ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ РФ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИИ, УПРАВЛЕНИЯ И ДИЗАЙНА
УЛЬЯНОВСКОГО ГОСУДАРСТВЕННОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА
Кафедра «Математики и информационных технологий»

Курсовая работа
по курсу: “Методы и средства защиты информации”

Тема: Сканер портов Nmap

Руководитель:

_____
(подпись) Ф.И.О.
_____(дата)

Студент:

Потеренко А.Г.

Группа ВТ-41

_____(дата)

_____(220400)_____
шифр направления (код специальности)

Проект защищен
_____
(дата)
с оценкой_____

Димитровград
2007

Содержание

Ведение

В современном мире очень сложно предотвратить атаку на собственную систему, в которой работает пользователь. Для понимания того, как происходит процесс исследования “системы-жертвы” перед атакой, и предназначена данная курсовая работа.

Сканирование портов - первый шаг в процессе взлома или предупреждения взлома, поскольку он помогает определить потенциальные цели. Вокруг каждого хоста - независимо от аппаратного и программного обеспечения или выполняемых функций - имеется некоторое количество идентифицируемых особенностей. Внимательный наблюдатель (скорее всего им может быть высококвалифицированный хакер), вооруженный соответствующим инструментарием, может исследовать работающие на машине службы (Web-сервер, FTP-сервер, mail-сервер, и т.д.), номера версий и даже операционную систему, посылая им пакеты данных и анализируя то, как они отвечают.

Несмотря на ежедневные инциденты со взломом систем, многие люди размещают свои компьютеры в Интернете совершенно без подготовки. Даже в сфере индустрии информационных технологий системные администраторы могут установить самую последнюю версию Linux на новейший сервер, дополнительное программное обеспечение и позволить ему что-то выполнять. Как только такая система буде подвергнута исследованию, некто получит возможность определить не только то, что система работает под управлением Linux, но и то, какая версия Linux работает, а также номер версии системы. Используя известные уязвимости ОС для данной версии, взломщик сможет получить контроль над системой.

Данная курсовая работа может быть адресована не только студентам, имеющих непосредственное отношение к информационным технологиям, но и продвинутым пользователям, желающих вникнуть в суть проблемы – как защитить свой компьютер при серфинге Интернета от возможности пригласить в свою систему сетевого маньяка.

1. Теоретическая часть

1.1. Понятие сканирования портов

Сканированием портов называется метод удаленного анализа, осуществляемый путем передачи тестовых запросов на создание соединения и позволяющий определить список активных служб предоставления удаленного сервиса на каком-либо хосте. Сканирование портов (или разведка) применяется на подготовительной стадии перед атакой, так как позволяет получить необходимые начальные сведения о потенциальном объекте воздействия: список открытых портов, а следовательно, и перечень потенциально атакуемых серверных приложений, загруженных на компьютере.

Все известные на сегодняшний день основные методы сканирования портов в зависимости от возможности определения объектом непосредственного инициатора сканирования (хоста, откуда осуществлялся удаленный анализ) можно разделить на две группы:

– *методы открытого сканирования*: непосредственный инициатор однозначно определяется объектом сканирования по IP-адресу приходящих запросов.

– *методы "невидимого" анонимного сканирования*. Непосредственный инициатор не определяется объектом сканирования (однозначно определяется только "промежуточный" источник сканирующих запросов), таким образом, гарантируется анонимность инициатора сканирования.

1.2. Введение в теорию сканирования портов

Сканирование позволяет осуществлять поиск каналов передачи данных. Идея сканирования заключается в том, чтобы исследовать как можно больше потенциальных каналов связи и определить, какие именно находятся в состоянии ожидания соединения.

В качестве канала связи теоретически может выступать любая совокупность приемопередающего оборудования со средой передачи данных. Мы рассмотрим лишь возможную его реализацию, представляющую собой компьютерный терминал (хост), подключенный к сети по коммутируемой либо выделенной линии.

Термин «порт» является абстрактным понятием, используемым для упрощенного описания механизма установления соединения между компьютерами. Следуя приведенной выше терминологии, порт представляет собой потенциальный канал передачи данных.

Использование механизма портов существенно облегчает процесс установления соединения и обмена информацией между компьютерами.

Как и везде, в организации механизма портов имеются свои недостатки. Любой пользователь имеет возможность исследовать сетевое окружения сервера методом опроса его портов. Достаточно лишь послать «лавину» пакетов на все возможные номера портов

сервера (1-65535), и по тому, от каких портов будут (или не будут) получены ответы, определить открытые порты и службы, работающие на исследуемом сервере.

Существует большое количество алгоритмов для поиска активных портов хоста и соответствующих им служб.

Существуют два общих способа сканирования – сканирование TCP-портов и сканирование UDP-портов.

1.3. Методы сканирования TCP-портов

Определение состояния сервера методом ICMP-сканирования

Перед непосредственным сканированием портов удаленного компьютера необходимо выяснить его состояние – работает он в сети или нет. Особенно это важно при сканировании группы хостов либо при сканировании определенного сегмента сети, где помимо определения функционирующих хостов необходимо также определить их адреса. Для этого необходимо отправить специфический запрос, означающий, что пользователю необходима информация о состоянии сервера (либо хоста) и исправности сетевых средств, обеспечивающих маршрутизацию пакетов.

В сетях, организованных на базе стека протоколов TCP/IP, для этой цели используется протокол ICMP. Данный протокол является вспомогательным и позволяет маршрутизатору сообщать конечному узлу об ошибках либо непредвиденных ситуациях, которые имели место при передаче IP-дейтаграммы от этого узла.

Обмен информацией между маршрутизатором и узлом реализован с помощью ICMP-сообщений. Помимо сообщений об ошибках, в протоколе ICMP предусмотрен ряд стандартных запросов, позволяющих хосту получить различного рода информацию о состоянии объектов сети.

В качестве упомянутого выше запроса хост отправляет серверу ICMP-сообщение и ожидает получения ответа, также представляющего собой ICMP-сообщение (ICMP-эхо). Сообщение ICMP пользователь может сформировать программным способом либо использовать для этого средства операционной системы.

Для программной реализации данного метода пользователю необходимо знать некоторые особенности построения подобных запросов.

В начале любого ICMP-сообщения находятся три поля: «Тип сообщения», «Код» (причина ошибки) и «Контрольная сумма». Поле «Тип» определяет смысл ICMP-сообщения и соответствующий ему формат.

Сканирование TCP-портов функцией connect()

Данный метод использовался в самом начале развития технологии сканирования, однако до сих пор является основным и единственным в некоторых операционных системах

(Windows), поддерживающих механизм сокетов, для сканирования портов по протоколу TCP. Функция connect() позволяет хосту соединиться с любым портом сервера. Если порт, указанный в качестве параметра функции, прослушивается сервером (т.е. порт открыт для соединения), то в результате выполнения функции connect(n) будет установлено соединение с сервером по указанному порту n. В противном случае, если соединение не установлено, то порт с номером n является закрытым.

Этот метод обладает некоторыми преимуществами. Во-первых, его может применить любой пользователь, не обладающий никакими привилегиями на хосте. Во-вторых, данный метод обеспечивает довольно высокую скорость исследования. Последовательный перебор портов путем вызова функции connect() для каждого номера порта, определение его состояния и закрытие соединения – достаточно долгий процесс. Однако его можно ускорить, применив метод «параллельного просмотра» с использованием неблокированного ввода/вывода (non-blocked I/O). Такой метод позволяет практически одновременно определить состояние всех портов сервера.

Большим недостатком данного метода является возможность обнаружения и фильтрации такого рода сканирования, причем сделать это достаточно легко. Log-файл сканируемого сервера укажет службам, отвечающим за внешние подключения, на наличие многочисленных запросов на соединение с одного и того же адреса и ошибок создания соединения с ним, поскольку хост исследующего после создания соединения с сервером сразу же обрывает его. Службы внешних подключений, в свою очередь, немедленно заблокируют доступ к серверу для хоста с данным адресом.

Сканирование TCP-портов флагом SYN

Данный метод известен еще как «сканирование с установлением наполовину открытого соединения» (half-open scanning), поскольку полное установление TCP-соединения не производится.

Алгоритм сканирования следующий. Хост отправляет на определенный порт сервера SYN-пакет, как бы намереваясь создать соединение, и ожидает ответ. Наличие в ответе флагов SYN|ACK означает, что порт открыт и прослушивается сервером. Получение в ответ TCP-пакета с флагом RST означает, что порт закрыт и не прослушивается.

В случае приема SYN|ACK-пакета хост немедленно отправляет RST-пакет для сброса устанавливаемого сервером соединения и не продолжает процесс обмена синхропакетами. Таким образом, производится проверка способности сканируемого сервера установить соединение по указанному порту.

Преимущество данного метода заключается в том, что лишь немногие серверы способны зарегистрировать такого рода сканирование без использования специальных

средств защиты. Метод возможно использовать только в случае, если на хосте, с которого производится сканирование, установлена операционная система из семейства UNIX. Кроме того, пользователь должен обладать статусом root, в противном случае пользователь попросту не сможет программно сформировать одиночный SYN-пакет.

Сканирование TCP-портов флагом FIN

Лишь немногие серверы способны отследить попытку SYN-сканирования их портов. Так, некоторые файрволлы и пакетные фильтры «ожидают» поддельные SYN-пакеты на закрытые порты защищенного ими сервера, и специальное программное обеспечение распознает попытку SYN-сканирования. Если сервер обрывает соединение после опроса нескольких портов, используется FIN-сканирование.

В этом методе используются FIN-пакеты, используемые в процедуре закрытия соединения. Пакет предусматривает установку в TCP-сообщении флага FIN.

FIN-пакеты способны обойти средства защиты сети. Идея заключается в том, что на прибывший на закрытый порт FIN-пакет сервер должен ответить RST-пакетом (TCP-пакет с установленным в нем флагом RST). FIN-пакеты на открытые порты игнорируются сервером.

Сканирование TCP-портов флагами SYN(FIN) с использованием IP-фрагментации

Данный метод представляет собой комбинацию SYN и FIN-сканирования с небольшим усовершенствованием. Он основан на использовании функциональной особенности протокола IP, называемой *фрагментацией*.

Фрагментация – это процесс разделения большого пакета данных на несколько частей перед непосредственной передачей его в сеть для получения размера фрагмента, соответствующего стандарту используемой сети (*параметр MTU* – Maximum Transmission Unit, максимальный размер блока). Фрагментация пакета на стороне источника и его сборка на стороне приемника осуществляется автоматически. Каждая фрагментированная часть исходного пакета имеет одинаковый формат. Этот метод позволяет маршрутизировать фрагменты независимо друг от друга.

Таким образом, TCP-пакет (SYN или FIN-пакет) разбивается на стороне хоста на пару IP-фрагментов меньшего размера, и эта пара IP-фрагментов отправляется серверу. На стороне сервера IP-фрагменты «собираются» в один TCP-пакет и производится его обработка (те же действия, как и при SYN или FIN-сканировании).

В этом случае фрагментация позволяет уменьшить вероятность обнаружения сканирования фильтрами пакетов и другим подобным оборудованием. Однако при этом следует быть очень осторожным, поскольку некоторые программы имеют обыкновение «зависать» при попытке обработки такого маленького IP-фрагмента.

Сканирование TCP-портов методом reverse-ident (обратной идентификации)

Протокол ident позволяет определить имя (username или login, указанное при входе в систему) владельца любого запущенного на сервере процесса, связанного с ним, даже если сам этот процесс не инициализировал TCP-соединение.

Протокол ident иначе называется *протоколом аутентификации сервера*. За ним зарезервирован 113-й TCP-порт, который используется демоном identd, выполняющим функции аутентификации согласно протокола ident, для приема запросов и передачи ответов на них. Этот процесс происходит следующим образом.

Сервер прослушивает 113 порт и ожидает прихода запроса на соединение. Как только соединение установлено, сервер считывает блок данных, характеризующий соединение, для которого необходимо получить информацию аутентификации. В запросе, помимо информации, находящейся в IP и TCP-заголовках, передается небольшой текстовый блок данных, состоящий из двух полей:

<Порт сервера>,<Порт клиента>

где <Порт сервера> - это номер порта сервера, на котором запущен identd и о котором необходимо получить информацию, а <Порт клиента> - номер порта на хосте, посылающем запрос серверу, на который сервер должен прислать ответ.

Как и большинство программ, identd имеет некоторые интересные особенности функционирования, позволяющие получить требуемую информацию, не используя стандартный 113 порт и уж тем более не проходя процедуру аутентификации. Так, например, имеется возможность подключиться к http-порту и затем использовать identd, чтобы определить, работает ли на сервере пользователь root.

К сожалению, это может быть сделано только при установлении «полного» TCP-соединения к порту исследуемого сервера, что позволяет системному администратору отследить действия злоумышленника.

1.4. Методы сканирования UDP-портов

Сканирование UDP-портов проверкой ICMP-сообщения «Порт недостижим»

Этот метод предназначен для определения состояния портов сервера. Основным отличием является использование протокола UDP вместо протокола TCP. Не смотря на то, что организация протокола UDP проще, чем TCP, сканировать UDP-порты гораздо труднее. Это связано прежде всего с концепцией протокола UDP как протокола с негарантированной доставкой данных. Поэтому UDP-порт не посылает подтверждение приема запроса на установление соединения, и нет никакой гарантии, что отправленные UDP-порту данные успешно дойдут до него.

К счастью, большинство серверов в ответ на пакет, прибывший на закрытый UDP-порт, отправляют ICMP-сообщение «Порт недоступен» (Port Unreachable - PU). Таким образом, если в ответ на UDP-пакет пришло ICMP-сообщение «PU», то сканируемый порт является закрытым, в противном случае (при отсутствии «PU») порт открыт. Поскольку нет гарантии, что запросы хоста дойдут до сервера, пользователь должен позаботиться о повторной передаче UDP-пакета, который, по всей видимости, оказался потерянным.

Этот метод работает очень медленно из-за использования на некоторых машинах т.н. «компенсации», ограничивающей частоту генерирования ICMP-сообщений об ошибке. Например, ядро Linux ограничивает частоту генерирования ICMP-сообщения «адресат недостижим» (Destination Unreachable) до 80 сообщений за 4 секунды, с простоем ¼ секунды, если это ограничение было превышено. Кроме того, для использования данного метода (а именно – для обнаружения ICMP-сообщений об ошибке) пользователь должен обладать статусом root на хосте, с которого производится сканирование.

Сканирование UDP-портов с использованием функций recvfrom() и write()

Этот метод используется в случае, когда пользователь, проводящий сканирование, не обладает статусом root на хосте. Поскольку не - root пользователь не может «читать» ICMP-сообщение PU, в ОС, поддерживающих механизм сокетов (например в Linux), имеется возможность получения информации о состоянии UDP-порта косвенным способом. Так, например, попытка вызова функции write() на закрытый порт обычно приводит к возникновению ошибки.

Функция recvfrom() в этом плане более информативна. Вызов ее на неблокированный UDP-сокет сервера обычно возвращает ошибку EAGAIN (Try Again – «попытайтесь еще раз», код 13) в случае, когда ICMP-сообщение не было принято, и ECONREFUSED (Connection Refused – «соединение закрыто», код 111), если ICMP-сообщение было принято.

Таким образом, по всем вышеперечисленным признакам возможно определить состояние портов сканируемого сервера. Наибольшая эффективность достигается при использовании комплексного метода сканирования, предусматривающего выбор конкретного метода либо их совокупности в зависимости от конкретной ситуации.

1.5. Эталон TCP-взаимодействия между клиентом и сервером

Когда для определенного порта создано TCP-соединение, клиент посылает TCP-пакет с установленным флагом SYN для инициализации соединения. Если сервер прослушивает этот порт, он посылает пакет с установленными флагами SYN и ACK, подтверждая клиентский запрос на соединение, одновременно запрашивая установление обратного соединения. Затем клиент может послать пакет с установленным флагом ACK, подтверждая запрос SYN от сервера. Такой способ известен как *трехходовое установление связи TCP*. Когда одна из

сторон выполнит передачу другой, она может послать пакет FIN. Другая сторона должна подтвердить этот пакет FIN и послать свое собственное сообщение FIN, ожидая от другой стороны подтверждения того, что соединение действительно закрыто. Для экстренного прерывания соединения любая из сторон может передать пакет с флагом RST. Простое TCP-взаимодействие между клиентом и сервером представлено ниже.

1. Клиент посылает серверу сообщение SYN: «Я запрашиваю соединения».
2. Сервер посылает SYN/ACK-клиенту: «Okay; мне необходимо соединение с тобой».
3. Клиент посылает сообщение ACK-серверу: «Okay».
4. Клиент и сервер посылают информацию вперед-назад, подтверждая каждую передачу сигналом ACK. Если одна из сторон посылает сообщение RST, соединение прерывается немедленно.
5. Клиент желает завершить взаимодействие; клиент посылает серверу сообщение FIN: «Goodbye».
6. Сервер посылает сообщение ACK-клиенту (подтверждая получение сообщения FIN). Затем сервер посылает собственное сообщение FIN: «Okay. Goodbye».
7. Клиент посылает подтверждение ACK-серверу (подтверждая получение его сообщения FIN): «Okay».

Таблица 1.1

Определение флагов TCP

Признак	Описание
SYN	Используется для обозначения начала соединения TCP
ACK	Используется для уведомления о получении предыдущего пакета или цикла передачи
FIN	Используется для закрытия соединения TCP
RST	Используется для внезапного прекращения соединения TCP

2. Практическая часть

2.1. Установка и удаление NMAP

У нас имеется дистрибутив программы как для платформы Linux, так и для Windows. Установка для Windows заключается распаковке пакета Nmap 4.11.zip и инсталляции пакета WinPcap 3.1. Последний нужен для доступа к сырым сокетам и другим сетевым библиотекам.

Для платформы RED HAT ENTERPRISE LINUX 4 AS существует дистрибутив nmap-4.11-1.i386.rpm, программf которого работает в режиме командной строки и графический пакет nmap-frontend-4.11-1.i386.rpm к нему. Ниже приводятся команды для установки:

```
[root@DBADOMAIN ~]# rpm -i /root/nmap-4.11-1.i386.rpm
[root@DBADOMAIN ~]# rpm -i /root/nmap-frontend-4.11-1.i386.rpm
```

и удаления пакетов rpm:

```
[root@DBADOMAIN ~]# rpm -e nmap-frontend
[root@DBADOMAIN ~]# rpm -e nmap
```

2.2. Применение NMAP

Nmap один из наиболее доступных сканеров портов. Его можно загрузить по адресу http://www.insecure.org и с легкостью установить его на большинстве Unix-систем. В нашей работе мы будем использовать версию NMAP для Linux номер 4.11 beta 1.

NMAP (Network Mapper) - открытый исходный инструмент для исследования сети и анализа безопасности. Он был разработан, чтобы быстро анализировать большие сети, хотя он также хорошо работает и с одним хостом. Nmap использует *«сырые IP пакеты»*, чтобы определить доступность того или иного хоста в сети, какие сервисы предоставляют сервера, какие операционные системы они используют и множество других характеристик. Администраторы также предпочитают использовать этот инструмент для обслуживания сети.

Напомним, что «сырые IP пакеты» – это пакеты, сформированные приложениями самостоятельно, т.е. в этом пакете можно указывать произвольные протоколы, порты и адреса. При перехвате IP пакетов установить принадлежность таких пакетов соединениям и приложениям невозможно.

Большинство типов сканирования портов доступно только привилегированным пользователям. Это происходит потому, что они используют сырые пакеты, которые требуют прав root в ОС. Использование учетной записи администратора рекомендуется, хотя Nmap иногда работает для непривилегированных пользователей на той платформе, когда WinPcap уже был загружен в операционной системе.

Одна из причин, по которой NMAP так широко используется, состоит в том, что он предоставляет много приемов для сканирования. Вы можете сканировать работающие хосты на предмет наличия TCP-портов, UDP-портов и любых других IP-протоколов.

2.3. Проверяем доступность хостов

Начнем с простого определения присутствия хостов в сети. Для этого можно использовать метод сканирования с использованием Ping (-sP). Этот метод работает также как fping, когда он посылает ICMP эхо-запросы по заданному промежутку IP-адресов и ожидает ответа. Однако многие хосты на сегодняшний день блокируют ICMP-запросы. В этом случае nmap дает возможность установить с хостом TCP-соединение по 80 порту (по умолчанию). Если он получает что-либо (SYN/ACK или RST), значит хост работает. Если он ничего не получает, хост маркируется как не работающий или отключенный от сети.

Важно понимать, как работает TCP Ping-сканирование для IP-адреса. Если сервис прослушивает порт, и кто-то пытается установить соединение с ним (посылая SYN пакет), сервис может послать в ответ пакет SYN/ACK. Это наглядно показывает, что по этому IP-адресу есть машина. Однако если отсутствует сервис, который прослушивал бы указанный порт, а машина находится в сети, в ответ будет послан пакет RST. Независимо от того, отвечает машина на запрос, или по заданному порту нет работающего сервиса, сам факт такого ответа подтверждает, что по заданному IP-адресу есть машина. Если в ответ на посланный SYN-пакет ничего не получено, это может означать, что по заданному IP-адресу нет никакой машины, или что такой трафик блокируется брандмауэром. 80 порт задан по умолчанию, поскольку большинство брандмауэров и фильтров пропускают Web-трафик.

```
[root@DBADOMAIN ~]# nmap -sP 192.168.50.1
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 04:46 MSK
Host 192.168.50.1 appears to be up.
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 13.374 seconds
```

Если никакого ответа не получено, nmap с достаточной степенью уверенности может предположить, что хост не работает.

```
[root@DBADOMAIN ~]# nmap -sP 192.168.50.10
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 04:47 MSK
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0
Nmap finished: 1 IP address (0 hosts up) scanned in 0.239 seconds
```

Для получения имен хостов для заданного IP-интервала можно использовать опцию -sL.

```
[root@DBADOMAIN ~]# nmap -sL 192.168.50.1-3
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 05:05 MSK
Host 192.168.50.1 not scanned
Host 192.168.50.2 not scanned
```

Host 192.168.50.3 not scanned
Nmap finished: 3 IP addresses (0 hosts up) scanned in 13.014 seconds

2.4. Сканирование TCP-портов

Основной метод сканирования TCP-портов - это установление TCP-соединения connect() (-sT) с портом, чтобы посмотреть, будет ли получен ответ. То же самое делает TCP-клиент, желая установить соединение (законченное трехходовое соединение), за исключением того, что nmap может разорвать соединение, послав пакет RST, как только соединение будет установлено.

[root@DBADOMAIN ~]# **nmap -sT 192.168.50.3**
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2006-11-18 05:12 MSK
Interesting ports on 192.168.50.3:
Not shown: 1678 closed ports
PORT STATE SERVICE
22/tcp open ssh
111/tcp open rpcbind
Nmap finished: 1 IP address (1 host up) scanned in 13.176 seconds

Можно использовать сканирование RPC (-sR) для сканирования любого открытого для RPC-сервиса порта.

[root@DBADOMAIN ~]# **nmap -sR 192.168.50.3**
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2006-11-18 05:11 MSK
Interesting ports on 192.168.50.3:
Not shown: 1678 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh
111/tcp open rpcbind (rpcbind V2) 2 (rpc #100000)
Nmap finished: 1 IP address (1 host up) scanned in 13.245 seconds

Таблица 2.1

Сканирование -sT,-sR,-sP

NMAP посылает на порт хоста	NMAP получает от порта хоста	NMAP отвечает	NMAP принимает решение
SYN	SYN/ACK	ACK следом за RST	Порт открыт, хост работает
SYN	RST	-	Порт закрыт, хост работает
SYN	Ничего	-	Порт блокирован брандмауэром или хост не работает

Поскольку вы всего лишь создали TCP-соединение, оно, скорее всего, будет зарегистрировано службой, предоставившей его.

Nmap позволяет сделать много интересного с TCP-пакетами, которые вы используете для сканирования портов. Во-первых, есть *SYN-сканирование* (-sS), которое создает первую половину TCP-соединения (посылая TCP-пакет с установленным флагом SYN) но затем ведет себя несколько иначе. Если приходит TCP-пакет с установленным флагом RST, nmap

решает, что порт закрыт и ничего больше не предпринимает. Однако если приходит ответ (о чем свидетельствует пакет с установленным флагом SYN/ACK), вместо того чтобы подтвердить получение этого пакета, как это было бы при установлении нормального соединения, посылается RST-пакет, как это показано в следующей таблице. Поскольку трехходовое TCP-соединение не завершено, многие сервисы не регистрируют соединение. Поскольку вы осуществляете манипуляции с некоторыми из этих TCP-флагов на низком уровне, вы не можете реализовать эти типы сканирования, не имея полномочий пользователя root в системе.

Таблица 2.2

Сканирование -sS (SYN-сканирование)

NMAP посылает на порт хоста	NMAP получает от порта хоста	NMAP отвечает	NMAP принимает решение
SYN	SYN/ACK	RST	Порт открыт, хост работает
SYN	RST	-	Порт закрыт, хост работает
SYN	Ничего	-	Порт блокирован брандмауэром или хост не работает

```
[root@DBADOMAIN ~]# nmap -sS 192.168.50.1
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 05:34 MSK
Interesting ports on 192.168.50.1:
Not shown: 1677 closed ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 14.792 seconds
```

Так же как сервисы могут не регистрировать «незавершенное» соединение, некоторые брандмауэры или системы определения вторжений (IDS) могут быть настроены на поиск таких типов сканирования. У Nmap есть несколько способов сканирования, но и хорошие IDS-системы могут поймать вас. Вдобавок, брандмауэр может фильтровать подозрительные пакеты и искажать результаты сканирования.

Вы должны были уже понять, что в любой момент, как только вы посылаете TCP-пакет на закрытый порт, стек TCP/IP на другой стороне готов послать в ответ RST-пакет. Что же в этом случае происходит с разрешенными TCP пакетами? Если закрытый порт всегда отвечает пакетами RST, почему бы просто не посылать не имеющие смысла пакеты и не смотреть, что придет в ответ?

FIN-сканирование (-sF) посылает пакеты FIN, которые обычно используются для закрытия соединения. Однако, поскольку мы посылаем его прежде, чем соединение было

установлено, открытый порт должен игнорировать такой мусор. Закрытый порт по-прежнему будет отвечать пакетом RST, как это показано в следующей таблице. Nmap предлагает два способа мусорного сканирования: *Xmas tree* (-sX) (рождественское) сканирование (которое устанавливает FIN, URG, и PUSH флаги TCP-пакета, расцвечивая его подобно новогодней елке) и *null-сканирование* (-sN) (которое выключает все флаги, подобно тому, как это делает hping по умолчанию). Поскольку мы совершаем некоторые манипуляции с пакетами на низком уровне, то такое сканирование также требует полномочий пользователя root. Имейте в виду, что не все стеки TCP/IP реализованы корректно. Даже притом, что открытые порты не посылают RST-пакеты в ответ на такие виды проверок, некоторые стеки TCP/IP не следуют этим правилам и посылают ответ в любом случае. Это означает, что вы можете ошибочно сделать положительное предположение при использовании такого сканирования для некоторых типов хостов. Также любой хост, защищенный брандмауэром, может вернуть ложный ответ. Nmap предполагает, что порт открыт, если не получает ничего в ответ. Что если брандмауэр блокирует этот ответ? Такое сканирование более скрытое, но оно и менее аккуратное.

Таблица 2.3

Сканирование -sF (FIN-сканирование)

NMAP посылает на порт хоста	NMAP получает от порта хоста	NMAP принимает решение
FIN	Ничего	Порт открыт, если хост работает и не защищен брандмауэром
FIN	RST	Порт закрыт; хост работает

```
[root@DBADOMAIN ~]# nmap -sX 192.168.50.1
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 05:36 MSK
All 1680 scanned ports on 192.168.50.1 are closed
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 14.786 seconds
[root@DBADOMAIN ~]# nmap -sN 192.168.50.1
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 05:37 MSK
All 1680 scanned ports on 192.168.50.1 are closed
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 15.019 seconds
```

Иногда nmap может сообщить вам, что порт подвергается фильтрации. Это означает, что влияние брандмауэра или фильтра накладывается на возможность nmap точно определить, открыт или закрыт порт. Некоторые брандмауэры, тем не менее, могут только фильтровать входящие соединения (это означает, что они просматривают только входящие SYN-пакеты на конкретном порту). Если вы хотите проверить правила брандмауэра, запустите ACK-сканирование для хоста, находящегося за брандмауэром. Всякий раз, как

ACK-пакет (подтверждение) посылается не как часть существующего соединения, принимающая сторона предположительно посылает пакет RST. Сканирование ACK (-sA) может использовать этот факт для определения, осуществляется ли блокирование или фильтрация порта. Если получен пакет RST, порт не фильтруется; в противном случае, осуществляется фильтрация, как это показано в следующей таблице. Именно сканирование-ACK может показать вам, что конкретный хост защищен брандмауэром.

Таблица 2.4

Сканирование -sA (ACK-сканирование)

NMAP посылает на порт хоста	NMAP получает от порта хоста	NMAP принимает решение
ACK	RST	Порт не защищен брандмауэром; порт может быть открыт или закрыт; хост работает
ACK	Ничего или ICMP unreachable	Порт блокирован брандмауэром и хост работает

Поскольку такое сканирование не может сказать, закрыт ли на самом деле порт или открыт, возможно, вы захотите использовать другие виды сканирования в комбинации с ACK-сканированием. Например, вы можете использовать ACK-сканирование в комбинации с SYN-сканированием (-sS), чтобы определить, что хост защищен брандмауэром, который использует полную проверку пакетов или только проверку входящих соединений (SYN-флаг). Выключим файрволл на машине 192.168.181.1 и проведем сканирование:

C:\Nmap 4.11>**nmap –sA 192.168.181.1**
Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2007-01-10 19:03 Московское время (зима)
All 1680 scanned ports on 192.168.181.1 are Unfiltered
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 2.250 seconds

Теперь включим: экран заблокировал хост атакующего, и NMAP ушел в бесконечное ожидание – это говорит о том, что порты блокированы и хост 192.168.181.1 работает.

Рис. 1. Отчет экрана при ACK-сканировании

2.5. Сканирование UDP-портов

Естественно, NMAP это тоже делает. Используя флаг -sU, можно посылать пустые UDP-пакеты и ожидать в ответ ICMP-сообщения «port unreachable». Если не получено

никаких сообщений в ответ, порт считается открытым, как это показано в следующей таблице. Здесь вы можете видеть несколько возможных ошибок. Если возвращаемые ICMP-сообщения блокируются брандмауэром, это может означать, что все UDP-порты хоста открыты. Также, если UDP-трафик непосредственно блокируется брандмауэром, это по-прежнему означает, что все UDP-порты открыты. Вдобавок, многие хосты могут посылать наружу только ограниченное число ICMP-сообщений об ошибках в секунду для предотвращения перегрузки сети. NMAP может автоматически корректировать эту ситуацию, если она определяется, но это может очень сильно замедлить сканирование. Поскольку UDP-протокол работает без установления соединения и не ограничен необходимостью подтверждения получения входящих пакетов, нет полного решения этой проблемы.

Таблица 2.5

Сканирование -sU (UDP-сканирование)

NMAP посылает на порт хоста	**NMAP получает от порта хоста**	**NMAP принимает решение**
Пустой UDP пакет	Ничего	Возможно, порт открыт, если хост отвечает на Ping (хост работает); возможно, порт закрыт, если брандмауэр блокирует ICMP
Пустой UDP пакет	ICMP порт недоступен	Порт закрыт

C:\Nmap 4.11>**nmap -sU 192.168.181.1**
Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2007-01-10 19:25 Московское время (зима)
All 1487 scanned ports on 192.168.181.1 are open|filtered
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 34.235 seconds

Рис. 2. Отчет экрана при UDP-сканировании

Nmap по умолчанию пытается послать Ping хосту перед тем, как начать его сканировать. Это особенно важно для правильной интерпретации результатов UDP-сканирования. Если nmap не может послать Ping хосту (или потому, что хост блокирован брандмауэром, или вы отключили такую возможность, вручную используя флаг -P0), он не сможет получить правильные результаты.

2.6. Сканирование протоколов

Если вы предприняли неудачную попытку соединения с UDP-портом, хост вернет ICMP-сообщение «port unreachable». То же самое можно сказать и про IP-протоколы. Каждый IP-протокол транспортного уровня имеет соответствующий номер. Наиболее распространены ICMP (1), TCP (6) и UDP (17). У всех IP-пакетов есть поле «protocol», которое показывает тип заголовков пакетов и номер протокола транспортного уровня. Если мы посылаем серию пакетов без заголовка протокола транспортного уровня и с номером протокола 130, который означает протокол семейства IPSEC, называемый SPS (Secure Packet Shield), мы можем определить, какой из протоколов реализован на данном хосте. Если мы получаем ICMP-сообщение «protocol unreachable», то этот протокол недоступен. В противном случае - доступен. Этот метод сканирования, называемый *сканированием протоколов* (-sO), страдает теми же самыми проблемами, что и UDP-сканирование в том случае, если брандмауэр блокирует ICMP-сообщения или непосредственно сам протокол дает ложные ответы.

При выключенном брандмауэре результат таков:

```
C:\Nmap 4.11>nmap 192.168.181.1 -sO
Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2007-01-10 19:39 Московское время
(зима)
Interesting protocols on 192.168.181.1:
Not shown: 250 closed protocols
PROTOCOL STATE         SERVICE
1        open          icmp
2        open|filtered igmp
6        open          tcp
17       filtered      udp
47       open|filtered gre
255      open|filtered unknown
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 1.844 seconds
```

При включенном:

```
C:\Nmap 4.11>nmap 192.168.181.1 -sO
Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2007-01-10 19:46 Московское время
(зима)
All 256 scanned ports on 192.168.181.1 are open|filtered
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 7.328 seconds
```

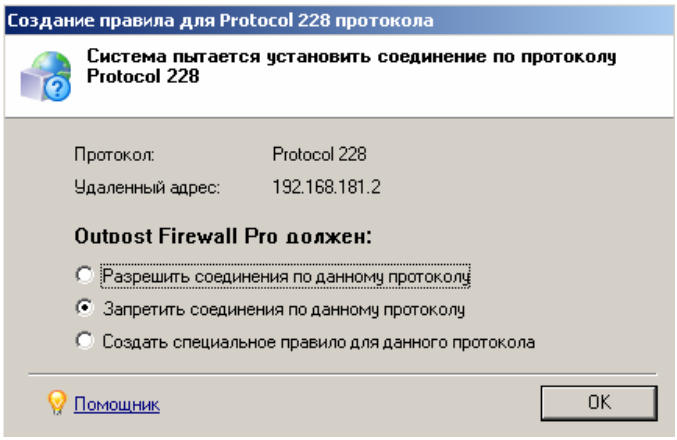

Рис. 3. Отчет администратору хоста 192.168.181.1

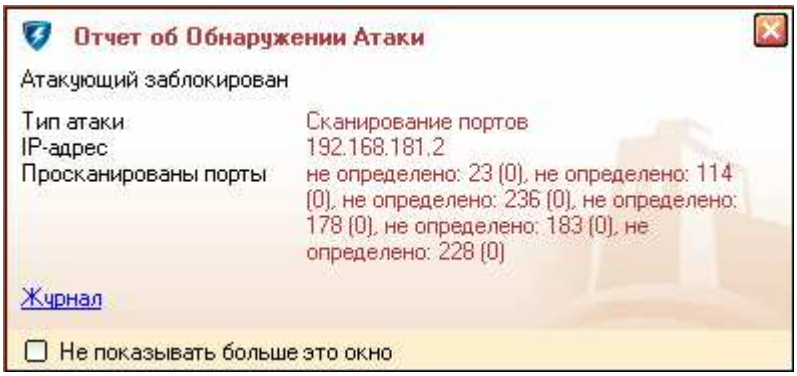

Рис. 4. Отчет при сканировании протоколов

2.7. Обман брандмауэров и IDS

У nmap есть несколько опций сканирования. Некоторые из них могут помочь скрыть сканирование портов от определения по системным журналам с помощью брандмауэров и IDS-систем. Вдобавок, NMAP предоставляет некоторые возможности рандомизации и маскировки, которые предоставляются такими программами, как Netcat и hping.

Фрагментация пакетов при сканировании

Тип атаки «Short fragments» – пакет разбивается на несколько фрагментов, которые затем изменяются таким образом, что после сборки пакет приводит к зависанию системы.

Опция -f NMAP, указывает на необходимость выполнения скрытого сканирования с использованием фрагментированных IP-пакетов, у которых разорваны TCP-заголовки. Идея состоит в предохранении таких «искаженных» TCP-пакетов с необычными флагами от блокировки брандмауэрами. Эта опция может привести к нарушению в работе некоторых систем и не может корректно работать на всех разновидностях Unix.

Обнаружение брандмауэром

Да, брандмауэр воспринял сканирование как атаку «Short fragments»

Рис. 5. Отчет при сканировании

Результат работы

[root@DBADOMAIN ~]# **nmap -f 192.168.50.1**
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2006-11-17 21:15 MSK
All 1680 scanned ports on 192.168.50.1 are filtered
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 48.477 seconds

Заманивание

NMAP дает возможность определить хост «ловушку», используя опцию -D. Идея состоит в определении нескольких «мистических жертв» - хостов в Интернете - и задания их в разделенном запятыми списке после флага -D. NMAP будет, как обычно, выполнять роль сканера порта, но при этом определяться среди замаскированных сканеров портов с IP-адресами - ловушками. Системный администратор увидит несколько различных сканеров портов, но только один из них будет настоящим.

Обнаружение брандмауэром

Список заблокированных узлов

Время	Атака	Узел	Действие
19:57:27	Сканирование портов	192.168.50.12	Атакующий заблокирован на 5 мин. (Разблокировать)
19:57:27	Сканирование портов	192.168.50.11	Атакующий заблокирован на 5 мин. (Разблокировать)
19:57:27	Сканирование портов	192.168.181.2	Атакующий заблокирован на 5 мин. (Разблокировать)
19:57:27	Сканирование портов	192.168.50.10	Атакующий заблокирован на 5 мин. (Разблокировать)

Рис. 6. Заблокированные узлы после сканирования с узла 192.168.181.2

Результат работы

C:\Nmap 4.11>**nmap 192.168.181.1 -D 192.168.50.10,192.168.50.11,192.168.50.12**

Подмена исходящего адреса

Опция –S позволяет задать исходящий IP-адрес для пакетов. Данный способ можно использовать для создания впечатления, что сканирование производится с заданного адреса. Эта функция имеет смысл для хоста с несколькими адресами, но может быть использована для маскировки и создания общего беспорядка. При этом способе невозможно получить обратно результаты сканирования, но можно выставить кого-то, как вредителя. В случае если системы обнаружения вторжений или брандмауэры оснащены функциями против приманки, то атакующего смогут обнаружить.

Обнаружение брандмауэром

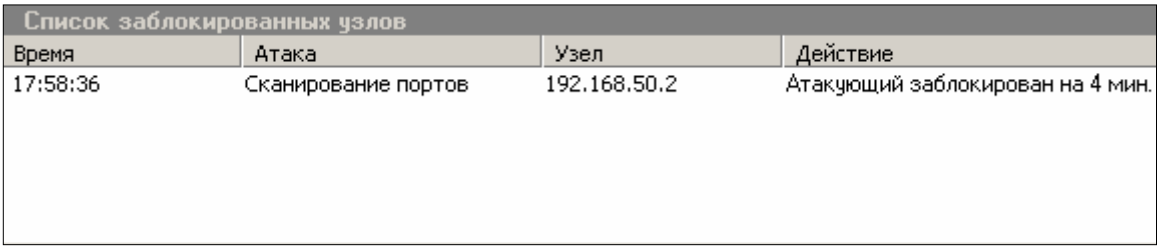

Рис. 7. Заблокированные узлы после сканирования с узла 192.168.50.3

Рис. 8. Отчет о сканировании

Результат работы

[root@DBADOMAIN ~]# **nmap 192.168.50.1 -S 192.168.50.2 -e eth0**
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2006-11-17 22:31 MSK
All 1680 scanned ports on 192.168.50.1 are filtered
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 48.653 seconds

Пакеты с поддельной контрольной суммой TCP/UDP

Опция --badsum позволяет использовать недействительную TCP или UDP контрольную сумму для пакетов, посланных целевым хостам. Те брандмауэры и IDS, которые не проверяют контрольную сумму пакетов, пропустят ответ системы атакуемому хосту.

Обнаружение брандмауэром

Firewall обнаружил сканирование портов.

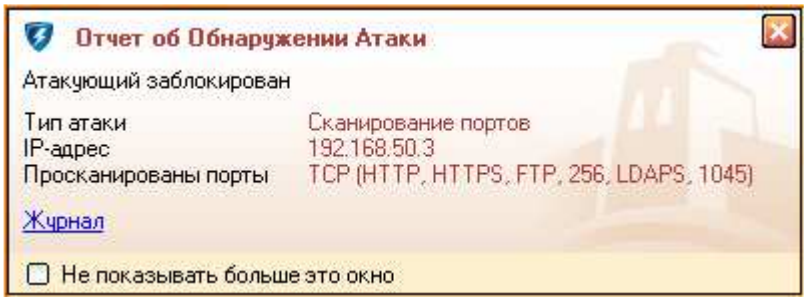

Рис. 9. Отчет о сканировании

Результат работы

[root@DBADOMAIN ~]# **nmap 192.168.50.1 --badsum**
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2006-11-17 22:49 MSK
All 1680 scanned ports on 192.168.50.1 are filtered
MAC Address: 00:50:56:C0:00:01 (VMWare)

2.8. Режим определения версии служб

Опция -sV (scan Version) позволяет включить режим определения версий служб, за которыми закреплены сканируемые порты. После окончания сканирования будет получен список открытых TCP и/или UDP-портов. Без этой опции в списке напротив каждого порта будет указана служба, которая обычно использует данный порт (эта информация берется из базы данных "общеизвестных" портов, файл *nmap-services*). При включенной опции будет запущена подсистема определения версий служб, которая проведет последовательное тестирование этих портов с целью определения типов и версий служб, за которыми закреплены обнаруженные порты. Файл, названный *nmap-service-probes* используется для определения оптимальных тестов, с помощью которых можно получить максимально точную информацию в данных условиях. Nmap пытается определить протокол службы (ftp, ssh, telnet, http), имя приложения (ISC Bind, Apache httpd, Solaris telnetd), номер версии и дополнительную информацию (версия протокола SSH). Если Nmap откомпилирован с поддержкой OpenSSL, он осуществит подключение к SSL-серверу и попытается определить, какая служба работает "за" шифрованием. Если обнаружена служба RPC, Nmap предпримет атаку на RPC по методу "грубой силы" для определения номера программы RPC и номера версии.

```
[root@DBADOMAIN ~]# nmap 192.168.50.1 -sV
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 03:56 MSK
Interesting ports on 192.168.50.1:
Not shown: 1677 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:50:56:C0:00:01 (VMWare)
Service Info: OS: Windows
Nmap finished: 1 IP address (1 host up) scanned in 20.895 seconds
```

Опция --version-trace для получения полной информации о передаваемых и принимаемых в процессе тестирования данных.

```
[root@DBADOMAIN ~]# nmap 192.168.50.1 --version-trace
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-11-18 04:15 MSK
---------------- Timing report ----------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  scan-delay: TCP 1000, UDP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
-----------------------------------------------
Packet capture filter (device eth0): arp and ether dst host 00:0C:29:5A:C4:60
mass_rdns: Using DNS server 192.168.202.2
mass_rdns: 13.00s 0/1 [#: 1, OK: 0, NX: 0, DR: 0, SF: 0, TR: 3]
```

DNS resolution of 1 IPs took 13.00s. Mode: Async [#: 1, OK: 0, NX: 0, DR: 1, SF: 0, TR: 3, CN: 0]
Packet capture filter (device eth0): dst host 192.168.50.3 and (icmp or (tcp and (src host 192.168.50.1)))
Increased max_successful_tryno for 192.168.50.1 to 1 (packet drop)
Interesting ports on 192.168.50.1:
Not shown: 1677 closed ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
MAC Address: 00:50:56:C0:00:01 (VMWare)
Final times for host: srtt: 929 rttvar: 394 to: 100000
Nmap finished: 1 IP address (1 host up) scanned in 15.042 seconds

2.9. Удаленная идентификация хоста

Одной из наиболее часто употребляемых возможностей, которые предоставляет NMAP, является удаленная идентификация хоста. Nmap, просто осуществляя сканирование по сети, зачастую может сообщить вам какая операционная система выполняется на хосте, а также дать некоторую информацию о номере версии и релиза. Как это делается? Если задан флаг -O, NMAP использует несколько различных приемов для поиска некоторых характерных признаков в TCP/IP-пакетах, возвращаемых хостом. Посылая специально созданные TCP- и UDP-заголовки, NMAP может получить некоторые сведения о том, как удаленный хост осуществляет взаимодействие по протоколу TCP/IP. Если затем проанализировать информацию, то ее можно сравнить с известными признаками, которые хранятся в файле *nmap-os-fingerprints*. Этот файл поддерживается разработчиками NMAP. Если вы сканируете хост с известной операционной системой, используя флаг -O, и NMAP не может ее распознать, то он выводит зашифрованный результат тестирования и сетевой адрес, чтобы можно было послать информацию разработчикам.

Флаг определения ОС также может обеспечить анализ TCP-пакетов с целью определения информации типа общей продолжительности работы системы с момента последней перезагрузки (используя отметку времени в TCP/IP) и предсказуемость последовательного номера. Предсказуемая последовательность номеров упрощает возможность имитации TCP-соединения посредством перехвата пакетов и угадывания последовательности номеров.

[root@DBADOMAIN ~]# **nmap -O 192.168.50.1**
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2006-11-18 04:39 MSK
Interesting ports on 192.168.50.1:
Not shown: 1677 closed ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
MAC Address: 00:50:56:C0:00:01 (VMWare)

Device type: general purpose
Running: Microsoft Windows 2003/.NET|NT/2K/XP
OS details: Microsoft Windows 2003 Server or XP SP2
Nmap finished: 1 IP address (1 host up) scanned in 16.363 seconds

2.10. Idlescan-сканирование с использованием компьютера-зомби

Потрясающей возможностью обладает NMAP – сканирование хоста-жертвы с помощью компьютера-зомби. Этот прием нужно применять очень осторожно, т.к. нужно быть уверенным, что компьютер-зомби не администрирует опытный пользователь, способный обнаружить, что сканирование ведется с помощью его системы. Обычно – это компьютеры, не защищенные ничем и никем.

C:\Nmap 4.11>**nmap 192.168.181.1 -sI 192.168.181.3**
Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2007-01-10 20:40
Idlescan using zombie 192.168.181.3 (192.168.181.3:80); Class: Incremental
All 1680 scanned ports on 192.168.181.1 are closed|filtered
MAC Address: 00:50:56:C0:00:01 (VMWare)
Nmap finished: 1 IP address (1 host up) scanned in 33.500 seconds

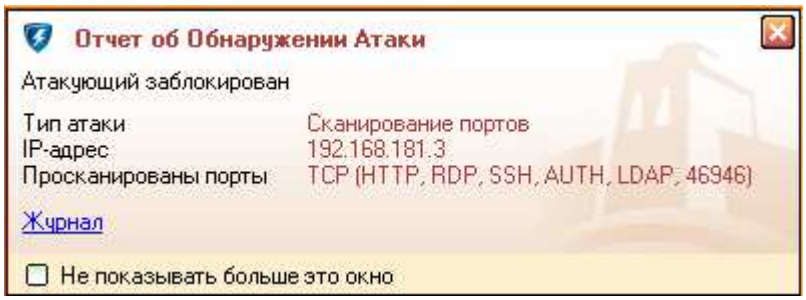

Рис. 10. Отчет при Idle-сканировании с компьютера 192.168.181.2

2.11. Дополнительные опции

1. **-P0 -PT -PS -PI -PB**. Перед тем как сканировать другие порты, протоколы NMAP всегда пытаются осуществить проверку хоста с помощью команды Ping. Такое тестирование не занимает много времени для неработающих хостов. Но многие хосты и брандмауэры блокируют ICMP Ping-трафик, поэтому необходимо иметь возможность управления тем, какую политику тестирования использовать, чтобы определить статус хоста.

-P0 указывает на необходимость отменить использование Ping -использовать только слепое сканирование.

-PT указывает на необходимость использования протокола TCP (с использованием telnet по 80 порту, если у вас нет полномочий суперпользователя или ACK-сканирования по 80 порту, если такие полномочия имеются). Задав число после параметра -PT, можно задать для сканирования номер порта, отличный от 80.

-PS посылает SYN-пакеты (также в случае наличия полномочий суперпользователя).

-PI определяет немедленное проведение тестирования ICMP Ping.

-PB по умолчанию осуществляет попытку тестирования с использованием и ICMP и TCP Ping.

2. **-v -d**. Параметр -V задает расширенный вывод, параметр -d добавляет отладочную информацию. Вы можете использовать оба параметра для получения расширенной и отладочной информации одновременно.

3. **-p ports**. Разумеется, если вы хотите указать конкретные порты для сканирования, то с помощью значения <ports> можно определить единственный порт, список портов, заданный через запятую, или интервал портов, заданный через тире, или использовать любую из возможных комбинаций. Если этот параметр не определен, то осуществляется быстрое сканирование первых 1024 портов.

4. **-e interface**. Для хоста с несколькими сетевыми адресами вы можете определить, какой из сетевых интерфейсов используется для связи. Обычно NMAP самостоятельно обрабатывает такую ситуацию.

5. **-g port**. Позволяет выбрать порт - источник, с которого будет осуществляться сканирование. Обычно эта возможность используется для скрытия сканирования от брандмауэров, которые разрешают входящий трафик от портов TCP/20 (предназначенного для данных ftp), TCP/80 (предназначенного для Web-трафика) или UDP/53 (предназначенного для работы DNS).

6. **-F**. Указывает nmap на необходимость сканирования только «известных» (описанных в файле *nmap-services*) портов. Вез этого параметра nmap сканирует порты с номерами от 1 до 1024 и любые другие порты, которые включены в файл nmap-services (или в файле /etc/services). Если для сканирования протоколов используется параметр -sO, nmap вместо последовательного сканирования всех 256 протоколов использует встроенный файл протоколов (*nmap-protocols*).

Заключение

Автором данной курсовой работы были рассмотрены основные опции программного продукта NMAP, применяемого для сканирования портов хоста, определения их состояния и работающих на этом хосте служб. Комплексное использование рассмотренных методов позволяет не только получить достоверную информацию об открытых портах, но и обойти возможные средства защиты исследуемого сегмента сети.

В руках хакера данное средство сканирования сети является едва ли не основным инструментом для осуществления взлома. NMAP позволяют взломщику до тонкостей разобраться в структуре сети, получить информацию о самой системе, версии ее ядра, и затем применить эксплоиты для известных уязвимостей данной системы.

Системный администратор может воспользоваться данной программой для сканирования своего сегмента сети с целью выявления брешей, которыми может воспользоваться хакер.

При использовании программы автор столкнулся с такой проблемой. При реализации NMAP на платформе Linux можно воспользоваться графическим интерфейсом, а в Windows – нет. Т.к. опций у программы много, при работе в режиме командной строки приходится применять man, что не есть хорошо для обычных пользователей. Держать в памяти все опции в состоянии только администратор, а для пользователя в Windows использовать эту утилиту будет проблематично. Это не должно быть поводом, чтобы пользователи меняли платформу при необходимости использовать возможности данной программы.

Интеллект программы очень продуман. Не всякая система обнаружения вторжений обнаружит сканирование. До недавнего времени не существовало таких брандмауэров, которых NMAP не смог бы обмануть. Теперь вроде бы ситуация нормализовалась и известны все методы сканирования, так что продукты IDS известных фирм с трудом определяют изощренные методы сканирования, которые основаны на дырах в реализации стека TCP/IP.

Список используемых источников

Научная литература

Три автора

К. Дж. Джонс, М. Шема, Б.С. Джонсон. Анти-Хакер. Средства защиты компьютерных сетей. Справочник профессионала. Издательство OSBORNE 2003г.

Описание электронных ресурсов

Электронная статья

Описание основных методов сканирования портов.
http://www.cherepovets-city.ru

Официальный сайт Nmap

http://www.insecure.org

Электронная статья

Описание основных методов сканирования портов.
http://www.bytemag.ru

Приложение 1.

Используемое ПО

Для реализации поставленной задачи мы будем использовать ОС Red Hat Enterprise Linux 4 и ОС Windows XP SP2. В некоторых ситуациях преднамеренно будет включать Outpost Firewall Pro ver. 3.51.748.6419 (462). Для дампа пакетов будем использовать Ethereal ver. 0.10.14.

Для выполнения необходимых задач установим соединение между компьютерами, как показано ниже. Для решения поставленных задач будет использовать две конфигурации.

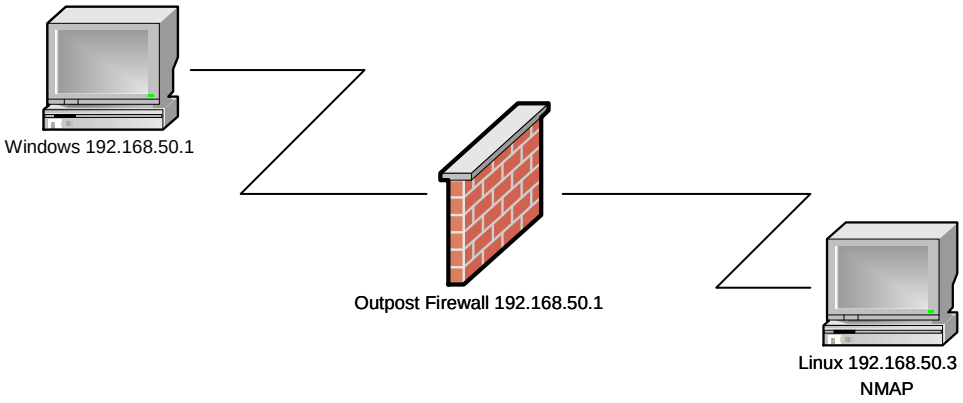

Рис. 11. Соединение компьютеров Linux и Windows

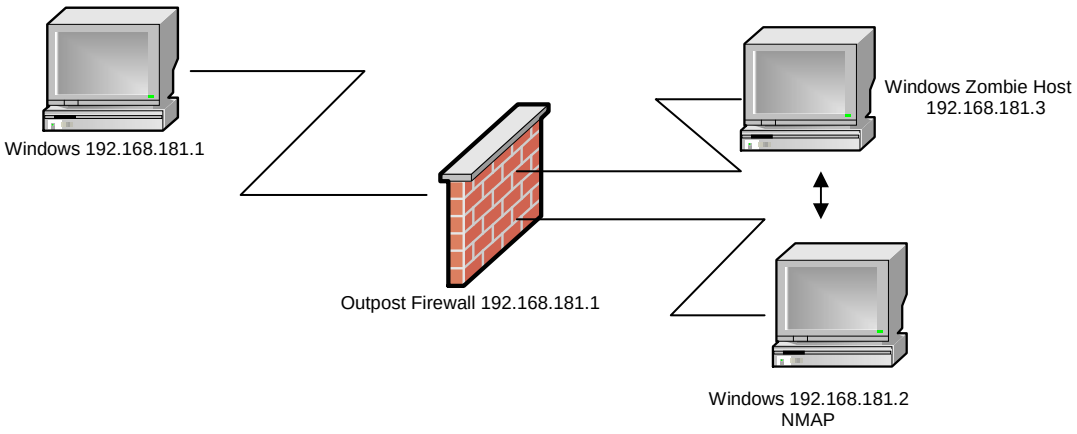

Рис. 12. Соединение компьютеров Windows

Приложение 2.

Графическая часть как дополнение к инструменту NMAP

Для NMAP существует графическая оболочка, позволяющая быстро освоить саму утилиту. Ее интерфейс очень удобен и позволяет даже новичку начать работать со сканером. Вид программы изображен ниже:

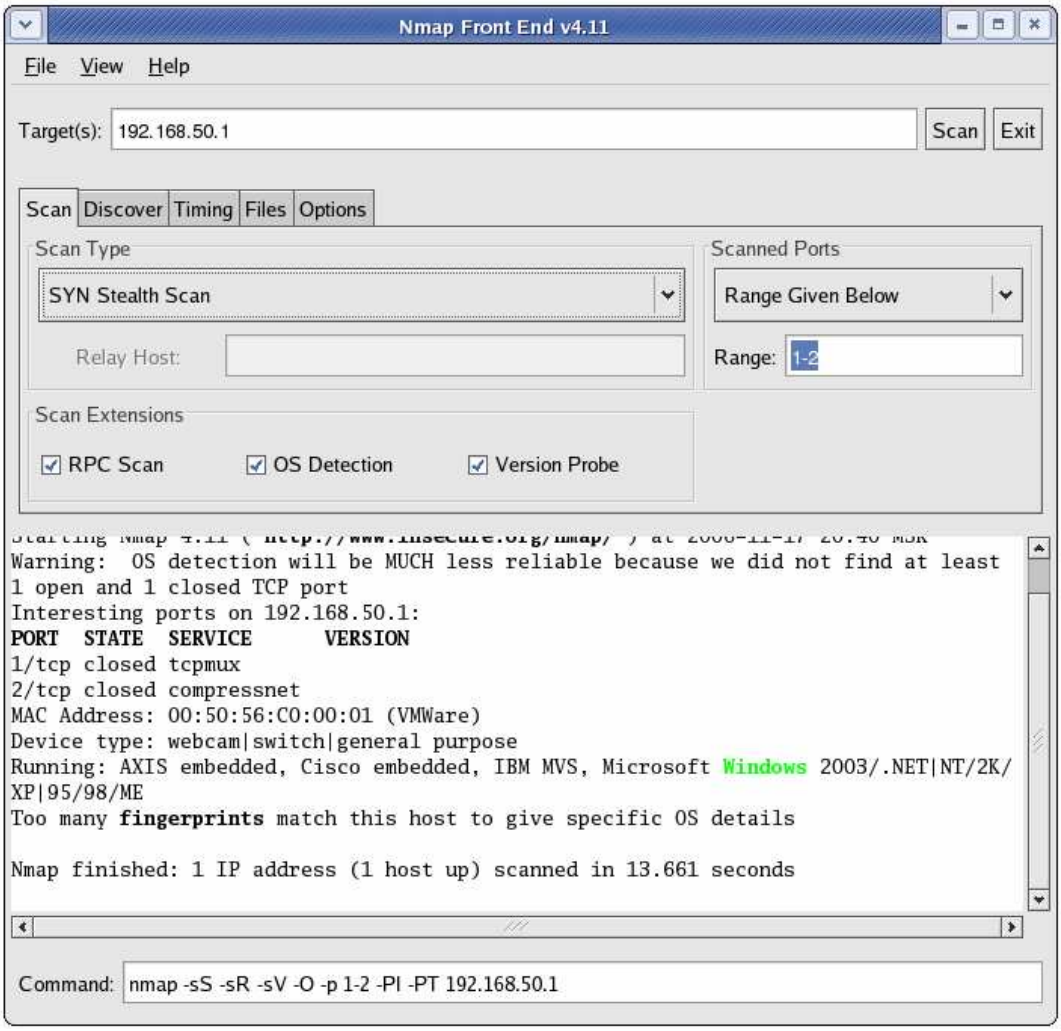

Рис. 13. Nmap Front End v.4.11

Дамп пакетов в Ethereal при различных методах анализа портов

Приведем части отпечатков пакетов, полученные с помощью Ethereal при различных методах сканирования (ответ хоста не рассматривается, брандмауэр жертвы выключен). Сканирование проводилось командами:

nmap <опция> 192.168.181.1 –p 80

– sS (SYN)

```
Internet Protocol, Src: 192.168.181.2 (192.168.181.2), Dst: 192.168.181.1 (192.168.181.1)
Transmission Control Protocol, Src Port: 63484 (63484), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0
    Source port: 63484 (63484)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 24 bytes
    Flags: 0x0002 (SYN)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgment: Not set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 4096
    Checksum: 0xfc97 [correct]
    Options: (4 bytes)
```

– sM (Maimon)

```
Internet Protocol, Src: 192.168.181.2 (192.168.181.2), Dst: 192.168.181.1 (192.168.181.1)
Transmission Control Protocol, Src Port: 63947 (63947), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0
    Source port: 63947 (63947)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Acknowledgement number: 0    (relative ack number)
    Header length: 20 bytes
    Flags: 0x0011 (FIN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...1 = Fin: Set
    Window size: 1024
    Checksum: 0xd86f [correct]
```

– sX (Xmas)

```
Internet Protocol, Src: 192.168.181.2 (192.168.181.2), Dst: 192.168.181.1 (192.168.181.1)
Transmission Control Protocol, Src Port: 42659 (42659), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0
    Source port: 42659 (42659)
    Destination port: http (80)
```

```
    Sequence number: 0    (relative sequence number)
    Header length: 20 bytes
    Flags: 0x0029 (FIN, PSH, URG)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..1. .... = Urgent: Set
        ...0 .... = Acknowledgment: Not set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...1 = Fin: Set
    Window size: 1024
    Checksum: 0xbfbf [correct]
    Urgent pointer: 0
```

– sN (Null)

```
Internet Protocol, Src: 192.168.181.2 (192.168.181.2), Dst: 192.168.181.1 (192.168.181.1)
Transmission Control Protocol, Src Port: 37624 (37624), Dst Port: http (80), Seq: 0, Ack: 0, Len: 0
    Source port: 37624 (37624)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 20 bytes
    Flags: 0x0000 ()
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgment: Not set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 3072
    Checksum: 0xe25e [correct]
```

Приложение 4.

Исследование хоста www.crackmafia.com

Для полноценного исследования какого-либо хоста необходимо в первую очередь сохранить полную анонимность в сети, иначе можно попасть в “черный список флудеров” (flood – непрерывный поток пакетов на хост) или еще что хуже - напороться на злопамятных администраторов. Поэтому необходимо использовать цепь из Socks-серверов. Но это уже тема для отдельного разговора. Поэтому ограничимся хостом www.crackmafia.com.

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -sS www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:54

Note: Host seems down. If it is really up, but blocking our ping probes, try -P0

Nmap finished: 1 IP address (0 hosts up) scanned in 4.953 seconds

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -sU -P0 www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:52

All 1487 scanned ports on cracked.by.the.crackmafia.com (85.17.42.12) are open|filtered

Nmap finished: 1 IP address (1 host up) scanned in 306.907 seconds

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -sA www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:56

Note: Host seems down. If it is really up, but blocking our ping probes, try -P0

Nmap finished: 1 IP address (0 hosts up) scanned in 6.937 seconds

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -P0 www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:44

All 1680 scanned ports on cracked.by.the.crackmafia.com (85.17.42.12) are filtered

Nmap finished: 1 IP address (1 host up) scanned in 345.406 seconds

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -sX -O -P0 www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:49

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port

All 1680 scanned ports on cracked.by.the.crackmafia.com (85.17.42.12) are open|filtered

Too many fingerprints match this host to give specific OS details

Nmap finished: 1 IP address (1 host up) scanned in 390.656 seconds

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -sF -O -P0 www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:41

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port

All 1680 scanned ports on cracked.by.the.crackmafia.com (85.17.42.12) are open|filtered

Too many fingerprints match this host to give specific OS details

Nmap finished: 1 IP address (1 host up) scanned in 397.516 seconds

C:\Documents and Settings\ADMIN>D:\DOC\nmap-4.11**nmap -sN -O -PT www.crackmafia.com**

Starting Nmap 4.11 (http://www.insecure.org/nmap) at 2006-11-25 11:40

Note: Host seems down. If it is really up, but blocking our ping probes, try -P0

Nmap finished: 1 IP address (0 hosts up) scanned in 4.562 seconds