Лабораторная работа
«Утилита для сканирования и исследования безопасности сети NMAP»

Цель работы: изучение и практическое применение утилиты для сканирования и исследования безопасности сети nmap.

Содержание

1 Описание

1.1 Введение

Nmap предназначен для сканирования сетей с любым количеством объектов, определения состояния объектов сканируемой сети, а также портов и соответствующих им служб. Для этого nmap использует много различных методов сканирования, таких, как:

- UDP connect(),
- TCP connect(),
- TCP SYN (полуоткрытое),
- FTP proxy (прорыв через ftp),
- Reverse-ident,
- ICMP (ping),
- FIN-сканирование,
- ACK-сканирование,
- Xmas tree-сканирование,
- SYN-сканирование,
- NULL-сканирование.

Nmap также поддерживает большой набор дополнительных возможностей, а именно:

- определение операционной системы удаленного хоста с использованием TCP/IP fingerprint,
- «невидимое» сканирование,
- динамическое вычисление времени задержки и повтор передачи пакетов,
- параллельное сканирование,
- определение неактивных хостов методом параллельного ping-опроса,
- сканирование с использованием ложных хостов,
- определение наличия пакетных фильтров,
- прямое (без использования portmapper) RPC-сканирование,
- сканирование с использованием IP-фрагментации,
- произвольное указание IP-адресов и номеров портов сканируемых сетей.

Результатом работы Nmap является список отсканированных портов удаленной машины с указанием номера и состояния порта, типа используемого протокола, а также названия службы, закрепленной за этим портом. Порт характеризуется тремя возможными состояниями «открыт», «фильтруемый» и «нефильтруемый»:

- открыт (open)- удаленная машина прослушивает данный порт;

- фильтруемый (filtered) – межсетевой экран, пакетный фильтр или другое устройство блокирует доступ к этому порту и nmap не смог определить его состояние;
- «Нефильтруемый» (closed) - по результатам сканирования nmap воспринял данный порт как закрытый, при этом средства защиты не помешали nmap определить его состояние. Это состояние nmap определяет в любом случае (даже если большинство сканируемых портов хоста фильтруются).

В зависимости от указанных опций, nmap также может определить следующие характеристики сканируемого хоста:
- операционная система хоста,
- метод генерации TCP ISN,
- имя пользователя владельца процесса, зарезервировавшего сканируемый порт,
- символьные имена, соответствующие сканируемым IP-адресам и т.д.

1.2 Методы сканирования

1.2.1 TCP connect()

Наиболее общий метод сканирования TCP портов. Функция connect(), присутствующая в любой ОС, позволяет создать соединение с любым портом удаленной машины. Если указанный в качестве аргумента порт открыт и прослушивается сканируемой машиной, то результат выполнения connect() будет успешным (т.е. соединение будет установлено), в противном случае указанный порт является закрытым, либо доступ к нему заблокирован средствами защиты.

Для того, чтобы использовать данный метод, пользователь может не иметь никаких привилегий на сканирующем хосте. Этот метод сканирования легко обнаруживается целевым (т.е. сканируемым) хостом, поскольку его log-файл будет содержать запротоколированные многочисленные попытки соединения и ошибки выполнения данной операции. Службы, обрабатывающие подключения, немедленно заблокируют доступ адресу, вызвавшему эти ошибки.

1.2.2 TCP SYN

Этот метод часто называют «полуоткрытым» сканированием, поскольку при этом полное TCP-соединение с портом сканируемой машины не устанавливается. Nmap посылает SYN-пакет, как бы намереваясь открыть настоящее соединение, и ожидает ответ.

Наличие флагов SYN|ACK в ответе указывает на то, что порт удаленной машины открыт и прослушивается. Флаг RST в ответе означает обратное. Если nmap принял пакет SYN|ACK, то в ответ немедленно отправляет RST-пакет для сброса еще не установленного соединения (реально эту операцию выполняет сама ОС). Очень немного сайтов способны обнаружить такое сканирование.

Пользователь должен иметь статус root для формирования поддельного SYN-пакета.

1.2.3 «Невидимое» FIN, Xmas Tree и NULL-сканирование

Эти методы используются в случае, если SYN-сканирование по каким-либо причинам оказалось неработоспособным. Так, некоторые межсетевые экраны и пакетные фильтры «ожидают» поддельные SYN-пакеты на защищенные ими порты, и программы типа Synlogger или Courtney способны отследить SYN-сканирование.

Идея заключается в следующем. В FIN-сканировании в качестве запроса используется FIN-пакет. В Xmas Tree используется пакет с набором флагов FIN|URG|PSH, а NULL-сканирование использует пакет без флагов. Согласно рекомендации RFC 973 п. 64, ОС сканируемого хоста должна ответить на такой пакет, прибывший на закрытый порт, пакетом RST, в то время как открытый порт должен игнорировать эти пакеты.

Разработчики Microsoft Windows, как обычно, решили полностью игнорировать все общепринятые стандарты и пойти своим путем. Поэтому любая ОС семейства Windows не посылает в ответ RST-пакет, и данные методы не будут работать с этими ОС. Однако во всем есть свои плюсы, и в nmap этот признак является основным для различения операционных систем, обладающих таким свойством. Если в результате FIN-сканирования вы получили список открытых портов, то это не Windows. Если же все эти методы выдали результат, что все порты закрыты, а SYN-сканирование обнаружило открытые порты, то вы скорей всего имеете дело с ОС Windows. Существует еще несколько ОС, обладающих данным недостатком.

1.2.4 Ping-сканирование

Иногда необходимо лишь узнать адреса активных хостов в сканируемой сети. Nmap может сделать это, послав ICMP-сообщение echo-request на каждый указанный IP-адрес. Хост, отправивший ответ на эхо, является активным. Некоторые сайты (например, microsoft.com) блокируют эхо-пакеты. По этой причине nmap также посылает TCP ACK-пакет на 80-й порт сканируемого хоста (по умолчанию). Если в ответ вы получили RST-пакет, хост активен. Третий метод использует SYN-пакет и ожидает в ответ RST либо SYN|ACK. Для пользователей, не обладающих статусом root, используется метод connect().

Для root-пользователей nmap по умолчанию использует параллельно оба метода - ICMP и ACK. Вы можете изменить это, используя опцию P.

Заметим, что ping-сканирование по умолчанию выполняется в любом случае и только активные хосты подвергаются сканированию.

1.2.5 Определение версий

После того, как определены открытые TCP и/или UDP порты с помощью какого-либо метода сканирования, nmap взаимодействует с этими портами пытаясь определить что именно скрывается за ними. Nmap пытается определить протокол обмена службы (например, ftp, ssh, telnet, http), имя приложения (например, ISC Bind, Apache http, Solaris telnetd), номер версии и иногда различные детали, например, есть ли возможность подключится к X-серверу или номер версии SSH-протокола. Если nmap собран с поддержкой OpenSSL, он создает соединение с SSL-серверами пытаясь определить что скрыто за шифрованным каналом. Если обнаружены RPC-службы, nmap определяет программы, обслуживающие RPC-порты, и их версии. Некоторые UDP порты остаются в состоянии «open|filtered» после UDP-сканирования, если сканирование не смогло определить, является ли порт открытым или фильтруемым. Определение версий пытается получить «ответ» с этих портов (как оно делает и с открытыми портами) и, в случае успеха, меняет их состояние на «открыт».

1.2.6 UDP-сканирование

Этот метод используется для определения, какие UDP порты на сканируемом хосте являются открытыми. На каждый порт сканируемой машины отправляется UDP-пакет без данных. Если в ответ было получено ICMP-сообщение «порт недоступен» (port unreachable), это означает, что порт закрыт. Если на посланный UDP-пакет получен ответ, считается, что сканируемый порт открыт. Если на запрос не получено никаких ответов, то состояние порта будет «opened|filtered», что означает, что порт либо открыт, либо пакетные фильтры блокируют обмен данными. В данном случае определение версий поможет различить действительно открытые порты от фильтруемых.

1.2.7 Сканирование протоколов IP

Данный метод используется для определения IP-протоколов, поддерживаемых сканируемым хостом. Метод заключается в передаче хосту IP-пакетов без какого-либо заголовка для каждого протокола сканируемого хоста. Если получено сообщение «протокол недоступен» (protocol unreachable), то данный протокол хостом не

используется. В противном случае nmap предполагает, что протокол поддерживается хостом.

Некоторые ОС и межсетевые экраны могут блокировать передачу сообщений «протокол недоступен». По этой причине все сканируемые протоколы будут «открыты» (т.е. поддерживаются).

1.2.8 Сканирование «вхолостую»

Позволяет произвести абсолютно невидимое сканирование портов. Атакующий может просканировать цель, не посылая при этом пакетов от своего IP-адреса. Вместо этого используется метод IdleScan, позволяющий просканировать жертву через так называемый хост-«зомби». Кроме абсолютной невидимости, этот тип сканирования позволяет определить политику доверия между машинами на уровне протокола IP. Листинг результатов показывает открытые порты со стороны хоста-«зомби».

Таким образом, можно просканировать цель с использованием нескольких «зомби», которым цель может «доверять», в обход межсетевых экранов и пакетных фильтров. Такого рода информация может быть самой важной при выборе целей «первого удара».

1.2.9 ACK-сканирование

Этот дополнительный метод используется для определения набора правил (ruleset) межсетевого экрана. В частности, он помогает определить, защищен ли сканируемый хост межсетевым экраном или просто пакетным фильтром, блокирующим входящие SYN-пакеты.

В этом методе на сканируемый порт хоста отправляется ACK-пакет (со случайными значениями полей acknowledgement number и sequence number). Если в ответ пришел RST-пакет, порт классифицируется как «нефильтруемый». Если ответа не последовало (или пришло ICMP-сообщение о недоступности порта), порт классифицируется как «фильтруемый». Обращаем ваше внимание, что этот метод никогда не покажет состояние порта «открыт» в результатах сканирования.

1.2.10 TCP Window

Этот метод похож на ACK-сканирование, за исключением того, что иногда с его помощью можно определять открытые порты точно так же, как и фильтруемые/нефильтруемые. Это можно сделать, проверив значение поля Initial Window TCP-пакета, возвращаемого хостом в ответ на посланный ему запрос, ввиду наличия определенных особенностей обработки данного поля у некоторых ОС.

1.2.11 RPC-сканирование

Этот метод используется совместно с другими методами сканирования и позволяет определить программу, которая обслуживает RPC-порт, и номер ее версии. Для этого все открытые TCP/UDP порты хоста затопляются NULL-командами оболочки SunRPC, после чего определяются RPC-порты и закрепленные за ними программы. Таким образом, вы легко получаете информацию, которую могли бы получить с помощью команды 'rpcinfo -p', даже если portmapper сканируемого хоста закрыт межсетевым экраном или TCP-wrapper'ом.

1.2.12 «Прорыв через FTP»

Интересной «возможностью» протокола является поддержка «доверенных» (proxy) ftp-соединений. Другими словами, с доверенного хоста source.com можно соединиться с FTP-сервером target.com и отправить файл, находящийся на нем, на любой адрес Internet. Nmap использует эту возможность для сканирования портов с «доверенного» FTP-сервера. Итак, вы можете подключиться к FTP-серверу «над» межсетевым экраном и затем просканировать заблокированные им порты (например, 139-й). Если ftp-сервер позволяет читать и записывать данные в какой-либо каталог (например, /incoming), вы также можете отправить любые данные на эти порты.

1.3 Определение операционной системы удаленного хоста

Проблема определения типа и версии операционной системы удаленного хоста является весьма актуальной на начальном этапе реализации атаки на хост. В зависимости от того, какая ОС установлена на удаленном хосте, атакующий будет планировать свои дальнейшие действия, воздействуя на известную «дыру» (если таковая имеется) в безопасности установленной на хосте ОС. При этом, чем точнее атакующий определит тип и версию ОС удаленного хоста, тем эффективней будет выполнен его «взлом». В подтверждении этого, рассмотрим несколько возможных ситуаций.

Допустим, осуществляется попытка проникновения на удаленный хост. В результате сканирования портов было обнаружено, что 53-й порт хоста открыт. На основании данного признака можно предположить, что на хосте установлена одна из версий ОС UNIX, и выполняется одна из уязвимых версий демона bind. Если это весьма условное предположение является верным, атакующий имеет только одну попытку использовать обнаруженную «дыру», поскольку неудавшаяся попытка атаки «подвесит» демона и порт окажется закрытым, после чего атакующему придется искать новые «дыры» в безопасности удаленного хоста.

Если атакующий точно определит тип и версию ОС удаленного хоста (например, Linux kernel 2.0.35 или Solaris 2.51), он может соответствующим образом скоординировать свои действия, проанализировав информацию, касающуюся известных проблем в безопасности определенной ОС.

Используя программные средства, обеспечивающие определение ОС удаленного хоста, атакующий способен просканировать множество хостов и определить тип и версию ОС, установленную на каждом из них. Затем, когда кто-нибудь опубликует в сети Интернет информацию об обнаруженной «дыре» в безопасности конкретной ОС, атакующий автоматически получает список уязвимых хостов, на которых установлена данная версия ОС.

1.3.1 Метод опроса стека TCP/IP удаленного хоста

Как правило, реакцией сервера на любое удаленное воздействие (входящий пакет данных, запрос) является пакет данных, посылаемый источнику данного воздействия (в дальнейшем под термином «сервер» понимается атакуемый хост, а под термином «хост» - хост атакующего).

Как показывает практика, различные ОС при работе в сети по-разному реагируют на один и тот же запрос. Исследовав особенности реакций на запрос ОС, версии которых заранее известны, можно набрать определенную статистику, сопоставив реакции на запрос с типом ОС. При использовании комбинированного воздействия, статистическая информация становится более конкретизированной.

В дальнейшем, исследуя реакцию сервера с неизвестной ОС, с использованием накопленной статистики можно определить не только тип, но и версию установленной на сервере ОС. Например, возможно точно отличить Solaris 2.4 от Solaris 2.50 или Linux kernel version 2.0.30 (для всех Linux далее указывается версия ядра) от Linux 2.0.35.

Рассмотрим более подробно основные методы исследования ОС сервера.

1.3.1.1 FIN-исследование

Перед началом непосредственного исследования хост сканирует порты сервера и определяет, какие порты являются открытыми. Затем на любой открытый порт сервера хост посылает FIN-пакет (TCP пакет на завершение соединения) или любой другой пакет без флагов SYN и ACK. В соответствии с RFC 793 сервер должен ответить на такой пакет

RST-пакетом, однако некоторые ОС типа Windows, BSDI, CISCO, HP/UX, MVS и IRIX не посылают ничего в ответ.

1.3.1.2 Исследование BOGUS- флагом

Хост посылает на сервер SYN-пакет с установленным в TCP-заголовке неиспользуемым «флагом» BOGUS. «Флаг» BOGUS не является настоящим флагом. На самом деле этот термин подразумевает установку бит в поле Reserved заголовка TCP-пакета как 1000000 (вместо всех нулей согласно RFC 793). ОС Linux до 2.0.35 сохраняет в ответе этот «флаг». Некоторые ОС обрывают соединение при получении такого пакета.

1.3.1.3 Определение закона изменения ISN сервера

Хост посылает на сервер SYN-пакет с запросом на соединение, записав в пакет свое (известное) значение ISN. Сервер, получив запрос на соединение, прибавляет к полученному ISN единицу и записывает полученное значение в поле ACK ответа (т.е. SYN|ACK-пакета), а в поле ISS ответа - свой собственный ISN, и передает пакет устанавливающему соединение хосту. На приемной стороне (т.е. на стороне хоста) пакет анализируется. Операция повторяется до тех пор, пока не будет выявлен закон изменения ISN сервера.

Возможны следующие закономерности:

- закон «постоянного приращения» (традиционный закон «+64» - старые версии UNIX): значение ISN сервера, записываемого в поле ISS ответа на запрос «установление соединения», увеличивается на постоянную величину (либо на 64) с каждым обрабатываемым запросом;
- закон «случайных приращений» (новые версии Solaris, IRIX, FreeBSD, DigitalUNIX, Cray): приращение ISN носит случайный характер;
- истинно случайные значения (Linux 2.0.x, OpenVMS, новые AIX): значение ISN является случайной величиной;
- закон «время-зависимых приращений» (Windows): значение ISN периодически во времени увеличивается на некоторую небольшую величину;
- постоянный (концентраторы 3Com [ISN=0x803], принтеры Apple LaserWriter [0xC7001]): значение ISN остается постоянным.

1.3.1.4 Исследование поля Window TCP-пакета

Анализируя принятые от сервера TCP-пакеты целесообразно обратить внимание на поле Window в их заголовках, поскольку значение этого поля является своеобразной константой, характеризующей ОС. В некоторых случаях для однозначного определения типа ОС достаточно извлечь значение поля Window в TCP-заголовке принятого от сервера пакета.

Так, ОС AIX - единственная ОС, имеющая значение Window=0x3F25. «Полностью переписанный» стек TCP/IP в ОС Windows 2000, равно, как и OpenBSD и FreeBSD, имеют Window=0x402E.

1.3.1.4 Исследование поля ACK в TCP-пакете

Рекомендацией RFC 793 определено стандартное изменение поля ACK в TCP-пакетах при установлении соединения, передаче данных и закрытии соединения. Однако в нестандартных ситуациях различные ОС по-разному устанавливают значение этого поля. Исследование проводится следующим образом. На закрытый TCP порт хост отправляет FIN|PSH|URG-пакет с известным значением ISN в поле ISS. Большинство ОС скопируют значение ISN, прибывшее в ISS, в поле ACK ответа. Однако ОС Windows и некоторые сетевые принтеры отправят в поле ACK ответа ISN+1. Если хост пошлет

SYN|FIN|PSH|URG-пакет, поведение Windows предсказать трудно. Иногда эта ОС отправляет в поле ACK ответа прибывший ISN, иногда - ISN+1, иногда - по всей видимости, случайное значение. Остается только догадываться, какой код написала Microsoft для обработки подобной ситуации.

1.3.1.5 Исследование скорости генерирования ICMP-сообщений

Согласно RFC 792, протокол ICMP использует протокол IP в качестве средства доставки. Очевидно, что ICMP-сообщения занимают определенную часть полосы пропускания канала связи, что снижает общую скорость передачи данных. По этой причине некоторые продвинутые ОС, следуя рекомендации RFC 1812, ограничивают количество отправляемых в канал связи ICMP-сообщений об ошибках.

Так, Linux ограничивает количество ICMP-сообщений об ошибке типа «получатель недоступен» (destination unreachable) до 80 сообщений в 4 секунды, с простоем 0,25 секунды, если это ограничение было превышено.

Единственный способ проверить скорость генерирования ICMP-сообщений сервером - послать на некоторый закрытый UDP-порт с большим номером набор пакетов и подсчитать количество принятых ICMP-сообщений. Этот тест является очень медленным, и, кроме того, вызывает относительно большую нагрузку на сеть.

1.3.1.6 Исследование формата ICMP-сообщений

Для снижения общей нагрузки на сеть рекомендациями было установлено, что дейтаграмма с ICMP-сообщением об ошибке должна иметь меньший размер, чем дейтаграмма с ICMP-сообщением, вызвавшая ошибку. Так, в качестве ICMP-сообщения «порт недостижим» (port unreachable) практически все ОС генерируют дейтаграмму, представляющую собой необходимый IP-заголовок и 8 байт данных, которые и являются непосредственно ICMP-сообщением. Однако ОС Solaris формирует ICMP-сообщение немного большего размера, а Linux - еще больше, чем Solaris. Таким образом, имеется возможность распознавать ОС Linux и Solaris даже в том случае, если сервер не осуществляет прослушивание портов.

1.3.1.7 Исследование эха в ICMP-сообщениях

Как известно, в ICMP-сообщении об ошибке должна присутствовать часть дейтаграммы, вызвавшей эту ошибку. Эта часть состоит из IP-заголовка дейтаграммы и первых 64-х бит данных дейтаграммы, и называется «эхом» дейтаграммы.

Некоторые ОС используют IP-заголовок прибывшей дейтаграммы в качестве «рабочего пространства» на начальном этапе ее обработки. Это приводит к искажению IP-заголовка, и дейтаграмма с искаженным заголовком отправляется как эхо ICMP-сообщения. Различные ОС по-разному искажают заголовок дейтаграммы. Например, ОС AIX и BSDI в IP-заголовке эха возвращают значение поля TotalLength на 20 байт больше первоначального значения исходной дейтаграммы. Некоторые версии ОС BSDI, FreeBSD, OpenBSD, ULTRIX и VAXen стирают поле ID в эхо-IP заголовке.

Не смотря на то, что поле «контрольная сумма» IP-заголовка так или иначе изменяется в связи с изменением параметра TTL, некоторые ОС типа AIX, FreeBSD отправляют в эхо-дейтаграмме несоответствующую либо нулевую контрольную сумму. То же самое происходит и с контрольной суммой в UDP-пакете.

В целом, возможно проведение девяти различных тестов для проверки эха дейтаграммы в ICMP-сообщении и выявления различных закономерностей для разных ОС.

1.3.1.8 Исследование поля Type Of Service в заголовке ICMP-сообщения

В ICMP-сообщении, наряду с уже упомянутыми признаками, можно проанализировать поле Type Of Service (Тип сервиса, TOS). Подавляющее большинство ОС устанавливают поле TOS=0. Однако старые версии Linux ставят TOS=0xC0 (следует отметить, что это значение не является указателем на какой-либо тип сервиса). Таким образом, данный признак можно использовать совместно с другими тестами для различения старых и новых версий Linux.

1.3.1.9 Исследование обработки фрагментов дейтаграммы

Как известно, протокол IP делит пакет на фрагменты для дальнейшей передачи их в сеть. На практике различные ОС могут по-разному осуществлять обработку перекрытия фрагментов. Так, некоторые ОС заменяют старый фрагмент, прибывший без ошибок, повторно прибывшим аналогичным. Другие ОС считают, что старый пакет имеет привилегию над аналогичным новым и игнорируют его. Исследуя закон перекрытия фрагментов можно сделать определенные выводы относительно типа ОС исследуемого сервера.

1.3.1.10 Исследование поля Options заголовка TCP-пакета

Поле Options (опции) TCP пакета является едва ли не самым важным каналом утечки информации от хоста относительно ОС, установленной на нем. Денное поле имеет некоторые особенности:

- опции TCP-протокола не являются обязательными, и не все ОС поддерживают их;
- узнать, поддерживает ли ОС опции TCP можно, послав на сервер запрос с указанием в соответствующем поле TCP-заголовка некоторый набор опций (а лучше всего - полный набор). Сервер укажет на поддержку определенных опций, установив соотв. значение в поле Options TCP-заголовка ответа и сбросит все остальные.

Некоторые ОС, например, новые версии FreeBSD и последние версии Linux 2.1.x , поддерживают все опции, другие (например, Linux 2.0.x) - лишь небольшой набор опций.

1.3.1.11 Исследование флага DontFragment в IP-заголовке

Многие ОС в определенных ситуациях не используют фрагментацию пакетов, и поэтому устанавливают флаг DontFragment (DF) в IP-заголовке отправляемой (нефрагментированной) дейтаграммы. Это повышает производительность ОС при работе в сети в связи с уменьшением времени обработки передаваемых пакетов. Установив зависимость наличия (или отсутствия) данного признака в конкретной ситуации от типа ОС, можно в дальнейшем использовать его в одном из тестов на определение ОС сервера.

1.3.1.12 Исследование возможности «борьбы с затоплением» SYN-пакетами

«Затопление» SYN-пакетами является достаточно известным способом «завала» сервера, вызвав у него состояние «отказа в обслуживании». Суть этой атаки заключается в том, что при отправлении на некоторый открытый порт сервера определенного числа SYN-пакетов (запрос на установление соединения) с указанием несуществующего IP-адреса сервер перестает отвечать на все входящие на этот порт запросы.

Большинство ОС могут успешно обработать не более 7 таких пакетов. Однако некоторые новые версии ОС (например, новые Linux) способны бороться с «затоплением» SYN-пакетами различными методами (например, SYN-cookies) для предотвращения «отказа в обслуживании».

Поэтому имеется возможность исследовать ОС сервера, послав на него 8 SYN-пакетов с указанием несуществующего IP-адреса в поле «источник» IP-заголовка (указание ложного

источника позволяет избежать разрыва соединения между сервером и хостом) и затем проверить возможность установления соединения с сервером по тому порту, на который были отправлены SYN-пакеты.

1.3.1.13 Особенности ОС Windows

Несмотря на все выше перечисленные методы определения ОС сервера, практически невозможно различить стек TCP/IP у ОС Windows 95, Windows 98 и Windows NT всех версий, несмотря на то, что Windows 98 вышла позже Windows 95 на 4 года. Этот факт позволяет сделать вывод о том, что стек TCP/IP, положенный в основу Windows 95, был скопирован в Windows NT 4.0 и, может быть, слегка изменен в Windows 98.

Поэтому атакующему, определив ОС сервера как Windows95/NT, достаточно опробовать известные методы атаки на эти ОС (Ping of Death, WinNuke, Teardrop, Land). Тем самым, работа сервера так или иначе будет нарушена.

1.3.2 Реализация метода комплексного опроса стека TCP/IP в NMAP

Рассмотрим реализацию метода исследования ОС удаленного хоста путем комплексного опроса его TCP/IP стека, называемым иначе методом снятия «отпечатков» (fingerprint) стека TCP/IP и используемым в сканере NMAP.

Для определения ОС удаленного хоста, версия которой неизвестна, необходимо иметь определенную информацию о том, как ОС известных версий реагируют на определенные виды запросов, описанных выше, иначе говоря - составить «отпечаток» стека TCP/IP операционной системы.

Для этого необходимо удаленный либо локальный хост, тип и версия ОС которого заранее известны, протестировать всеми описанными выше способами, проанализировать результаты тестов и на основе полученных данных составить общую характеристику (или т.н. «отпечаток») стека TCP/IP удаленного хоста, привязав его к конкретному типу и версии ОС.

Собрав достаточно большое количество таких отпечатков (хотя можно начинать и с одним), возможно теми же методами исследовать хост, тип и версия ОС которого заранее неизвестна. Составив из полученных результатов отпечаток и сопоставив его с уже имеющимися, можно определить, какой ОС соответствует полученный отпечаток и на основании этого сделать вывод об ОС исследуемого хоста.

Алгоритм получения отпечатка стека TCP/IP следующий. Вначале проводится сканирование портов удаленного хоста с целью определения открытых портов и служб, функционирующих на исследуемом хосте. Затем проводится несколько тестов, поэтапно выполняющих опрос стека TCP/IP удаленного хоста с целью выявления рассмотренных выше признаков.

На основе полученных от хоста ответов составляется отпечаток, который затем сравнивается с уже имеющейся базой отпечатков, и принимается решение о типе и версии ОС исследуемого хоста.

Заметим, что нет никакой разницы между алгоритмом получения отпечатка для хоста с известной ОС и для хоста с ОС, версия которой неизвестна. Вот пример одного из таких отпечатков, полученного для ОС IRIX версии 6.2 - 6.4.

```
FingerPrint IRIX 6.2 - 6.4
Tseq(Class=i800)
T1(DF=N%W=C000|EF2A%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
```

T3(Resp=Y%DF=N%W=C000|EF2A%ACK=0%Flags=A%Ops=NNT)
T4(DF=N%W=0%ACK=0%Flags=R%Ops=)
T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=0%Flags=R%Ops=)
T7(DF=N%W=0%ACK=S%Flags=AR%Ops=)
PU(DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)

Для получения отпечатка было проведено 9 тестов. Далее подробно рассмотрен каждый из них.

1. Tseq (Class = i800) - тест определения закона изменения ISN хоста.

Указатель Tseq определяет закон изменения ISN сервера. Описание закона изменения ISN хранится в переменной Class (здесь и далее значения параметров указаны для ОС IRIX; для остальных ОС параметры те же, отличаются лишь значения). Для ОС IRIX закон изменения ISN описан как i800 (Increment 800). Это означает, что каждое последующее значение ISN на 800 больше, чем предыдущее.

2. T1(DF=N%W=C000|EF2A%ACK=S++%Flags=AS%Ops=MNWNNT) - тест определения TCP-опций.

В данном тесте на открытый порт сервера хост посылает SYN-пакет с набором TCP-опций. В скобках записаны параметры, возвращаемые в ответе на посланный SYN-пакет:

DF = N - состояние флага DontFragment в IP-заголовке ответа (N, т.е. 0)

W = C000|EF2A - значение поля Window в TCP-заголовке ответа (C000 либо EF2A)

ACK = S++ - значение поля ACK в TCP-заголовке ответа (S++, т.е. посланный хостом ISN+1)

Flags = AS - состояние флагов в TCP-заголовке ответа (должны быть установлены флаги ACK (A) и SYN (S))

Ops = MNWNNT - набор TCP-опций (учитывается наличие опций и порядок их следования), указанных в TCP-заголовке ответа

3. T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=) - тест обработки NULL-пакета.

На открытый порт сервера хост отправляет «пустой» пакет с указанием TCP-опций, аналогичных предыдущему тесту.

Resp = Y - указатель, определяющий наличие или отсутствие ответа от сервера на подобный запрос. Данный указатель используется в том случае, когда конкретная ОС, для которой составлен отпечаток, может не ответить на запрос, используемый в тесте, тогда как другие ОС отвечают на подобный запрос (это и устанавливается указателем Resp=Y/N). В случае, если Resp не присутствует среди переменных, подразумевается, что на подобный запрос любая ОС пошлет ответ.

Ops = - набор TCP-опций в ответе на запрос. «Пустое» значение данной переменной означает отсутствие в ответе каких-либо опций.

4. T3(Resp=Y%DF=N%W=C000|EF2A%ACK=0%Flags=A%Ops=NNT) - тест обработки SYN|FIN|PSH|URG-пакета.

На открытый порт сервера хост посылает пакет с указанием соотв. набора флагов и без

указания TCP-опций. Расшифровка ожидаемого ответа следующая: ответ на запрос должен быть получен, флаг DontFragment сброшен, поле Window=0, значение поля ACK содержит посланный хостом в запросе ISN, набор флагов - ACK и RST, TCP-опции в ответе должны отсутствовать.

5. T4(DF=N%W=0%ACK=0%Flags=R%Ops=) - тест обработки ACK-пакета.

На открытый порт сервера хост отправляет ACK-пакет (здесь и далее расшифровка результатов по аналогии с предыдущими тестами, за исключением переменных, смысл которых объяснен по тексту).

6. T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=) - тест обработки SYN-пакета.

На закрытый порт сервера хост отправляет SYN-пакет.

7. T6(DF=N%W=0%ACK=0%Flags=R%Ops=)- тест обработки ACK-пакета на закрытый порт.

На закрытый порт сервера хост отправляет ACK-пакет.

8. T7(DF=N%W=0%ACK=S%Flags=AR%Ops=) - тест обработки FIN|PSH|URG-пакета.

На закрытый порт сервера хост отправляет соответствующий пакет.

9. PU(DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134 %DAT=E) - тест формата ICMP-сообщения Port Unreachable.

На закрытый порт сервера с большим номером хост отправляет запрос (TCP и UDP-пакет), и анализируется прибывшее в ответ ICMP-сообщение Port Unreachable (порт недоступен).

DF = N - состояние флага DontFragment в IP-заголовке ICMP-сообщения

TOS = 0 - значение поля TypeOfService в прибывшем ICMP-сообщении (равно 0)

IPLEN = 38 - шестнадцатиричное значение поля TotalLength в IP-заголовке прибывшего ICMP-сообщения (составляет 38h)

RIPTL = 148 - значение поля TotalLength в IP-заголовке эха ICMP-сообщения (составляет 148h)

RID = E - проверка значения поля ID в IP-заголовке эха ICMP-сообщения (E-совпадает с посланным значением, F-не совпадает)

RIPCK = E - проверка значения поля CheckSum в IP-заголовке эха ICMP-сообщения (E-совпадает с посланным значением, F-не совпадает)

UCK = E - проверка значения поля CheckSum в UDP-заголовке (при отправлении на сервер UDP-пакета) UDP-эха ICMP-сообщения (E-совпадает с посланным значением, F-не совпадает)

ULEN = 134 - проверка значения поля CheckSum в UDP-заголовке UDP-эха ICMP-сообщения (составляет 134h)

DAT = E - проверка данных UDP-эха ICMP-сообщения (E-совпадает с посланным значением, F-не совпадает). В общем случае, совокупность значений переменных UCK=E, ULEN=0x134h (для IRIX) и DAT=E означает, что данные эхо-UDP были приняты

верно. Поскольку большинство ОС не отправляют посланные в UDP-пакете данные в качестве эха, решение о его «верности» принимается ни основании значений UCK и ULEN, а в поле DAT устанавливается значение по умолчанию (т.е. DAT = E).

2 Использование

nmap [Метод(ы) сканирования] [Опции] <Хост или сеть #1,[#N]>

2.1 Опции (частично)
2.1.1 Опции выбора метода сканирования
-sT TCP Connect()

-sS TCP SYN

-sF FIN-сканирование

-sX Xmas Tree сканирование

-sN NULL-сканирование

-sP Ping-сканирование

-sV Определение версий

-sU UDP-сканирование

-sO Сканирование протоколов IP

-sI - Сканирование «вхолостую»

-sA ACK-сканирование

-sW TCP Window

-sR RPC-сканирование

-b - «Прорыв через FTP». В качестве аргумента передаетсяURL ftp-сервера, используемого в качестве «доверенного» (имя_пользователя:пароль@сервер:порт)

2.1.2 Некоторые опции настройки и выбора дополнительных возможностей
-P0 Не производить ping-опрос хостов перед их непосредственным сканированием. Эта опция позволяет просканировать сети, блокирующие обработку ICMP-эха с помощью межсетевых экранов. Примером такой сети является microsoft.com.

-O Эта опция позволяет определить операционную систему сканируемого хоста с помощью метода TCP/IP fingerprint. Другими словами, nmap активизирует мощный алгоритм, функционирующий на основе анализа свойств сетевого программного обеспечения установленной на нем ОС. В результате сканирования получается формализованный «отпечаток», состоящий из стандартных тестовых запросов и «ответов» хоста на них. Затем полученный отпечаток сравнивается с имеющейся базой стандартных ответов известных ОС, и на основании этого принимается решение о типе и версии ОС сканируемого хоста. Этот метод требует наличия хотя бы одного закрытого и одного открытого порта на целевом хосте.

-p <диапазон(ы)_портов>
Эта опция указывает nmap, какие порты необходимо просканировать. Например, '-p 23' означает сканирование 23 порта на целевой машине. Если указано выражение типа '-p

20-30,139,60000-' Nmap будет сканировать порты с номерами с 20 по 30 включительно, 139 и от 60000 и выше (до 65535). По умолчанию nmap сканирует все порты в диапазоне 1-1024

2.2 Способы задания целевого хоста
Все, что не является опцией или ее аргументом, nmap воспринимает как адрес или имя целевого хоста (т.е. хоста, подвергаемого сканированию). Простейший способ задать сканируемый хост - указать его имя или адрес в командной строке после указания опций и аргументов. Если вы хотите просканировать подсеть IP-адресов, вам необходимо указать параметр '/mask' («маска») после имени или IP-адреса сканируемого хоста.

Nmap позволяет также гибко указать целевые IP-адреса, используя списки и диапазоны для каждого их элемента. Например, необходимо просканировать подсеть класса B с адресом 128.210.*.*. Задать эту сеть можно любым из следующих способов: 128.210.*.*, 128.210.0-255.0-255, 128.210.1-50,51-255.1,2,3,4,5-255,128.210.0.0/16

Приведем еще один пример. Если вы указали в качестве целевого IP-адреса строку '*.*.5.6-7', nmap отсканирует все IP-адреса, оканчивающиеся на 5.6 либо 5.7.

3 Рабочее задание
1. Получите список открытых портов (TCP и UDP) машины, за которой выполняется лабораторная работа (localhost). Воспользуйтесь несколькими методами сканирования. Сравните результаты.
2. Определите операционную систему этой же машины.
3. Узнайте адреса активных хостов в сети лаборатории (без сканирования портов).
4. Просканируйте хост №1, разными методами. Сравните результаты.
5. Для четырех хостов (№2-5), выполните следующие действия:
    1. Определите, защищен ли хост межсетевым экраном.
    2. Определите поддерживаемые протоколы.
    3. Выберите оптимальный метод сканирования портов, чтобы избежать обнаружения. Определите открытые и фильтрованные порты и по возможности версии служб. При необходимости используйте RPC-сканирование.
    4. Определите операционную систему.

4 Форма отчета по лабораторной работе
В отчете должны содержаться параметры, с которыми вызывался nmap, полученные результаты и пояснения по каждому пункту задания.

Пример отчета:
1.1 Сканирование localhost методом TCP SYN
Команда «nmap –sS localhost»
Результаты:
Starting nmap 3.70 (http://www.insecure.org/nmap/) at 2005-02-10 13:05 MSK
Interesting ports on localhost (127.0.0.1):
(The 1658 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
Nmap run completed -- 1 IP address (1 host up) scanned in 0.264 seconds
1.2 Сканирование localhost методом ACK
Команда «nmap –sA localhost»
Результаты:
Starting nmap 3.70 (http://www.insecure.org/nmap/) at 2005-02-10 13:07 MSK

All 1660 scanned ports on localhost (127.0.0.1) are: UNfiltered

Nmap run completed -- 1 IP address (1 host up) scanned in 0.263 seconds
и т.д.

5 Список рекомендуемой литературы

1. Nmap Documentation - Free Security Scanner For Network Exploration & Security Audits. – http://www.insecure.org/nmap/nmap_documentation.html