I DISCOVERED A HOLE IN OUR INTERNET SECURITY.

GOOD GRIEF, MAN! HOW COULD YOU PUT A HOLE IN OUR INTERNET?

WHAT?!!!

I DIDN'T PUT IT THERE. I FOUND IT... AND IT'S NOT...

IT'S YOUR JOB TO FIX THAT HOLE. I WANT YOU TO WORK 24-7!

ACTUALLY, THAT'S NOT MY JOB. BUT I'LL INFORM OUR NETWORK MANAGEMENT GROUP.

PASSING THE BUCK!!! YOU'RE A BUCK PASSER!!!
CSE 534
Network Security

Vyas Sekar
Stony Brook University

With content from: Srini Seshan
Assigned reading

• TVA
  – Yang et al
  – Problem?
  – Solution?

• Traceback
  – Savage et al
  – Problem?
  – Solution?
Likes/Dislikes

• Like
  – Security is an important problem
  – Architectural solutions rather than patches

• Dislike
  – Too fanciful not really deployed
  – What’s the real threat?
  – Implementation/Eval depth is pretty low
Outline

• Basic overview of network security

• DDoS Overview

• Traceback

• Capabilities/TVA
1. Interconnection
2. Failure resilience
3. Types of service
4. Variety of networks

Network designed with implicit trust
→ Not designed for “bad” players
The World has Changed

• Growth
  – Beyond just university experiments

• Use-cases
  – Critical services (e.g., banking)
  – E-commerce

• Attackers!
Threats on the Internet

- Eavesdrop
- Impersonate
- DDoS
- Modify
- Exfiltrate
- Traffic hijack
- Infect

ISP A
ISP B
ISP C
ISP D
What do attackers do?
Vulnerabilities at every layer

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

• Transport-layer attacks
  – TCP vulnerabilities

• Application-layer attacks

• End-to-end attacks
Worms

• Self-propagate through network
  – Without user intervention

• Typical steps in worm propagation
  – Probe host for vulnerable software
  – Exploit the vulnerability (e.g., buffer overflow)
    • Attacker gains privileges of the vulnerable program
  – Launch copy on compromised host

• Spread at exponential rate
  – 10M hosts in < 5 minutes
  – Hard to deal with manual intervention
Defense against the Dark Arts?

Security mechanisms “overlaid” over existing untrusted Internet
End-to-End
Properties of a secure communication channel

• Authentication (Who am I talking to?)

• Confidentiality (Is my data hidden?)

• Integrity (Has my data been modified?)
Firewalls

• Block/filter/modify traffic at network-level
  – Limit access to the network
  – Installed at perimeter of the network
• Allows traffic specified in the policy
• Drops everything else
Intrusion Detection Systems

- Firewalls allow traffic only to legitimate hosts and services
- Traffic to the legitimate hosts/services can have attacks
- Solution?
  - Intrusion Detection Systems
  - Monitor data and behavior
  - Report when identify attacks
Miscellany

- TCP attacks
- Spam
- Phishing
- Censorship-resistance
- Anonymous communication
- Route hijacks
- Any other netsec topic you want to know about?
Outline

• Basic overview of network security

• DDoS Overview

• Traceback

•Capabilities/TVA
Denial of Service

• Make a service unusable/unavailable

• Disrupt service by taking down hosts
  – E.g., ping-of-death

• Consume host-level resources
  – E.g., SYN-floods

• Consume network resources
  – E.g., UDP/ICMP floods
Reflector Attacks

• Spoof source address
• Send query to service
• Response goes to victim
• If response >> query, “amplifies” attack
• Hides real attack source from victim
• Amplifiers:
  – DNS responses (50 byte query → 400 byte resp)?
  – ICMP to broadcast addr (1 pkt → 50 pkts) (“smurf”)

• Recent spamhaus attack!
Reflector Attack

Unsolicited traffic at victim from legitimate hosts
Smurf Attack

Attacking System

Victim System

Internet

Broadcast Enabled Network

15-411: security
Distributed DoS

Attacker

Handler

Agent

Handler

Agent

Agent

Agent

Agent

Victim

15-411: security
Inferring DoS Activity: Backscatter

IP address spoofing creates random backscatter.
Backscatter Analysis

• Use a big block of addresses (N of them)?
  – People often use a /16 or /8

• Observe x backscatter packets/sec
  – How big is actual attack?
    • x * (2^32 / N)?
    • Assuming uniform distribution

• Sometimes called “network telescope”
Bandwidth DOS Attacks - Solutions

- Ingress filtering – examine packets to identify bogus source addresses
- Link testing – have routers either explicitly identify which hops are involved in attack or use controlled flooding and a network map to perturb attack traffic
- Logging – log packets at key routers and post-process to identify attacker’s path
- ICMP traceback – sample occasional packets and copy path info into special ICMP messages
- Capabilities
- IP traceback + filtering
Cool Stuff 😊


Outline

- Basic overview of network security
- DDoS Overview
- Traceback
- Capabilities/TVA
Security Flaws in IP

• The IP addresses are filled in by the originating host
  – Address spoofing

• Can A claim it is B to the server S?
• Can C claim it is B to the server S?
The Need for Traceback

• Internet hosts are vulnerable
  – Many attacks consist of *very few packets*
  – Fraggle, Teardrop, ping-of-death, etc.

• Internet Protocol permits anonymity
  – Attackers can “spoof” source address
  – IP forwarding maintains no audit trails

• Need a separate *traceback* facility
  – For a given packet, find the path to *source*
Approaches to Traceback

- Path data can be noted in several places
  - In the packet itself [Savage et al.],
  - At the destination [I-Trace], or
  - In the network infrastructure

- Logging: a naïve in-network approach
  - Record each packet forwarding event
  - Can trace a single packet to a source router, ingress point, or subverted router(s)
Marking procedure at router $R$:
for each packet $w$, append $R$ to $w$

Path reconstruction procedure at victim $v$:
for any packet $w$ from attacker
extract path $(R_i \ldots R_j)$ from the suffix of $w$

High overhead at router + Too much space
Extension: Node Sampling

Marking procedure at router \( R \):
for each packet \( w \)
  let \( x \) be a random number from \([0..1)\)
  if \( x < p \) then,
    write \( R \) into \( w \cdot \text{node} \)

Path reconstruction procedure at victim \( v \):
let \( \text{NodeTbl} \) be a table of tuples \((\text{node, count})\)
for each packet \( w \) from attacker
  \( z := \text{lookup } w \cdot \text{node in } \text{NodeTbl} \)
  if \( z \neq \text{NIL} \) then
    increment \( z \cdot \text{count} \)
  else
    insert tuple \((w \cdot \text{node}, 1)\) in \( \text{NodeTbl} \)
  sort \( \text{NodeTbl} \) by count
  extract path \((R_i \ldots R_j)\) from ordered node fields in \( \text{NodeTbl} \)
IP Traceback

• Node append (record route) – high computation and space overhead
• Node sampling – each router marks its IP address with some probability $p$
  – $P(\text{receiving mark from router } d \text{ hops away}) = p(1 - p)^{d-1}$
  – $p > 0.5$ prevents any attacker from inserting false router
  – Must infer distance by marking rate $\rightarrow$ relatively slow
  – Doesn’t work well with multiple routers at same distance $\rightarrow$ i.e. multiple attackers
IP Traceback

• Edge sampling
  – Solve node sampling problems by encoding edges & distance from victim in messages
  – Start router sets “start” field with probability $p$ and sets distance to 0
  – If distance is 0, router sets “end” field
  – All routers increment distance
  – As before, $P(\text{receiving mark from router d hops away}) = p(1 - p)^{d-1}$

• Multiple attackers can be identified since edge identifies splits in reverse path
Edge Sampling

• Major problem – need to add about 72bits (2 address + hop count) of info into packets

• Solution
  – Encode edge as xor of nodes $\rightarrow$ reduce 64 bits to 32 bits
  – Ship only 8bits at a time and 3bits to indicate offset $\rightarrow$ 32 bits to 11bits
  – Use only 5 bit for distance $\rightarrow$ 8bits to 5bits
  – Use IP fragment field to store 16 bits
    • Some backward compatibility issues
    • Fragmentation is rare so not a big problem
Outline

• Basic overview of network security threats

• DDoS Overview

• Traceback

• Capabilities/TVA
Capabilities

• Filters: prevent the bad stuff
• Capabilities: must have permission to talk
• Sender must first ask dst for permission
  – If OK, dst gives capability to src
  – capability proves to routers that traffic is OK
• Good feature: stateless at routers
Unforgeable Capabilities

• It is required that a set of capabilities be not easily forgeable or usable if stolen from another party

• Each router computes a cryptographic hash when it forwards a request packet

• The destination receives a list of pre-capabilities with fixed source and destination IP, hence preventing spoofed attacks
TVA (Capability)

PreCapability (Pi) =
hash(srcIP, destIP, time, secret)

- RTS rate limited
  - 1-5% of bandwidth
- Pi Queue at Router
  - Most recent Pi
Fine-Grained Capabilities

- False authorizations even in small number can cause a denial of service until the capability expires
- An improved mechanism would be for the destination to decide the amount of data (N) and also the time (T) along with the list of pre-capabilities
TVA (Capability)

Capability =
timestamp || Hash (N, T, PreCap)

• *N bytes, T seconds*
• Stateless receiver
  – Does not store *N, T*
Bounded Router State

• The router state could be exhausted as it would be counting the number of bytes sent

• Router state is only maintained for flows that send faster than N/T
  – When new packets arrive, new state is created and a byte counter is initialized along with a time-to-live field that is decremented/incremented
Balancing Authorized Traffic

• It is quite possible for a compromised insider to allow packet floods from outside

• A fair-queuing policy is implemented and the bandwidth is decreased as the network becomes busier

• To limit the number of queues, a bounded policy is used which only queues those flows that send faster than $\frac{N}{T}$

• Other senders are limited by FIFO service
Basic Idea for Queue Management

- Requests
- Path-identifier queue
- Regular packets
- Capability checking
- Yes
- Per-destination queue
- No
- Legacy packets
- Low priority queue
Short, Slow or Asymmetric Flows

• Even for short or slow connections, since most byte belong to long flows the aggregate efficiency is not affected

• No added latency are involved in exchanging handshakes

• All connections between a pair of hosts can use single capability

• TVA experiences reduced efficiency only when all the flows near the host are short; this can be countered by increasing the bandwidth
Takeaways

• Internet security is hard
  – Often retrofitted, not intrinsic
  – Worms, DDoS, Spoofing, Hijack, Eavesdrop ..

• Papers on DoS/DDoS
  – Detecting spoofed origins -- Traceback
  – Preventing bandwidth floods – TVA

• Other solutions:
  CDN, caching, new internet arch