

Network Management Workshop – APRICOT 2004, Malaysia



Nmap

www.insecure.org/nmap/

- it's a Network MAPper
- powerful utility for network exploration or security auditing
- rapidly scan large networks or single host
- determine what hosts are available on the network
- what services (ports) they are offering
- what operating system (and OS version) they are running
- what type of packet filters/firewalls are in use
- runs on most types of computers
- both console and graphical versions are available
- free software, available with full source code - GNU GPL

Network Management Workshop – APRICOT 2004, Malaysia

◆ Nmap Features

- ◆ Flexible - Supports advanced techniques for mapping out networks filled with IP filters, firewalls, routers
- ◆ Port scanning mechanisms - TCP & UDP, OS detection, pings sweeps
- ◆ Powerful - scan huge networks of hundreds of thousands of machines
- ◆ Portable - Most operating systems are supported – Linux, BSD, MacOS
- ◆ Easy – can be used with simple commands or GUI options
- ◆ Free – freely downloadable, comes with full source code, GNU GPL
- ◆ Well documented - comprehensive and up-to-date man pages, whitepapers, and tutorials
- ◆ Supported – well supported by the author
- ◆ Acclaimed - has won numerous awards, featured in magazines
- ◆ Popular – thousands download everyday, included in many OS distros

Network Management Workshop – APRICOT 2004, Malaysia

◆ Using Nmap

NMAP does three things:

- 1 - ping a number of hosts to determine if they are alive or not
 - 2 - portscan hosts to determine what services are listening
 - 3 - attempt to determine the OS of hosts
- NMAP is very configurable, and any of these steps may be omitted
 - Although portscanning is necessary in order to do an OS scan
 - There are multiple ways to accomplish most of these
 - Many command line switches to tweak the way that NMAP operates

Network Management Workshop – APRICOT 2004, Malaysia

◆ Nmap network setup



Nmap client

Scan hosts/services



Mail
server



DNS
server



Web
server

Network Management Workshop – APRICOT 2004, Malaysia

◆ Using Nmap

Target Selection

- ◆ Specify targets on the command line or in a filename with the -i option
- ◆ Range of hosts - cert.org/24, 192.88.209.5/24, 192.88.209.0-255

Ping Scans

- ◆ Default behavior - ICMP ping sweep and TCP port 80 ACK ping sweep
- ◆ ICMP ping sweep - the usual kind of ping, -PI
- ◆ TCP port ACK ping sweep - sends an ACK to port, expects a RST, -PT
- ◆ random high-numbered port may work *much* better thru firewalls
- ◆ both an ICMP ping scan and an ACK scan to a high port, -PB32523
- ◆ intelligent firewall may block your “illegal” ACK packet
- ◆ then you may do a TCP SYN sweep with -PS
- ◆ Try ICMP pings, if not TCP ACK pings, if not TCP SYN pings...

Network Management Workshop – APRICOT 2004, Malaysia

◆ Using Nmap

Port Scanning

The vanilla scan is a TCP connect() scan (-sT) - loggable – don't use this

- ◆ **SYN** scans (-sS) - workhorse of scanning methods
also called "half-open" scans -
send a SYN packet, look for the return SYN|ACK (open) or RST (closed) packet and then you tear down the connection before sending the ACK that would normally finish the TCP 3-way handshake
They are also harder to detect, packet filters like ipfwadm, firewall can
- ◆ **FIN** (-sF), **NULL** (-sN) and **XMAS** (-sX) scans are all similar
work by getting a RST back (closed) or a dropped packet (open)
- ◆ **UDP** scanning (-sU) - packet-filtered ports turn up as being open ports
runs extremely slowly against machines with UDP packet filters

Network Management Workshop – APRICOT 2004, Malaysia

◆ Using Nmap

Port Scanning

The vanilla scan is a TCP connect() scan (-sT) - loggable – don't use this

- ◆ **SYN** scans (-sS) - workhorse of scanning methods
also called "half-open" scans -
send a SYN packet, look for the return SYN|ACK (open) or RST (closed) packet and then you tear down the connection before sending the ACK that would normally finish the TCP 3-way handshake
They are also harder to detect, packet filters like ipfwadm, firewall can
- ◆ **FIN** (-sF), **NULL** (-sN) and **XMAS** (-sX) scans are all similar
work by getting a RST back (closed) or a dropped packet (open)
- ◆ **UDP** scanning (-sU) - packet-filtered ports turn up as being open ports
runs extremely slowly against machines with UDP packet filters

Network Management Workshop – APRICOT 2004, Malaysia

◆ Nmap lab

- ◆ Scan all reserved TCP ports on target.example.com in verbose mode

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

- ◆ Launches a stealth SYN scan against 255 hosts in target's network with OS detection - requires root privileges

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

- ◆ Launch a stealth scan with OS detection on specified ports against 255 hosts in the network, in verbose mode

```
nmap -ss -O -v 192.168.10.0/24 -p '1-1024,1080,3128'
```

- ◆ Launch a stealth scan with OS detection on all privileged ports against 255 hosts in the network, output the results into the file /root/nmap.scan

```
nmap -ss -O 192.168.10.0/24 -oN /root/nmap.scan
```


Network Management Workshop – APRICOT 2004, Malaysia



Ndiff

www.vinecorp.com/ndiff/

- compares two nmap scans and outputs the differences
- allows monitoring of your network(s) for interesting changes in port states and visible hosts
- eliminates the need to examine voluminous raw scan output in search of the few noteworthy differences
- useful to network administrators to monitor large networks in an organized fashion
- known to work on Linux/x86, other POSIX/UNIX platforms
- requires perl 5.005_03 or later and nmap 2.53 or later
- supports HTML output for viewing results

Network Management Workshop – APRICOT 2004, Malaysia

◆ Ndiff usage

Use the machine-parseable output of two nmap runs on the same net:

```
nmap -m first_scan.nm 10.0.0.0/24
```

later...

```
nmap -m second_scan.nm 10.0.0.0/24
```

OK, now we have two scans of the same net at different moments in time.
Now to see the changes:

```
ndiff -baseline first_scan.nm -observed second_scan.nm
```

We designate **first_scan** as the ``**baseline**'' for comparison.
Changes are reported as differences from **first_scan**.

Network Management Workshop – APRICOT 2004, Malaysia

◆ Ndiff results

... ndiff outputs: ...

missing hosts:

< hosts present in first_scan, but missing in second_scan >

new hosts:

< hosts present in second_scan, but missing from first_scan >

changed hosts:

< hosts present in both scans, but whose port states have changed >

[for each host, a list of changes in port states]

Ndiff has additional options, features for controlling output detail & format

Network Management Workshop – APRICOT 2004, Malaysia



Ngen

www.vinecorp.com/ndiff/ngen.html

- synthetically create baseline nmap results
- Using the results of a previous scan as your baseline for comparison is fine for many purposes
- but if you never knew or liked the state of the scanned net
- the previous scan probably didn't yield a satisfactory baseline
- you really want as your baseline is a description of your ideal net
- one which reflects your firewall rules and/or security policy

- Ngen accepts host and port specifications, and outputs an equivalent nmap scan result to be used as an ndiff baseline
- Comparisons with this output then will show how your net varies from your ideal net

```
ngen -o baseline.nm -h 10.0.2.128/25:80 -h  
10.0.2.144-150:22,53,53u
```

Network Management Workshop – APRICOT 2004, Malaysia



Nrun

www.vinecorp.com/ndiff/nrun.html

- housekeeping tasks necessary to use ndiff in an automated fashion via a cron job
- execute nmap
- name the results something reasonable
- store the results somewhere reasonable
- optionally run ndiff against those results to generate a report