Distributed Denial of Service

John Mitchell

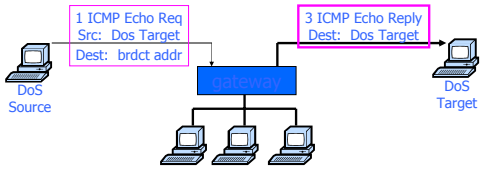
Outline

- Point-to-point network denial of service
  - Smurf, TCP syn flooding, TCP reset
  - Congestion control attack
- Distributed denial of service attacks
  - Coordinated attacks
  - Trin00, TFN, Stacheldraht, TFN2K
- IP traceback
  - Edge Sampling techniques
- Commercial products
  - Monitor and filter traffic

Sources

- Analysis of a Denial of Service Attack on TCP
  - Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spafford, Aurobindo Sundaram, Diego Zamboni, Security & Privacy 1997
- Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants)
  - Aleksandar Kuzmanovic and Edward W. Knightly, SIGCOM 2003
- Practical Network Support for IP Traceback
  - Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson. SIGCOMM 2000
- Advanced and Authenticated Marking Schemes for IP Traceback
  - Dawn X. Song, Adrian Perrig. Proceedings IEEE Infocomm 2001

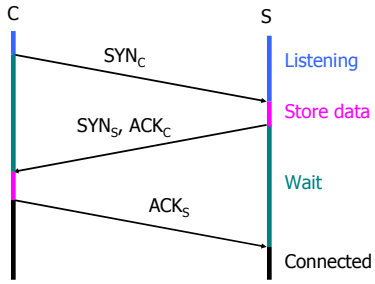
Smurf DoS Attack



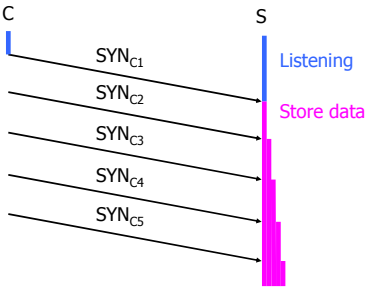
- Send ping request to brdcst addr (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim
  - Ping reply stream can overload victim

Prevention: reject external packets to brdcst address.

TCP Handshake



SYN Flooding



TCP Reset vulnerability [Watson'04]

- Attacker sends RST packet to reset connection
  - Need to guess seq. # for an existing connection
    - Naively, success prob. is $1/2^{32}$ for 32-bit seq. number
    - Most systems allow for a large window of acceptable seq. #'s ⇒ much higher success probability

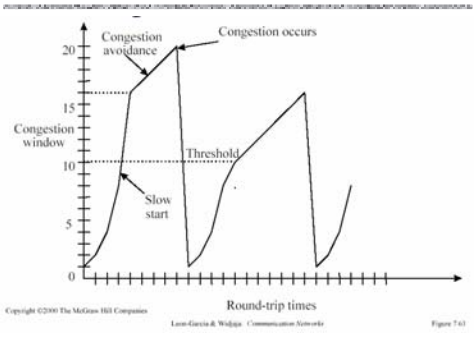
Attack is most effective against long lived connections, e.g. BGP

Block with stateful packet filtering?

TCP Congestion Control

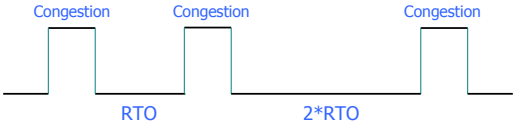
- Source estimates available bandwidth
  - Starts slow and increases based on ACKS
  - Reduces rate if congestion
- Two time scales
  - If RTT is 10-100 ms, TCP performs AIMD
    - Additive Increase Multiplicative Decrease
    - Drops quickly (by half), rises slowly
  - Severe congestion ⇒ Retransmission Timeout (RTO)
    - Send one packet and wait for period RTO
    - If further loss
      - RTO = 2*RTO
    - If packet successfully received, TCP enters slow start
    - Minimum value for RTO is 1 sec

Pattern



Congestion control attack

- Generate TCP flow to force target to repeatedly enter retransmission timeout state



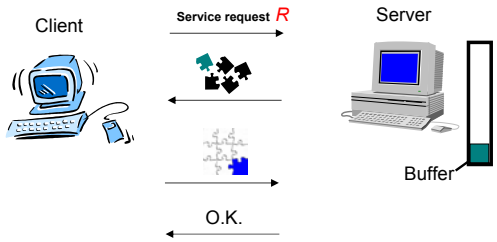
- Difficult to detect because packet rate is low
  - Degrade throughput significantly
  - Existing solutions only mitigate the attack

Using puzzles to prevent DOS

- Basic idea
  - Sender must solve a puzzle before sending
  - Takes some effort to solve, but easy to confirm solution (e.g., hash collision)
- Example use (RSA client puzzle protocol)
  - Normally, server accepts any connection request
  - If attack detected, server responds to request with puzzle
  - Supports connection only to clients that solve puzzle within some regular TCP timeout period

http://www.rsasecurity.com/rsalabs/node.asp?id=2050

The client puzzle protocol



http://www.rsasecurity.com/rsalabs/node.asp?id=2050

DDOS: How disaster strikes …

To: nanog@merit.edu
Subject: Yahoo network outage
From: Declan McCullagh <declan@wired.com>
Date: Mon, 07 Feb 2000 16:22:41 -0500
Delivered-To: nanog-outgoing@merit.edu
Sender: owner-nanog@merit.edu

… I was wondering whether anyone has some insight into what happened with Yahoo. The main site (although not all properties) has been offline since 10:30 am pt Monday. It doesn't *appear* to be Global Crossing's problem, though I can't be sure. GC is mum on the phone.
-Declan

To: Declan McCullagh <declan@wired.com>
Subject: Re: Yahoo network outage
From: Richard Irving <rirving@onecall.net>
Date: Mon, 07 Feb 2000 16:34:44 -0500

To Quote my Noc:
I just got off the phone with Global Center NOC. Global Center Sunnyvale Router is down. Both Yahoo! and Global Center are working on the problem at this time. No ETA for repair

To: nanog@merit.edu
Subject: Re: Yahoo network outage
From: Kai Schlichting <kai@pac-rim.net>
Date: Mon, 07 Feb 2000 16:37:10 -0500
Delivered-To: nanog-outgoing@merit.edu

Yahoo seems to be down by itself, but GC (The former Exodus?) was majorly hosed for a couple of hours today, at least when seen from UUnet. This has cleared up since. The way it looked, they must have lost a larger circuit and traffic was falling back onto something smaller. I certainly heard about it from customers today.

To: <nanog@merit.edu>
Subject: Yahoo offline because of attack (was: Yahoo network outage)
From: Declan McCullagh <declan@wired.com>
Date: Mon, 07 Feb 2000 20:31:24 -0500

Yahoo told me on the phone that it's a malicious attack, and Global Center says the same thing. In Yahoo's words: "a coordinated distributed denial of service attack." We've got a brief story up at: http://www.wired.com/news/business/0,1367,34178,00.html The problem apparently originated with a router. But what kind of attack could have taken the network offline for that period of time and not affected other Global Center customers? I mean, there had to have been a gaping security hole somewhere: It looks like the routes got lost for (nearly) all of the Yahoo network, but no other non-Yahoo sites...
-Declan

Routers Blamed for Yahoo Outage by Declan McCullagh and Joan

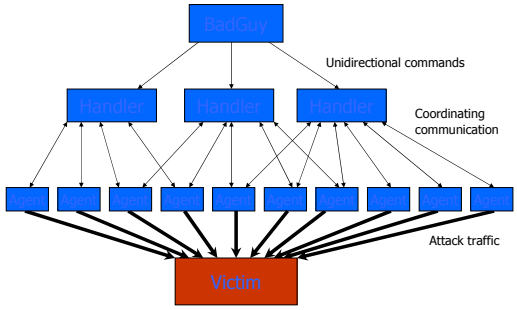
Wired news article

- Most of Yahoo unreachable for three hours
- Attackers reportedly laid siege…
  - Denying millions of visitors access …
- An engineer at another company …
  - told Wired News outage due to misconfigured equipment
- Details remained sketchy…
- A Yahoo spokesperson called it a "coordinated distributed denial of service attack"

What happened?

- Coordinated effort from many sites
- Attacking sites were compromised
  - According to Dittrich's DDoS analysis,
    - trinoo and tfn daemons found on of Solaris 2.x systems
    - systems compromised by exploitation of buffer overrun in the RPC services statd, cmsd and ttdbserverd
- Compromised machines used to mount attack

DDOS



Trin00

- Client to Handler to Agent to Victim
  - Multi-master support
  - Attacks through UDP flood
- Restarts agents periodically
- Warns of additional connects
- Passwords protect handlers and agents of Trin00 network, though sent in clear text

Attack using Trin00

- In August 1999, network of > 2,200 systems took University of Minessota offline for 3 days
  - Tools found cached at Canadian firm
  - Steps:
    - scan for known vulnerabilities, then attack
    - once host compromised, script the installation of the DDoS master agents
- According to the incident report
  - Took about 3 seconds to get root access
  - In 4 hours, set up > 2,200 agents

Tribal Flood Network (TFN)

- Client to Daemon to Victim
- TCP, SYN and UDP floods
- No passwords for client
- Client-Daemon communication only in ICMP
- Needs root access
- Fixed payload size
- Does not authenticate incoming ICMP

Stacheldraht

- Combines Trin00 and TFN features
- Communication is symmetric key encrypted
- Able to upgrade agents on demand
- Client to Handler to Agent to Victim topology, just like Trin00
- Authenticates communication

Traffic Characteristics

- Trinoo
  - Port 1524 tcp Port 27665 tcp
  - Port 27444 udp Port 31335 udp
- TFN
  - ICMP ECHO and ICMP ECHO REPLY packets.
- Stacheldraht
  - Port 16660 tcp Port 65000 tcp
  - ICMP ECHO and ICMP ECHO REPLY
- TFN2K
  - Ports supplied at run time or chosen randomly
  - Combination of UDP, ICMP and TCP packets.

Possible firewall actions

- Only allow packets from known hosts
- Check for reverse path
  - Block packets from IP addr X at the firewall if there is no reverse connection going out to addr X
- Ingress/egress filtering
  - Packets in must have outside source destination
  - Packets out must have inside source destination
- Rate limiting
  - Limit rate of ICMP packets and/or SYN packets

All of these steps may interfere with legitimate traffic

Can you find source of attack?

- Hard to find BadGuy
  - Originator of attack compromised the handlers
  - Originator not active when DDOS attack occurs
- Can try to find agents
  - Source IP address in packets is not reliable
  - Need to examine traffic at many points, modify traffic, or modify routers

Methods for finding agents

- Manual methods using current IP routing
  - Link testing
  - Input debugging
  - Controlled flooding
  - Logging
- Changing router software
  - Instrument routers to store path
  - Provides automated IP traceback

Link Testing

- Start from victim and test upstream links
- Recursively repeat until source is located
- Assume attack remains active until trace complete

Input Debugging

- Victim recognizes attack signature
- Install filter on upstream router
- Pros
  - May use software to help coordinate
- Cons
  - Require cooperation between ISPs
  - Considerable management overhead

Controlled Flooding

- Flooding link during attack
  - Add large bursts of traffic
  - Observe change in packet rate at victim
- Pros
  - Eventually works if attack continues
- Cons
  - Add denial of service to denial of service

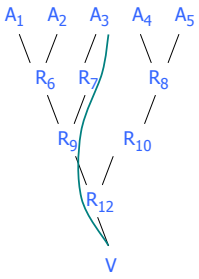
Logging

- Key routers log packets
- Use data mining to find path
- Pros
  - Post mortem – works after attack stops
- Cons
  - High resource demand

Modify routers to allow IP traceback

Traceback problem

- Goal
  - Given set of packets
  - Determine path
- Assumptions
  - Most routers remain uncompromised
  - Attacker sends many packets
  - Route from attacker to victim remains relatively stable

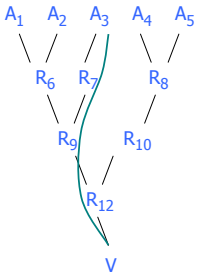


Simple method

- Record path
  - Each router adds 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 are not practical

Better idea

- Many packets
  - 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



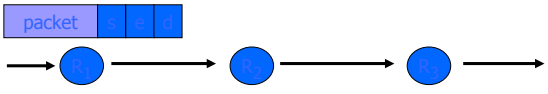
Edge Sampling

- Data fields
  - Edge: *start* and *end* IP addresses
  - Distance: number of hops since edge stored
- Marking procedure for router R

with probability p
write R into start address
write 0 into distance field
else
if distance ==0 write R into end field
increment distance field

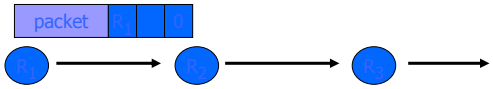
Edge Sampling: picture

- Packet received
  - R_1 receives packet from source or another router
  - Packet contains space for start, end, distance



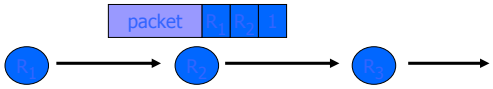
Edge Sampling: picture

- Begin writing edge
  - R_1 chooses to write start of edge
  - Sets distance to 0



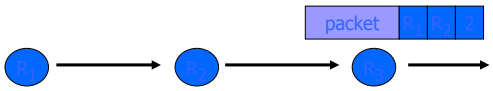
Edge Sampling

- 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
  - R_3 chooses not to overwrite edge
  - Distance >0
    - Increment distance to 2



Path reconstruction

- Extract identifiers from attack packets
- Build graph rooted at victim
  - Each (start,end,distance) tuple provides an edge
  - Eliminate edges with inconsistent distance
  - Traverse edges from root to find attack paths
- # packets needed to reconstruct path

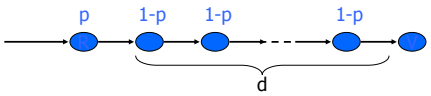
$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

where p is marking probability, d is length of path

Optimal p is 1/d … can vary probability by distance

Node Sampling?

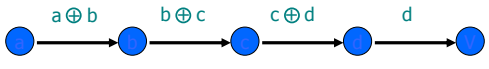
- Less data than edge sampling
  - Each router writes own address with probability p
- Infer order by number of packets
  - Router at distance d has probability $p(1-p)^d$ of showing up in marked packet



- Problems
  - Need many packets to infer path order
  - Does not work well if many paths

Reduce Space Requirement

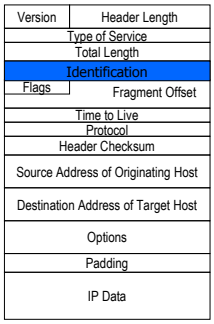
- XOR edge IP addresses
  - Store edge as start ⊕ end
  - Work backwards to get path:
    (start ⊕ end) ⊕ end = start
- Sample attack path



Details: where to store edge

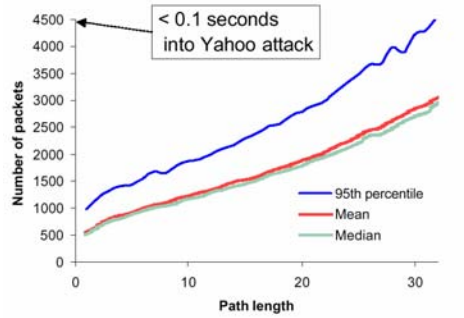
- Identification field
  - Used for fragmentation
  - Fragmentation is rare
  - 16 bits
- Store edge in 16 bits?

- Break into chunks
- Store start ⊕ end



[Savage et al]

Experimental convergence time



Summary of Edge Sampling

- Benefits
  - Practical algorithm for tracing anonymous attacks
  - Can reduce per-packet space overhead (at a cost)
  - Potential encoding into current IP packet header
- Weaknesses
  - Path validation/authentication
  - Robustness in highly distributed attacks
    - Both addressed nicely in [Song&Perrig00]
  - Compatibility issues (IPsec AH, IPv6)
  - Origin laundering (reflectors, tunnels, etc)

Song and Perrig

Advanced Marking Schemes

- Assumption
  - Map of upstream routers is known (www.caida.org)
- Encoding
  - 11 bit for the XOR of hashes of the IP addresses
  - 5 bits for the distance
- Improvement
  - use two *sets* of independent hash functions to minimize collision

Marking and detection

- Marking procedure for router R

  with probability p

  write h(R) into address field

  write 0 into distance field

  else

  if distance ==0 set field = field ⊕ h'(R)

  increment distance field
- Reconstruction
  - Use upstream router map
  - Guess last router, confirm by computing hash
  - Otherwise, same as

Authenticated Marking Schemes

- Packets not authenticated
  - Attacker can forge markings and mislead victim
- Possible solutions
  - Digital signatures: too expensive
  - Use message authentication codes (MACs)
    - Each router shares secret keys with the victim
    - Key management complex; Scheme impractical
  - Use time-released keys
    - Each router has sequence of keys
    - Publishes first key in digital certificate
    - Changes key periodically

Similar to S/Key passwords…

Time-Release Keys

- Router creates chain of keys K_0, K_1, … ,K_{N-1}
  - Selects a random key K_N
  - Using hash function, let $K_j = \text{hash}(K_{j+1})$
- Router publishes K_0 in public certificate
- Properties
  - Given K_j, cannot predict K_i for i>j
  - Given K_j, can compute K_0 and check
- Keys will be used in order K_1, K_2, …

Network World, 09/24/01

Commercial DDOS products

- Within the last three months, four companies
  - Arbor Networks
  - Asta Networks
  - Captus Networks
  - Mazu Networks

  have started shipping equipment to fight DDoS attacks
- The companies claim to be able to identify such attacks and help users take steps to stop them

Arbor Peakflow

- Anomaly detection
  - Create model of normal network behavior, compare traffic to perform anomaly detection
- Measure
  - Sample network traffic statistics and routing data across network, aggregating and correlating this information to create a distributed, network-wide perspective
  - Leveraging existing router interfaces to measure at high speed
- Visualize
  - builds graphs that display traffic and routing information
  - Organized by AS, router, interface, and protocol.
- Protect
  - flags misuse, protocol anomalies, usage or bandwidth anomalies.
  - Detects known threats (ICMP flood) and new or zero-day threats
  - fuel network troubleshooting

Mazu

- Mazu *Profiler*
  - Real-time Event Detection Module
    - observes changes in network behavior rather than comparing traffic against known signatures
  - Real-time Analysis Module
    - analytic and reporting capabilities
- Mazu *Enforcer* – Filtering capability
  - Mazu filters
    - E.g, TCP SYN Flood Filters
      - Tracks unacknowledged SYN packets entering the network
      - Sends resets to free up the downstream resources
    - Fast filters can eliminate specific packets once operator acts
  - Cisco router ACL filters

	High performance Mazu filters	Mazu expression filters	Mazu TCP SYN flood filters	Mazu Payload Filters	Cisco router ACL filters
Filters on addresses	✓	✓			✓
Filters on ports	✓	✓			✓
Filters on protocol	✓	✓			✓
Filters on TTL	✓	✓			
Filters on packet length	✓	✓			
Filters on ANY packet header field		✓			
Filters on payload hash	✓				
Filters on exact payload fragment				✓	
Proactively mitigates SYN floods			✓		
Supports tens of thousands of terms with no performance impact	✓				
Enforcer generates a recommendation	✓				✓
Can be used in passive mode					✓

Payload filter matches bit pattern in packet payload, can detect worms and viruses that have a static bit pattern as a subset of some other, more random packet payload (e.g., Code Red)