Network Denial of Service

John Mitchell

Course logistics

- Four more lectures
  - Today: Network denial of service
  - Tues: Firewalls, intrusion detection, traffic shapers
  - Thurs: Network security protocols
  - May 31: Paul Kocher, *Guest speaker*
- Project: due June 2
- Homework: due June 2
- Final exam: June 6

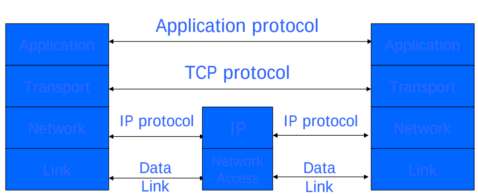
Outline

- 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
  - Bot networks
- Mitigation techniques
  - Firewall
  - IP traceback
    - Edge Sampling techniques
  - Overlay networks
    - Migration
    - Authentication

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
- MOVE: An End-to-End Solution To Network Denial of Service
  - A. Stavrou, A.D. Keromytis, J. Nieh, V.Misra, and D. Rubenstein

TCP Protocol Stack

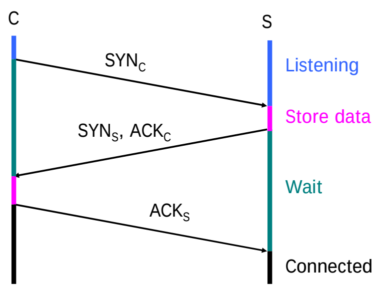


This lecture is about attacks on transport layer and below

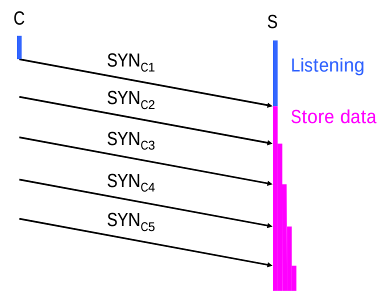
Point-to-point attacks

- Attacker chooses victim
- Sends network packets to isolate victim
- Goal of attacker
  - Small number of packets ⇒ big effect

TCP Handshake



SYN Flooding



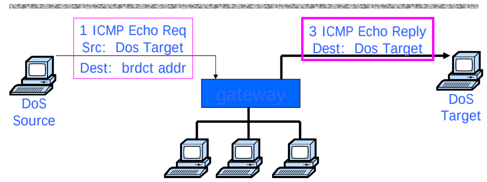
TCP Reset vulnerability [Watson'04]

- 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?

Smurf DoS Attack



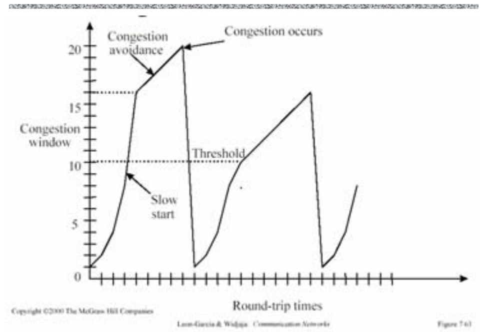
- Send ping request to broadcast addr (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim
  - Ping reply stream can overload victim

Prevention: reject external packets to broadcast address

TCP Congestion Control

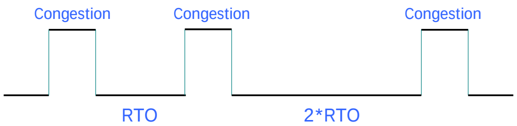
- Sender estimates available bandwidth
  - Starts slow and increases based on ACKS
  - Reduces rate if congestion is observed
- Two time scales
  - RTT is 10-100 ms ⇒ TCP performs AIMD
    - Additive Increase Multiplicative Decrease
    - Rises slowly, drops quickly (by half)
  - Severe congestion ⇒ 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

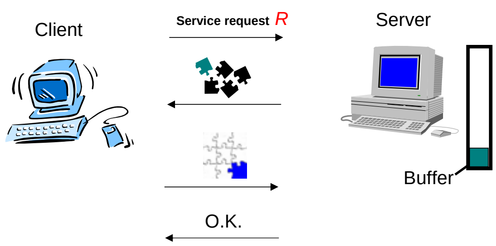
Defense against "connection depletion" attacks

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 suspected, server responds with puzzle
  - Allows connection only for 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

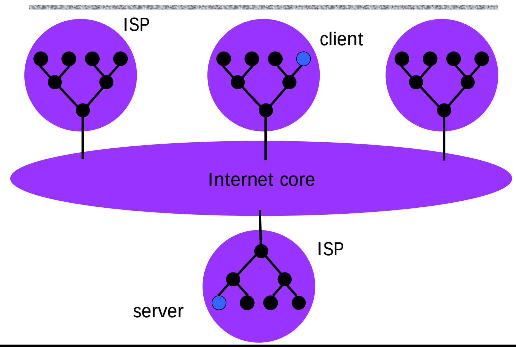
Outline

- 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
  - Bot networks
- Mitigation techniques
  - Firewall
  - IP traceback
    - Edge Sampling techniques
  - Overlay networks
    - Migration
    - Authentication

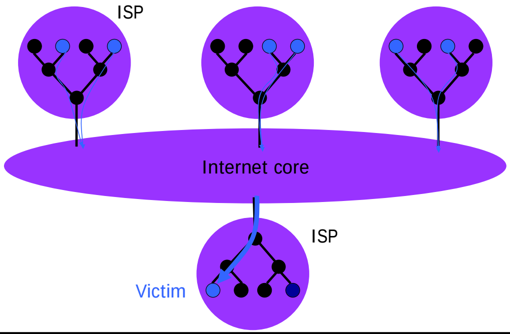
Distributed denial of service

- Attacker sets up network of machines
  - Break in by buffer overflow, etc.
- Attack machines bombard victim
- Attacker can be off line when attack occurs

Internet



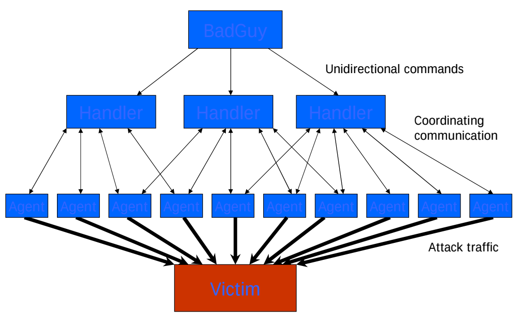
Distributed denial of service



Feb 2000 Distributed DOS Attack

- Observable effect
  - Most of Yahoo unreachable for three hours
  - Experts did not understand why
    - "An engineer at another company … told Wired News the outage was due to misconfigured equipment"
- 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 overlay network



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
  - Fixed payload size
- Client-Daemon communication only in ICMP
  - No passwords for client
  - Does not authenticate incoming ICMP

Stacheldraht

- Client to Handler to Agent to Victim
  - Like Trin00
- Combines Trin00 and TFN features
  - Authenticates communication
  - Communication encrypted by symmetric key
  - Able to upgrade agents on demand

Traffic Characteristics

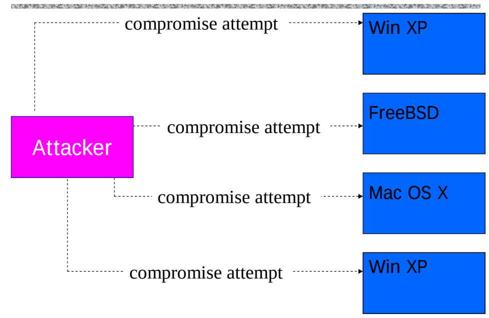
- Trinoo
  - Port 1524 tcp   Port 27665 tcp
  - Port 27444 udp   Port 31335 udp
- TFN
  - ICMP ECHO and ICMP ECHO REPLY packets.
- Stacheldraht
  - Port 16660 tcp   Port 65000 tcp
  - ICMP ECHO and ICMP ECHO REPLY
- TFN2K
  - Ports supplied at run time or chosen randomly
  - Combination of UDP, ICMP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

BOT Networks

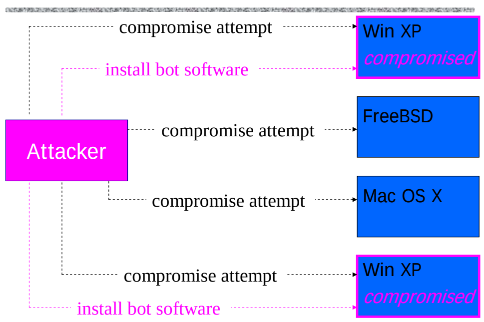
- What is a bot network?
  - Group of compromised systems with software installed on them to allow simple remote control
  - Software on zombies upgradeable via IRC or P2P
- Used as attack base for various activities
  - DDoS attacks
  - Spam forwarding
  - Launching pad for new exploits/worms
  - Install keylogger to capture passwords and product activation codes

Thanks: Alissa Cooper

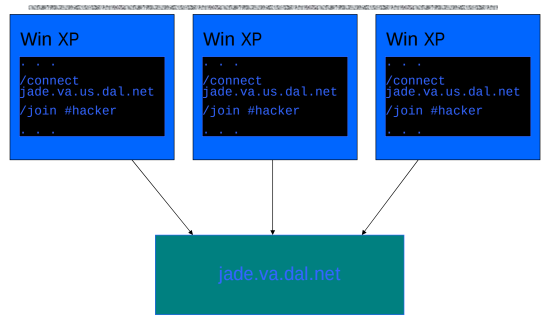
Building a Bot Network



Building a Bot Network



Step 2



Step 3

```
(12:59:27pm) -- A9-pcgbdv (A9-pcgbdv@140.134.36.124) has joined (#owned) Users : 1646

(12:59:27pm) (@Attacker) .ddos.synflood 216.209.82.62

(12:59:27pm) -- A6-bpxufrd (A6-bpxufrd@wp95-81.introweb.nl) has joined (#owned) Users : 1647

(12:59:27pm) -- A9-nzmpah (A9-nzmpah@140.122.200.221) has left IRC (Connection reset by peer)

(12:59:28pm) (@Attacker) .scan.enable DCOM

(12:59:28pm) -- A9-tzrkeasv (A9-tzrkeas@220.89.66.93) has joined (#owned) Users : 1650
```

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
  - Bot networks
- ➡ Mitigation techniques
  - Firewall
  - IP traceback
    - Edge Sampling techniques
  - Overlay networks
    - Migration
    - Authentication

Mitigation efforts

- Firewall
  - Protect server, not ISP
  - (More about firewalls next lecture)
- Find source of attack
  - Used to shut down attack
  - Sometimes possible to find culprit
- Overlay techniques
  - Preserve service to authenticating clients

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?

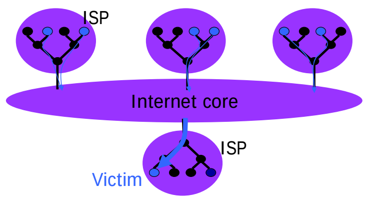
- Hard to find BadGuy
  - Originator of attack compromised the handlers
  - Originator not active when DDOS attack occurs
- Can try to find agents
  - Source IP address in packets is not reliable
  - Need to examine traffic at many points, modify traffic, or modify routers

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
  - Can provide 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 determines 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

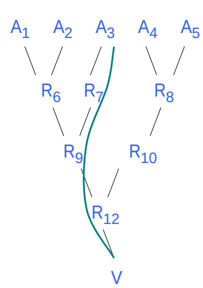
Logging

- Critical 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

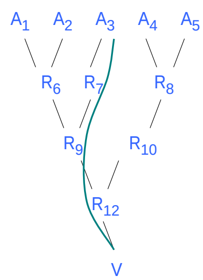


Simple method

- Write path into network packet
  - 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

  if coin turns up heads (with probability p) then

  write R into start address

  write 0 into distance field

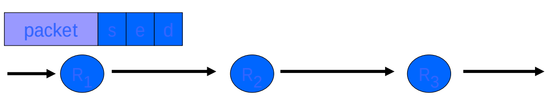
  else

  if distance ==0 write R into end field

  increment distance field

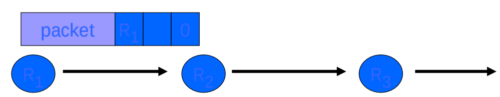
Edge Sampling: picture

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



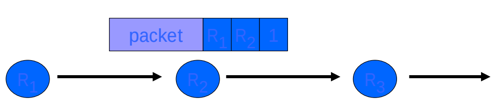
Edge Sampling: picture

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



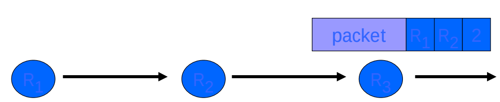
Edge Sampling

- 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 information 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

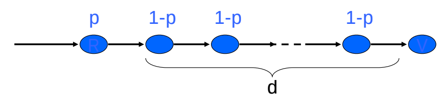
$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

where p is marking probability, d is length of path

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

Node Sampling?

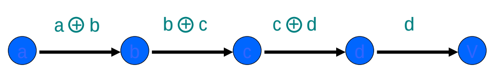
- Less data than edge sampling
  - Each router writes own address with probability p
- Infer order by number of packets
  - Router at distance d has probability $p(1-p)^d$ of showing up in marked packet



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

Reduce Space Requirement

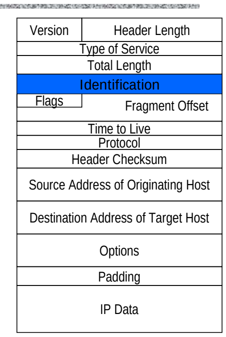
- XOR edge IP addresses
  - Store edge as start ⊕ end
  - Work backwards to get path:
    (start ⊕ end) ⊕ end = start
- Sample attack path



Details: where to store edge

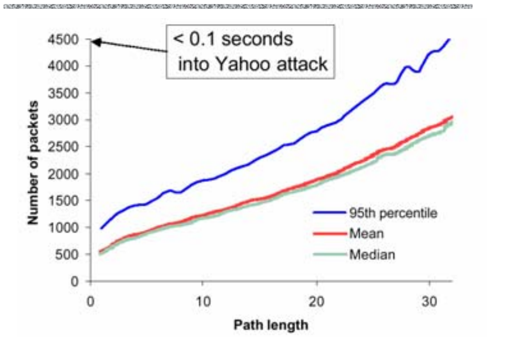
- Identification field
  - Used for fragmentation
  - Fragmentation is rare
  - 16 bits
- Store edge in 16 bits?

  - Break into chunks
  - Store start ⊕ end



Experimental convergence time

[Savage et al]



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

  if coins flip is heads (with probability p)
  write h(R) into address field
  write 0 into distance field
  else
  if distance ==0 set field = field $\oplus$ h'(R)
  increment distance field

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

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 = 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, …

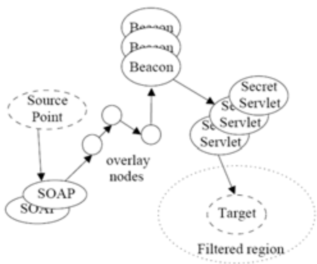
Outline

- 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
  - Bot networks
- Mitigation techniques
  - Firewall
  - IP traceback
    - Edge Sampling techniques
  - ➡ Overlay networks
    - Migration
    - Authentication

Secure Overlay Services (SOS)

- Maintain access in face of DDOS attack
  - Move site to another location on overlay network
  - Forward “good” traffic to new location
- Separate good from bad/unknown traffic
  - Authenticate users for entering the overlay
  - Route good traffic through overlay
- Assumptio
  - Attackers cannot saturate Internet core

SOS picture

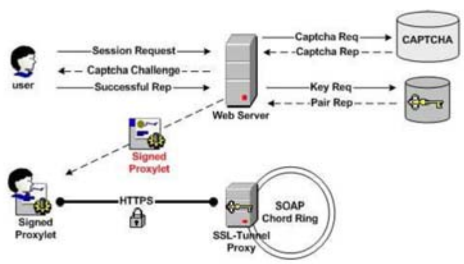


Angelos Keromytis

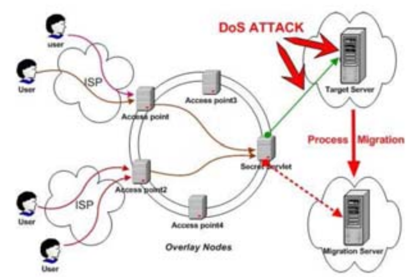
Authentication in SOS

- Requiring known users is too restrictive
- Goal: guarantee no "zombies"
- Graphic Turing Tests
  - Tests that humans can perform, but difficult for computers

CAPTCHA in secure overlay service

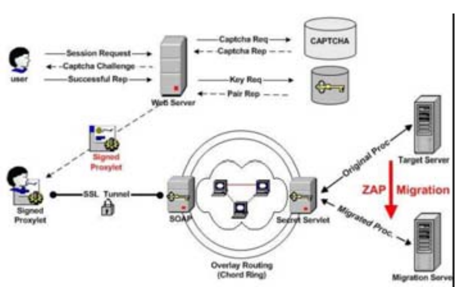


Migrating OVErlay (MOVE)



Columbia Univ project

Captcha and migration



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
  - Bot networks
- Mitigation techniques
  - IP traceback
    - Edge Sampling techniques
  - Overlay networks
    - Migration
    - Authentication