

Port Scanning

Open Source Security Tools for Information Technology Professionals

School of Professional Studies (SPS)

The City University of New York (CUNY)

Aron Trauring
Adjunct Professor
CEO, Zoteca

Class 7 November 7th, 2005

TCP/UDP Ports

- 16 bits — 1-65,535
- 0-1,023 — reserved for common applications
- Also known as **well-known** ports
- Assigned by IANA — Internet Assigned Numbers Authority
- Originally run by one man, Jon Postel
- Assigned Well Known Ports <http://www.iana.org/assignments/port-numbers>

Ephemeral (Dynamic, Private) Ports

- Every connection has two ends
- Server side is well-known port
- Client side is ephemeral assigned by IP stack
- Port taken for every connection
- Usually high end — 49152 through 65535 most common
- 32768-61000 — Linux
- 1024-4000 — Windows (problematic since it restricts number of connections)

Exploiting Open Ports

- Illegitimate traffic can come in on legitimate open ports
- Port 80 (web) is particularly vulnerable
- Buffer overflow — overflow allocated data storage buffer and change the data that follows buffer in memory

Buffer Overflow — Security Problem

- C subroutine pushes the return address onto the stack, so the subroutine knows where to return control to when it has finished
- When a dynamic buffer is allocated, the stack grows left by however big the buffer is.
- At the end of the function, the buffers are deallocated, everything pushed is popped, and a RET operation is called.
- This pops the return address off the stack and jumps there, shifting program control back to wherever the subroutine was called from.
- Overflow data will clobber the return address, overwriting part of it with the extra data.

- This changes where program control will go to continue execution when the subroutine has finished
- If random place in memory program “crashes” when the RET instruction attempts to jump control
- A technically inclined malicious user can ensure that the extra data points to a valid memory address, causing program control to be shifted to location of their choosing\
- There it can execute whatever arbitrary code the user has caused to be in that location with whatever privileges the currently executing program has

Buffer Overflow — History

- Morris Worm (1988) — first Internet worm (exploited fingerd)
- Phrack (1996) — "Smashing the Stack for Fun and Profit"
- Coder Red Work (2001) – exploited IIS (which kept administrative privileges after binding port 80)
- SQLSlammer (2003)

Buffer Overflow — Programming Solutions

- Choice of language — C and C++ do no memory checking; Java, Python, C# do
- Choice of safe libraries — particularly string handling routines
- Executable Space protection (kernel patches and processors)
- Stack Smashing protection (compilers)
- **Apply security patches!**

Buffer Overflow — Second-best Solutions

- Close unnecessary ports
- Reduce the number of services running to only what is needed

Port Scanners

- Poll TCP and UDP ports to see if an application answers back

- Simplest: `telnet <IP address> <portno>`

```
telnet microsoft.com 80  
a<enter>
```

- Port scanner tools scan multiple ports, even ephemeral ports

Why Do Full Scan

- Not all vendors stick to standards
- Malware may be hiding in high ports

TCP Fingerprinting

- Each vendor implements TCP/IP stack for its own OS
- Response to queries might differ based on OS and version
- Useful information for attackers
- Not always accurate (borrowed code, embedded OSes)

Port Scanning Considerations

- Network intensive so may degrade performance or shut computers down
- Legality not clear (is this considered an attempt to break in?)
- May be banned by ISP terms of service
- Don't do without permission
- iptables flag rules may block
- New iptables patch:

```
iptables -A INPUT -m psd -j DROP
```

Uses for Port Scanning

- Network inventory — check for number of machines and IP addresses in use
- Network/server optimization — discover services that are unnecessary
- Discovering malware
- Discovering unauthorized and illicit services

Malware

- Spyware — track user activities and report back to a central server
- Trojan Horse — back door for cracker, needs open port to communicate back (usually high port)
- Network Worms — open up ports as they spread from computer to computer

Unauthorized Services

- P2P
- IM
- Employee installed