Computer Networks

Security
Security Vulnerabilities

• At every layer in the protocol stack!

• Network-layer attacks
  – IP-level vulnerabilities
  – Routing attacks

• Transport-layer attacks
  – TCP vulnerabilities

• Application-layer attacks
Security Flaws in IP

• The IP addresses are filled in by the originating host
  – Address spoofing
• Using source address for authentication
  – r-utilities (rlogin, rsh, rhosts etc.)

• Can A claim it is B to the server S?
• ARP Spoofing
• Can C claim it is B to the server S?
• Source Routing
ARP Spoofing

• Attacker uses ARP protocol to associate MAC address of attacker with another host's IP address
• E.g. become the default gateway:
  – Forward packets to real gateway (interception)
  – Alter packets and forward (man-in-the-middle attack)
  – Use non-existent MAC address or just drop packets (denial of service attack)
Source Routing

- ARP spoofing cannot redirect packets to another network
  - if you spoof an IP source address, replies go to the spoofed host
- An option in IP is to provide a route in the packet: *source routing*.
  - Equivalent to tunneling.
- Attack: spoof the host IP address and specify a source route back to the attacker.
Smurf Attack

- Ping request to a broadcast address with source = victim's IP address
- Ping reply from every host
- Replies directed to victim
- Broadcast Enabled Network

Attacking System

Internet

Victim System
ICMP Attacks

• No authentication
• ICMP redirect message
• Oversized ICMP messages can crash hosts
• Destination unreachable
  – Can cause the host to drop connection
• Many more...
ICMP Redirect

- ICMP Redirect message: tell a host to use a different gateway on the same network (saves a hop for future packets)

Spoof an ICMP Redirect message from "Good" Gateway to redirect traffic through Attacker
TCP-level attacks

- **SYN-Floods**
  - Implementations create state at servers before connection is fully established

- **Session hijack**
  - Pretend to be a trusted host
  - Sequence number guessing

- **Session resets**
  - Close a legitimate connection
Session Hijack

First send a legitimate SYN to server
Session Hijack

1. SYN (ISN_X)
   SRC = T
   \[ \rightarrow \]

2. SYN(ISN_S2)
   ACK(ISN_X)

3. ACK(ISN_S2)
   SRC = T
   \[ \rightarrow \]

Using ISN_S1 from earlier connection guess ISN_S2!
TCP Layer Attacks

- TCP SYN Flooding
  - Exploit state allocated at server after initial SYN packet
  - Send a SYN and don’t reply with ACK
  - Server will wait for 511 seconds for ACK
  - Finite queue size for incomplete connections (1024)
  - Once the queue is full it doesn’t accept requests
TCP Layer Attacks

• TCP Session Poisoning
  – Send RST packet
    • Will tear down connection
  – Do you have to guess the exact sequence number?
    • Anywhere in window is fine
    • For 64k window it takes 64k packets to reset
    • About 15 seconds for a T1
Where do the problems come from?

• Protocol-level vulnerabilities
  – Implicit trust assumptions in design

• Implementation vulnerabilities
  – Both on routers and end-hosts

• Incomplete specifications
  – Often left to the imagination of programmers
Denial of Service Attacks
Questions

• What are the DoS attacks at different levels of the network architecture?

• How can we mitigate them?
DoS can happen at any layer

- Sample Dos at different layers (by order):
  - Link
  - TCP/UDP
  - Application
- There are some generic DoS solutions
- However, current Internet not designed to handle DDoS attacks
Internet Reality

• Distributed Denial-of-Service is a huge problem today!
  – Akamai reports DDOS against US banks peaking at 65 Gbps ...

• There are no great solutions
  – CDNs, network traffic filtering, and best practices all help
Examples

• Already discussed:
  – Smurf ICMP amplification attack
  – TCP SYN resource exhaustion attack
DNS Amplification attack: (×50 amplification)

Millions of open resolvers on Internet
A classic SYN flood example

- **MS Blaster worm** (2003)
  - Infected machines at noon on Aug 16\textsuperscript{th}:
    - SYN flood on port 80 to \texttt{windowsupdate.com}
    - 50 SYN packets every second.
      - each packet is 40 bytes.

- **MS solution:**
  - new name: \texttt{windowsupdate.microsoft.com}
  - Win update file delivered by Akamai
Low rate SYN flood defenses

• Non-solution:
  – Increase backlog queue size or decrease timeout

• Correct solution (when under attack):
  – **Syncookies**: remove state from server
  – Small performance overhead
Syncookies

[Bernstein, Schenk]

• Idea: use secret key and data in packet to gen. server SN

• Server responds to Client with SYN-ACK cookie:
  – $T = 5$-bit counter incremented every 64 secs.
  – $L = \text{MAC}_{\text{key}}(\text{SAddr}, \text{SPort}, \text{DAddr}, \text{DPort}, SN_{C}, T)$ [24 bits]
    • key: picked at random during boot
  – $SN_{S} = (T \cdot \text{mss} \cdot L)$ (|L| = 24 bits)
  – **Server does not save state** (other TCP options are lost)

• Honest client responds with ACK ($AN=SN_{S}$, $SN=SN_{C}+1$)
  – Server allocates space for socket only if valid $SN_{S}$. 
DoS Mitigation
Possible defenses I: Filtering

- Filtering at the victim’s firewall
  - Likely to be useless, firewall itself can be targeted

- Filtering at the attacker’s firewall
  - Routers drop packets with an “invalid” source IP address field
  - Would need near universal deployment to be effective
    - Besides, does not prevent subnet-level spoofing
  - Economic incentives?
Ingress/Egress Filtering

• RFC 2827: Routers install filters to drop packets from networks that are not downstream
• Feasible at edges; harder at “core”
Possible defenses II: Pushback

• Pushback: rate limit flows that compose large traffic aggregates to mitigate impact of DDoS
• Assumption: can identify anomalous traffic
• Distributed solution: the whole network benefits

• Requires router modifications
  – Deployment may take very long
  – Need authentication of filters
Possible Defenses III: Traceback

[Savage et al. ’00]

• Goal:
  – Given set of attack packets
  – Determine path to source

• How: change routers to record info in packets

• Assumptions:
  – Most routers remain uncompromised
  – Attacker sends many packets
  – Route from attacker to victim remains relatively stable
Simple method

- Write path into network packet
  - Each router adds its own IP address to packet
  - Victim reads path from packet

Problem:
- Requires space in packet
  - Path can be long
  - No extra fields in current IP format
    - Changes to packet format too much to expect
Better idea

- DDoS involves many packets on same path
- Store one link in each packet
  - Each router probabilistically stores own address
  - Fixed space regardless of path length
Edge Sampling

- Data fields written to packet:
  - Edge: start and end IP addresses
  - Distance: number of hops since edge stored

- Marking procedure for router R
  - if coin turns up heads (with probability p) then
    - write R into start address
    - write 0 into distance field
  - else
    - if distance == 0 write R into end field
    - increment distance field
Edge Sampling: picture

- Packet received
  - $R_1$ receives packet from source or another router
  - Packet contains space for start, end, distance
Edge Sampling: picture

- Begin writing edge
  - $R_1$ chooses to write start of edge
  - Sets distance to 0
Finish writing edge
  - $R_2$ chooses not to overwrite edge
  - Distance is 0
    - Write end of edge, increment distance to 1
Edge Sampling

- Increment distance
  - R₃ chooses not to overwrite edge
  - Distance >0
    - Increment distance to 2
Path reconstruction

• Extract information from attack packets

• Build graph rooted at victim
  – Each (start, end, distance) tuple provides an edge

• # packets needed to reconstruct path

\[
E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}
\]

where p is marking probability, d is length of path
Capability based defense

• Basic idea:
  – Receivers can specify what packets they want

• How:
  – Sender requests capability in SYN packet
    • Path identifier used to limit # reqs from one source
  – Receiver responds with capability
  – Sender includes capability in all future packets

  – **Main point:** Routers only forward:
    • Request packets, and
    • Packets with valid capability
Interdomain Routing Security
Interdomain Routing

- AS-level topology
  - Nodes are Autonomous Systems (ASes)
  - Edges are links and business relationships
TCP Connection Underlying BGP Session

- BGP session runs over TCP
  - TCP connection between neighboring routers
  - BGP messages sent over TCP connection
  - Makes BGP vulnerable to attacks on TCP
Validity of the routing information:
Origin authentication
IP Address Ownership and Hijacking

• IP address block assignment
  – Regional Internet Registries
  – Internet Service Providers

• Proper origination of a prefix into BGP
  – By the AS who owns the prefix or by its upstream provider(s) in its behalf

• However, what’s to stop someone else?
  – Prefix hijacking: another AS originates the prefix
  – BGP does not verify that the AS is authorized
  – Registries of prefix ownership are inaccurate
Prefix Hijacking

- Consequences for the affected ASes
  - Blackhole: data traffic is discarded
  - Snooping: data traffic is inspected, and then redirected
  - Impersonation: data traffic is sent to bogus destinations
Hijacking is Hard to Debug

• The victim AS doesn’t see the problem
  – Picks its own route
  – Might not even learn the bogus route
• May not cause loss of connectivity
  – E.g., if the bogus AS snoops and redirects
  – ... may only cause performance degradation
• Or, loss of connectivity is isolated
  – E.g., only for sources in parts of the Internet
• Diagnosing prefix hijacking
  – Analyzing updates from many vantage points
  – Launching traceroute from many vantage points
Sub-Prefix Hijacking

- Originating a more-specific prefix
  - Every AS picks the bogus route for that prefix
  - Traffic follows the longest matching prefix
How to Hijack a Prefix

• The hijacking AS has
  – Router with BGP session(s)
  – Configured to originate the prefix
• Getting access to the router
  – Network operator makes configuration mistake
  – Disgruntled operator launches an attack
  – Outsider breaks in to the router and reconfigures
• Getting other ASes to believe bogus route
  – Neighbor ASes do not discard the bogus route
  – E.g., not doing protective filtering
YouTube Outage on Feb 24, 2008

- **YouTube (AS 36561)**
  - Web site www.youtube.com
  - Address block 208.65.152.0/22
- **Pakistan Telecom (AS 17557)**
  - Receives government order to block access to YouTube
  - Starts announcing 208.65.153.0/24 to PCCW (AS 3491)
  - All packets directed to YouTube get dropped on the floor
- **Mistakes were made**
  - AS 17557: announcing to everyone, not just customers
  - AS 3491: not filtering routes announced by AS 17557
- **Lasted 100 minutes for some, 2 hours for others**
Timeline (UTC Time)

- 18:47:45
  - First evidence of hijacked /24 route propagating in Asia
- 18:48:00
  - Several big trans-Pacific providers carrying the route
- 18:49:30
  - Bogus route fully propagated
- 20:07:25
  - YouTube starts advertising the /24 to attract traffic back
- 20:08:30
  - Many (but not all) providers are using the valid route

http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml
Timeline (UTC Time)

- 20:18:43
  - YouTube starts announcing two more-specific /25 routes
- 20:19:37
  - Some more providers start using the /25 routes
- 20:50:59
  - AS 17557 starts prepending (“3491 17557 17557”)
- 20:59:39
  - AS 3491 disconnects AS 17557
- 21:00:00
  - All is well, videos of cats flushing toilets are available

http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml
Another Example: Spammers

• Spammers sending spam
  – Form a (bidirectional) TCP connection to a mail server
  – Send a bunch of spam e-mail
• But, best not to use your real IP address
  – Relatively easy to trace back to you
• Could hijack someone’s address space
  – But you might not receive all the (TCP) return traffic
  – And the legitimate owner of the address might notice
• How to evade detection
  – Hijack unused (i.e., unallocated) address block in BGP
  – Temporarily use the IP addresses to send your spam
Question

• What other attacks are possible with BGP?
Bogus AS Paths

• Remove ASes from the AS path
  – E.g., turn “701 3715 88” into “701 88”

• Motivations
  – Make the AS path look shorter than it is
  – Attract sources that normally try to avoid AS 3715
  – Help AS 88 look like it is closer to the Internet’s core

• Who can tell that this AS path is a lie?
  – Maybe AS 88 *does* connect to AS 701 directly
Bogus AS Paths

- Add ASes to the path
  - E.g., turn “701 88” into “701 3715 88”

- Motivations
  - Trigger loop detection in AS 3715
    - Denial-of-service attack on AS 3715
    - Or, blocking unwanted traffic coming from AS 3715!
  - Make your AS look like it has richer connectivity

- Who can tell the AS path is a lie?
  - AS 3715 could, if it could see the route
  - AS 88 could, but would it really care as long as it received data traffic meant for it?
Bogus AS Paths

• Adds AS hop(s) at the end of the path
  – E.g., turns “701 88” into “701 88 3”

• Motivations
  – Evade detection for a bogus route
  – E.g., by adding the legitimate AS to the end

• Hard to tell that the AS path is bogus...
  – Even if other ASes filter based on prefix ownership
Invalid Paths

• AS exports a route it shouldn’t
  – AS path is a valid sequence, but violated policy
• Example: customer misconfiguration
  – Exports routes from one provider to another
• Interacts with provider policy
  – Provider prefers customer routes
  – Directing all Internet traffic through customer
• Main defense
  – Filtering routes based on prefixes and AS path
Missing/Inconsistent Routes

• Peers require consistent export
  – Prefix advertised at all peering points
  – Prefix advertised with same AS path length
• Reasons for violating the policy
  – Trick neighbor into “cold potato”
  – Configuration mistake
• Main defense
  – Analyzing BGP updates or data traffic
BGP Security Today

• Applying best common practices
  – Securing the session (authentication, encryption)
  – Filtering routes by prefix and AS path
  – Packet filters to block unexpected control traffic

• This is not good enough
  – Doesn’t address fundamental problems
    • Can’t tell who owns the IP address block
    • Can’t tell if the AS path is bogus or invalid
    • Can’t be sure the data packets follow the chosen route
Proposed Enhancements to BGP
S-BGP Secure Version of BGP

- **Address attestations**
  - Claim the right to originate a prefix
  - Signed and distributed out-of-band
  - Checked through delegation chain from ICANN

- **Route attestations**
  - Distributed as an attribute in BGP update message
  - Signed by each AS as route traverses the network
  - Signature signs previously attached signatures

- **S-BGP can validate**
  - AS path indicates the order ASes were traversed
  - No intermediate ASes were added or removed
S-BGP Deployment Challenges

- Complete, accurate registries
  - E.g., of prefix ownership
- Public Key Infrastructure
  - To know the public key for any given AS
- Cryptographic operations
  - E.g., digital signatures on BGP messages
- Need to perform operations quickly
  - To avoid delaying response to routing changes
- Difficulty of incremental deployment
  - Hard to have a “flag day” to deploy S-BGP
Incrementally Deployable Solutions?

- Backwards compatible
  - No changes to router hardware or software
  - No cooperation from other ASes
- Incentives for early adopters
  - Security benefits for ASes that deploy the solution
  - ... and further incentives for others to deploy

- What kind of solutions are possible?
  - Detecting suspicious routes and then filtering or depreferencing them
Detecting Suspicious Routes

- Monitoring BGP update messages
  - Use past history as an implicit registry
- E.g., AS that announces each address block
  - Prefix 18.0.0.0/8 usually originated by AS 3
- E.g., AS-level edges and paths
  - Never seen the subpath “7018 88 1785”
- Out-of-band detection mechanism
  - Generate reports and alerts
  - Prefix Hijack Alert System: http://phas.netsec.colostate.edu/
Avoiding Suspicious Routes

• Soft response to suspicious routes
  – Prefer routes that agree with the past
  – Delay adoption of unfamiliar routes when possible

• Why is this good enough?
  – Some attacks will go away on their own
  – Give network operators time to investigate

• How well would it work?
  – If top ~40 largest ASes applied the technique
  – ... most other ASes are protected, too
  – ... since they mostly learn routes from the big ASes
Conclusions

• Border Gateway Protocol is very vulnerable
  – Glue that holds the Internet together
  – Hard for an AS to locally identify bogus routes
  – Attacks can have very serious global consequences

• Proposed solutions/approaches
  – Secure variants of the Border Gateway Protocol
  – Anomaly detection schemes, with automated response