Know your network

NebraskaCERT
May 2006 CSF

by

Aaron Grothe
CISSP/Security+

Near Random Sun Tzu Quote

- The general who wins the battle makes many calculations in his temple before the battle is fought. The general who loses makes but few calculations beforehand

Items/Agenda

- Intro
- Why?
- Description of 3 Tools
  - localscan
  - pbnj
  - ndiff
- Resources
- Contact

Why should I know what ports are Open?

- Many crackers will leave a process listening so they can get back in easily
- This is basically how the armies of zombie PCs that do the periodic DdoS attacks work
- Can also help tighten down network devices you get from other companies
  - E.g. custom appliances with telnet and ftp enabled
- Nmap is pretty much the standard tool for doing a scan

Nmap

- Nmap is an amazing tool
  - It is even used in the Matrix Reloaded
- Nmap can tell you the following about a server
  - What ports are open
  - What ports are closed
  - Used with amap can also tell you the versions of the software running on the port

Nmap Shortcomings

- Nmap has no history/diff functionality
  - was not designed for this
- Usually you run nmap then “grep -v” out the results you don't want. E.g. Server X has port 80 open so ignore that.
- Nmap's syntax is good, but isn't the most scalable

Tools to Diff/Automate Nmap

- 3 Major Tools exist
  - LocalScan
  - Ndiff
  - PBNJ
- All of these tools provide the following to nmap
  - Ability to diff the results of 2 scans
  - Ability to create a baseline for scans
  - Send summary output to admins

Why?

- New ports open
  - On a server peculiar
  - On a client brings up a lot of questions
  - Internally on a test/dev box is interesting
  - On a production box in the DMZ is very interesting
  - On the firewall is nuclear
- New machines appearing on the net
- Machines that appear on scans intermittently

LocalScan

- Written in Perl
- Reduces amount of output from nmap
  - Uses ignore/drop list for this
- Can create a baseline by either running a config script or creating a custom localscan.conf file

Using LocalScan

perl make_conf.pl
What subnet (specified by nmap scheme) do you want to scan? 192.168.0.0
What's your e-mail address?
admin@mainmachine.org
Do you want to receive “all clear” messages (Y/N)? Y
Where is nmap located (path only)? /usr/bin

You are now ready to run localscan.pl

Example LocalScan.conf file

```
# Example localscan file

subnet 192.168.0.0

mailto admin@mainmachine.org

allclr yes
```

Example LocalScan.conf file

```
# Ignore ssh servers on all machines

ignore 192.168.0.1-254 22

# ignore webserver, ftp on the following
   machine

ignore 192.168.0.240 80 21
```

LocalScan Caveats

- In the config file a '#' anywhere in the line makes the WHOLE line a comment, not from that point forward
- The syntax checking of the localscan file is a bit rough
- Localscan only checks for an open port. E.g. an ftp server running on port 22 will make it through the localscan.conf file I listed before

Ndiff

- Written in Perl
- Quite simply provides an intelligent output of the differences between any 2 nmap scans
- Also has several supporting tools like ngen and nrun to support ndiff

Setting up Ndiff

- Example Use

create baseline scan

nmap -m baseline.nm 192.168.0.0/24

create a second scan

nmap -m scan.nm 192.168.0.0/24

Running Ndiff

- Now compare the results using ndiff

```
# ndiff -baseline baseline.nm -observed
  scan.nm
```

Ndiff Output

... ndiff outputs: ...

missing hosts:

new hosts:

changed hosts:

Ngen – Generate a baseline

- Ngen – can be used to artificially create a baseline for ndiff
- generate a baseline of two machines both with ssh and one with a webserver on it

```
# ngen -o baseline.nm -h
  192.168.0.20/32:80,22 -h
  192.168.0.32/32:22
```

- Is a pretty powerful tool, very simple example

Nrun – automate nmap and ndiff

- Runs nmap, save results optionally run ndiff and can generate a report
- Can be used to easily save nmap result files over time and easily create reports

Ndiff Caveats

- Is currently an orphaned project
  - I'm starting a sourceforge project to make it available/supported again
  - If anybody is interested in helping out please let me know
- Parts like nrun don't currently work on many recent distros at the moment – E.g. ubuntu without hacking up the source code

PBNJ

- Ports Banners N' Junk
- Combines nmap with amap to determine what software and what version of the software is running on a port
- Also written in Perl

Using PBNJ

- Creating a baseline

pbnj -s 192.168.0.0/24 -r 1-9000 -o ofile

- Do a comparison scan and e-mail results

pbnj -i tmp –email-to admin –email-from admin –email-type both

What it does

- Provides version O/S version type information
- Is very configurable
- Harder to configure as a result of version tracking (still not that hard)
- Not always the most stable tool

PBNJ Caveats

- Also has a menu interface

pbnj -interactive

- Can be a useful tool
- Crackers can also compile up a program to send out a different header
  - Can compile up openssh to say it is Apache 2.0.36

Summary

- All of these are written in Perl cross-platform
- LocalScan is the easiest of the tools to configure and use
- Ndiff is my personal favorite
- PBNJ's banner analysis is interesting

- Experiment with PBNJ and deploy LocalScan and consider ndiff when it is up at sourceforge

Getting Started with Tools

- Several Linux LiveCDs offer these tools
  - Backtrack
  - nUbuntu

Resources

- NMAP - http://www.insecure.org
- LocalScan - http://staff.washington.edu/dgreene/localscan
- Ndiff - http://packages.debian.org/unstable/source/n
  http://www.ndiff.org (soon)
- PBNJ - http://pbnj.sourceforge.net

Contact Info

- Contact info
  - aaron@itinomaha.org
- Slides can be found on the NebraskaCERT website http://www.NEbraskaCERT.org/CSF