Module 5: Network Attacks II

Nitesh Saxena

Adopted from previous lectures by Keith Ross

Overview of the Module

L1 Sniffing

L2 Spoofing

L3 Session Hijacking

L4 DoS and DDoS

L5 Connection and Bandwidth Flooding

L6 DNS Attacks

Module 5, Lecture 1

Sniffing

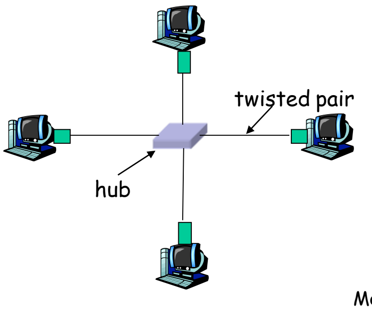
Interconnection devices

- Hubs
- Switches
- Routers

Hubs

Hubs are essentially physical-layer repeaters:

- bits coming from one link go out all other links
- at the same rate
- no frame buffering
- no CSMA/CD at hub: adapters detect collisions
- provides net management functionality



Sniffing

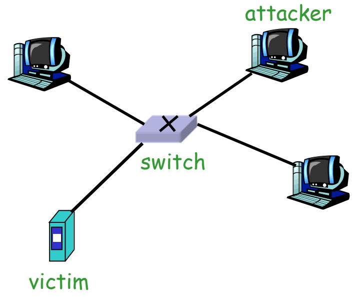
- Attacker is inside firewall
- Requirements
  - Attacker's host connected to shared medium
  - NIC should be in "promiscuous mode"
    - processes all frames that come to NIC
- Sniffer has two components
  - Capture
  - Packet analysis
- Grab and file away:
  - userids and passwords
  - credit card numbers
  - secret e-mail conversations
- Island hopping attack:
  - Take over single machine (eg virus)
  - Install sniffer, observe passwords, take over more machines, install sniffers

Passive sniffing

- Easy to sniff:
  - 802.11 traffic
  - Ethernet traffic passing through a hub
    - Any packets sent to hub is broadcast to all interfaces
    - Not true for a switch
- Popular sniffers
  - Wireshark
  - tcpdump (for unix)
  - Snort (sniffing and intrusion detection)

Active Sniffing through a switch

How does attacker sniff packets sent to/from the victim?



Have to get victim's packets to attacker!

Sniffing through a switch: flooding switch memory approach

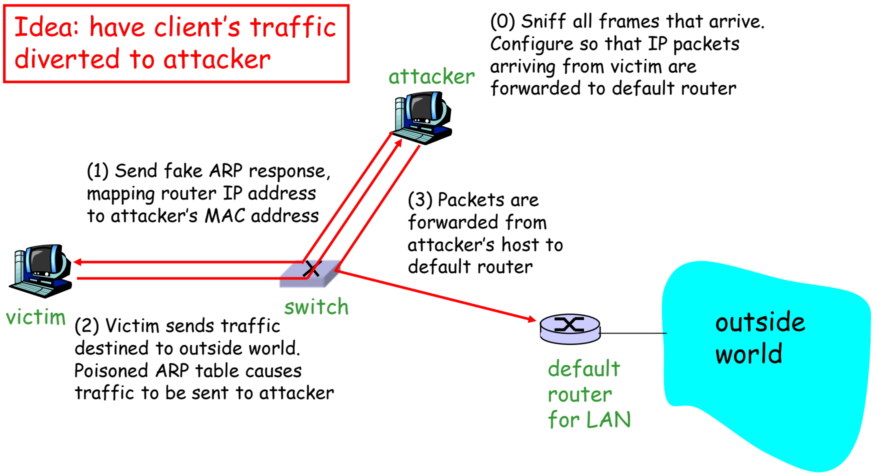
Host sends flood of frames with random source MAC addresses

- Switch's forwarding table gets filled with bogus MAC addresses
- When "good packet arrives," dest MAC address not in switch memory
- Switch broadcasts real packets to all links

- Sniff all the broadcast packets

Sniffing through LAN: poison victim's ARP table approach

Idea: have client's traffic diverted to attacker



Powerful sniffing tools

- Dsniff and ettercap
  - Flooding switch memory
  - ARP poisoning

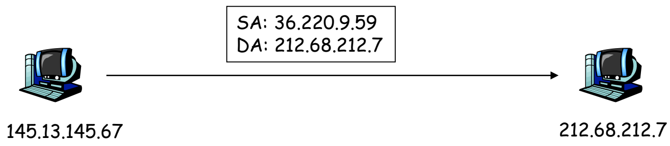
Sniffing defenses

- Encrypt data: IPsec, SSL, PGP, SSH
- Use encryption for wireless
- Get rid of hubs: complete migration to switched network
- Configure switches with MAC addresses
  - Turn off self learning (knowing mappings between ports and MAC addresses)
  - Eliminates flooding problem
- Intrusion detection systems:
  - Lookout for large numbers of ARP replies
- Honeypot
  - Create fake account and send password over network
  - Identify attacker when it uses the password

Module 5, Lecture 2

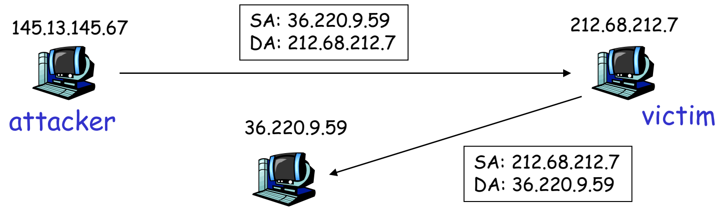
Spoofing

IP address spoofing (1)



- Attacker doesn't want actions traced back
- Simply re-configure IP address in Windows or Unix.
- Or enter spoofed address in an application
  - e.g., decoy packets with Nmap

IP address spoofing (2)

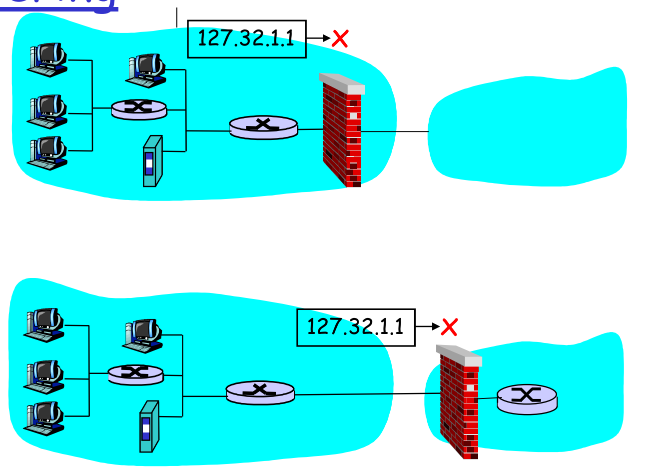


- But attacker cannot interact with victim.
  - Unless attacker is on path between victim and spoofed address.

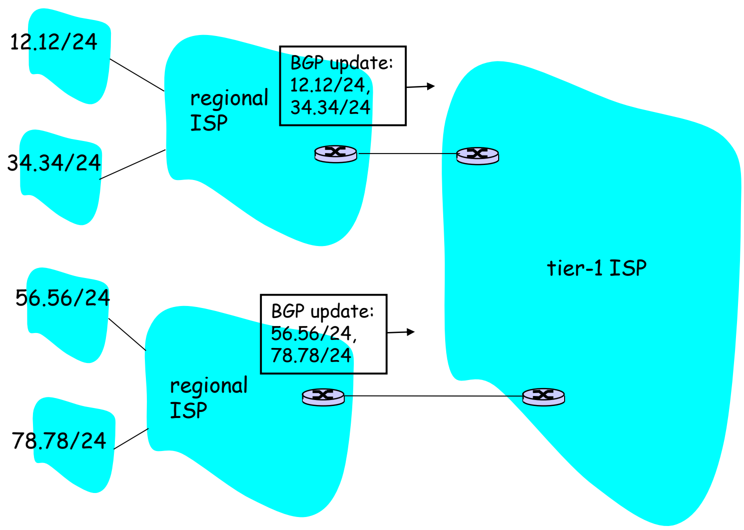
IP spoofing with TCP?

- Can an attacker make a TCP connection to server with a spoofed IP address?
- Not easy: SYNACK and any subsequent packets sent to spoofed address.
- If attacker can guess initial sequence number, can attempt to send commands
  - Send ACK with spoofed IP and correct seq #, say, one second after SYN
- But TCP uses random initial sequence numbers.

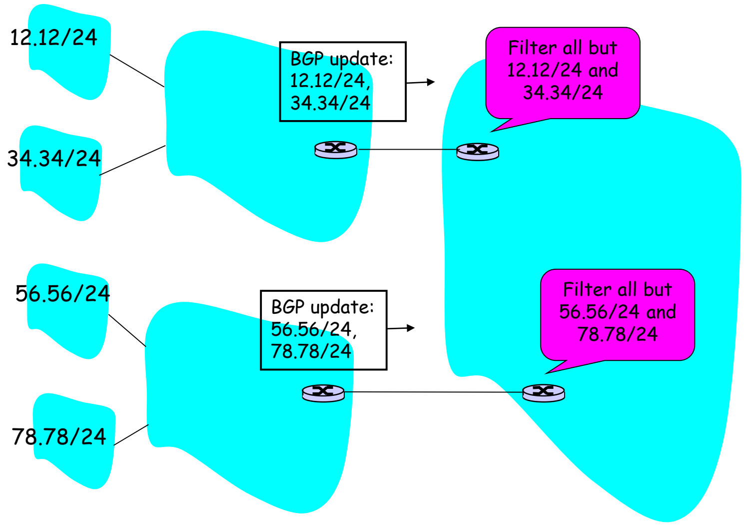
Defense: Ingress and egress filtering



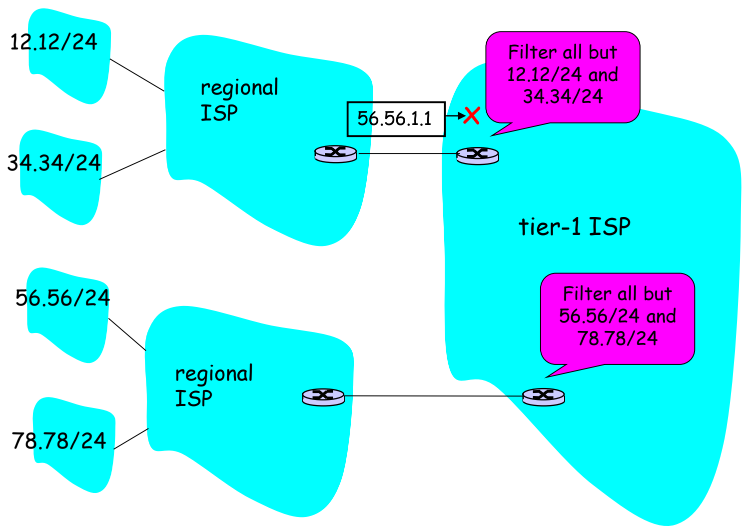
Ingress Filtering: Upstream ISP (1)



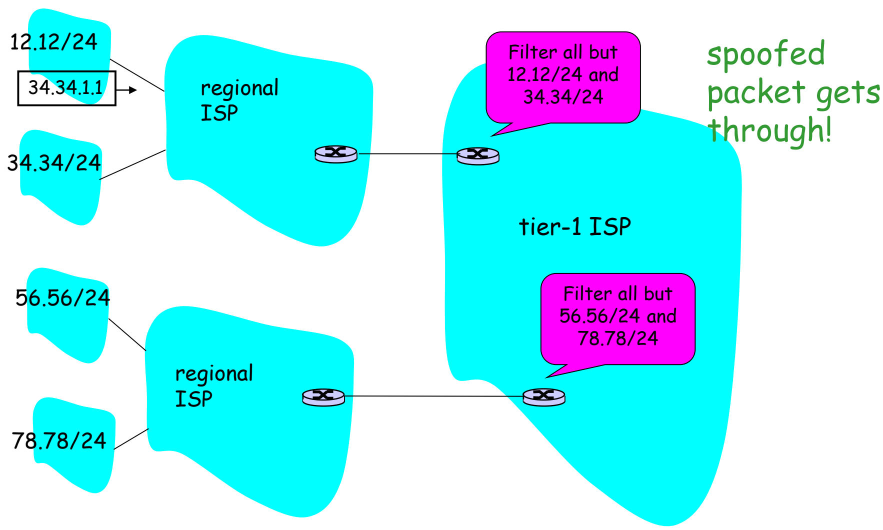
Ingress Filtering: Upstream ISP (2)



Ingress Filtering: Upstream ISP (3)



Ingress Filtering: Upstream ISP (3)



Ingress filtering: summary

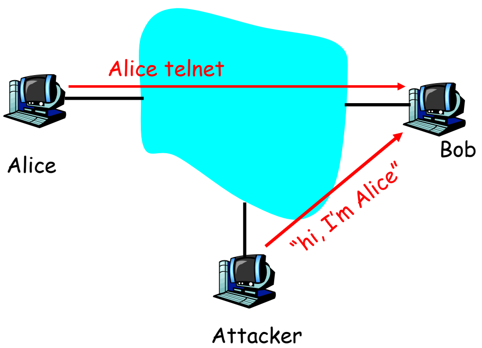
- Effectiveness depends on widespread deployment at ISPs
- Deployment in upstream ISPs helps, but does not eliminate IP spoofing
  - Filtering can impact router forwarding performance
- Even if universally deployed at access, hacker can still spoof another address in its access network 12.12/24
- See RFC 2827 "Network Ingress Filtering: Defeating DDoS"

Module 5, Lecture 3

Session Hijacking

Session hijacking

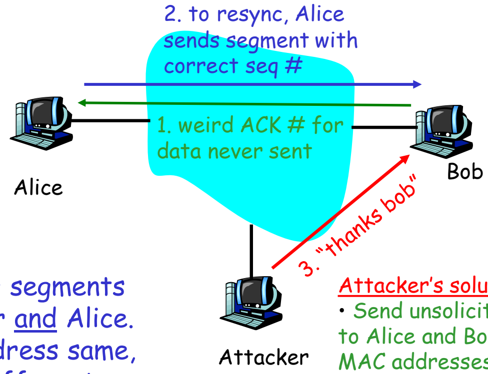
- Take control of one side of a TCP connection
- Marriage of sniffing and spoofing



Session hijacking: The details

- Attacker is on segment where traffic passes from Alice to Bob
  - Attacker sniffs packets
  - Sees TCP packets between Bob and Alice and their sequence numbers
- Attacker jumps in, sending TCP packets to Bob; source IP address = Alice's IP address
  - Bob now obeys commands sent by attacker, thinking they were sent by Alice
- Principal defense: encyrption + MAC
  - Attacker does not have keys to encrypt/authenticate and insert meaningful traffic

Session hijacking: limitation



Bob is getting segments from attacker and Alice. Source IP address same, but seq #'s different. Bob likely drops connection.

Attacker's solution:
- Send unsolicited ARP replies to Alice and Bob with non-existent MAC addresses
- Overwrite IP-to-MAC ARP tables
- Alice's segments will not reach Bob and vice-versa
- But attacker continues to hear Bob's segments, communicates with Bob

Session Hijacking Tools:

- Hunt
  - https://packetstormsecurity.com/sniffers/hunt
  - Provides ARP poisoning
- Netcat
  - General purpose widget
  - Very popular

Module 5, Lecture 4

DoS and DDoS

Denial-of-Service

Prevent access by legitimate users or stop critical system processes

- Implementation Vulnerability attack:
  - Send a few crafted messages to target app that has vulnerability
  - Malicious messages called the "exploit"
  - Remotely stopping or crashing services
- Connection flooding attack
  - Overwhelming connection queue with SYN flood
- Bandwidth flooding attack:
  - Overwhelming communications link with packets
  - Strength in flooding attack lies in volume rather than content

DoS and DDoS

- DoS:
  - source of attack small # of nodes
  - source IP typically spoofed
- DDoS
  - From thousands of nodes
  - IP addresses often not spoofed
- Good book:
  - Internet Denial of Service by J. Merkovic, D. Dittrich, P. Reiher, 2005

DoS: examples of vulnerability attacks

see http://www.cert.org/advisories/CA-1997-28.html

- Land: sends spoofed packet with source and dest address/port the same
- Ping of death: sends oversized ping packet
- Jolt2: sends a stream of fragments, none of which have offset of 0. Rebuilding consumes all processor capacity.
- Teardrop, Newtear, Bonk, Syndrop: tools send overlapping segments, that is, fragment offsets incorrect.

Patches fix the problem, but malformed packet attacks continue to be discovered.

LAND

- Local Area Network Denial
- Spoofed SYN packet with source and destination both being the victim
- On receipt, victim's machine keep on responding to itself in a loop
  - Causes the victim to crash
- Many OSs are vulnerable, e.g.,
  - Windows 95, NT, XP SP2
  - Mac OS MacTCP

Ping of Death

- ICMP Echo Request (Ping) is 56 bytes
- If a ping message is more than 65536 bytes (max for IP packet), this can cause some machines to crash
- Older windows systems

Solution: patch OS, filter out ICMP packets

"Teardrop", "Bonk" and kins

- TCP/IP fragments contain Offset field
- Attacker sets Offset field to:
  - overlapping values
    - Bad/old implementation of TCP/IP stack crashes when attempting to re-assemble the fragments
  - … or to very large values
    - Target system crashes

Solution: use up-to-date TCP/IP implementation

Module 5, Lecture 5

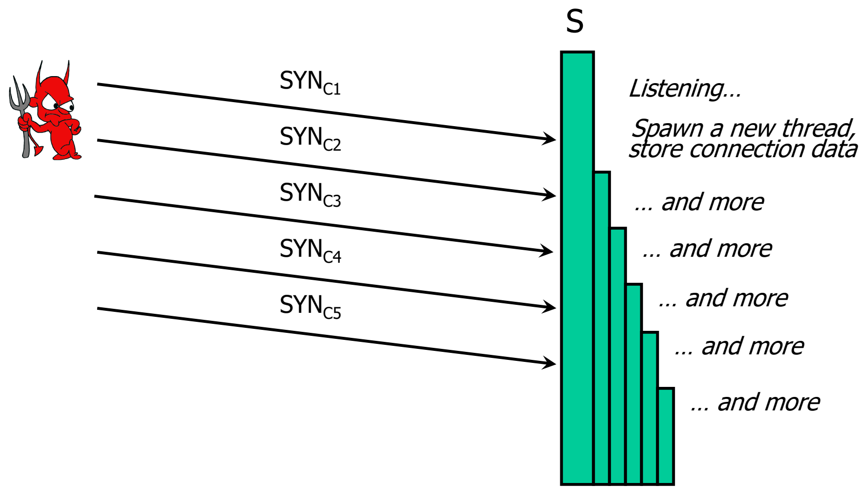
Connection and Bandwidth Flooding

Connection flooding: Overwhelming connection queue w/ SYN flood

- Recall client sends SYN packet with initial seq. number when initiating a connection.
- TCP on server machine allocates memory on its connection queue, to track the status of the new half-open connection.
- For each half-open connection, server waits for ACK segment, using a timeout that is often > 1 minute
- *Attack:* Send many SYN packets, filling connection queue with half-open connections.
  - Can spoof source IP address!
- When connection queue is exhausted, no new connections can be initiated by legit users.

Need to know of open port on victim's machine: Port scanning.

SYN Flooding Attack

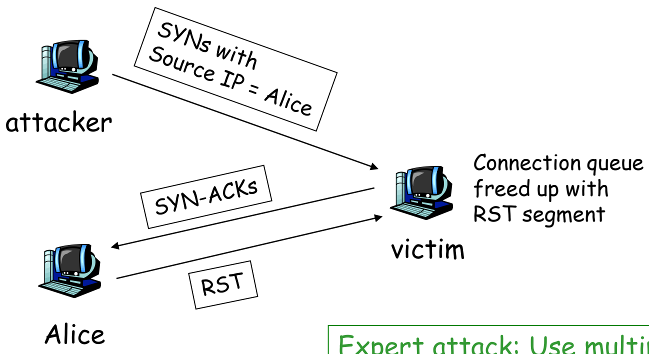


SYN Flooding Explained

- Attacker sends many connection requests (SYNs) with spoofed source addresses
- Victim allocates resources for each request
  - New thread, connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic denial of service attack
  - Common pattern: it costs nothing to TCP client to send a connection request, but TCP server must spawn a thread for each request - asymmetry!
  - What's another example of this behavior?

SYN flood Issue

amateur attack:

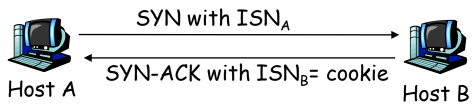


Expert attack: Use multiple source IP addresses, each from unresponsive addresses.

Preventing Denial of Service (SYN Flood)

- DoS is caused by asymmetric state allocation
  - If server opens new state for each connection attempt, attacker can initiate many connections from bogus or forged IP addresses
- Cookies allow server to remain stateless until client produces:
  - Server state (IP addresses and ports) stored in a cookie and originally sent to client
- When client responds, cookie is verified

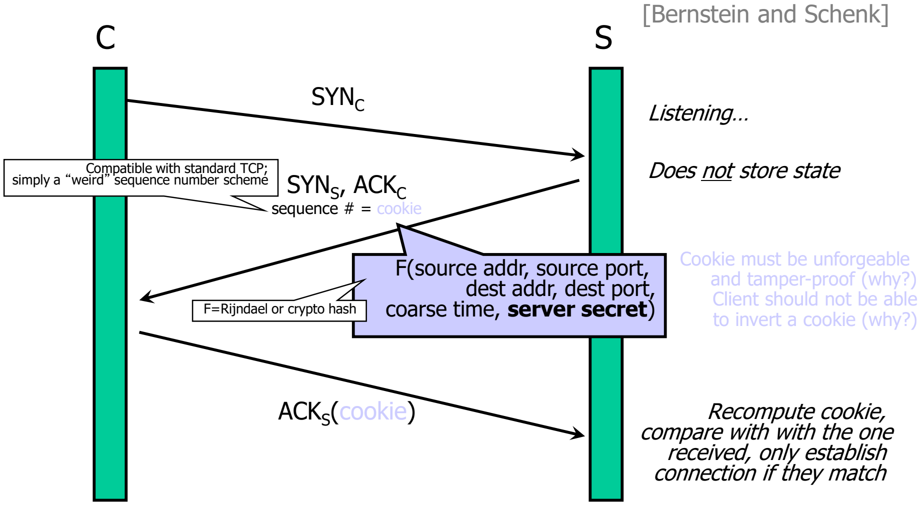
SYN flood defense: SYN cookies (1)



- When SYN segment arrives, host B calculates function (hash) based on:
  - Source and destination IP addresses and port numbers, and a secret number
- Host B uses resulting "cookie" for its initial seq # (ISN) in SYNACK
- Host B does not allocate anything to half-open connection:
  - Does not remember A's ISN
  - Does not remember cookie

SYN Cookies (2)

[Bernstein and Schenk]

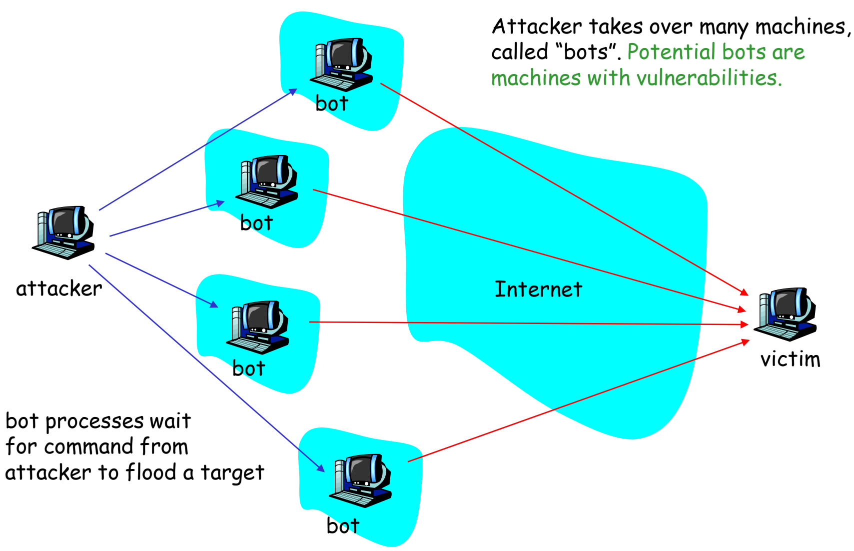


More info: http://cr.yp.to/syncookies.html

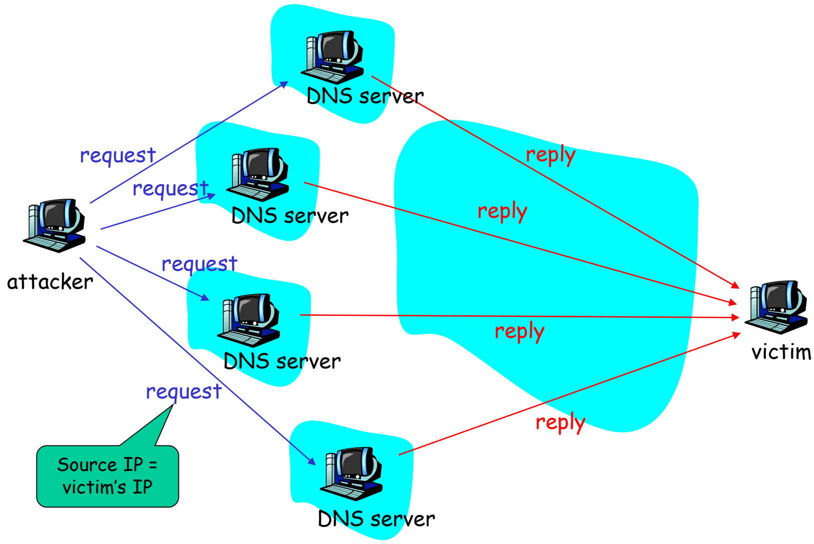
Overwhelming link bandwidth with packets

- Attack traffic can be made similar to legitimate traffic, hindering detection.
- Flow of traffic must consume target's bandwidth resources.
  - Attacker needs to engage more than one machine => DDoS
- May be easier to get target to fill-up its upstream bandwidth: async access
  - Example: attacking BitTorrent seeds

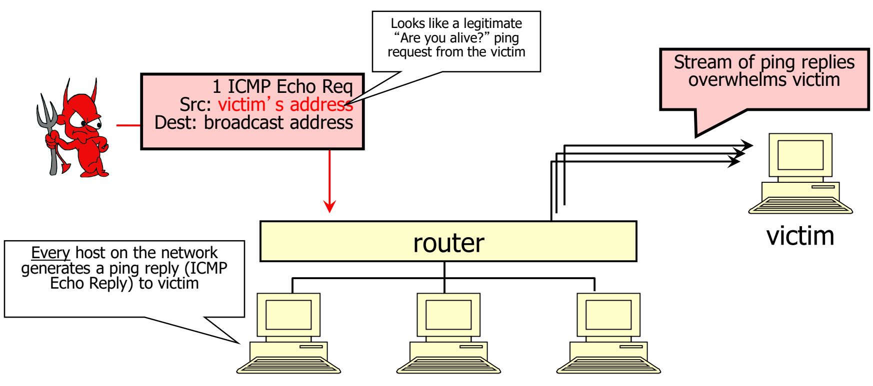
Distributed DoS: DDos



DDoS: Reflection attack



"Smurf" Attack



Solution: reject external packets to broadcast addresses

DDoS Defenses

- Don't let your systems become bots
  - Keep systems patched up
  - Employ egress anti-spoof filtering on external router.
- Filter dangerous packets
  - Vulnerability attacks
  - Intrusion prevention systems
- Signature and anomaly detection and filtering
- Rate limiting
  - Limit # of packets sent from source to dest
- CAPTCHAs
  - Could be useful against application level attacks (e.g., against web servers)

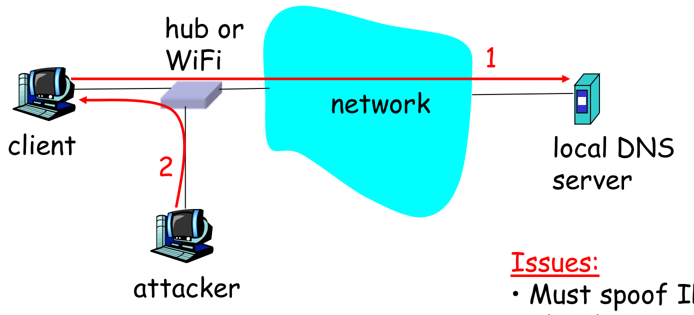
Module 5, Lecture 6

DNS Attacks

DNS attacks

- Reflector attack: already discussed
  - Leverage DNS for attacks on arbitrary targets
- Denying DNS service
  - Stop DNS root servers
  - Stop top-level-domain servers (e.g. .com domain)
  - Stop local (default name servers)
- Use fake DNS replies to redirect user
- Poisoning DNS:
  - Insert false resource records into various DNS caches
  - False records contain IP addresses operated by attackers

DNS attack: redirecting



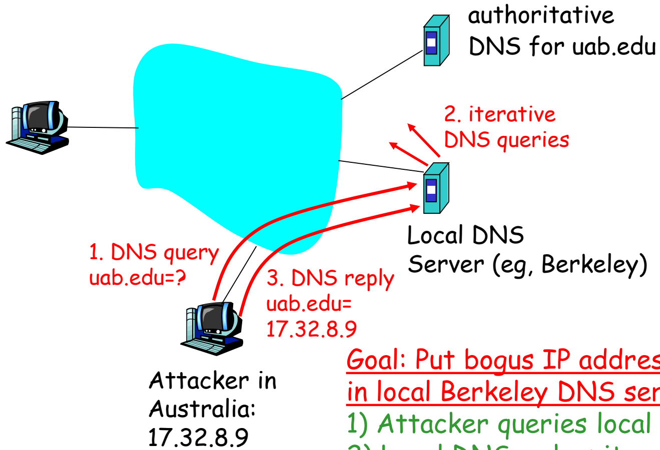
1. Client sends DNS query to its local DNS server; sniffed by attacker
2. Attacker responds with bogus DNS reply

Issues:
- Must spoof IP address: set to local DNS server *(easy)*
- Must match reply ID with request ID *(easy)*
- May need to stop reply from the local DNS server *(harder)*

Poisoning DNS Cache (1)

- Poisoning: Attempt to put bogus records into DNS name server caches
  - Bogus records could point to attacker nodes
  - Attacker nodes could phish
- But unsolicited replies are not accepted at a name server.
  - Name servers use IDs in DNS messages to match replies to queries
  - So can't just insert a record into a name server by sending a DNS reply message.
- But can send a reply to a request.

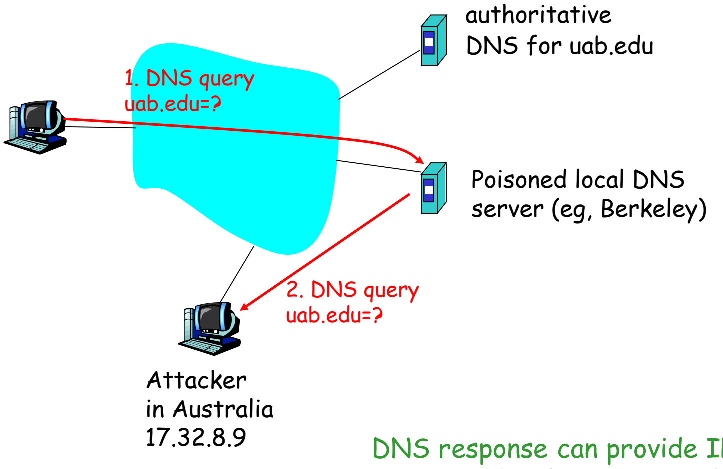
Poisoning local DNS server (2)



Goal: Put bogus IP address for uab.edu in local Berkeley DNS server

1) Attacker queries local DNS server
2) Local DNS makes iterative queries
3) Attacker waits for some time; sends a bogus reply, spoofing authoritative server for uab.edu.

Poisoning local DNS server (3)



DNS response can provide IP address of malicious server!

DNS Poisoning (4)

- Issues:
  - Attacker may need to stop upstream name server from responding
    - So that server under attack doesn't get suspicious
    - Ping of death, DoS, overflows, etc

DNS attacks: Summary

- DNS is a critical component of the Internet infrastructure
- But is surprisingly robust:
  - DDoS attacks against root servers have been largely unsuccessful
  - Poisoning and redirection attacks are difficult unless you can sniff DNS requests
    - And even so, may need to stop DNS servers from replying
- DNS *can be* leveraged for reflection attacks against non-DNS nodes