Network Security

CSE 561 Lecture 9, Spring 2002.
David Wetherall

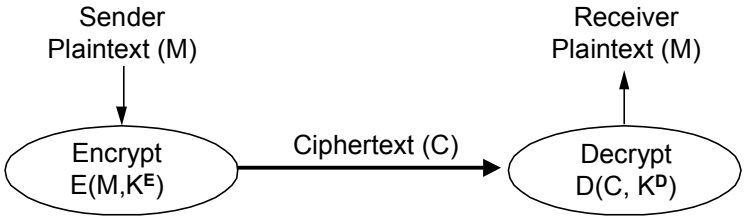
What is network security?

- Protecting information
  - Confidentiality
  - Integrity
  - Authenticity / Non-repudiation
- Protecting systems
  - Access (who is authorized to do what)
  - Availability (!denial of service)
  - Containment (detecting compromises, limiting their effects)
- These are very broad categories.

Why is it challenging?

- Fragility
  - Security is a negative goal. Any vulnerability (design, implementation, configuration) can defeat it.
  - Implementation flaws are a big deal in practice, e.g. viruses
- Exposure
  - The Internet is shared with many, mostly anonymous parties, e.g., IP address spoofing complicates denial of service
  - Compare to a standalone banking network …

Basic Cryptography (Peterson 8-1-8.3)



- Cryptographer chooses functions E, D and keys K^E, K^D
  - Mathematical basis rather than assumed secrecy of method
  - Private keys support encryption
  - Public keys along with a PKI support authentication
- These solutions are based on trust; the key is the principal

Example Systems

- We can protect information at different levels.

- Pretty Good Privacy (PGP)
  - For authentic and confidential email
- Secure Sockets (TLS nee SSL) and Secure HTTP (HTTPS)
  - For secure Web transactions
- IP Security (IPSEC)
  - Framework for encrypting / authenticating IP packets

Security Problems in TCP/IP [Bellovin89]

- Primarily concerned with protocol vulnerabilities relating to authenticity
  - Key issue: source addresses are taken at face value without strong evidence or proof

- Vulnerabilities
  - Sequence number guessing
  - Source routing
  - Routing protocol attacks
  - ICMP attacks
  - DNS, ARP

TCP Sequence # prediction

- Problem:
  - Many applications use IP address for access control (WebAuth, r-commands)
  - Easy to spoof IP address; TCP requires port and seq#
  - If you can guess initial sequence number (ISN) then can create "fake" TCP sessions as well
- Blind spoofing
  - Attacker->Server: SYN(ISNa) [spoof client]
  - Server->Client: SYN(ISNs), ACK(ISNa) [what happens?]
  - Attacker->Server: ACK(ISNs) [spoof client]
  - Attacker->Server: "echo" "*" >> ~/.rhosts" [spoof client]
  - Attacker -> Server: RST [spoof client]

TCP sequence # prediction

- How hard is to guess ISN?
- Traditional systems (< 1999)
  - Increment ISN by constant over time
  - Very easy to predict
- Most modern TCP stacks
  - Random increment
  - Still predictable (need more trials)
- Cryptographically secure RNG makes this hard
  - Overhead

Source routing

- Problem:
  - If source IP address is used for authentication, then attacker can pretend to be trusted host but route through attacker
- Solution:
  - Disable source routing

Routing attacks

- Problem: Attacker may advertise bogus routes
  - Claim to originate network / host
  - Intercept packets then re-route to true destination
  - May also cause denial-of-service
- Solutions
  - Policy about which routes you believe (don't accept routes for own network); have well-known neighbors
  - Authentication of routing protocol sessions
  - Open research problem to handle this problem efficiently…

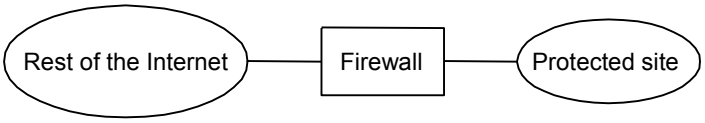
ICMP attacks

- Used to report errors / exceptional conditions from network to end hosts
- Problem: spurious ICMP messages
  - ICMP Redirect (optimization): send traffic to alternate router
  - ICMP TTL Exceeded, Dest / Net Unreachable: kill connection
- Solutions: ad hoc
  - Don't accept redirect (or only from same subnet)
  - Match packet body on ICMP errors

DNS/ARP

- Problem: name translation (DNS->IP and MAC->IP) is vulnerable to spoofing
  - DNS sequence # prediction (must also guess client port) allows attacker to spoof DNS server reply
  - Attacker can spoof reply to ARP who-has requests to intercept host traffic on same LAN
- Solutions:
  - Better DNS sequence # generation
  - No good solution currently for ARP spoofing (switches help)

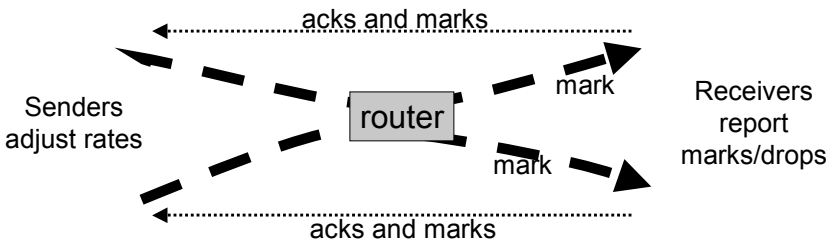
Firewalls – security in practice



- Firewalls selectively sever or allow connectivity between protected site and outside world based on a site policy.
- E2E access checks based on crypto authentication (rather than IP address) work fine in theory. But firewalls are it in practice! Why?
  - E2E solutions aren't deployed; a PKI is required
  - Centralized application and control of policy
- Intrusion detection systems are also used to spot attacks

Ex: Congestion Signaling with ECN

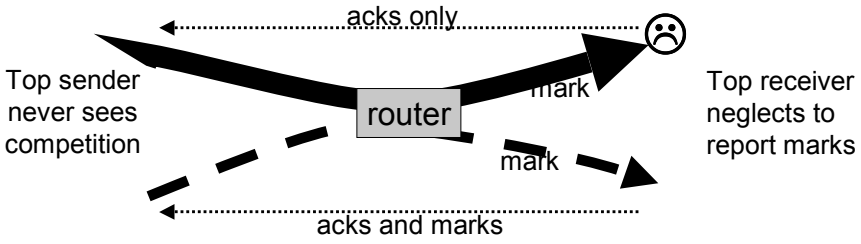
- Competing senders adjust the rate of TCP connections to share bandwidth based on router feedback (drops).



- With Explicit Congestion Notification (ECN), routers mark (rather than drop) packets to signal congestion.

Problem – marks can be erased

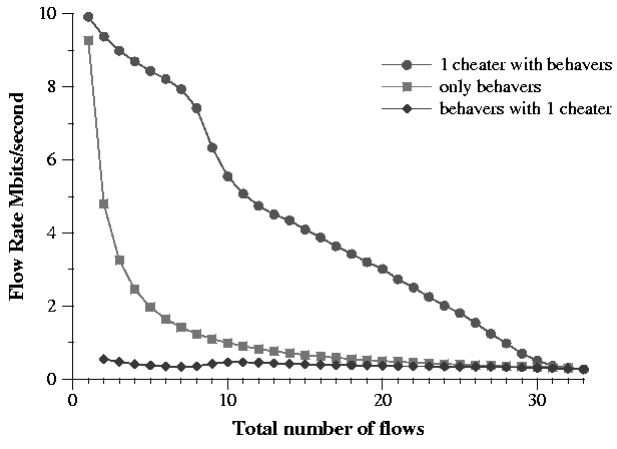
- Unlike drops, marks on packets can be erased, causing the sender to send too fast through no fault of its own



- During testing, VPNs/firewalls were found to do this.

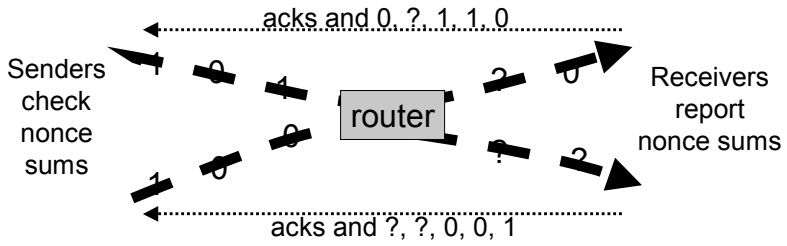
And it makes a big difference

- Bad receiver gets up to 10X throughput at others' expense



Solution – Robust Congestion Signaling (ICNP'01)

- Senders attach nonces to packets, which routers erase to mark. Receivers report nonce sums to prove no congestion.



- Now bugs slow faulty connection, but not others.
- This is the new IETF ECN design

Denial of Service

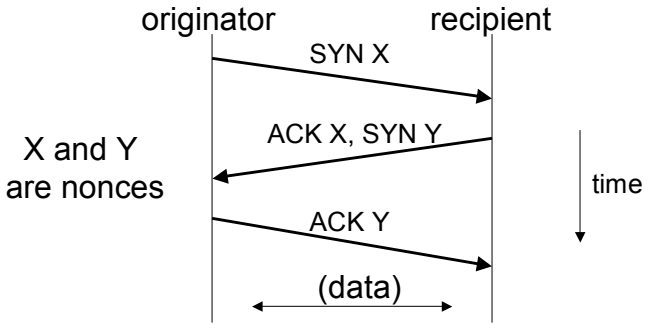
- Consume sufficient resources to render system unavailable, thus denying service to legitimate users.
- Q: Does authentication solve this problem?

- Serious problem in the Internet today
  - Lack of accountability (IP address spoofing, reflectors)
  - Ease of marshalling attack (amplifiers, zombies)
  - Lack of control (can't stop people sending you packets)

- This is a network problem, requiring network solutions
  - Ingress filtering, traffic "pushback"

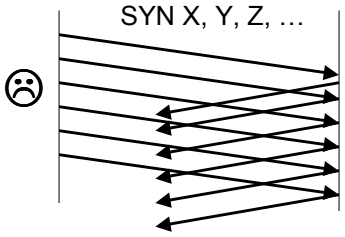
Ex: TCP Connection Establishment

- Two parties need to "SYNchronize" to form a connection
- TCP uses a three way handshake – for classic reliability!



Problem – SYN flooding

- If originator doesn't follow through, it burdens the recipient. Used for denial-of-service starting ~1996 through today.

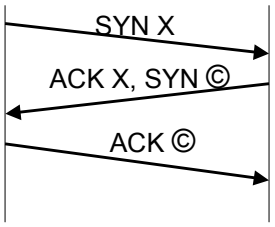


State kept for incomplete connections can exhaust resources …

- (Plus, if nonces are predictable then fake connections can be forged.)

Solution – offload state with cookies

- Don't keep the state. Send it back and let the originator return it later. But state must be an opaque, verifiable cookie …



No per connection state during handshake

- Linux SYN cookies ('97) : reply nonce © is used to carry a cookie. It is securely hashed with a secret so it can be checked on return.

Ingress filtering and Pushback

- Ingress filtering
  - Strict RPF check: Validate that source address is contained as next-hop in forwarding table on interface receiving packet; else drop packet
    - Only appropriate at network edges
  - Loose RPF check: Just validate source address is in forwarding table
    - More widely applicable, but less helpful
  - Automatically blocks many spoofed source address
    - But requires near-universal deployment to be effective
    - And doesn't stop attacks using legitimate addresses
- Pushback
  - Preferential drop of unwanted traffic at routers
  - Push drop requests back router-by-router from point of overload

Summary

- Security is a huge field, poorly fleshed out
- Mostly based on trust
  - Authenticity, confidentiality, integrity to establish trust with outsider
  - Firewalls / IDS define trusted vs untrusted infrastructure
  - If you don't have trust, these measures don't help
- ***Every*** protocol in use today likely has security holes
  - We don't design for the adversary
- How many of the flaws we discussed today still exist?