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

Сканер портов Nmap

http://www.insecure.org

http://cherepovets-city.ru/insecure/runmap/

Алексей Волков. Определение операционной системы удаленного хоста http://www.insecure.org/nmap/nmap-fingerprinting-article-ru.html

Программа **nmap** относится к числу сканеров портов и сканеров безопасности систем.

Программа позволяет администраторам сканировать отдельные хосты и целые сети, определяя поддерживаемые типы сервиса и другие параметры. Nmap поддерживает множество методов сканирования - UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), Reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep, IP Protocol, Null scan. Более подробное описание этих методов приводится ниже. Кроме обычного сканирования программа **nmap** может определять тип операционной системы удаленного хоста, выполнять скрытое сканирование, параллельное сканирование, детектирование фильтров, прямое сканирование RPC (без portmapper), сканирование с использованием фрагментов и др.

Для выполнения большинства операций nmap требуются полномочия пользователя **root**, поскольку многие интерфейсы ядра (в частности, сокеты **raw**) требуют привилегий **root**. При запуске **nmap** от имени обычного пользователя значительная часть функций программы теряется.

По результатам работы программа **nmap** генерирует отчет, содержащий сведения об интересных портах просканированных хостов, если таковые были обнаружены. Для “хорошо известных” портов **nmap** всегда указывает имя сервиса, номер порта, его состояние и протокол. Состояние порта может быть **open** (открыт), **filtered** (фильтруется) или **unfiltered** (не фильтруется). Порт считается открытым если хост принимает адресованные в этот порт соединения. К фильтруемым относятся порты, которые активны, но доступ к ним заблокирован межсетевым экраном, пакетным фильтром или иными системами контроля трафика, которые не позволили программе **nmap** организовать соединение с портом. К нефильтруемым портам относятся те, которые программа **nmap** определила как закрытые, не встретив при этом брандмауэра или иного средства предотвращения доступа к портам. Это состояние является обычным для большинства портов, поэтому они указываются в отчете лишь в тех случаях, когда большинство просканированных портов оказались фильтруемыми.

В зависимости от заданных опций **nmap** может также определять ряд характеристик удаленного хоста – операционную систему, порядковые номера TCP, имена пользователей, которые работают с программами, привязанными к портам, доменное имя DNS и другие параметры.

Синтаксис

```
nmap [<тип сканирования>] [<опции>] <хост или сеть #1 ... [#N]>
```

Опции

В силу широких возможностей программы число опций командной строки, управляющих режимом и параметрами сканирования весьма велико. Программа **nmap** проверяет заданный в командной строке набор опций и при наличии в них ошибок или противоречий выдает пользователю предупреждение. Список опций с краткими комментариями можно получить по команде **nmap -h**, более подробное описание вы получите с помощью команды **man nmap**.

Ниже приводятся описания поддерживаемых программой опций, объединенных в группы по их назначению.

Тип сканирования

В последующих параграфах описаны опции выбора типа сканирования или дополнительных операций, выполняемых программой **nmap**. Опции активизации этих методов указаны в скобках после названия метода.

Сканирование TCP SYN (-sS)

Этот метод часто называют сканированием с использованием полуоткрытых (half-open) соединений, поскольку при сканировании полные соединения TCP не организуются. Сканирующий хост передает пакет SYN как при обычной организации соединения и ожидает отклика. Полученный в ответ пакет **SYN ACK** говорит о том, что порт прослушивает входящие соединения, пакет **RST** показывает, что порт не прослушивается. При получении отклика **SYN ACK** незамедлительно передается пакет **RST** для сброса запрошенного соединения.

Основным преимуществом данного метода сканирования является то, что большинство сайтов не сохраняют записей о нем в своих журнальных файлах. Однако для использования метода требуются привилегии пользователя root, чтобы создавать пакеты SYN с нужными параметрами. Этот метод сканирования применяется по умолчанию для привилегированного пользователя.

Сканирование TCP connect (-sT)

Это один из основных методов сканирования TCP. Для организации соединения с каждым проверяемым портом служит системный вызов **connect()**. Если порт находится в состоянии **listening**, connect() возвращает позитивный результат, в противном случае функция сообщает о недоступности порта. Преимуществом этого метода является то, что он не требует каких-либо специальных привилегий для пользователя, поскольку вызов функции **connect** на большинстве систем UNIX доступен любому пользователю. Данный метод применяется по умолчанию для пользователей, не имеющих привилегий.

Сканирование с использованием этого метода легко обнаружить, поскольку проверяемый хост будет фиксировать в журнальных файлах многочисленные вызовы и сообщения об ошибках при обращении к закрытым портам.

Скрытое сканирование Stealth FIN, Stealth Xmas Tree, Stealth Null (-sF -sX -sN)

В ряде случаях сканирование **TCP SYN** не обеспечивает скрытности. Некоторые брандмауэры и системы фильтрации пакетов следят за пакетами SYN, направленными в закрытые порты, а программы типа **PortSentry** и **Courtney** способны детектировать сканирование **TCP SYN**. Эти методы сканирования достаточно эффективны и практически не оставляют следов.

Идея этих методов состоит в том, что при обращении к закрытым портам вы должны получить отклик **RST**, а открытые порты должны игнорировать такие пакеты в соответствии со стандартом (RFC 793, стр. 64[1]). При сканировании **Stealth FIN** в качестве зондов передаются пакеты с флагом **FIN**, метод **Stealth Xmas tree** использует пакеты с флагами **FIN**, **URG** и **PUSH**, а сканирование **Stealth Null** основано на передаче пробных пакетов без флагов.

Эти методы не позволяют сканировать большинство систем Windows, поскольку компания Microsoft, по своему обыкновению, проигнорировала стандарт и реализовала протокол как получилось[2]. Существуют и другие системы, в которых реакция на сканирование не соответствует стандарту. Системы Cisco, BSDI, HP/UX, MVS и IRIX передают пакет **RST** при сканировании открытых портов, хотя в соответствии со стандартом должны просто отбрасывать пакеты.

Вы можете видеть сравнить результаты сканирования одного хоста с использованием методов **TCP SYN** (рисунок 1) и **Stealth FIN** (рисунок 2).

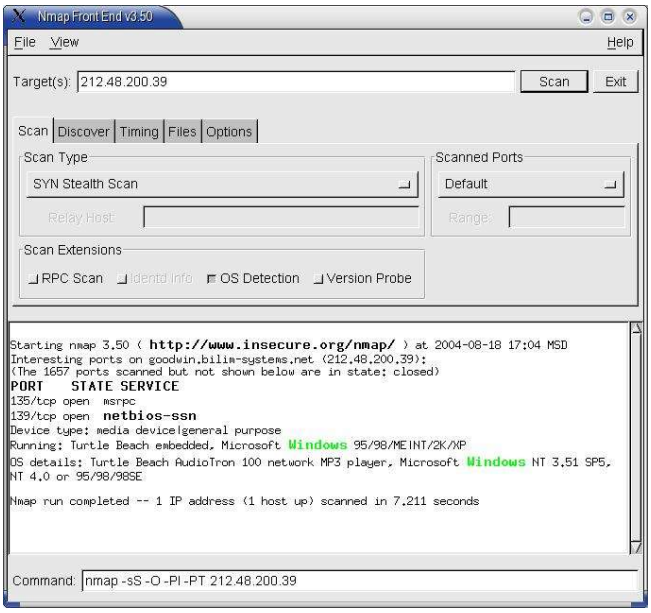

Рисунок 1 Результат сканирования TCP SYN

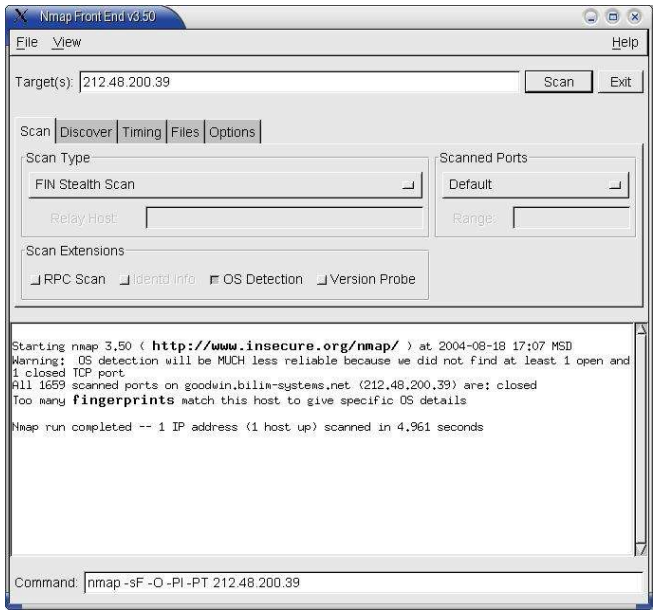

Рисунок 2 Результат сканирования Stealth FIN

Ping-сканирование (-sP)

Иногда бывает достаточно информации о наличии в сети активных хостов. Nmap может выполнять такую проверку путем передачи пакетов **ICMP echo request** по всем IP-адресам указанной сети. Получение отклика от хоста будет говорить о его активности. Однако, некоторые сайты блокируют пакеты **ICMP echo request**. В таких случаях **nmap** может передавать пакеты **TCP ACK** в указанный порт (по умолчанию 80). Получение в ответ пакета **RST** будет говорить об активности хоста. Третий вариант состоит в передаче пакетов **SYN** и ожидании отклика **RST** или **SYN/ACK**. Для пользователей, не имеющих привилегий **root**, применяется метод **connect()**.

По умолчанию для пользователя **root** программа **nmap** будет сканировать с помощью методов **ICMP** и **ACK** (параллельно). С помощью опции **-P** (стр. 5) вы можете выбрать метод проверки.

Отметим, что ping является стандартным способом проверки доступности хостов и обычно на такое сканирование никто не реагирует как на злой умысел.

Определение версии (-sV)

После определения портов TCP и UDP с помощью одного из методов сканирования система детектирования версий взаимодействует с открытыми портами, пытаясь определить реально работающие порты. Для выбора подходящего метода проверки используется информация из файла **nmap-service-probes**. **Nmap** пытается определить связанный с портом протокол (**FTP**, **SSH**, **telnet**, **HTTP** и т. п.), имя приложения (**ISC Bind**, **Apache httpd**, **Solaris telnetd** и т. п.), номер версии и другие доступные сведения. При компиляции **Nmap** с поддержкой **OpenSSL** программа будет пытаться организовать соединение с серверами SSL для определения возможности использования шифрованных соединений. При обнаружении служб RPC используется модуль **Nmap RPC grinder** для детектирования программы RPC и номера версии этой программы. Добавочная опция **--version_trace** обеспечит вывод отладочной информации в процессе детектирования версии для удаленного хоста.

Дополнительные сведения о системе детектирования версий вы найдете на сайте http://www.insecure.org/nmap/versionscan.html.

Сканирование UDP (-sU)

Этот метод применяется для детектирования открытых портов UDP. Метод основан на передаче пустых пакетов UDP проверяемому хосту. Получение в ответ пакета **ICMP port unreachable** будет говорить о том, что порт закрыт, а отсутствие такого отклика позволяет предположить наличие открытого порта. Однако зачастую пакеты **ICMP unreachable** фильтруются межсетевыми экранами, поэтому достоверной при таком сканировании можно считать только информацию о закрытых портах. В некоторых случаях Internet-провайдеры блокируют некоторые “опасные” порты (например 31337 - back orifice и 139 - Windows NetBIOS), что может

[1]Копию стандарта вы можете загрузить с сайта http://rfc-editor.org/rfc/rfc793.txt. Перевод документа на русский язык вы найдете на сайте http://www.protocols.ru.

[2]Нет худа без добра и поведение стека протоколов Microsoft позволяет с помощью этих методов сканирования достаточно достоверно идентифицировать хосты, работающие в среде Windows. Если после сканирования любым из этих методов вы получили информацию хотя бы об одном открытом порте, это говорит о том, что хост не использует Windows. Если же сканирование в режиме **-sF**, **-sX** или **-sN** говорит, что все порты закрыты, а сканирование SYN (**-sS**) показывает наличие открытых портов, это с высоким уровнем достоверности указывает на систему Windows. Польза от такой возможности невелика, поскольку **nmap** поддерживает эффективные средства детектирования ОС.

создать иллюзию открытости таких портов. Не впадайте в панику, если вы обнаружили нечто подобное при сканировании своей системы извне.

Иногда приходится слышать, что сканирование портов UDP не имеет никакого смысла. Автор nmap[3] в качестве контраргумента приводит ситуацию с уязвимостью Solaris **rpcbind**. Этот сервис можно связать с недокументированным портом UDP, имеющим номер более 32770. Найти такой порт без сканирования UDP достаточно сложно.

Сканирование портов UDP может занять продолжительное время, поскольку на большинстве хостов[4] реализованы рекомендации RFC 1812[5] (параграф 4.3.2.8) по ограничению скорости передачи откликов ICMP. Например, ядро Linux[6] ограничивает скорость генерации откликов **ICMP destination unreachable** 80 пакетами в течение 4 секунд с использованием паузы в 250 мсек после превышения заданного порога. В Solaris ограничение еще жестче (около 2 откликов в секунду). Такое ограничение скорости откликов существенно замедляет сканирования. **Nmap** старается определить скорость откликов и в соответствии с ней задать темп передачи запросов, поскольку в противном случае часть запросов пропадет втуне.

IP-сканирование (-sO)

Этот метод используется для определения поддерживаемых хостом протоколов IP. Метод основан на передаче raw-пакетов IP без дополнительных протокольных заголовков, адресованных всем протоколам проверяемого хоста. Получение отклика **ICMP protocol unreachable** говорит, что соответствующий протокол не поддерживается хостом, а отсутствие такого сообщения позволяет предположить наличие протокола[7].

Используемый этим методом вариант передачи пакетов похож на сканирование UDP и ему присущи те же ограничения, связанные с ограничением темпа генерации сообщений ICMP. Однако поле номера протоколов IP имеет размер 8 битов, поэтому проверяется лишь 256 протоколов и это не должно занять много времени.

Метод скрытого сканирования Idlescan (-sI)

`-sI <хост[:порт]>`

Этот метод позволяет полностью замести следы сканирования портов TCP и проверяемый хост не будет даже получать пакетов с IP-адресом сканирующей машины. Вместо передачи пакетов со сканирующего хоста используется подставной хост, доступный через Internet. Системы IDS будут показывать сканирование с указанного параметром подставного хоста и не смогут получить адрес вашего компьютера.

Предложенная в конце 1998 года технология сканирования **dumb host scan**[8] основана на предсказуемости значений поля IP ID в пакетах IP. Для реализации этого метода требуется промежуточный хост, имеющий по крайней мере один открытый порт TCP. Для успешного сканирования требуется, чтобы во время такой операции этот хост не проявлял собственной сетевой активности, но таких хостов в сети достаточно много. Для описания сути метода обозначим используемый для сканирования хост буквой **A**, промежуточный хост буквой **Z**, а проверяемый – **T**.

Хост **A** может осуществлять мониторинг хоста **Z** по значениям поля ID в заголовке передаваемых этим хостом пакетов IP. Дело в том, что большинство реализаций протокола IP просто увеличивают значение поля **ID** в заголовке пакетов IP на 1 для каждого следующего пакета. В этом легко убедиться с помощью описанной в Приложении программы **hping2**. Таким образом по значению поля **ID** в заголовке IP полученных от хоста откликов можно определить количество пакетов, переданных этим хостом в интервале между генерацией откликов.

Вспомним, что при получении пакета **SYN** для открытого порта хост передает в ответ пакет **SYN ACK**. Если же порт закрыт, в ответ на **SYN** передается пакет **RST ACK**. При получении неожиданного пакета **SYN ACK** хост передает в ответ пакет **RST**, а при получении неожиданного пакета **RST** просто отбрасывает такой пакет.

На основании сказанного легко построить модель скрытого сканирования.

1) Хост **A** генерирует серию запросов ICMP, позволяющую контролировать рост значений поля ID в заголовке IP полученных от **Z** откликов.
2) Хост **A** генерирует пакет SYN, адресованный в интересующий порт хоста **T**, используя в качестве адреса отправителя IP-адрес хоста **Z**.
3) Хост **T** при получении пакета **SYN** шлет хосту **Z** пакет **SYN ACK**, если проверяемый порт открыт и **RST ACK**, если этот порт закрыт.
4) Хост **Z** получает от хоста **T** неожиданный пакет **SYN ACK** или **RST ASK**.
   a) при получении **SYN ACK** хост **Z** передает отклик **RST**, вследствие чего увеличивается значение поля ID;
   b) при получении пакета **RST ASK** хост **Z** просто отбрасывает такой пакет и увеличения ID не происходит.
5) Хост **A**, анализируя значения поля ID в заголовках откликов ICMP от хоста **Z**, может фиксировать передачу хостом **Z** пакета в интервале между откликами. Исходя из предположения об отсутствии собственной активности хоста **Z**, это позволяет говорить о передаче отклика на пакет **SYN ACK** от хоста **T** и наличии у последнего открытого порта

Как видите, метод очень прост и при грамотной реализации метода позволяет с высокой степенью достоверности скрытно определять состояние портов проверяемого хоста без риска быть замеченным.

Необязательный параметр **<порт>**, передаваемый программе с этой опцией, позволяет указать порт подставного хоста, который будет использоваться при сканировании. По умолчанию **nmap** будет использовать **tcp ping**.

Поскольку состояние хоста **Z** во время сканирования играет достаточно важную роль, разумно будет сначала убедиться в том, что этот хост действительно подходит для наших целей. Сделать это можно, например, с помощью команды **hping -r <IP-адрес>**. Если

---

[3]Известный по имени Fyodor.
[4]Реализации протокола в продукции Microsoft, как обычно, не соответствуют RFC и скорость передачи откликов в системах Windows не ограничивается. Это позволяет просканировать все 65K портов UDP за достаточно короткое время
[5]Копию этого документа вы агрузить с сайта http://rfc-editor.org/rfc/rfc1812.txt.
[6]См. файл **<net/ipv4/icmp.h>**
[7]Некоторые ОС (AIX, HP-UX, Digital UNIX) и межсетевые экраны не передают сообщений о недоступности протокола, поэтому отсутствие таких сообщений не говорит однозначно о наличии протокола и лишь позволяет предположить его поддержку хостом.
[8]Сообщение о возможности такого метода можете найти на сайте http://seclists.org/bugtraq/1998/Dec/0082.html.

вывод этой команды будет подобен приведенному на рисунке (содержит **id=+1** в течение достаточно продолжительного времени), этот хост вполне подходит для использования в качестве подставного.

```
bash-2.05b# hping -r <IP-адрес>
HPING <IP-адрес> (eth0 <IP-адрес>): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=<IP-адрес> ttl=64 DF id=3064 sport=0 flags=RA seq=0 win=0 rtt=0.6 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=1 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=2 win=0 rtt=0.4 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=3 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=4 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=5 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=6 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=7 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=8 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=9 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=10 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=11 win=0 rtt=0.4 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=12 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=13 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=14 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=15 win=0 rtt=0.3 ms
len=46 ip=<IP-адрес> ttl=64 DF id=+1 sport=0 flags=RA seq=16 win=0 rtt=0.3 ms
```

Рисунок 3 Проверка подставного хоста

Программа nmap может корректно работать только с хостами, увеличивающими значение ID в каждом пакете на 1. Проверка показывает, что даже при стабильном значении **id=+1** программа просто не работает.

Если в процессе сканирования подставной хост будет генерировать пакеты по каким-либо иным причинам, кроме реакции на отклики от проверяемого хоста, это приведет к появлении в результатах ложной информации об открытых портах. Никто не мешает вам повторить сканирование, используя этот же хост или указав в качестве подставного другой хост.

ACK-сканирование (-sA)

Этот метод обычно используется для получения данных о политике межсетевого экрана. В частности, с помощью этого метода можно определить, учитывает брандмауэр состояние соединений (stateful inspection) или является простым пакетным фильтром, который блокирует входящие пакеты SYN.

В этом режиме программа передает пакеты ACK с кажущимися случайными номерами подтверждений и порядковыми номерами в сканируемые порты. При получении отклика **RST** порт считается нефильтруемым. Если же отклика просто не приходит или возвращается сообщение **ICMP unreachable**, порт считается фильтруемым. Программа **nmap** обычно не выводит сведений о нефильтруемых портах, поэтому отсутствие какого-либо списка портов в результате сканирования говорит о том, что ни один из проверенных портов не фильтруется.

В этом режиме список открытых портов обычно не выводится программой.

Window-сканирование (-sW)

Этот метод основан на определении размера окна TCP и похож на ACK-сканирование, но отличается от него тем, что наряду с детектированием состояния портов **filtered/unfiltered** он иногда может детектировать порты в состоянии **open** (вследствие получения аномальных размеров окна TCP, возвращаемых некоторыми ОС, включая AIX, Amiga, BeOS, BSDI, Cray, Tru64 UNIX, DG/UX, OpenVMS, Digital UNIX, FreeBSD, HP-UX, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.X, Ultrix, VAX, VxWorks).

Сканирование RPC (-sR)

Этот метод используется в сочетании с другими методами сканирования, поддерживаемыми программой **nmap** и служит для проверки всех обнаруженных при сканировании открытых портов TCP/UDP на предмет поддержки функций RPC[9]. Проверка осуществляется путем передачи в порт команды **NULL** с помощью **SunRPC**. Для поддерживающих RPC портов предпринимается попытка идентификации связанной с портом программы и номера ее версии. Таким образом можно определить порты и функции RPC даже в тех случаях когда порт **sunrpc** (**111**, **portmapper**) закрыт с помощью межсетевого экрана или иными средствами. В режиме сканирования RPC не работают приманки (decoy), описанные на стр. 7.

Сканирование по списку (-sL)

В этом режиме просто выводится список адресов IP или имен хостов, заданных другими параметрами командной строки без реального сканирования этих хостов.

Сканирование FTP bounce attack (-b)

-b <промежуточный сервер FTP>

Этот метод основан на возможности опосредованной передачи файлов по протоколу FTP (RFC 959[10]). Этот протокол имеет странную по сегодняшним меркам особенность, позволяющую пользователю хоста A подключиться к серверу FTP на другом хосте и запросить у этого сервера передачу файла **любому** хосту Internet. Как было отмечено еще в 1995 году, протокол FTP можно использовать для неконтролируемой рассылки электронной почты и новостных сообщений, заполнения чужих дисков ненужными файлами, попыток обхода межсетевых экранов и иных анонимных пакостей.

Программа **nmap** использует эту особенность протокола для сканирования портов TCP с использованием промежуточных серверов FTP. Сканер соединяется с сервером FTP, находящимся за брандмауэром, который закрывает сканируемый хост, и сканирует с

[9]Remote Procedure Call – удаленный вызов процедур.
[10]Копию документа вы можете агрузить с сайта http://rfc-editor.org/rfc/rfc959.txt.

его помощью закрытые межсетевым экраном порты (например, порт 139). Если сервер FTP имеет открытый для записи каталог, можно организовать передачу в порты сканируемого хоста любых данных, позволяющих найти в системе открытые порты.

Опция используется с параметром, указывающим параметры подключения к промежуточному серверу FTP в стандартной нотации URL (**username:password@server:port**). Все компоненты этого параметра, за исключением адреса или имени сервера, являются необязательными.

Следует отметить, что далеко не все современные серверы FTP пригодны для этого типа сканирования.

Опции общего назначения

Ни одна из перечисленных в таблице 1 опций не является обязательной, но многие опции весьма полезны. Отметим, что опции **-P** можно объединять – это поможет преодолеть даже весьма изощренные брандмауэры за счет использования различных портов и флагов TCP и кодов ICMP.

Таблица 1 Опции команды nmap

Опция	*Описание*
-P0	Эта опция отключает попытки использования команды **ping** перед сканированием хоста. С помощью этой опции вы сможете сканировать сети, блокирующие пакеты **ICMP echo** на межсетевом экране. Примером такой сети является **microsoft.com** и при сканировании хостов Microsoft всегда следует задавать опцию **-P0** или **-PT80**. Отметим, что **ping** в контексте сканирования портов может означать не только традиционную передачу запросов **ICMP echo**. **Nmap** поддерживает множество типов проб, включая зонды TCP, UDP и ICMP. По умолчанию **Nmap** передает пакеты **ICMP echo request** и пакеты **TCP ACK**, адресованные в порт 80.
-PT [<порты>]	Эта опция задает использование “TCP ping” с указанными номерами портов для определения доступности хоста. Взамен передачи запросов **ICMP echo** генерируются пакеты **TCP ACK** и анализируются отклики на эти запросы. Активные хосты должны передавать в качестве отклика на такие запросы пакет **RST**. Эта опция может быть весьма полезна для проверки доступности хостов в сетях, где брандмауэры блокируют пакеты ICMP. Если эту опцию указал пользователь, не имеющий привилегий **root** для сканирования будет вызываться функция **connect()**. Используемые для сканирования порты задаются в виде списка номеров или имен, разделенных запятыми **-PT<порт1>[,порт2][...]**. По умолчанию используется порт 80, поскольку его фильтруют достаточно редко.
-PS [<порты>]	Эта опция задает использование пакетов **SYN** вместо пакетов **ACK**, доступных только для пользователя **root**. Хосты должны отвечать на такие запросы пакетами **RST** или **SYN ACK**. Вы можете указать номера портов так же, как для опции **-PT**.
-PU [<порты>]	Эта опция задает передачу пакетов UDP в заданные порты и ожидание откликов **ICMP port unreachable** (порт закрыт) или UDP (порт открыт), если хост активен. Поскольку многие службы UDP не отвечают на пустые пакеты, эта опция полезна скорее для поиска закрытых портов, нежели открытых.
-PE	Задает использование стандартной операции **ping** (пакеты **ICMP echo request**) для определения доступности хостов.
-PP	Задает использование запросов **ICMP timestamp** (тип 13) для поиска активных хостов.
-PM	Подобна опциям **-PE** и **-PP**, но использует запросы **ICMP netmask** (тип 17).
-PB	Эта опция задает использование стандартной операции **ping** (**-PE**) в параллель с пакетами **ACK** (**-PT**). Такой способ позволяет обойти брандмауэры, которые блокируют один из этих вариантов проб. Для пакетов **ACK** можно задать номер порта, как было описано выше для опции **-PT**.
-O	Эта опция задает определение операционной системы сканируемых хостов с помощью методов **TCP/IP fingerprinting**[11]. Для детектирования ОС используется множество методов анализа стека протоколов сканируемого хоста. Полученная при сканировании информация сравнивается с “отпечатками” известных ОС[12] для идентификации операционной системы данного хоста. Эта опция также включает несколько дополнительных тестов, в частности – определение времени с момента загрузки хоста (Uptime) с помощью опции **TCP timestamp** (RFC 1323[13]), если она поддерживается проверяемым хостом. Кроме того, опция **-O** определяет уровень предсказуемости порядковых номеров TCP, определяющий сложность организации обманных соединений TCP с проверяемым хостом. Информация о предсказуемости порядковых номеров выводится только при наличии в командной строке опции **-v**. При одновременной использовании опций **-v** и **-O** определяется также алгоритм генерации порядковых номеров IP ID. Большинство хостов относится к классу **incremental**, использующему увеличение значения поля **ID** в заголовках IP на 1 для каждого генерируемого пакета. Такой алгоритм генерации порядковых идентификаторов может оказать весьма большую услугу злоумышленникам при организации атаки на хост или использовании этого хоста для скрытого сканирования других сетей (см. стр. 3).
-A	Этот флаг позволяет использовать расширенные возможности программы по детектированию ОС (**-O**), определению версии (**-sV**) и др. Эта опция не влияет на опции синхронизации программы, описанные ниже.

[11]Буквально - отпечатки пальцев.
[12]См. файл **nmap-os-fingerprints** из пакета **nmap**
[13]Копию RFC 1323 вы можете загрузить с сайта http://rfc-editor.org/rfc/rfc1323.txt.

Опция	*Описание*
-6	Эта опция включает поддержку протокола IPv6. Все цели сканирования должны задаваться адресами IPv6 (например, **3ffe:501:4819:2000:210:f3ff:fe03:4d0**) или полными доменными именами (записи AAAA). Версия **nmap6** доступна на сайте http://nmap6.sourceforge.net/.
-I	Эта опция включает режим сканирования **TCP reverse ident**. Дейв Голдсмит (Dave Goldsmith) в 1996 году отметил, что протокол **ident** (RFC 1413[14]) позволяет раскрыть имена пользователей, владеющих любыми процессами, подключенными по протоколу TCP, даже если этот процесс не был инициатором соединения. Таким образом можно, подключившись, например, к порту **http**, с помощью **identd** определить на сервере наличие процессов пользователя **root**. При использовании опции **-I** удаленному демону **identd** передаются запросы для каждого найденного на сервере открытого порта. Опция -I может использоваться только в режиме **TCP connect (-sT)**.
-f	Эта опция задает использование мелких фрагментов IP при сканировании в режимах **SYN** (стр. 1), **FIN** (стр. 1), **XMAS** (стр. 1) или **NULL** (стр. 1). Смысл заключается в разбиении заголовка TCP на множество компонент, передаваемых в разных фрагментах IP для затруднения работы пакетных фильтров, систем IDS и других способов обнаружения фактов сканирования. Эту опцию следует использовать с осторожностью, поскольку многие программы недостаточно корректно обрабатывают мелкие фрагменты.
-v	Задает вывод дополнительной информации в процессе сканирования и по завершении. Для дополнительного увеличения объема выводимых данных можно указать опцию дважды. Вы можете также указать в командной строке одну или несколько опций **-d** для вывода отладочной информации.
-h	Выводит на экран краткую справку о работе с программой.
-oN <файл>	Задает запись результатов сканирования в указанный текстовый файл.
-oX <файл>	Задает запись результатов сканирования в указанный файл XML. В качестве параметра опции можно указать символ **-** для вывода информации на **stdout** (например, в канал и т. п.); в этом случае программа отключает вывод информации на экран, а сообщения об ошибках будут передаваться на **stderr**. Описание вывода результатов в формате XML вы можете найти на сайте http://www.insecure.org/nmap/nmap.dtd.
-oG <файл>	Задает запись результатов сканирования в указанный файл, пригодный для операций поиска с помощью команды **grep**. В этом случае все результаты выводятся в одну строку файла. Такой формат может быть удобен для передачи результатов сканирования в другие программы, но формат XML (опция **-oX**) обеспечивает более эффективное решение. При использовании вместо имени файла символа **-** весь вывод будет направлен на **stdout** (например, в канал), а сообщения об ошибках будут направляться на **stderr**.
-oA <имя>	Говорит программе о необходимости записи результатов во всех поддерживаемых форматах (**-oN, -oG**, **-oX**). Параметр опции задает имя файла, к котором добавляется расширение **.nmap**, **.gnmap** и **.xml**, соответственно.
-oS <файл>	Задает запись результатов сканирования в указанный файл с использованием формата ScriptKiddie. Вместо имени файла можно указать символ **-** для вывода результатов на **stdout**.
--resume <файл>	С помощью этой опции может быть возобновлено сканирование, прерванное по тем или иным причинам, если его результаты были сохранены в указанном параметром опции текстовом файле (запись с опцией **-oN** или **-oG**). **Nmap** возобновит прерванное сканирование с использованием исходного набора опций.
--append_output	Говорит программе **nmap** о необходимости добавления информации в конец файла вместо перезаписывания этого файла.
-iL <файл>	Задает использование целей сканирования из указанного файла, который должен содержать список адресов и/или имен, разделенных пробелами, символами табуляции или новой строки (см. стр. 8). Если вы укажете вместо имени файла символ **-**, программа будет ждать список целей от устройства **stdin**.
-iR <количество>	Эта опция задает программе **nmap** сканирование указанного количества случайно выбранных адресов. Для бесконечного сканирования случайных адресов можно задать **-iR 0**. Такой способ может быть полезен для статистической оценки того или иного интересующего вас параметра (задается другими опциями) в сети Internet. Например, команда **nmap -sS -PS80 -iR 0 -p 80** поможет сделать случайную выборку web-серверов.
-p <порты>	Эта опция задает порт для проверки. Например, **-p 23** будет указывать программе, что на сканируемом хосте нужно проверить лишь порт 23, а **-p 20-30,139,60000-** будет проверять порты с 20 по 30, порт 139 и все порты с номерами выше 60000. По умолчанию программа сканирует порты с номерами от 1 до 1024 и все порты, указанные в файле **services** из пакета **nmap**. В режиме сканирования IP (**-sO**) эта опция задает номера проверяемых протоколов (0-255). При одновременном сканировании портов TCP и UDP вы можете задать номера портов отдельно для каждого протокола с помощью префиксов **T:** и **U:**. Все номера портов после префикса относятся к указанному протоколу, пока в строке не будет найден префикс другого протокола.
-F	Эта опция задает режим быстрого сканирования при котором проверяются только порты, указанные в файле **services** из пакета **nmap** (или в файле **protocols** при использовании режима **-sO**). Ограниченный набор проверяемых портов позволяет существенно ускорить процесс сканирования.

[14]Копию документа вы можете загрузить с сайта http://rfc-editor.org/rfc/rfc1413.txt. На сайте http://www.protocols.ru имеется перевод спецификации протокола на русский язык.

Опция	*Описание*
-D <decoy1[,decoy2][,ME],...>	Задает режим обмана сканируемого хоста, при котором последнему кажется, что сканирование осуществляется не только с вашего хоста, но и с хостов, указанных параметрами **decoy**. В результате системы IDS будут выдавать список из множества сканирующих хостов с уникальными адресами IP, среди которых ваш хост может просто затеряться. Для разделения подставных адресов используются запятые, а параметр **ME** указывает позицию списка адресов, в которую вы хотите поместить свой реальный адрес IP. Если вы укажете **ME** после пятого элемента списка или далее, некоторые детекторы сканирования просто никогда не покажут ваш адрес. Если параметр **ME** не указан, **nmap** будет помещать реальный адрес в случайную позицию. Не используйте в качестве подставных адреса неактивных хостов, поскольку это может привести к возникновению SYN-атаки на сканируемый хост. Кроме того, если вы укажете бездействующие адреса, среди них будет гораздо проще идентифицировать ваш реальный адрес. Некоторые детекторы сканеров (например, **portsentry**) будут подавлять маршрут к сканирующему хосту. Учитывая, что адрес сканирующего хоста может быть подставным, не следует принимать таких мер. Подставные адреса используются как при начальном ping-сканировании (с использованием **ICMP**, **SYN**, **ACK** и т. д.), так и при последующем реальном сканировании портов исследуемого хоста. Можно использовать подставные адреса и при определении ОС (опция **-O**). Нет ничего плохого в использовании большого числа подставных адресов, но это будет замедлять сканирование, а в некоторых случаях – снижать достоверность результатов. Отметим также, что некоторые операторы не выпускают из сети пакеты с обманными адресами отправителя и эта опция в таком случае не будет работать.
-S <адрес>	В некоторых случаях **nmap** не может определить адрес отправителя (вы получите сообщение об этом) и данная опция позволяет задать IP-адрес интерфейса, который будет использоваться для передачи пакетов в сеть.
-e <интерфейс>	Указывает программе **nmap** интерфейс, который следует использовать для передачи пакетов. Обычно интерфейс определяется автоматически.
-g <порт>	Задает номер порта отправителя для используемых при сканировании пакетов. Это может помочь в тех случаях, когда проверяемый хост защищен брандмауэром, но на последнем открыты некоторые порты[15]. Отметим, что в иногда задание порта ведет к снижению производительности сканирования.
--data_length <number>	Обычно **nmap** передает пакеты минимального размера, достаточного для включения заголовка транспортного уровня (для TCP 40 байтов, для ICMP – 28). Эта опция указывает программе на необходимость дополнения пакета случайными значениями до заданного размера. Опция не влияет на пакеты, используемые для определения ОС (режим **-O**), но влияет на большинство других пакетов. Отметим, что использование пакетов заданного размера несколько снижает производительность, но более крупные пакеты обычно привлекают меньше внимания, поскольку напоминают обычный трафик.
-n	Отключает преобразование адресов в символьные имена с помощью DNS. Эта опция может существенно ускорить процесс сканирования.
-R	Задает обязательное преобразование адресов в доменные имена с помощью DNS. Обычно преобразование осуществляется только для активных адресов.
-r	Отключает случайный выбор порядка сканируемых портов.
-ttl <value>	Устанавливает значение TTL в заголовках передаваемых пакетов IPv4.
--randomize_hosts	Говорит программе о необходимости перемешивания перед сканированием адресов каждой группы, содержащей до 2048 хостов. Такое перемешивание позволит избавиться от пристального внимания некоторых систем сетевого мониторинга, особенно если ее использовать совместно с опциями синхронизации (стр. 7).
-M <max sockets>	Задает максимальное число сокетов, которые будут использоваться при параллельном сканировании **TCP connect()**. Такое ограничение несколько замедляет сканирование, но снижает риск возникновения критических ошибок на сканируемых хостах.
--packet_trace	Говорит программе о необходимости вывода информации о всех передаваемых пакетах. Эта опция может быть полезна при отладке и обучении.
--datadir [<каталог>]	Задает имя каталога, в котором хранятся используемые программой файлы **nmapservices**, **nmap-protocols**, **nmap-rpc** и **nmap-os-fingerprints**. Nmap сначала ищет файлы в каталоге, заданном этой опцией, после чего просматривается каталог, указанный в переменной окружения **NMAPDIR**, далее – каталог **~/nmap**, и после этого - **/usr/share/nmap**. В качестве последней попытки **nmap** просматривает текущий каталог.

Опции синхронизации

В большинстве случаев программа **nmap** способна подстроить параметры работы с учетом состояния сети. Однако существуют ситуации, когда используемая программой по умолчанию политика синхронизации не будет соответствовать вашим задачам. В таких случаях вы можете самостоятельно выбрать политику синхронизации с помощью опции:

-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>

[15]Обычно открыты порты **DNS** (53), **HTTP** (80), **FTP-DATA** (20).

Таблица 2 Варианты политики синхронизации nmap

Политика		*Описание*
`Paranoid`	`0`	Медленное сканирование с целью избежать внимания систем IDS. Все операции выполняются последовательно с паузами не менее 5 минут между передачей пакетов.
`Sneaky`	`1`	Аналогичен предыдущему режиму, но паузы уменьшены до 15 секунд.
`Polite`	`2`	Эта политика синхронизации обеспечивает невысокий уровень загрузки сети и малую вероятность возникновения критических ошибок. Пакеты передаются с паузами не менее 4 мсек. Сканирование в этом режиме по крайней мере на порядок медленней, нежели в используемом по умолчанию режиме **Normal**.
`Normal`	`3`	Используемый по умолчанию режим, при котором сканирование осуществляется быстро и без перегрузки сети или пропуска хостов/портов.
`Aggressive`	`4`	Режим параллельного сканирования для уменьшения времени работы и снижения шансов на принятие ответных мер.
`Insane`	`5`	Этот режим пригоден только для очень скоростных сетей в тех случаях, когда потеря части информации не имеет существенного значения. Время ожидания для каждой пробы снижено до 300 мсек, а по истечение 15 минут сканирование хоста прекращается (тайм-аут).

Опция **-T0** будет задавать режим **Paranoid**, а **-T5** – **Insane**.

Кроме выбора политики, вы можете явно указать временные параметры работы программы с помощью опций, перечисленных в таблице 3.

Таблица 3 Опции синхронизации

Опция	*Описание*
`--host_timeout <мсек>`	Задает максимальное время сканирования одного хоста до перехода к следующему адресу. По умолчанию в режиме **Normal** время сканирования не ограничивается.
`--max_rtt_timeout <мсек>`	Задает максимальное время ожидания отклика хоста на пробный пакет **nmap** до передачи повторного пакета или констатации тайм-аута. В используемом по умолчанию режиме синхронизации это время составляет приблизительно 9 секунд (9000).
`--min_rtt_timeout <мсек>`	Задает минимальную паузу между передачей последовательных пробных пакетов. Обычно **nmap** сокращает интервал между передачей пакетов, если сканируемый хост отвечает достаточно быстро.
`--initial_rtt_timeout <мсек>`	Задает тайм-аут для первого пробного пакета. Обычно такое ограничение полезно при сканировании хостов, закрытых межсетевым экраном, в режиме -P0. Обычно **nmap** оценивает RTT по результатам **ping** и откликам на несколько первых пакетов. По умолчанию используется время ожидания 6 сек. (6000).
`--max_parallelism <number>`	Задает ограничение количества параллельных операций сканирования. При установке значения 1 **nmap** будет выполнять операции сканирования последовательно. Значение этого параметра влияет на все операции, которые могут выполняться в параллельном режиме (ping sweep, RPC scan и т. п.).
`--min_parallelism <number>`	Задает минимальное число сканируемых одновременно портов. Параллельное сканирование ускоряет процесс, но может снижать достоверность результатов.
`--scan_delay <мсек>`	Задает минимальную между передачей последовательных пакетов для снижения нагрузки на сеть и привлечения меньшего внимания со стороны IDS.

Выбор цели сканирования

Цель сканирования является единственным обязательным параметром команды **nmap**. В простейшем случае программа сканирует единственный хост, заданный именем или адресом IP в командной строке. Вы можете также задать сканирование подсети с указанной маской (**IP-адрес/маска**).

Nmap поддерживает широкие возможности задания цели сканирования. Например, для проверки сети класса B **192.168.*.*** вы можете указать цель как **192.168.*.***, **192.168.0-255.0-255**, **192.168.0.0/16** и даже **192.168.1-50,51-255.1,2,3,4,5-255**. Не забывайте, что многие командные процессоры при наличии в параметре символов * или / требуют использования двойных кавычек (**“”**).

Программа поддерживает и совсем экзотические варианты задания целей. Например, параметр ***.*.5.6-7** будет задавать сканирование хостов с номерами **.5.6** и **.5.7** во всех сетях класса B.

Примеры

```
nmap -v target.example.com
```

сканирование всех зарезервированных портов TCP хоста **target.example.com**; опция **-v** задает вывод подробного отчета.

```
nmap -sS -O target.example.com/24
```

сканирование **stealth SYN** каждого хоста сети, в которой находится хост **target.example.com**; при сканировании предпринимаются попытки определения ОС; использование такой команды требует привилегий **root**.

```
nmap -sX -p 22,53,110,143,4564 198.116.*.1-127
```

сканирование **Xmas tree** первой половины хостов (1 – 127) каждой сети класса C в сети класса B **198.116.0.0**; проверяются порты **ssh**, **DNS**, **pop3**, **imap** и **4564**[16].

[16]Напомним, что режим **Xmas** не обеспечивает сканирование хостов Windows по причине некорректной реализации стека протоколов TCP/IP, а также хостов CISCO, IRIX, HP/UX, BSDI.

```
nmap -v --randomize_hosts -p 80 *.*.2.3-5
```

определяет хосты с открытым портом **http** (80), имеющие значения **.2.3**, **.2.4** или **.2.5** в двух последних байтах адреса IP; адреса хостов выбираются случайно среди всех сетей класса B.

```
host -l company.com | cut  -d  -f 4 | ./nmap -v -iL -
```

копирует зону DNS для определения хостов домена **company.com** и выполняет сканирование хостов этого домена; в разных вариантах ОС детали команды могут отличаться.

Графические интерфейсы nmap

Программа nmap поддерживает большой набор опций командной строки и может показаться слишком сложной в использовании. Для тех, кто предпочитает работать с графическими интерфейсами, существует по крайней мере два варианта.

Модуль Webmin

Для программы Webmin существует модуль **Network Utilities**, в состав которого входит интерфейс управления сканером nmap. К сожалению, это интерфейс не поддерживает всех возможностей сканера, зато он позволяет работать с удаленными хостами. Вид интерфейса показан на рисунке 4.

Модуль **Network Utilities** вы можете загрузить с сайта http://www.niemueller.de/webmin/modules/nettools.

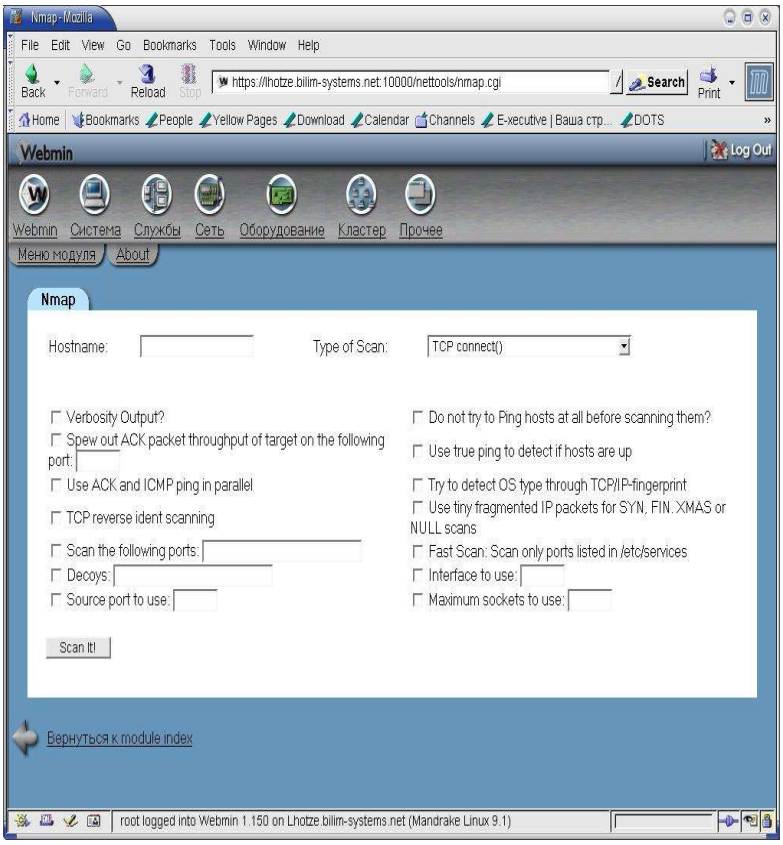

Рисунок 4 Интерфейс Webmin для сканера nmap

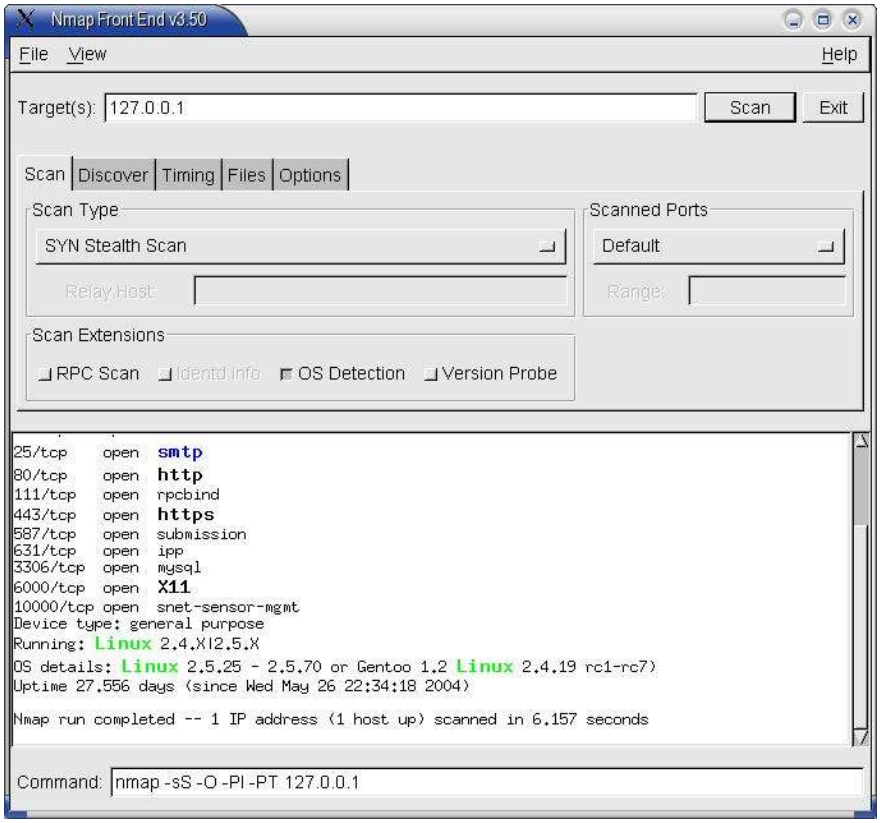

Рисунок 5 Интерфейс nmapfe

nmapfe

Программа **nmapfe** (xnmap) обеспечивает графический интерфейс для сканера безопасности **nmap**, построенный на базе GTK+.

Синтаксис

```
nmapfe [опции Glib]
```