6.829 Security Recitation

Rob Beverly <rbeverly@mit>

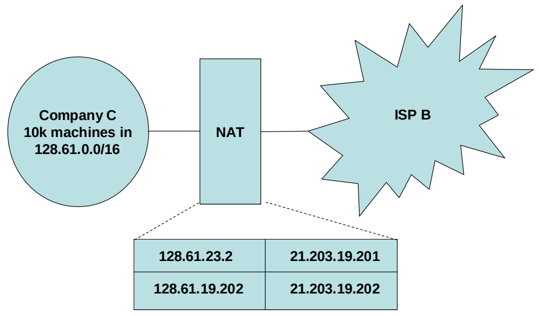
November 17, 2006

Network Address Translation

Basic NAT

- Basic NAT rewrites either the source or destination addresses of IP packets
- Example:
  - Translate private RFC1918 address to public
- Example:
  - company C with ~10k machines in 128.61.0.0/16
  - 128.61.0.0/16 owned by provider A (non-portable)
  - company moves to ISP B
  - B gives C 21.203.0.0/16
  - C doesn't want to renumber

Basic NAT Example

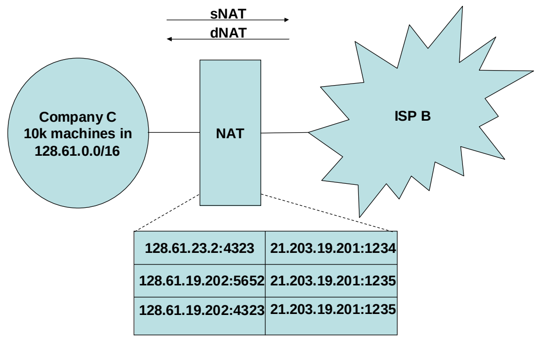


128.61.23.2	21.203.19.201
128.61.19.202	21.203.19.202

NAT with Port Translation

- Rewrite IP address and Transport Port number
- NAT maintains state on IP and port
- Allows for overloading *n* IPs into *m* IPs
- dNAT: destination NAT
- sNAT: source NAT
- dNAT, sNAT often used together
- Example: Company C is given *m=1* IP from ISP B

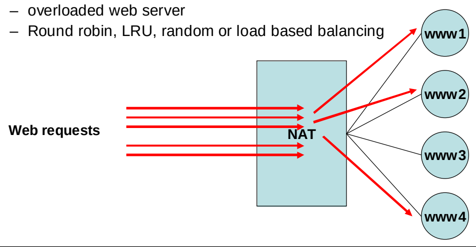
NAT with Port Translation



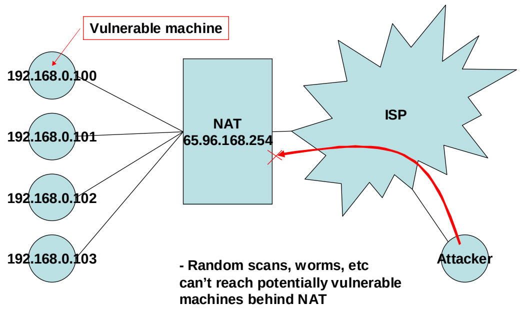
128.61.23.2:4323	21.203.19.201:1234
128.61.19.202:5652	21.203.19.201:1235
128.61.19.202:4323	21.203.19.201:1235

Other NAT Applications

- Load balancing
- Failover
- Example:
  - overloaded web server
  - Round robin, LRU, random or load based balancing

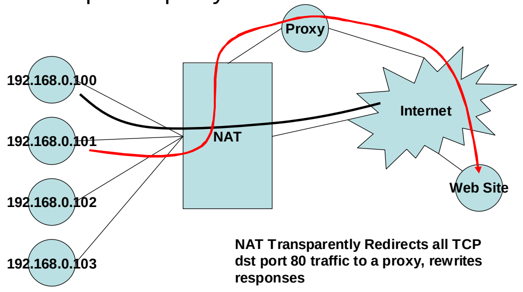


NAT Security(?)

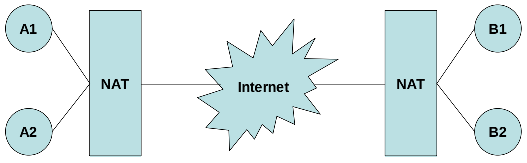


Other NAT Applications

- Transparent proxy



NAT End-to-End Brokenness



- How can A1 establish a TCP connection to B1?

NAT Architectural Issues

- NATs are architecturally contentious
- Advantages:
  - Slows IPv4 address depletion (disadvantage to IPv6 adoption)
  - Map many machines to one public address
  - Security
- Disadvantages:
  - Breaks end-to-end communication model
  - Breaks some applications

Applications Affected by NAT

- Requires E2E:
  - P2P
- Network layer information in application:
  - SIP
  - FTP
- Example: FTP server uses port 20 for TCP control connections. In active mode, client and server negotiate port on client; server initiates connection to client, but has bad info

Resource Attack Defenses

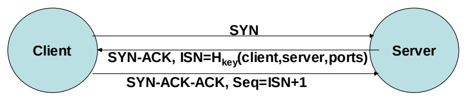
TCP SYN Cookies

- Server's SYN ACK ISN =
  - H_{key}(src_addr, src_port, dst_addr, dst_port)
  - Where H is cryptographic secure hash
- Note, that some TCP options are negotatied in 3-way handshake. If server keeps no state, throws away TCP options
- TCP options not a problem in practice:
  - Server only uses SYN cookies when under duress, SYN queue overloaded
- Note, no need for client TCP implementation to change
- Frequency of key rotation?

Resource Attack Defenses

- *Fundamental problem*: attacker can force victim to do work without doing any work itself
- Defenses:
  - Postpone devoting resources as late as possible
    - Ex: SYN cookies
  - Force attacker to do work first
    - Ex: Client puzzles; we'll see some of this next lecture

TCP SYN Cookies



key **is secret private to server**

On receipt of SYN-ACK-ACK, server checks:
seq-1 = H_{key}(client,server,ports)

TCP SYN Flood

- Client sends TCP SYN to server
  - Client source may be spoofed, so that client never receives SYN-ACK
  - Client may ignore SYN-ACK
- Result: half-open connection
- Server maintains connection state even when client source may be spoofed!
- Must maintain state in case SYN-ACK was lost, congestion, etc
- Client may exhaust server's SYN queue

Worms

Worms

- Worm: self-propagate across the Internet by exploiting security flaws
- Viruses: require user action to propagate
- Worm example: Code Red found and exploited vulnerable Windows IIS web servers on the Internet
- Worms interesting to research community because of the rate of spread

Worms: Permutation Scanning

- Propagation technique
- How to propagate exhaustively without duplicating work?
- Permutation Scanning:
  - All worms share common pseudo random permutation of IP address space
  - Infected via hit-list: Start scanning after their point in permutation
  - Infected via permutation scan: Start scanning at random point in permutation
  - Whenever worm sees an already infected machine, choose a new random start point

Worms

- Worms may be modeled just like infectious diseases
- How to bootstrap?
  - Corresponds to speed of infection
- How to propagate?
  - Corresponds to spread of infection
- Worm designers often mess up at least one of the two; we've been lucky

Permutation Benefits

- Scan all space
- Self-coordinating using local information
- Maintain benefits of random probing
- Keep infection rate high

Worms: Hit-list Scanning

- Bootstrap technique
- How to find initially vulnerable hosts to start spread?
- Hit-list scanning:
  - Build list of vulnerable targets
  - Stealth port scans
  - Distributed scans
  - DNS searches
  - Spiders
  - Just listen

Warhol Worm

- Andy Warhol: "*In the future everyone will have 15 minutes of fame*"
- Hit-list + Permutation: Warhol worm
- Warhol worms shown able to infect all vulnerable machines possible on Internet in less than 15 minutes
- Faster than any human solution: dictates requirement for automated response mediation

Other Types of Worms

- Flash worm
  - Worm maintains a complete hit list of all vulnerable machines
  - May require large worm
  - More efficient techniques exist
- Contagion worm
  - Slowly spreading
  - Exhibit no communication pattern
  - Example: infected server has two exploits Ec and Es. A vulnerable client goes to server and is infected via Ec. Client then gets copies of Ec and Es. Next server that client visits which is vulnerable to Es gets infected and receives copies of Ec and Es.

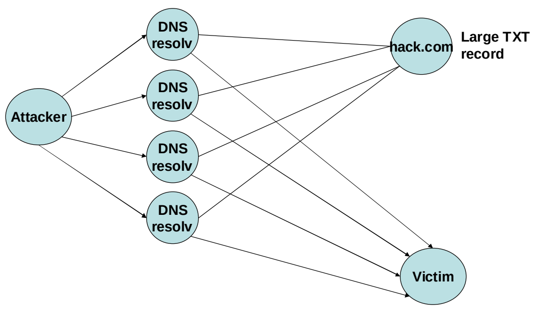
IP Address Spoofing

- Several recent new attacks:
  - DNS Reflector Attacks
  - In-Window TCP RST
  - Spam filter circumvention

Packet Filtering

- NOTE:
  - (Ran out of time; This topic not covered in recitation)
  - (Read if curious about filtering and some recent IP spoofing attacks)

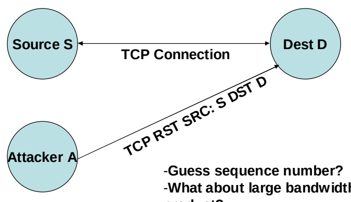
Spoofing: DNS Reflector Attack



Packet Filtering

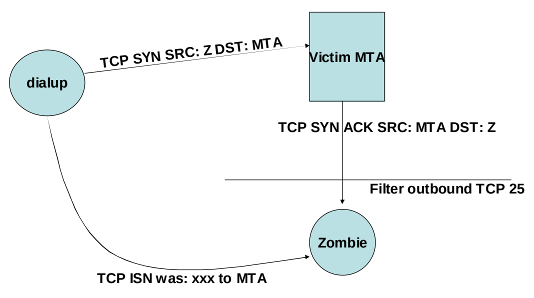
- One defense for blocking “bad” traffic: packet filtering on routers
- Ingress, egress filters
- Examples of traffic to block:
  - RFC1918
  - Bogons
  - Spoofed Packets
- Generally done at edge because of traffic asymmetry problem
- Hard because an ISP can properly filter, but still receive bogus traffic
- By filtering, an ISP doesn’t necessarily protect itself from bogus traffic

Spoofing: In-Window TCP RST



-Guess sequence number?
-What about large bandwidth delay product?
-BGP router exploit

Spoofing: Spam Circumvention


dialup
TCP SYN SRC: Z DST: MTA
Victim MTA
TCP SYN ACK SRC: MTA DST: Z
Filter outbound TCP 25
Zombie
TCP ISN was: xxx to MTA