Санкт-Петербург, ул. Седова, 80, телефон (812) 449-0770, факс (812) 449-0771, E-mail: info@bilim.com

ICMP-атаки на приложения UDP

Николай Малых

nmalykh@bilim.com

Чуть более года назад Фернандо Гонт обратил внимание на возможность разрыва соединений TCP или существенного снижения скорости передачи данных через такие соединения с помощью специально сформированных сообщений ICMP Destination Unreachable. Я уже писал об этом в небольшом обзоре [1], опубликованном несколько дней назад на этом сайте. Гонт также написал несколько простых программ для проверки стека TCP/IP и приложений на предмет наличия данной уязвимости. Многие производители отреагировали на работу Гонта и внесли соответствующие поправки в свою продукцию.

Однако, насколько мне известно, никто не предпринял серьезных попыток исследования возможности организации таких атак на приложения, работающие на основе транспорта UDP. На первый взгляд эта мысль может показаться бредовой – в UDP же не используется явных соединений. Однако, нужно учесть по крайней мере два обстоятельства, которые позволяют предположить возможность успешного проведения таких атак:

1) Работу приложений, использующих модель “запрос-отклик” также можно нарушить, при запросе клиент будет получать поток пакетов ICMP Destination Unreachable. Совсем не очевидно, что все клиенты смогут адекватно отреагировать на этот поток ложной информации.

2) Некоторые приложения (в частности, системы телефонной связи через Internet) организуют “соединения” на уровне инкапсулируемых в дейтаграммы UDP данных (например, “телефонное” соединение между парой абонентов). Для таких приложений поток информации о недоступности удаленной стороны может приводить к разрыву соединения на уровне инкапсулированного в UDP протокола.

3) Для некоторых приложений может играть важную роль целостность сессии (“соединения” между конечными точками на уровне 5). Здесь также есть простор для экспериментов.

К перечисленным категориям относится значительное число широко используемых приложений и желающие могут поэкспериментировать самостоятельно. Я для своих экспериментов взял первый попавшийся под руку объект – TFTP-клиента Windows XP. Приведенные ниже результаты показывают, что достаточно небольшим потоком сообщений ICMP Destination Unreachable можно существенно помешать работе TFTP, а в некоторых случаях совсем заблокировать возможность использования сервиса TFTP.

Немного о методологии атаки.

Протокол ICMP предназначен для проверки доступности хостов и передачи сообщений об ошибках, возникающих при передаче дейтаграмм между узлами IP-сетей.

В соответствии со стандартом Internet “Требования к хостам” [2] протокол UDP должен передавать информацию об ошибке, полученную в сообщении ICMP, на вышележащий уровень[1].

> Протокол UDP **должен** передавать на уровень приложений все сообщения ICMP об ошибках, полученные от уровня IP. Для реализации этого может использоваться вызов процедуры ERROR_REPORT (см. 4.2.4.1).
>
> **Обсуждение**:
>
> > Отметим, что сообщения ICMP об ошибках в результате передачи дейтаграмм UDP принимаются асинхронно. Приложения на базе протокола UDP, желающие получать сообщения ICMP об ошибках, сами отвечают за поддержку состояния, которое обеспечит демультиплексирование принятых сообщений (например, программа может сохранять операцию приема незавершенной). Приложения также отвечают за предотвращение конфликтов в результате задержки сообщений ICMP об ошибках по причине занятости портов.

Из приведенного фрагмента перевода RFC 1122 видно, что приложения UDP сами отвечают за демультплексирование и трактовку полученных хостом сообщений ICMP об ошибках. Таким образом, предполагаемая уязвимость не связана напрямую в недостатками в спецификации или реализациях протокола UDP, а может быть обусловлена обработкой сообщений ICMP в UDP-приложениях.

В соответствии со спецификацией протокола [3] сообщение ICMP должно содержать в себе полный заголовок IP из пакета, вызвавшего ошибку, и 64 бита (8 байтов) из этого пакета. Очевидно, что в эти 64 бита попадает лишь часть заголовка транспортного уровня. Для случая UDP эти 8 дополнительных байтов будут включать номера портов отправителя и получателя (по 2 байта каждый), а также поля размера и контрольной суммы дейтаграммы UDP (по 2 байта каждое). Требования к хостам [2] позволяют включать в сообщение ICMP дополнительную информацию из вызвавшего ошибку пакета, но не требуют такого включения. Требования к маршрутизаторам [4] говорят, что в сообщения об ошибках **следует** включать максимальное число байтов из вызвавшего ошибку пакета, которое не приведет к тому, что размер дейтаграммы, содержащей сообщение ICMP превысит 576 байтов. Таким образом хост или маршрутизатор, получающий сообщения ICMP об ошибке, может надеяться лишь на наличие в них первых 8 байтов заголовка транспортного уровня.

Протокол ICMP

Как уже было сказано, протокол ICMP используется, в частности, для передачи сообщений об ошибках, возникающих в сети. В соответствии со стандартом Internet “Требования к хостам” [5] относятся к критическим ошибкам[2]. Следовательно, любое из этих сообщений может привести к разрыву соединения TCP (Reset).

Спецификация протокола ICMP [3] включает несколько типов сообщений ICMP, с помощью которых атакующий может попытаться создать у приложения ложное представление о наличии проблем в сети. К таким сообщениям относятся, в частности, сообщения о недоступности адресата (Type 3, Destination Unreachable) с кодами 2 (Protocol Unreachable), 3 (Port Unreachable) и 4

[1]Цитируется по переводу, опубликованному на сайте http://www.protocols.ru.
[2]В стандарте используется термин hard error.

(Fragmentation needed but DF bit set[3]), а также сообщения Source Quench (тип 4, код 0), которые используются для управления потоком данных и предотвращения перегрузки в сети[4].

Сообщения ICMP используются также в механизме определения значений MTU для пути (Path MTU), описанном в RFC 1191 [6]. Для определения MTU применяются сообщения ICMP типа 3 (Destination Unreachable) с кодом 4 (Fragmentation needed but DF bit set). В силу такого использования сообщений указанного типа системы, реализующие механизмы Path MTU Discovery, не рассматривают ошибки данного типа как критические.

Собственно атака

На основании отмеченных выше особенностей можно предпринять попытку бомбардировки хостов, использующих приложения UDP потоком специально сформированных сообщений ICMP, содержащих сведения о якобы имеющих место проблемах при передаче данных через сеть. Очевидно, что такая атака будет иметь смысл лишь в тех случаях, когда принимаемые сообщения ICMP будут приняты хостом и связаны с каким-то приложением. Для этого нужно как минимум узнать или подобрать 4 параметра. Два из этих параметров являются адресами IP и, как правило, эти адреса известны атакующему, поскольку один из них задает объект атаки. Вторым адресом должен быть IP-адрес другой стороны, участвующей в обмене данными между приложениями UDP. Зачастую это адрес какого-либо сервера, который также несложно узнать. Два оставшиеся параметра представляют собой номера портов. Зачастую по крайней мере на одном из хостов используется стандартный (well-known) номер порта, а в некоторых случаях номера обеих сторон известны заранее. Если же известен номер только на одной стороне, второй придется подобрать. Это может несколько снизить эффективность атаки, но отнюдь не предотвращает ее. Следует отметить, что во многих случаях на станциях (не серверах) Microsoft Windows по умолчанию используется весьма небольшой диапазон портов от 1025 до 5000, что существенно упрощает задачу подбора.

Осталось совсем немного – создать поток сообщений ICMP, содержащих дезинформацию о недоступности некого хоста и отправленных якобы от имени этого хоста. Предположим, что на вашей станции в качестве сервера DNS указан адрес 192.168.1.1, а сама станция имеет адрес 172.16.0.33. Если злоумышленник, находящийся где-то в пределах домена, в котором маршрутизируется трафик обеих этих сетей (адреса взяты из приватных блоков) направит в адрес 172.16.0.33 поток сообщений ICMP о недоступности адресата 192.168.1.1 через порт 53 (domain) и укажет в заголовках IP значения 192.168.1.1:53 для отправителя, существует вероятность, что станция 172.16.0.33 в какой-то момент поверит в недоступность сервера DNS.

Приложения, которые представляются уязвимыми

Как было отмечено выше, уязвимыми для таких атак могут оказаться различные приложения, в которых информация о недоступности удаленной стороны принимается во внимание (инкапсулированные соединения, целостность сессий, своевременность отклика на запрос) и может так или иначе повлиять на работу приложения в целом. К таким приложениям на мой взгляд относятся:

- nfs
- esp
- sip
- dns
- dhcp
- tftp

Список этот наверняка неполон и желающие могут внести свою лепту в его расширение.

Очевидно, что наиболее привлекательными объектами атак являются DNS, DHCP, SIP. Я провел эксперименты с клиентом TFTP ОС Microsoft Windows XP[5]. Приложение TFTP представляется мне достаточно интересным, поскольку этот протокол достаточно широко используется для обновления программного кода в устройствах. Кроме того, этот протокол использует динамически выделяемые номера портов на обеих сторонах. Для организации соединения обычно используется стандартный порт 69, а передача данных осуществляется через порты с номерами более 1024 (непривилегированные порты). Оказалось, что для нарушения работы клиента достаточно весьма небольшого потока сообщений ICMP – около 10 кбит/с[6]. В качестве серверов использовались несколько вариантов свободно распространяемых программ для среды Windows и встроенный TFTP-сервер Linux. Существенной зависимости от типа используемого сервера отмечено при атаках на существующие соединения не было замечено. Однако заблокировать возможность возможность организации соединений для случая с сервером Windows мне не удалос. Для тестирования использовалась простая программа, написанная по образу и подобию icmp-reset Фернандо Гонта.

Для существующих сессий TFTP требуется подбор номеров портов обеих сторон, поэтому влияние паразитного трафика ICMP сравнительно невелико – оно приводит лишь к уменьшению скорости передачи данных в несколько раз. Однако, если использовать в пакетах ICMP фиксированный номер порта на стороне сервера (69) даже совсем незначительный поток пакетов ICMP (около 1 кбит/с) приводил к тому, что клиенту просто не удавалось соединиться с сервером. Повторю, что такого типа атака мне удалась только для случая работы к TFTP-сервером Linux.

Проводились эксперименты также с другими приложениями, но публиковать их результаты я не считаю возможным до того, как производители как-то прореагируют на отправленные им уведомления об уязвимостях.

Способы защиты

Не берусь предложить все возможные варианты защиты от такого рода атак, но хотелось бы отметить некоторые способы.

1. Более жесткий контроль параметров принимаемых пакетов ICMP на уровне IP – требует внесения изменений в стек TCP/IP. Для контроля можно использовать, например, значения IP ID из заголовков IP, включенных в сообщения ICMP. Для этих же целей может послужить контроль и сравнение содержимого полей TTL в заголовке дейтаграммы, содержащей сообщение ICMP, и в заголовке IP, включенном в это сообщение. Однако логика такого контроля может оказаться достаточно сложной. Хотя за обработку сообщений ICMP и несет ответственность само приложение UDP, перечисленные меры могут (по крайней мере, теоретически) повлиять на повышение уровня безопасности UDP-приложений. Кроме того, эти меры могут также повысить устойчивость соединений TCP к атакам с помощью ICMP-сообщений.

[3]Требуется фрагментация пакета, но в заголовке установлен флаг запрета фрагментации.

[4]Такие сообщения передаются в адрес отправителя, если получатель или промежуточный маршрутизатор не способен обрабатывать данные с той скоростью, которую выбрал отправитель. По сути дела эти сообщения являются запросом на снижение скорости передачи данных.

[5]Для тех, кто предпочитает объяснять проблемы использованием нелицензионных копий, отмечу, что данная копия честно куплена с ноутбуком Compaq/HP, поэтому такая отговорка не пройдет.

[6]Клиент и сервер располагались в коммутируемой ЛВС FastEthernet, поэтому такой поток данных практически не уменьшал полосу соединения и не влиял опосредованно на результаты.

2. Сохранение значений контрольных сумм UDP для последующего сравнения со значениями контрольных сумм, полученных в сообщении ICMP. Этот метод требует внесения изменений в код приложений и приведет к дополнительному расходу памяти, что может оказаться неприемлемым для серверов TFTP, встроенных в различные устройства.
3. Ingress-фильтрация входящего трафика по периметру сети в соответствии с RFC 2267. Этот способ не требует внесения изменений в программный код, но обеспечивает лишь частичную защиту от внешних атак. На атаки в локальной сети такая фильтрация не оказывает никакого воздействия.
4. Фильтрация принимаемых пакетов ICMP с учетом их типа и частоты доставки. Этот способ также не требует внесения изменений в программы и может быть реализован как на граничных шлюзах, так и на конечных станциях и серверах.

Заключение

Оценить уровень угрозы от атак этого типа достаточно сложно, тем более, что экспериментов пока явно недостаточно для какого-либо обобщения. Из общих соображений угроза представляется наиболее значительной для протоколов типа SIP, в которых поверх транспорта UDP организуются настоящие соединения. Однако с приложениями SIP мне пока поработать просто не удалось.

В заключение хотелось бы поблагодарить Александра Лопатина за ценные замечания и комментарии при обсуждении потенциальной уязвимости приложений UDP.

Библиография

1: Н. Малых, Атаки на соединения TCP с помощью ICMP-пакетов, http://www.protocols.ru/Security/ICMPagainstTCP.shtml
2: Braden, R., Requirements for Internet Hosts - Communication Layers, STD 3, RFC 1122, октябрь 1989 Перевод этого стандарта имеется на сайте http://www.protocols.ru
3: Postel, J., Internet Control Message Protocol, STD 5, RFC 792, сентябрь 1981 Перевод этого документа имеется на сайте http://www.protocols.ru
4: Baker, F., Requirements for IP Version 4 Routers, RFC 1812, июнь 1995 Перевод этого документа имеется на сайте http://www.protocols.ru
5: Braden, R., Requirements for Internet Hosts - Communication Layers, STD 3, RFC 1122, октябрь 1989 Перевод этого стандарта имеется на сайте http://www.protocols.ru
6: Mogul, J., S. Deering, Path MTU discovery, RFC 1191, ноябрь 1990 Перевод этого документа имеется на сайте http://www.protocols.ru