# CSci 5271 Introduction to Computer Security Day 21: Malware and Denial of Service

Stephen McCamant University of Minnesota, Computer Science & Engineering

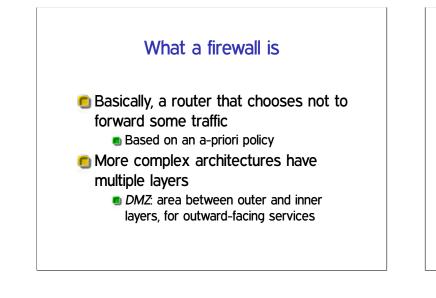
# Outline

Firewalls and NAT boxes

Intrusion detection systems

Malware and the network

Denial of service and the network



# Inbound and outbound control

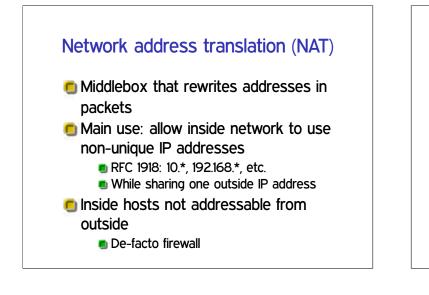
Most obvious firewall use: prevent attacks from the outside
 Often also some control of insiders
 Block malware-infected hosts
 Employees wasting time on Facebook
 Selling sensitive info to competitors
 Nation-state Internet management
 May want to log or rate-limit, not block

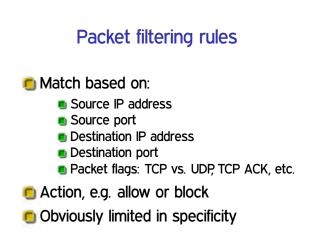
# Default: deny

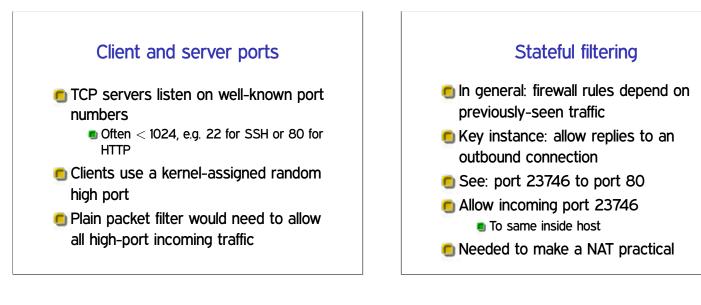
- Usual whitelist approach: first, block everything
- 🖲 Then allow certain traffic
- Basic: filter packets based on headers
- More sophisticated: proxy traffic at a higher level

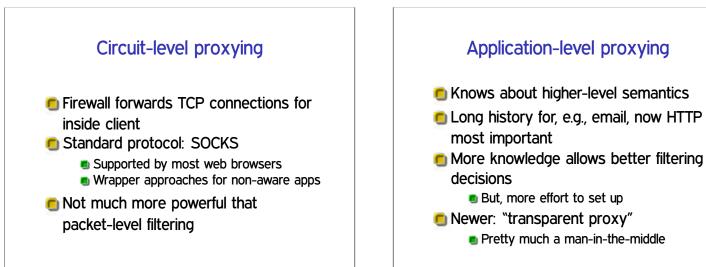
# IPv4 address scarcity

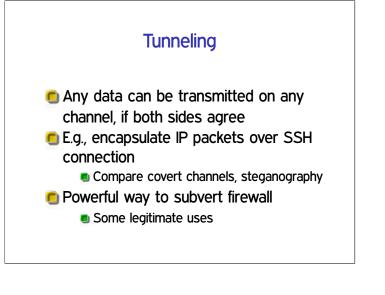
- Design limit of 2<sup>32</sup> hosts
   Actually less for many reasons
- Addresses becoming gradually more scarce over a many-year scale
- Some high-profile exhaustions in 2011
- IPv6 adoption still very low, occasional signs of progress











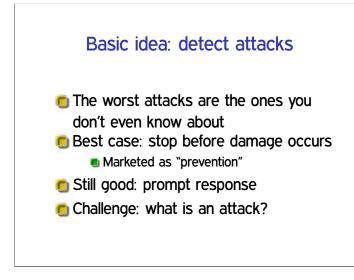
# Outline

Firewalls and NAT boxes

Intrusion detection systems

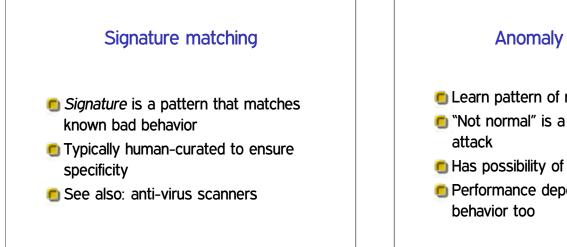
Malware and the network

Denial of service and the network



# Network and host-based IDSes

- Network IDS: watch packets similar to firewall
  - But don't know what's bad until you see it More often implemented offline
- Host-based IDS: look for compromised process or user from within machine



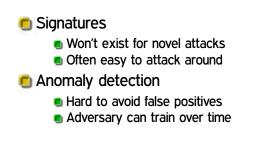


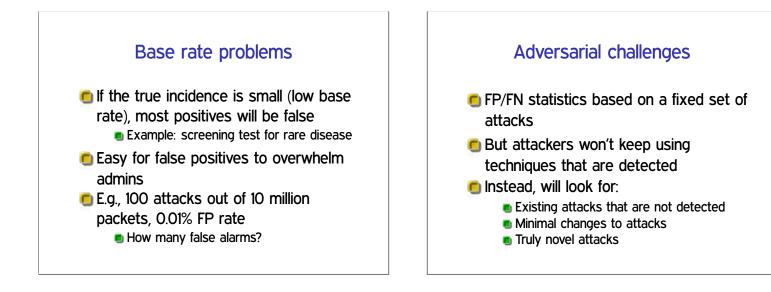
- Learn pattern of normal behavior
- "Not normal" is a sign of a potential
- Has possibility of finding novel attacks
- Performance depends on normal

# Recall: FPs and FNs

- False positive: detector goes off without real attack
- False negative: attack happens without detection
- Any detector design is a tradeoff between these (ROC curve)

# Signature and anomaly weaknesses





# Wagner and Soto mimicry attack Host-based IDS based on sequence of syscalls Compute A ∩ M, where: A models allowed sequences M models sequences achieving attacker's goals Further techniques required:

- Many syscalls made into NOPs
  - Replacement subsequences with similar
    - effect

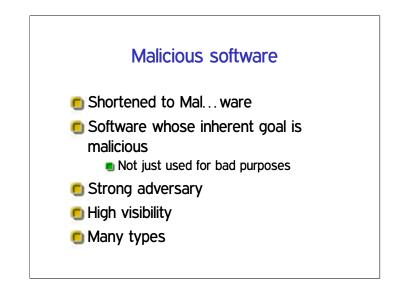
# Outline

Firewalls and NAT boxes

Intrusion detection systems

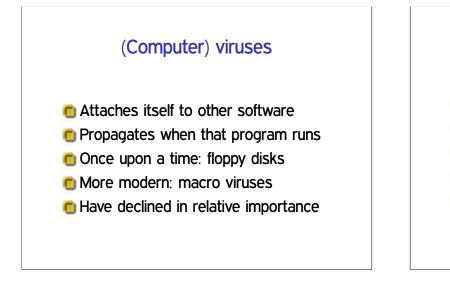
Malware and the network

Denial of service and the network



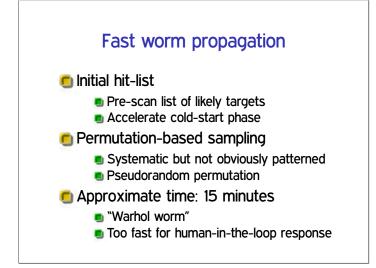
# Trojan (horse)

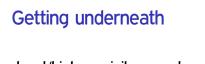
- Looks benign, has secret malicious functionality
- Key technique: fool users into installing/running
- Concern dates back to 1970s, MLS



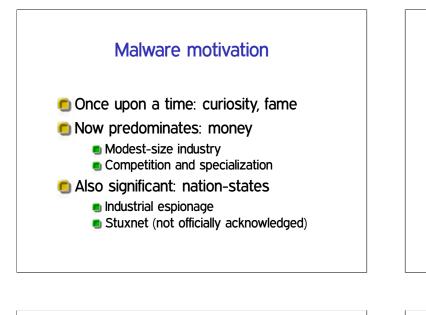
# Worms

- Completely automatic self-propagation
- Requires remote security holes
- 🖲 Classic example: 1988 Morris worm
- "Golden age" in early 2000s
- Internet-level threat seems to have declined





- Lower-level/higher-privilege code can deceive normal code
- Rootkit: hide malware by changing kernel behavior
- MBR virus: take control early in boot
- Blue-pill attack: malware is a VMM running your system

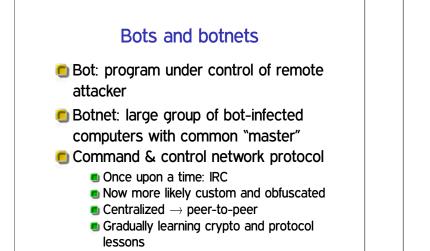


# User-based monetization

- Adware, mild spyware
- Keyloggers, stealing financial credentials

## Ransomware

- Application of public-key encryption
- Malware encrypts user files
- Only \$300 for decryption key



# Bot monetization

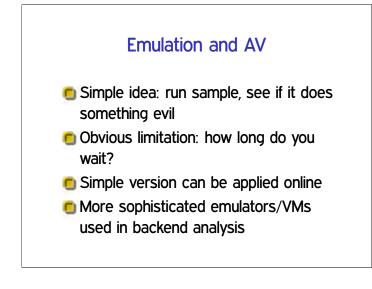
- 🖲 Click (ad) fraud
- Distributed DoS (next section)
- Bitcoin mining
- Pay-per-install (subcontracting)
- Spam sending

# Malware/anti-virus arms race

- "Anti-virus" (AV) systems are really general anti-malware
- Clear need, but hard to do well
- No clear distinction between benign and malicious
- Endless possibilities for deception

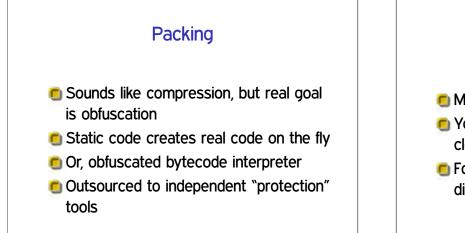
# Signature-based AV Similar idea to signature-based IDS Would work well if malware were static In reality: Large, changing database Frequent updated from analysts Not just software, a subscription

Malware stays enough ahead to survive



# Polymorphism

- Attacker makes many variants of starting malware
- Different code sequences, same behavior
- One estimate: 30 million samples observed in 2012
- But could create more if needed



# Fake anti-virus

- Major monentization strategy recently
- Your system is infected, pay \$19.95 for cleanup tool
- For user, not fundamentally distinguishable from real AV

# Outline

Firewalls and NAT boxes

Intrusion detection systems

Malware and the network

Denial of service and the network

# DoS versus other vulnerabilities

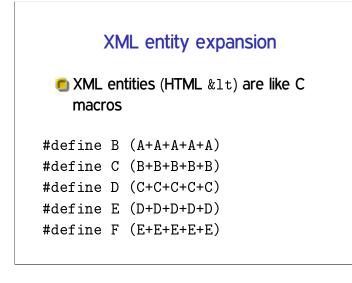
- Effect: normal operations merely become impossible
- Software example: crash as opposed to code injection
- Less power that complete compromise, but practical severity can vary widely
   Airplane control DoS, etc.

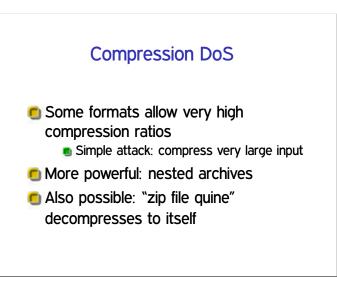
# When is it DoS?

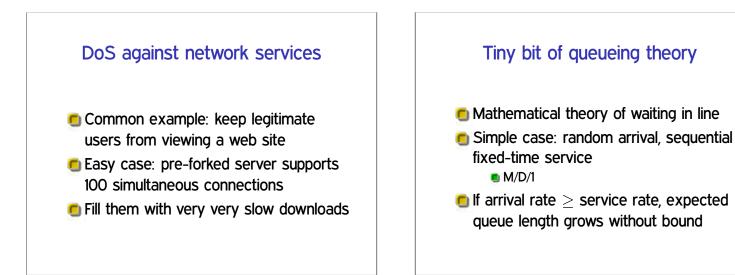
- Very common for users to affect others' performance
- Focus is on unexpected and unintended effects
- Unexpected channel or magnitude

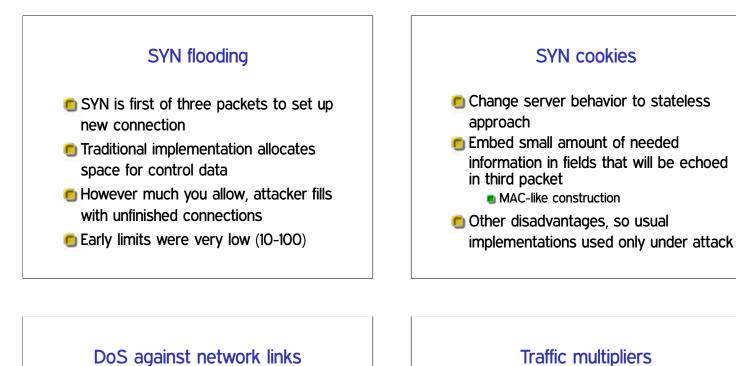
# Algorithmic complexity attacks

- Can an adversary make your algorithm have worst-case behavior?
- $\bigcirc O(n^2)$  quicksort
- Hash table with all entries in one bucket
- Exponential backtracking in regex matching







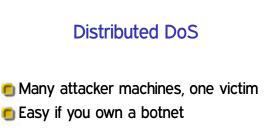


- Try to use all available bandwidth, crowd out real traffic
- Brute force but still potentially effective
- Baseline attacker power measured by packet sending rate

- Third party networks (not attacker or victim)
- One input packet causes n output packets
- Commonly, victim's address is forged source, multiply replies
- Misuse of debugging features



- ICMP echo request with forged source
- Sent to a network broadcast address
- Every recipient sends reply
- Now mostly fixed by disabling this feature



- Impractical to stop bots one-by-one
- May prefer legitimate-looking traffic over weird attacks
  - Main consideration is difficulty to filter

# Next time

Networking layers to protect privacy