25 февраля 1985 г.

A Weakness in the 4.2BSD Unix TCP/IP Software

Недостатки реализации протокола TCP в ОС 4.2BSD Unix

Robert T. Morris
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

Операционная система Unix версии 4.2 Berkeley Software Distribution (далее 4.2BSD) поддерживает широкий набор программ, работа которых основана на использовании семейства протоколов TCP/IP. В частности, каждая система 4.2BSD "доверяет" некоторому множеству других систем, позволяя пользователям этих систем выполнять программы через сеть TCP/IP без проверки пароля. В этой статье описано, как TCP/IP и реализация этого стека протоколов в 4.2BSD позволяют пользователям непроверенных[1] и, возможно, весьма удаленных хостов замаскироваться под доверенных пользователей. Лаборатории Bell используют постоянно расширяющуюся сеть TCP/IP, связывающую машины с различными требованиями безопасности, поэтому следует принять меры по снижению влияния описанной здесь уязвимости.

Стандарт протокола TCP/IP[2] был разработан в 1979 году для реализации в "internet" – группе сетей, сильно различающихся по надежности и скоростным характеристикам и соединенных между собой через компьютеры, действующие в качестве шлюзов. Одной из наиболее популярных реализаций стека TCP/IP в среде Unix стала система 4.2BSD, используемая, в частности, Лабораториями Bell и Министерством обороны США. Программы 4.2BSD Unix TCP/IP весьма гибки и удобны, но слишком сильно доверяют другим компьютерам сети, не обеспечивая высокого уровня безопасности. Описанная здесь технология атаки не требует изменения ОС и не зависит от используемого в сети оборудования.

Стек TCP/IP концептуально делится на два уровня TCP (Transmission Control Protocol) и IP (Internet Protocol). Уровень IP обеспечивает передачу пакетов данных (дейтаграмм) между хостами через соединяющие их сети и шлюзы. Уровень TCP поддерживает множество "портов" на каждом хосте IP, обеспечивающих создание надежных и управляемых соединений (виртуальных каналов) между хостами. Соединения TCP организуются на основе сервиса передачи дейтаграмм IP. Каждый пакет TCP или IP содержит заголовок с управляющей информацией, за которым следуют передаваемые в пакет данные. В случае TCP данные предоставляются пользователем, а данными IP служат пакеты TCP. Важными элементами заголовка TCP являются номера портов получателя и отправителя, порядковый номер, номер подтверждения и некоторые флаги. Номера портов служат для идентификации "виртуальных устройств". Порядковые номера и номера подтверждений обеспечивают корректность порядка доставки пакетов, а флаги влияют на состояние "виртуальных устройств". Заголовок IP содержит адреса отправителя и получателя (32-битовые уникальные идентификаторы хоста и сети), а также номер протокола (например, TCP), котором протокол IP должен передать данные.

4.2BSD поддерживает функции удаленного сервера, который прослушивает запросы на организацию соединений TCP через порт 514[3]. При получении такого запроса сервер проверяет наличие хоста, указанного адресом отправителя в заголовке IP, в списке пользующихся доверием компьютеров. Если хост найден в этом списке, идентификатор пользователя и переданная им команда принимаются сервером к исполнению. Недостаток такой схемы состоит в том, что хост-отправитель сам указывает адрес в заголовке IP, а 4.2BSD и протокол TCP/IP не могут проверить корректность этого адреса.

Идеальным случаем была бы передача обманных пакетов непосредственно с использованием стека TCP/IP. 4.2BSD не обеспечивает такой возможности, поэтому требуются дополнительные программы для передачи обманных пакетов от систем 4.2BSD. Однако 4.2BSD предоставляет привилегированным пользователям возможность передачи пакетов IP, поэтому с минимальными усилиями можно организовать передачу пакетов с корректным номером протокола (6) и подставным адресом отправителя в заголовке IP. Для этого достаточно создать в 4.2BSD сокет типа SOCK_RAW, внести в структуру данных ядра изменение, позволяющее указывать для сокета SOCK_RAW номер протокола 6 (TCP) и подменить адрес отправителя. Это требует наличия у пользователя привилегий, однако в большой сети всегда можно найти по крайней мере один хост с недостаточным уровнем защиты для получения таких привилегий.

Имея соответствующий доступ к IP, пользовательский процесс сможет создавать и поддерживать соединения TCP без обращений к модулю протокола TCP в ядре Unix. Каждый заголовок TCP содержит контрольную сумму для обнаружения ошибок при передаче. Эта контрольная сумма учитывает не только заголовок и данные TCP, но и часть заголовка IP. Следовательно, пользовательская программа должна предсказать содержимое заголовка IP, который ядро использовало бы для инкапсуляции пакета TCP. После этого пользовательский процесс может передавать отдельные пакеты TCP.

Соединение TCP может находиться в состояниях LISTEN, SYN_SENT, SYN_RCVD и ESTABLISHED. Для каждого соединения TCP также поддерживается порядковый номер, как часть данных о состоянии этого соединения. На состояние соединения оказывают влияние флаги SYN, ACK, RST (синхронизация, подтверждение, сброс), а также порядковые номера подтверждений. Одна сторона[4] инициирует соединение передачей пакета с флагом SYN и переходом в состояние SYN_SENT; другая сторона[5] начинает соединение в состоянии LISTEN. В приведенной ниже диаграмме состояний каждое сообщение представлено набором флагов, порядковым номером и номером подтверждения. Каждая комбинация "состояние-событие" обычно ведет к передаче пакета и смене состояния (возможно с возникновением ошибки); ячейки таблицы показывают передаваемые пакеты и состояния соединения. M означает порядковый номер принятого пакета; N – порядковый номер, сохраненный как часть данных о состоянии порта TCP.

[1] В оригинале - untrusted – не пользующиеся доверием. *Прим. перев.*
[2] Протокол IP описан в RFC 791, а TCP - в RFC 793. *Прим. перев.*
[3] Служба shell в современной терминологии. *Прим. перев.*
[4] Клиент. *Прим. перев.*
[5] Сервер. *Прим. перев.*

	SYN,X,Y	ACK,X,Y,данные
LISTEN	SYN,N++,M+1 SYN_RCVD	ошибка
SYN_SENT	ACK,N,M+1 ESTABLISHED	ошибка
SYN_RCVD	RST,N,M ошибка	ESTABLISHED
ESTABLISHED	RST,N,M ошибка	ACK,N,M+размер данных ESTABLISHED (передача данных пользователю)

Данные (ACK,N,M,данные) передаются, когда обе стороны находятся в состоянии ESTABLISHED и после передачи данных значение N увеличивается на размер этих данных. Соединение также может находиться в других состояниях и использовать дополнительные флаги, не имеющие отношения к теме статьи.

4.2BSD поддерживает глобальную переменную для последовательных номеров, значение которой увеличивается на 128 каждую секунду и 64 при старте каждого соединения. Когда хост передает пакет SYN с обманным адресом, получатель такого пакета будет передавать отклик по указанному в пакете подставному адресу. Передающий пакеты с подставными адресами хост должен определить или предсказать порядковый номер неполученного им отклика, чтобы подтвердить его получение и перевести порт TCP атакуемого хоста в состояние ESTABLISHED. Предсказание порядкового номера для хоста 4.2BSD является простой задачей – достаточно организовать с атакуемым хостом реальное соединение, посмотреть полученный для него порядковый номер и увеличить его значение на 64. После этого обманная программа подтверждает этот порядковый номер, завершая тем самым процесс организации соединения, и может передавать данные (не получая, правда, откликов на них).

Дополнительные сложности связаны с тем, что SYN-пакет от атакуемого хоста не исчезает бесследно. Получив такой пакет, хост будет передавать в ответ пакет с флагом RST при получении которого атакуемый хост разорвет обманное соединение. Например, хост A шлет хосту B обманные пакеты с указанным в них адресом хоста C. Хост B будет передавать пакет SYN хосту C, в ответ на это C передаст пакет RST по адресу хоста B. В результате хост B сбросит соединение, которое хост A создал обманным путем. Однако порты хоста C, находящиеся в состоянии прослушивания входящих соединений могут не передавать пакета RST при получении неожиданного пакета SYN. Дело в том, что размер очереди ожидающих организации соединений ограничен и при получении пакета SYN, выходящего за пределы очереди такой пакет просто будет отброшен без генерации пакета RST в предположении повтора пакета SYN[1]. Таким образом, если обманный пакет отправлен с указанием ждущего соединений порта на подставном хосте, разрыва обманного соединения может не произойти.

Предположим, что атакующий хост A передает хосту B пакеты с адресом хоста C в поле отправителя. Пакеты адресуются в порт 514 хоста B и отправлены якобы из порта 21[2] (который обычно находится в состоянии ожидания) хоста C. Цепочка событий на хосте A может иметь вид:

1) Забросать порт 21 хоста C запросами на организацию соединения для переполнения очереди.
2) Создать реальное соединение с портом хоста B и запомнить порядковый номер, полученный от B.
3) Создать raw-сокет IP, сменить для него номер протокола на TCP (6) и задать в качестве отправителя адрес хоста C.
4) Передать обманный пакет SYN “из порта 21 хоста C” в порт 514 хоста B и передать пакет SYN в порт 21 хоста C для переполнения очереди.
5) Передать пакет ACK хосту B с номером подтверждения, соответствующим сохраненному порядковому номеру, увеличенному на 64.
6) Передать данные хосту B, не забывая увеличивать порядковый номер с учетом переданной информации. Порт 514 ожидает пустую строку, за которой следует строка с именем пользователя и строка команды.
7) Если хост B доверяет хосту C, полученная от хоста B команда будет выполнена.

В целях обеспечения ясности некоторые детали процесса были опущены.

При добавлении опущенных деталей описанная схема будет работать достаточно надежно. Она позволяет машине, подключенной к сети TCP/IP выполнить команду на любой подключенной к этой сети системе 4.2BSD, которая “доверяет” другим системам. Таким образом можно организовать различные атаки. Порядковые номера, которые должен предсказать атакующий, могут быть более случайными и 32-битовые номера создают большой простор для предсказаний. Однако атакующий может организовать достаточно большое число пробных соединений для определения алгоритма генерации случайных чисел; уровень случайности начальных порядковых номеров для соединения определяет сложность предсказания этих номеров. Еще лучше будет потребовать от всех сетей IP использовать только корректные адреса, но это зависит от используемого в сети оборудования и в некоторых случаях может не работать. Реальным выходом из положения может послужить ограничение доверительных отношений пределами локальной сети и изменение сетевых шлюзов таким образом, чтобы они отвергали пакеты, пришедшие снаружи и имеющие в поле отправителя адрес из локальной сети.

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1] Отметим, что пакеты SYN инициатора соединения и отвечающего сервера идентичны.
[2] Порт ftp. *Прим. перев.*