Эссе по курсу «Защита информации»
4й курс МФТИ, гр.013 Пашко Е.Б.

Безопасность TCP-IP

В данной работе очень кратко рассматривается безопасность стека протоколов TCP/IP. Описываются возможные типы атак, и некоторые способы борьбы с ними. Предполагается что читатель уже имеет некоторое представление о работе TCP/IP.

1. Вступление.

Около 20и лет назад, по инициативе Министерства обороны США была разработана сеть передачи данных, по принципу коммутации пакетов, удовлетворявшая требованиям того времени по «живучести» и универсальности - ARPAnet. Буквально за десятилетие небольшая экспериментальная сеть выросла в глобальную среду передачи данных – Internet. Основой как и первой сети ARPAnet, так и современной Internet является стек протоколов TCP/IP. Поскольку изначально он разрабатывался для маленькой сети, с взаимно - доверяющими друг другу пользователями, при столь резком увеличении масштаба сети вылезли наружу многие недоработки в данном протоколе, и особенно касающиеся безопасности. В данной работе будут рассмотрены проблемы безопасности TCP/IP версии 4. Многие из рассматриваемых проблем были решены, или приобрели иной вид в дальнейшем развитии стека – в IPv6, но, к сожалению, ввиду многих факторов данная версия протокола пока не прижилась в Internet, равно как и в локальных сетях, построенных на основе TCP/IP, поэтому ее рассмотрение не является актуальным на настоящий момент.

2. Классификация по уровню в стеке.

По аналогии с моделью OSI, в стеке TCP/IP можно выделить 4 уровня(мы не рассматриваем уровень приложений):

- TCP
- IP
- Канальный (среда передачи, например Ethernet, Token Ring, PPP итп.)
- Физический (провод/оптика/радиоканал).

Начать классификацию можно с разделения видов атак по этим уровням. Под уровнем атаки будем понимать самый высокий из уровней, который подвергся атаке непосредственным образом.
Рассмотрим их, снизу вверх.

2.1 Физический

История атак на физический уровень начинается с момента появления первой среды передачи данных, еще до нашей эры не брезговали тем, чтобы побить гонца (оборвать канал связи), или, его же, прилично напоив, прочитать секретный свиток (подслушивание). Суть атаки тривиальна, и более на ней останавливаться не будем.

2.2 Канальный

Канальный уровень, уязвим для прослушивания уже более цивилизованным путем.

Рассмотрим Ethernet. Ввиду идеологии данной технологии все пользователи одного сегмента сети (участок до первого коммутатора(switch)) составляют т.н. коллизионный домен – т.е. «видят» Ethernet-кадры всех своих соседей. Сетевой адаптер лишь по своей воле уже принятый кадр соседа удаляет из своей памяти, после того как он проверит соответствие своего адреса канального уровня(MAC) и адреса получателя, и обнаружит несовпадение. Любой современный адаптер Ethernet можно перевести в режим, когда он будет успешно принимать все подряд кадры, и вынимать из них интересную информацию.

Существует довольно большое количество программных продуктов, в которых уже реализован данный метод подслушивания они носят общее название- *сниферы(Sniffers).*

Как способ решения проблемы – проектировать сеть так, чтобы она состояла из множества сегментов, по 1му компьютеру в каждом.

В сетях Token Ring ситуация похожая, ввиду того что там данные циркулируют по всей сети, их тоже можно прослушать.

При применении PPP такой способ не возможен, т.к. в обмене участвуют лишь 2 точки- компьютер и сервер.

3. Активные атаки на уровне TCP и IP

Все вышеперечисленные атаки можно отнести в «пассивным», т.к. у злоумышленника не было никакого активного взаимодействия с каналом передачи или хотя бы с одной из передающих стороной. Теперь же рассмотрим «активные атаки», на протокол IP и все вышележащие протоколы данного стека.

- Проблемы, связанные с реализацией TCP (TCP State machine Problems)
- Затопление SYN-пакетами (SYN Flooding)
- Предсказание TCP sequence number (IP Spoofing)
- IP Hijacking
- Атаки на протоколы маршрутизации
- Затопление ICMP-пакетами
- Атаки на DNS

3.1. Проблемы, связанные с реализацией TCP (TCP State machine Problems)

Работу протокола TCP можно представить в виде некоторой «машины состояний» (См. RFC 793, стр. 23). Как и любая машина состояний, машина TCP имеет состояния и условия переходов между ними.

Уязвимость заключается в том, что событие, которое соответствует условию выхода из какого либо состояния может никогда не наступить, и машина TCP «повиснет». Вообще говоря существует механизм борьбы с такими зависаниями, и возник он не из соображения безопасности. Любой канал передачи данных подвержен помехам, или может просто оборваться. Таким образом посылка, активирующая условие выхода из какого-либо состояния, может не дойти. Стандартный способ борьбы с этим явлением – введения механизма таймаутов. После перехода в некоторое состояние активируется таймер, и если по истечению некоторого времени не придет дальнейшей команды от удаленной стороны, TCP переходит в исходное состояние.

Несмотря на гибкий механизм таймаутов, может случиться совершенно тривиальная вещь – у какого либо состояния не окажется таймера, и TCP, при умелых манипуляциях злоумышленника, повиснет в этом состоянии на веки (т.е. до перезагрузки). Правда в современных реализациях TCP такое практически не возможно- за долгую историю многих операционных систем, реализующих TCP/IP эта проблема была устранена. Зато все еще актуальной остается следующая: таймаут на некоторое состояние может иметь очень большое значение, и таким образом зависание TCP будет не вечным, а «сего лишь» очень долгим. Но это совсем не мешает злоумышленнику по истечению таймаута повторить атаку.

Как пример можно рассмотреть состояние CLOSE_WAIT (ожидание подтверждения о получении клиентом FIN+ACK - посылки для закрытия сессии). Там присутствует *Keep-alive* таймер, отвечающий за поддержку TCP сессии. Если злоумышленник корректно не закроет сессию, другая сторона будет ждать его посылок по умолчанию 2 часа, заморозив при этом весь TCP.

3.2 Затопление SYN-пакетами (SYN Flooding)

Установление соединения TCP происходит следующим образом:

1) Клиент посылает серверу запрос на установление соединения (SYN пакет, содержащий sequence number со стороны клиента C-SYN)
2) Сервер отвечает клиенту подтверждением, и своим запросом (SYN+ACK пакет, содержащий в поле ACK C-SYN+1, и sequence number со стороны сервера S-SYN)
3) Клиент подтверждает получение посылки сервером (ACK пакет, содержащий S-SYN +1).

Для поддержки медленных каналов связи время ожидания ответа клиента сделано довольно большим(по умолчанию 75 секунд).

Затопление SYN-пакетами основано на том, что злоумышленник может непрерывно посылать серверу SYN пакеты, не отвечая на его запросы. Для каждого принятого SYN запроса будет открываться сессия. Ввиду того, что количество открытых сессий всегда ограничено, злоумышленник (или группа злоумышленников) может открыть сессий ровно столько, сколько поддерживает сервер, и каждые 75 секунд обновлять свои соединения.

Таким образом можно вывести из строя какой ни будь TCP сервер, он просто не сможет принять запросы нормальных пользователей т.к. все свои ресурсы направит на поддержание «ложных» сессий.

Для борьбы с SYN Flooding можно предложить уменьшение таймаута удержания сессии или принудительное «разрежение» переполненной очереди сервера.

3.3 Предсказание TCP sequence number (IP Spoofing)

Еще раз посмотрим на схему установления соединения TCP, в пункте 3.2.

Предположим, что злоумышленник может предсказать, какой sequence number (S-SYN) будет выслан сервером, тогда он сможет не получив пакета сервера, посылать ему свои пакеты от имени другого IP адреса, которому клиент «доверяет». Это возможно сделать на основе знаний о конкретной реализации TCP/IP. Например, в 4.3BSD значение sequence number, которое будет использовано при установке следующего значения, каждую секунду увеличивается на 125000. Таким образом, послав один пакет серверу, злоумышленник получит ответ и сможет (возможно, с нескольких попыткок и с поправкой на скорость соединения) предсказать sequence number для следующего соединения.

Если реализация TCP/IP использует специальный алгоритм для определения sequence number, то он может быть выяснен с помощью посылки нескольких десятков пакетов серверу и анализа его ответов.

Итак, предположим, что система A доверяет системе B, так, что пользователь системы B может получить доступ к A, не вводя пароля. Предположим, что злоумышленник расположен на системе C. Система A выступает в роли сервера, системы B и C - в роли клиентов.

Первая задача злоумышленника - ввести систему B в состояние, когда она не сможет отвечать на сетевые запросы. Это может быть сделано несколькими способами, в простейшем случае нужно просто дождаться перезагрузки системы B. Нескольких минут, в течении которых она будет неработоспособна, должно хватить. Другой вариант - использовать способ, описанный в разделе 3.2.

После этого злоумышленник может попробовать притвориться системой B, для того, что бы получить доступ к системе A (хотя бы кратковременный).

- злоумышленник высылает несколько IP-пакетов, инициирующих соединение, системе A, для выяснения текущего состояния sequence number сервера.
- злоумышленник высылает IP-пакет, в котором в качестве обратного адреса указан уже адрес системы B.
- система A отвечает пакетом с sequence number, который направляется системе B. Однако система B никогда не получит его (она выведена из строя), как, впрочем, и

злоумышленник. Но он на основе предыдущего анализа догадывается, какой sequence number был выслан системе B.

- злоумышленник подтверждает "получение" пакета от A, выслав от имени B пакет с предполагаемым S-ACK

Следует заметить, что если системы располагаются в одном сегменте сети, злоумышленнику для выяснения sequence number достаточно перехватить пакет, посланный системой A, как это было описано в пункте 2.2.

После этого, если злоумышленнику повезло и sequence number сервера был угадан верно, соединение считается установленным.

3.4 IP Hijacking

Как известно, при передаче данных постоянно используются sequence number и acknowledge number (оба поля находятся в IP-заголовке). Исходя из их значения, сервер и клиент проверяют корректность передачи пакетов.

Существует возможность ввести соединение в "десинхронизированное состояние", когда присылаемые сервером sequence number и acknowledge number не будут совпадать с ожидаемым значениеми клиента, и наоборот. В данном случае злоумышленник, "прослушивая" линию, может взять на себя функции посредника, генерируя корректные пакеты для клиента и сервера и перехватывая их ответы. Некорректные пакеты, с «неправильным» sequence number или acknowledge number, даже если и дойдут до клиента или сервера, по умолчанию будут проигнорированы (точнее говоря будет отправлен ack-пакет, см. ниже).

Метод позволяет полностью обойти такие системы защиты, как, например, одноразовые пароли, поскольку злоумышленник начинает работу уже после того, как произойдет авторизация пользователя. Также очень удобно производить атаки «человек посередине» на шифры с открытым ключом.

Существует 2 основных метода рассинхронизации сессии – ранняя десинхронизация и десинхронизация «нулевыми данными».

В первом случае соединение десинхронизируется на стадии его установки. Злоумышленник принудительно сбрасывает запрос сервера на установку сессии, посылая затем запрос на установку уже своей сессии(от имени клиента), параллельно генерируя необходимые пакеты для клиента. Таким образом получается установленная рассинхронизированая сессия.

При десинхронизация «нулевыми данными» злоумышленник прослушивает сессию и в какой-то момент посылает серверу пакет с "нулевыми" данными, т.е. такими, которые фактически будут проигнорированы на уровне прикладной программы и не видны клиенту (например, для telnet это может быть данные типа IAC NOP IAC NOP IAC NOP...). Аналогичный пакет посылается клиенту. Очевидно, что после этого сессия переходит в десинхронизированное состояние.

Данный метод имеет довольно существенный недостаток, позволяющий его обнаружить.

Он заключается в том, что любой пакет, высланный в момент, когда сессия находится в десинхронизированном состоянии вызывает так называемый ACK-бурю. Например, пакет выслан сервером, и для клиента он является неприемлимым, поэтому тот отвечает ACK-пакетом. В ответ на этот неприемлимый уже для сервера пакет клиент вновь получает ответ... И так до бесконечности. Но современные сети строятся по технологиям, когда допускается потеря отдельных пакетов. Поскольку ACK-пакеты не несут данных, повторных передачи не происходит и "буря стихает".

3.5 Атаки на протоколы маршрутизации

Хотя протоколы маршрутизации и не являются на прямую компонентом TCP/IP, они обеспечивают работу практически всех более менее крупных TCP/IP сетей. Но так же как и TCP/IP в большинстве из них не предусмотрена аутентификация. Посылая ложные RIP пакеты злоумышленник может настроить маршрутизацию так, чтобы весь поток информации шел через него, со всеми вытекающими из этого последствиями.

3.6 Затопление ICMP-пакетами

Традиционный англоязычный термин -- "ping flood". Появился он потому, что программа "ping", предназначенная для оценки качества линии, имеет ключ для "агрессивного" тестирования. В этом режиме запросы посылаются с максимально возможной скоростью и программа позволяет оценить, как работает сеть при максимальной нагрузке.

Данная атака требует от злоумышленника доступа к быстрым каналам в Интернет.

При стандартном режиме работы пакеты выслаются через некоторые промежутки времени, практически не нагружая сеть. Но в "агрессивном" режиме поток ICMP echo request/reply-пакетов может вызвать перегрузку небольшой линии, лишив ее способности передавать полезную информацию.

Естественно злоумышленник может также подделывать обратный адрес подобных пакетов, затрудняя его обнаружение.

3.7 Атаки на DNS

Подобные атаки направлены на сервисы, которые запрашивают услуги по «имени», вместо IP адреса (на пример WWW). В таком случае ОС делает запрос DNS серверу на преобразование имени в IP. В DNS так же как и в большинстве других протоколов не предусмотрена аутентификация, таким образом данный запрос можно перехватить, и отправить свой ответ.

Используемая литература:

1) Вадим Колонтсов «Безопасность TCP/IP» 2000г, http://www.citforum.ru/internet/securities/tcpip.shtml
2) CHRIS CHAMBERS, JUSTIN DOLSKE, and JAYARAMAN IYER «TCP/IP Security» http://www.linuxsecurity.com/resource_files/documentation/tcpip-security.html
3) S.M. Bellovin «Security Problems in the TCP/IP Protocol Suite» Computer Communication Review, Vol. 19, No. 2, pp. 32-48, April 1989 http://www.ja.net/CERT/Bellovin/TCP-IP_Security_Problems.html
4) И. Д. Медведовский, П. В. Семьянов, В. В. Платонов «АТАКА ЧЕРЕЗ INTERNET», 1997 г.