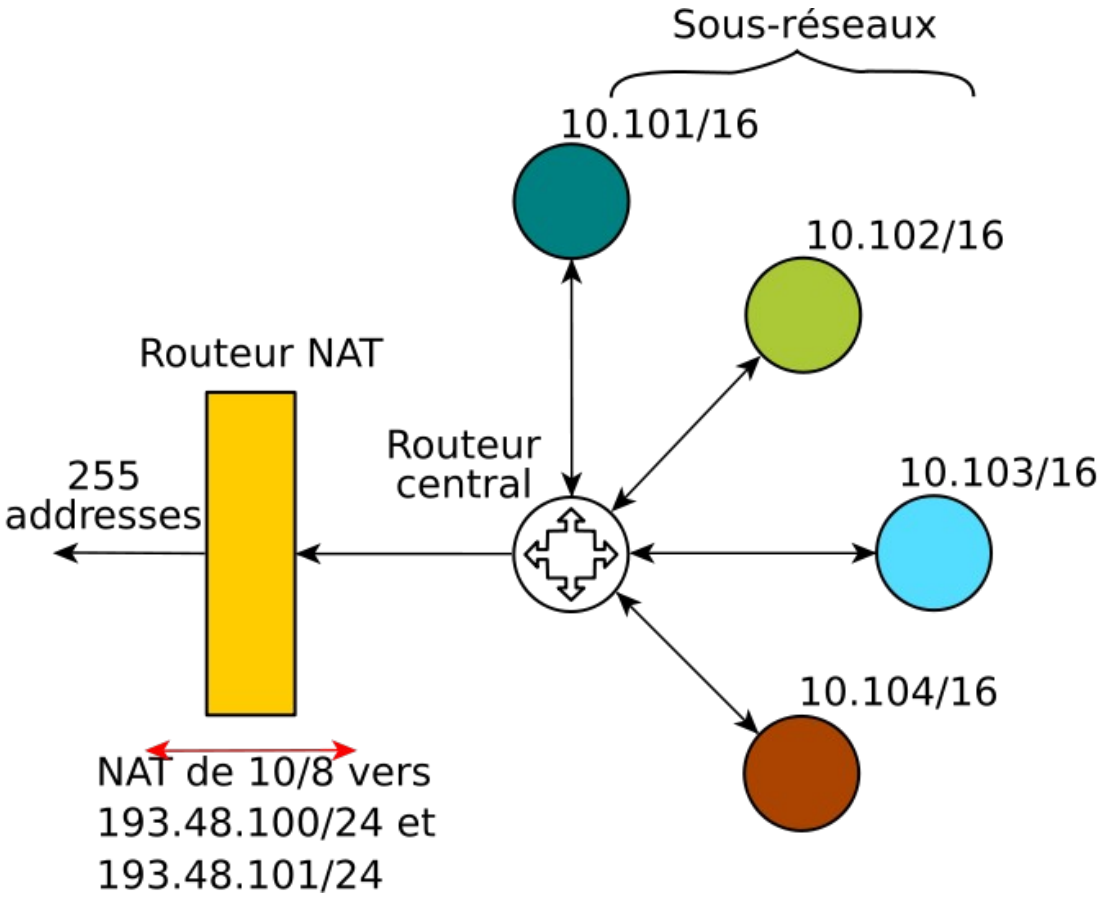

Рисунок 1 за одним NAT'ом может скрываться целый легион "партизан"

Требовать от _всех_ клиентов отдельного DSL-подключения на каждый компьютер, не только не гуманно, но и технически невозможно — это же ведь сколько телефонных линий нужно тянуть!!! Кроме того, некоторые DSL-модемы уже содержат встроенный NAT, который работает всегда, независимо от того сколько узлов к нему подключено — восемь или один.

Наконец, не стоит забывать про виртуальные машины типа VM Ware или Virtual PC. Гостевым операционным системам тоже нужен выход в сеть! А разные брандмауэры, банерорезалки, web-ускорители и прочие программы зачастую работают как proxy-сервер, даже если за ним сидит всего один пользователь!!!

Таким образом, само по себе наличие NAT'ов или proxy-серверов на клиентской машине — еще не повод отрубать последнего от сети (даже если их использование явным образом запрещено в договоре). Тут необходим комплексный анализ и тщательное расследование всех обстоятельств. В конце концов, существует тысячи способов "обуть" провайдера, не нарушив при этом договор. Скажем, получить заказы на скачку файлов по мылу и качать 24 часа в непрерывном режиме без всяких там proxy и NAT'ов, а сами скаченные файлы нарезать на DVD – не слишком удобно, зато честно.

"Правильные" провайдеры, обнаружив факт использования NAT'а, прежде всего смотрят на объем трафика и, если клиент реально "борзеет", пишут ему письмо с просьбой прокомментировать ситуацию. Быть может, это действительно небольшая домашняя сеть или просто аппетит у клиента такой. Ни о каких NAT'ах он не слышал, просто купил модем со встроенным транслятором. Так за что же его отрубать?!

Но довольно слов, перейдем к делу и опишем простые и доступные методики обнаружения NAT'ов и Proxy, которыми может воспользоваться каждый провайдер.

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="4">Версия</td><td colspan="4">IHL</td><td colspan="8">Тип обслуживания</td><td colspan="16">Длина пакета</td></tr>
<tr><td colspan="16">Идентификатор</td><td colspan="3">Флаги</td><td colspan="13">Смещение фрагмента</td></tr>
<tr><td colspan="8">Число переходов (TTL)</td><td colspan="8">Протокол</td><td colspan="16">Контрольная сумма заголовка</td></tr>
<tr><td colspan="32">IP-адрес отправителя (32 бита)</td></tr>
<tr><td colspan="32">IP-адрес получателя (32 бита)</td></tr>
<tr><td colspan="16">Параметры (до 320 бит)</td><td colspan="16">Данные (до 65535 байт минус заголовок)</td></tr>
</table>

Рисунок 2 структура IPv4-пакета (поля, по которым можно определить наличие NAT'а, выделены красным цветом)

IP: TTL

Поле TTL (Time-to-Live – время жизни) в заголовке IP-пакета при проходе через каждый узел уменьшается на единицу, включая узел на котором расположен NAT. Следовательно, значение TTL пакетов, отправленные с машины с NAT-сервером, окажется на единицу больше, чем значение TTL пакетов, отправленных остальными узлами, находящимися за NAT'ом, что легко обнаруживается анализатором трафика.

Судя по форумам, некоторые провайдеры считают этот способ достаточно надежным, а хакеры, пытающиеся их обломать, пишут драйверы-фильтры, перехватывающие IP-пакеты и корректирующие значение TTL (но ведь драйвера еще необходимо уметь писать!). Намного проще указать начальное значение TTL в настройках TCP/IP-стека, что по силам любому пользователю, взявшему в руки твикер. Это раз!

А теперь два — DSL-модемы с Ethernet-портами, имеющее встроенный NAT, уменьшают TTL _всех_ пользователей, а потому данная методика их не обнаруживает.

Рисунок 3 DSL-модем с несколькими Ethernet-портами и встроенным (причем _неотключаемым_) NAT'ом на борту

IP: ID

Поле идентификатора IP пакета согласно RFC 791 должно быть уникально для "*this source, destination pair and protocol for the time the datagram (or any fragment of it) could be alive in the internet*" ("IP-адреса узла-источника/узла-приемника, протокола, дейтаграммы, включая любой ее фрагмент, в течении срока жизни дейтаграммы в сети"). И хотя RFC 791 не указывает пути достижения заданной уникальности, оставляя это на откуп конкретным реализациям TCP/IP стека, все современные ОСи (Linux, BSD и Windows, начиная с W2K) просто генерируют некоторое число, а потом увеличивают его с каждым посланным пакетом на единицу, в результате чего мы получаем простую последовательность, конечно, при условии, что с данным узлом, ассоциирована только одна машина. Две машины, находящиеся за NAT'ом, сгенерируют две последовательности, а если счет машин идет на десятки, то провайдер видит в идентификаторах рандомный мусор, что дает ему все основания прищемить пользователя, крышующего "партизан".

Теоретически, можно написать драйвер-фильтр, корректирующие идентификаторы всех уходящих пакетов, но… он должен быть запущен на машине с NAT-сервером. Если же "злоумышленник" использует DSL-модем с кучей Ethernet-портов, ему придется не по детски извратиться, запустив драйвера фильтры на _всех_ "партизанских" машинах, но в этом случае, некоторые NAT'ы могут поехать крышей, отказавшись функционировать, да и сложность разработки подобного драйвера соответствующая.

Кажется, что анализ идентификаторов IP пакетов идеально подходит для выявления "партизан", но увы… Windows 9x, Me, NT 4.x используют различные алгоритмы генерации IP-идентификатора и скрипты, написанные администраторами, зачастую ошибочно принимают их за толпу "партизан". Конечно, 9x сегодня большая редкость и основная масса народа сидит под XP, однако, это еще не повод, чтобы рубить с плеча. Как уже говорилось выше, прежде чем отрубать клиенту доступ в сеть необходимо на 100% быть уверенным, что он действительно нарушил хотя бы один пункт договора, иначе однажды можно нарваться на типа, знающего законы, лучше чем ты — таблицы марштуизации. По судам затаскает — не отмажешься!

Формат TCP-сегмента

Бит	**0 — 3**	**4 — 7**	**8 — 15**	**16 — 31**
0	Порт источника			Порт назначения
32	Номер последовательности			
64	Номер подтверждения			
96	Смещение	Зарезервировано	Флаги	Окно

	данных	
128	Контрольная сумма	Указатель важности
160	Опции (необязательное)	
160/192+	Данные	

Рисунок 4 формат TCP-сегмента (поля, по которым можно определить наличие NAT'а, выделены красным цветом)

TCP/UDP – диапазон портов источника

Поскольку 99,9% приложений, работающих с сетью, предоставляют операционной системе право самостоятельного назначения порта источника (из числа свободных), то выбирая номера портов определенным образом, можно спрятать за одним IP-адресом очень много узлов. Собственно говоря, идея NAT'ов как раз и основана на том, что они обеспечивают уникальность связки IP-источник:порт-источника → IP-приемник:порт-приемника, "отлавливая" пакеты, поступающих с разных внутренних узлов на один внешний узел за счет "маппинга" номеров портов источника и никой путаницы "чей пакет?" не возникает.

Проблема (если, конечно, это проблема) в том, что большинство NAT'ов использует фиксированный диапазон портов для маппинга, который намного уже диапазона портов, назначаемых операционной системой, а потому, если у провайдера имеется достаточное количество клиентского трафика, и этот трафик сосредоточен в узком диапазоне портов отправителя, то можно предположить, что тут замешан NAT.

Данная методика считается очень надежной, хотя ей присущи свои недостатки. Начнем с того, что NAT'ы бывают встроены не только в DSL-модемы с Ethernet-портами, но даже в модемы, подключаемые по USB!!! То есть, узкий диапазон портов указывает лишь на наличие транслятора, но ничего не говорит о том, сколько пользователей за ним сидит: один или несколько (TTL поле при всей его незатейливости подобных ложных срабатываний не допускает).

Далее, если на машине, генерирующей большое количество трафика, установлен программный NAT, провайдер получит более или менее нормальное распределение по портам и ему нужно очень-очень долго собирать трафик, чтобы заподозрить, что тут что-то не так. С другой стороны, некоторые приложения (например, клиенты файлообменных сетей) поддерживают настройку диапазона используемых портов источника, что с точки зрения провайдера выглядит как наличие NAT'а. Очередная ложная тревога! Так что пользоваться данной методикой следует с оччень большой осторожностью.

Формат UDP-сегмента

+	**Биты 0—15**	**16—31**
0	Порт отправителя	Порт получателя
32	Длина пакета	Контрольная сумма
64	Данные	

Рисунок 5 формат UDP-сегмента (поля, по которым можно определить наличие NAT'а, выделены красным цветом)

прикладной уровень – в охоте за реальными IP

Компьютеры, находящиеся на NAT'ом, должны иметь различные IP-адреса, иначе NAT поедет крышей и запутаться в пакетах, хотя NAT, встроенный в DSL-модем, может, и не запутается, т. к. определяет узлы не по IP-адресам, а по физическим портам (хотя и не обязан этого делать). Следовательно, задача отлова "партизан" сводится к определению реальных IP-адресов клиентских узлов и, если у одного клиента окажется несколько IP…

А как можно определить реальный IP-клиента, ведь NAT автоматически подменяет его во всех IP-пакетах? На IP-уровне нам ловить, действительно, нечего, но вот если подняться на уровень прикладных протоколов, можно обнаружить, что многие программы внедряют IP-

адреса в "свои" пакеты. Так поступают, в частности некоторые почтовые клиенты, instant messenger'ы (MSN, ICQ) и другие "товарищи", на которых партизаны палятся как молодые.

Народ поумнее юзают открытый софт, который ничего и некуда не вставляет — там с этой проблемой разобрались уже давно. К тому же, если у компьютера имеется несколько интерфейсов (локальная сеть, сотовый телефон, периодически работающий как GPRS модем, несколько беспроводных устройств для связи с фотокамерами, etc), то независимо от наличия/отсутствия NAT'а или Proxy, клиентские приложения очень часто ошибаются с определением "настоящего" IP, потому что понятие "настоящего" IP абсурдно и применимо лишь к узлам, имеющим всего один сетевой интерфейс. Компьютеры, имеющие несколько интерфейсов, имеют более одного IP и все они "настоящие". А таблица маршрутизации — штука сложная и несовершенная. Windows вполне может попытаться послать пакет, адресованный внешнему узлу, на беспроводной адаптер домашней локалки, и только убедившись, что он ни хвоста не маршрутизируется, попытать счастья на другом интерфейсе.

Если приложение определяет "свой" IP уже после установки соединения, то все ОК, но… ведь не все приложения такие правильные и многие из них запрашивают у операционной системы список сетевых интерфейсов _до_ установки соединения и в качествен "настоящего" IP берут адрес первого интерфейса, а, поскольку, таблица маршрутизации может меняться при подключении/отключении сетевых устройств, вместе с ней будет меняется и "настоящий" IP. И все это на компьютере, за которым сидит всего один пользователь!!! Ложная тревога… ну сколько же можно?! Увы, против современных технологий не попреешь!

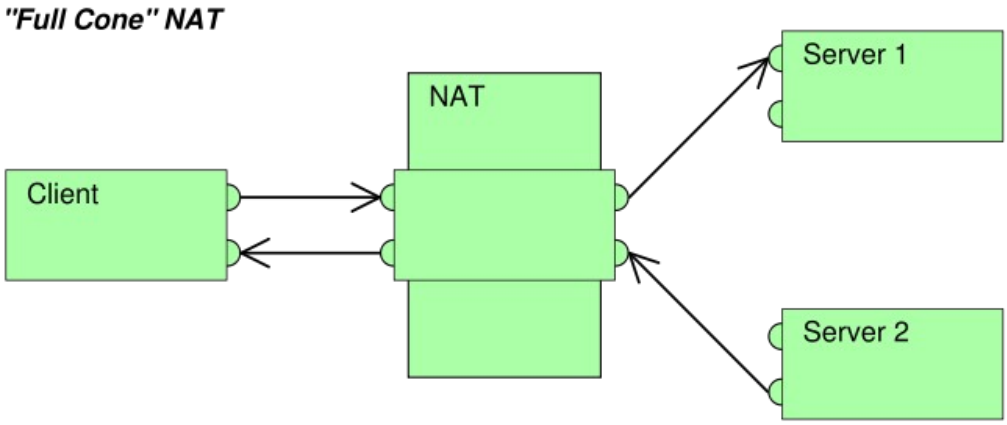

Рисунок 6 принцип действия типового NAT'а

proxy-сервера

Подавляющее большинство proxy-серверов явно прописывает свое присутствие в HTTP-запросах и обнаружить их — не проблема, однако, достаточно многие пользователи устанавливают proxy-сервер не только для совместного доступа Интернет-соединения, но и для кэширования запросов. Горящий Лис, Опера и IE кэшировать тоже умеют, но… далеко не так хорошо, как это делают некоторые proxy-серверы, которые к тому же ведут статистику, отображая ее в наглядной графической форме. А если клиент попеременно использует несколько браузеров, то для экономии трафика и дискового пространства, разумнее всего "запитать" все браузеры от одного proxy-сервера, запретив им самостоятельное кэширование страниц. Бразурер — это же ведь не религия. Если позволено иметь двойное гражданство, почему нельзя использовать несколько браузеров?

Ладно, предположим, что Proxy-сервер скрывает факт своего присутствия. Может ли провайдер его обнаружить? Сканирование портов — метод простой, но увы… Во-первых, если пользователь не полный лох, то: а) повесит proxy на нестандартный порт; б) запретит подключение со всех IP-адресов, кроме локальных. И хотя продвинутые сканеры портов (типа nmap) все-таки обнаружат присутствие закрытого порта, определить его назначение ни за что не смогут (если, конечно, proxy-сервер при попытке подключения со внешних адресов не выдает страничку со злобной надписью "access denied").

Другими словами, тщательно замаскированный Proxy-сервер, обслуживающий закрытую сеть, со стороны провайдера обнаружить невозможно. Все методики либо ненадежны, либо выдают огромное количество ложных позитивных срабатываний, реагируя на различные утилиты, устроенные по принципу proxy-серверов.

заключение

В идеале, провайдер вообще не должен ограничивать свободу клиентов, а если и ограничивать, то в разумных пределах. Провайдер, ставящий клиента раком, и не позволяющий ему разделить трафик с женой, сыном, виртуальной машиной и собачкой жучкой, никому не интересен и к нему идут только лохи, не читающие договора и не думающие, что они будут делать, если захотят установить VM Ware или протянуть локальную сеть. (К тому же закон о защите потребителей никто не отменял и суды чаще всего выносят решения именно в пользу обиженных пользователей).

С другой стороны, бороться с "партизанами" все-таки надо, особенно, если внутрисетевой трафик дешевле грязи или вообще не тарифицируются. Тогда к держателю NAT'а могут подключаться не только соседи по лестничной площадки, но и все остальные клиенты провайдера, выбравшие соответствующий тарифный план (кстати говоря, составление тарифных планов — это настоящее искусство).

Как мы уже убедились, ни одна методика обнаружения NAT'ов, не без изъяна и никакую из них по отдельности применять нельзя. Но вот совокупность всех описанных методик дает неплохой результат, вполне пригодный для обнаружения нарушителей.

А если сюда подключить еще и психологию, то количество ложных срабатываний вообще упадет до нуля. Как наверняка известно опытным провайдерам, каждый (или, практически каждый) пользователь, дорвавшегося до "безлимитки" проходит через три стадии: сначала качает все-все-все, не выключая компьютер ни ночью, ни днем. Затем интерес начинает спадать. Пользователь становится более разборчивым и всякую дрянь уже не качает (потому как свободное дисковое пространство уменьшается со страшной скоростью, а DVD-R/CD-R между прочим денег стоят!). Наконец, пользователь окончательно успокаивается и потребление трафика значительно сокращается. В противном случае, подозрения, что пользователь не один, резко усиливаются (хотя встречаются и такие типы, которые не успокаиваются и через год). Так что некоторая неоднозначность все-таки остается…

>>> врезка NAT в RFC

Ниже представлен перечить RFC-стандартов, в которых описывается NAT и чтение которых облегчает его выявление:

- RFC1631 The IP Network Address Translator (NAT)
- RFC2766 Network Address Translation - Protocol Translation (NAT-PT)
- RFC3022 Traditional IP Network Address Translator (Traditional NAT)
- RFC3235 Network Address Translator (NAT)-Friendly Application Design Guidelines

>>> врезка полезные ссылки

- **Network Address Translation**:
  - основная информация о NAT'ах (энциклопедическая статья на английском): http://en.wikipedia.org/wiki/Network_address_translation;
- **Как обнаружить использование NAT?**
  - топик на форуме "SWAMP Board" (на русском языке): http://forum.swamp.ru/viewtopic.php?t=75236;
- **Скрыть прокси от провайдера**:
  - хакерские методики сокрытия NAT'ов от провайдера (на русском языке): http://www.xakep.ru/post/29448/default.asp?page=2;