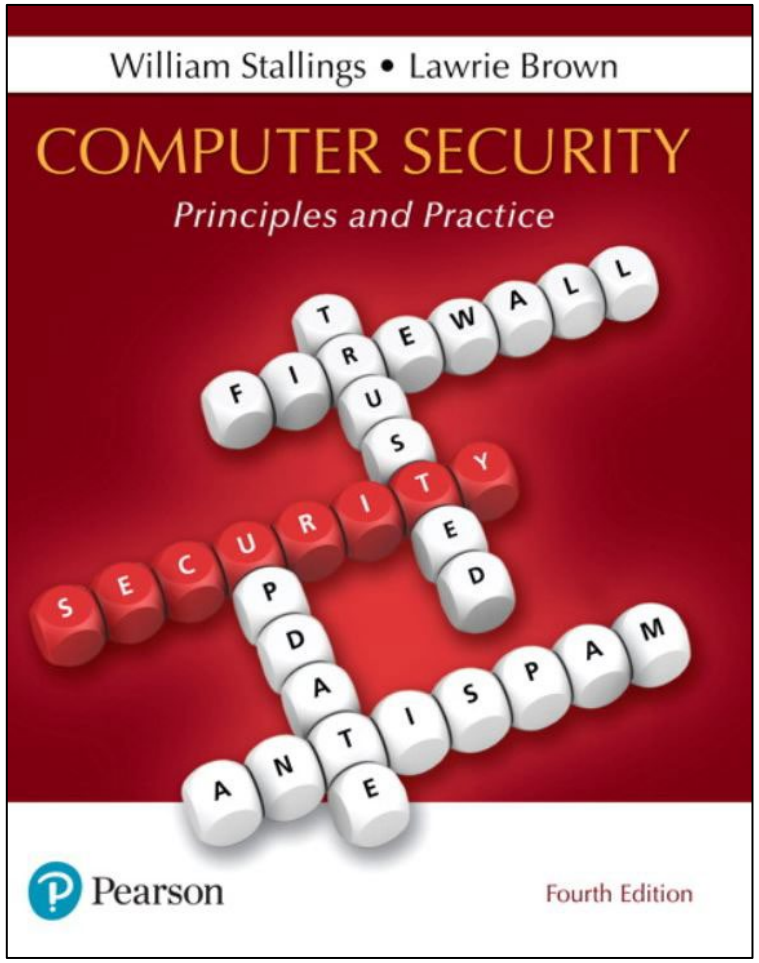
Denial of Service attacks

CalPolyPomona

Computer Security: Principles and Practice



- **Chapter 7**

- Denial-of-Service Attacks

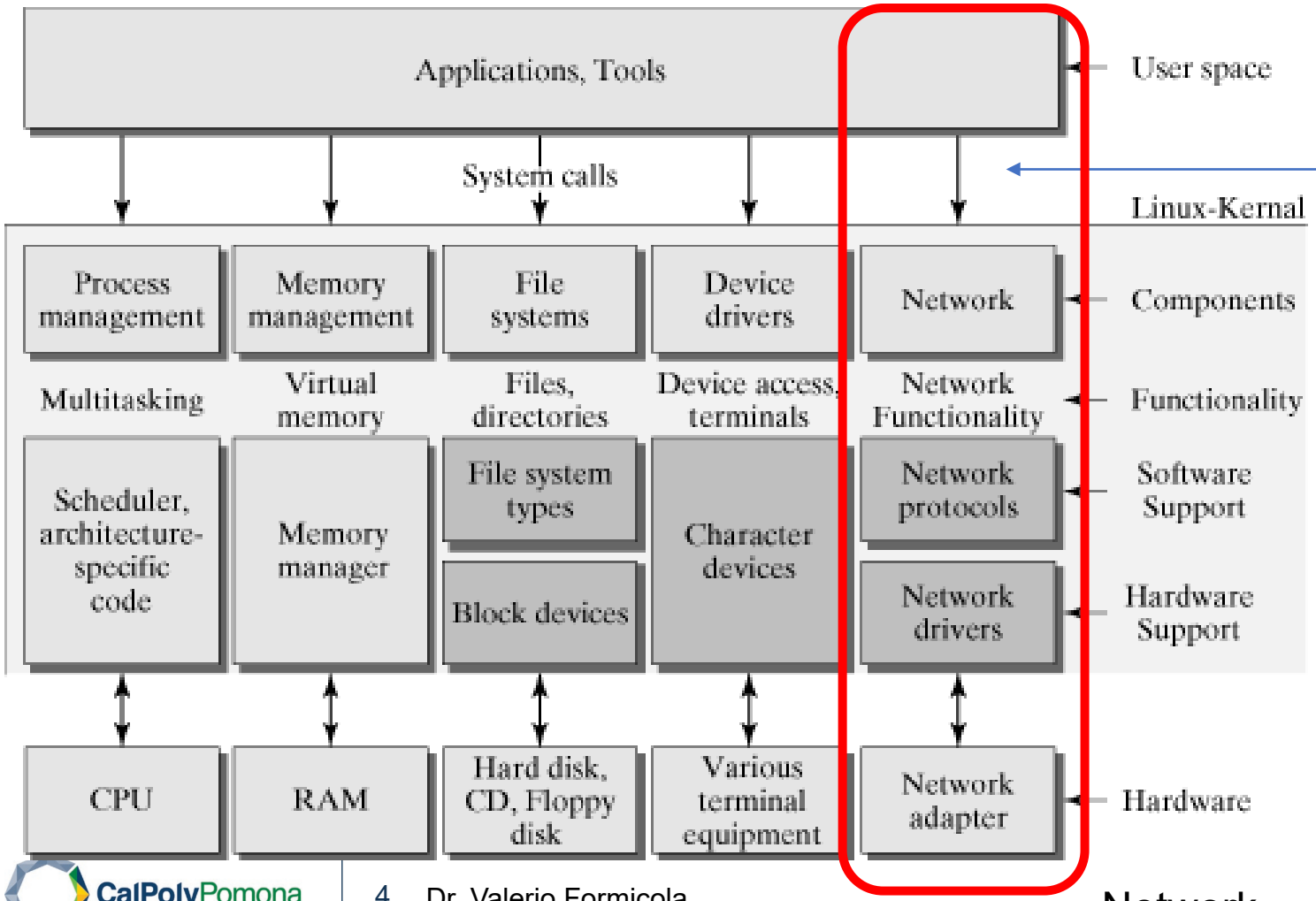
Denial-of-Service (DoS) Attack

The NIST Computer Security Incident Handling Guide defines a DoS attack as:

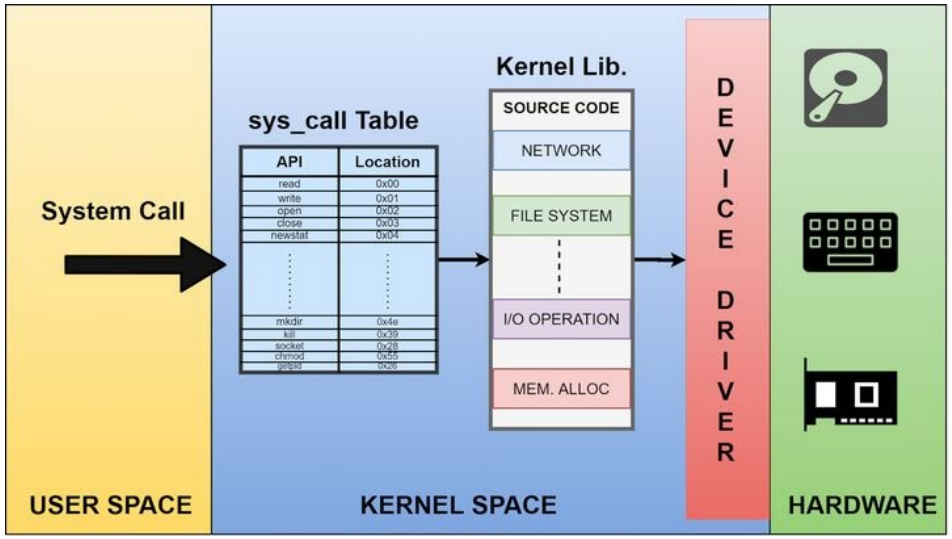
"An action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space."

DoS attacks are attacks to the AVAILABILITY
(remember, availability is one of the three main security pillars CIA)
Note: here, we will refer to the denial of services available on a network of computers, not on isolated machines

Recall: Operating System – Linux example



System calls = O.S. "APIs"

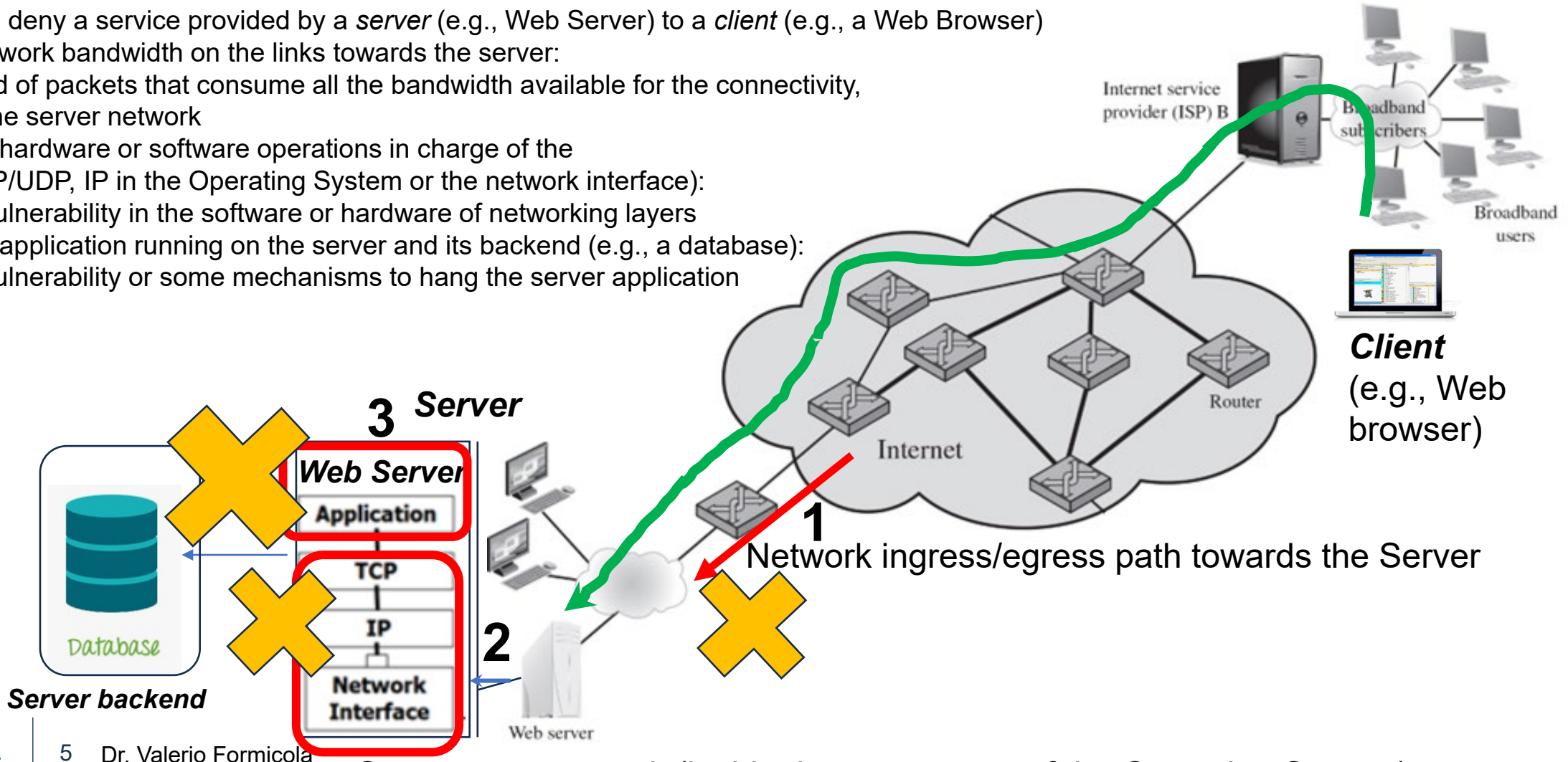


Network

How to stop a client from receiving a service in a remote host

Three alternatives to deny a service provided by a *server* (e.g., Web Server) to a *client* (e.g., a Web Browser)

1. Congest the network bandwidth on the links towards the server:
   how: with a flood of packets that consume all the bandwidth available for the connectivity, in proximity of the server network
2. Crash/hang the hardware or software operations in charge of the networking (TCP/UDP, IP in the Operating System or the network interface):
   how: exploit a vulnerability in the software or hardware of networking layers
3. Crash/hang the application running on the server and its backend (e.g., a database):
   how: exploit a vulnerability or some mechanisms to hang the server application

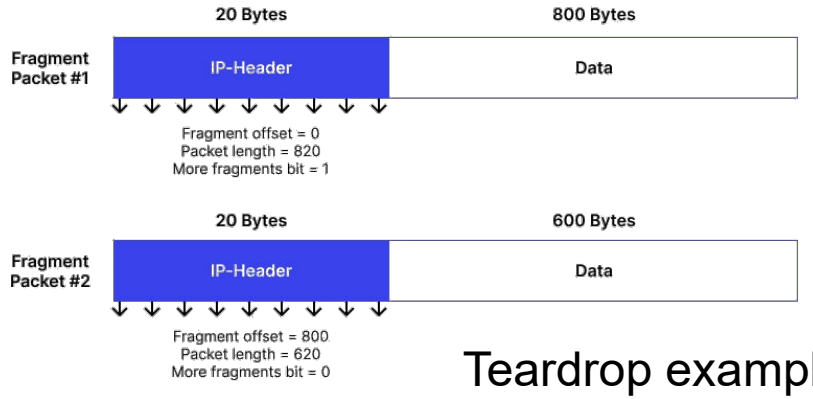


Denial-of-Service (DoS) - overview

- A form of attack on the availability of some service
- Categories of resources that could be attacked are:
    1. 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)
        - Examples: SYN Spoofing, ICMP/SYN/UDP floodings, Broadcast Reflections, DNS Amplifications, Smurf/Fraggle;
    2. System resources
        - Aims to overload or crash the network handling software (e.g., the Operating System cannot handle some corrupted packets and crashes) or alter networking system operation
        - Examples: Teardrop and IP fragmentation attacks, Land, Targa3, Ping of Death
          Wireless specific: Blackhole, Rogue AP, Disassociation attack
    3. Application resources
        - Typically involves a number of valid requests, each of which consumes significant resources, thus limiting the ability of the server to respond to requests from other users
        - Example: SIP Invite flood, SIP Invite-of-death, HTTP cyberslam, HTTP Slowris

IP Packet attacks (not network flooding) – System resource exhaustion or crash

- **IP malformation attack:**
  - Random optional fields of IP packet:
    E.g., all qualities of service bits are set to 1, victims apply some additional time to analyze the packet and slow processing.
    Flood of packets will hang the OS.
- **IP fragmentation attack** (notably, **Teardrop**)
  - Example: Fragments have overlapping sequences
  - There are many others:
    https://en.wikipedia.org/wiki/IP_fragmentation_attack
- **Land attack**
  - TCP/IP version: Src IP + Src Port == Dst IP + Dst Port
  - Requires *spoofing* of source IP address: fake source address is declared
  - Spoofed TCP SYN packet (connection initiation) with the target host's IP address to an open port as both source and destination. This causes the machine to reply to itself continuously till exhausting resources.
- **Targa3**
  - Uncommon IP packets consist of invalid fragmentation, protocol, packet size, header values, options, offsets, TCP segments, and routing flag. It caused crash of Windows.



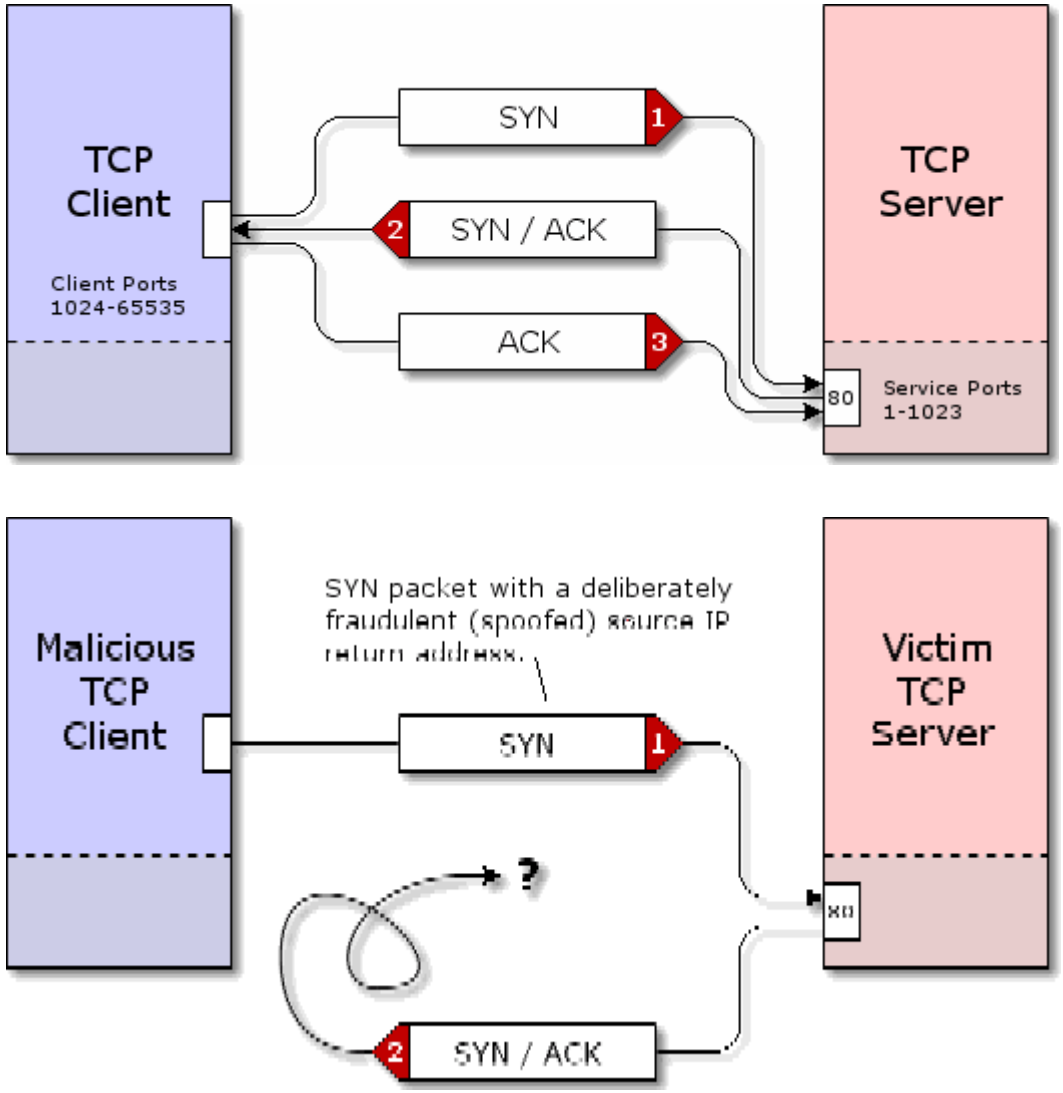
Teardrop example

ICMP – single packet attack
Ping of Death

- **Ping of Death:**
  - ICMP fragment size larger than the max size (65507, some bytes are used for ICMP header)
  - Crash of old versions of Unix, Linux, Mac, Windows, etc.
  - Solution:
    - Firewalls might drop malformed packets
    - Limit ICMP packet traffic only from some sources or IP ranges

SYN spoofing (TCP/IP) – 1/2 – (not network flooding)

- Common DoS attack
- It is NOT BASED on exhausting of network bandwidth, but the resources on a server machine
- Attacks the ability of a server to respond to future connection requests by overflowing the tables used to manage them
- Thus, legitimate users are denied access to the server
- Hence an attack on system resources, specifically the network handling code in the operating system



SYN spoofing (TCP/IP) – 2/2

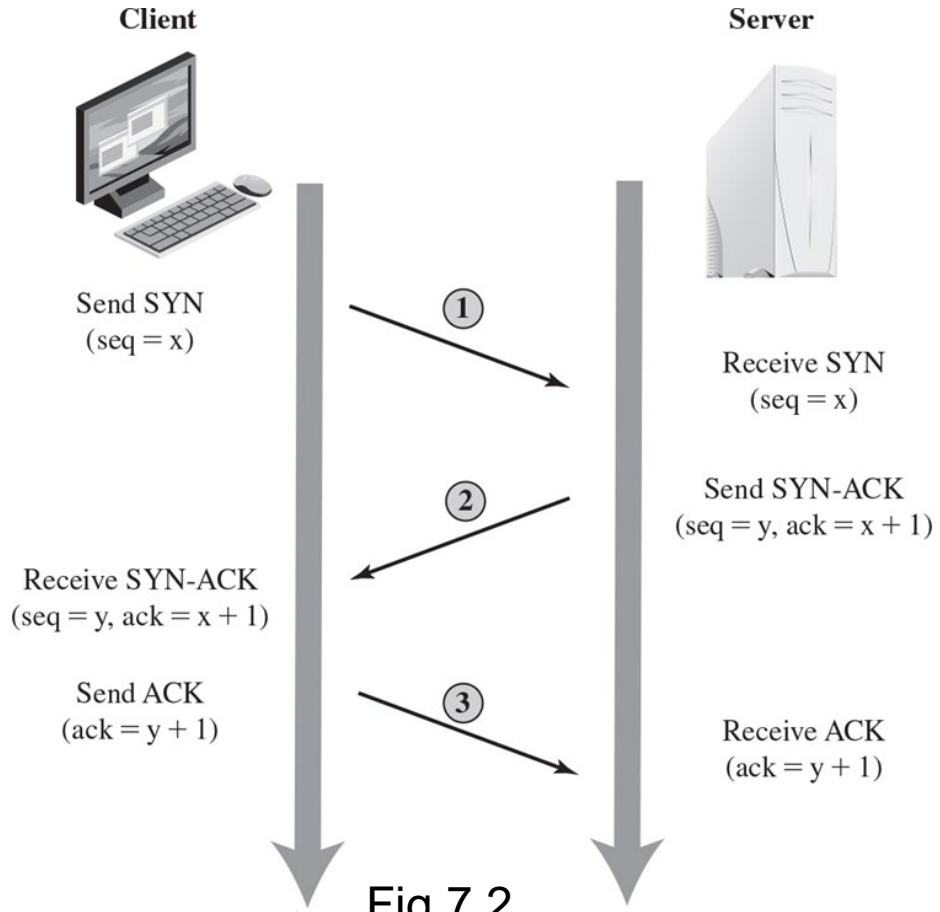


Fig 7.2

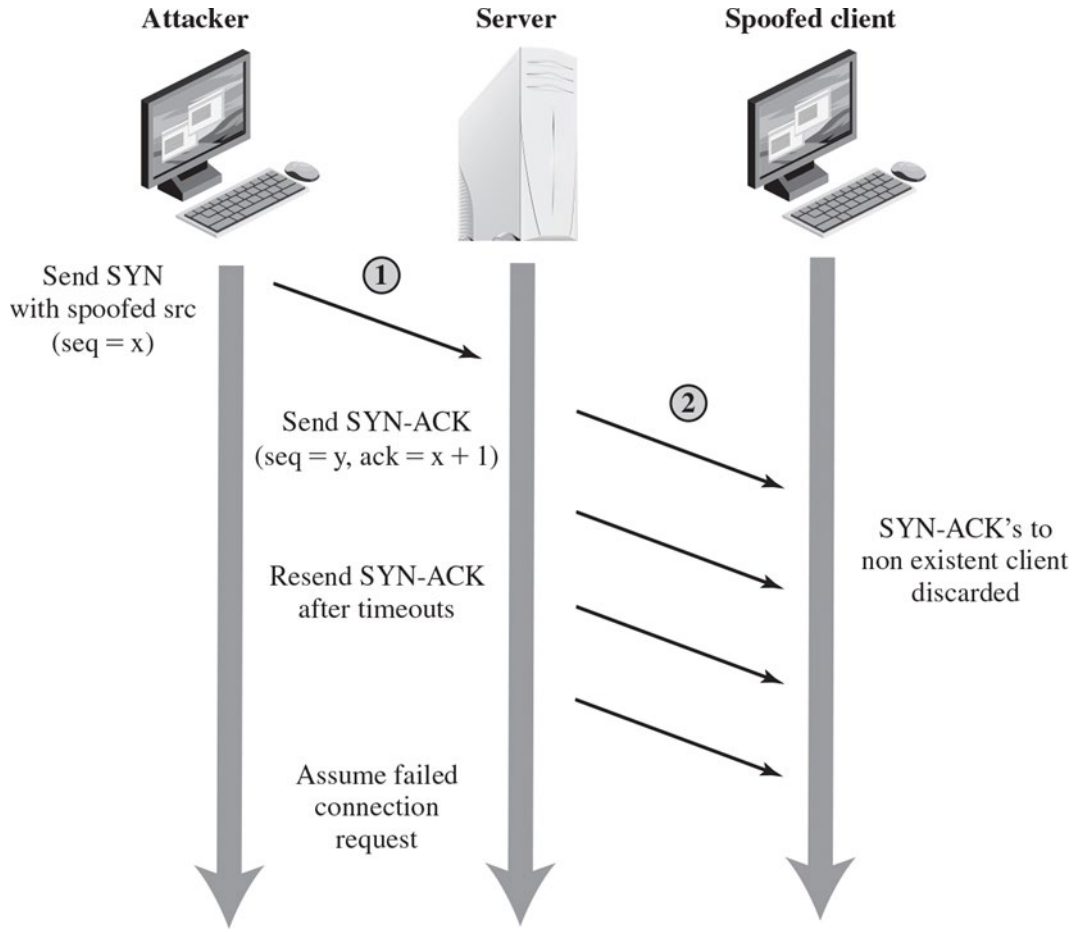
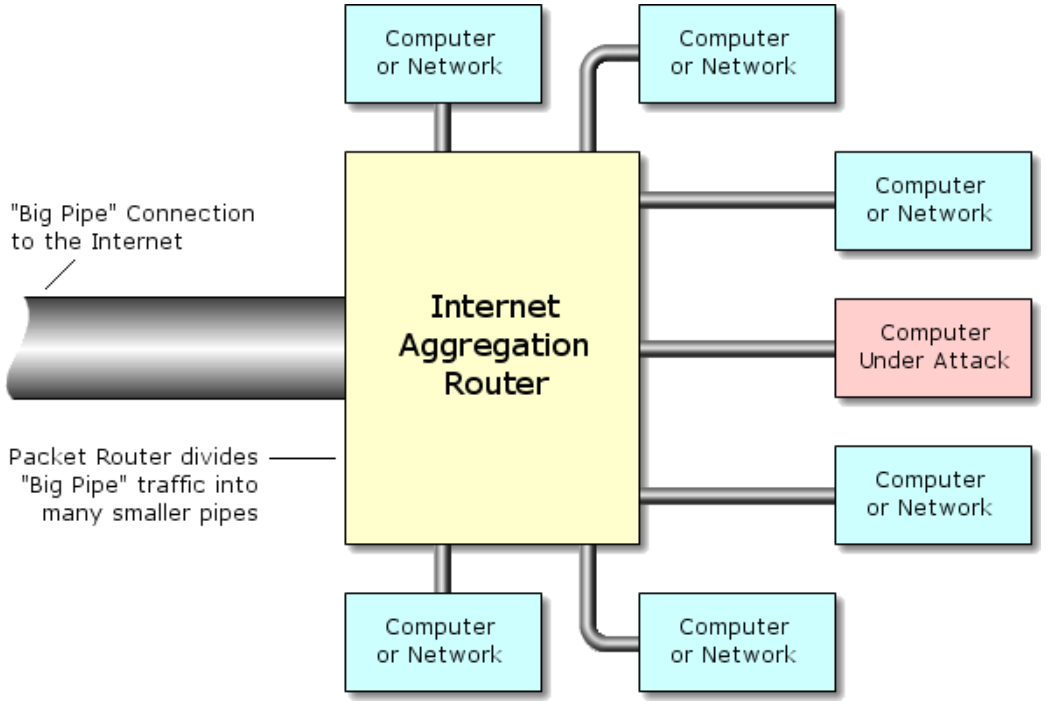


Fig 7.3

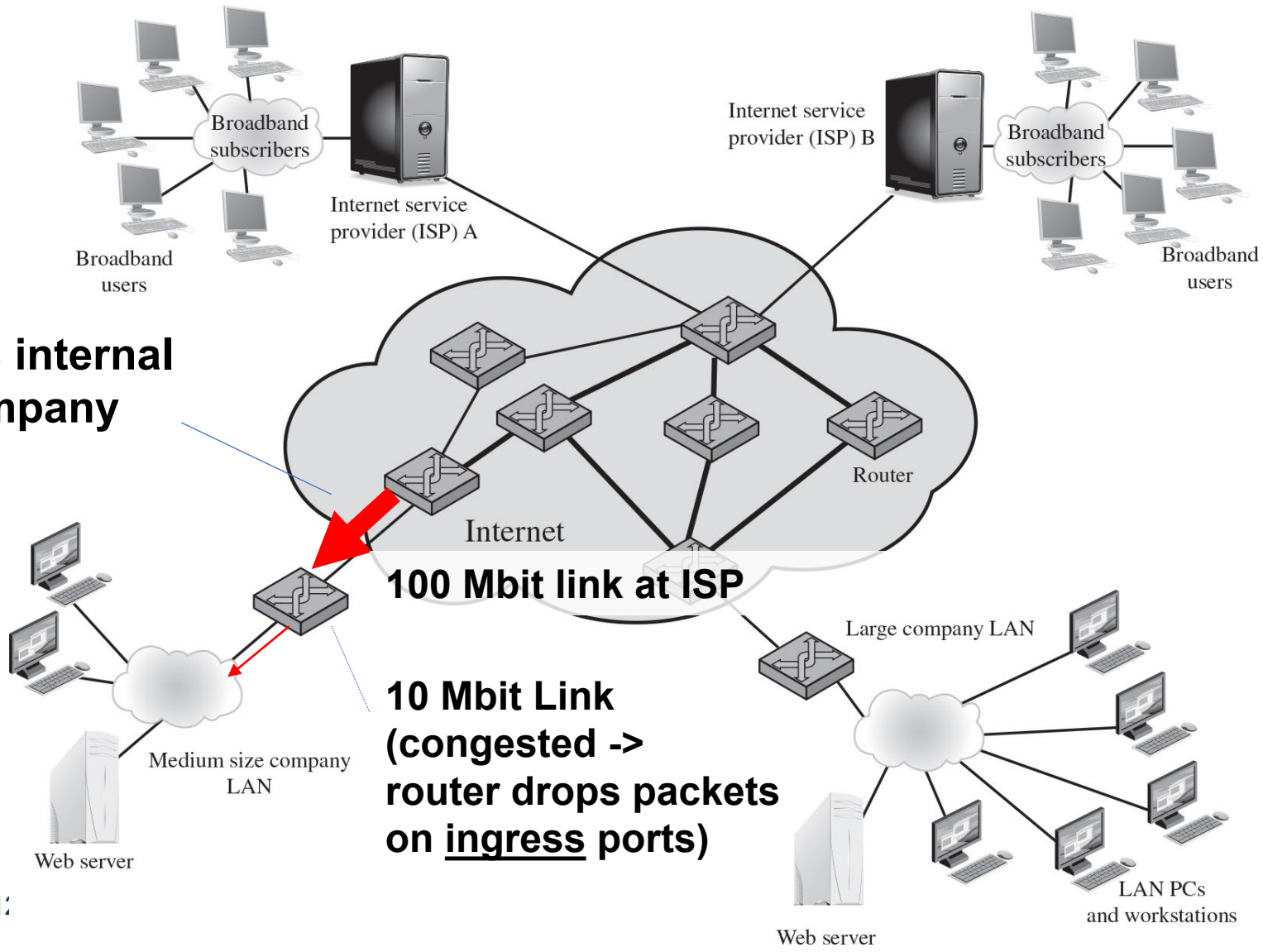
Flooding attacks: ICMP, SYN and UDP flooding

- Can use any kind of protocol to generate a high volume of traffic towards the victim IP address
- They are all based on the principle of congesting the destination network link



ICMP flooding – Network Bandwidth congestion

flooding
packets towards internal machines of company from outside (e.g., servers)



Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination.
Packet loss is either caused by errors in data transmission, typically, across wireless networks, or network congestion.

To avoid congestion, routers simply ***drop*** excess of packets arriving at the overloaded port.

ICMP flooding - case 1: single system generated with real source IP

- One source sends a large volume of ICMP messages from high-capacity links towards smaller capacity links:
  - E.g., ping to a small medium company server
  - Additional messages on the congested link will be discarded, including regular user messages
- Using a real source IP is not a smart move (for attackers) since:
  - The victim can take legal action if corresponding to known entity
  - The victim can stop ICMP traffic from sender source or network
  - The source will receive a flood of ICMP replies
    - However, it will likely survive since has high bandwidth

ICMP flooding – case 2: spoofed source IPs from a single system

- Spoofing: Technique consisting in altering the source address
  - In the case of ICMP, attackers change the source IP address, hence hiding their own
- One attacker generates large volumes of packets that have the target system as the destination IP address but fake source IP addresses
- Source addresses are usually totally random
  - Might correspond to existing addresses or not
  - Reply is sent to random addresses, but if the initial fake source is not existing, the victim will receive also a Destination unreachable ICMP message due to unsuccessful replies
    - Flooding due to 2 kind of messages: initial ICMP Ping + Destination unreachable ICMP
- Defensive strategy:
  - Requires network engineers to specifically query flow information from their routers to stop traffic from inexistent IP addresses -> Requires cooperation between ISP to lock fake egress traffic
  - **Defensive *Backscatter* traffic analysis:** Advertise routes to unused IP addresses to monitor attack traffic. Traffic towards unused IPs must come from attackers.

UDP and SYN Flooding

- Rather than ICMP, the attacker sends UDP packets or TCP packets (SYN packets) to random destination ports on the victim machine
  - Note: SYN packets are only in TCP; UDP uses ports but there is no SYN message
- The victim will not have likely any service running on those ports, but this will take away network bandwidth
  - Also, ICMP destination unreachable or service unreachable messages will be generated on the target machine

SYN spoofing (TCP)

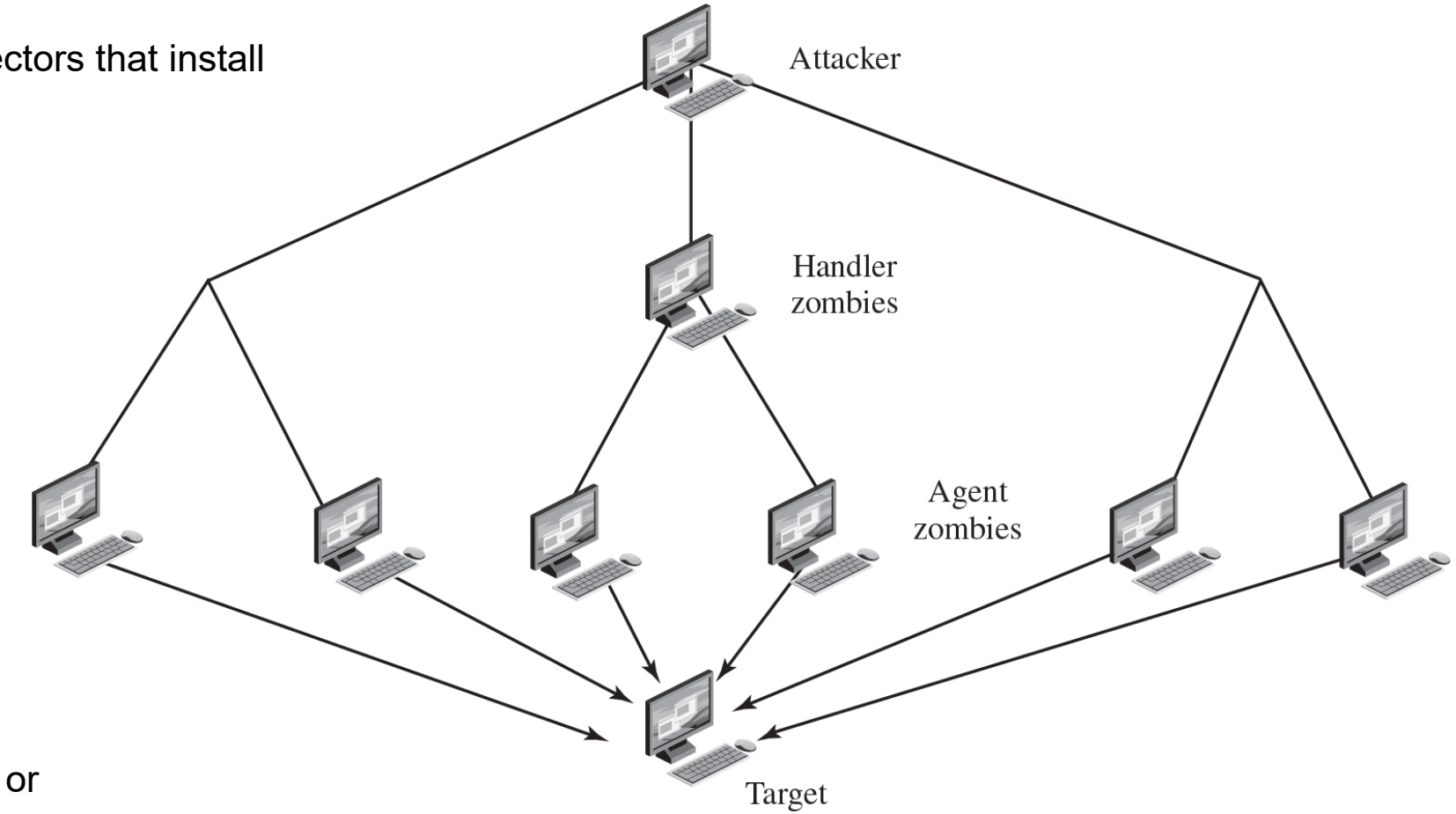
- Service running on target machine (i.e., port open)
- Connection open and server resources exhausted
- Not a flooding attack

SYN flooding (TCP)

- Service not running on target machine (port not open)
- SYN packets are dropped and destination unreachable
- A flooding attack

Distributed Denial of Service – DDoS

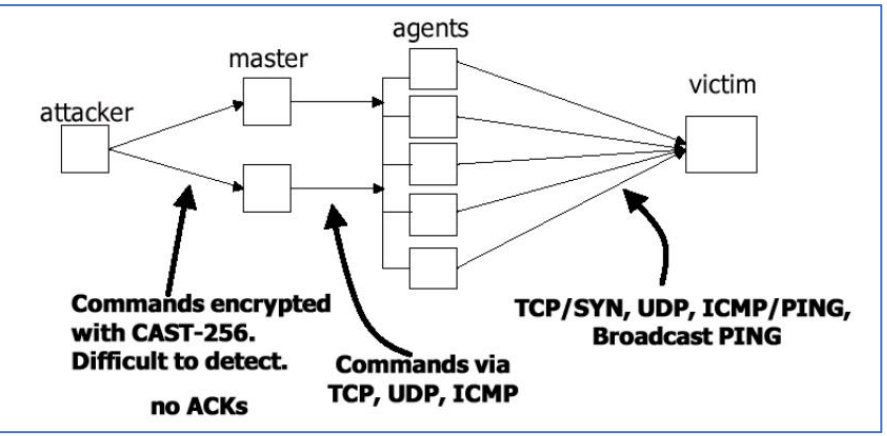
- Use of multiple systems to generate attacks, previously infected somehow
  - Trojans, worms, viruses can be vectors that install or download the zombie agents
- A Botnet is a common strategy to implement the attack
- Each handler controls a subset of zombies
- The attacker simply connects to the handler to control the zombies.
- Handler to zombies connections might be encrypted
- The zombies then execute the flooding or spoofing attack



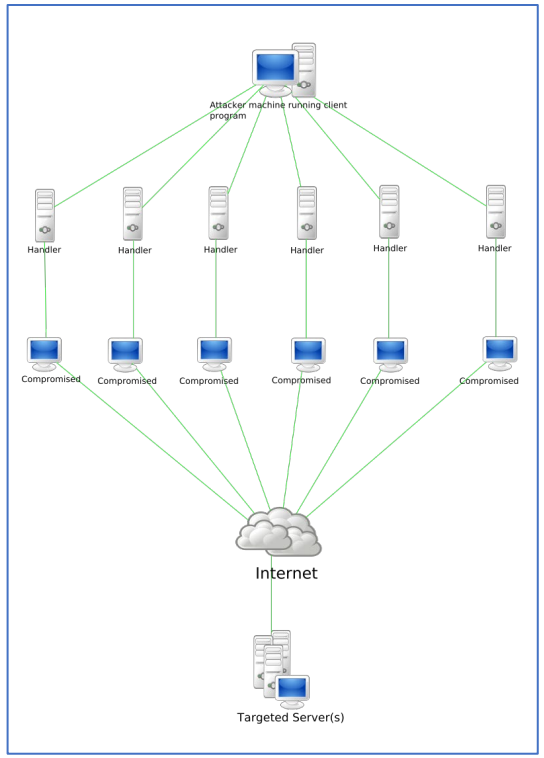
General model of a Botnet, some components might be combined

Examples of DDoS botnets – old tools

Tribe Flood Network 2000 - TFN2k

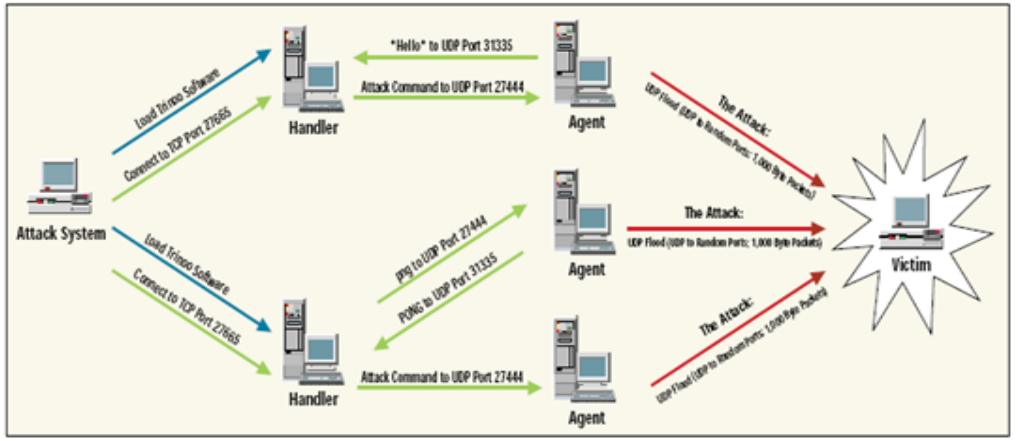


Stacheldraht



Stacheldraht combines the features of Trin00 and TFN and adds encrypted communication between the client and the handler, plus automated remote update of the agents.
Zombies attacks:
Ping flood, UDP flood, TCP SYN flood, and Smurf attack

Trin00

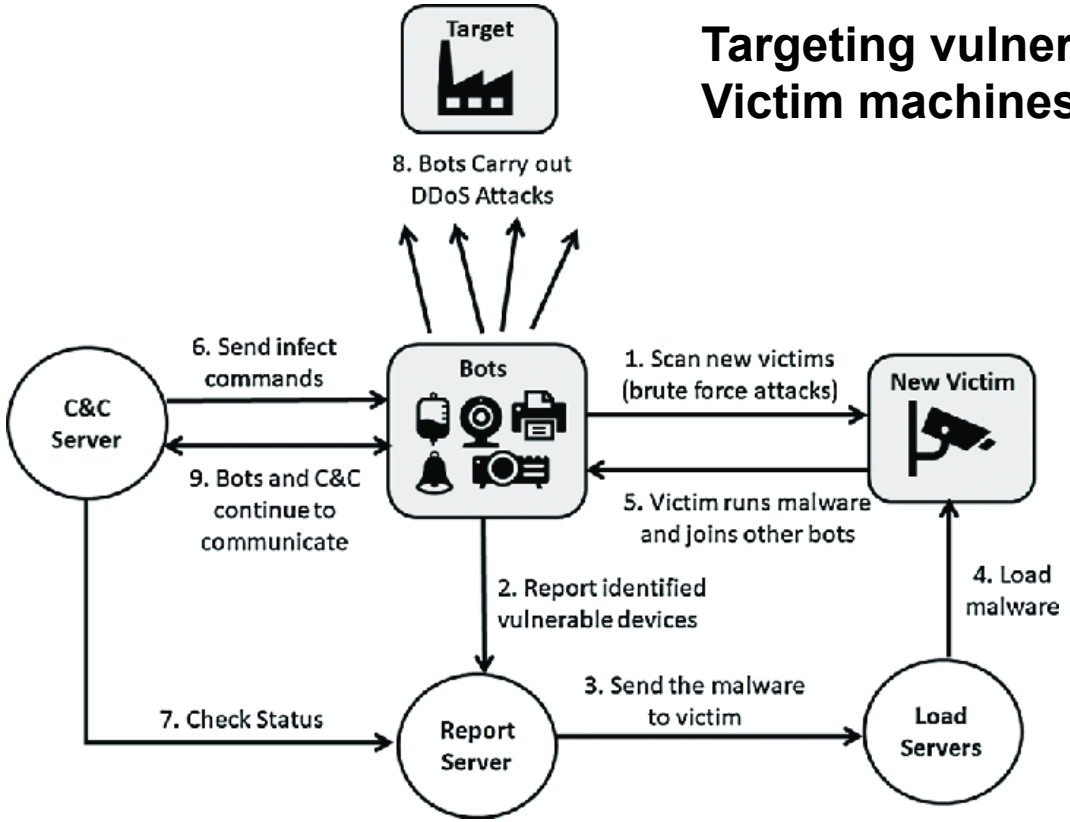


The Trinoo architecture

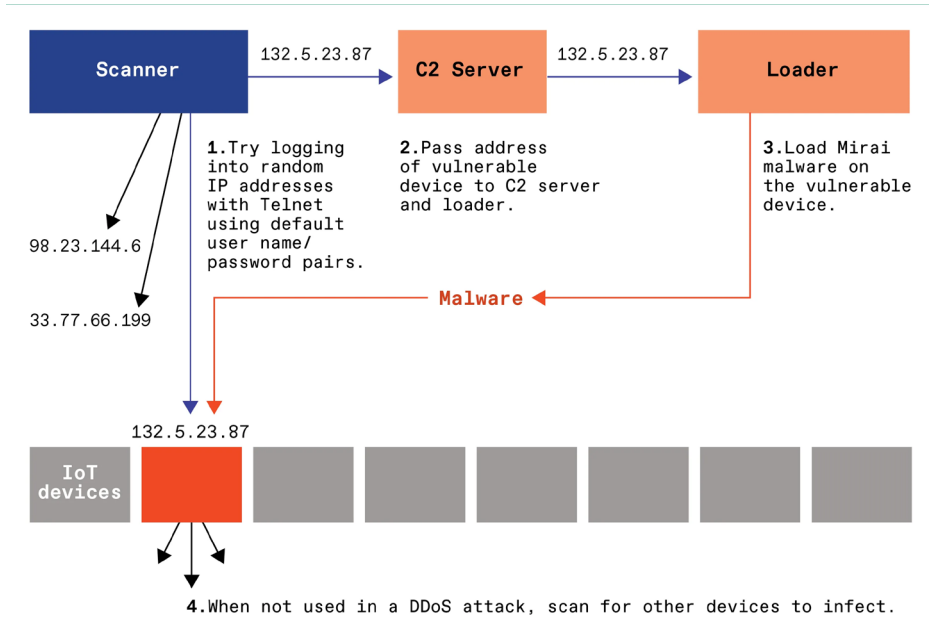
1. Attack the master which will host tools
2. Scan vulnerable machines to be elected as handlers
3. Attack zombies from handlers to install daemons
4. Execute commands from handlers to zombies
5. Zombies perform *UDP flood*

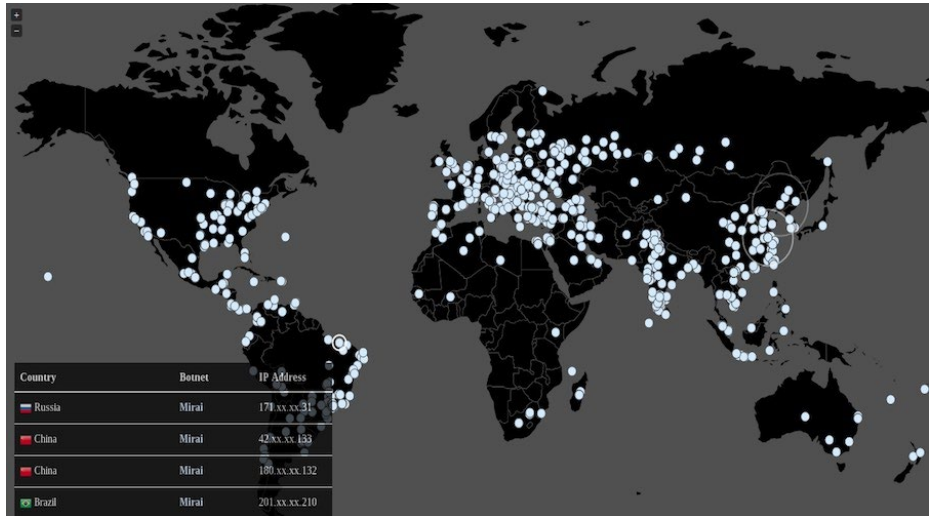
More here: https://digital.library.unt.edu/ark:/67531/metadc739628/m2/1/high_res_d/792253.pdf

Example recent DDoS – Mirai (2016)



Targeting vulnerable IoT.
Victim machines were DNS servers



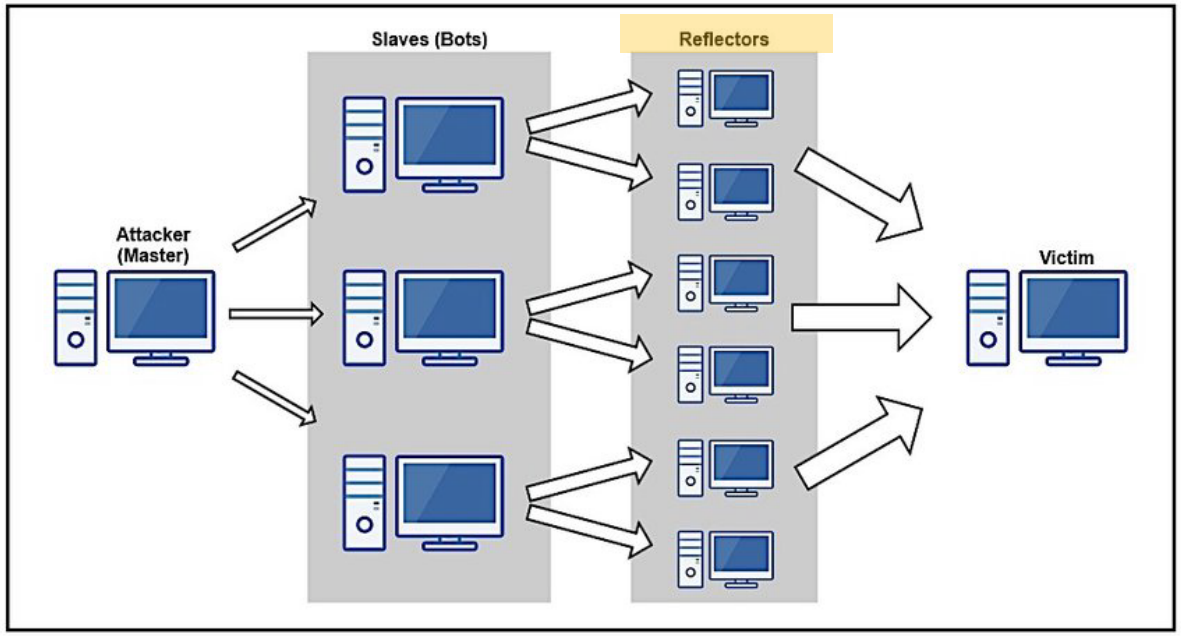


https://www.youtube.com/watch?v=cQkQpbkhzIo&t=9s\
Infection progression

DoS/DDoS - Reflection Attacks

- Attacker sends packets to a known service on the intermediary with a spoofed source address of the actual target system
- When intermediary responds, the response is sent to the target
- "Reflects" the attack off the intermediary (reflector)
  - Reflectors are NOT compromised hosts that simply reply to (spoofed) sender requests
- Goal is to generate enough volumes of packets to flood the link to the target system without alerting the intermediary
  - Very few reflectors are chosen, but with high-capacity bandwidth
- Lack of backscatter traffic: Very hard to detect based on the source
  - The basic defense against these attacks is blocking spoofed-source packets

Spoofed source IP == victim IP



For this attack, main categories of protocols are used to generate replies:

- ICMP messages, like ping
- UDP based protocols, e.g., DSN, SNMP, ISAKMP, NTP, UDP echo
- TCP SYN

UDP Echo protocol

- Similar to ICMP Echo Request and Reply, but in TCP and UDP
- UDP Echo measures round-trip delay.
- UDP Echo tests return Latency and Packet Loss results.
- UDP tests must be configured to target a destination port that is listening for UDP traffic. For example, many Cisco devices running IOS IP SLAs use port 1967 and UNIX systems use port 7. Failure to configure the port can result in timeouts for UDP tests. Set the Destination Port parameter in the General options for UDP tests.
- Existing also in TCP

```
nc -uvn <IP> 7
Hello echo    #This is what you send
Hello echo    #This is the response
```

Example: DNS Reflection Attack with UDP Echo

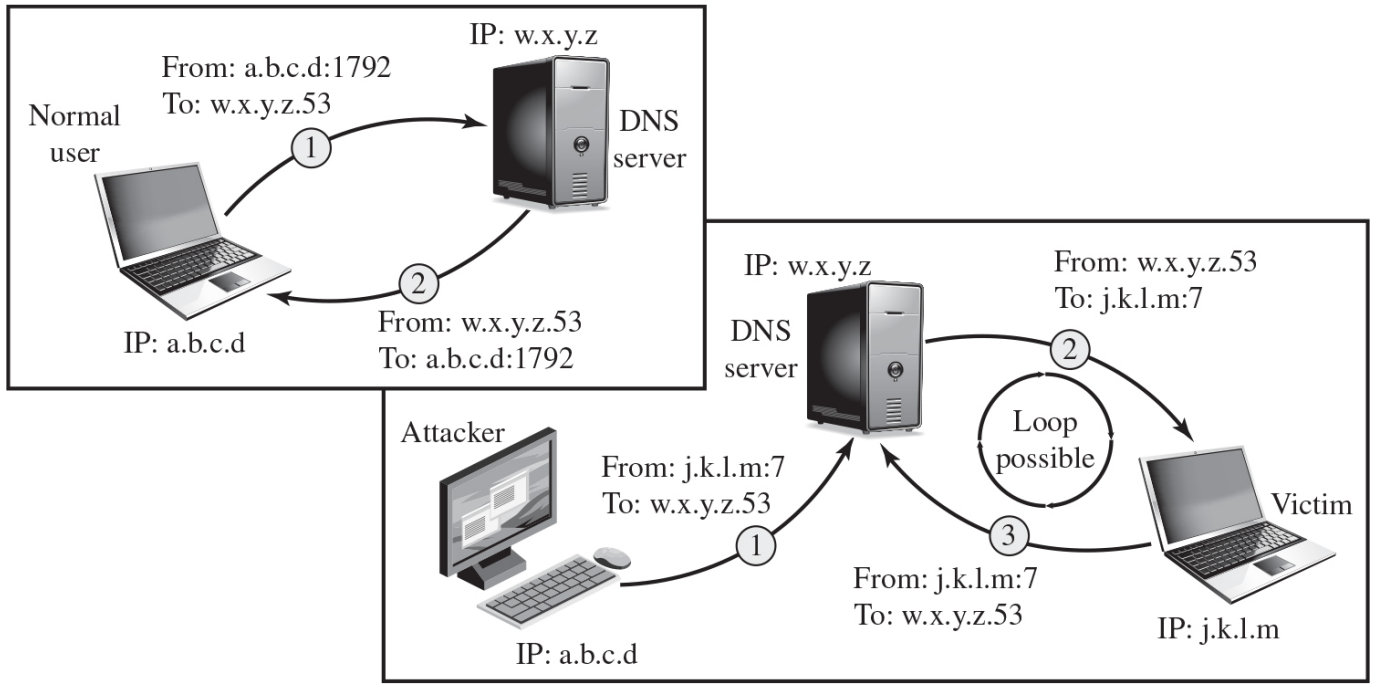
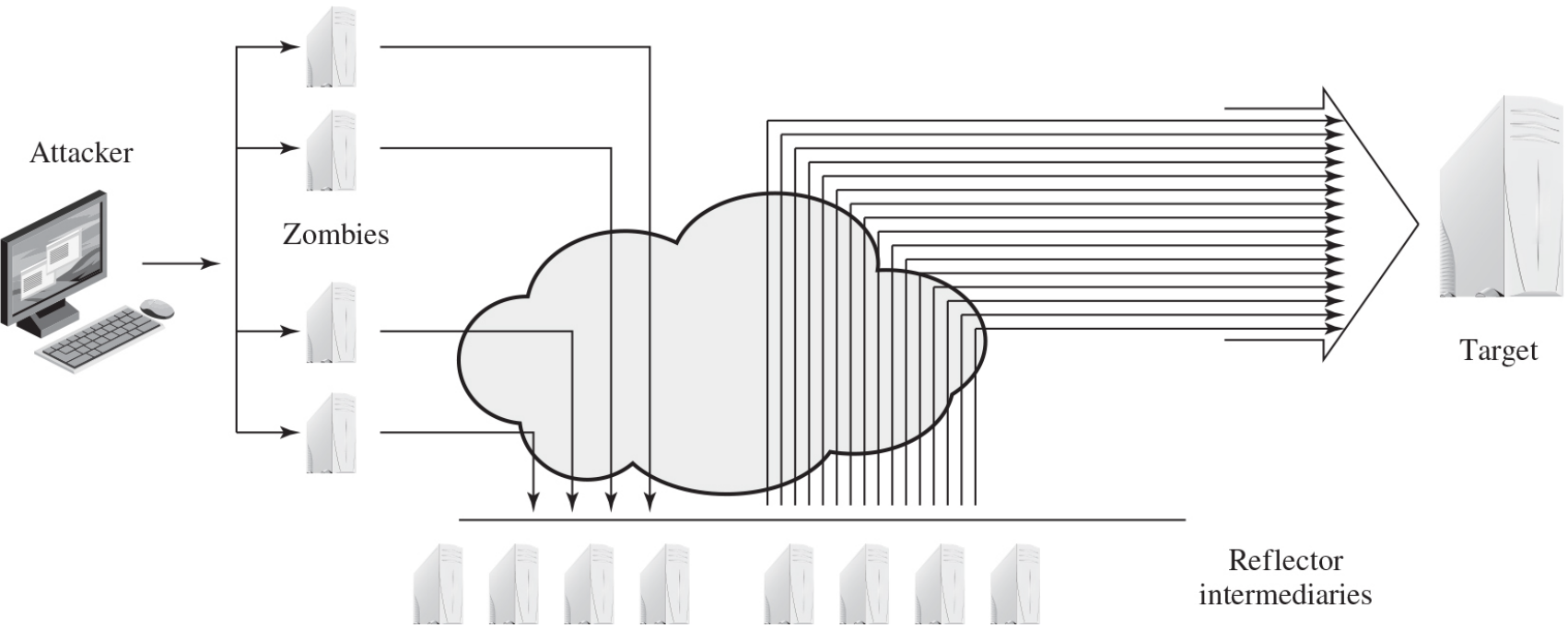


Figure 7.6

Attack Model:

- Spoofed DNS request (over UDP protocol) from attacker towards a DSN server (reflector):
  Source IP = Victim IP
  Source Port = 7
- DNS server will reply to Victim IP and send the UDP message to Port 7 of Victim
- If Victim has UDP service activated, then victim will reply to the UDP Echo, by sending another UDP Echo to DNS server 7
- This activates an infinite loop
- Can be stopped:
  - Disable UDP echo from external networks on any machines
  - Detect DNS (UDP) requests with 7 as a source port

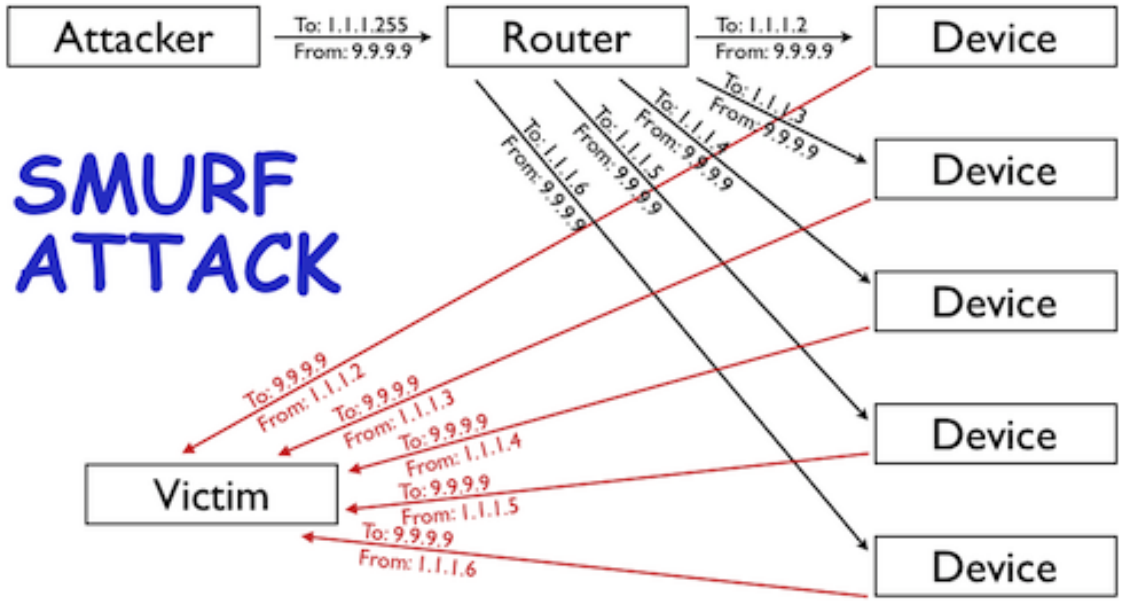
DoS/DDoS - Amplification Attacks



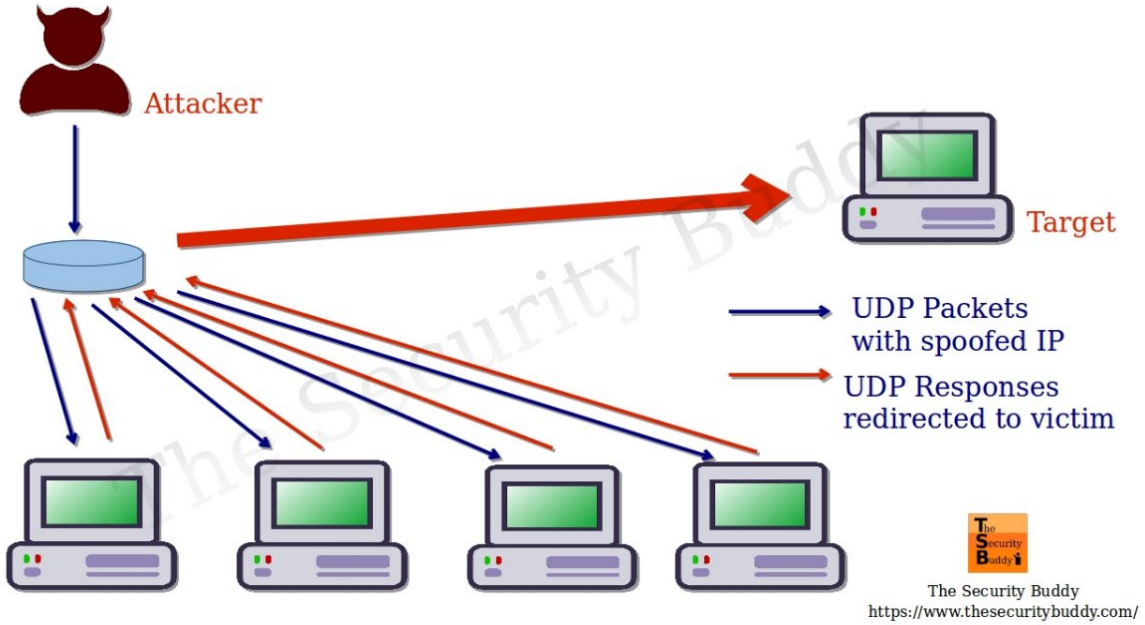
The basic mechanisms are:

1. send a limited number of messages in broadcast: since the receivers are many, the reflection will come from many machines
2. Another approach to amplify the reflection, is to send small amount of bytes and trigger large replies from the reflectors to the victim/target.
3. Since it's based on broadcast, it's not possible in TCP, only UPD and ICMP (Broadcast messages are always connectionless)

Example of DDoS Amplification: Smurf and Fraggle attacks



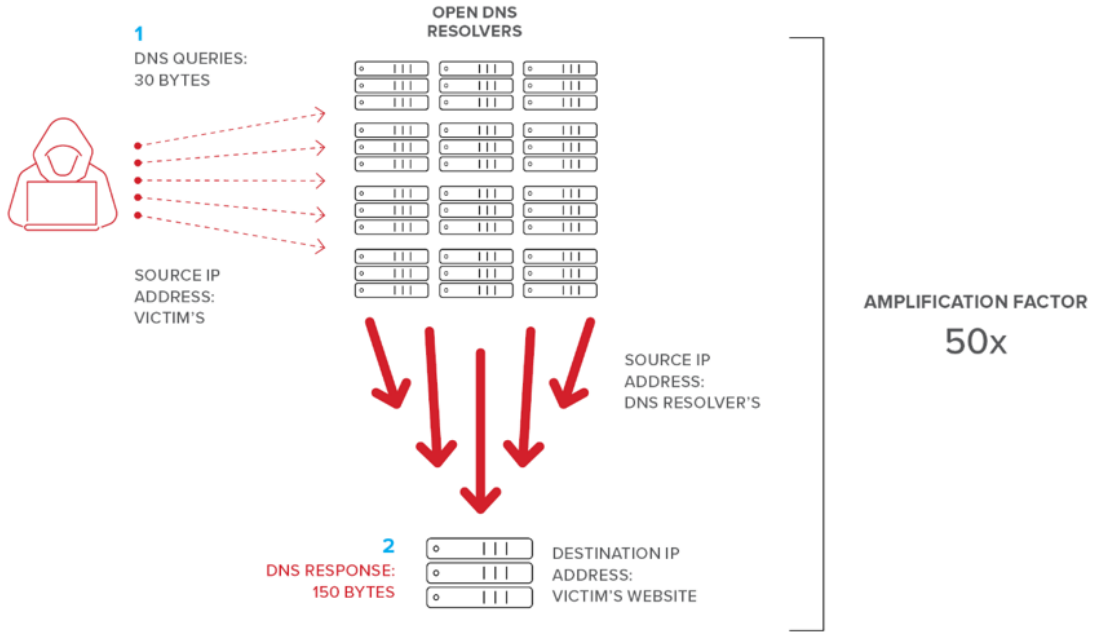
- Smurf attack sends ICMP with unicast (victim) source and broadcast as a destination to reach the reflectors
- Can also be the other way around: broadcast source, unicast reflectors



Fraggle is like Smurf but using UDP Echo as a message request and reply (still IP broadcast used as a destination)

Example of DDoS Amplification: DNS Amplification Attacks

- Use packets directed at a legitimate DNS server as the intermediary system
- Attacker creates a series of DNS requests containing the spoofed source address of the target system
- Exploit DNS behavior to convert a small request to a much larger response (amplification)
- Target is flooded with responses
- Basic defense against this attack is to prevent the use of spoofed source addresses

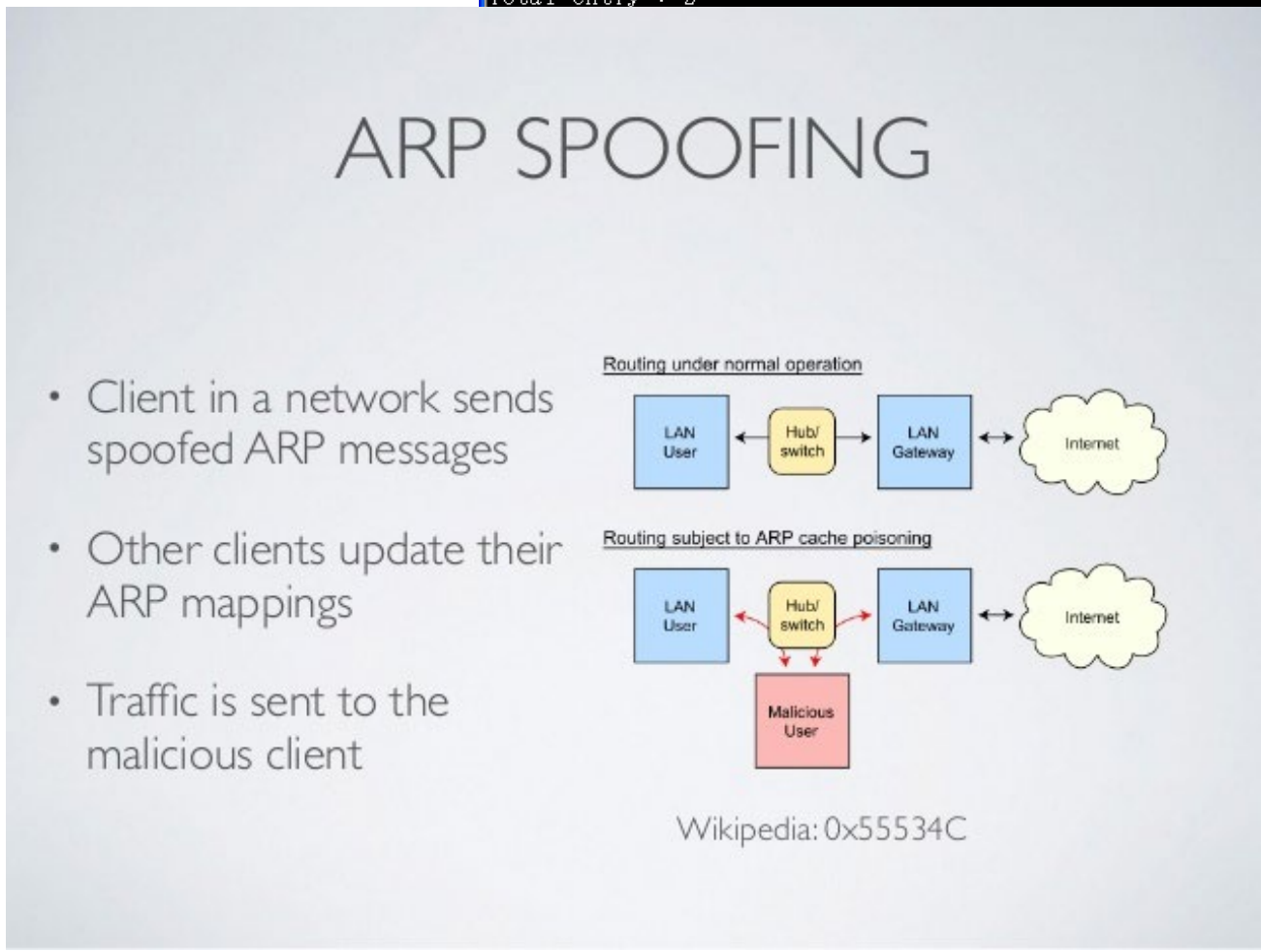


https://cyberhoot.com/cybrary/dns-reflection-attack/

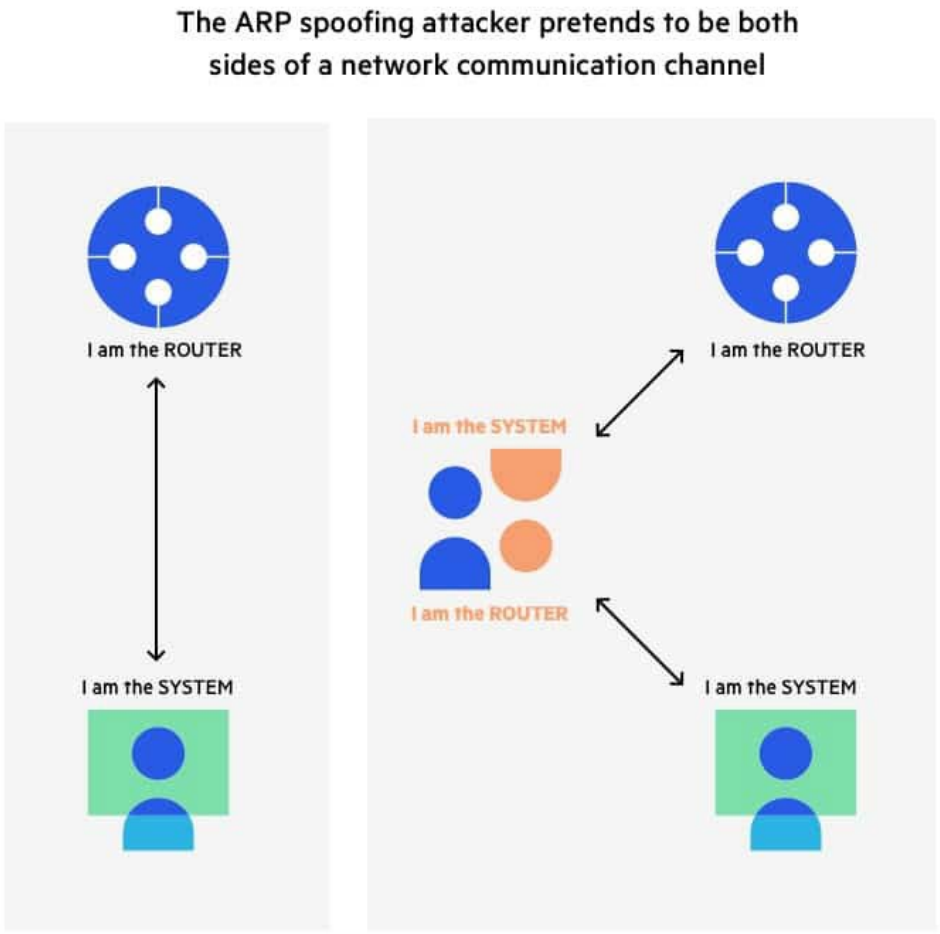
ARP spoofing/ARP poisoning (1/2)

```
Console#show arp
ARP Cache Timeout: 1200 (seconds)

IP Address      MAC Address       Type      Interface
--------------- ----------------- --------- -----------
192.168.1.20    CC-5D-4E-39-8A-A2 dynamic   VLAN 1
192.168.1.100   E0-CB-4E-E8-F3-6D dynamic   VLAN 1

Total entry : 2
```

- ARP protocol is responsible for resolving an IP address to a Mac address within a local network in order for devices to communicate with each other.
- Remember: once a packet travels or arrives to a Local Area Network, only MAC addresses are used to send the messages from/to routers from/to individual hosts on the network
- ARP spoofing is a hacking technique where an attacker can poison the ARP cache (which holds temporary associations IP-MACs) on other devices, for example, on the switches.
- ARP Spoofing alters the association IP-MAC address inside a local area network (LAN):
  - The attacker declares to have the MAC address associated to the victim IP, for example during an ARP request (aka, *gratuitous poisoning*)



ARP spoofing/ARP poisoning (2/2): Man-In-The-Middle (MITM) example



The attacker can advertise ARP responses to have:

- the MAC address associated to the router IP
- the MAC address associated to the victim IP

If the attacker declares to be the router, it can intercept all the messages towards the router (usually packets that leave the LAN).

If the attacker declares also to have the MAC of an internal victim host, it can intercept all the messages for the victim.

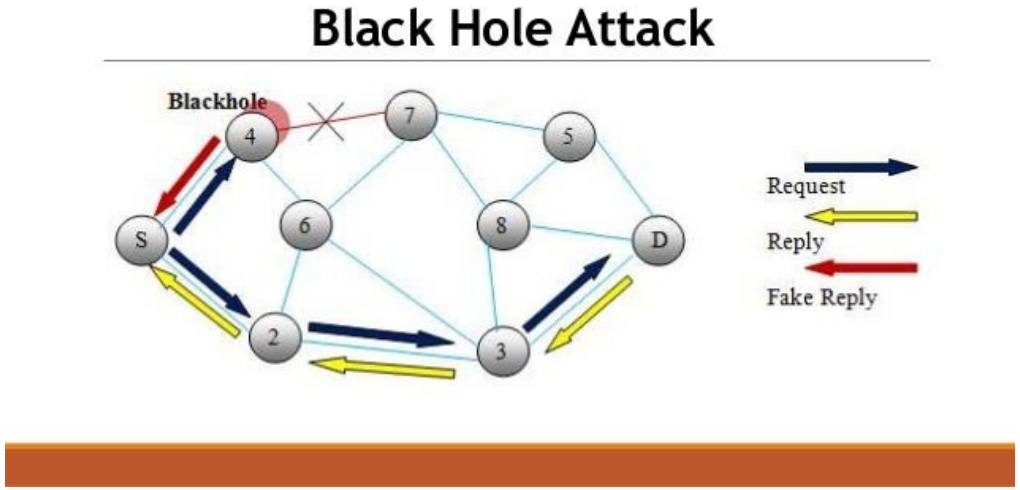
Perfect example of how to implement a **Man-In-The-Middle (MITM)** attack in a Local Area Network.

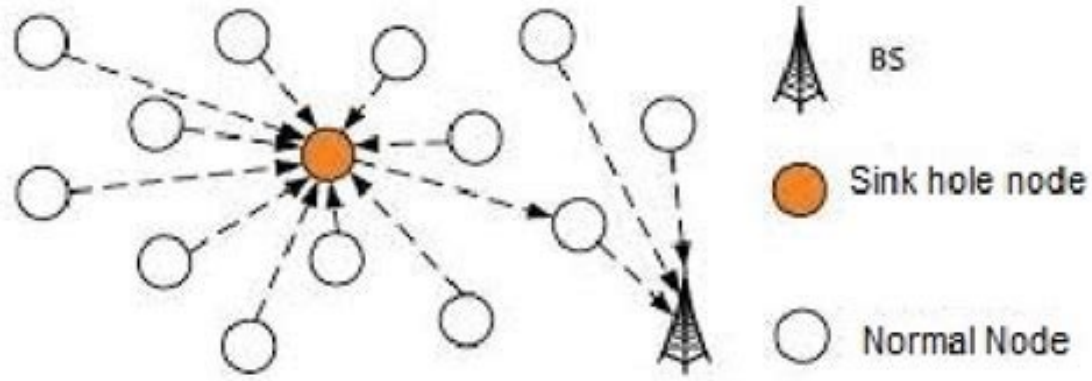
What the attacker can do now on:

1. Sniff the packets and steal cleartext data
2. Perform session hijacking—if the attacker obtains a session ID, they can gain access to accounts the user is currently logged into.
3. Alter communication—for example pushing a malicious file or website to the workstation.
4. DDoS - an attacker might send out ARP Response messages that falsely map hundreds or even thousands of IP addresses to a single MAC address, potentially overwhelming the target machine.
5. Blackhole DoS attack – attacker might simply drop packets for the victim, hence interrupting services, especially if the victim might is a server.

DoS in mobile ad hoc networks – Blackhole and Sinkhole

- In wireless networks some protocols (e.g., Zigbee) build routing paths based on quality of wireless paths and hop counts (e.g., AODV routing protocol)
- By advertising fake routes or less hop counts, a malicious node will force an attacker to send all data to the malicious node that can discard all the packets

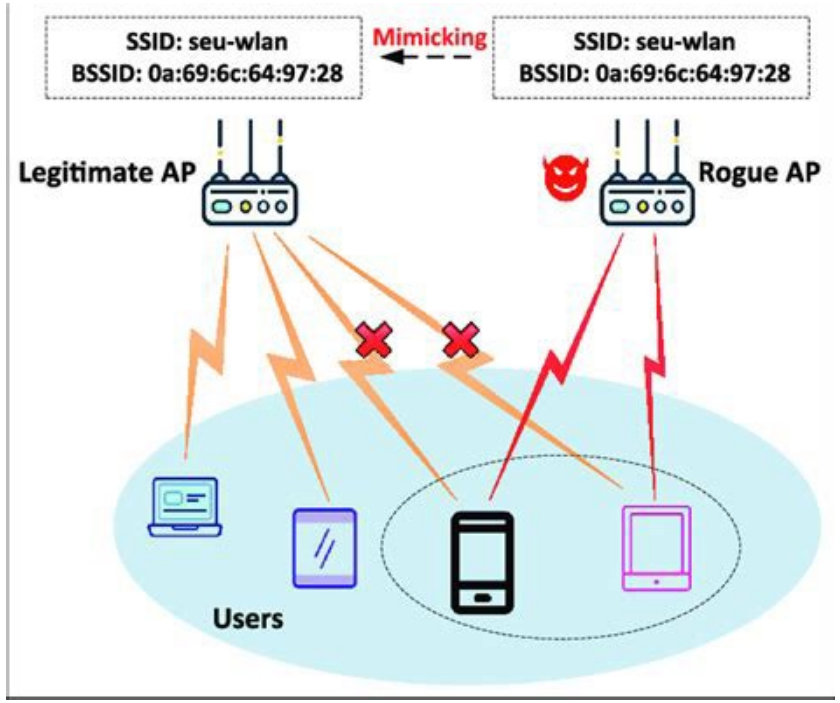


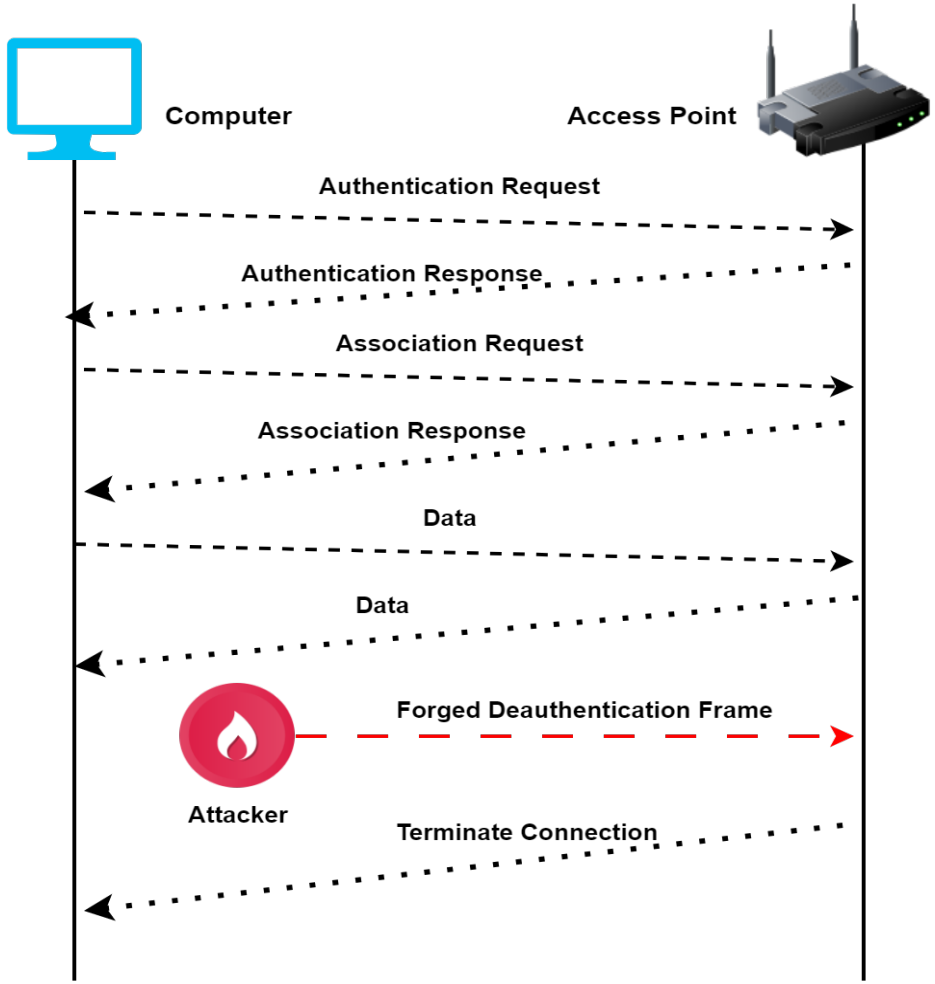


Normally, all paths reach the BS - Base station
During the attack, all traffic goes to the Sinkhole which can discard packets or perform Man in the Middle

DoS in WiFi – Rogue AP and Disassociation attacks

Rogue AP can act as the legitimate AP
and disconnect or drop packets to the victims.
Note: BSSID is the wifi MAC of the AP





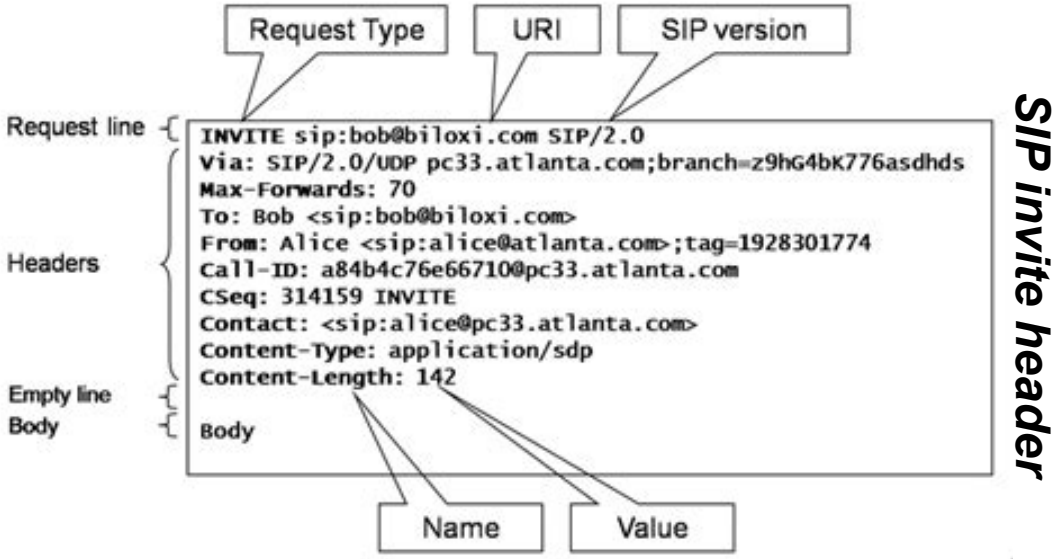
Disassociation attack performs a MAC spoofing of victim machine, and performs a request of disassociation to the AP, hence disconnecting the victim from the wireless network

Application-level DoS attacks

- In theory, DNS attacks are also Application-level attacks since DNS is an application using UDP (well-known port 53)
  - This attack aims at consuming network bandwidth toward the victim
- On the other hand, application-level DoS attacks are meant to exhaust some mechanisms in the application to generate unproductive operation of a server
- Examples:
  - Huge **amount of invitations** in voice and conferencing applications: SIP invite flood
  - **Heavy requests** from clients, which will end up being overloaded **in the backend servers:**
    - Cyberslam, Slowris, HTTP flood
  - Generate a **crash** in the backend server**:** SIP Invite of death, RegularExpressionDoS (ReDoS)

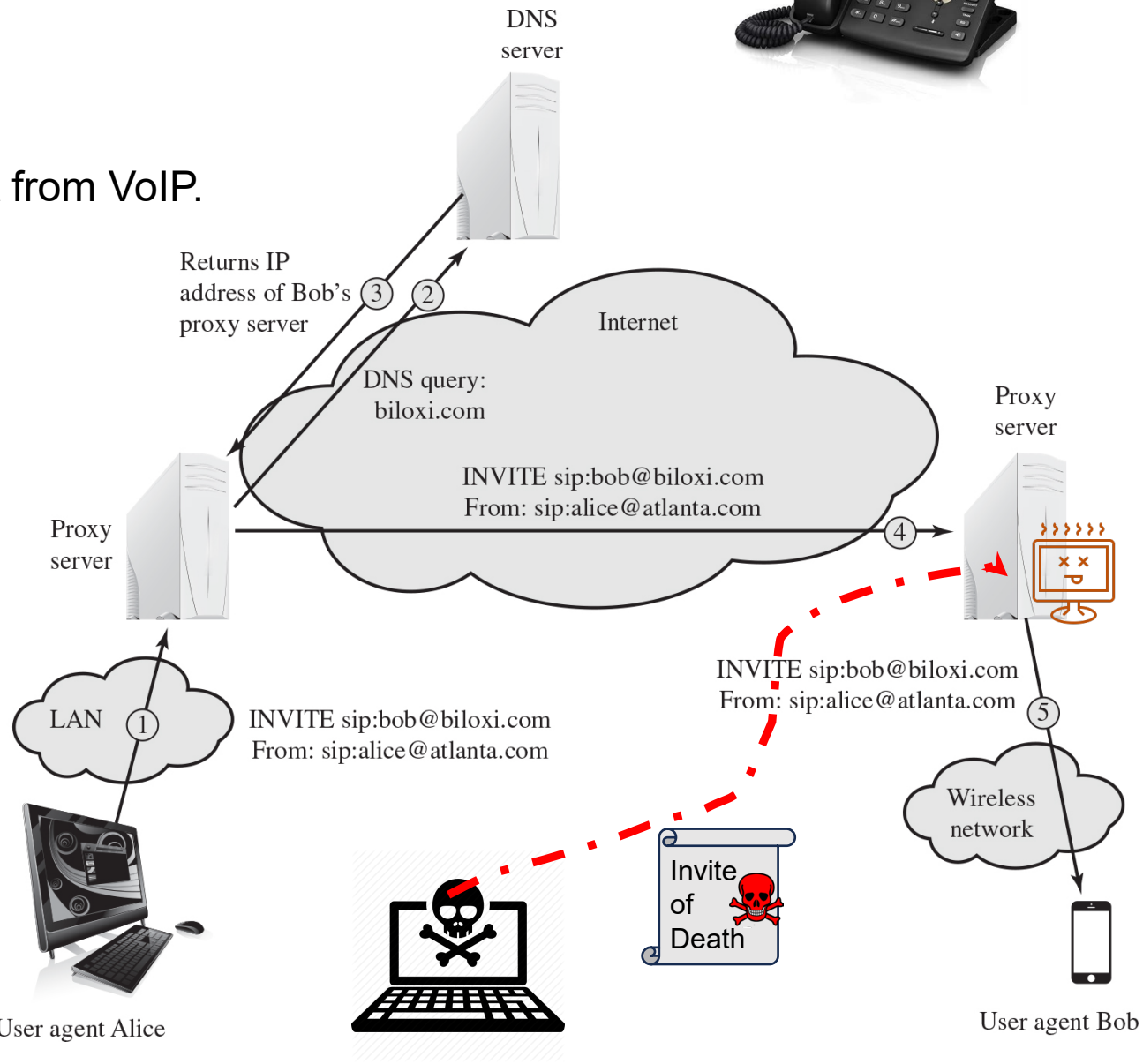
Example of Application DoS - SIP INVITE-of-Death

SIP is a protocol for telephony over IP, usually different from VoIP.



Invite of death:

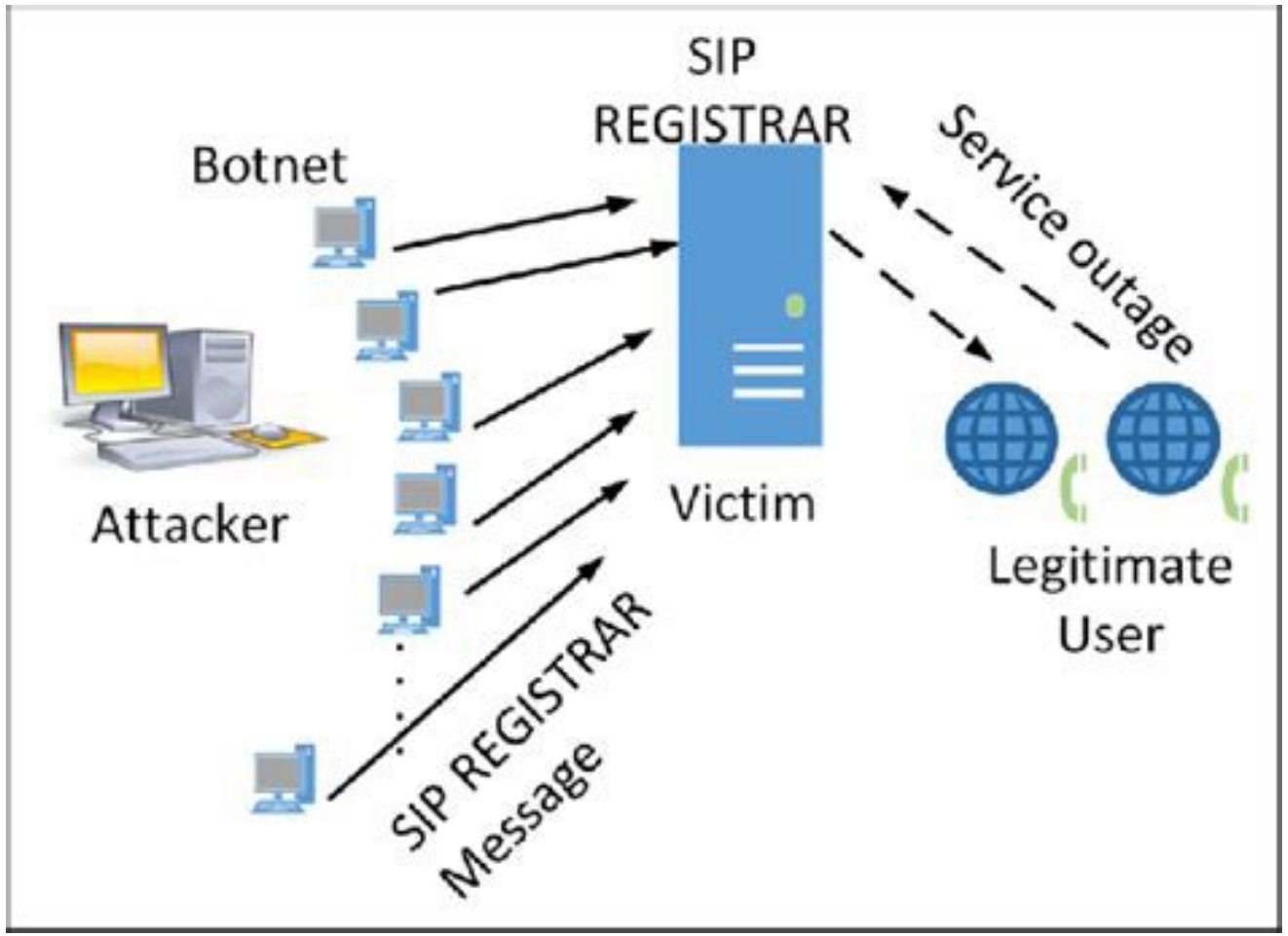
Via (in the header) contains an extraneous number of colon (:) characters (i.e., tokens); it causes a vulnerable OpenSBC server to crash.



Example App DoS: SIP Invite flood

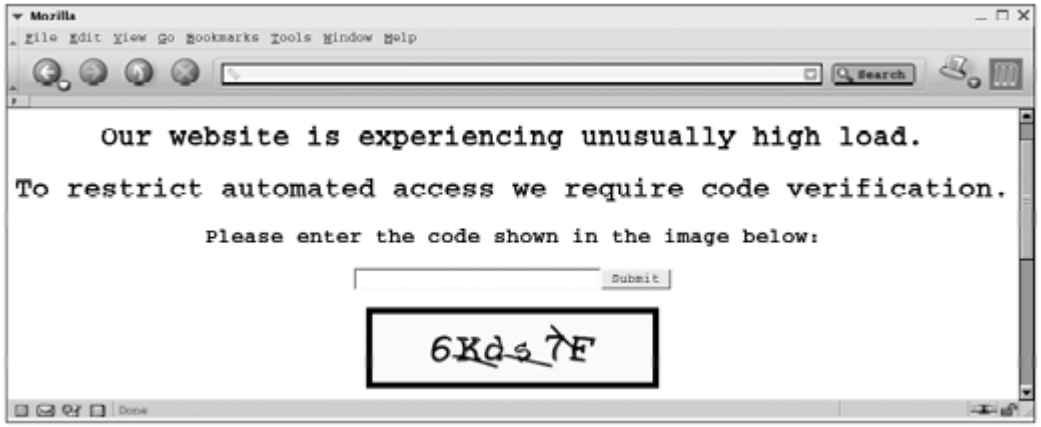
Either SIP Invite or
SIP Registrar messages

Effect:
- Server crashes
- Network might congest
- User receives a zillion calls



Hypertext Transfer Protocol (HTTP) Based Attacks – CyberSlam

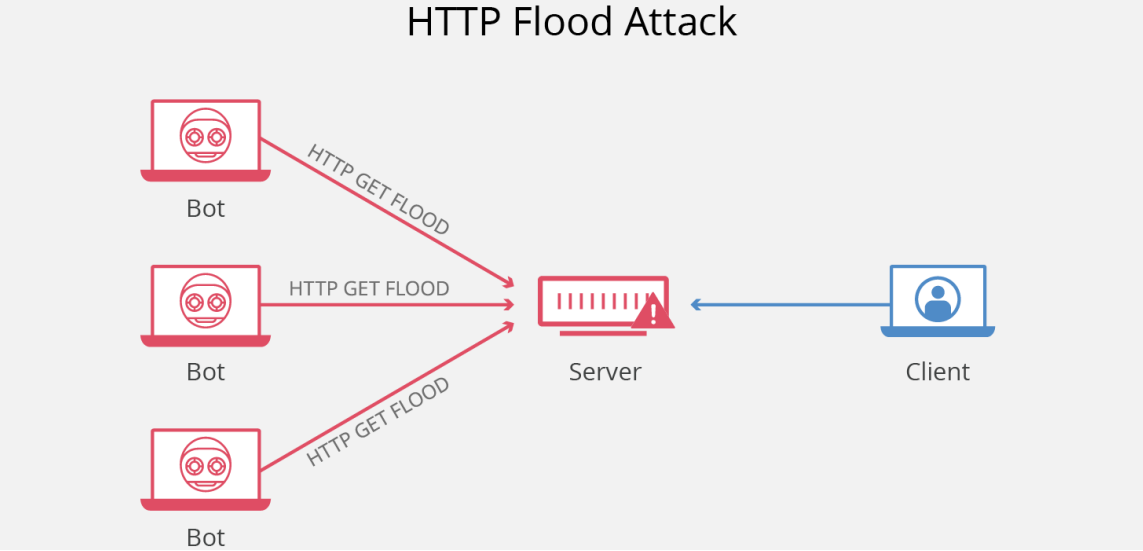
- Mimic legitimate Web browsing behavior and consume higher layer server resources such as CPU, memory, database and diskband width.
- Small amount of requests which take busy a WebServer towards the Databases with stressful queries
  - Recent FBI case from a web site competitor against another
- From the Web server's perspective, these zombie requests (from a botnet) look exactly like legitimate requests, so the server ends up spending a lot of its time serving the zombies, causing legitimate users to be denied service.



Potential solution: Smart Captcha sent only to human-like patterns

- Note: using captcha to query bots might not be enough to kill them, as a server might consume resources to generate thousands captcha

HTTP flood and recursive HTTP flood

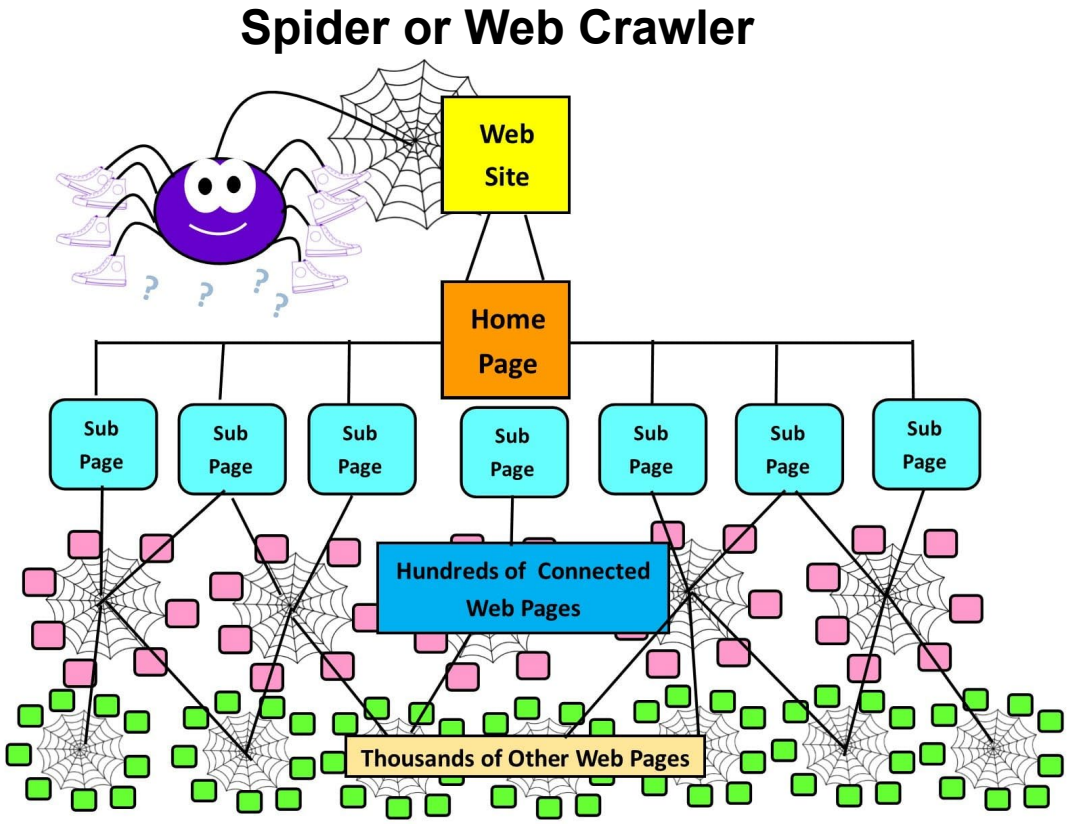


Based on HTTP GET and POST message types sent from a botnet.

Recursive variant is HTTP request sent to all the URLs linked inside the initial page

note

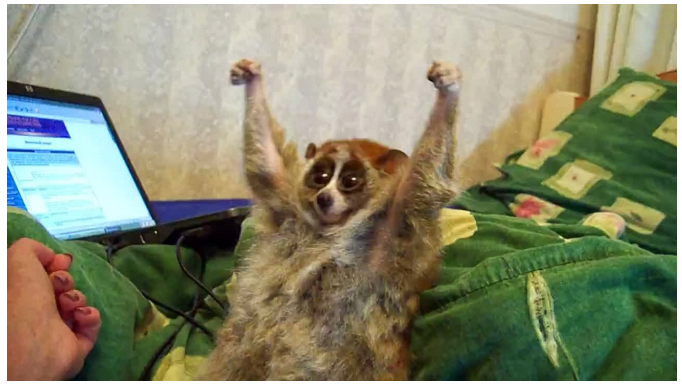
Example of hacker/stresser tools:
Low Orbit Ion Cannon (LOIC), High Orbit Ion Cannon (HOIC)



A WebCrowler or Spider is a bot that extracts all the http links (hyperlinks) in a starting website (HTTP seed), with the intent to download all of them, but not to flood them in a DoS

Hypertext Transfer Protocol (HTTP) Based Attacks – Slowris

- Attempts to monopolize by sending HTTP requests that never complete
- Eventually consumes Web server's connection capacity
- Utilizes legitimate HTTP traffic
- Existing intrusion detection and prevention solutions that rely on signatures to detect attacks will generally not recognize Slowloris



Stream Content

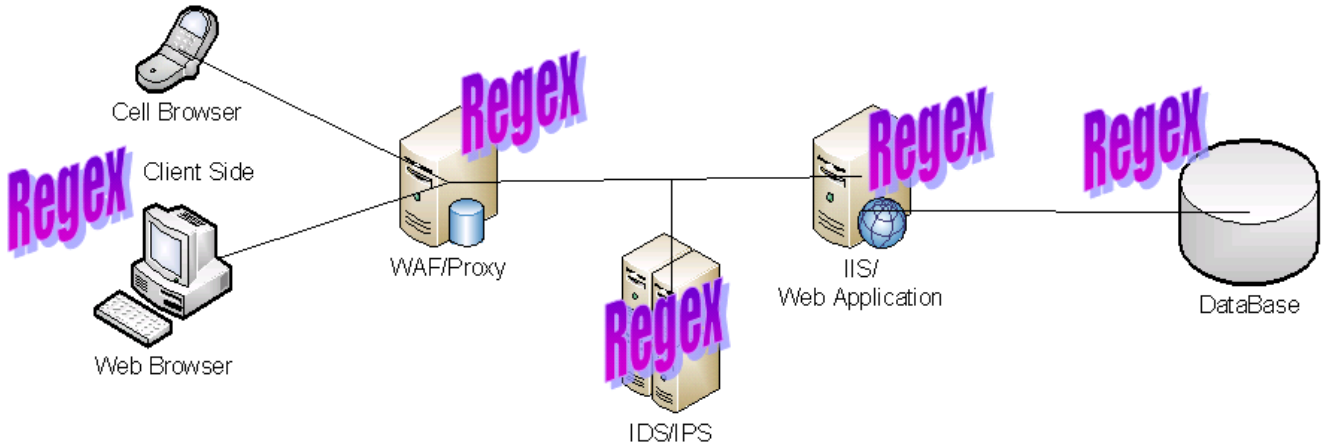
```
GET /?67128010893156 HTTP/1.1
Host: bbc.com
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR
1.1.4322; .NET CLR 2.0.503l3; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)
Content-Length: 42
X-a: b
X-a: b
X-a: b
X-a: b
```

https://kb.mazebolt.com/knowledgebase/slowloris-attack/

Complexity DoS - Regular expression denial of service (ReDoS)

- The Regular expression Denial of Service (ReDoS) is a Denial of Service attack, that exploits the fact that most Regular Expression implementations may reach extreme situations that cause them to work very slowly (exponentially related to input size). An attacker can then cause a program using a Regular Expression (Regex) to enter these extreme situations and then hang for a very long time.

A Regex pattern is called **Evil Regex** if it can get stuck on crafted input.



Components that might contain Evil Regex can get hit by a sequence that hangs the node (e.g., a web server or an IDS)

Examples of Evil Regex:

(a+)+
([a-zA-Z]+)*
(a|aa)+
(a|a?)+
(.*a){x} for x \> 10

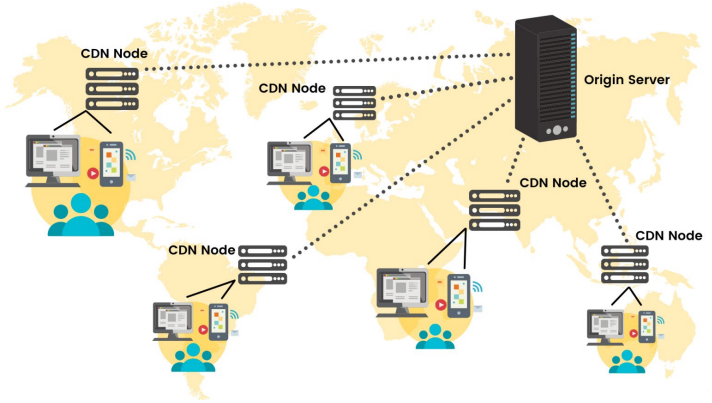
All the above are susceptible to the input aaaaaaaaaaaaaaaaaaaaaaa!

Defensive measures against DoS

CalPolyPomona

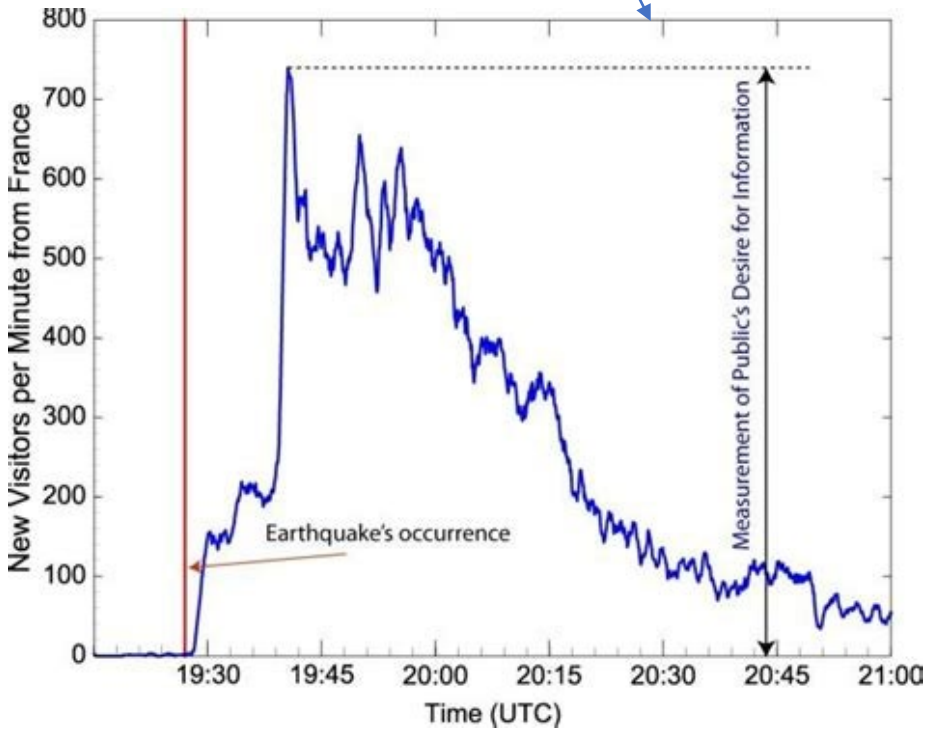
Observation: Flashcrowd events

- High traffic volumes may be legitimate
  - High publicity about a specific site
  - Activity on a very popular site
  - Described as **slashdotted, flash crowd, or flash event**
  - Solutions are **Load Balancers** in datacenters
    And **Content Delivery Networks (CDN)** (e.g., Akamai)



Example:
- at 7 PM Amazon will offer product 50% of price on a specific day
- world cup final free to watch at website X
- some natural/human related event occurring, and everybody connects to the most reliable info source

Access to the website of EMSC - European-Mediterranean Seismological Centre after an earthquake somewhere in Europe



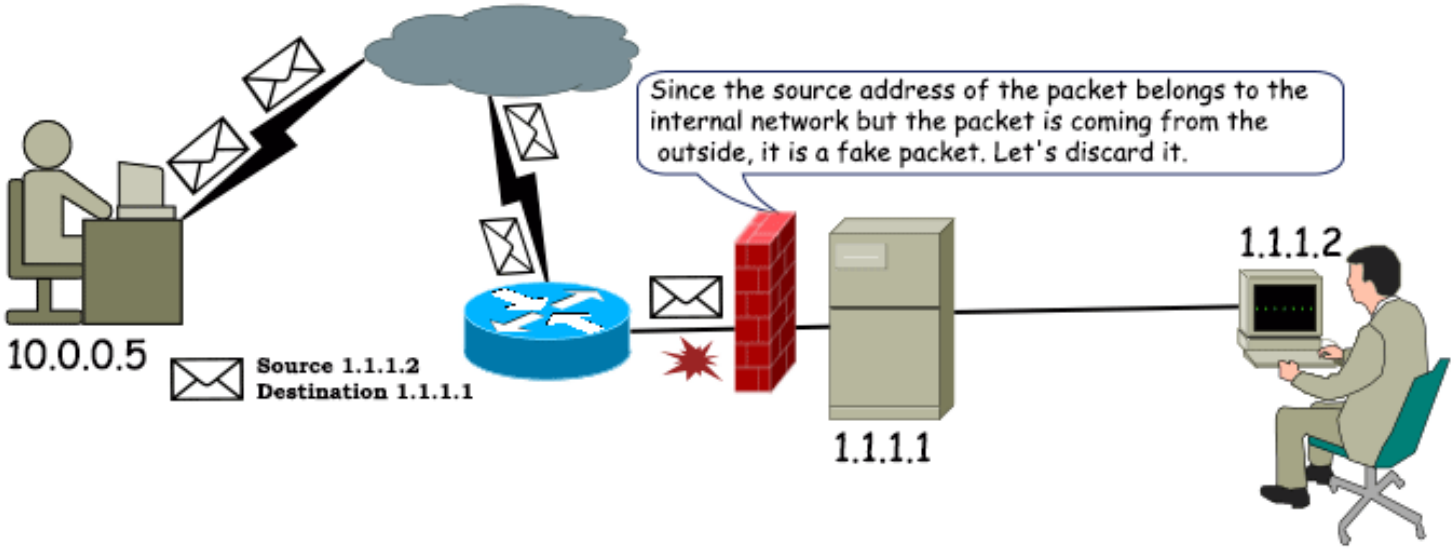
DoS Attack Defense strategies

Four lines of defense against DDoS attacks

1. Attack prevention and preemption
   - Before attack.
2. Attack detection and filtering
   - During the attack
3. Attack source traceback and identification
   - During and after the attack
4. Attack reaction
   - After the attack

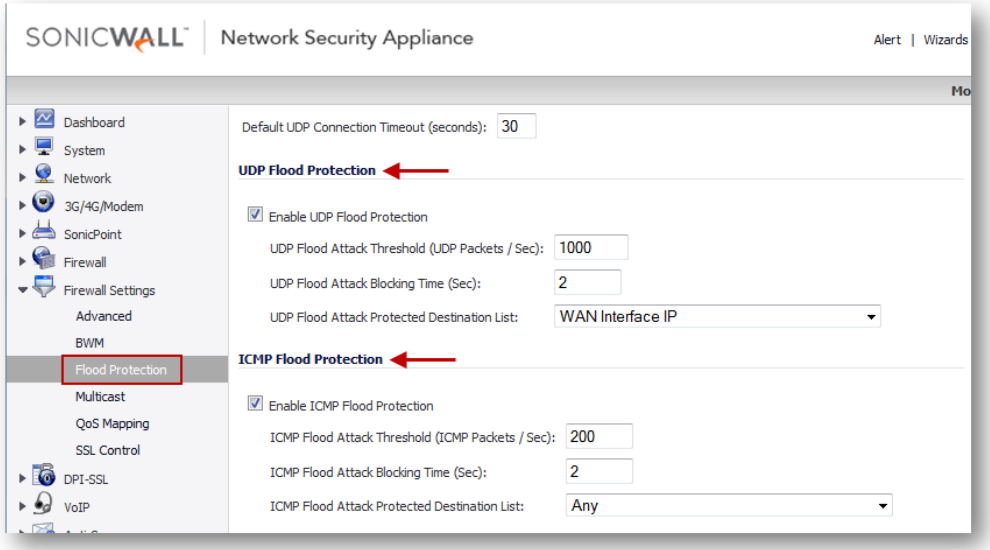
DoS Attack Prevention – Spoofed IP filtering

- Block spoofed source addresses, i.e., drop packets before they leave the spoofed machine:
  - On routers as close to source as possible, example with "Reverse Path Filtering" strategy: https://tldp.org/HOWTO/Adv-Routing-HOWTO/lartc.kernel.rpf.html
  - An ISP (Internet Service Provider) should filter out all packets it is known they are not generated within the possible assigned IP addresses
    (an ISP knows what is assigned since it is its own network)



DoS Attack Prevention – ICMP and UDP throttling

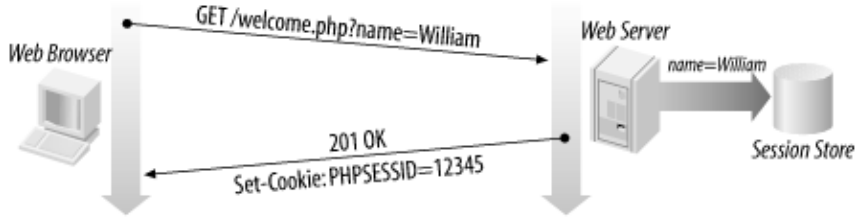
- Attacks using particular packet types, such as ICMP floods or UDP floods to diagnostic services, can be throttled by imposing limits on the rate at which these packets will be accepted. In normal network operation, these should comprise a relatively small fraction of the overall volume of network traffic



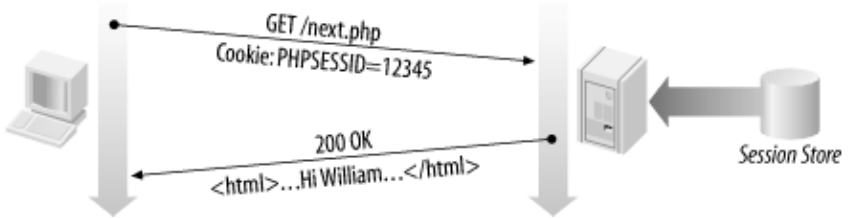
Network Throttling = Transmission Rate-limiting

DoS Attack Prevention – TCP SYN flood limit

- Use modified TCP connection handling code
  - **SYN Cookie:** No Information about a connection is saved; the server cryptographically encodes critical information in a cookie that is sent as the server's initial sequence number
    - Legitimate client responds with an ACK packet containing the incremented sequence number cookie
    - This still requires some effort and not everything fits the cookie
  - **Random Drop:** Drop an entry for an incomplete connection from the TCP connections table when it overflows

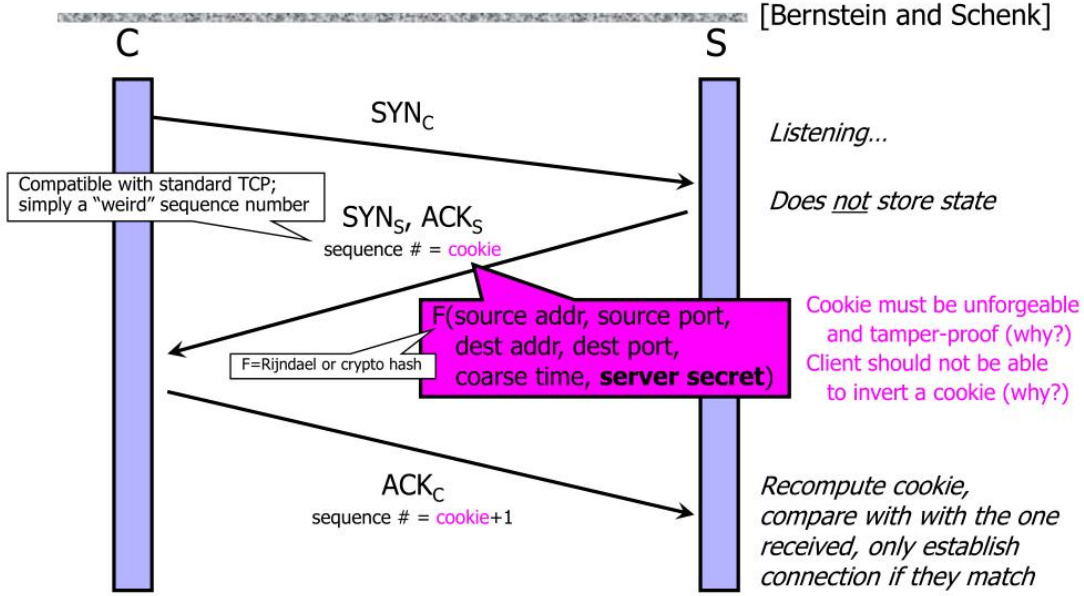


Normal cookie operation in HTTP (e.g., Web)



SYN Cookies (not only for HTTP)

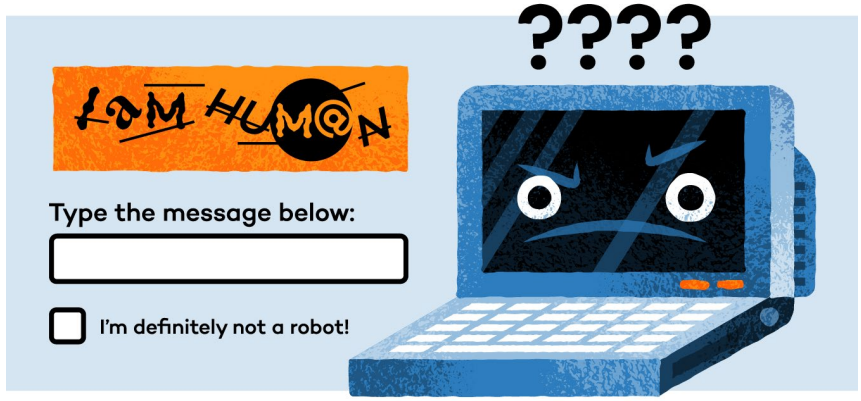
[Bernstein and Schenk]



More info: http://cr.yp.to/syncookies.html

DoS Attack Prevention – Amplification attacks

- Block IP **directed broadcasts**
- **Block suspicious services and combinations**
  - E.g., suspicious src IP and dst IP combinations
- Manage application attacks with a form of graphical puzzle **(captcha)** to distinguish legitimate human requests
  - Track also response time and modes, humans might behave differently than a bot
- Good general system security practices
  - Avoid to be infected and become a bot
- Use mirrored and replicated servers when high-performance and reliability is required
  - CDN and load balancing
- Cyber Threat intelligence:
  - **Darweb monitoring for Botnet services** available on the market



Captcha

Botnet as a service

DoS Attacks Response (1 of 2)

- Good Incident Response Plan
  - Details on how to contact technical personal for ISP
  - Needed to impose traffic filtering upstream
  - Details of how to respond to the attack
- Antispoofing, directed broadcast, and rate limiting filters should have been implemented
- Ideally have network monitors and IDS to detect and notify abnormal traffic patterns

Responding to DoS Attacks (2 of 2)

- Identify type of attack
  - Capture and analyze packets
  - Design filters to block attack traffic upstream
  - Or identify and correct system/application bug
- Have ISP trace packet flow back to source
  - May be difficult and time consuming
  - Necessary if planning legal action
- Implement contingency plan
  - Switch to alternate backup servers
  - Commission new servers at a new site with new addresses
- Update incident response plan
  - Analyze the attack and the response for future handling

Appendix

CalPolyPomona

ICMP-based DoS attacks