

# SOME NMAP SCAN DETECTION & A TCP/IP PRIMER IN C

By David Weinman  
Thanks to Richard Weiss & Jon Erickson

4th February, 2015

# OUTLINE

Outcomes

Some Networking

Client / Server model

3-way Handshake

NMAP

Sniffers

Detection Method

My Scan Detector

# Outcomes

Just enough Networking

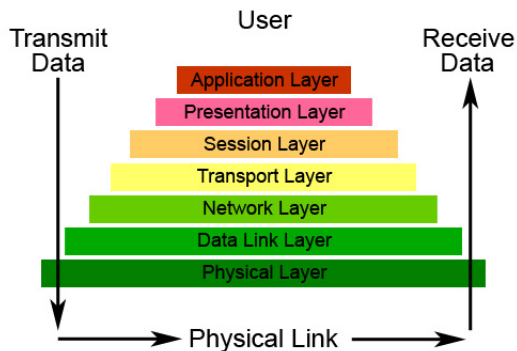
How do some common NMAP scans work?

How do sniffers work? (In C)

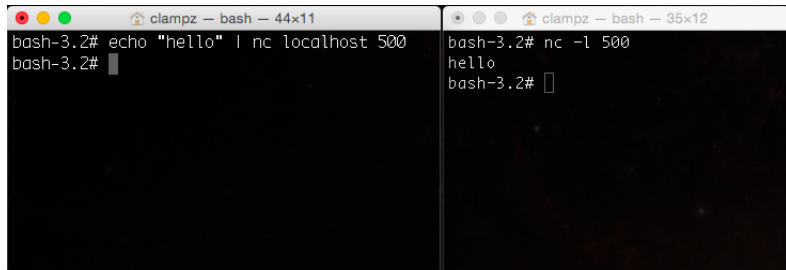
How does one detect NMAP Scans? (Or how one might choose a method)

How my NMAP scan detector works.

## The Seven Layers of OSI



# Client / Server model



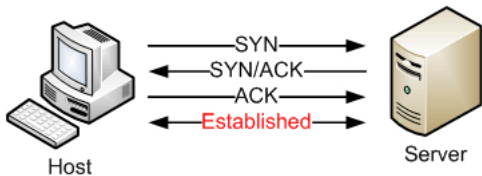
The image shows two terminal windows side-by-side. The left window, titled 'clampz - bash - 44x11', shows the command 'echo "hello" | nc localhost 500' being executed, followed by a prompt 'bash-3.2#'. The right window, titled 'clampz - bash - 35x12', shows the command 'nc -l 500' being executed, followed by the output 'hello' and a prompt 'bash-3.2#'. This demonstrates a simple client-server communication using netcat.

```
clampz - bash - 44x11
bash-3.2# echo "hello" | nc localhost 500
bash-3.2#

clampz - bash - 35x12
bash-3.2# nc -l 500
hello
bash-3.2#
```

# 3-way Handshake

## TCP Three-Step Handshake



# How does NMAP's SYN scan work?

- (AKA Half-Open Scan because the connection is closed after becoming half-open)
- NMAP sends a SYN packet
- target returns SYN-ACK if port is open
- NMAP responds to SYN-ACK with RST to avoid DoS'ing target.

## How does NMAP's FIN, X-MAS & NULL scans work?

- NMAP sends a FIN, X-MAS or NULL packet
- if the port is open, the packets are ignored.
- if port is closed then target returns RST (according to RFC-793)
- (note: these scans can be unreliable because some OS's have TCP implementations which do not send RSTs)



## Lets look at a real SYN scan :)

- download syn\_scan.pcap from <http://ada.evergreen.edu/~weidav02>
- look at this pcap in Wireshark

# Let's use some sniffers

- Common CLI sniffers are tcpdump & dsniff.
- dsniff supports attacks and can be used to filter for interesting data.
- Lets examine some network traffic for comparison?

# Let's build some sniffers

- download and compile `raw_tcpsniff.c` and then `pcap_sniff.c`, `hacking.h` can also be found here.
- I will upload these examples to <http://ada.evergreen.edu/~weidav02>
- These examples are from 'Hacking: The Art of Exploitation' by Jon Erickson in Ch 0x400.

# Deciding a Method

- What kinds of scans do you hope to detect?
- There are many variables to consider (pros/cons lists can help)
- For example you may want to limit the size of your logs or print them from your system for protection.
- How reliable do you want your detections to be? (remember type 1 and type 2 errors)

# Known Methods

- Two research papers on this topic that I have found are
- "Designing and Attacking Port Scan Detection Tools" by solar designer (1998)
  - outlines the design of their tool 'scanlogd'
  - 'scanlogd' uses a count-based attack signature to keep low risk of false-negatives occurring.
- "Detection and Characterization of Port Scan Attacks" by Cynthia Bailey Lee et al. (2003)

# My Method & Reasoning

- I want to detect scans even if the scan was spread out over time.
- I can worry about avoiding false negatives after.
- How can I find key differences between NMAP scans and the rest of the normal network traffic?

# References

- [1] 'Hacking: The Art of Exploitation' by Jon Erickson
- [2] <http://www.windowsnetworking.com/img/upl/image0011210155736818.jpg> OSI Model Image
- [3] <http://www.georgecoding.com/wp-content/uploads/2013/04/handshake.gif> 3-way Handshake Image

THANK YOU ...