Version 1.0

David Fifield (david@bamsoftware.com)

Nmap, the well-known free security scanner, lately comes with a set of companion tools. Zenmap is a graphical user interface for Nmap, with the ability to combine scans and create a visualization of the network. Ncat is a flexible, modern replacement for the Netcat utility. Ndiff is diff for Nmap scans; it takes two Nmap XML output files and displays the differences between them. NSE, The Nmap Scripting Engine, uses external scripts to gather more information about hosts and ports. This paper describes these tools and how to make the best use of them in combination with Nmap.

Work and play with Nmap and friends

Nmap is a free network security scanner. It is best known as a port scanner, but it also has a variety of related abilities that complement this basic function. These include OS detection, service version identification, and custom script scanning. It comes with some companion tools, Nmap's "little brothers," that complement it in its purpose of network exploration and security auditing.

The tools are: Zenmap, a graphical frontend; Ncat, a reimplementation of Netcat; and Ndiff, a scan comparison tool. These are all included in the source code distribution and binary packages. Go to http://nmap.org/download.html and follow the instructions for your platform.

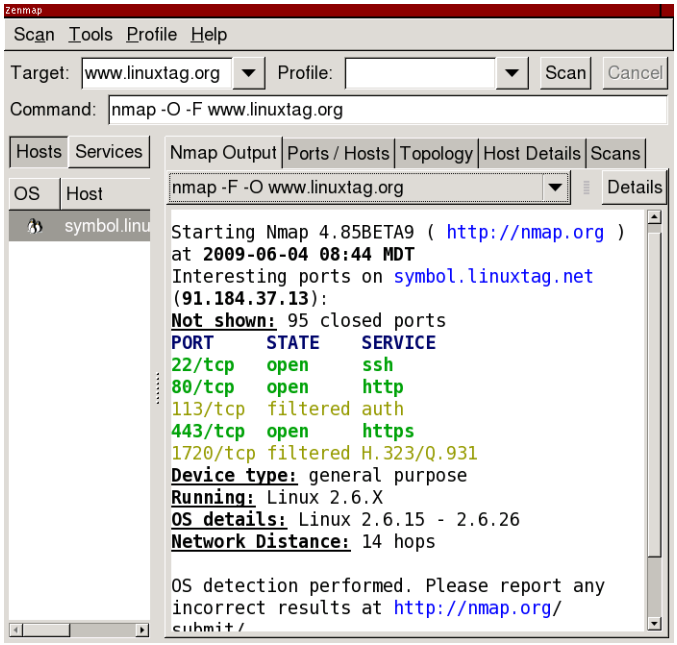
Readers not familiar with the capabilities and purpose of Nmap will want to gain some background by reading some of the documentation from http://nmap.org/docs.html, installing the program, and trying a few sample scans. For now, this sample will give a taste of what is possible with Nmap and motivation for the discussion of the auxiliary tools.

```
# nmap -F -O www.linuxtag.org
Starting Nmap 4.85BETA9 ( http://nmap.org )
Interesting ports on symbol.linuxtag.net (91.184.37.13):
Not shown: 95 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
113/tcp  filtered auth
443/tcp  open     https
1720/tcp filtered H.323/Q.931
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15 - 2.6.26
Network Distance: 14 hops

Nmap done: 1 IP address (1 host up) scanned in 5.38 seconds
```

In just a few seconds, Nmap has sent a flurry of packets to a remote host: a SYN request to each of the 100 ports most likely to be open and a dozen or so crafted probes designed to uncover differences in TCP/IP implementations. There are three open ports (ssh, http, and https), and the host is likely running a recent revision of Linux. Even more information can be requested with different command line options.

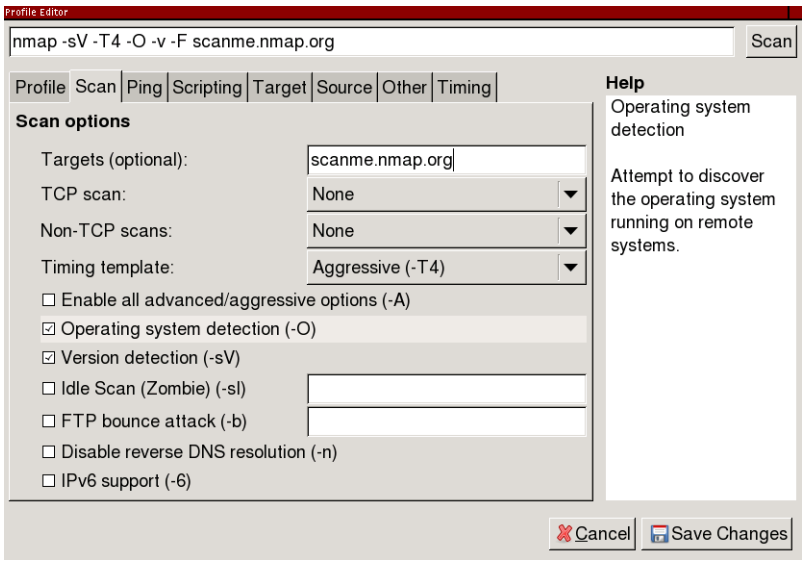
Zenmap, Nmap’s graphical user interface



Why have a graphical frontend for Nmap, a program whose users are more likely than most to be partisans of the command line? One reason is to open the door to beginners, so that you don’t have to know the command line before you can start using Nmap. Many of Zenmap’s features are geared towards the beginner, like interactive command line construction and preset scan profiles. But Zenmap also aims to offer compelling advantages over plain Nmap, like scan aggregation and topology viewing, so that even advanced users will want to use it for some purposes.

Zenmap’s most basic ability is to run Nmap scans and open scan results saved in Nmap’s XML output format. It offers syntax highlighting of text output and interactive browsing of results.

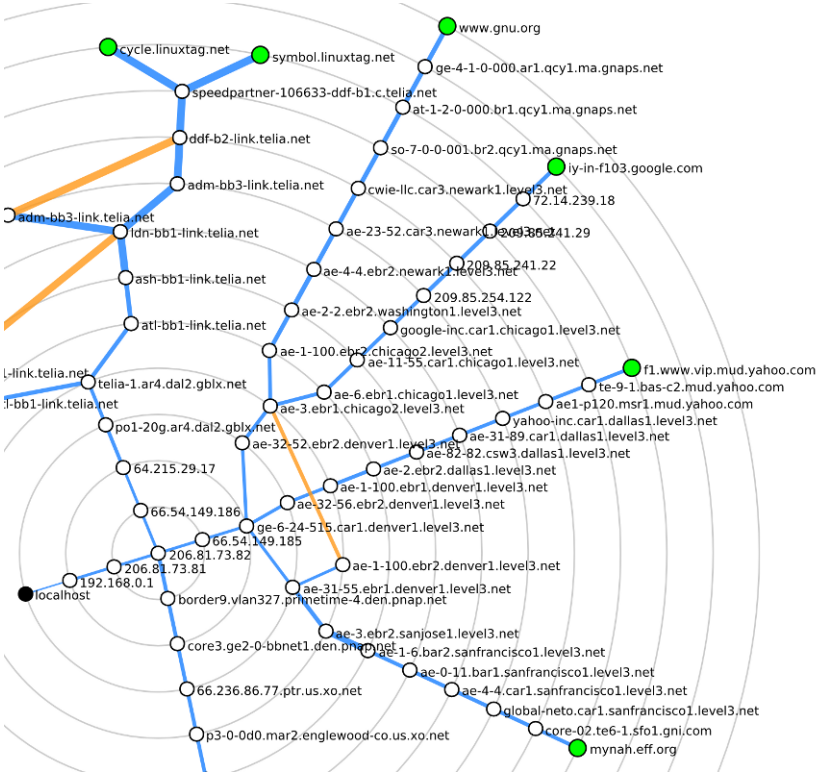
A conspicuous feature of Zenmap is its scan profiles. These allow you to save an Nmap command line and run it several times, possibly against different targets. Zenmap ships with profiles for lots of typical scans, with names like “Quick scan,” “Ping scan,” and “Intense scan.” These is aimed at being helpful for beginners, who don’t know the options for the scan they want to run, but also for advanced users, who can save a canned scan as a profile to run it again and again.



A beginner-friendly feature is the profile editor, which allows interactive building of an Nmap command line. All of Nmap's most important options have a clickable control and help text; clicking the control adds or removes the option from the command line in progress. Once the command is ready, it can be run immediately or saved as a profile. It is hoped that the profile editor has the side effect of slowly turning beginners into experts, by always displaying the command line as it is built up and establishing a mental correlation between the options and their descriptions.

Experienced users will like a feature called scan aggregation. When several scans are run in a single window, their results are combined into one cohesive view that shows all the hosts and ports that are in any of the scans. In other words, there is no need to page through scan results one at a time; results are presented as if they were from one big "meta-scan" that had collected all the information. You can, for example, ping scan a large network, then scan one selected host more intensively. That one host will have detailed port scan results, while the others will continue to have just an up/down indicator and a reverse DNS name. Similarly, you can scan two different networks separately, then view them together. You can even have several scans running at the same time; as each one finishes, its results will be added into the aggregation. To disable scan aggregation, just run scans in separate windows.

Zenmap can draw an interactive map of the network, called the "Topology" in the user interface. This function works best with scans that include route information; use Nmap's `--traceroute` option (included in the "Quick traceroute" scan profile). Scan aggregation causes the topology to be updated whenever a new scan is added.



The topology display shows the scanning host, the targets, and all the intermediate hosts that traceroute was able to find. Clicking on any host recenters the graph to radiate outward from the host selected, showing the network from its point of view. Above is a map of the routes to several major web servers from the author's home computer.

Zenmap, at the time called Umit, was first distributed with Nmap in version 4.22SOC1 in July 2007.

> http://nmap.org/zenmap/
> http://nmap.org/book/zenmap.html

Ncat, a featureful Netcat

Ncat is a general-purpose network connection and troubleshooting tool inspired by the Netcat program and its many derivatives. Its most basic function is to operate as a raw client or server, relaying data between a socket and its own standard input and output. It is suitable for interactive use, for example in debugging a mail server, or as a backend network connector for non-network-aware programs. Ncat supports all the features of traditional Netcat programs and adds some innovations. Connections may be made over TCP, UDP, or SSL; over IPv4 or IPv6. The client is able to connect through a SOCKS or HTTP proxy, and the server itself can act as an HTTP proxy.

Ncat can work as a network client (connect mode) or as a server (listen mode). Connect mode is the default; listen mode is enabled with the `-l` option. In connect mode you specify the host and port to connect to. In listen mode provide the address and port to listen on. If the port is not provided, it defaults

to 31337.

Elementary usage is discussed in the online Ncat Users' Guide, and most online Netcat tutorials will be applicable. This paper presents uses that are fun, unusual, or unique to Ncat.

Normally the Ncat server reads data from the network and writes it to stdout. A slight modification to this behavior enables new creative uses. In "connection brokering" mode, enabled by the `--broker` option, Ncat accepts connections from multiple clients. Any message received by one client is not written to stdout, but rather broadcast to all other connected clients (clients do not get their own messages).

What is brokering good for? One thing is to circumvent firewalls that don't allow incoming connections. Suppose that host1 wants to send a file to host2, but both are behind firewalls such that neither can connect directly to the other. If there is a host3 out on the Internet, it can broker a connection between the two other hosts. Transferring a file would work like this:

```
host3$ ncat -l --broker
host2$ ncat host3 > outputfile
host1$ ncat host3 < inputfile
```

host3 acts like a network hub, copying data received from host1 and sending it to host2 (and any other clients that happen to be connected). Anything that host2 sends will be relayed back to host1 as well.

A trivial hack to connection brokering allows Ncat to work as a (very) rudimentary chat server.

```
$ ncat -l --chat
```

Each connecting client receives a unique ID. The server prefixes each message received with the ID before broadcasting it to the other clients. This makes it easier to determine who is saying what. Here is an example transcript between two users:

```
$ ncat localhost
<announce> 127.0.0.1 is connected as <user5>.
<announce> already connected: 127.0.0.1 as <user4>.
<user4> Guten Tag.
Was ist los?
<user4> Wo gibt es viele Sehensw\303\274rdigkeiten?
<announce> <user4> is disconnected.
```

It is a pity that non-ASCII characters are escaped, but because it is assumed that `--chat` output is written directly to a terminal, anything that might be a control character is sanitized to prevent various terminal-based security exploits.

Most of Ncat's operating modes support securing connections with SSL. To

connect to an SSL server, just do

```
$ ncat --ssl host port
```

By default no certificate verification is done, so the connection is vulnerable to man-in-the-middle attacks. To enforce verification, use `--ssl-verify` instead:

```
$ ncat --ssl-verify host port
```

An SSL server on the default port is started with

```
$ ncat -l --ssl
```

This generates a temporary private key and certificate. To use an established certificate, use the `--ssl-cert` and `--ssl-key` options.

Here is how you might read your email securely over SSL using Ncat. The `-C` option transforms the line ending characters you type into the CRLFs required by many Internet protocols.

```
$ ncat -v -C --ssl-verify pop.gmail.com 995
Ncat version 4.85BETA9 ( http://nmap.org/ncat )
SSL connection to 209.85.201.111:995. Google Inc.
SHA-1 fingerprint: 5121 45CE CE99 1987 7DCE 3F52 C031 0F7E FBB4 6A6F
+OK Gpop ready for requests from 66.7.171.173 27pf10591584wff.25
```

Ncat can invoke an external program and relay its standard input and output to a socket. With certain limitations, mostly related to buffering, this makes it possible to use terminal applications over the network. (Use this feature with care!) When acting as a server, Ncat will accept multiple simultaneous connections and handle them independently, making it work like a one-port inetd. All kinds of creative uses are possible. One section of the Ncat Users' Guide shows how to implement several TCP services in as little as one line. For example, here is a TCP echo server:

```
# ncat --listen 7 --exec "/bin/cat"
```

Anything received on the socket is given to cat on stdin; cat writes it back to stdout which is connected back to the socket. More complicated operations are possible, too. Here we use Perl to create a "shout server" that capitalizes its input:

```
$ ncat -l --sh-exec 'perl -e "$| = 1; while (<>) { print uc; }"'
```

The `$| = 1` is necessary to disable block buffering. Here is an example interaction with the server.

```
$ ncat localhost
Hallo.
HALLO.
Wo gibt es viele Sehenswürdigkeiten?
```

```
WO GIBT ES VIELE SEHENSWÜRDIGKEITEN?
```

This command execution ability can be used to “unwrap” SSL services so they can be used by clients without SSL support. Suppose you have a mail client that supports POP but not SSL. Here is how to establish an SSL tunnel to the mail server, connected to an unencrypted local port.

```
$ ncat -l 995 --sh-exec "ncat --ssl-verify pop.gmail.com 995"
```

After doing this, configure the mail client to connect to pop3://localhost:995. When the local Ncat receives a connection on port 995, it will spawn another Ncat to communicate with pop.gmail.com over SSL. The new Ncat will relay all data over the secure connection. This is a subset of the functionality provided by a program like stunnel.

A similar technique makes Ncat work as an IPv4-to-IPv6 gateway.

```
$ ncat -4 -l <port> --sh-exec "ncat -6 <host> <port>"
```

Ncat was first distributed with Nmap in version 4.85BETA1 in January 2009. Ncat has a complete users’ guide covering everything described here and more.

> http://nmap.org/ncat/
> http://nmap.org/ncat/guide/

Ndiff, a scan comparison utility

Ndiff is a tool for comparing Nmap scans. Given two Nmap scan logs, it shows how they differ: what hosts came up or went down, which ports became open or closed, DNS name changes, changes to server software (when available with Nmap’s `-sV` option), and changes in operating system (when available with `-O`). It is designed to work just like the diff utility that compares text files. This example shows how a few random Internet hosts changed over a day (the names and addresses are fictitious).

```
$ ndiff -v scan-1.xml scan-2.xml
-Nmap 4.85BETA4 at 2009-03-24 17:34
+Nmap 4.85BETA4 at 2009-03-25 16:35

+10.181.218.66:
+Host is up.
+Not shown: 998 closed ports
+PORT     STATE    SERVICE    VERSION
+222/tcp  open     rsh-spx
+8080/tcp filtered http-proxy

-utkjlegbx-701.example.com (10.196.172.89):
+cdgzhwik-216.example.com (10.196.172.89):
 Host is up.
```

```
 Not shown: 995 filtered ports
 PORT     STATE  SERVICE    VERSION
 21/tcp   open   ftp        Dreambox ftpd
 80/tcp   open   http       Dreambox httpd

-bpdygf-130.example.com (10.188.226.230):
-Host is up.
-Not shown: 1000 filtered ports
```

Ndiff works on the XML output format of Nmap, so be sure to use `-oX` (XML output) or `-oA` (all output formats) to produce `.xml` files. The XML output format is also used by Zenmap, and various third-party parsers for it exist.

Here is a use case for Ndiff: A system administrator could set up a script to scan a network with Nmap once a day, then mail a report containing the diff against the previous day, week, or month. The report would contain information useful to someone managing a network: hosts going down (where did the mail server go?), hosts coming up (where did that wireless access point come from?), and hosts offering new services (hmm, the firewall should have blocked that). The example at the end of this section shows how to do it.

Another use for Ndiff is in learning to use Nmap. Scan a network using the default options, then scan it again with `-F` (fast scan of only 100 ports). Use Ndiff to see if any ports were missed in the fast scan. Run a service scan with `-sV`, then run another, adding the option `--version-all`, and use Ndiff to see if the more thorough version detection discovered anything additional.

Ndiff was first distributed with Nmap in version 4.85BETA1 in January 2009.

> http://nmap.org/ndiff/

Ndiff example

It is easy to combine Nmap, Ndiff, and cron to scan a network daily and produce a report of differences. A shell script runs the scan and compares it to the previous one. This script is based on one created by Fyodor, the author of Nmap.

```
#!/bin/sh
date=`date +%F`
cd /root/scans
nmap -v -T4 -F -sV -oA scan-$date targets > /dev/null
if [ -f scan-prev.xml ]; then
     ndiff scan-prev.xml scan-$date.xml > diff-$date
     echo "*** NDIFF RESULTS ***"
     cat diff-$date
     echo
fi
echo "*** NMAP RESULTS ***"
cat scan-$date.nmap
```

```
ln -sf scan-$date.xml scan-prev.xml
```

Add a line to your crontab to run the script periodically. cron will mail the script's output to the user who scheduled it. This line makes the script run daily at 12:00.

```
0 12 * * * /root/scan-ndiff.sh
```

NSE, the Nmap Scripting Engine

The Nmap Scripting Engine, or NSE, while not exactly new, is still not as well known as some other Nmap features and has been undergoing many changes and rapid growth. NSE is an embedded Lua interpreter and a set of network-specific libraries. A general-purpose programming environment with access to Nmap's data structures enables more detailed information gathering than just a port scan.

As a simple example of what a script can do, consider `ftp-anon.nse`. This script is activated when Nmap has found open port 21, or any port that service detection has identified as FTP. All it does is try to log in to the FTP server anonymously, and report if it was successful. This may be compared to a much more complex script like `smb-check-vulns.nse`, which uses multi-thousand-line libraries to build custom MSRPC requests and check for a variety of Windows vulnerabilities.

To activate NSE, use the `-sC` option to run just the default scripts, or use `--script` with a list of the scripts you want to run.

```
# nmap -sC target
# nmap --script=ftp-anon,smb-check-vulns target
```

A list of all scripts that come with Nmap is online at http://nmap.org/nsedoc/. That site contains documentation for for both users and developers of NSE scripts. As of this writing, there were 58 scripts and 32 libraries.

NSE gained some press exposure in April 2009 when `smb-check-vulns.nse` became able to remotely detect infections by the Conficker worm. Nmap was one of the first programs able to do this after the technique was discovered. The command to run is

```
# nmap -PN -T4 -p139,445 -n -v --script=smb-check-vulns \
  --script-args safe=1 networks
```

Implementing the detection in NSE means that it automatically benefits from Nmap's multiple target specification and NSE's parallelism, with no extra effort on the part of the script author. The script itself only needs to work against one port on one host. NSE takes care of the rest, reducing the effort needed to turn a proof of concept into a workaday scanner. It is hoped that NSE will become a preferred means of security researchers to distribute new vulnerability detection and exploit code.

NSE was first released with Nmap 4.21ALPHA1 in December 2006, but it been

revised and updated extensively since then.

http://nmap.org/book/nse.html
http://nmap.org/nsedoc/

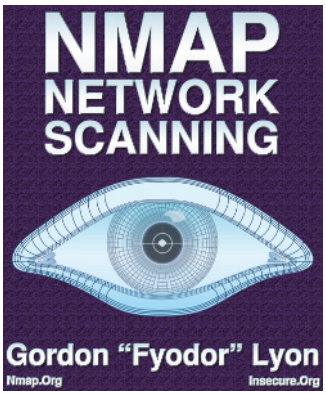
Future work

More additions are planned to the Nmap family. Two new applications are under development: Nping, a flexible raw packet sending program in the spirit of hping; and Ncrack, an efficient authentication cracker supporting many protocols.

More information and support

The Nmap Reference Guide (man page) is detailed and complete, and kept up to date despite the fast pace of development. Type **`man nmap`** or read it online.

http://nmap.org/book/man.html



The Nmap book, *Nmap Network Scanning*, was published in January 2009. It contains not only an in-depth treatment of Nmap itself, but also general information on network scanning and security. About half of the chapters are available free online, including those covering OS detection, the scripting engine, and Zenmap. A German translation is available since May 2009.

http://nmap.org/book/
http://www.opensourcepress.de/index.php?26&tt_products=270

Most project discussion and user support occurs on the development mailing list, nmap-dev@insecure.org. The list receives around 200–300 messages per month. It is the preferred place to send bug reports and support questions. Another list is nmap-hackers@insecure.org, a read-only list used for major announcements; it receives less than one message per month. Subscribe to the lists or read the archives at http://seclists.org/.