Offense-Based Cybersecurity: Reconnaissance

CS 459/559: Science of Cyber Security
3rd Lecture

Instructor:

Guanhua Yan

Project 1

- Demonstrate a cyber attack to the TA
- In your first project report, please:
  - Explain what are the vulnerability, exploit, attack surface, and attack vector in your project
  - Explain how the attack (exploitation) occurs in your project
  - Explain why the attack (exploitation) works
- Each of your project reports should be **at least five pages, excluding bibliography**
- **Due time: TBD**
  - **Four days of grace period, with 2.5% penalty each half day late**
- **Grading criteria:**
  - **Results, novelty, difficulty, presentation**

How to choose your project?

- You should be familiar with the concepts behind the attack (software, network, system, or human)
- You should be comfortable with developing defensive mechanisms on the target being attacked
- It's OK to work on known vulnerabilities using known exploit code (plenty of online resources)
  - Known vulnerabilities: National Vulnerability Database (https://nvd.nist.gov)
  - Known exploits: Exploit database (https://exploit-db.com)
  - The metasploit framework (https://www.metasploit.com) makes exploitation easy!
- Start early and work on your project hard!
  - **You will continue working on the same project with defensive techniques introduced in this course**

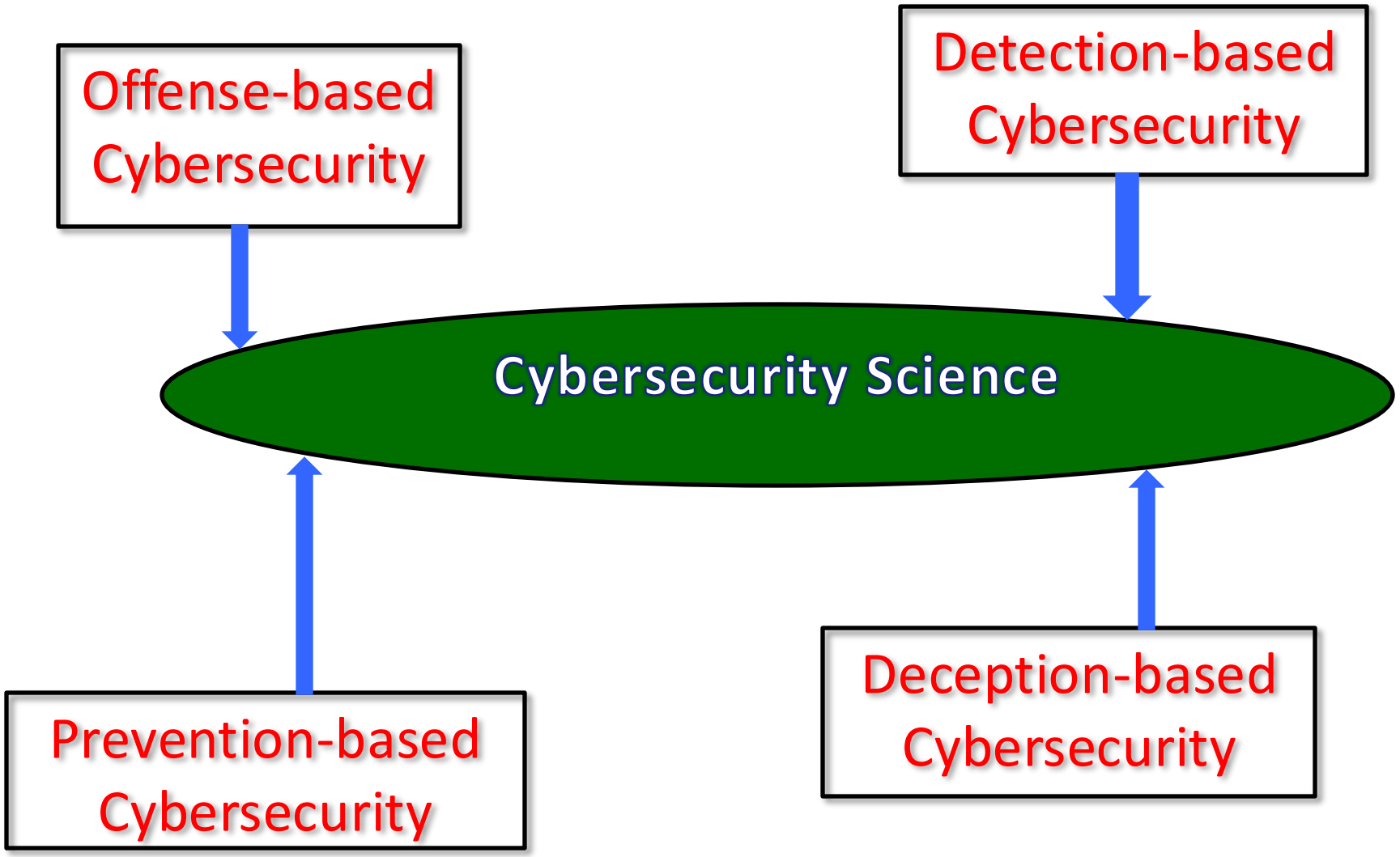
What we have learned from last lecture?

- Attack surface and vector
- Security risk analysis
- Security models

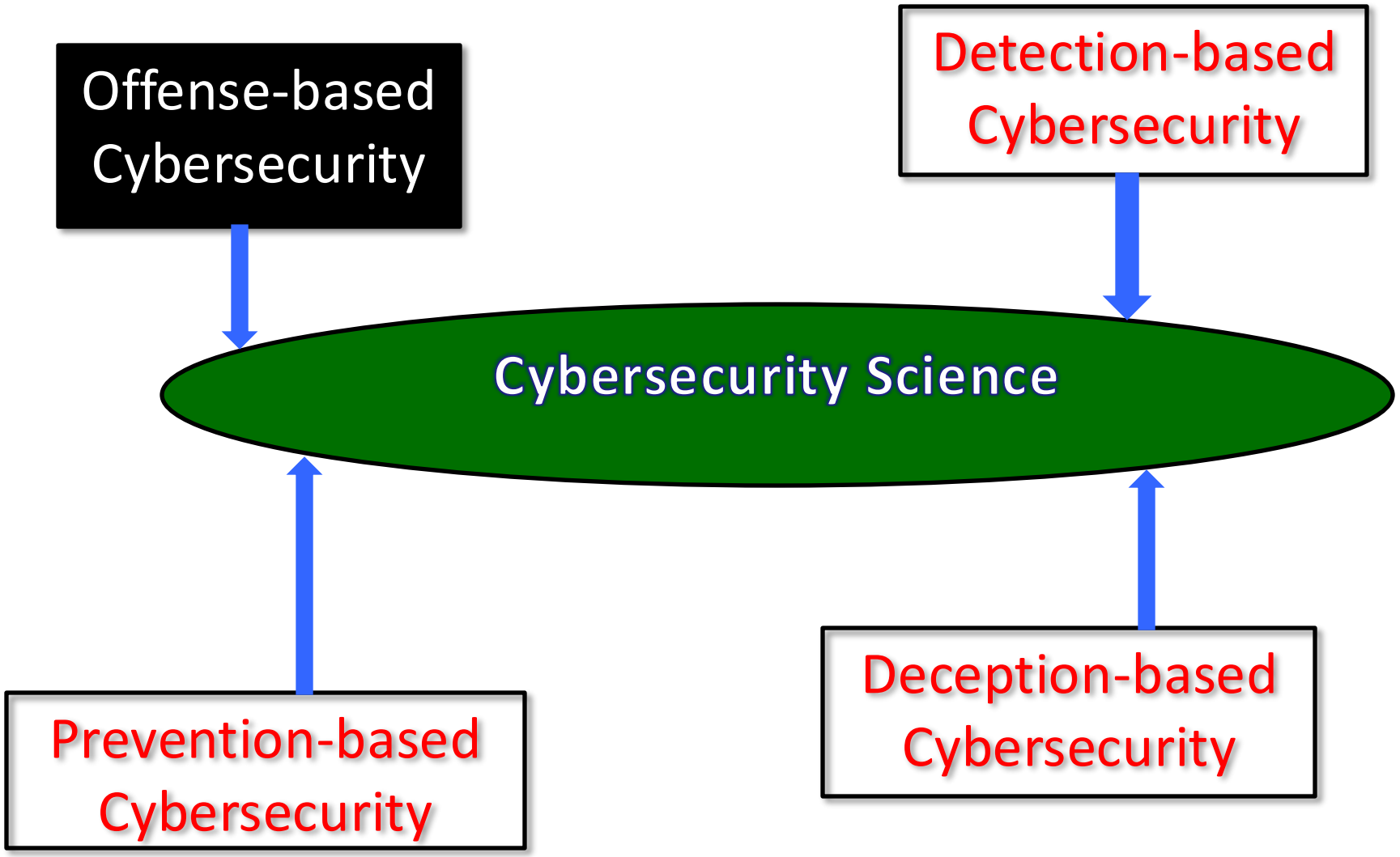
Reconnaissance Tools

0) Public/Online Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

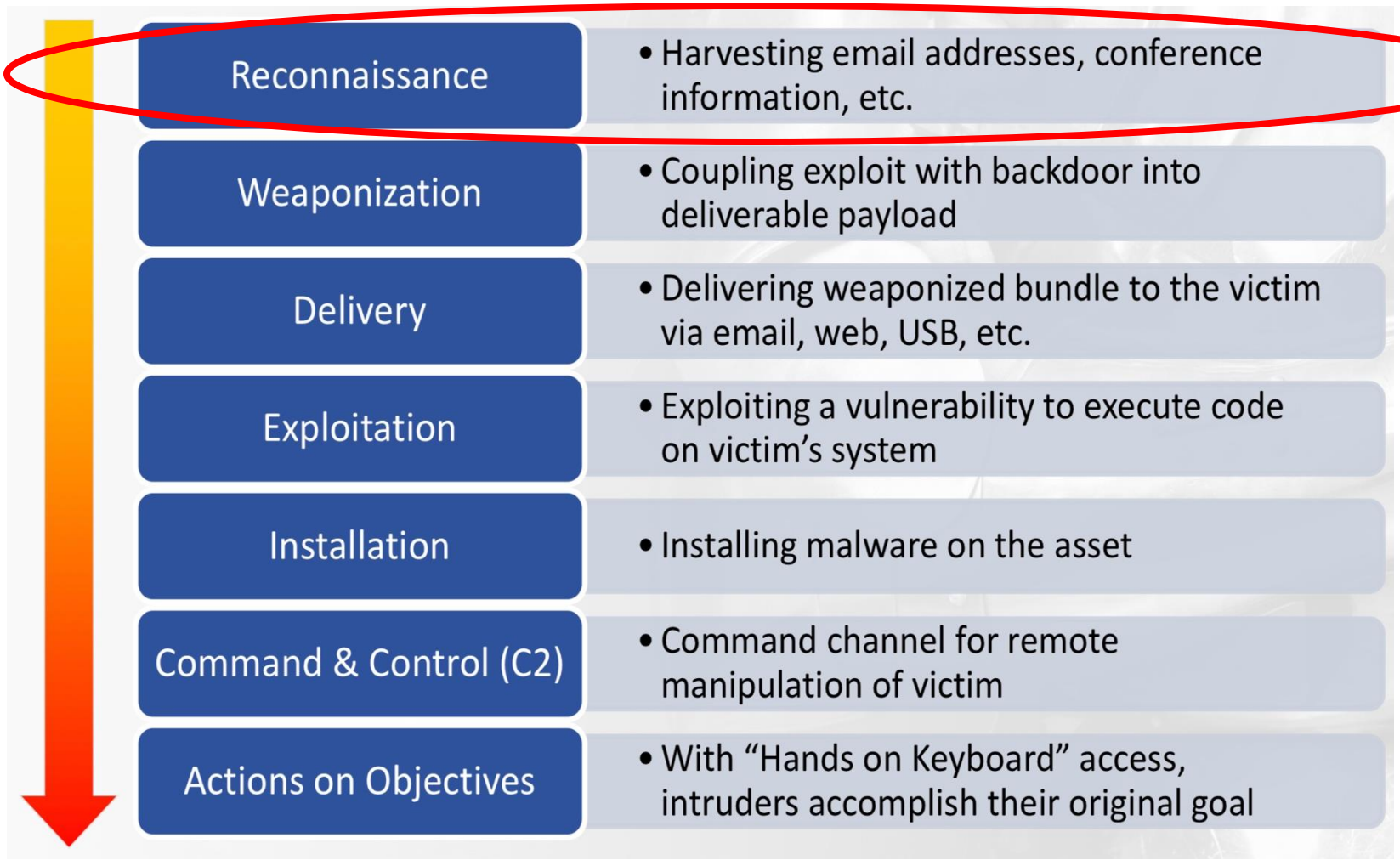
Overview of cybersecurity science



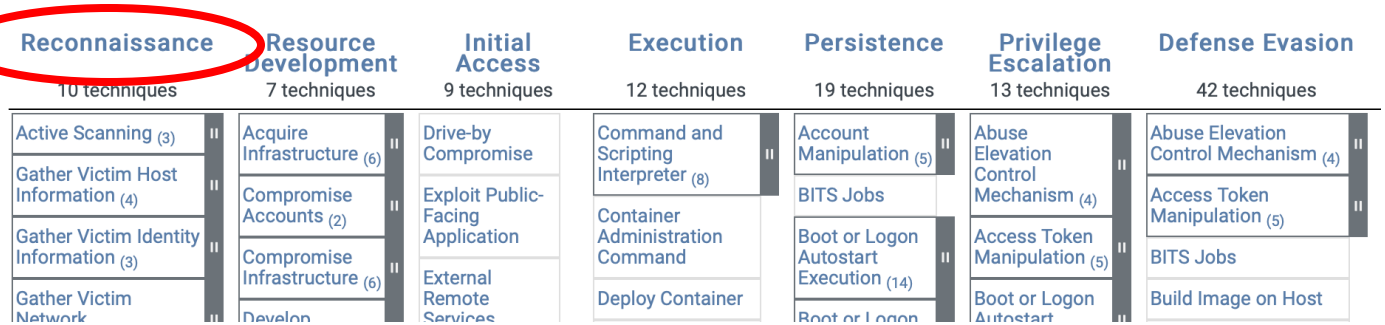
Overview of cybersecurity science



Cyber Kill Chain



MITRE ATT&CK



Reconnaissance

The adversary is trying to gather information they can use to plan future operations.

Reconnaissance consists of techniques that involve adversaries actively or passively gathering information that can be used to support targeting. Such information may include details of the victim organization, infrastructure, or staff/personnel. This information can be leveraged by the adversary to aid in other phases of the adversary lifecycle, such as using gathered information to plan and execute Initial Access, to scope and prioritize post-compromise objectives, or to drive and lead further Reconnaissance efforts.

Reconnaissance Tools

0) Public/Online Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Public Reconnaissance Tools

- **Quiet Recon** – **hackers often don't want to start off by generating lots of noise that can be picked up by IDS devices**
  - search engines can be a great recon tool - they allow stealthy probing that likely won't be detected or traced back to the hacker
  - **Google-Fu**
    - **ability to use search directives (advanced operators) effectively in order to filter out Google search results**

http://playfulcleverness.blogspot.ca/2013/04/google-fu.html

Public Reconnaissance Tools (cont.)

- **Google Directives** – **keywords that enable more accurate extraction of info from Google Index**
  - formatting of Google directives:

```
<directive>:<value> <term(s) to search>
```

  - **examples of directives:**
    - site:<domain-name> <term(s) to search>
    - intitle:<term(s) to search>
    - inurl:<term(s) to search>
    - filetype:<extension of the file type>
    - link:<URL that searched pages point to>

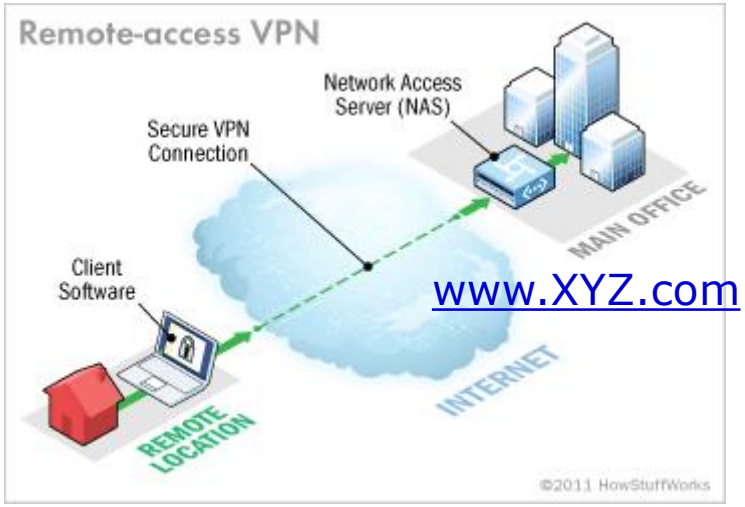
http://www.googleguide.com/advanced_operators_reference.html

Public Reconnaissance Tools (cont.)

Example

Assume Company XYZ's main web page is: www.XYZ.com.

Which Google directive should have the attacker used to narrow down the search for remote web server 'pointing' to the main enterprise web server.



Google Canada

link:www.XYZ.com -site:www.XYZ.com

Google Search I'm Feeling Lucky

Public Reconnaissance (cont.)

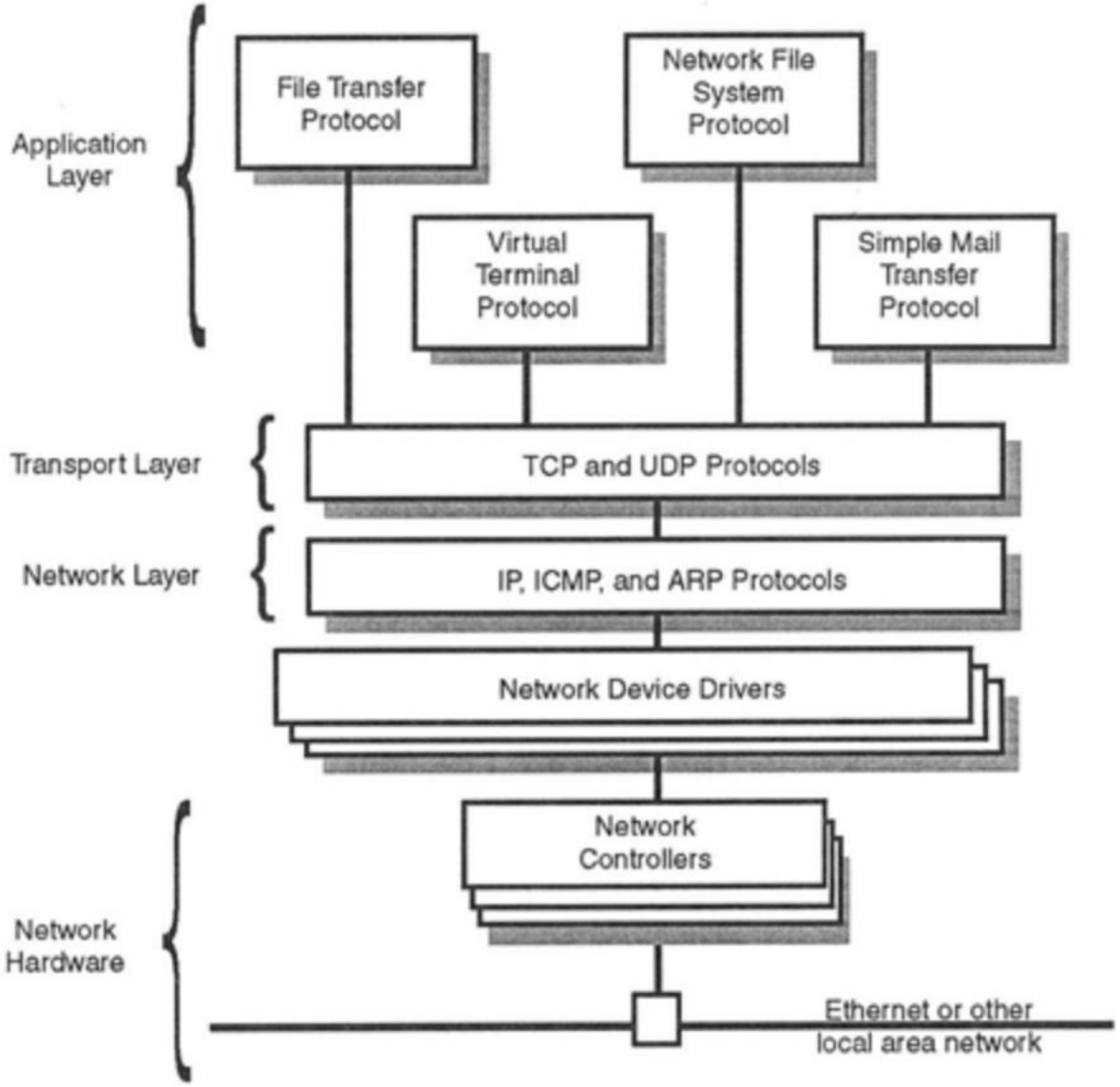
Example

What is the intention behind the following commands:

```
site:www.yorku.ca   inurl:login.php

site:www.yorku.ca   filetype:xls   inurl:"email.xls"

filetype:dat   site:www.yorku.ca   "password.dat"

filetype:sql   intext:password

ext:(doc | pdf | xls | txt | ps | rtf | odt | sxw |
psw | ppt | pps | xml) (intext:"confidential salary"
| intext:"budget approved")
```

Reconnaissance Tools

0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers



Internet protocol stack

Port Scanners

- **Footprinting** – **systematic research of IP addresses owned by a target organization**
  - passive and non-intrusive process of discovering which machines are out there (part of initial attack reconnaissance)
  - Web is a good starting point

examples of **'doorknob rattling'**

- **Scanning (Fingerprinting)** – **systematic scanning of a target organization's IP addresses/hosts**
  - discovering of individual hosts as well as services/ports running on them
  - **may include OS Fingerprinting** – determining which OS runs on a host
    - **typically done through detailed inspection of packets crafted by the host**

Port Scanners (cont.)

- **Port Scanner** – **a tool used by both attackers and defenders to identify active hosts and services on a network**
  - results of a scan on a port can be:
    - **open or accepted:** host sends reply indicating that service is available on a port
    - **closed / denied / not listening:** host sends reply indicating unavailable service on a port
    - **filtered / dropped / blocked:** there is no reply

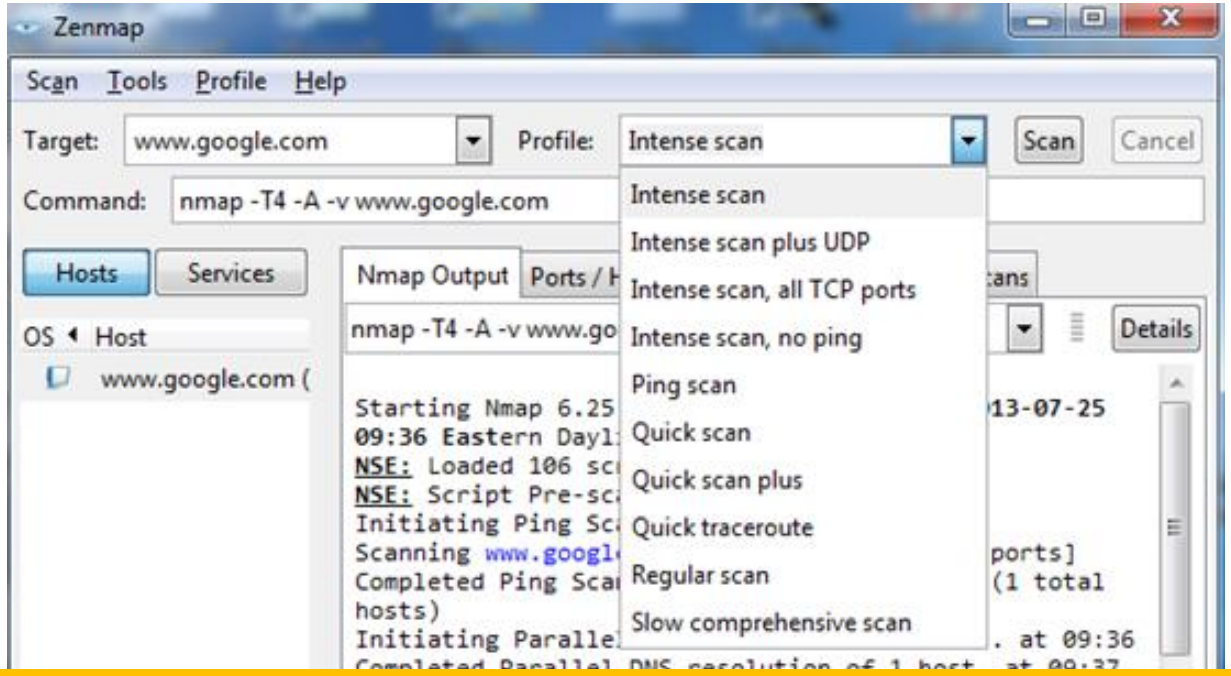
Port Scanners (cont.)

- **Nmap** – **Network Mapper - a security scanner written by Gordon Lyon** (aka Fyodor) **in 1997**
  - the most respected and well-known port scanner in both **black- and white- hat** community for its efficiency, flexibility and scanning speed
  - in addition to doing host and port scanning, also capable of determining:
    - OS of the target
    - name and versions of the listening services
    - presence of firewall, etc.
  - runs on Windows, Linux, Solaris, Mac, etc.

http://nmap.org/download.html

Port Scanners (cont.)

Example: Nmap/Zenmap Port Scanner



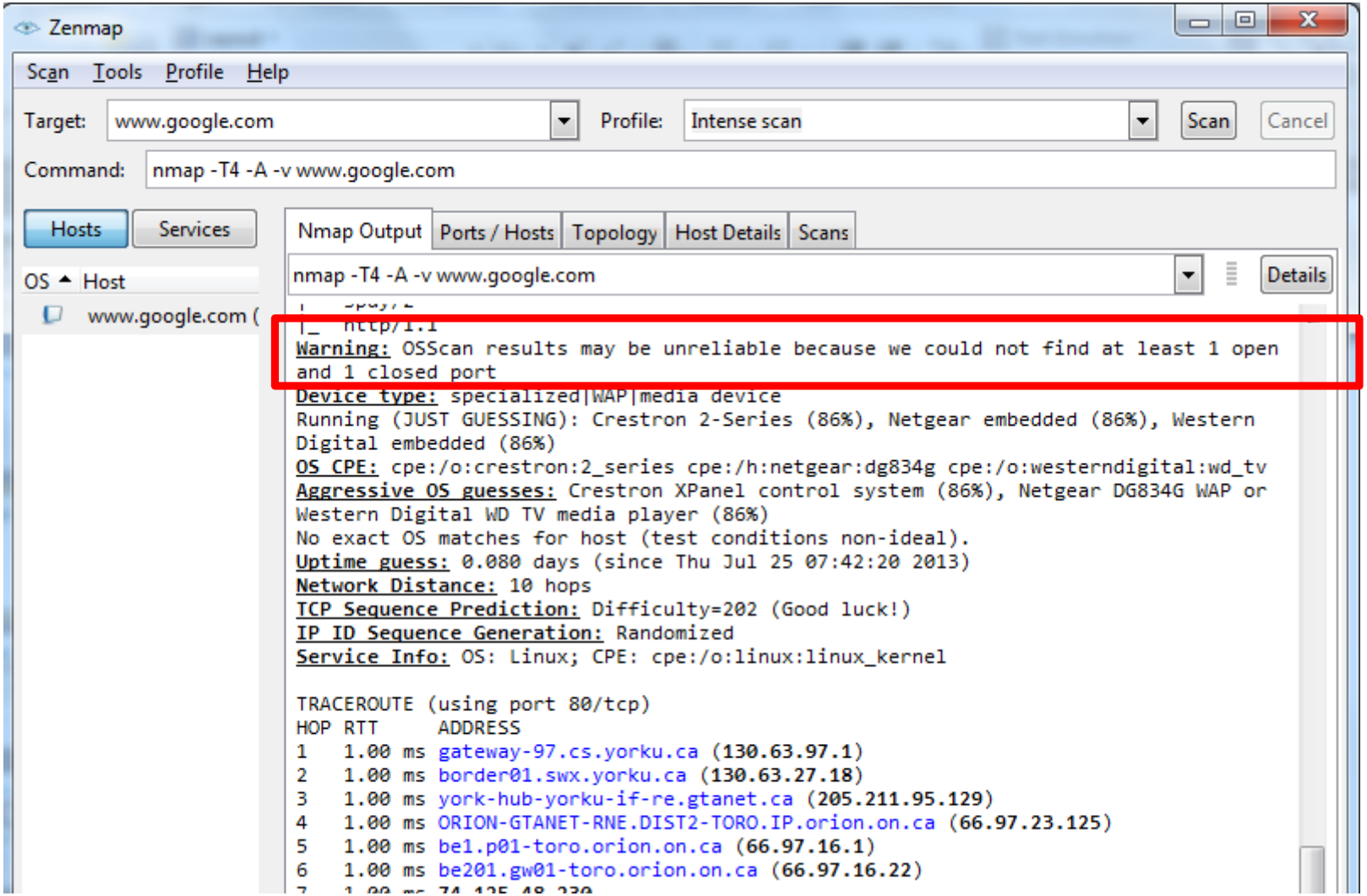
-T4: Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network

-A: Enables OS detection, version detection, script scanning, and traceroute

-v: Increase the verbosity level (use -vv or more for greater effect)

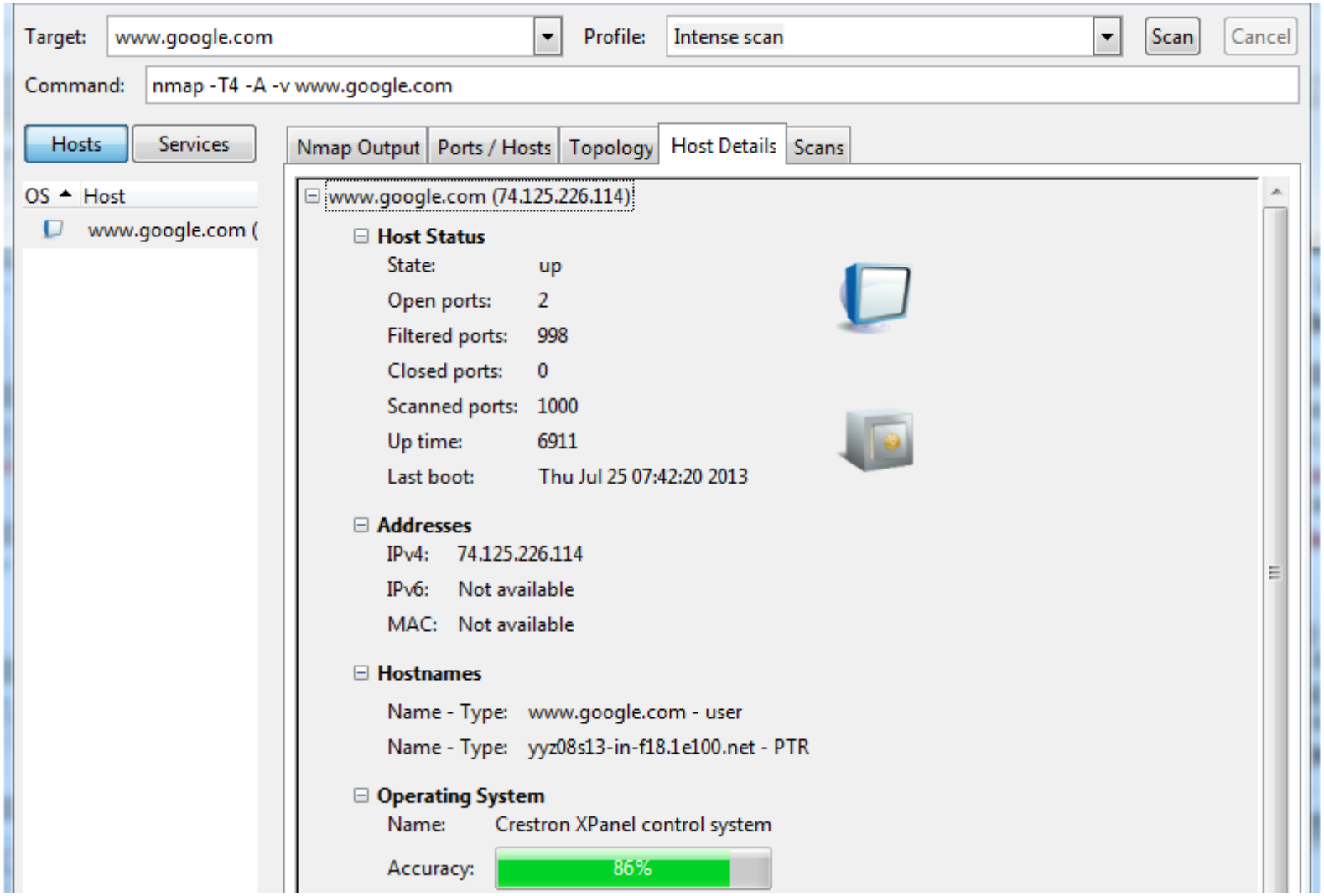
Port Scanners (cont.)

Example: Nmap/Zenmap Port Scanner (cont.)



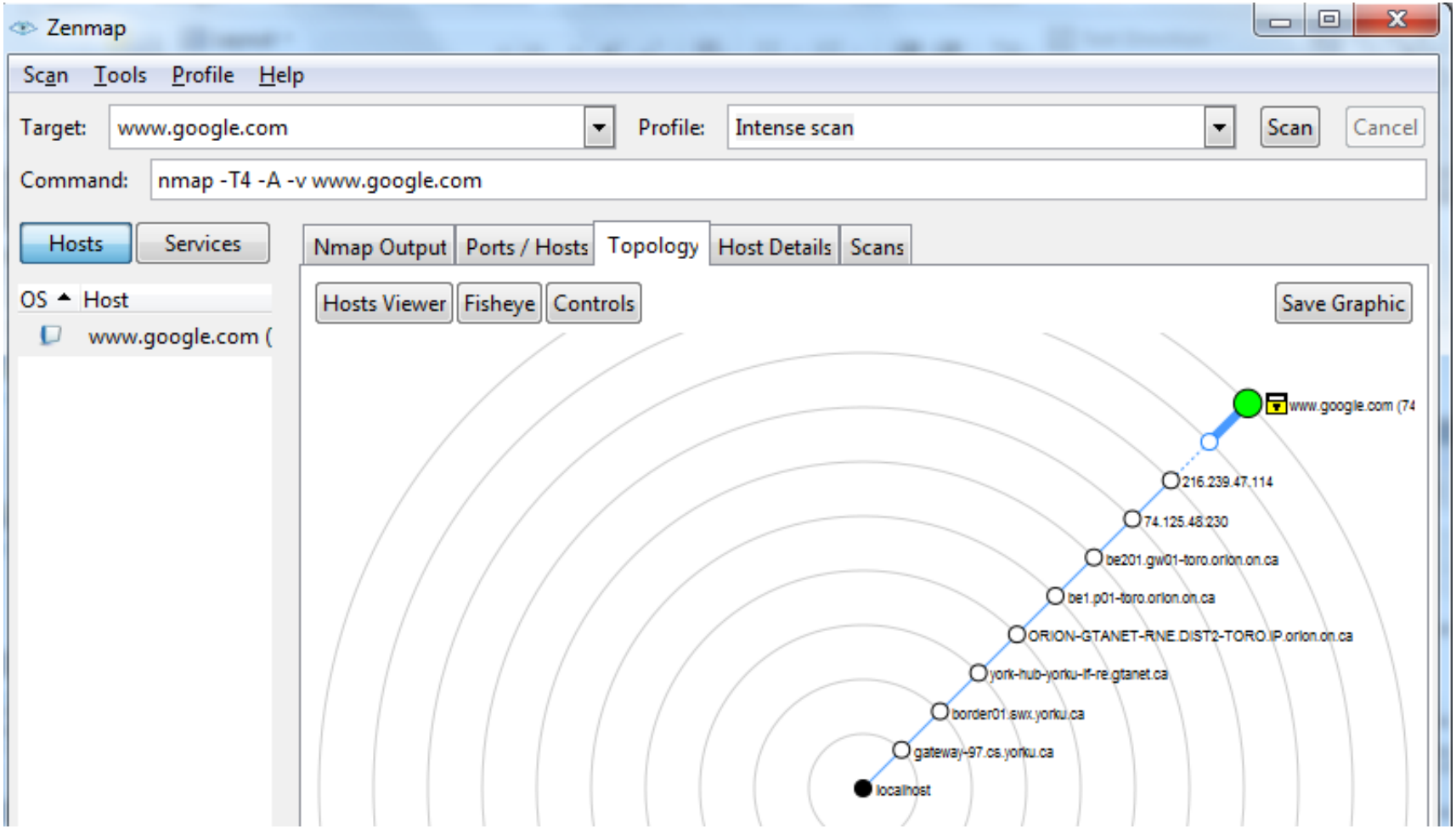
Port Scanners (cont.)

Example: Nmap/Zenmap Port Scanner (cont.)



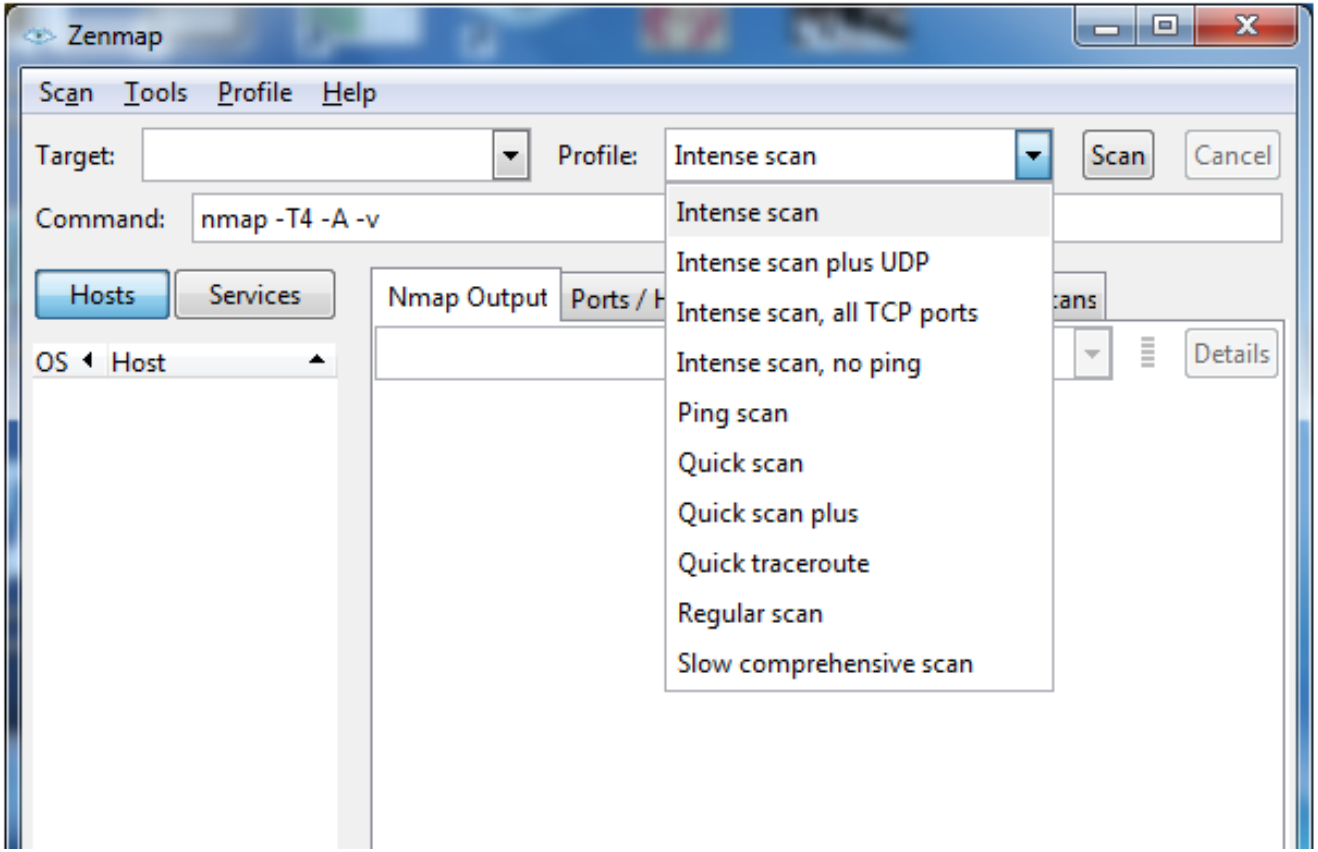
Port Scanners (cont.)

Example: Nmap/Zenmap Port Scanner (cont.)



Port Scanners (cont.)

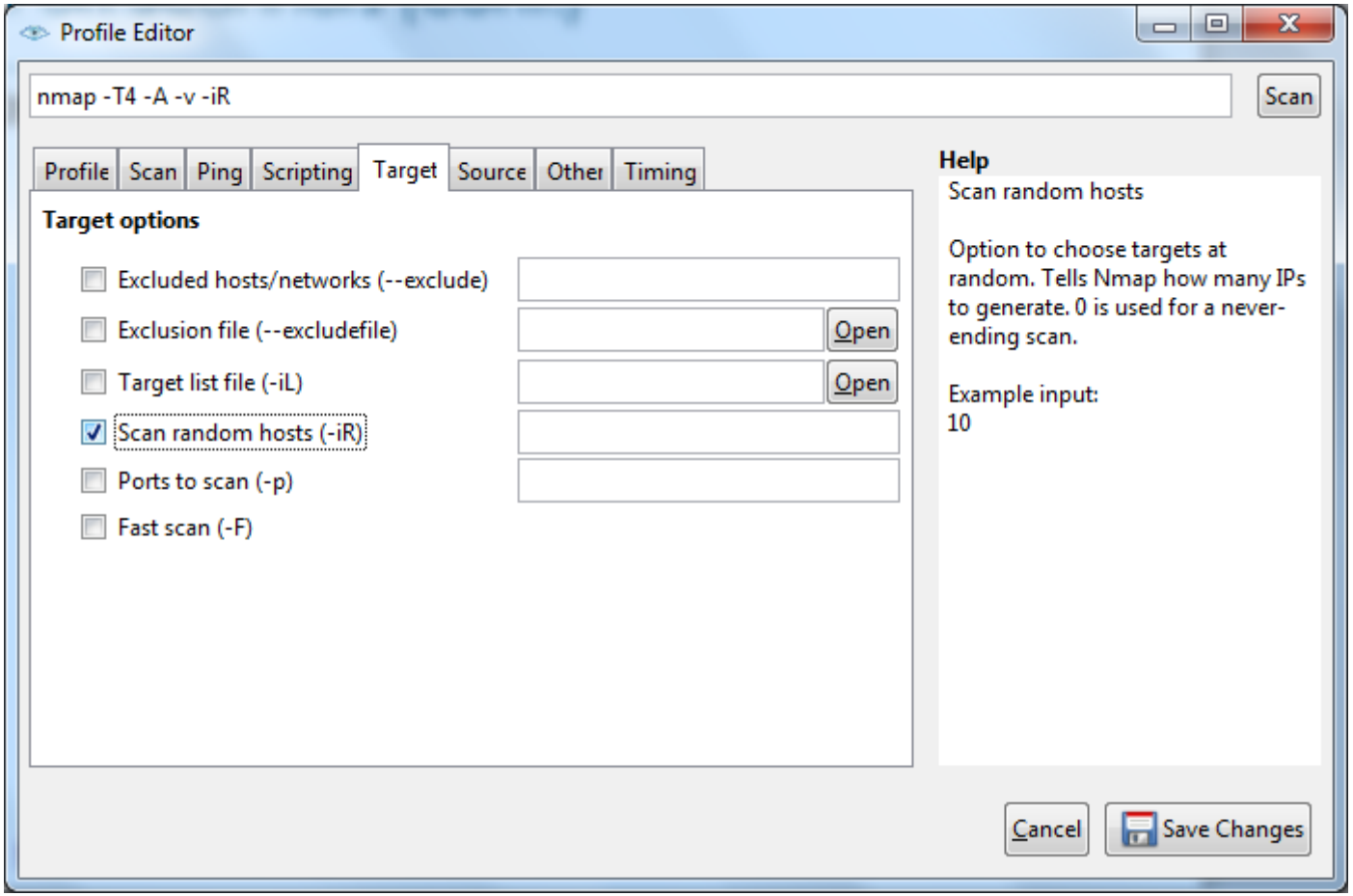
Example: Nmap/Zenmap Port Scanner (cont.)



https://svn.nmap.org/nmap/zenmap/share/zenmap/config/scan_profile.usp

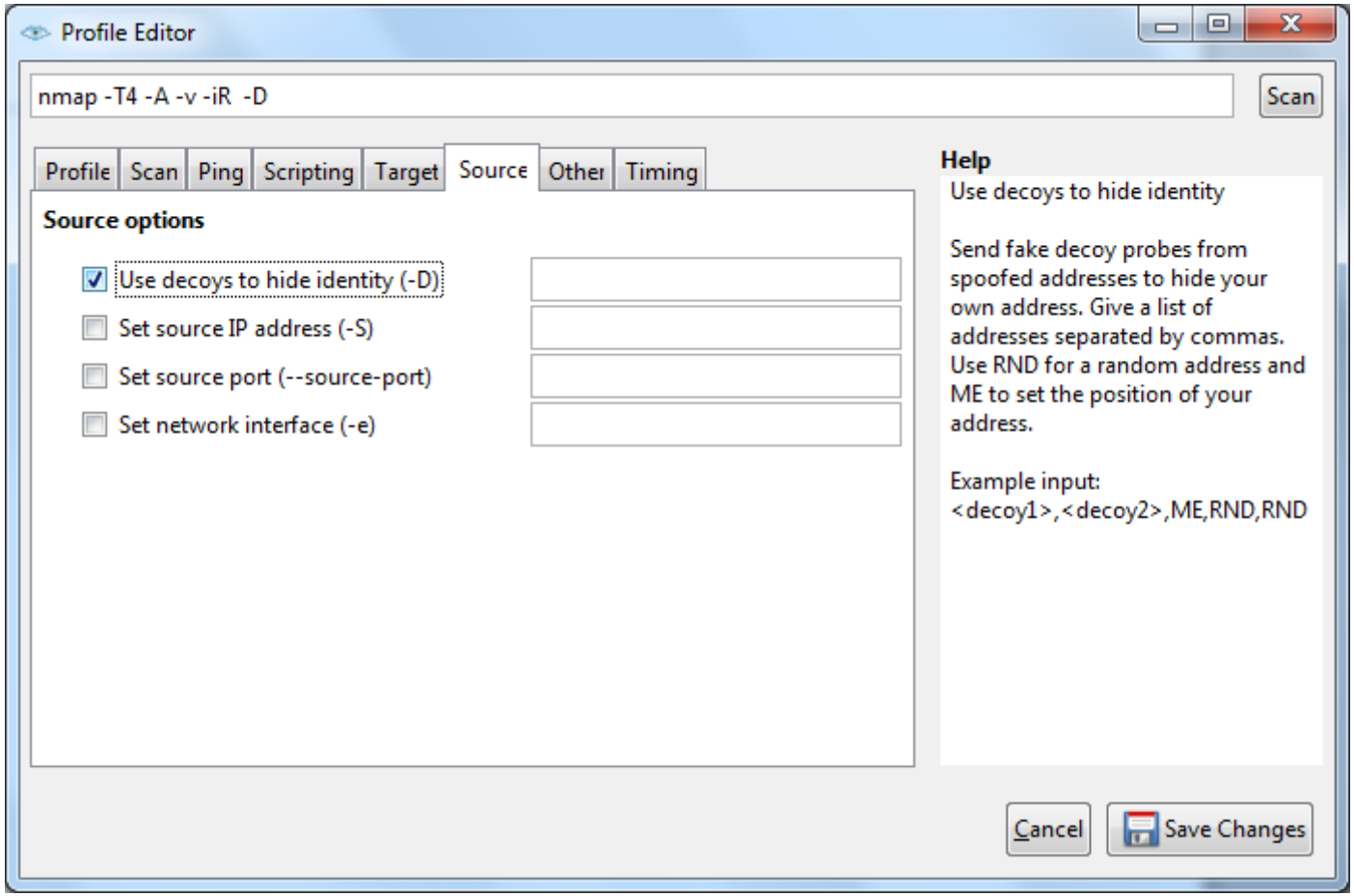
Port Scanners (cont.)

Example: Nmap/Zenmap Port Scanner (cont.)



Port Scanners (cont.)

Example: Nmap/Zenmap Port Scanner (cont.)



http://nmap.org/book/man-bypass-firewalls-ids.html

Port Scanners (cont.)

Port Scanner Techniques

1) ICMP Ping Scan

- **not really port scanning, as ICMP is Layer 3 protocol, but useful for probing of all active hosts in a network – host scanning**
- **scanner sends a single ICMP request to a destination; an ICMP response will arrive back unless the destination is not available or ICMP protocol is filtered**
- **potentially faster than other footprinting technique – only one sent packet per machine**
- **does not provide lots of information …**

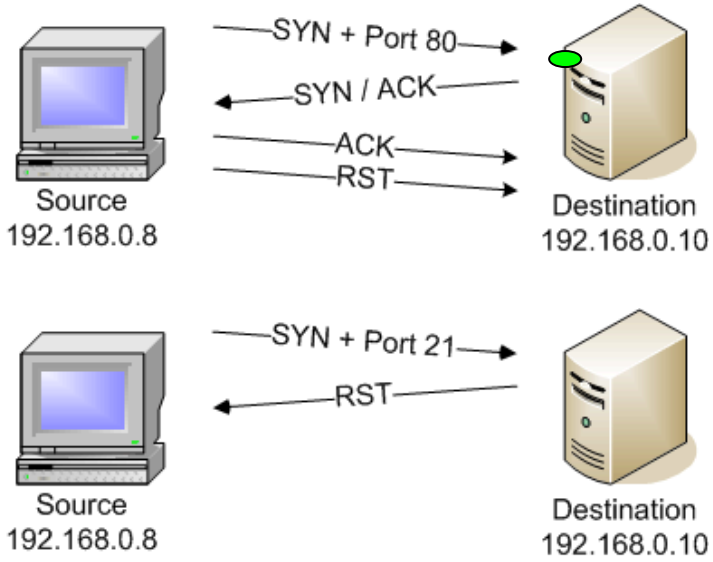
ICMP Ping scan to an active/on and inactive/off machine
http://www.networkuptime.com/nmap/page3-8.shtml

Port Scanners (cont.)

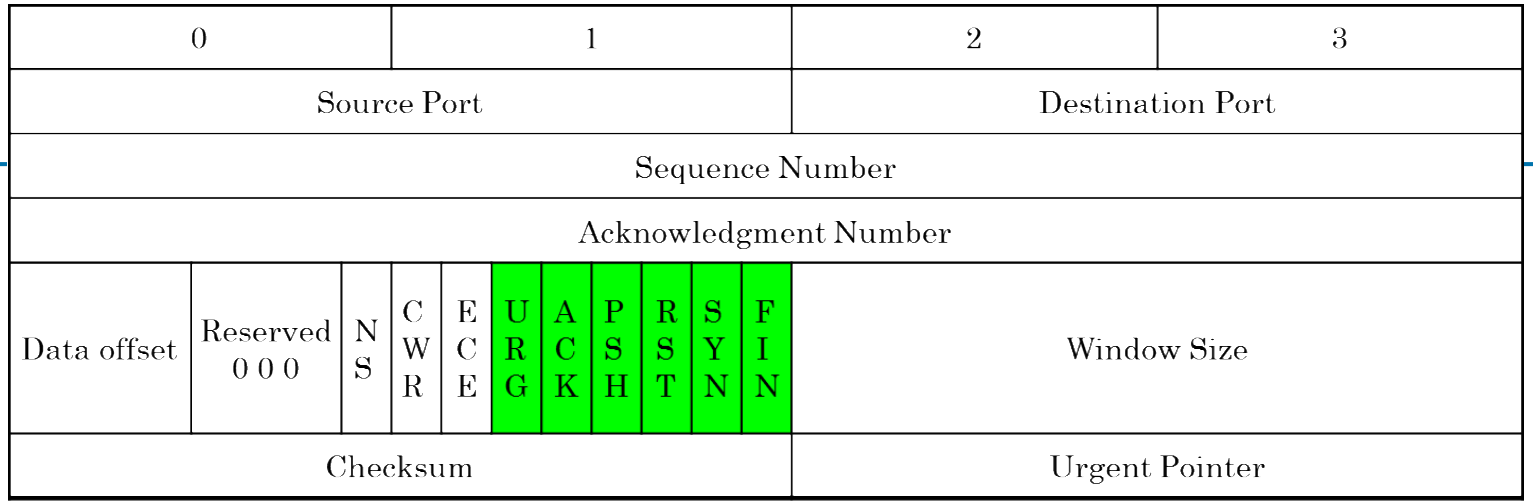
- Port Scanning Techniques

2) TCP connect() Scan

- most basic form of TCP scanning
- performed from 'application layer' - OS's connect() function is used to connect to a desired port
- easy to implement; however, very slow and detected (loged) by most sites/firewalls



TCP connect() scan to an open and to a closed port
http://www.networkuptime.com/nmap/page3-3.shtml



0	1	2	3
Source Port		Destination Port	
Sequence Number			
Acknowledgment Number			
Data offset	Reserved 0 0 0	NS CWR ECE URG ACK PSH RST SYN FIN	Window Size
Checksum		Urgent Pointer	

According to TCP RFC (specification):

TCP to a closed port => TCP RST arrives back
(SYN=0, ACK=0 or RST=0)

TCP to an open port => nothing back
(SYN=0, ACK=0 and RST=0)

UDP to a closed port => ICMP unreachable arrives back

Port Scanners (cont.)

- Port Scanning Techniques

3) TCP SYN Scan

- 'half-open' scanning
- instead of using OS' network function, scanner itself generates TCP-SYN packets; upon receiving a TCP-ACK, scanner immediately sends a RST to close the connection
  - handshake is never completed!
- most popular form of TCP scan since most sites do not log half-open connections - much 'quieter' than connect() scan ☺
- requires programming at OS level ☹

TCP SYN scan to an open and to a closed port
http://www.networkuptime.com/nmap/page3-2.shtml

Port Scanners (cont.)

- Port Scanner Techniques

4) TCP FIN Scan (Stealth Scan)

- **stealth scan = scan that sends a single frame to a TCP port without any TCP handshaking / additional packets**
- **TCP FIN scan sends a FIN packet; a closed port will reply with a proper RST, an open port will ignore the packet**
- **silence indicates an open port!!!**
- **UNIX vulnerable, but Microsoft is immune to this type of attack – RST sent regardless of the port state**

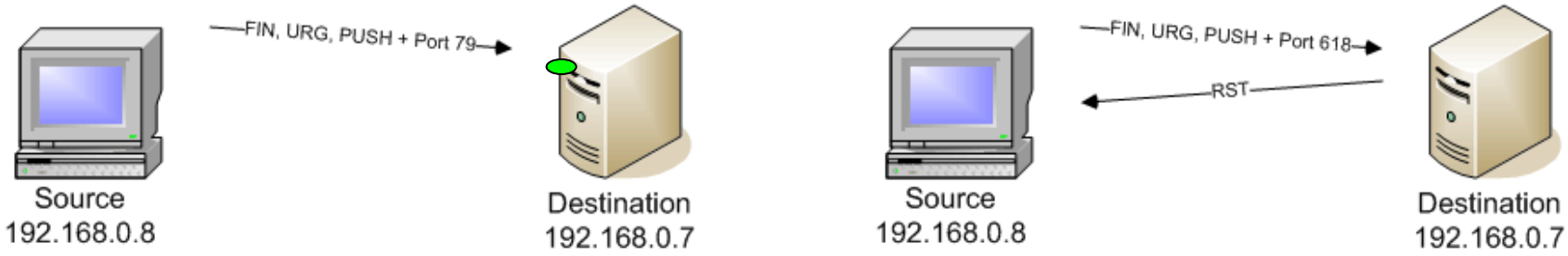
TCP FIN scan to an open and to a closed port
http://www.networkuptime.com/nmap/page3-4.shtml#3.3.1

Port Scanners (cont.)

- Port Scanner Techniques

5) Xmas-Tree Scan (Stealth Scan)

- **scanner sends a TCP frame with URG, PUSH and FIN flags set – Xmas tree packet (flags: 00101001, 'supper case' of FIN TCP packet)**
- **as in TCP FIN scan, <u>silence indicates an open port</u>!!!**
- **as TCP FIN scan, Xmas-Tree scan can 'sneak through' some non-statefull firewalls; but should not be used on Microsoft hosts**



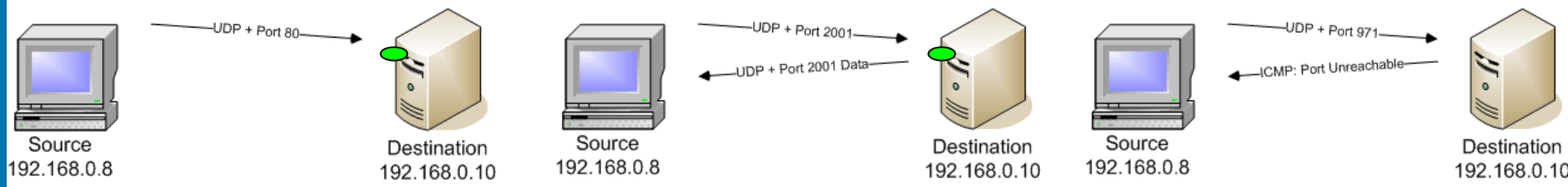
Xmas-Tree scan to an open and to a closed port
http://www.networkuptime.com/nmap/page3-5.shtml

Port Scanners (cont.)

- Port Scanner Techniques

6) UDP ICMP Scan

- previous scans find TCP ports/services; this scan looks for UDP ports/services
- scanner sends <u>empty</u> UDP datagrams – if port is listening, system sends back an error UDP message or nothing; if port is closed system sends an 'ICMP Port Unreachable'
- both UDP and ICMP are not guaranteed to arrive – lots of false positives possible
- also, a rather slow scan, as some systems limit the ICMP error message rate



UDP ICMP scan to an open and to a closed port
http://www.networkuptime.com/nmap/page3-10.shtml

Port Scanners (cont.)

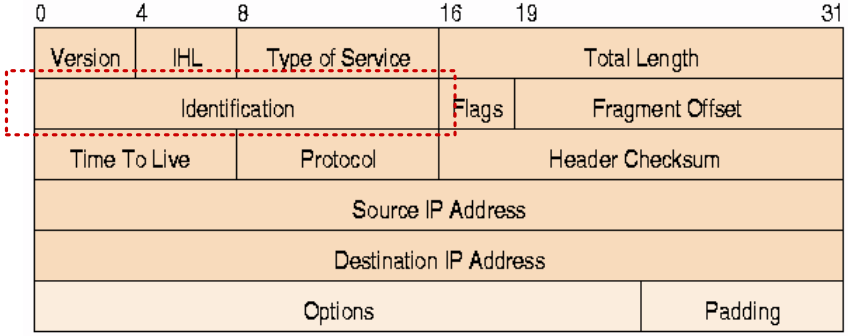
- Port Scanner Techniques

7) Idle Scan by Nmap

Used to:
1) hide the identity of the attacker (scanning machine); and/or
2) scan behind a firewall.

Requires the access to (communication with) at least one **zombie/dumb** host that can communicate directly with the target machine and sends/receives little traffic.

Exploits the fact that in many OSs, for every IP packet sent, **the value in packet's IP ID filed is incremented by one.**

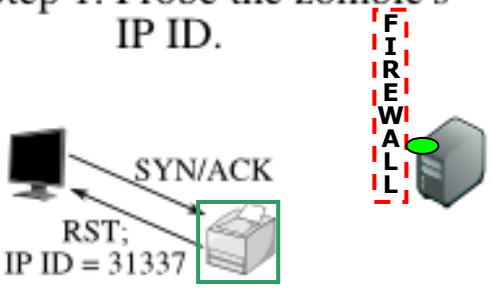


Port Scanners (cont.)

Example **Idle Scan** (cont.)

Idle scan of an open port:

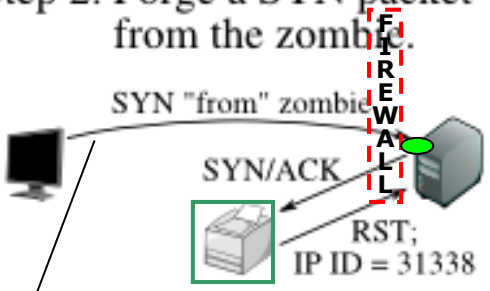
Step 1: Probe the zombie's IP ID.



trusted host/IP

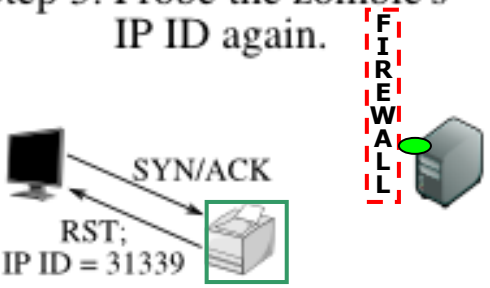
The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID.

Step 2: Forge a SYN packet from the zombie.



The target sends a SYN/ACK in response to the SYN that appears to come from the zombie. The zombie, not expecting it, sends back a RST, incrementing its IP ID in the process.

Step 3: Probe the zombie's IP ID again.



The zombie's IP ID has increased by 2 since step 1, so the port is open!

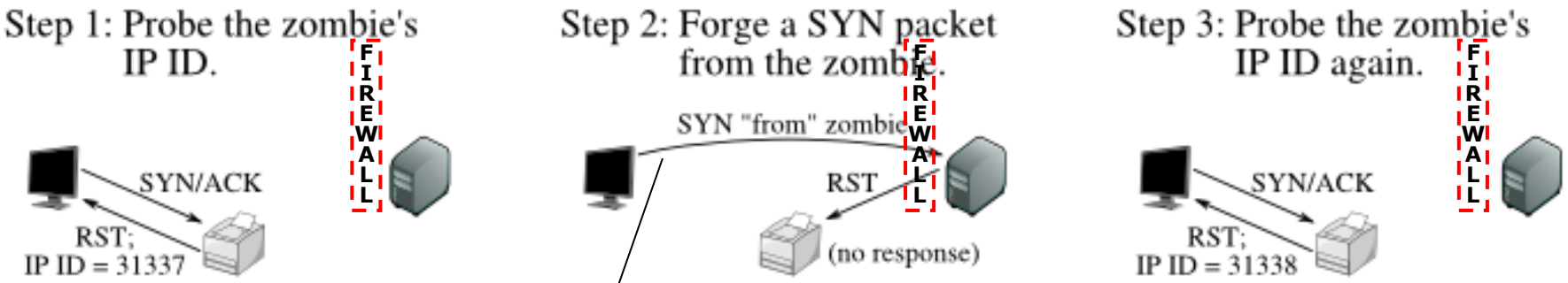
Source IP = Zombie IP
Destination Port = open

Zombie's IP ID increased by 2!

Port Scanners (cont.)

Example **Idle Scan** (cont.)

Idle scan of a closed port:



The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID. This step is always the same.

The target sends a RST (the port is closed) in response to the SYN that appears to come from the zombie. The zombie ignores the unsolicited RST, leaving its IP ID unchanged.

The zombie's IP ID has increased by only 1 since step 1, so the port is not open.

Source IP = Zombie IP
Destination Port = closed

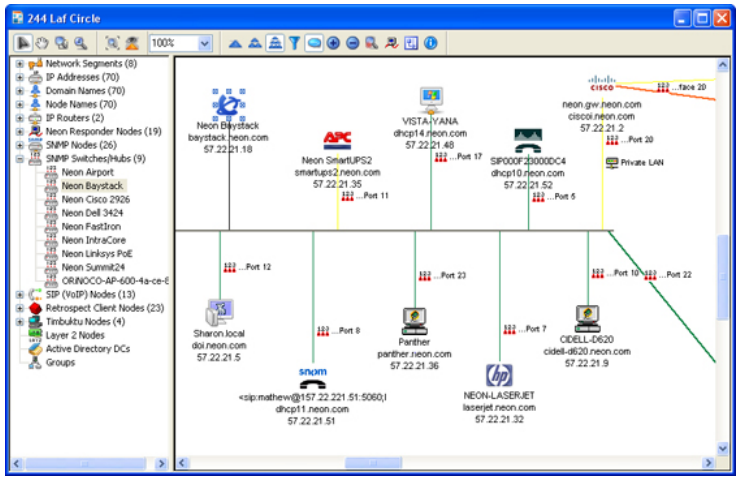
Zombie's IP ID increased by 1!

Reconnaissance Tools

0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Network Mappers

- **Network Mappers** – **software tools that identify all systems connected to a network**
  - most mappers use ICMP Ping …
  - most port scanners can be used as network mappers
  - examples of network mappers:
    - **Nmap**
    - **LanState**
    - **SolarWinds' LanSurveyor**



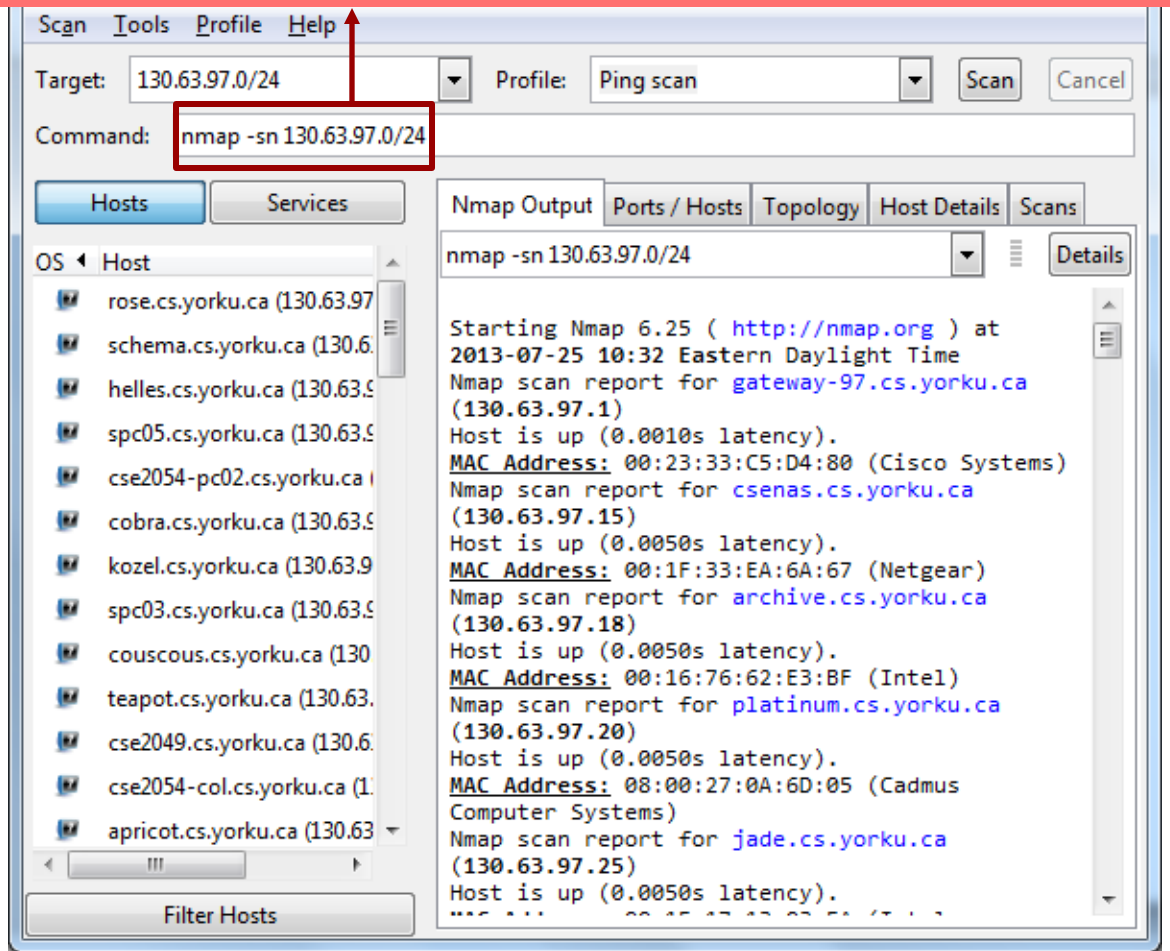
http://finaldownload.com/graphicsfile/screenshotimages/lansurveyor-101102.jpg

Network Mappers (cont.)

Example: Network mapping using Nmap

Target: IP/24, Profile: Ping scan

Host discovery: disable port scanning. Host discovery only.



Reconnaissance Tools

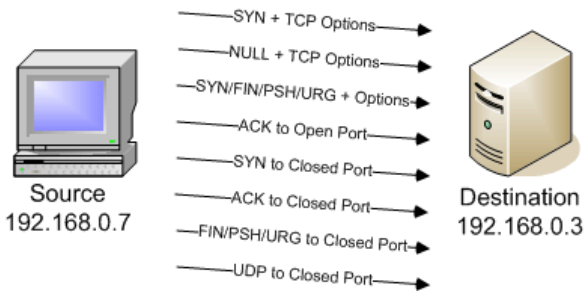
0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Operating System Detection Tools

- **OS Detection Tools** – **aka OS Fingerprinting Tools - aim to detect target host's OS**
  - knowing a host's OS is critical if one is to exploit the host's vulnerabilities (e.g. known bugs of that OS)
  - active fingerprinting – directly query host machine; replies are matched against database on known responses
  - passive fingerprinting – occurs without obvious querying of host machine (e.g. obtain information through sniffing)
  - examples of OS detection tools:
    - **Nmap**
    - **Xprobe**

Operating System Detection Tools (cont.)

- **OS Detection Techniques in Active Fingerprinting** – **TCP/IP stack is pretty much a fixed standard (given in RFC-s), but different OS vendors interpret the standard differently**
  - (active) fingerprinting takes advantage of differences in TCP/IP implementation
    - a crafted packet is sent to a remote system to elicit a unique response from the TCP/IP stack of the underlying OS
    - the unique response is referred to as an **OS fingerprint** or **signature**
    - the attacker than carefully analyzes and and compares the fingerprint to a data-base – comprising a wide range of known OS fingerprints …

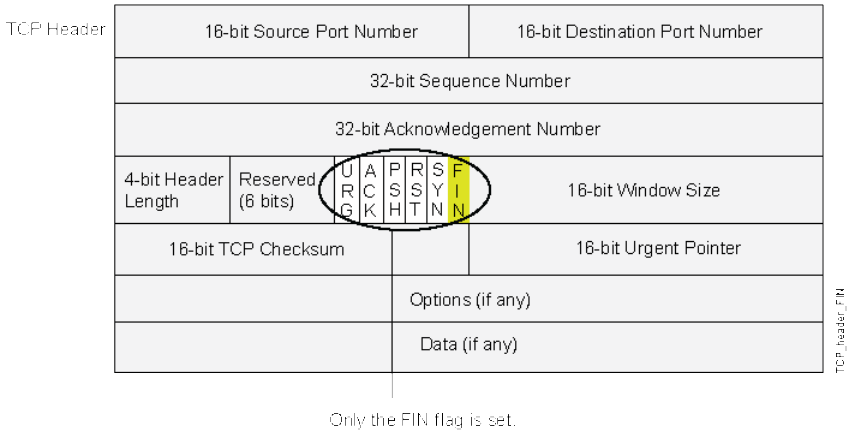


Operating System Detection Tools (cont.)

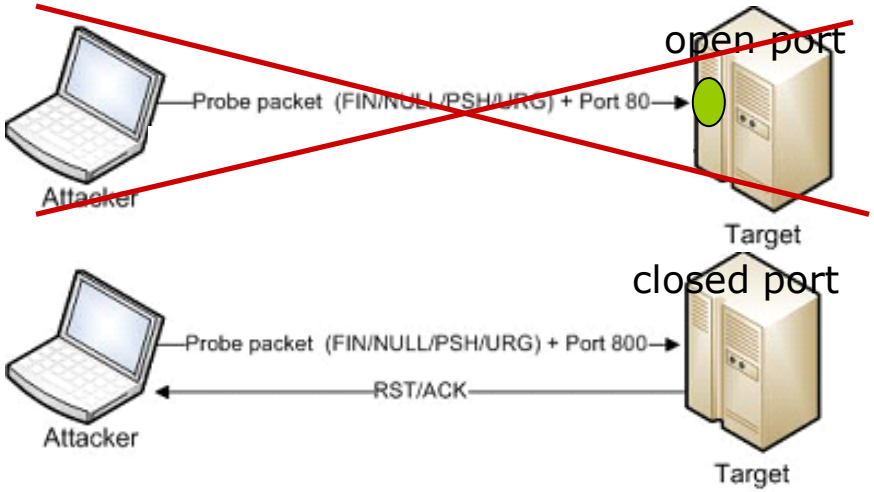
Nmap Fingerprint Methods

TCP FIN Probing

- TCP RFC 793 requires that a system with an open port ignores (not respond to) a FIN packet if received at the start of a connect.
- Microsoft Windows (Windows 7, Windows 2000, Vista) disregard this requirement and replies to the FIN packet with a RST packet



In Windows always response!



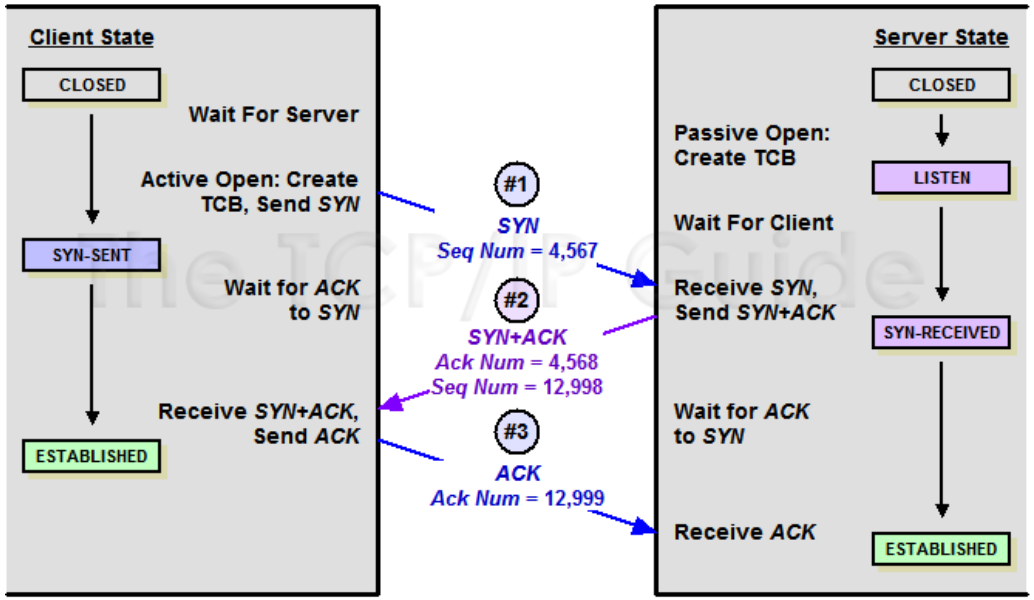
Probe a machine with a FIN packet on a port that is KNOW to be open.
If you receive a response → Windows OS!

Operating System Detection Tools (cont.)

Nmap Fingerprint Methods (cont.)

TCP Initial Sequence Number (ISN)

- when receiving a request to establish a connection, an OS must choose an ISN to respond and continue the 3-way handshake
- some OS choose ISN based on randomized values, while others (Windows) generate the ISN based on system's internal clock (ISN is incremented by a small fixed amount each time period)



http://www.tcpipguide.com/free/diagrams/tcp3waysynch.png

Operating System Detection Tools (cont.)

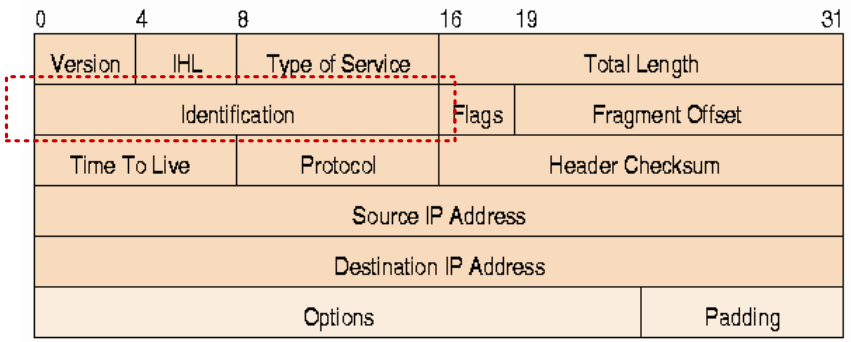
Nmap Fingerprint Methods (cont.)

TCP Initial Window Size

- some OSs are known to use a unique Window size
- e.g. Linux 2.4 IWS=5840 bytes, Linux 2.2 IWS=32120 bytes

IP ID Sampling

- Windows OS usually use a predictable IP ID sequence numbers, such as increasing the number by 1 or 256 for each packet
- other OS, e.g. Linux, randomize IP ID numbers



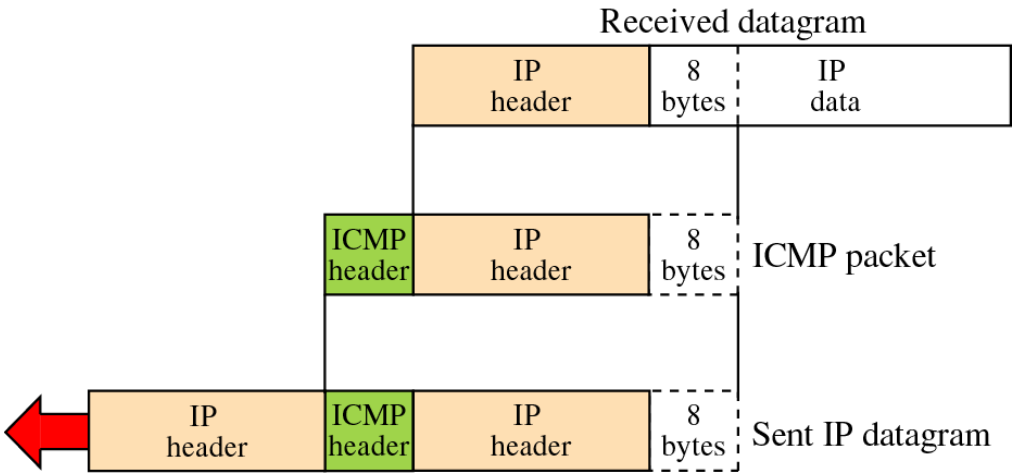
Operating System Detection Tools (cont.)

Nmap Fingerprint Methods (cont.)

ICMP Error Message Quoting

- according to ICMP RFC, OS must quote some parts of the original (ICMP) message - first 8 bytes - when generating an ICMP error message
- Linux and Solaris include much more information than required

e.g. UDP packet to a closed port



Operating System Detection Tools (cont.)

Nmap Fingerprint Methods (cont.)

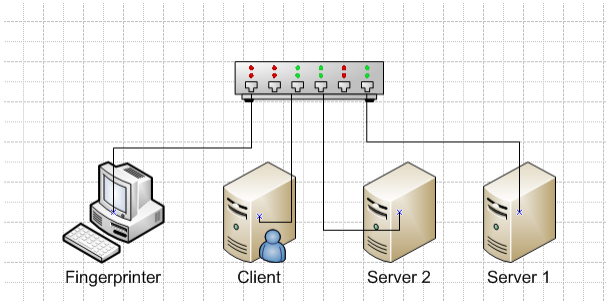
ICMP Error Message Quenching

- some operating systems follow the RFC 1812 suggestion to limit the rate at which various error messages are sent
- e.g. Linux limits destination unreachable message generation to 80 per 4 seconds, with a ¼ second penalty if that is exceeded
- to test a machine, send a bunch of packets to some random high UDP port and count the number of unreachables received

For more see: http://nmap.org/nmap-fingerprinting-article.txt

Operating System Detection Tools (cont.)

- **OS Detection Techniques in in Passive Fingerprinting** – **less intrusive way to gather info about the OS of a remote host**
  - instead of actively querying, attacker (only) sniffs remote host's packets
  - generally less precise/effective than active fingerprinting because:
    - have to accept whatever communication happens - there may not be much of it!
    - has fewer header parameters/options to work with than active fingerprinting
    - some of those parameters often get modified by firewall or proxy
  - Nmap does not passive fingerprinting per se – Idle Scan, the closest …



Operating System Detection Tools (cont.)

Passive Fingerprint Methods

Time-to-Live (TTL) in IP packets

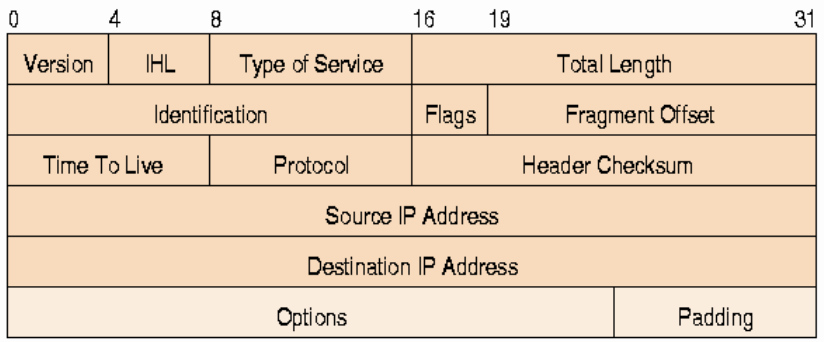
- normally **Linux** sets TTL = 64, and **Windows** TTL = 128

Don't Fragment Bit in IP packet

- Most systems set it to 1; in **OpenBSD** set to 0

Type of Service (TOS)

- Normally set to 0; a few OS reported using different value.
  (Generally not reliable as the TOS value is often set by application.)



Operating System Detection Tools (cont.)

Example Passive Fingerprinting

Assume a sniffer/attacker has capture a packet with the following parameters:

Time-to-Live (TTL): 51
TCP Window Size: 57344
Don't Fragment Bit: 1
Type of Service (TOS): 0

How would you go about determining the host's OS?

Solution:

Use Traceroute to determine the actual number of hops between itself and the host – say you have observed 13.

Hence, original TTL = 51 + 13 = 64.

Host's OS: (likely) Linux

Operating System Detection Tools (cont.)

- **OS Detection Counter-measures** – **to reduce chances of an OS being 'fingerprinted', OS's responses to various network requests/packets must be modified**

  - **IP Personality** – a patch for Linux kernel – allows changes to TCP/IP stack
    - **IP ID field, TCP Initial Window, TCP initial Sequence Number … values can be changed**

    http://ippersonality.sourceforge.net/

  - **Morph** and **IP Scrubber** – operate in firewall manner
    - **any traffic traveling from local net. will be 'scrubbed' & any OS-related information will be removed**

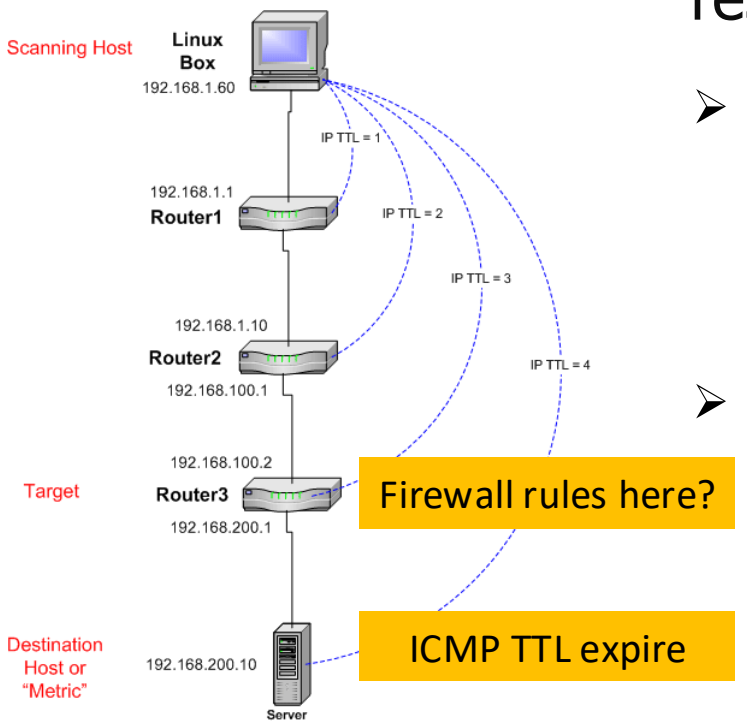
    http://www.synacklabs.net/projects/morph/

Reconnaissance Tools

0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Firewall Analysis Tools

- **Why Firewall Analysis?** – **help to understand / detect current set of a firewall's rules**
  - **Firewalk** – detects a firewall and its respective rules, in two phases
    - **phase 1: network discovery** through a traceroute-like procedure on ANY host inside firewall's network, TTL count to the target firewall is found
    - **phase 2: firewalking / scanning** TCP/UDP probe packets with TTL of 1-hop past firewall are sent
      - (a) if firewall does NOT allow packets in - packet will be dropped & firewall sends no message message back
      - (b) if firewall DOES allow packets in, ICMP TTL Expired message is sent by the binding host



Firewall Analysis Tools

Example

We want to test whether the firewall allows traffic for host H on port P in.

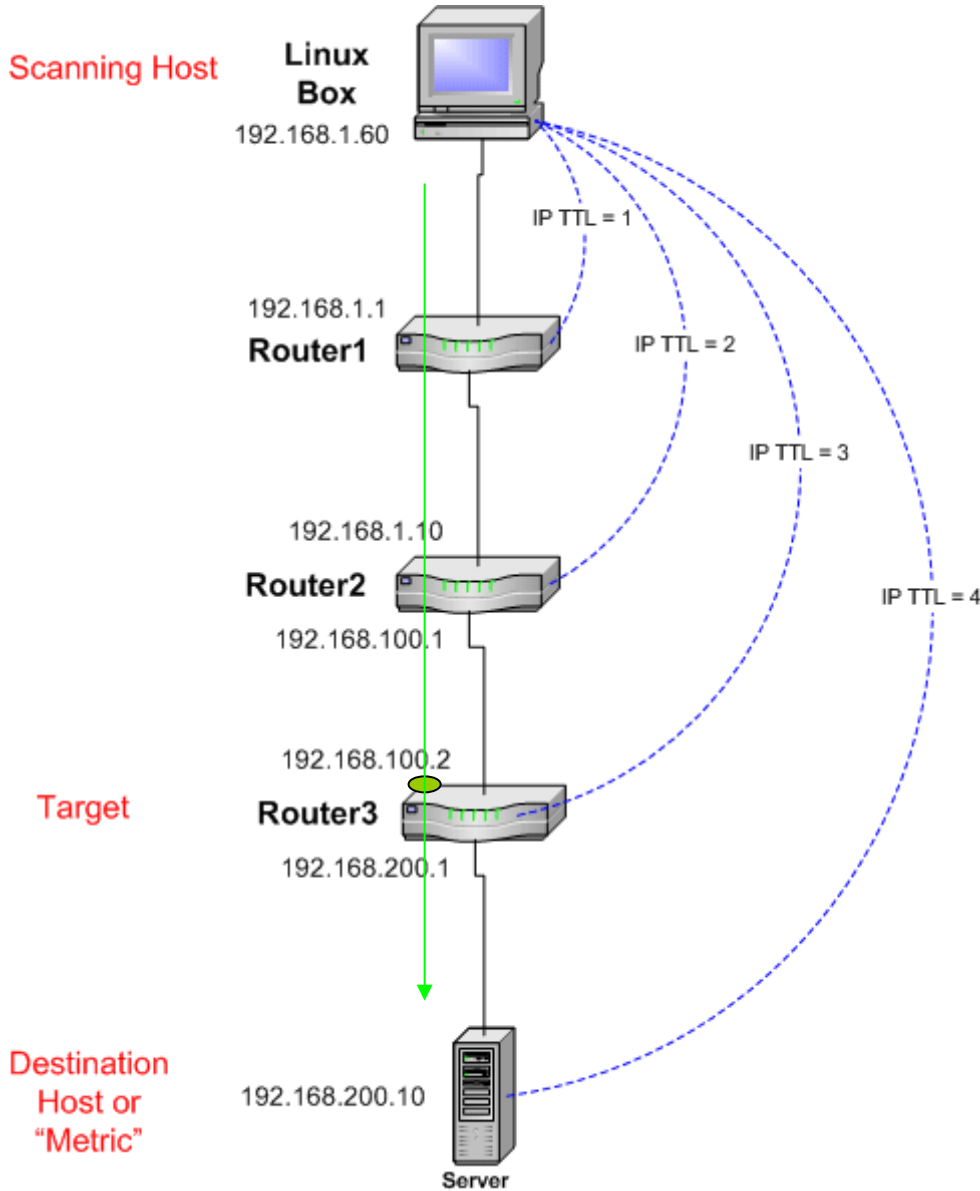
Hence, we send a UDP packet to H and port P with a very big TTL.

No response comes back.

What can we conclude?!

a) Firewall blocks packets intended for port P on H.

b) Firewall does not block such packet, but port P on on H is not open.



Firewall Analysis Tools (cont.)

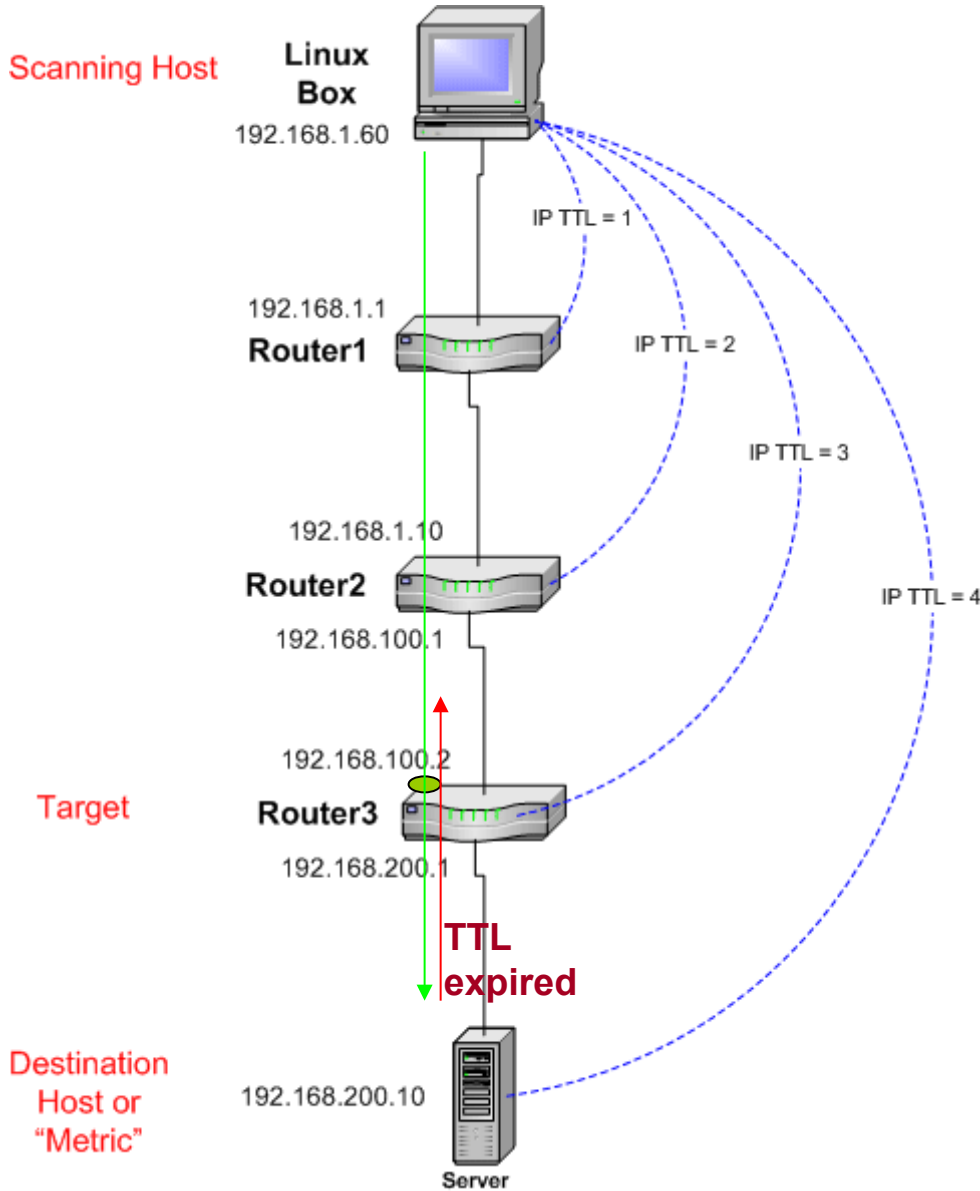
Now, assume we send the same UDP packet (to H, port P), but with $TTL=TTL_{firewall}+1$.

Possible outcomes:

a) Firewall blocks packets intended for port P on H, thus no response arrives back.

b) Firewall lets packet(s), in but the network-layer module of H's OS sends an ICMP TTL Expired error message back.

Outcomes a) and b) are different!!!



Firewall Analysis Tools (cont.)

Example: Firewalk (http://www.e-cq.net/wp/scanning-encored.pdf)

```
[bash]# traceroute www.gotrice.com
traceroute to www.gotrice.com (203.162.168.130), 30 hops max, 38 byte
1  202.155.7.1 (202.155.7.1)  80.147 ms  74.863 ms  59.949 ms
2  202.155.7.162 (202.155.7.162)  140.144 ms  139.960 ms  139.863 ms
3  202.84.154.57 (202.84.154.57)  144.885 ms  114.808 ms  109.942 ms
4  134.159.129.174 (134.159.129.174)  380.082 ms  334.85 ms  345.20 ms
5  203.162.231.233 (203.162.231.233)  349.61 ms  344.908 ms  354.983 ms
6  203.162.95.46 (203.162.95.46)  354.922 ms  339.922 ms  349.736 ms
7  203.162.168.130 (203.162.168.130)  365.106 ms 384.931 ms 354.884 ms

[bash]# firewalk -n -p TCP -S 21,22,23,25,53,80,110,143 203.162.95.46 \
203.162.168.130

Firewalk 5.0 [gateway ACL scanner]
Firewalk state initialization completed successfully.
TCP-based scan.
Ramping phase source port: 53, destination port: 33434
Hotfoot through 203.162.95.46 using 203.162.168.130 as a metric.
Ramping Phase:
1 (TTL 1): expired [202.155.7.1]
2 (TTL 2): expired [202.155.7.162]
3 (TTL 3): expired [202.84.154.57]
4 (TTL 5): expired [134.159.129.174]
5 (TTL 5): expired [203.162.231.233]
6 (TTL 6): expired [203.162.95.46]
Binding host reached.
Scan bound at 7 hops.
Scanning Phase:
port  21: A! open (port not listen) [203.162.168.130]
port  22: A! open (port listen) [203.162.168.130]
port  23: A! open (port not listen) [203.162.168.130]
port  25: A! open (port not listen) [203.162.168.130]
port  53: A! open (port not listen) [203.162.168.130]
port  80: A! open (port listen) [203.162.168.130]
port 110: A! open (port not listen) [203.162.168.130]
port 143: A! open (port not listen) [203.162.168.130]
```

firewall / gateway router

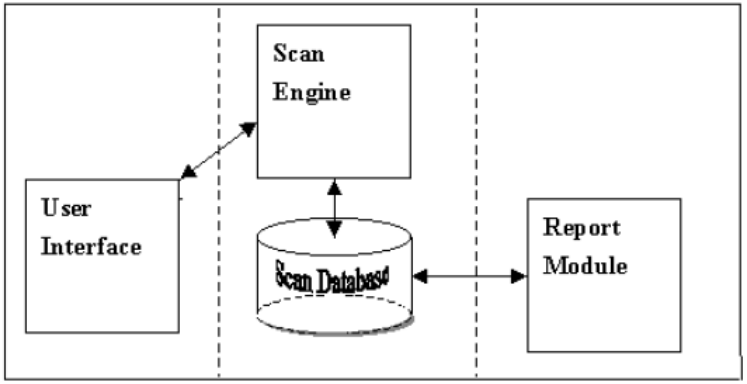
one known host on target network

Reconnaissance Tools

0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Vulnerability Scanners

- **Vulnerability Scanners** – **software tools that assess security vulnerabilities in networks & hosts & produce a set of scan results**
  - functionality of port scanners and more!
    - **e.g. tell you not only which ports are open, but also the name and version of software running on the port, and its vulnerabilities**
    - **components of a scanner:**

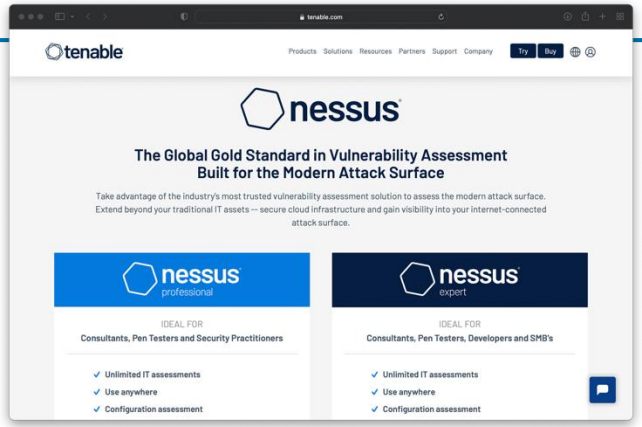


Components of Scanner

http://www.infosec.gov.hk/english/technical/files/vulnerability.pdf

Vulnerability Scanners (cont.)

Example:

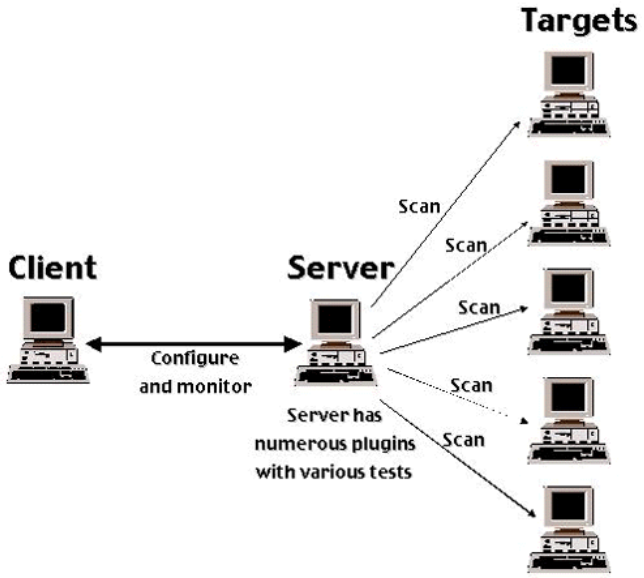


- Leader in vulnerability scanners – used by over 75,000 companies.
- Freeware! (not any more!)
- Can scan for vulnerabilities on either a local or a remote host.
- Comes in different flavors for UNIX, Mac and Windows.
- Able to detect:
  - open ports / available services
  - misconfigurations (e.g. missing patches)
  - default passwords
  - presence of viruses and back-door programs, etc.

Vulnerability Scanners (cont.)

Example: Nessus (cont.)

- Employs client-server architecture:
  - Nessus server includes a vulnerability database & a scanning engine.
  - Nessus client includes a user config. tool and a report-gener. tool.
  - Client & server can run on same or different machines (e.g. in case of a slow link).



Reconnaissance Tools

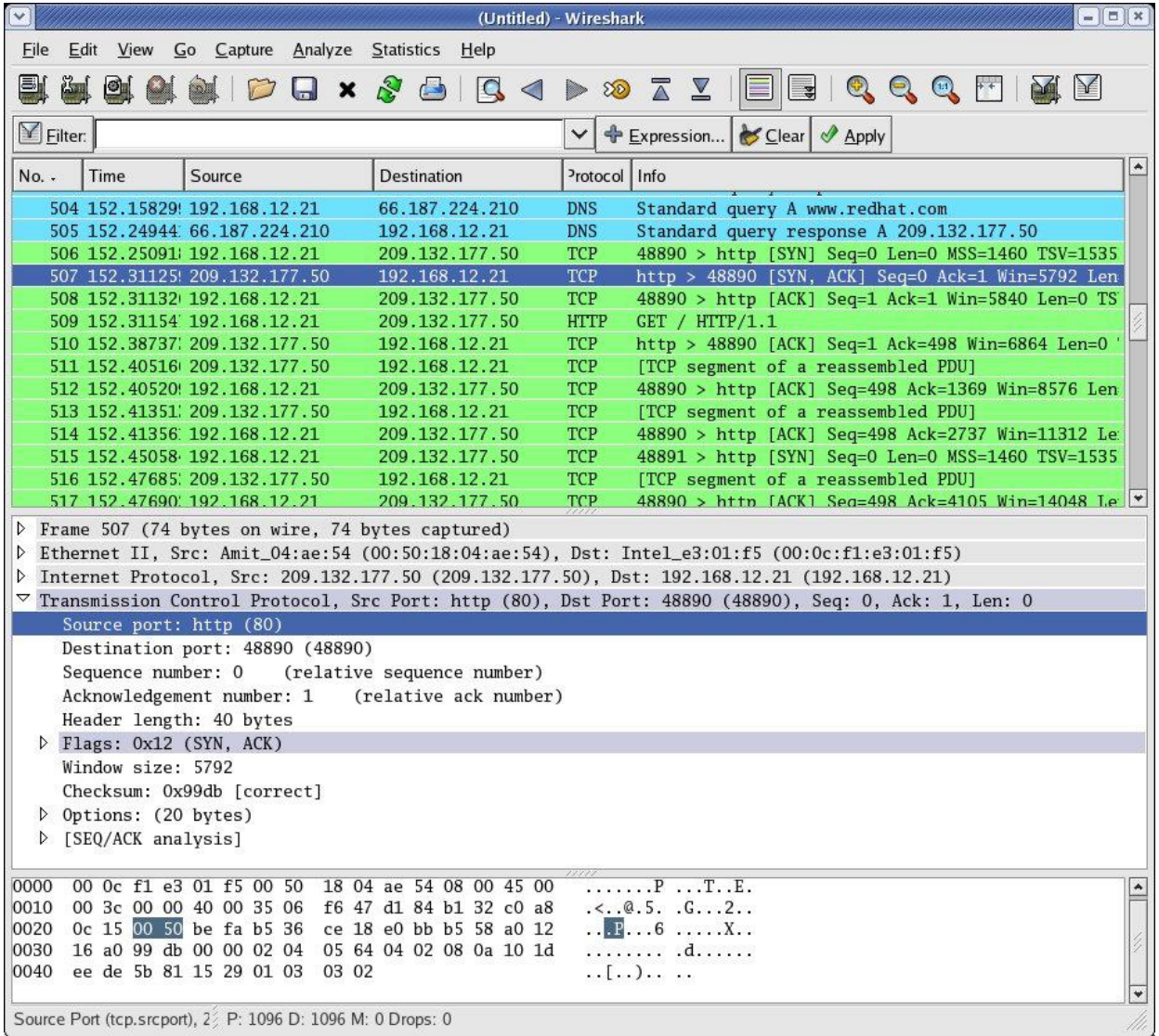
0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Packet Sniffers

- **Packet Sniffers** – **aka network protocol analyzers – collect copies of packets from a network and decode their content**
  - common use:
    - **troubleshooting – e.g. diagnose protocol configuration mistakes**
    - **network traffic characterization – obtain a picture of type and make of network traffic to fine-tune/manage bandwidth**
    - **security analysis – e.g. detect DoS attacks by observing a large number of specific-type packets**
  - to be able to 'sniff' all LAN packets packet sniffer should be put into promiscuous mode

Packet Sniffers (cont.)

Example: Wireshark



Reconnaissance Tools

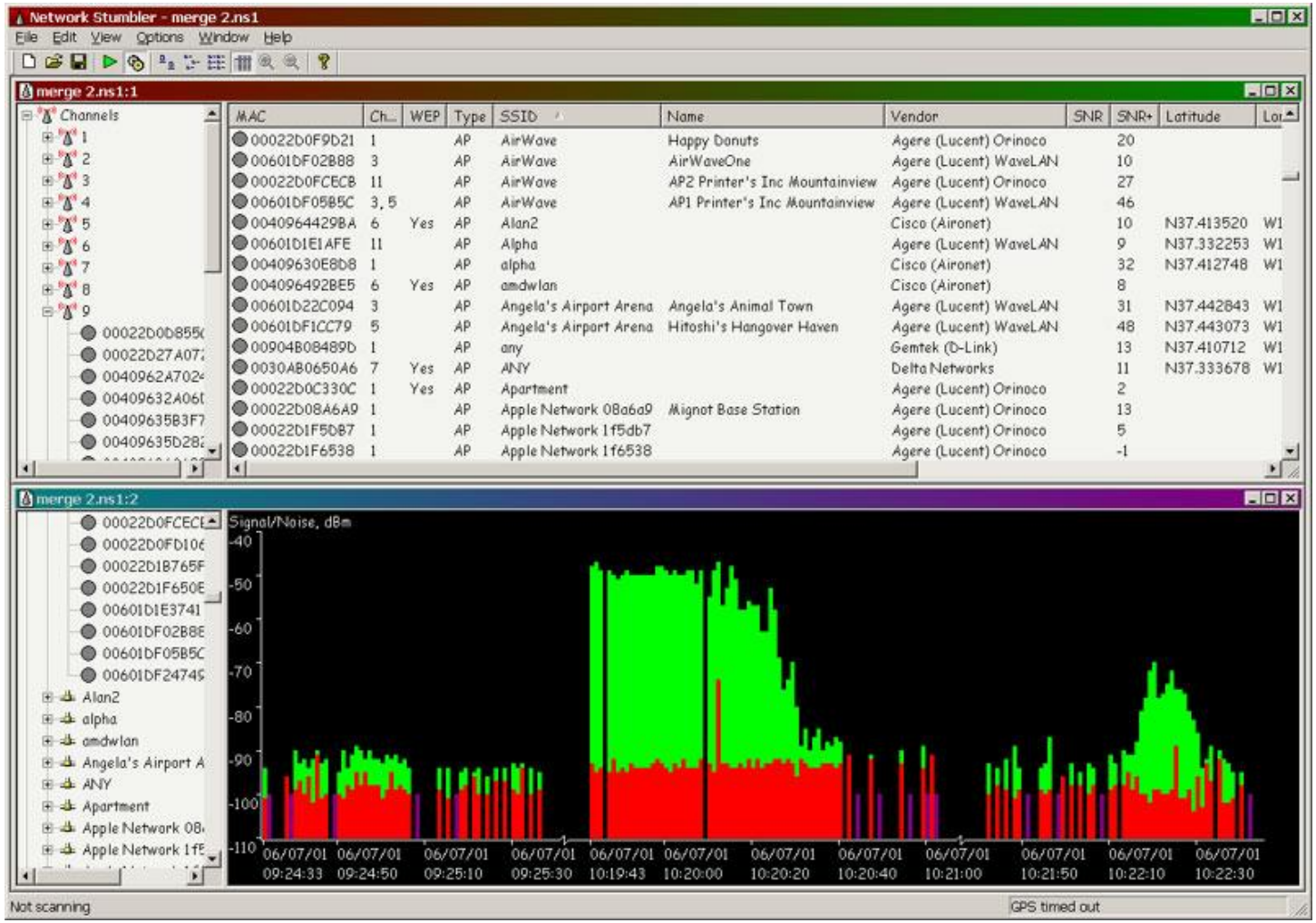
0) Public Reconnaissance Tools
1) Port Scanners
2) Network Mappers
3) Operating System Detection Tools
4) Firewall Analysis Tools
5) Vulnerability Scanners
6) Packet Sniffers
7) Wireless Sniffers

Wireless Sniffers

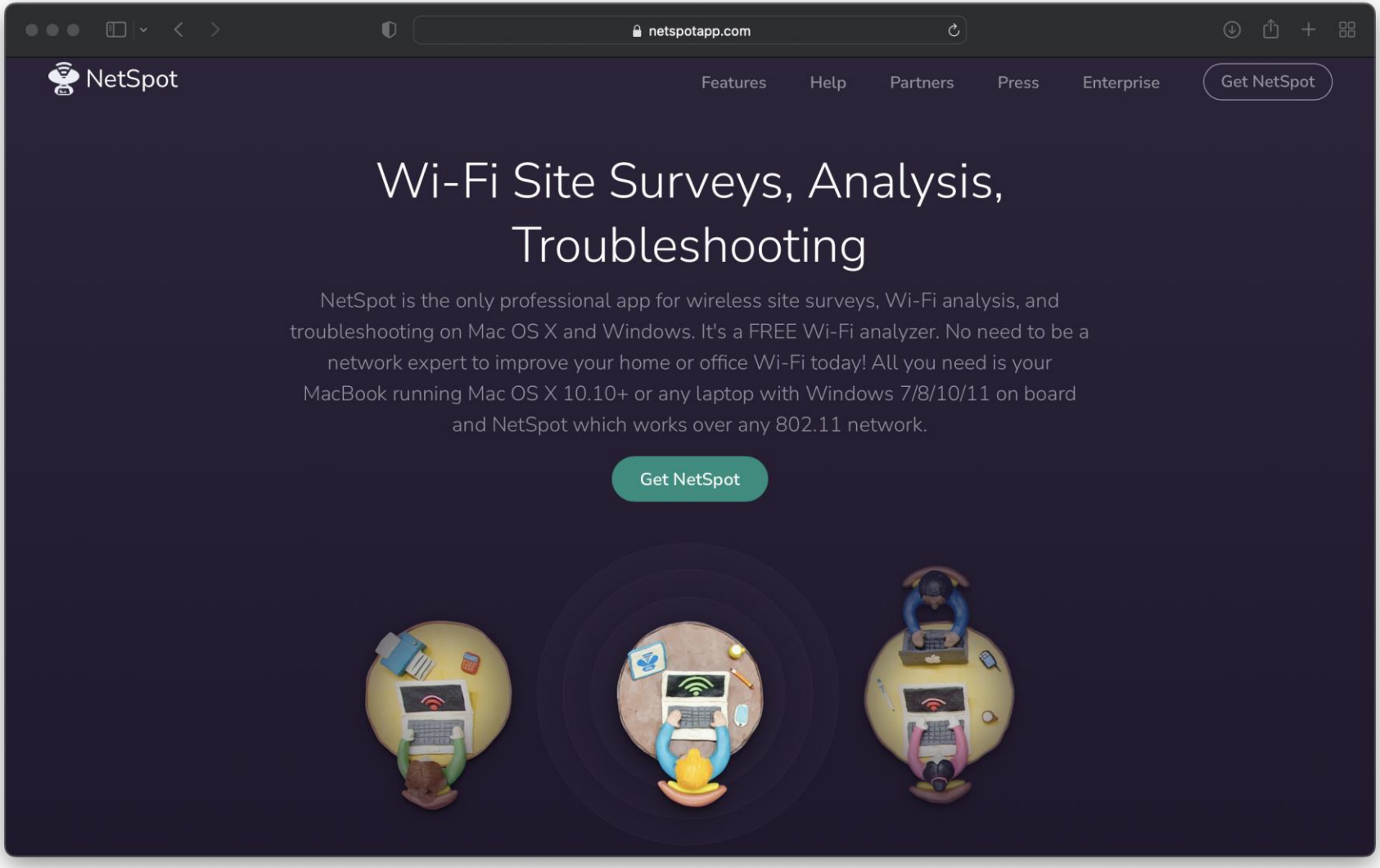
- **Wireless Sniffers** – **software / hardware capable of capturing & decoding packets as they pass over airwaves**
  - wireless sniffing is much easier than wired sniffing
    - **wireless medium is broadcast medium: everybody sees everything**
    - **in a wired network, the attacker must find a way to install a sniffer on a host or in a target subnet**
  - detection of wireless sniffing is extremely difficult – leaves no traceable evidence
  - name typically refers to WiFi (IEEE 802.11) sniffers

Wireless Sniffers (cont.)

Example: NetStumbler



NetSpot


netspotapp.com
NetSpot
Features
Help
Partners
Press
Enterprise
Get NetSpot
Wi-Fi Site Surveys, Analysis, Troubleshooting
NetSpot is the only professional app for wireless site surveys, Wi-Fi analysis, and troubleshooting on Mac OS X and Windows. It's a FREE Wi-Fi analyzer. No need to be a network expert to improve your home or office Wi-Fi today! All you need is your MacBook running Mac OS X 10.10+ or any laptop with Windows 7/8/10/11 on board and NetSpot which works over any 802.11 network.
Get NetSpot

End of Lecture 3