

Nmap Compilation, Installation, And Removal

By Fyodor

Introduction

Nmap can often be installed or upgraded with a single command, so don't let the length of this guide scare you. Most readers will use the table of contents to skip directly sections that concern them. This guide describes how to install Nmap on many platforms, from Windows and OpenBSD to the Sharp Zaurus PDA, including both source code compilation and binary installation methods. Graphical and command-line versions of Nmap are described and contrasted. Nmap removal instructions are also provided in case you change your mind.

Testing whether Nmap is already installed

The first step toward obtaining Nmap is to check whether you already have it. Many free operating system distributions (including most Linux and BSD systems) come with Nmap, although it may not be installed by default. On UNIX systems, open a terminal window and try executing the command **nmap --version**. If Nmap exists and is in your \$PATH, you should see output similar to Example 1, "Checking for Nmap and determining its version number".

Example 1. Checking for Nmap and determining its version number

```
felix~>nmap --version
nmap version 3.95 ( http://www.insecure.org/nmap )
felix~>
```

If Nmap does *not* exist on the system (or if your \$PATH is incorrectly set), an error message such as `nmap: Command not found` displays. As the example above shows, Nmap responds to the command by printing its version number (here 3.95).

Even if your system already has a copy of Nmap, you should consider upgrading to the latest version available from <http://www.insecure.org/nmap/download.html>. Newer versions often run faster, fix important bugs, and feature updated operating system and service version detection databases. A list of changes since the version already on your system can be found at <http://www.insecure.org/nmap/changelog.html>.

Command-line and graphical interfaces

Nmap has traditionally been a command-line application run from a UNIX shell or (more recently) Windows command prompt. This allows experts to quickly execute a command that does exactly what they want without having to maneuver through a bunch of configuration panels and scattered option fields. This also makes Nmap easier to script and enables easy sharing of useful commands among the user community.

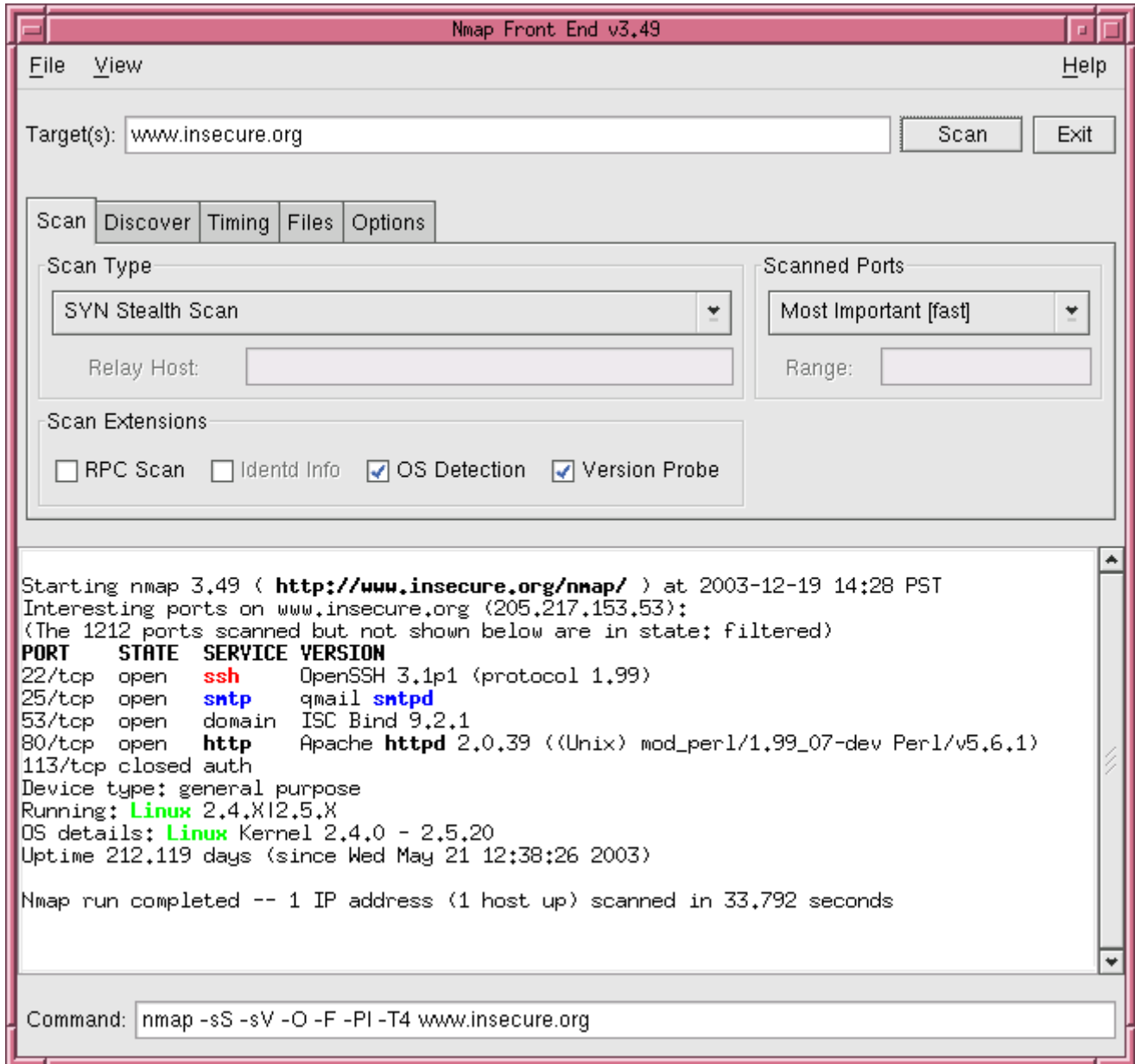
One downside of the command-line approach is that it can be intimidating for new and infrequent users. Nmap offers more than a hundred command-line options, although many are obscure features or debugging controls that most users can ignore. Many graphical frontends have been created for those users who prefer a GUI interface. The most common GUI for UNIX is NmapFE, which is distributed as part of the Nmap project. It offers a number of option panes (Scan, Discover, Timing, Files, and Options), which are all used to build an appropriate Nmap command. The Nmap command-line is shown at the bottom of the window as it is constructed. This feature helps people learn the syntax in case they wish to migrate to the command-line version. There is not presently a field for entering arbitrary Nmap options, but one trick is to stick them in the big Target(s) field. Once the

Nmap Compilation, Installation, And Removal

By Fyodor

command is constructed to your liking, press the Scan button to launch Nmap. Raw Nmap output (with added color for service emphasis) is shown in a large white window, as seen in Figure 1, "NmapFE presents a simple graphical interface to Nmap".

Figure 1. NmapFE presents a simple graphical interface to Nmap



Unfortunately, NmapFE does not yet work well on the Windows platform. Several projects are underway to produce powerful, cross-platform Nmap GUIs and results viewers, but none are yet ready for wide-scale use.

Once you understand how the command-line options work and can interpret the output, using any of the available Nmap GUIs is trivial. The options are all the same whether you choose them from radio buttons and menus or type them at a command-line.

Nmap Compilation, Installation, And Removal

By Fyodor

Downloading Nmap

Insecure.Org is the official source for downloading Nmap source code and binaries for Nmap and NmapFE. Source code is distributed in Bzip2 and Gzip compressed tar files, and binaries are available for Windows and Linux (RPM format). Find all of this at <http://www.insecure.org/nmap/download.html>.

Verifying the integrity of Nmap downloads

It often pays to be paranoid about the integrity of files downloaded from the Internet. Popular packages such as Sendmail^[1], OpenSSH^[2], tcpdump, libpcap, BitchX, Fragrouter, and many others have been infected with malicious trojans. Software distributions sites at the Free Software Foundation, Debian, and SourceForge have also been successfully compromised. This has never happened to Nmap, but one should always be careful. To verify the authenticity of an Nmap releases, consult the PGP detached signatures or cryptographic hashes (including SHA1 and MD5) posted for each release to the Nmap sigs directory at <http://www.insecure.org/nmap/dist/sigs/?C=M&O=D>.

The most secure verification mechanism is detached PGP signatures. As the signing key is never stored on production servers, even someone who successfully compromises the web server couldn't forge and properly sign a trojan release. While numerous applications are able to verify PGP signatures, I recommend the GNU Privacy Guard (GPG).

Nmap releases are signed with a special Nmap Project Signing Key, which can be obtained from they major key servers or http://www.insecure.org/nmap/data/nmap_gpgkeys.txt. My key is included in that file too. The keys can be imported with the command **gpg --import nmap_gpgkeys.txt**. You only need to do this once, then you can verify all future Nmap releases from that machine. Before trusting the keys, verify that the fingerprints match the values shown in Example 2, "Verifying the Nmap and Fyodor PGP Key Fingerprints".

Example 2. Verifying the Nmap and Fyodor PGP Key Fingerprints

```
flog~> gpg --fingerprint nmap fyodor
pub 1024D/33599B5F 2005-04-24
    Key fingerprint = BB61 D057 C0D7 DCEF E730 996C 1AF6 EC50 3359 9B5F
uid      Fyodor <fyodor@insecure.org>
sub 2048g/D3C2241C 2005-04-24

pub 1024D/6B9355D0 2005-04-24
    Key fingerprint = 436D 66AB 9A79 8425 FDA0 E3F8 01AF 9F03 6B93 55D0
uid      Nmap Project Signing Key (http://www.insecure.org/)
sub 2048g/A50A6A94 2005-04-24
```

For every Nmap package download file (e.g. nmap-3.95.tar.bz2 and nmap-3.95-win32.zip), there is a corresponding file in the sigs directory with .gpg.txt appended to the name (e.g. nmap-3.95.tar.bz2.gpg.txt). This is the detached signature file.

With the proper PGP key in your keyring and the detached signature file downloaded, verifying an Nmap release takes a single GPG command, as shown in Example 3, "Verifying PGP Key Fingerprints (Successful)". If the file has been tampered with, the results will look like Example 4, "Detecting a bogus file"

Nmap Compilation, Installation, And Removal

By Fyodor

Example 3. Verifying PGP Key Fingerprints (Successful)

```
flog~> gpg --verify nmap-3.95.tar.bz2.gpg.txt nmap-3.95.tar.bz2
gpg: Signature made Thu 08 Dec 2005 12:26:23 AM PST using DSA key ID 6B9355D0
gpg: Good signature from "Nmap Project Signing Key (http://www.insecure.org/)"
```

Example 4. Detecting a bogus file

```
flog~> gpg --verify nmap-3.95.tar.bz2.gpg.txt nmap-3.95-hacked.tar.bz2
gpg: Signature made Thu 08 Dec 2005 12:26:23 AM PST using DSA key ID 6B9355D0
gpg: BAD signature from "Nmap Project Signing Key (http://www.insecure.org/)"
```

While PGP signatures are the recommended validation technique, SHA1 and MD5 (among other) hashes are made available for more casual validation. An attacker who can manipulate your Internet traffic in real time (and is extremely skilled) or who compromises Insecure.Org and replaces both the distribution file and digest file, could defeat this test. However, it can be useful to check the authoritative Insecure.Org hashes if you obtain Nmap from a third party or feel it might have been accidentally corrupted. For every Nmap package download file, there is a corresponding file in the sigs directory with .digest.txt appended to the name (e.g. nmap-3.95.tar.bz2.digest.txt). An example is shown in Example 5, "A typical Nmap release digest file". This is the detached signature file. The hashes can be verified with common tools such as md5sum, sha1sum, or gpg, as shown in Example 6, "Verifying Nmap hashes".

Example 5. A typical Nmap release digest file

```
flog~> cat nmap-3.95.tar.bz2.digest.txt
nmap-3.95.tar.bz2: MD5 = A1 6E 9D 7F 79 12 AF 7C 93 FA A7 94 BD 63 B6 D1
nmap-3.95.tar.bz2: SHA1 = 1D8A 10FF B295 80E3 EA04 06D5 2A38 86C8 DE8B F21B
nmap-3.95.tar.bz2: RMD160 = BA57 2025 2743 18A6 8CB3 215D 7E67 511A 92A5 DC20
nmap-3.95.tar.bz2: SHA256 = EC48A1D9 AE7D34BB CAB61F75 527A36FE 58695813
36E201D2 CA8E9C68 0D3D5D2F
nmap-3.95.tar.bz2: SHA384 = 6B0A6E9D 0FACBCD7 F7482253 461A4E95 12701A6B
0AE308CF B3E827F5 4474049F 3563D35B 3E550E9E
FFD74069 516E459D
nmap-3.95.tar.bz2: SHA512 = B57BF296 56AEB0E3 02FE05DA 7D738886 A3AC3180
8F4C2299 3E170F28 542E9E37 43AEE424 1B362560
6248F4A2 7096A9A4 7BAD3E08 B97F9D14 B420D32E
08E6A2B3
```

Example 6. Verifying Nmap hashes

```
flog~> gpg --print-md sha1 nmap-3.95.tar.bz2
nmap-3.95.tar.bz2: 1D8A 10FF B295 80E3 EA04 06D5 2A38 86C8 DE8B F21B
flog~> sha1sum nmap-3.95.tar.bz2
1d8a10ffb29580e3ea0406d52a3886c8de8bf21b nmap-3.95.tar.bz2
flog~> md5sum nmap-3.95.tar.bz2
a16e9d7f7912af7c93faa794bd63b6d1 nmap-3.95.tar.bz2
```

While releases from Insecure.Org are signed as described in this section, certain Nmap add-ons, interfaces, and platform-specific binaries are developed and distributed by other parties. They have different mechanisms for establishing the authenticity of their downloads.

Nmap Compilation, Installation, And Removal

By Fyodor

[¹] <http://www.cert.org/advisories/CA-2002-28.html>

[²] <http://www.cert.org/advisories/CA-2002-24.html>

UNIX Compilation and installation from source code

While binary packages discussed in later sections are available for most platforms, compilation and installation from source code is the traditional and most powerful way to install Nmap. This insures that the latest version is available and allows Nmap to adapt to the library availability and directory structure of your system. For example, Nmap uses the OpenSSL cryptography libraries for version detection when available, but most binary packages do not include this functionality. On the other hand, binary packages are generally quicker and easier to install, and allow for consistent management (installation, removal, upgrading, etc.) of all packaged software on the system.

Source installation is usually a painless process - the build system is designed to auto-detect as much as possible. Here are the steps required for a default install:

1. Download the latest version of Nmap in .tar.bz2 (bzip2 compression) or .tgz (gzip compression) format from <http://www.insecure.org/nmap/download.html>.
2. Decompress the downloaded tarball with a command such as:

```
bzip2 -cd nmap-VERSION.tar.bz2 | tar xvf -
```

If you downloaded the .tgz version, replace bzip2 with gzip in the command above. With GNU tar, the simpler command **tar xvjf nmap-VERSION.tar.bz2** does the trick.

3. Change into the newly created directory: **cd nmap-VERSION**
4. Configure the build system: **./configure**
5. Build Nmap (and GUI nmapfe if requirements met): **make**

Note that GNU Make is required. On BSD-derived UNIX systems, this is often installed as *gmake*. So if **make** returns a bunch of errors such as "Makefile, line 1: Need an operator", try running **gmake** instead.

6. Become a privileged user for systemwide install: **su root**
7. Install Nmap, support files, docs, etc.: **make install**

Congratulations! Nmap is now installed as /usr/local/bin/nmap! Run it with no arguments for a quick help screen.

As you can see above, a simple source compilation and install consists of little more than **./configure;make;make install**. However, there are a number of options available to configure that affect the way Nmap is built.

Configure directives

Most of the UNIX build options are controlled by the configure script, as used in step number four above. There are dozens of command-line parameters and environmental variables which affect the

Nmap Compilation, Installation, And Removal

By Fyodor

way Nmap is built. Run **./configure --help** for a huge list with brief descriptions. Here are the ones that are specific to Nmap or particularly important:

--prefix=directoryname

This option, which is standard to the configure scripts of most software, determines where Nmap and its components are installed. By default, the prefix is `/usr/local`, meaning that nmap is installed in `/usr/local/bin`, the man page (`nmap.1`) is installed in `/usr/local/man/man1`, and the data files (`nmap-os-fingerprints`, `nmap-services`, `nmap-service-probes`, etc.) are installed under `/usr/local/share/nmap`. If you only wish to change the path of certain components, use the options `--bindir`, `--datadir`, and/or `--mandir`. An example usage of `--prefix` would be to install Nmap in my account as an unprivileged user. I would run **./configure --prefix=/home/fyodor**. Nmap creates subdirs like `/home/fyodor/man/man1` in the install stage if they do not already exist.

--without-nmapfe

This option prevents the NmapFE graphical X-Window frontend from being built. Normally the build system checks your system for requirements such as the GTK graphical widget library and then build NmapFE if they are all available.

--with-openssl=directoryname

The version detection subsystem of Nmap is able to probe SSL-encrypted services using the free OpenSSL libraries. Normally the Nmap build system looks for these libraries on your system and include this capability if they are found. If they are in a location your compiler does not search for by default, but you still want them to be used, specify `--with-openssl=directoryname`. Nmap then looks in `directoryname/libs` for the OpenSSL libraries themselves and `directoryname/include` for the necessary header files. Specify `--without-openssl` to disable SSL entirely.

--with-libpcap=directoryname

Nmap uses the Libpcap library for capturing raw IP packets. Nmap normally looks for an existing copy of Libpcap on your system and use that if the version number and platform is appropriate. Otherwise Nmap includes its own recent copy of Libpcap, which has been modified for improved Linux functionality. The specific changes are described in `libpcap-possiblymodified/CHANGES` in the Nmap source directory. Because of these Linux-related changes, Nmap always uses its own Libpcap by default on that platform. If you wish to force Nmap to link with your own Libpcap, pass the option `--with-libpcap=directoryname` to configure. Nmap then expects the Libpcap library to be in `directoryname/lib/libpcap.a` and the include files to be in `directoryname/include`. Nmap will always use the version of Libpcap included in its tarball if you specify `--with-libpcap=included`.

--with-libpcre=directoryname

LibPCRE is a Perl-compatible regular expression library available from <http://www.pcre.org>. Nmap normally looks for a copy on your system, and then fall back to its own copy if that fails. If your PCRE library is not in your compiler's standard search path, Nmap probably will not find it. In that case you can tell Nmap where it can be found by specifying the option `--with-libpcre=directoryname` to configure. Nmap then expects the library files to be in

Nmap Compilation, Installation, And Removal

By Fyodor

directoryname/lib and the include files to be in *directoryname/include*. In some cases, you may wish to use the PCRE libraries included with Nmap in preference to those already on your system. In that case, specify `--with-libpcre=included`.

`--with-libdnet=directoryname`

Libdnet is an excellent networking library that Nmap uses for sending raw Ethernet frames. The version in the Nmap tree is heavily modified (particularly the Windows code), so the default is to use that included version. If you wish to use a version already installed on your system instead, specify `--with-libdnet=directoryname`. Nmap then expects the library files to be in *directoryname/lib* and the include files to be in *directoryname/include*.

`--with-localdirs`

This simple option tells Nmap to look in `/usr/local/lib` and `/usr/local/include` for important library and header files. This should never be necessary, except that some people put such libraries in `/usr/local` without configuring their compiler to find them. If you are one of those people, use this option.

If you encounter compilation problems

In an ideal world, software would always compile perfectly (and quickly) on every system you maintain. Unfortunately, society has not yet reached that state of nirvana. Despite all the efforts to make Nmap portable, compilation issues occasionally arise. Here are some suggestions in case the source distribution compilation fails.

Upgrade to the latest Nmap

Check <http://www.insecure.org/nmap/download.html> to make sure you are using the latest version of Nmap. The problem may have already been fixed.

Read the error message carefully

Scroll up in the output screen and examine the error messages given when commands fail. It is often best to find the first error message, as that often causes a cascade of further errors. Read the error message carefully, as it could indicate a system problem such as low disk space or a broken compiler. Users with programming skills may be able to resolve a wider range of problems themselves. If you make code changes to fix the problem, please send a patch (created with `diff -uw oldfile newfile`) and any details about your problem and platform to me at fyodor@insecure.org. Integrating the change into the base Nmap distribution allows many other users to benefit, and prevents you from having to make the changes with each new Nmap version.

Ask Google and other Internet resources

Try searching for the exact error message on Google or other search engines. You might also want to browse recent activity on the Nmap development (nmap-dev) list -- archives are available at <http://seclists.org>.

Ask nmap-dev

Nmap Compilation, Installation, And Removal

By Fyodor

If none of your research has led to a solution for your problem, try sending a report to the Nmap development (nmap-dev) list. If you subscribe first, your message gets through faster because it does not go through moderation. Subscribe by sending a blank email to <nmap-dev-subscribe@insecure.org> and post to the list by mailing <nmap-dev@insecure.org>. Be sure to describe your problem in full, including the Nmap version number, platform you are running on, and any relevant output snippets showing the error.

Consider binary packages

Binary packages of Nmap are available on most platforms and are usually easy to install. The downsides are that they may not be as up-to-date and you lose some of the flexibility of self-compilation. Previous sections of this chapter describe how to find binary packages on many platforms, and even more are available via Internet searching.

Linux Distributions

Linux is far and away the most popular platform for running Nmap. In one user survey, 86% said that Linux was at least one of the platforms on which they run Nmap.

Linux users can choose between a source code install or using binary packages provided by their distribution. The binary packages are generally quicker and easier to install, and are often slightly customized to use the distribution's standard directory paths and such. These packages also allow for consistent management in terms of upgrading, removing, or surveying software on the system. A downside is that packages created by the distributions are necessarily behind the Insecure.Org source releases. Most Linux distributions (particularly Debian and Gentoo) keep their Nmap package relatively current, though a few are way out of date. Choosing the source install allows for more flexibility in determining how Nmap is built and optimized for your system. To build Nmap from source, see the section called "UNIX Compilation and installation from source code". Here are simple package instructions for the most common distributions.

RPM-based distributions (Red Hat, Mandrake, Suse, Fedora)

I build RPM packages for every release of Nmap and post them to the Nmap download page at <http://www.insecure.org/nmap/download.html>. I build two packages: The nmap package contains just the command-line executable and data files, while the nmap-frontend package contains the optional X-Window graphical frontend named nmapfe. The nmap-frontend package is optional and only necessary for those who want a GUI interface to Nmap. It does require that the nmap package be installed first. One down side to installing the RPMs rather than compiling from source is that the RPMs don't support OpenSSL for version detection of SSL services.

Installing via rpm is quite easy - it even downloads the package for you when given the proper URLs. The following example downloads and installs Nmap 3.95, including the frontend. Of course you should use the latest version at the download site above instead. Any existing RPM-installed versions are upgraded. Example 7, "Installing Nmap from binary RPMs" demonstrates this installation process.

Example 7. Installing Nmap from binary RPMs

```
# rpm -vhU http://download.insecure.org/nmap/dist/nmap-3.95-1.i386.rpm
Retrieving http://download.insecure.org/nmap/dist/nmap-3.95-1.i386.rpm
Preparing...      ##### [100%]
 1:nmap          ##### [100%]
```

Nmap Compilation, Installation, And Removal

By Fyodor

```
# rpm -vhU http://download.insecure.org/nmap/dist/nmap-frontend-3.95-1.i386.rpm
Retrieving http://download.insecure.org/nmap/dist/nmap-frontend-3.95-1.i386.rpm
Preparing...      ##### [100%]
 1:nmap-frontend  ##### [100%]
core/home/fyodor#
```

As the filenames above imply, these binary RPMs were created for normal PCs (X86 architecture). I also distribute x86_64 binaries of some releases for users with 64-bit Linux running on an AMD Opteron or Athlon64 processor. These binaries won't work for the relatively few Linux users on other platforms such as SPARC, Alpha, or PowerPC. They also may refuse to install if your library versions are sufficiently different from what the RPMs were initially built on. One option in these cases would be to find binary RPMs prepared by your Linux vendor for your specific distribution. The original install CDs or DVD are a good place to start. Unfortunately, those may not be current or available. Another option is to install Nmap from source code as described previously, though you lose the binary package maintenance consistency benefits. A third option is to build and install your own binary RPMs from the source RPMs distributed from the download page above. Example 8, "Building and installing Nmap from source RPMs" demonstrates this technique with Nmap 3.95.

Example 8. Building and installing Nmap from source RPMs

```
> rpm --rebuild http://download.insecure.org/nmap/dist/nmap-3.95-1.src.rpm
[ hundreds of lines cut ]
Wrote: /home/fyodor/rpmdir/RPMS/i386/nmap-3.95-1.i386.rpm
Wrote: /home/fyodor/rpmdir/RPMS/i386/nmap-frontend-3.95-1.i386.rpm
[ cut ]
> su
Password:
# rpm -vhU /home/fyodor/rpmdir/RPMS/i386/nmap-*3.95-1.i386.rpm
Preparing...      ##### [100%]
 1:nmap           ##### [ 50%]
 2:nmap-frontend  ##### [100%]
#
```

Removing RPM packages is as easy as `rpm -e nmap nmap-frontend`.

Debian Linux and Derivatives

LaMont Jones does a fabulous job maintaining the Nmap .deb packages, including keeping them reasonably up-to-date. The proper upgrade/install command is `apt-get install nmap`. This works for debian derivatives such as Ubuntu too. Information on the latest Debian "stable" Nmap package is available at <http://packages.debian.org/stable/net/nmap.html> and the development ("unstable") package info is available from <http://packages.debian.org/unstable/net/nmap.html>.

Gentoo Linux

I believe Gentoo uses "emerge nmap" or some such. Can anyone send me details?

Other Linux distributions

There are far too many Linux distributions available to list here, but even many of the obscure ones include Nmap in their package tree. If they don't, you can simply compile from source code as

Nmap Compilation, Installation, And Removal

By Fyodor

described in the section called "UNIX Compilation and installation from source code". *If I am missing any important distributions, please send me details on installing their Nmap binary package*

Windows

While Nmap was once a UNIX-only tool, a Windows version was released in 2000 and has since become the second most popular Nmap platform (behind Linux). Because of this popularity and the fact that many Windows users do not have a compiler, binary executables are distributed for each major Nmap release. While it is improving rapidly, the Windows port is still not as efficient or stable as on UNIX. Here are some known limitations (at the time of this writing):

- You cannot generally scan your own machine from itself (using a loopback IP such as 127.0.0.1 or any of its registered IP addresses). This is a Windows limitation that we haven't found a way around. If you really want to do this, use a TCP connect scan without pinging (-sT -P0) as that uses the high level socket API rather than sending raw packets.
- Nmap only supports Ethernet interfaces (including many 802.11 wireless cards) unless you use the -sT -P0 options. RAS connections (such as PPP dialups) are not supported. This support was dropped when Microsoft removed raw TCP/IP socket support in Windows XP SP2. Now Nmap must send lower-level ethernet frames instead.
- Version detection cannot use SSL scan-through
- Scans from Windows often take longer than on UNIX

Scans speeds on Windows are generally comparable to those on UNIX, though the latter often has a slight performance edge. One exception to this is connect scan (-sT), which is often much slower than on UNIX because of deficiencies in the Windows networking API. This is a shame, since that is the one TCP scan that works against localhost and over all networking types (not just ethernet, like the raw packet scans). Connect scan performance can be improved substantially by applying the registry changes in the nmap_performance.reg file included with Nmap. It is in the nmap-VERSION directory of the Windows binary zip file, and nmap-VERSION/mswin32 in the source tarball. These changes increase the number of ephemeral ports reserved for user applications (such as Nmap) and decreases the amount of time before a closed connection can be reused. Apply the by double-clicking on nmap_performance.reg, or run the command **regedt32 nmap_performance.reg**. Or you can make the changes by hand. Simply add these three registry DWORD values to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:

MaxUserPort

Set a large value such as 65534 (0x0000fffe). See MS KB Q196271.

TCPTimedWaitDelay

Set the minimum value (0x0000fffe). See MS KB Q196271.

StrictTimeWaitSeqCheck

Set to 1 so TcpTimedWaitDelay is checked.



Note

I would like to thank Ryan Permech of eEye, Andy Lutomirski, and Jens Vogt for their hard work on the Nmap Windows port. For many years, Nmap was a UNIX-only tool, and it would likely still be

Nmap Compilation, Installation, And Removal

By Fyodor

that way if not for their efforts.

Windows users have two choices for installing Nmap, both of which are available from the download page at <http://www.insecure.org/nmap/download.html>.

Command line .zip binaries

Every major "stable" Nmap release comes with Windows command-line binaries and associated files in a Zip archive. No graphical interface is included, so you need to run nmap.exe from a DOS/command window. Or you can download and install a superior command shell such as those included with the free Cygwin system available from <http://www.cygwin.com>. Here are the step-by-step instructions for installing and executing the Nmap .Zip binaries.

Installing the Nmap .Zip binaries

1. Read the Nmap Win32 support page for the latest updates
2. Download the .Zip binaries from <http://www.insecure.org/nmap/download.html>.
3. Uncompress the zip-file into the directory you want Nmap to reside in. An example would be "C:\Program Files\". A directory called nmap-VERSION should be created, which includes the Nmap executable and data files. If you do not have a Zip decompression program, there is one (called unzip) in Cygwin above, or you can download the open source and free 7-zip utility from <http://www.7-zip.org>. Commercial alternatives are Winzip and PKZIP from <http://www.winzip.com> and <http://www.pkware.com> respectively.
4. For improved performance, apply the Nmap registry changes discussed previously.
5. Nmap requires the free WinPcap packet capture library. Obtain and install the latest version from <http://www.winpcap.org>. They distribute an executable installer which makes this easy. You must have version 3.1 or later.

Executing Nmap as installed above

1. Make sure the user you are logged in as has administrative privileges in the box (should be in the administrators group).
2. Open a command/DOS Window. Though it can be found in the program menu tree, the simplest approach is to choose Start -> Run and type cmd<enter>. Opening a Cygwin window (if you installed it) by clicking on the Cygwin icon on the desktop works too, although the necessary commands differ slightly from those shown below.
3. Change to the directory you installed Nmap into. Assuming the example directory name used in the install section above, type the following commands.

```
C:  
cd "\program files\nmap-VERSION" (replace VERSION with the Nmap version number)
```

4. Execute nmap.exe. Figure 2, "Executing Nmap from a Windows command shell" is a screen shot showing a simple example

Figure 2. Executing Nmap from a Windows command shell

Nmap Compilation, Installation, And Removal

By Fyodor

```
cmd.exe (running as PLAYGROUND\root)
E:\>cd nmap
E:\nmap>nmap -A -T4 scanme.insecure.org

Starting nmap 3.48 < http://www.insecure.org/nmap > at 2003-12-20 03:20 Pacific
Standard Time
Interesting ports on scanme.insecure.org (205.217.153.55):
<The 1652 ports scanned but not shown below are in state: filtered>
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 3.1p1 <protocol 1.99>
25/tcp    open  smtp     gmail smtpd
53/tcp    open  domain   ISC Bind 9.2.1
80/tcp    open  http     Apache httpd 2.0.39 <<Unix> mod_perl/1.99_07-dev Perl/v5.
6.1)
113/tcp   closed auth
Device type: general purpose
Running: Linux 2.4.X!2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 212.653 days <since Wed May 21 12:40:35 2003>

Nmap run completed -- 1 IP address <1 host up> scanned in 54.588 seconds
E:\nmap>
```

If you execute Nmap frequently, you can the Nmap directory (c:\program files\nmap-VERSION in this case) to your command execution path. The exact place to set this varies by Windows platform. On my Windows XP box, I do the following:

1. From the desktop, right click on My Computer and then click properties.
2. In the System Properties window, click the Advanced tab.
3. Click the Environment Variables button.
4. Choose Path from the System variables section, then hit edit.
5. Add a semi-colon and then your Nmap directory (such as c:\program files\nmap-VERSION) to the end of the value.
6. Open a new DOS window and you should be able to execute a command such as **nmap scanme.nmap.org** from any directory.

Compile from source code

Most Windows users prefer to use the Nmap binary distribution, but compilation from source code is an option. Compilation requires Microsoft Visual C++, which is part of their Visual Studio suite. Microsoft distributes a free version named Visual C++ 2005 Express, which works fine. So does the standard commercial version of Visual C++, though you do need the 2003 edition or later.

Compiling Nmap on Windows from Source.

1. Download the latest Nmap source distribution from <http://www.insecure.org/nmap/download.html>. It has the name `nmap-version.tar.bz2` or `nmap-version.tgz`. Those are the same tar file compressed using gzip or bzip2, respectively. The Bzip2-compressed version is smaller.
2. Uncompress the source code file you just downloaded. Recent releases of the free Cygwin distribution can handle both the `.tar.bz2` and `.tgz`. Use the command **tar xvjf nmap-version.tar.bz2** or **tar xvzf nmap-version.tgz**, respectively. Alternatively, the common Winzip application can decompress the `.tgz` version.
3. Open Visual Studio and the Nmap solution file (`nmap-VERSION/mswin32/nmap.sln`)
4. From the Build Menu, select Configuration Manager and set Active Solution Configuration to Release. Use Debug instead if you are trying to debug a problem with Nmap.

Nmap Compilation, Installation, And Removal

By Fyodor

5. Choose Build Solution from the Build Menu. Nmap should begin compiling, and end with the line "- Done --" saying that all projects built successfully and there were 0 failures.
6. The executable and data files can be found in map-VERSION/mswin32/Release/ (or Debug). You can copy them to a preferred directory as long as they are all kept together.
7. Instructions for executing your compiled Nmap are the same as given previously for the .zip binaries. Take special note of the WinPcap requirement.

Many people have asked whether Nmap can be compiled with the gcc/g++ included with Cygwin or other compilers. Some users have reported success with this, though we don't actively maintain build instructions.

Sun Solaris

Solaris has long been well-supported by Nmap. Sun even donated a complete SPARCstation to the project, which is still being used to test new Nmap builds. For this reason, many Solaris users compile and install from source code as described in the section called "UNIX Compilation and installation from source code".

Users who prefer native Solaris packages will be pleased to learn that Steven Christensen does an excellent job of maintaining Nmap packages over at <http://www.sunfreeware.com>. Instructions are on his site, and are generally very simple: download the appropriate Nmap package for your version of Solaris, decompress it, and then run **pkgadd -d packagename**. As is generally the case with contributed binary packages, these Solaris packages are simple and quick to install. The advantages of compiling from source are that a newer version may be available and you have more flexibility in the build process. Certain optional features such as OpenSSL version detection are often not available in prebuilt packages.

Apple Mac OS X

Thanks to several people graciously donating shell accounts on their OS X boxes, Nmap usually compiles on that platform without problems. Doing this does require the Apple Developer Tools system. If you are not careful, Apple tries to charge for them. Brian Hatch sent me the following steps for obtaining the Developer Tools for free (as of September 2003).

1. Browse to <http://connect.apple.com> and join the ADC (Apple Developer Connection)
2. Fill out several forms to create a new account
3. Eventually you reach a page for buying support and/or CD media. Ignore this page and return to <http://connect.apple.com>.
4. Log in with your new account credentials.
5. Hit the Download link on the left and then choose Developer Tools.
6. Download the most recent Dev Tools and install.
7. Download the most recent Dev Tools Updates and install.

Verify that these steps have not changed

These exact steps may change, but it is hoped that this general approach will continue to work.

Once you have the developer tools installed, you can follow the compilation instructions found in the section called "UNIX Compilation and installation from source code". Note that on some older versions of Mac OS X, you may have to replace the command **./configure** with **./configure CPP=/usr/bin/cpp**.

Nmap Compilation, Installation, And Removal

By Fyodor

Users who prefer binary packages may want to have a look at the Fink project. Their stated goal is “to bring the full world of Unix Open Source software to Darwin and Mac OS X,” and so they offer Nmap and hundreds of other useful packages. As with all contributed binary packages, the disadvantage is that they may not be up-to-date with the latest Nmap releases and you have less flexibility in the build process. But it is certainly worth a look if you want to install many popular UNIX tools at once.

FreeBSD / OpenBSD / NetBSD

The BSD flavors are well supported by Nmap, so you can simply compile it from source as described in the section called “UNIX Compilation and installation from source code”. This provides the normal advantages of always having the latest version and a flexible build process. If you prefer binary packages, these *BSD variants each maintain their own Nmap packages. Many BSD systems also have a “ports” tree which standardizes the compilation of popular applications. Instructions for installing Nmap on the most popular *BSD variants follow.

OpenBSD binary packages and source ports instructions

According to the OpenBSD FAQ, users “are HIGHLY advised to use packages over building an application from ports. The OpenBSD ports team considers packages to be the goal of their porting work, not the ports themselves.”. That same FAQ contains detailed instructions for each method. Here is a summary.

Installation using binary packages

1. Choose a mirror from <http://www.openbsd.org/ftp.html>. FTP in and grab the Nmap package from `/pub/OpenBSD/version/packages/platform/nmap-version.tgz`. Or obtain it from the OpenBSD distribution CD-ROM.
2. As root, execute: **pkg_add -v nmap-version.tgz**

Installation using the source ports tree

1. If you do not already have a copy of the ports tree, obtain it via CVS using instructions at <http://www.openbsd.org/faq/faq8.html#CVS>.
2. As root, execute the following command (replace `/usr/ports` with your local ports directory if it differs):

```
cd /usr/ports/net/nmap && make install clean
```

FreeBSD binary package and source ports instructions

The FreeBSD project has a whole chapter in their Handbook describing the package and port installation processes. A brief summary of the process follows.

Installation of the binary package

The easiest way to install the binary Nmap package is to run **pkg_add -r nmap**. You can then run the same command with an `nmapfe` option if you want the X-Window front-end. If you wish to obtain the package manually instead, retrieve it from <http://www.freebsd.org/cgi/ports.cgi?query=nmap> or the CDROM and run **pkg_add packagename.tgz**.

Nmap Compilation, Installation, And Removal

By Fyodor

Installation using the source ports tree

1. The ports tree is often installed with the system itself (usually in /usr/ports). If you do not already have it, specific installation instructions are provided in the FreeBSD Handbook chapter referenced above.
2. As root, execute the following command (replace /usr/ports with your local ports directory if it differs):

```
cd /usr/ports/security/nmap && make install clean
```

NetBSD binary package instructions

NetBSD has packaged Nmap for an enormous number of platforms, from the normal i386 to Playstation 2, PowerPC, Vax, SPARC, MIPS, Amiga, ARM, and several platforms that I have never even heard of! Unfortunately they are not very up-to-date. A list of NetBSD Nmap packages is available from <ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/net/nmap/README.html> and a description of using their package system to install applications is available at <http://www.netbsd.org/Documentation/pkgsrc/using.html#id2956484>.

Amiga, HP-UX, IRIX, and Other Platforms

One of the wonders of Open Source development is that resources are often biased towards what people find exciting rather than having an exclusive focus on profits as most corporations do. It is along those lines that the Amiga port came about. Diego Casorran performed most of the work and sent in a clean patch which was integrated into the main Nmap distribution. In general, AmigaOS users should be able to simply follow the source compilation instructions in the section called "UNIX Compilation and installation from source code". You may encounter a few hurdles on some systems, but I presume that must be part of the fun for Amiga fanatics.

Nmap supports many proprietary UNIX flavors such as HP-UX and SGI IRIX. The Nmap project mostly depends on the user community to maintain adequate support for these systems. If you have trouble, try sending a report with full details to the nmap-dev mailing list (<nmap-dev@insecure.org>). If you develop a patch which improves support on your platform, please email it to nmap-dev or to me at <fyodor@insecure.org>.

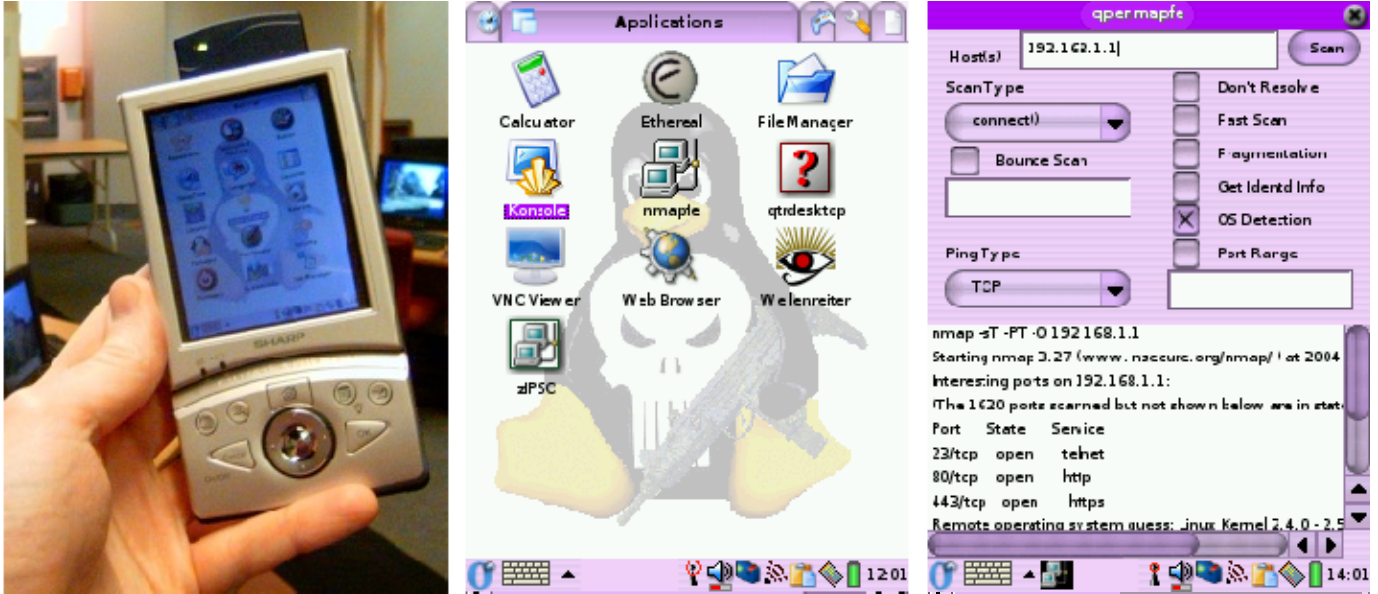
Installing Nmap on a PDA

Previous sections have described the installation of Nmap on notebook and desktop computers running a wide variety of operating systems. However, some users want greater portability and stealth than even the smallest notebook computers provide. They wish to do their security auditing from a personal digital assistant (PDA) small enough to fit in their pocket or to hide near an ethernet jack in a corporate office or datacenter. Walking around while using a notebook can raise eyebrows. With a PDA, passers by may assume you are just checking your calendar or shopping list while you locate insecure wireless access points or scan their internal network for vulnerabilities. Thanks in a large part to enthusiastic user communities, Nmap supports numerous PDAs. Two of the best supported are the Sharp Zaurus and Compaq IPAQ. Nmap has not been ported to PalmOS systems.

Figure 3. The Sharp Zaurus is an excellent platform for highly mobile security applications

Nmap Compilation, Installation, And Removal

By Fyodor



This recipe focuses on the Sharp Zaurus because it is the most popular PDA for running Nmap. Users of the Compaq IPAQ may wish to investigate the Familiar Linux distribution for similar functionality. Many other PDAs have active developer communities that are easily found with Google or through sites such as <http://www.handhelds.org>. The Zaurus is popular with mobile security auditors for many reasons.

Advantages of the Sharp Zaurus for hackers

- Keyboard (sliding or folding) allows easy use of Linux console commands
- Lightweight, compact form factor is convenient and inconspicuous
- Reasonably fast (200Mhz+) ARM processor and adequate RAM (32MB+) provide plenty of power for running Nmap and other security tools
- A wide variety of CF networking cards are supported without bulky adapters. Secure Digital cards are also supported for extra flash storage.
- Ships with Linux pre-installed, making it compatible with a wide variety of popular free security tools (and other software).
- The OpenZaurus project provides convenient support for Nmap, NmapFE, and many other security tools

Many thanks go to Kevin Milne, Adrian Crenshaw (AKA IronGeek), and David Malcher (KillingJoke), avid Zaurus users who provided much of the content and screenshots for this recipe.

Installing Nmap on the Sharp Zaurus

Before beginning, make sure you have sufficient hardware.

System Requirements

- A Sharp Zaurus (any model)
- 64MB or larger Compact Flash (CF) card for the OpenZaurus ROMS
- A CF networking card such as a basic ethernet card and/or wireless 802.11X. Wireless cards with the Prism2 chipset are recommended. Kevin uses a Xircom 10MB ethernet card and a Netgear

Nmap Compilation, Installation, And Removal

By Fyodor

MA701 wireless card. Adrian uses an Ambicom WL1100C-CF Wi-Fi card and an TRENDnet/TRENDware TE-CF100 ethernet card.

The most common way to install Nmap is using the OpenZaurus project. They provide an alternative ROM image (Linux kernel and filesystem) with a greater emphasis on development and open source tools than the ROM Sharp provides. OpenZaurus is based on the popular Debian Linux distribution. Many other Zaurus Linux distributions are available to suit different needs and preferences. The OpenZaurus project may be subsumed by the more general OpenEmbedded distribution.

Rather than describe the installation process here, readers are advised to follow the directions in the OpenZaurus Install Guide available from http://www.openzaurus.org/oz_website/content/installguide. Follow those directions carefully to avoid damaging your Zaurus.

Once OpenZaurus is installed, thousands of open source applications are available for easy installation as IPK files (the file extensions should be .ipk). These are available for download from the OpenZaurus site, or a number of 3rd party sites such as <http://www.killefiz.de/zaurus/>. While finding IPK files on the Internet is quite convenient, they are not always up-to-date. At the time of this writing, the latest IPK of Nmap available via the sites OpenZaurus.Org and Killefiz.de is almost 2 years old. A bit of Internet searching turned up IronGeek's excellent resource site at <http://www.irongeek.com/all.php>. He includes instructions for installing the very latest Nmap version.

After downloading IPK files to the Zaurus, they can be installed with the ipkg program. An example execution would be **ipkg install nmap_3.27-1_armv4l-strongarm.ipk**. Type **ipkg** with no arguments for help.

A convenient alternative to manually downloading and installing IPK files is the Zaurus Package Manager. It makes installing a large number of packages simple and quick.

Installing Nmap using the Zaurus Package Manager

1. Click on the Settings tab and choose Packages.
2. The first time you start the Package Manager, activate the feeds through the Options -> Configure screen and configure the stable, testing, and/or unstable feeds. Nmap is currently part of the testing feed.
3. Since the available packages are frequently updated, perform a feed update by choosing Actions and then Update Lists.
4. A huge list of available software is provided. You may have to switch to the testing feed to obtain Nmap and NmapFE. NmapFE is the official UNIX GUI frontend that is distributed with Nmap. What OpenZaurus.Org calls NmapFE is actually Qpenmapfe, a simplified clone written by Dennis Webb. Scroll down the list of software packages and check Nmap, NmapFE, and anything else that catches your fancy. Many excellent security tools are available.
5. After selecting all of the appropriate software, click the GO (green arrow) icon on the top right hand of the screen. The installation manager screen shows the progress of software download and installation.

Using Nmap and NmapFE on the Sharp Zaurus

Once NmapFE and Nmap have been properly installed, a new NmapFE icon should appear in the Applications menu. If it does not appear, try restarting Opie. Simply click the icon to use it. If you prefer the command-line version of Nmap, start the console and execute the appropriate command. The screenshots at the top of this recipe demonstrate both methods. Except for a simplified option set

Nmap Compilation, Installation, And Removal

By Fyodor

in qpenmapfe, usage of Nmap is the same as on any other computer. The following figures show another type of Zaurus (the SL-C760) and how to run Nmap on it.

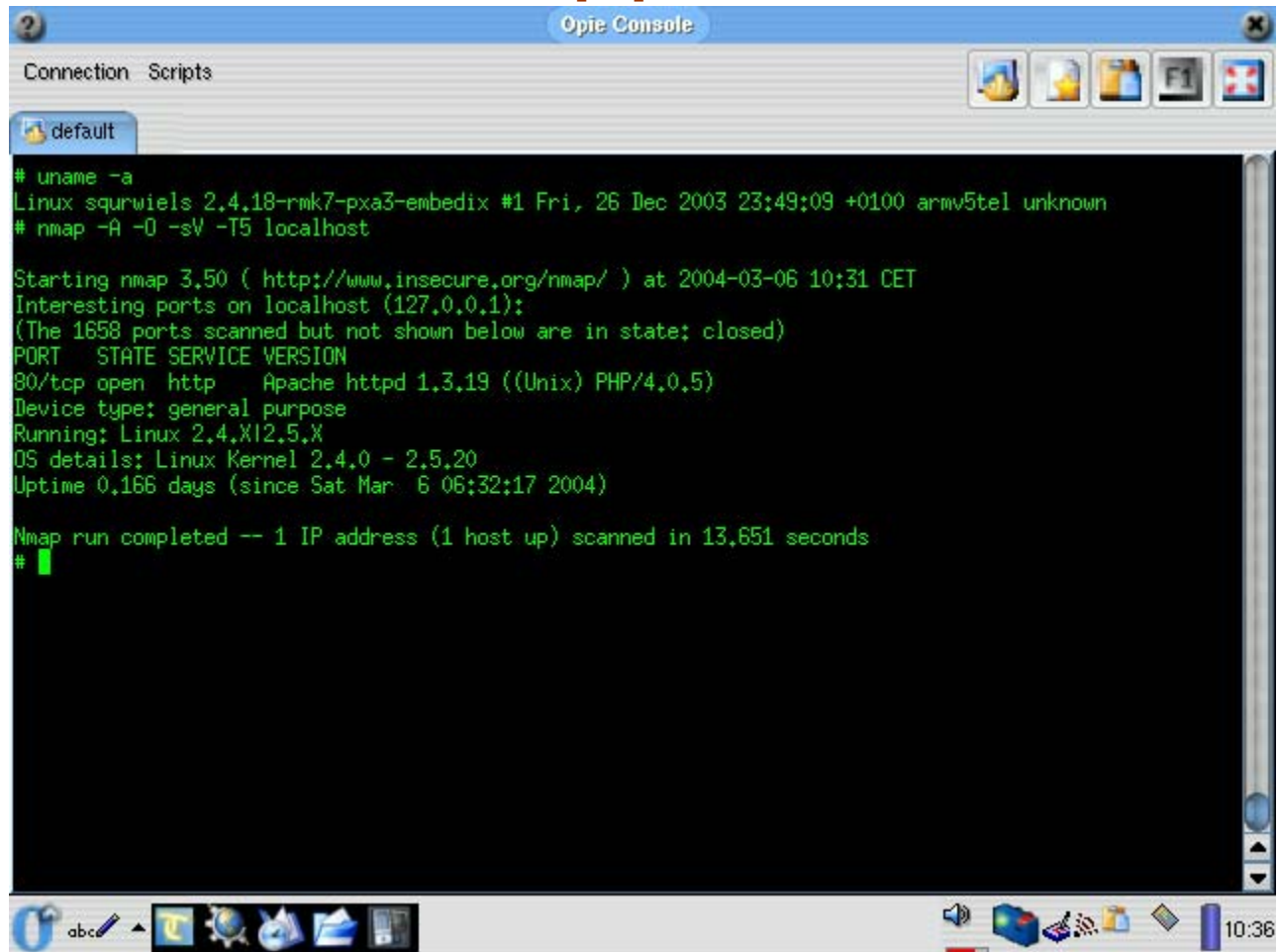
Figure 4. The Sharp Zaurus SL-C760 PDA



Figure 5. The SL-C760 executing Nmap in a terminal window

Nmap Compilation, Installation, And Removal

By Fyodor



```
Optix Console
Connection Scripts
default
# uname -a
Linux squirwiels 2.4.18-rmk7-pxa3-embedix #1 Fri, 26 Dec 2003 23:49:09 +0100 armv5tel unknown
# nmap -A -O -sV -T5 localhost

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-03-06 10:31 CET
Interesting ports on localhost (127.0.0.1):
(The 1658 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 1.3.19 ((Unix) PHP/4.0.5)
Device type: general purpose
Running: Linux 2.4.X12.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.166 days (since Sat Mar  6 06:32:17 2004)

Nmap run completed -- 1 IP address (1 host up) scanned in 13.651 seconds
# █
```

Removing Nmap

If your purpose for removing Nmap is simply to upgrade to the latest version, you can usually use the "upgrade" option provided by most binary package managers. Similarly, installing the latest source code (as described in the section called "UNIX Compilation and installation from source code") generally overwrites any previous from-source installations. Removing Nmap is a good idea if you are changing install methods (such as from source to RPM or vice versa) or if you are not using Nmap anymore and you care about the few megabytes of disk space it consumes.

How to remove Nmap depends on how you installed it initially (see previous sections). Ease of removal (and other maintenance) is a major advantage of most binary packages. For example, when Nmap is installed using the RPM system common on Linux distributions, it can be removed by running the command **rpm -e nmap nmap-frontend** as root. Analogous options are offered by most other package managers -- consult their documentation for further information.

If you installed Nmap from source code, removal is slightly more difficult. If you still have the build directory available (where you initially ran **make install**, you can remove Nmap by running **make uninstall**. If you no longer have that build directory, type **nmap -V** to obtain the Nmap version number. Then download that source tarball for that version of Nmap from <http://download.insecure.org/nmap/dist/>. Uncompress the tarball and change into the newly created directory (nmap-VERSION). Run **./configure**, including any install-path options that you specified the

Nmap Compilation, Installation, And Removal

By Fyodor

first time (such as --prefix or --datadir). Then run **make uninstall**. Alternatively, you can simply delete all the Nmap-related files. If you used a default source install of Nmap versions 3.00 or higher, the following commands remove it.

```
# cd /usr/local
# rm -f bin/nmap bin/nmapfe bin/xnmap
# rm -f man/man1/nmap.1 man/man1/nmapfe.1 man/man1/xnmap.1
# rm -rf share/nmap share/gnome/apps/Utilities/nmapfe.desktop
```

You may have to adjust the above commands slightly if you specified --prefix or other install-path option when first installing Nmap. The files relating to nmapfe/xnmap do not exist if you did not install the NmapFE frontend initially.