

# IT 4823

## Information Security Administration

### Denial of Service Attacks



Notice: This session is  
being recorded.

Some lecture slides prepared by Dr Lawrie Brown for “*Computer Security: Principles and Practice*”, 1/e, by William Stallings and Lawrie Brown



Copyright © 2016 by Bob Brown



# Denial of Service

- **Denial of service (DoS)** is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources.
- Attacks can be against:
  - network bandwidth
  - system resources (CPU, disk, etc.)
  - application resources
- Not new, although methods may change

# Resources Attacked

## Network bandwidth

Relates to the capacity of the network links connecting a server to the Internet. For most organizations this is their connection to their Internet Service Provider (ISP)

## System Resources

Aims to overload or crash the network handling software, operating system resources, disk space, or other resource

## Application Resources

Typically involves a number of valid requests which consume significant resources, limiting the ability of the server to respond to requests from other users

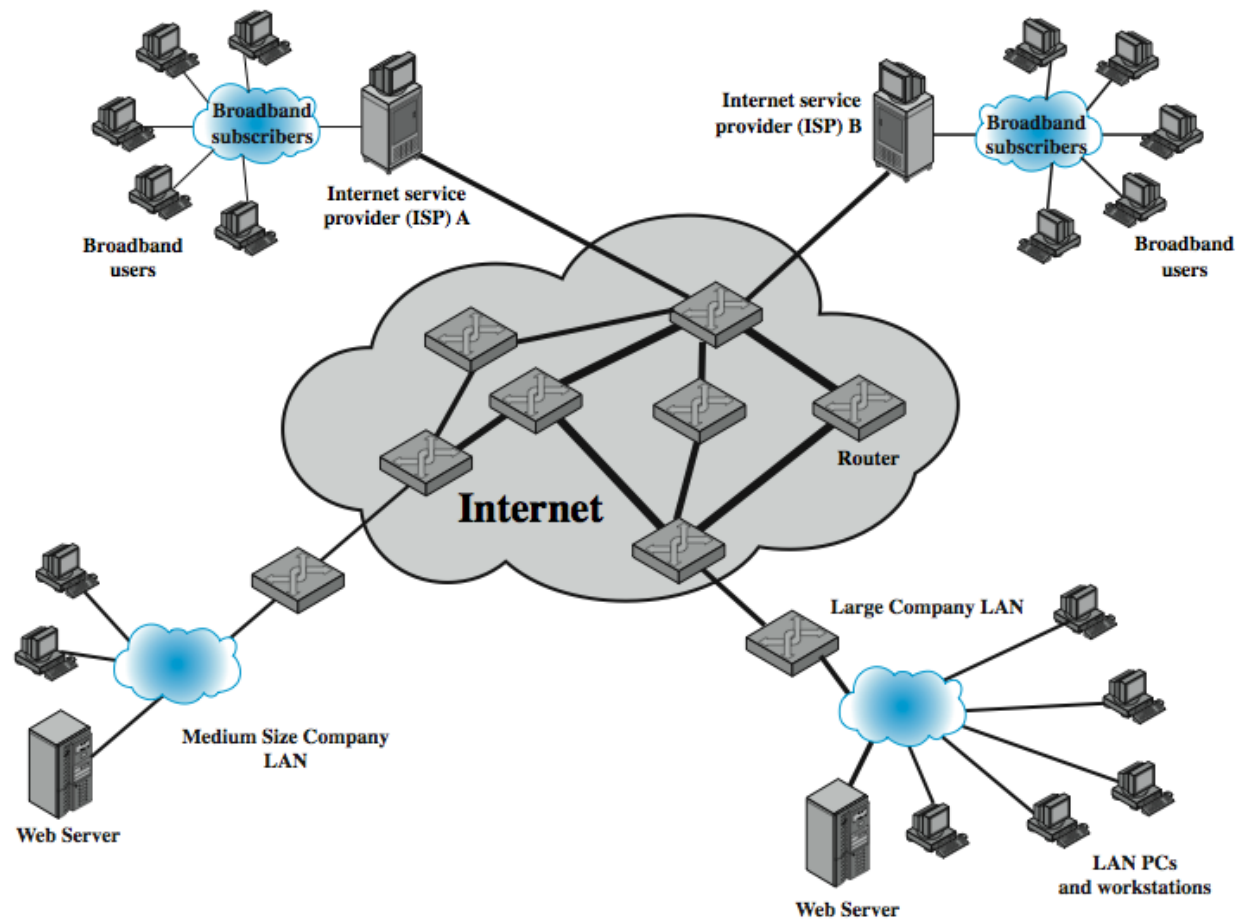
# Thinking About Denial of Service

- Not all service-denial events are attacks. (What does it mean to be “slash-dotted?”)
- Not all DoS attacks succeed in denying service. (How will you identify these?)
- First reports usually describe symptoms, *e.g.* “The Internet is slow.”

# Classic Denial of Service Attacks

- An attacker can use simple ping flooding from higher capacity link to one of lower capacity, causing loss of traffic.
- The source of flood traffic is easily identified when the mode is ping flooding.

# Classic Denial of Service Attacks



# Problems

- Malformed packets
- Source address spoofing
- Flooding attacks
- Distributed denial of service attacks (DDoS)
- Reflection attacks
- Amplification attacks
- Lack of appropriate response to attacks
- Unprotected computers as attack platforms.

<http://www.sans.org/dosstep/roadmap.php>

# Malformed Packets

- Also called poison packets, magic strings
- A single, specially-crafted packet triggers an error in in operating system or application.

Consequences:

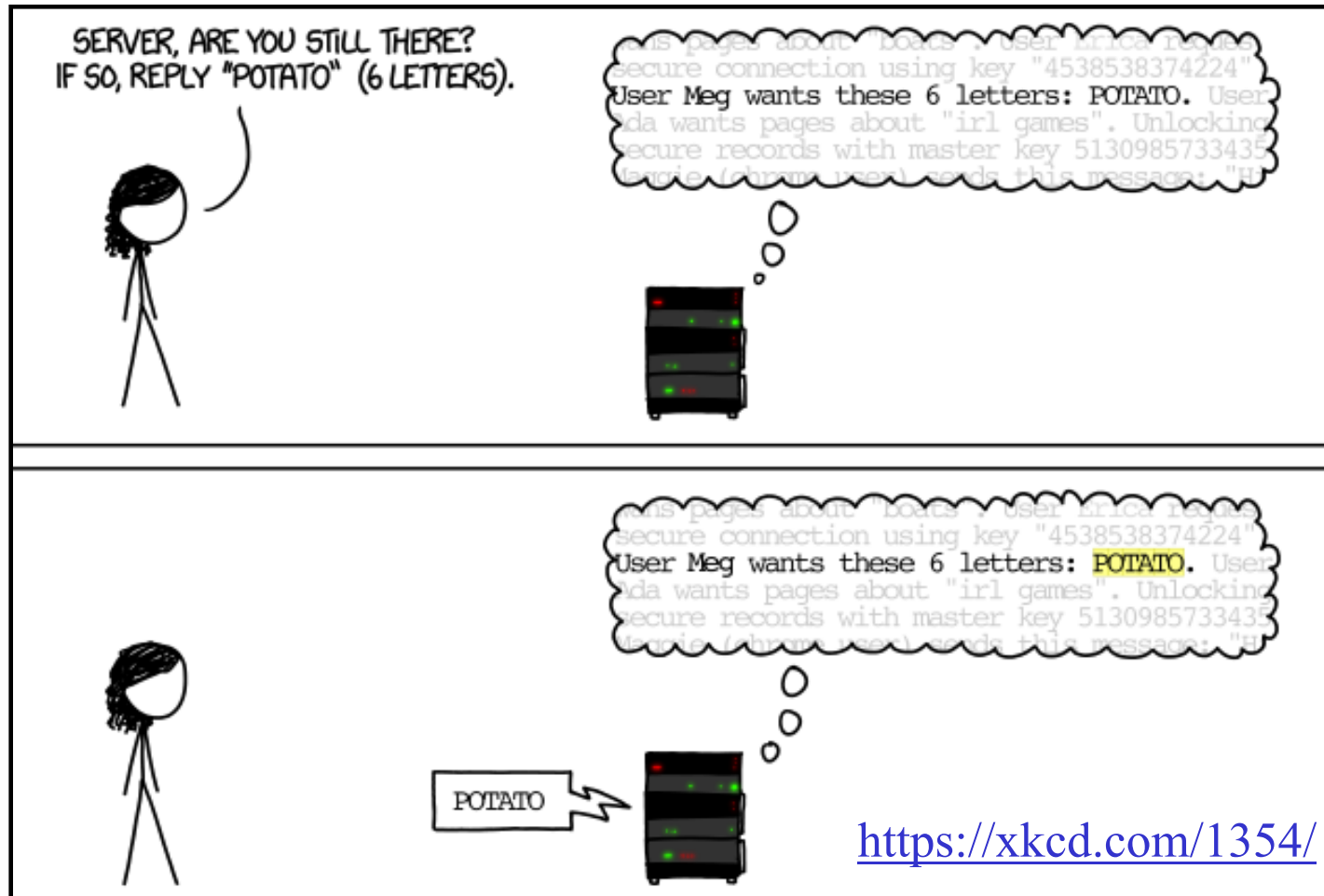
- Crash
- Arbitrary code execution
- Examples
  - Shellshock (Linux systems, 2014)
  - Heartbleed (SSL, 2014)
  - Ping of death (TCP, Windows 95)
  - Back attack (Apache, 1999 or so)



# The Shellshock Vulnerability

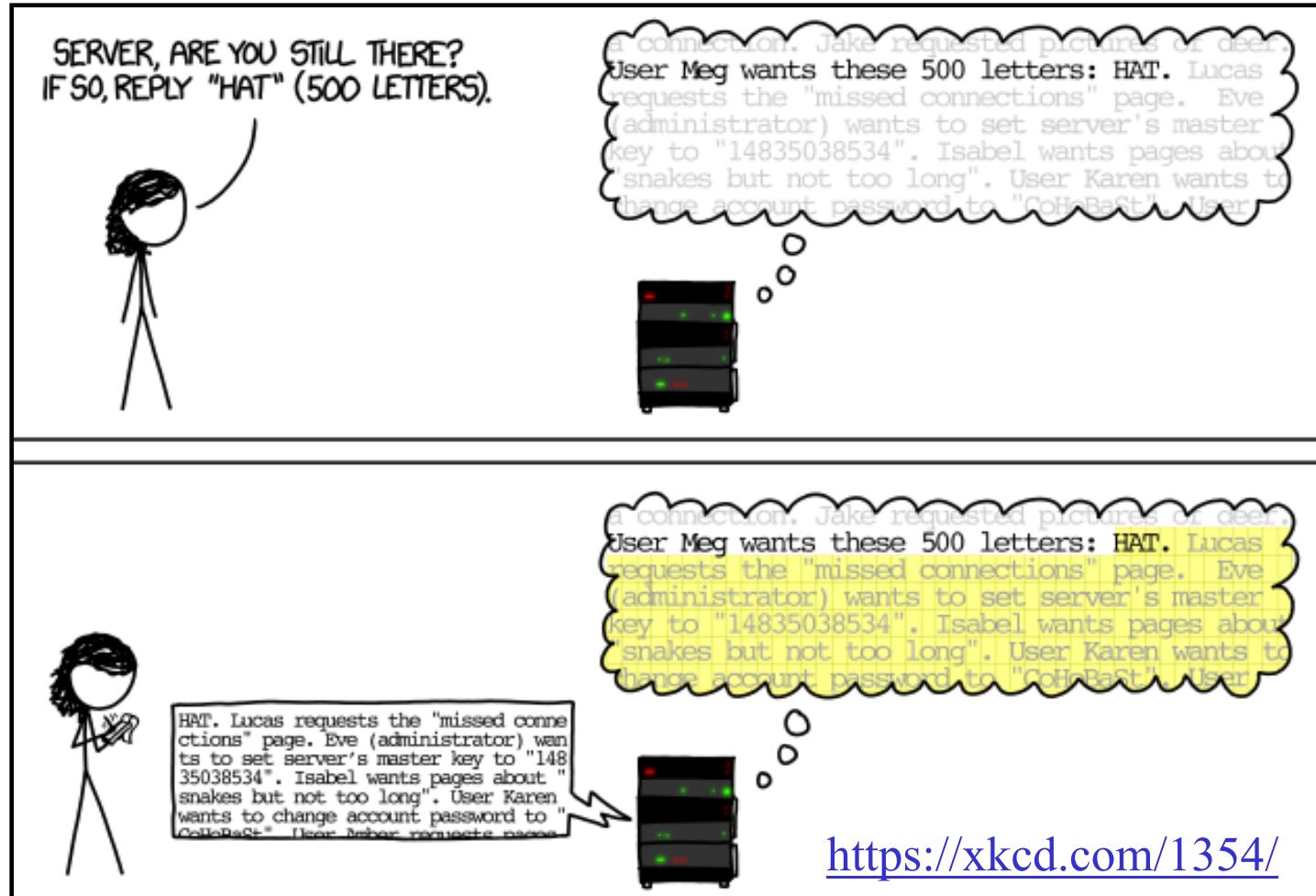
- An error in the bash shell (widely used in Linux)
- Can allow execution of arbitrary shell commands, *e.g.* ‘rm -rf /’
- This ejects the CD/DVD drive:  
`curl -H "User-Agent: () { :; }; /bin/eject"  
http://example.com/`
- The bash shell incorrectly interprets what should be a function, executes unintended commands.

# SSL Keepalive



<https://xkcd.com/1354/>

# The Heartbleed Vulnerability



# Defending Against Poison Packets

- Keep software up to date.
- Do not run unneeded services.
- Keep abreast of security news:  
<https://isc.sans.edu/>
- Run anti-malware software.
- Practice good computer security in general.

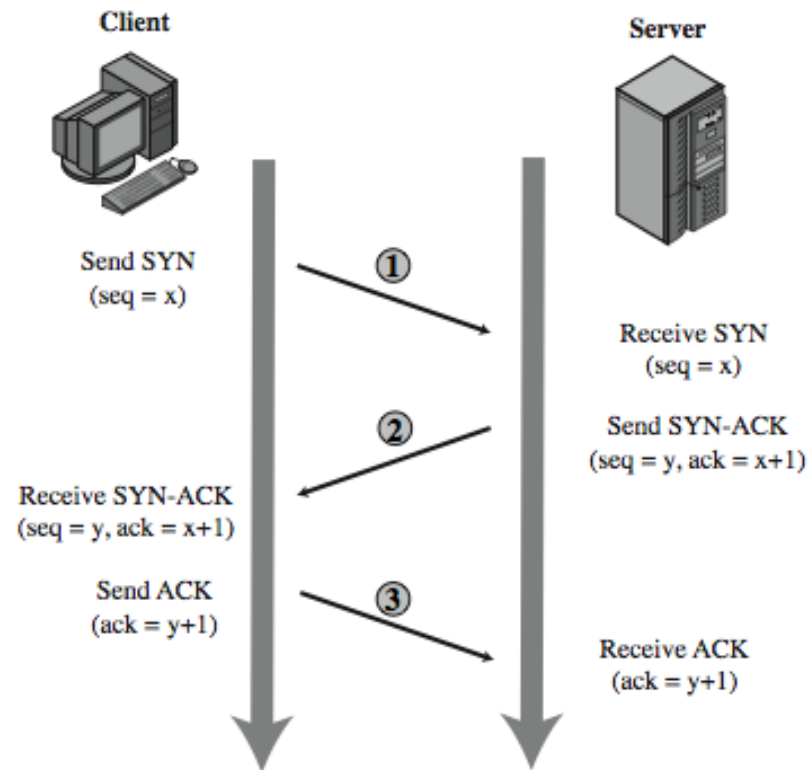
# Source Address Spoofing

- Uses forged source addresses
  - Needs sufficient privilege to create “raw sockets”
  - Easy to create, especially on MS operating systems
- Attacker generates large volumes of packets
  - Directed at target
  - With different, random, source addresses
- Causes the same congestion
- Responses are scattered across Internet
- Real source is much harder to identify

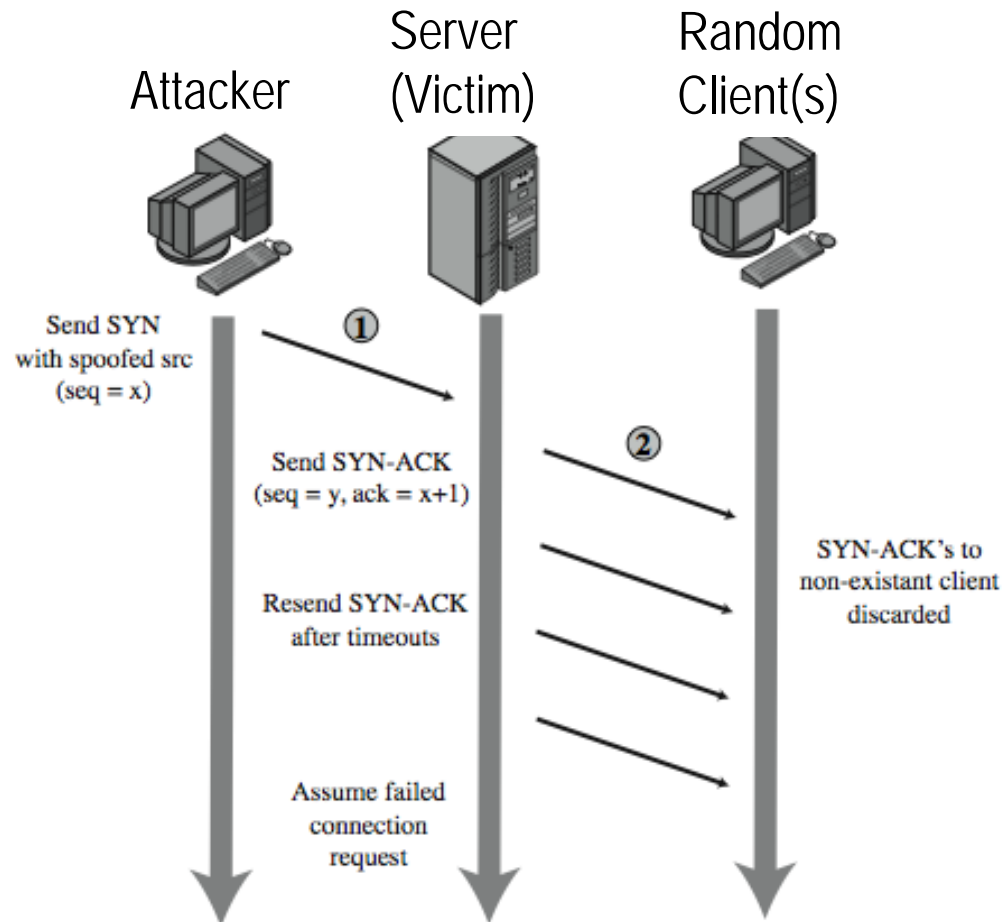
# SYN Flooding

- Attacks ability of a server to respond to future connection requests by overflowing tables used to manage them.
- Hence, an attack on an operating system resource.

# TCP Connection Handshake



# SYN Flooding Attack





# SYN Flooding Attack

- Attacker often uses either random source addresses or that of an overloaded server to block return of (most) reset packets.
- Needs much lower traffic volume  
attacker can be on a much lower capacity link.

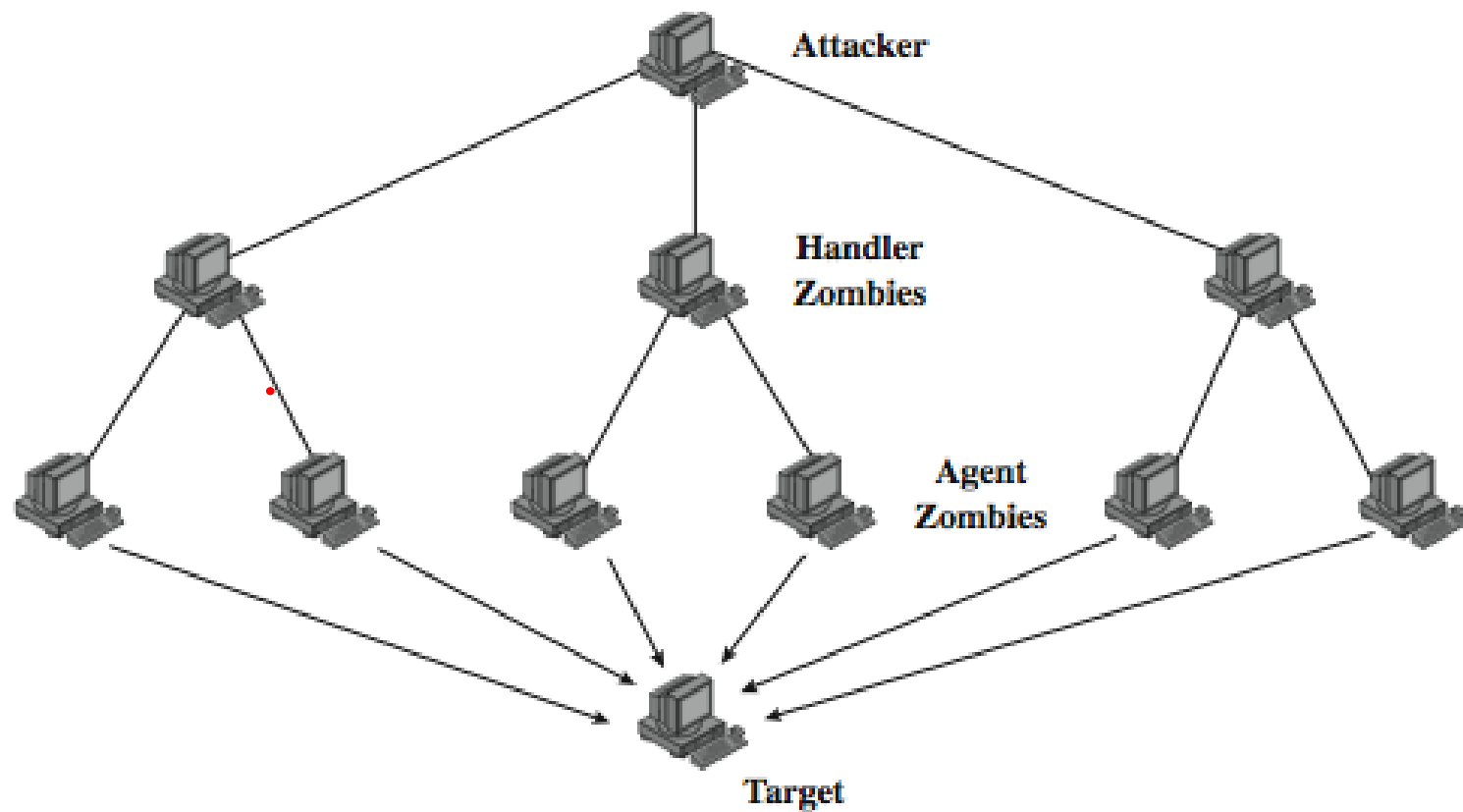
# Other Types of Flooding Attacks

- Classified based on network protocol used
- ICMP Flood
  - Uses ICMP packets, *e.g.* echo request (ping.)
  - ICMP packets are typically allowed through well-managed routers; some are required for proper network operation
- UDP Flood
  - Uses UDP packets to some port

# Distributed Denial of Service Attacks

- Denial of service attacks have limited volume if single source used
- Multiple systems allow much higher traffic volumes to form a Distributed Denial of Service (DDoS) Attack
- Often compromised PC's / workstations
  - zombies with backdoor programs installed
  - forming a botnet

# DDoS Control Hierarchy



# SIP Flood

- The Session Initiation Protocol (SIP) is a standard protocol for VoIP telephony.
- A single SIP INVITE transaction consumes a relatively large amount of resources.
- So, even a modest number of spoofed SIP INVITE requests can cause denial of service.

# HTTP Attacks

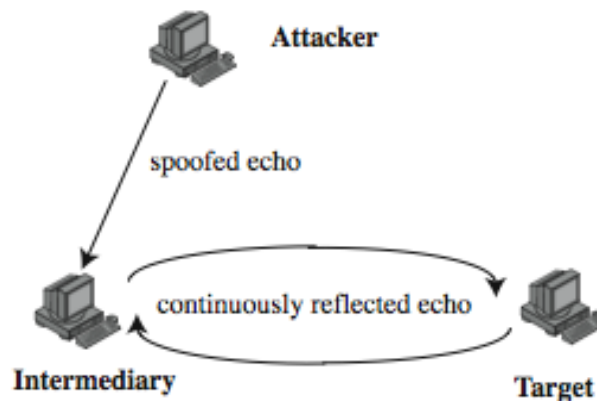
- Similar to others; consume resources of Web servers.
- Can use botnets, etc. for distributed attacks.
- “Slowloris” can bog down a Web server using only one or a few attacker machines:
  - Attacker sends a partial set of headers – server waits for the rest.
  - Attacker uses numerous threads to send requests.
  - Eventually the server’s connection pool is exhausted.

# Reflection Attacks

- Use the normal behavior of the network
- The attacker sends packet with spoofed source address being that of target to a server
- Server response is directed at target
- If there are many requests to multiple servers, responses can flood target
- Various protocols *e.g.* UDP or TCP/SYN
- Ideally, the attacker want response larger than the request
- Prevented by blocking source spoofing

# Reflection Attacks

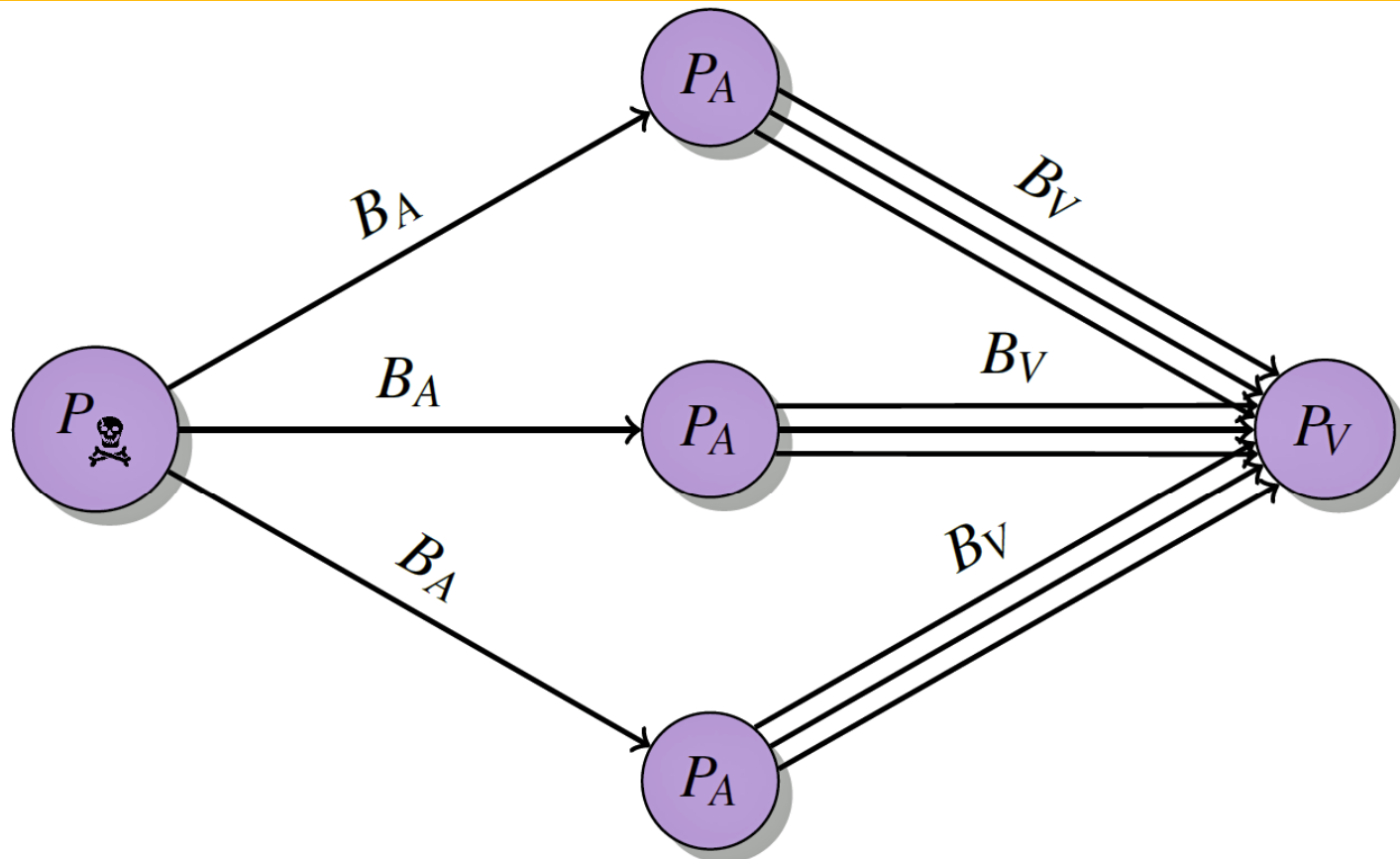
- Further variation creates a self-contained loop between intermediary and target
- Fairly easy to filter and block



See: CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack. Attacker connected the CHARGEN port of one machine to the ECHO port of another.



# BitTorrent DRDoS



<http://arstechnica.com/security/2015/08/how-bittorrent-could-let-lone-ddos-attackers-bring-down-big-sites/>

Attacker

Amplifiers

Victim

# Prevention

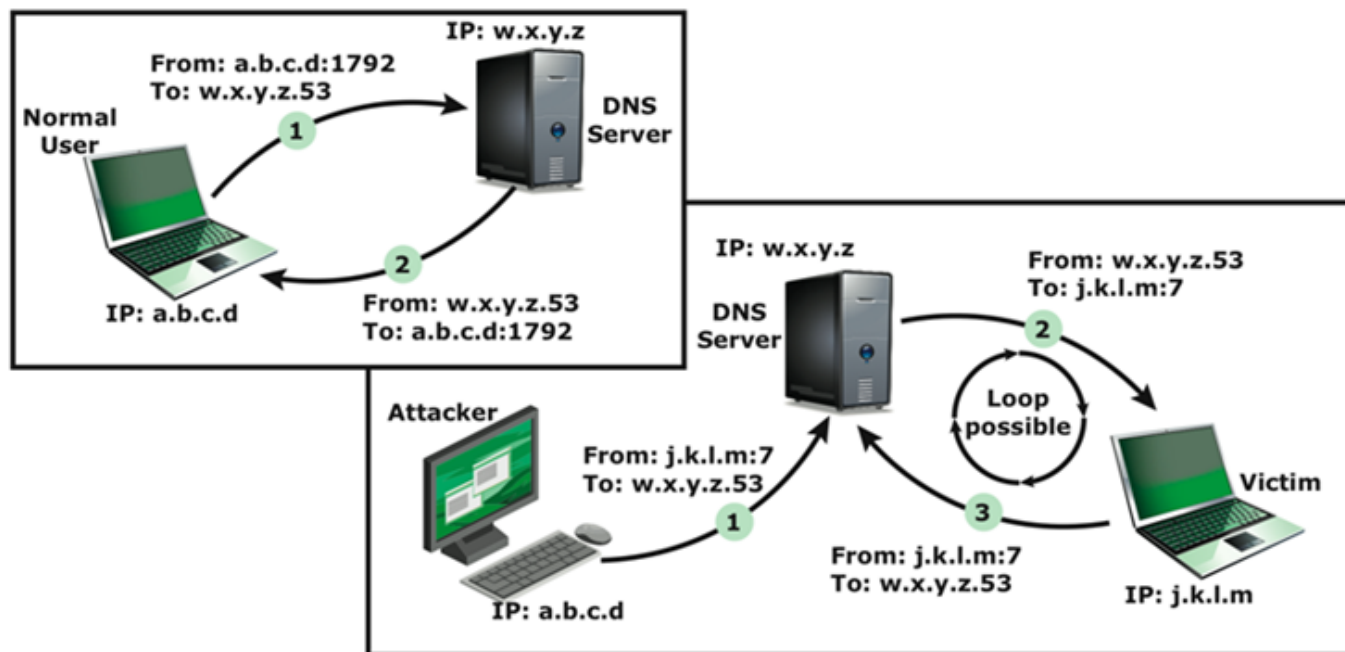
- Verify Reverse Path: A feature of Cisco and other routers, packets that do not have a route *back* to the source address on the incoming interface are dropped. Also called Reverse Path Forward, or RPF.
- Block RFC-1918 source addresses.
- Block own-network addresses in and off-network addresses out (source spoofing.)
- Rate-limit ICMP packets.

# Prevention

- Block UDP ports not needed for regular operation (*e.g.* CHARGEN and ECHO.)
- Disable unused/unneeded UDP services on hosts.

# DNS Reflection

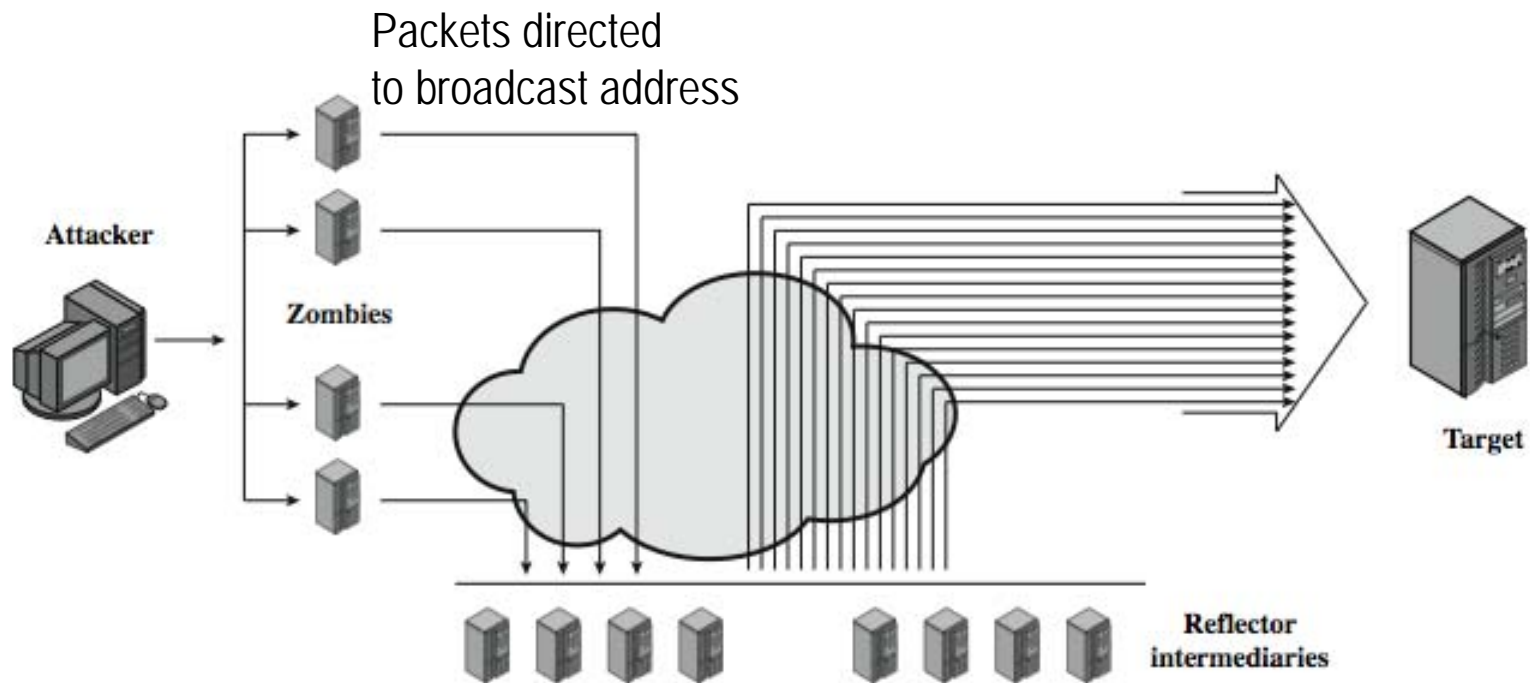
- Attacker sets up a loop between a DNS server and the “echo” port on the attacked machine.
- Prevention: Firewall rules.



# DNS Amplification Attacks

- Uses DNS requests with spoofed source address being the target
- Exploit normal DNS behavior: a small request produces much larger response; 60 byte request to 512 – 4,000 byte response
- Attacker sends requests to multiple well-connected servers, which flood target with responses
  - need only moderate flow of request packets
  - DNS servers will also be loaded

# Amplification Attacks



Example: Send “ping” to the broadcast address of a large network. Spoof the source address of the target.

# Thinking About DoS Attacks

- High traffic volumes may be legitimate
  - result of high publicity, *e.g.* “slash-dotted”
  - or to a very popular site, *e.g.* Olympics etc.
- Or legitimate traffic created by an attacker
- Three lines of defense against (D)DoS:
  - Attack prevention and preemption
  - Attack detection and filtering
  - Attack source traceback and identification (?)

# Attack Prevention

- Block spoofed source addresses *at output*
  - On routers as close to source as possible
  - [BCP 38 \(RFC 2827\)](#)
  - Still far too rarely implemented
- Example:
  - I operate network 12.232.184.0, with 254 host addresses.
  - My own edge router should block (drop) any outbound packets with source addresses *not* in 12.232.184.0–255.
  - If so, my network can never be a source of attacks based on address spoofing.
- ***Other networks*** prevent the attack.



# Attack Prevention

- Rate limiting in upstream distribution nets
  - on specific packets types, *e.g.* ICMP, some UDP, TCP/SYN
- Use modified TCP connection handling
  - Use SYN cookies when table full
  - Selective or random drop when table full
  - Shortened time for connection setup

# Attack Prevention

- Block IP directed broadcast packets; no *incoming* packets to broadcast addresses.
- Block suspicious services and combinations
- Manage application attacks with “puzzles” to (try to) distinguish legitimate human requests
- Good general system security practices
- Use mirrored and replicated servers when high-performance and reliability are required

# Legitimate Overloads

- We've talked about a Web site that has been “slash-dotted?”
- What other kinds of events can cause similar spikes in traffic?
- Operator's dilemma:
  - Prepare for such events (costly)
  - Ride through them (also costly!)
  - Use cloud services, maybe...

# Detection

- How can one tell whether an “overload” condition is an attack or a legitimate increase in traffic?
- Some traffic, in volume, is always an indication of an attack. Example: SYN floods, “Martian” DNS responses.
- Other traffic may be harder to analyze.

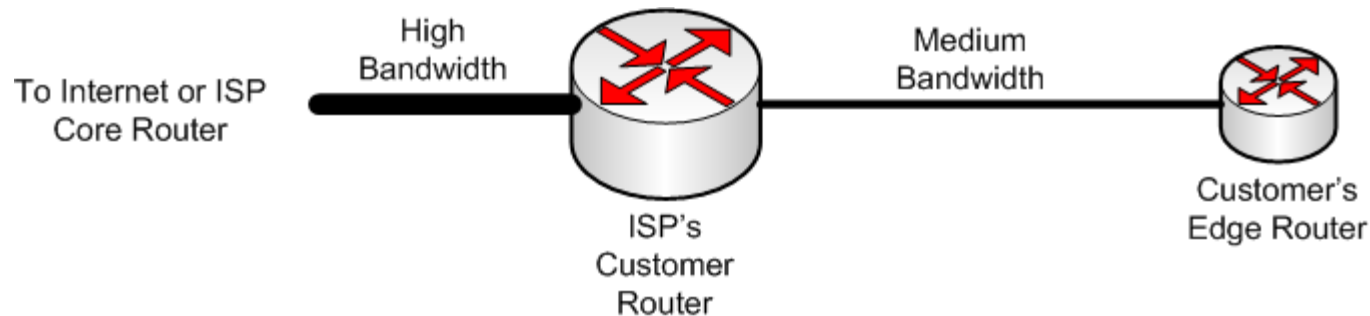
# Responding to Attacks

- Need a good incident response plan...
  - With prearranged contacts for ISP
  - Needed to impose traffic filtering upstream
  - Details of response process must be planned
- Have standard filters in place
- Ideally have network monitors and IDS to detect and notify of abnormal traffic patterns

# Responding to Attacks

- Identify type of attack
  - capture and analyze packets (*e.g.* with TCPDUMP)
  - Have ISP install filters to block attack traffic upstream
  - or identify and correct system/application bug
- Consider having ISP trace packet flow back to source:
  - may be difficult and time consuming
  - necessary if legal action is contemplated
- Implement contingency plan
- Update incident response plan

# Filtering: Where?



- Some attacks can be filtered at the edge router.
- Attacks that consume bandwidth can *only* be filtered effectively at the ISP's customer router, or before. Why?

# Mitigation

- Reduce Maximum Segment Life (MSL) to mitigate SYN floods.
- Drop (instead of sending RST) TCP packets to closed ports.
- Drop UDP packets to closed ports.
- Rate-limit outgoing ICMP “unreachable” and TCP RST packets.



# Being a Good Netizen

- Be alert for malicious software
- Keep patches up to date (on routers and other network gear as well as operating systems.)
- Deny outbound malicious traffic at your edge router (*example*: filtering spoofed addresses.)
- Disable and also block unused services (defense in depth.)
- Supply training for systems and network administrators.

# Longer-Term Solutions

- Adoption of IPsec. (Included in IPv6)
- Adoption of DNSSec. (DNSSec root zones deployed July 15, 2010)
- Load and volume monitoring by ISPs for early warning.
- Further research in intrusion detection.

<http://www.sans.org/dosstep/roadmap.php>

# Postscript: Thinking About Security

[Do not fall into] the classic security misapprehension error: the idea that either you're "secure" or you're not.

The real question, as we all know, should be, "against what sort of attacks am I vulnerable?"

*—Curt Sampson*

# Questions

