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

SYN cookies[1]

D. J. Bernstein

Работа почтового сервиса Panix, принадлежащего ISP в Нью-Йорке, была блокирована атакой SYN flood, начавшейся 6 сентября 1996. Неделей позже история повторилась с RISKS Digest, Wall Street Journal, Washington Post и многими другими газетами.

Атаки SYN flood[2] были предсказаны экспертами до их реального обнаружения. Многие считают проблему таких атак неразрешимой. Например, Garfinkel и Spafford в книге "Practical UNIX and Internet Security" (стр. 778) пишут:

> *На стороне адресата[3] возникает большое число полуоткрытых соединений, поглощающих ограниченные системные ресурсы. Обычно в таких соединениях указываются подставные адреса отправителей, которые указывают на несуществующие или недоступные хосты. Таким образом, атакующий не имеет возможности отследить источник атаки. В такой ситуации вы сможете сделать очень мало для предотвращения атаки ... любой ограниченный ресурс когда-либо будет исчерпан.*

Увеличение очередей SYN и случайное упреждающее отбрасывание соединений (drop) усложняет жизнь инициаторам атак SYN flood, но не решает проблему полностью.

Функции SYN cookie используют криптографические алгоритмы для решения проблемы. Я писал как это можно сделать[4] 16 сентября 1996; Вместе с Эриком Шенком (Eric Schenk) в течение нескольких следующих недель были выработаны детальные предложения[5] по решению проблемы. Джеф Вайсберг (Jeff Weisberg) реализовал это в программе для SunOS в октябре 1996, а Эрик Шенк создал в феврале 1997 программную реализацию для 1997.

Функции SYN cookie в настоящее время являются стандартным решением для операционных систем Linux и FreeBSD. К сожалению, в ОС Linux эти функции по умолчанию отключены. Для того, чтобы включить их, достаточно добавить команду

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

в сценарий загрузки[6].

Что такое SYN cookie?

SYN cookie – это одно из решений задачи выбора начальных порядковых номеров TCP серверами TCP. Разница между начальными порядковыми порядковыми номерами на серверах и клиентах заключается в:

- 5 старших битов: значение t mod 32, где t – 32-разрядный счетчик временных интервалов, значение которого увеличивается на 1 каждые 64 секунды;
- следующие 3 бита: кодированное значение MSS[7], выбранное сервером в ответ на MSS клиента;
- младшие 24 бита: выбранная сервером на основе IP-адресов и номеров портов отправителя и получателя, а также величины t значение секретной функции.

Такой алгоритм выбора начального порядкового номера соответствует основным требованиям протокола TCP, в соответствии с которыми номера должны увеличиваться достаточно медленно и начальные порядковые номера для серверов растут несколько быстрее, нежели порядковые номера для клиентов.

Серверы, использующие функции SYN cookie, не отвергают соединения при заполнении очереди SYN. Взамен они передают инициатору соединения пакет SYN+ACK, в точности соответствующий пакету, который был бы передан при большем размере очереди SYN (исключения: сервер должен отвергать (reject) опции TCP такие, как большое окно, и должен использовать 1/8 значения MSS, которое он может кодировать). При получении пакета ACK, сервер убеждается в работе секретной функции для последнего (recent) значения t и перестраивает запись очереди SYN в соответствии со значением MSS.

Атаки SYN flood представляют собой просто серии пакетов SYN с подставными адресами IP. Эти адреса выбираются случайным образом и не содержат никакой информации об атакующей стороне. Атаки SYN flood заполняют SYN-очереди серверов. Обычно это приводит к тому, что сервер отвергает входящие соединения. Использующий функции SYN cookie сервер будет продолжать нормально работать во время таких атак, Максимальное воздействие атаки SYN flood на такой сервер будет состоять в блокировке использования больших окон.

Атаки вслепую

Если атакующий сможет угадать порядковый номер, переданный какому-либо из хостов, он сможет организовать обманное соединение от имени этого хоста.

Атакующие могут попытаться предпринять криптоанализ выбранной сервером секретной функции, просматривая последовательность корректных cookie и пытаясь предсказать следующее значение cookie. При эффективной реализации функций шанс корректно предсказать порядковый номер незначительно превышает шансы на угадывание случайного числа при равномерном распределении. Для обеспечения безопасности были созданы средства аутентификации секретных ключей (Secret-key message authenticator). Достаточную скорость и безопасность обеспечивает функция, кодирующая входные данные в 16 байтов, обрабатывающая их с помощью алгоритма Rijndael и представляющая в качестве результата первые 24 бита.

Независимо от используемой функции атакующий для достижения успеха в попытках организации подставного соединения должен принять миллионы случайных пакетов ACK. Серверы могут осложнить организацию таких атак двумя путями:

---

[1] Оригинал этого документа можно найти на сайте http://cr.yp.to/syncookies.html
[2] Интенсивный поток пакетов TCP с установленным битом SYN (попытка организации соединения). *Прим. перев.*
[3] Объекта атаки.
[4] http://cr.yp.to/syncookies/idea
[5] http://cr.yp.to/syncookies/archive
[6] Требуется также включить поддержку этой функции я ядре Linux при его компиляции.
[7] Максимальный размер сегмента TCP.

- Хранение информации о времени последнего переполнения очереди SYN (для каждой очереди отдельно, а не в глобальной переменной). Отсутствующие записи очереди SYN не создаются заново, если не было недавнего переполнения очередей. Это позволяет предотвратить прохождение подставных пакетов ACK через брандмауэры с блокировкой SYN.
- Добавление другого числа в cookie: выбранная сервером 32-битовая секретная функция от адресов клиента и сервера (без учета текущего времени). Это потребует от атакующего подбора 32 битов взамен 24.

Новый протокол с поддержкой 128-битовых порядковых номеров сделает атаки вслепую практически невозможными.

Кто создал SYN cookie?

Как мне известно, Фил Кэрн (Phil Karn) был первым, кто разработал протокол Internet, использовавший cookie для защиты от DoS-атак вслепую. Однако идея более стара.

По моему мнению, я был первым, кто указал, что серверы TCP могут использовать cookie без каких-либо изменений протокола TCP. (Perry Metzger впоследствии заявлял, что он сделал это раньше. Однако Metzger не ответил мне, когда я попросил его прислать[8] мне копии связанных с этим сообщений. Архивы NANOG содержат информацию, что Metzger заявлял 9 сентября 1996, что функции cookie требуют создания нового протокола “TCP++”, а 17 сентября 1996, - что ISP должны фильтровать исходящие от них пакеты).

Мое первое предложение не соответствует требованиям TCP по использованию возрастающих порядковых номеров. У Эрика Шенка (Eric Schenk) возникла идея по добавлению “чего-либо” к порядковым номерам на стороне клиентов.

Я предложил зависящие от времени функции SYN cookie. Временная зависимость не дает атакующим возможности (1) собирать корреткные cookie на компьютере общего пользования и (2) впоследствии повторно использовать эти cookie для атаки с другого компьютера.

Страшилки о SYN cookie

Некоторые люди (в частности Alexey Kuznetsov, Wichert Akkerman и Perry Metzger) распространяют дезинформацию и функциях SYN cookie. Ниже приведено несколько примеров таких некорректных заявлений[9]:

- *Функции SYN cookie “являются серьезным нарушением протокола TCP.”* В реальности функции SYN cookie полностью соответствуют требованиям протокола TCP. Каждый пакет, передаваемый сервером с поддержкой SYN cookie представляет собой совокупность данных, которая может быть передана и сервером, не поддерживающим SYN cookie.
- *Фунции SYN cookie “не позволяют использовать расширения TCP" такие, как большой размер окна.* В реальности SYN cookie не оказывают влияния на расширения TCP. Соединения, сохраненные с помощью SYN cookie, не могут использовать окна большого размера, но то же самое произойдет и без SYN cookie, поскольку соединение будет просто уничтожено.
- *Функции SYN cookies могут вызывать “масштабное “зависание” соединений.”* В реальности соединения время от времени “зависают”, независимо от использования SYN cookie, при перегрузке компьютеров или сетей. Приложения в таких случаях просто отбрасывают “умершие” соединения.
- *Функции SYN cookie вызывают “серьезное снижение производительности служб.”* Реально функции SYN cookie повышают эффективность сервиса. Во время вычислений функции отнимают незначительные ресурсы CPU, но это процессорное время так или иначе было бы потрачено на создание труднопредсказуемых порядковых номеров (см. RFC 1948[10]).
- *SYN cookie может вызывать “магический сброс (magic reset).”* В реальности функции SYN cookie никогда не приводят к сбросу.

Эти люди распространяют свои заблуждения другим людям, в частности, мне. Я не знаю, что является причиной – умысел или заблуждение, - но в любом случая буду рад установлению истины.

Я предлагал Кузнецову отказаться от его заявления или отстаивать свое мнение в дискуссии. Он отказался от этого предложения. Я уверен, что сейчас он понимает ошибочность своего заявления и любые попытки отстаивать эту точку зрения обречены на неудачу. Очень жаль, что он не захотел участвовать в установлении истины.

Был приглашен к обсуждению и Akkerman, но он просто не ответил.

[8] http://cr.yp.to/syncookies/metzger
[9] Некоторые из этих заявлений приводятся в файле ip-sysctl.txt из дистрибутива ядра Linux.
[10] Документ RFC 1948 можно загрузить с сайта http://rfc-editor/rfc/rfc1948.txt, а его перевод имеется на сайте http://www.protocols.ru.