Professional Masters in Information and Cyber Security (PMICS)

Computer Science and Engineering,
University of Dhaka

CSE 802
Information Security Fundamentals

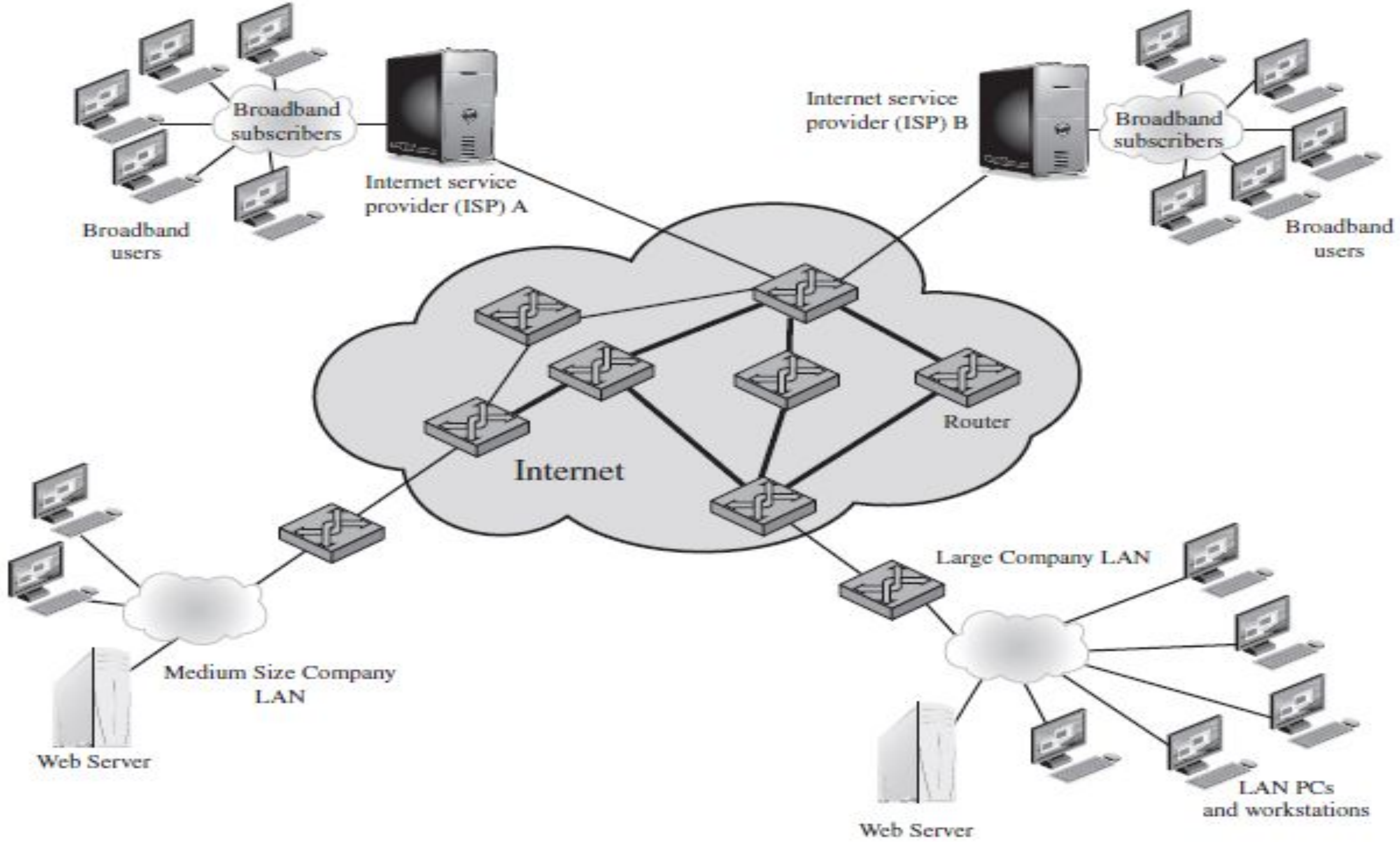
Topic: Wisreshark Study Material (LAB)

Denial-of-Service Attack

- Compromise the availability by blocking some services completely.
- Attempts to exhaust some critical resources associated with the service.
- With the improvement of bandwidth, large scale DoS attacks are now available.
- According to the NIST

“A **denial of service (DoS)** is 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.”

Example Network to Illustrate DoS Attack



Victim of DoS Attack

- **Network Bandwidth:**

  -- Network links connected to a server and ISP router are exhausted with huge volume of malicious traffic.

- **System Resources:**

  -- Overload or crash network handling software by occupying resources like buffer table, table of open connections, etc.

  -- Use packets (poison packet) that has structure to trigger a bug in network handling software to crash.

- **Application Resources:**

  -- Server is overloaded with a number of valid requests.

  -- Construct queries that can trigger a bug in the server code and cause it crash.

Classic Denial-of-Service Attack

- Flooding attack to an organization that can overwhelm the capacity of the network connection to that company.

- An attacker with access to a system with higher capacity network connection can distort the low-capacity network connection of a small-scale company.

- Can simple as sending ***ping*** command to the webserver of the target company.

- The destination router discards some packets as the network capacity of the neighboring company is low. The legitimate packets are discarded by the destination router to handle congestion.
- Also known as volumetric DoS attack.

Classic Denial-of-Service (DoS) Attack

- Problems for the attackers:
  - ICMP echo request contains the address of the source and thus can be detected easily.
  - ICMP echo response is reflected back to the destination causing large amount of traffic to the destination.
- Attackers try to hide the identity.

Source Address Spoofing

- Packets used for DoS attack use forged source address.
- Need sufficient privileged access to network handing code of the operating system.
  - Usually through raw socket interface.
  - The attackers need to install operating system specific device driver.
- In case of ***ping***, the response packets from the source address (real/false) are added to the flood toward the destination target.
- Source of attack cannot be identified by inspecting the packet header. Manual inspection of flow table of the routers along the path is required which is a manual process and time consuming.
- Source address spoofing is possible due to the vulnerabilities of TCP/IP.
- ISP can prevent address spoofing through egress filtering (Historically also known as Ingress filtering in some papers).

SYN Spoofing

- Attacks the ability of a network server to respond to TCP connection requests.
- An attack to system resources specifically the network handling software.

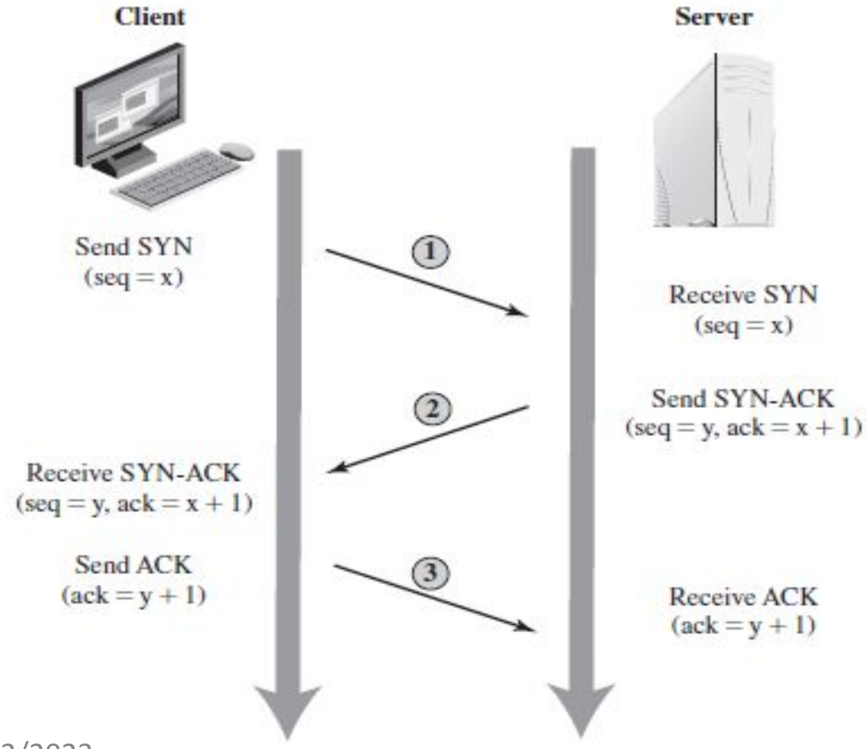


Figure 7.2 TCP Three-Way Connection Handshake

- Packet may be lost due to the transportation using IP.

SYN Spoofing

- The attacker generates a number of SYN connection request packets with forged source addresses.

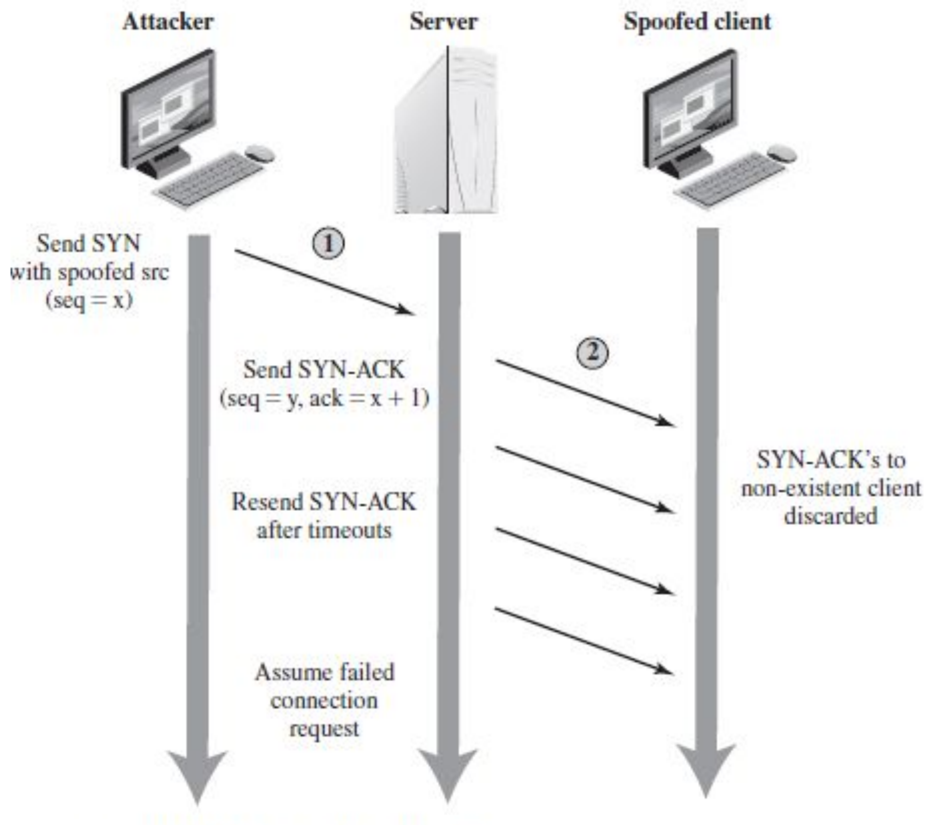

Attacker
Server
Spoofed client
Send SYN with spoofed src (seq = x)
1
Send SYN-ACK (seq = y, ack = x + 1)
2
SYN-ACK's to non-existent client discarded
Resend SYN-ACK after timeouts
Assume failed connection request

Figure 7.3 **TCP SYN SpoofingAttack**

- The attacker ideally wishes to

use addresses that will not respond to the SYN-ACK with a RST.

- The actual volume of SYN traffic can be comparatively low.
- Also known as TCP State Exhaustion attack.

Flooding Attack

- Overload network link capacity to a server.
- Alternatively, overload the server ability to handle and respond to this attack.
- Congestion caused to some routers on the path to target can discard packets.
- Network connection request may degrade severely or entirely fall.
- Virtually any type of network packet can be used in a flooding attack: simply needs to be of a type that is permitted to flow over the links toward the targeted system.
- Types:
  - ICMP Flood
  - UDP Flood
  - TCP SYN Flood

ICMP Flood

- Ping flood using ICMP echo request is a classical example of ICMP Flood.
- Nowadays ICMP echo request and response packets are restricted to pass through the firewall.
- Attackers started to use other ICMP packets: destination unreachable and time limit exceeds packet.
- These packets are required for correct operation of TCP/IP and thus allowed through firewall.
- Attacker can generate a large volume of one of these packets.

UDP Flood

- UDP packets are directed toward some port number. For example, echo service enabled on many server.
- Echo service returns a UDP packet containing the original packet back to the source address. If service is not allowed, then returns destination unreachable ICMP packet.
- Any packets generated in response only serve to increase the load on the server and its network links.
- Spoofed source addresses are normally used if the attack is generated using a single source system.
- If multiple systems are used for the attack, often the real addresses of the compromised, zombie, systems are used.
- When multiple systems are used, the consequences of both the reflected flow of packets and the ability to identify the attacker are reduced.

TCP SYN Flood

- Send normal TCP connection requests, with either real or spoofed source addresses.
- difference between a **SYN spoofing** attack and a **SYN flooding** attack: total volume of packets.
- could also use TCP data packets, which would be rejected by the server as not belonging to any known connection.
- **Traffic generated by a single system:** attacker is easier to trace and need mechanism to generate high amount of traffic.
- **Traffic generated by multiple systems:** attacker is difficult to trace and easier to generate huge volume of traffic.

Attack using Multiple Systems

- Distributed Denial of Service Attacks (DDoS attack).
- Reflector Attacks.
- Amplifier Attacks.

Distributed Denial-of-Service Attacks (DDoS)

- Popular tool for DDoS attack: botnet.
- The attacker uses malware to subvert the system (zombie) and install an attack agent which they can control.
- Large collections of such systems under the control of one attacker is known as **botnet**.

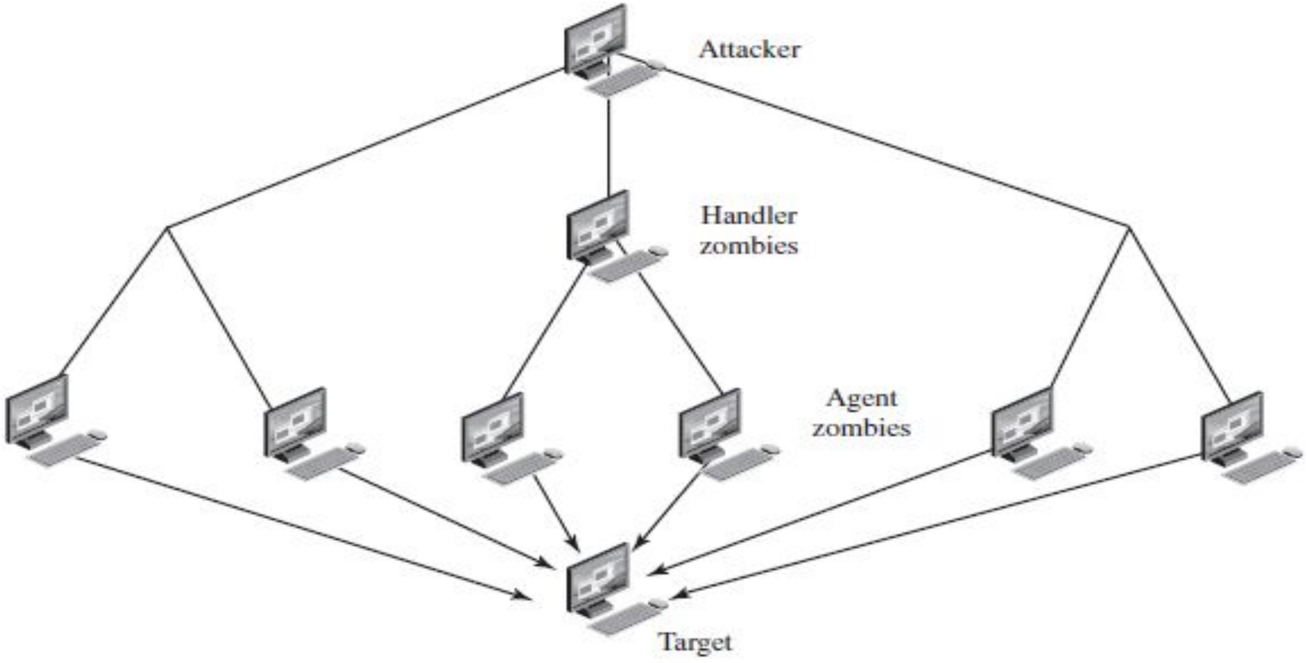


Figure 7.4 **DDoS Attack Architecture**

Distributed Denial-of-Service Attacks (DDoS)

- Well known tools to perform DDoS: Tribe Flood Network (TFN) and TFN2000 can run on Solaris, Unix and Windows machine.
- use a version of the two-layer command hierarchy.
- The agent was a Trojan program that was copied to and run on compromised, zombie systems capable of implementing ICMP flood, SYN flood, UDP flood, and ICMP amplification forms of DoS attacks.
- TFN did not spoof source addresses in the attack packets. Rather it relied on a large number of compromised systems and the layered command structure, to obscure the path back to the attacker.

Distributed Denial-of-Service Attacks (DDoS)

- Communications between the handler and its agents was encrypted and intermixed with other packets to avoid detection by analyzing control traffic.
- Both these communications and the attacks themselves could be sent via randomized TCP, UDP, and ICMP packets.
- The handler was simply a command-line program run on some compromised systems. The attacker accessed these systems using any suitable mechanism giving shell access, and then ran the handler program with the desired options.
- Many DDoS tools use an IRC or similar instant messaging server program, or web-based HTTP servers to communications with the agents.
- Many recent tools also use cryptographic mechanisms to authenticate the agents to the handlers.
- The prevention is to use good system security practices and keep the OS and applications current and patched.

Application-based Bandwidth Attack: SIP Flood

- Force the target to execute resource-consuming operations that are disproportionate to the attack effort.
- Session Initiation Protocol (SIP) was used to setup call in VoIP.

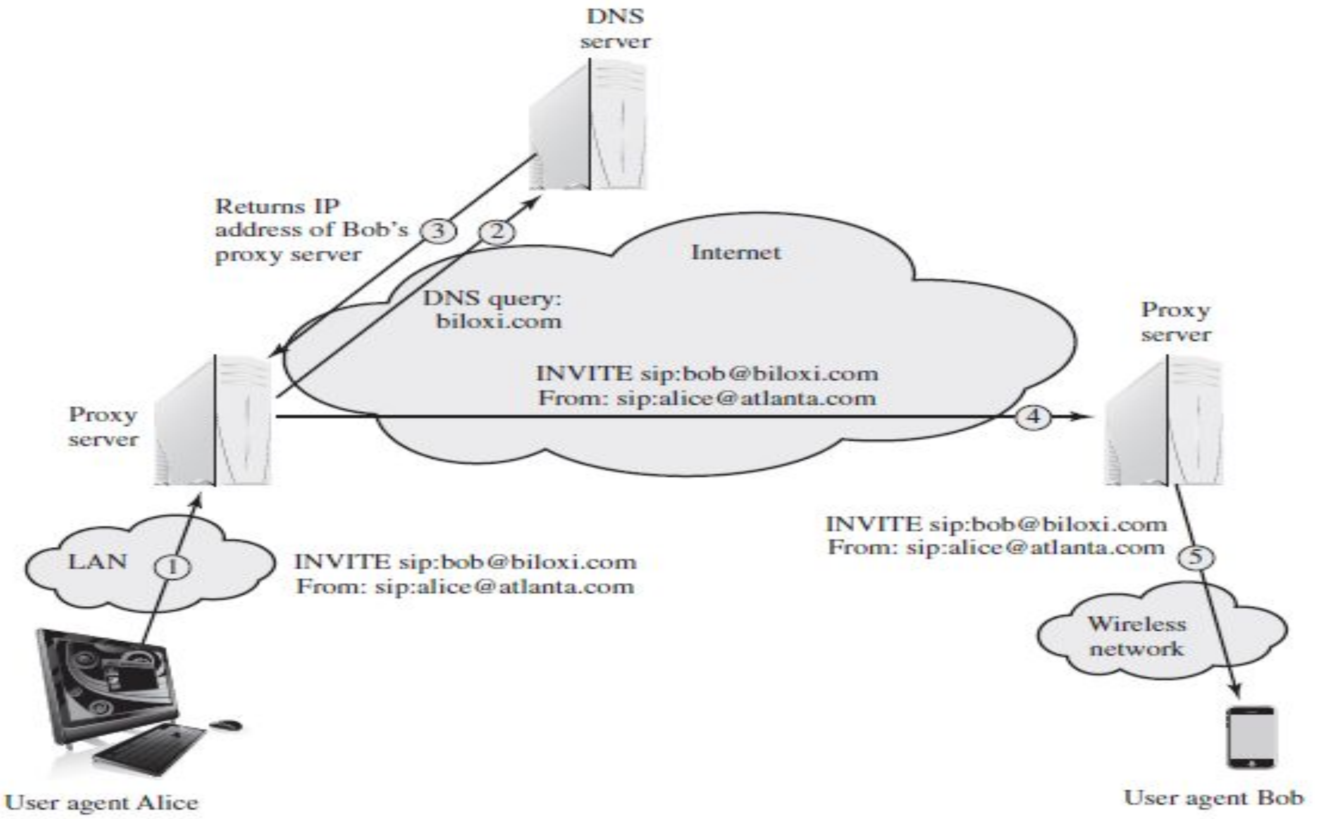

DNS server
Returns IP address of Bob's proxy server
3
2
Internet
DNS query: biloxi.com
Proxy server
INVITE sip:bob@biloxi.com From: sip:alice@atlanta.com
4
Proxy server
LAN
1
INVITE sip:bob@biloxi.com From: sip:alice@atlanta.com
INVITE sip:bob@biloxi.com From: sip:alice@atlanta.com
5
Wireless network
User agent Alice
User agent Bob

Figure 7.5 **SIP INVITE Scenario**

Application-based Bandwidth Attack: SIP Flood

- The attacker can flood a SIP proxy with numerous INVITE requests with spoofed IP addresses, or alternately a DDoS attack using a botnet.
- Effects:
  - First, their server resources are depleted in processing the INVITE requests.
  - Second, their network capacity is consumed.
  - Call receivers are also victims of this attack. A target system will be flooded with forged VoIP calls, making the system unavailable for legitimate incoming calls.

Application-based Bandwidth Attack

- **HTTP Flood:**
- An HTTP flood bombards Web servers with HTTP requests originated from many different bots.
- The requests can be designed to consume considerable resources. For example, an HTTP request to download a large file from the target causes the Web server to read the file from hard disk, store it in memory, convert it into a packet stream, and then transmit the packets.
- Consumes memory, processing, and transmission resources.
- **Recursive HTTP Flood:**
- The bots start from a given HTTP link and then follows all links on the provided Web site in a recursive way. This is also called spidering.

Application-based Bandwidth Attack: SlowLoris

- Exploits the common server technique of using multiple threads to support multiple requests to the same server application.
- It tries to monopolize all of the available request handling threads on the Web server by sending HTTP requests that never complete.
- The HTTP protocol specification requires a blank line to indicate the end of the request headers and the beginning of the payload.
- On each connection, slowloris sends an incomplete request that does not include the terminating newline sequence
- Web server keeps the connection open, expecting more information to complete the request.

Application-based Bandwidth Attack: SlowLoris

- Slowloris uses valid HTTP traffic and thus intrusion detection and prevention system based on signature cannot detect it.
- Counter measures:
  - limiting the rate of incoming connections from a particular host
  - varying the timeout on connections as a function of the number of connections
  - delayed binding performed by load balancing software.

Amplifier Attack

- The attacker sends a number requests with spoofed source address (target: valid machine) to a number of servers, all the servers will reply back to the target machine.
- This attack is easier to deploy and difficult to trace back the attacker.
- Two types:
  - Simple reflection attack
  - Amplification attack.

Reflection Attack

- The attacker sends a request with spoofed source address (target: valid machine) to a server and it replies back to the target machine.
- the attacker would like to use a service that created a larger response packet than the original request. Example, UDP service, DNS, Chargen, SNMP, ISAKMP, etc.
- The intermediary router or server is chosen as high capacity system.
- the attacker spreads the attack over a number of intermediaries in a cyclic manner, then the attack traffic flow may well not be easily distinguished from the other traffic flowing from the system.
- Attack can use TCP SYN packets and exploits the normal three-way handshake used to establish a TCP connection.
- Attack with TCP SYN is a flooding attack which is different from SYN Spoofing. The attacker

Reflection Attack

- Any TCP service can be used for attack. TCP connection requests are indistinguishable from normal traffic.

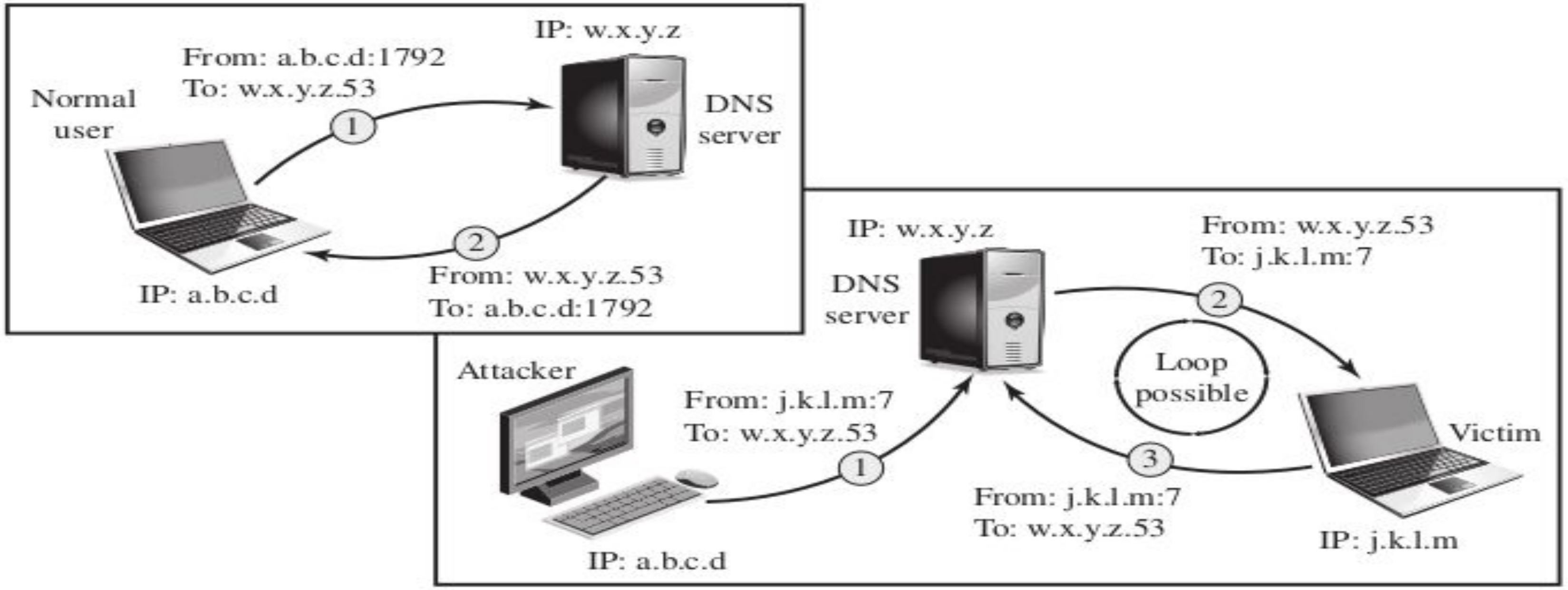


Figure 7.6 **DNS Reflection Attack**

- Although effective but easier to detect.

Reflection Attack

- Attacker can use single system or multiple system (botnet).
- Basic counter measure is to block spoofed source address.

Amplification Attack

- A variant of reflector attacks.
- Involves sending a packet with a spoofed source address for the target system to intermediaries.
- **Difference**: Generates multiple response packets for each original packet sent.
- Achieved by directing the original request to the broadcast address for some network. As a result, all hosts on that network can potentially respond to the request.
- Example: ping, UDP echo service.

Amplification Attacks

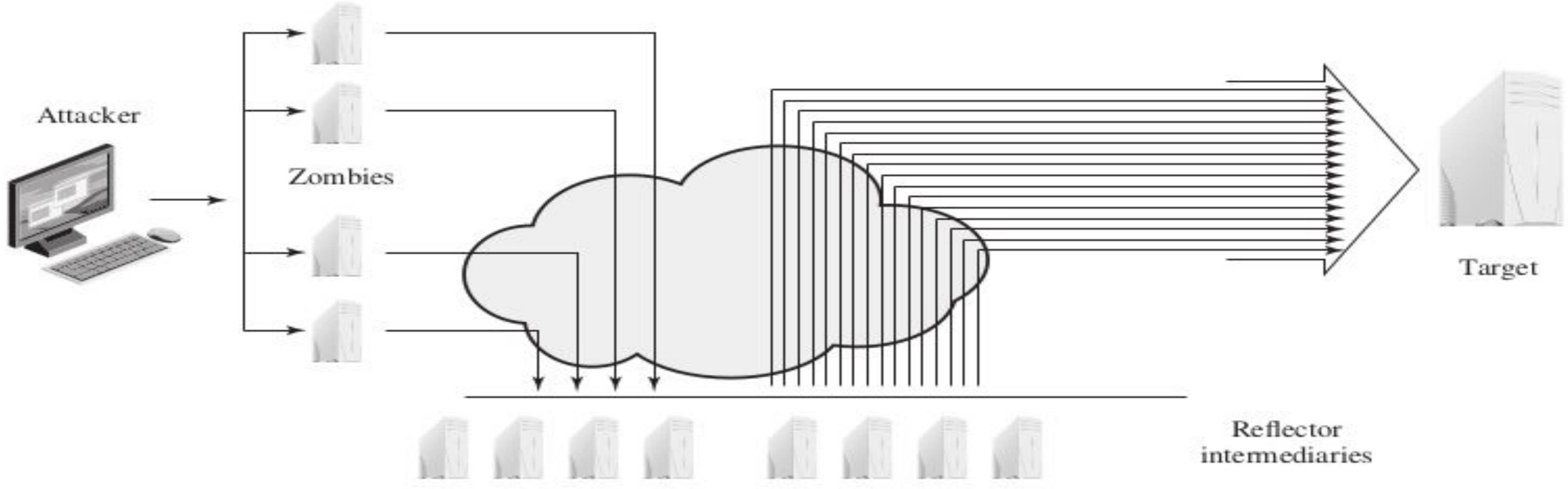


Figure 7.7 **Amplification Attack**

Counter measures:

- Block broadcast originated outside the network.
- Block ping and echo being accessed from outside an organization.

DNS Amplification Attacks

- Attack uses packets directed at a legitimate DNS server as the intermediary system.
- Exploits the behavior of the DNS protocol to convert a small request into a much larger response.
- Using the classic DNS protocol, a 60-byte UDP request packet can easily result in a 512-byte UDP response.
- More recently, the DNS protocol has been extended to allow much larger responses of over 4000 bytes to support extended DNS features such as IPv6, security, and others.

DNS Amplification Attack

- In this attack, a selection of suitable DNS servers with good network connections are chosen.
- The attacker creates a series of DNS requests containing the spoofed source address of the target system. These are directed at a number of the selected name servers.
- Because of the amplification achieved, the attacker need only to generate a moderate flow of packets to cause a larger, amplified flow to flood and overflow the link to the target system.
- Intermediate systems will also experience significant loads.

DNS Amplification Attack

- A variant of this attack exploits recursive DNS name servers.
- Many DNS systems support recursion by default for any requests. They are known as open recursive DNS servers.
- Attackers may exploit such servers for a number of DNS-based attacks, including the DNS amplification DoS attack.
- Counter measure:
  - Prevent spoofed source address.
  - Allow recursive queries only for local clients.

Defenses against DoS Attack

- **Attack prevention and preemption (before the attack):**
  - enable the victim to endure attack attempts without denying service to legitimate clients.
  - Techniques include enforcing policies for resource consumption and providing backup resources available on demand. In addition, prevention mechanisms modify systems and protocols on the Internet to reduce the possibility of DDoS attacks.
- **Attack detection and filtering (during the attack):**
  - These mechanisms attempt to detect the attack as it begins and respond immediately. This minimizes the impact of the attack on the target.
- **Attack source traceback and identification (during and after the attack):**
  - This is an attempt to identify the source of the attack to prevent future attacks.
- **Attack reaction (after the attack):**
  - This is an attempt to eliminate or curtail the effects of an attack.

Attack prevention and preemption (before the attack)

- Prevent the usage of spoofed IP address by applying egress and ingress filtering by appending explicit access control rules in ISP router.

- Filters may be used to ensure that the path back to the claimed source address is the one being used by the current packet. For example, using the “ip verify unicast reverse-path” command.
  - ISP often tries to avoid providing such filter.

- In case of flooding, the filters must be applied to traffic before it leaves the ISP’s network, or even at the point of entry to their network.

Attack prevention and preemption (before the attack)

- Limit packet rate for ICMP flood or UDP flood.
- Use modified version of the TCP connection handling code to handle SYN Spoofing. Instead of saving the connection details, critical information about the requested connection is cryptographically encoded in a cookie that is sent as the server's initial sequence number. This is sent in the SYN-ACK packet from the server back to the client.
- When a legitimate client responds with an ACK packet containing the incremented sequence number cookie, the server is then able to reconstruct the information about the connection.
  - Consume resources of server.
  - Cannot support extensions of TCP.

Attack prevention and preemption (before the attack)

- ***Selective or random drop*** of incomplete TCP connections from the connection table when the table overflows.
- Modify TCP/IP parameter such as connection table size and timeout period and limit rate on the network link.
- block the use of IP-directed broadcasts to stop broadcast amplification.
- Limit or block traffic to suspicious services, or combinations of source and destination ports.
- Defending against attacks on application resources generally requires modification to the applications targeted, such as Web servers. These often take the form of a graphical puzzle, a captcha, which is easy for most humans to solve but difficult to automate.

Attack prevention and preemption (before the attack)

- Good system security practice.
  - Ensure that the systems are not compromised and used as zombie systems.
  - Suitable configuration and monitoring of high performance, well-connected servers.
  - An organization should consider mirroring and replicating these servers over multiple sites with multiple network connections.

Responding to a Denial-of-Service Attack

- Requires proper incidence plan- contacting technical persons in ISP, Division of managing resources between company’s personnel and ISP personnel.
- An organization should implement the standard antispoofing, directed broadcast, and rate limiting filters.
- There should have some form of automated network monitoring and intrusion detection system to notify abnormal traffic (on the basis of changes in patterns of flow information, source addresses, or other traffic characteristics) be detected.
- An organization should know its normal traffic patterns to determine a baseline with which to compare abnormal traffic flows.

Responding to a Denial-of-Service Attack

- When a DoS attack is detected, the first step is to identify the type of attack and hence the best approach to defend against it.

- Typically this involves capturing packets flowing into the organization and analyzing them, looking for common attack packet types. This may be done by organizational personnel using suitable network analysis tools.

- From this analysis the type of attack is identified, and suitable filters are designed to block the flow of attack packets. These have to be installed by the ISP on its routers.

- The organization may ask its ISP to trace the flow of packets back in an attempt to identify their source to take legal steps.

Responding to a Denial-of-Service Attack

- In case of attack from a large number of distributed or reflected systems, the organization needs a contingency strategy either to switch to alternate backup servers or to rapidly commission new servers at a new site with new addresses, in order to restore service.

DDoS Mitigation using Multi-Layer Switching and Content Delivery Network (CDN)

- Modern enterprises employ multi-layer switching and content delivery network to mitigate DDoS.
- A multi-layer switch acts like a router, except for two very important differences:

(1) A router carries out its functions through software running in an embedded microprocessor and a multi-layer switch uses dedicated hardware to do the same.

(2) A router works only at Layer 3 (the IP Layer) of the OSI TCP/IP protocol stack and a multi-layer switch can route a packet on the basis of information corresponding to any of the layers 3 and above in the protocol stack.

- A Layer 4 switch carries out port translation for sending incoming packets to one or more machines hidden behind a single IP address. Packet forwarding takes place at wirespeed.
- Layers 4-7 switches that are now commonly used in enterprise level server systems are also referred to as “content switches.”
- Content switches are used for load balancing when services are provided through a Content Delivery Network (CDN). With a content switch, a client can be connected to the least loaded node of a CDN at network speed.

DDoS Mitigation using Multi-Layer Switching and Content Delivery Network (CDN)

- If a network of servers (providing the same service) are connected behind a multi-layer switch, the switch will deliver the flow to the least loaded server to provide response and to mitigate volumetric DDoS attack.
- The same idea is used in content delivery network (CDN).

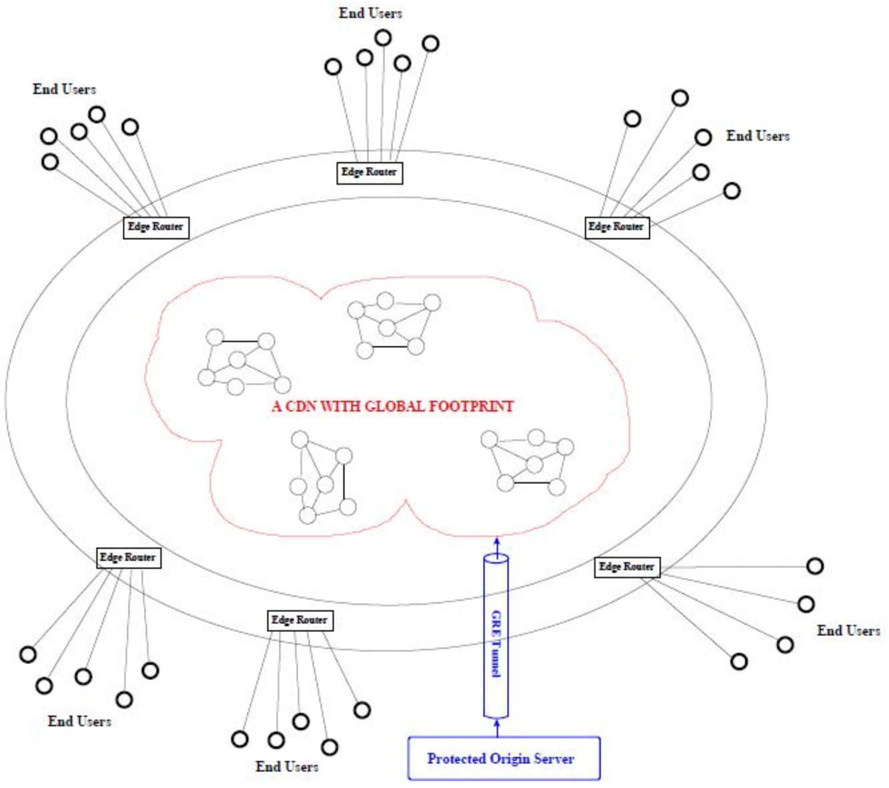


Figure 3: *Delivering Web Content through a Geographically Distributed CDN* (*This figure is from Lecture 29 of "Lecture Notes on Computer and Network Security" by Avi Kak*)

DDoS Mitigation using Multi-Layer Switching and Content Delivery Network (CDN)

- An attacker could try to defeat a CDN by making rapid-fire requests for content that would cause the CDN to have to go back to the origin server. **Solution:** Caching data.
- There are certain types of requests that the CDN would need to send back to the origin server: login requests, search, etc. **Solution**: Rate limiting Firewall rule.

DDoS Attack Mitigation with Manual Reconfiguration of BGP Routing

- BGP stands for Border Gateway Protocol.
- The internet is divided into a network of Autonomous Systems (AS).
- A large network controlled by a single organization would qualify as an Autonomous System (AS). For example, AT&T, Comcast, Verizon, Level 3, Google, and even Purdue University are examples of ASs.
- An AS will have direct network connections with multiple other ASs through the internet backbone.
- Each AS is identified by a unique number called the AS Number (ASN) provided by the Regional Internet Registries (APNIC for Asia).
- AS is also characterized by IP Prefixes.

DDoS Attack Mitigation with Manual Reconfiguration of BGP Routing

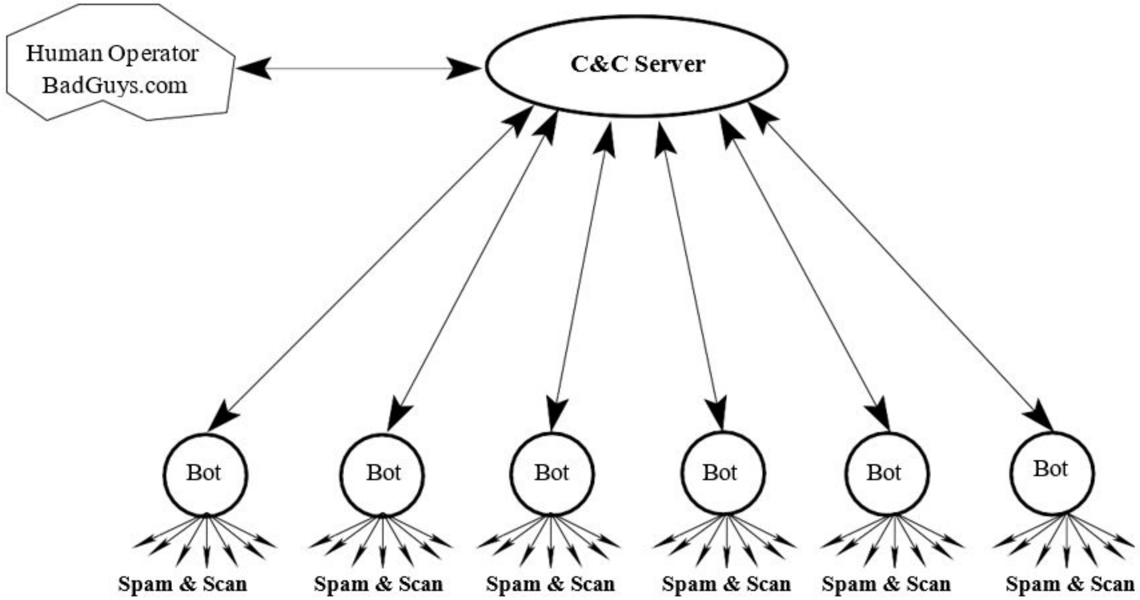
- Each AS enters into bilateral business relationships with a certain number of other ASs. The relationships are of two types: *provider-customer* and *peer-peer*.
- The business (or otherwise) relationships that an AS enters define its network neighborhood. Each AS broadcasts its ASN and its IP prefix to its neighbors through what are referred to as BGP announcements.
- If an AS ***A*** has an AS ***B*** as its immediate neighbor, ***A*** transmits to ***B*** its BGP announcement **(ASN_A,IP_prefix_A)**. And if ***C*** is a neighbor of ***B***, ***B*** forwards to ***C*** a BGP announcement **(ASN_B, ASN_A,IP_prefix_A)**; and so on. As these announcements propagate from AS to AS in the internet, eventually all the ASs would know how to route the packets to the IP prefix for *A*.
- From the standpoint DDoS attack mitigation, since the routing policies regarding the exchange of traffic between the peers are configured and updated manually, an instant routing policy change can re-route an ongoing DDoS attack through a longer route in which the DDoS traffic can be subject to greater filtering.

Bot and Bot Master

- Viruses and worms are equipped with a certain fixed behavior. Any time they migrate to a new host, they try to engage in that same behavior.
- A bot, on the other hand, is usually equipped with a larger repertoire of behaviors.
- A bot maintains, directly or indirectly, a communication link with a human handler, known typically as a bot-master.
- The specific behavior that a bot shows at any given time on any specific host depend, in general, on what commands it receives from some human.
- A collection of bots working together for the same bot-master constitutes a botnet.
- A bot master can harness the power of several bots working together to bring about a result that could be more damaging than what can be accomplished by a single bot (or a worm or a virus) working all by itself.
  - DDoS attack.
  - Spread virus and worm infections.
  - Spam spreading.

Command and Control Needs of a BotNet

- To follow the instructions of a bot master, a bot must have embedded in it some communication capabilities.
- There are two different ways in which a bot may receive commands from its master: (1) the push mode; and (2) the pull mode.
- Both of these modes require a command-and-control (C&C) server that "talks" to the individual bots.



- In push mode, IRC (Internet Relay Chat) Servers are preferred mode of servers.
- In pull mode, HTTPD servers are preferred.
- A botnet exploit is more likely to go undetected if the communication between the bots and the C&C server uses standard protocols as opposed to some custom designed protocol.

Command and Control Needs of a BotNet

- In push mode, IRC (Internet Relay Chat) Servers are preferred mode of servers.

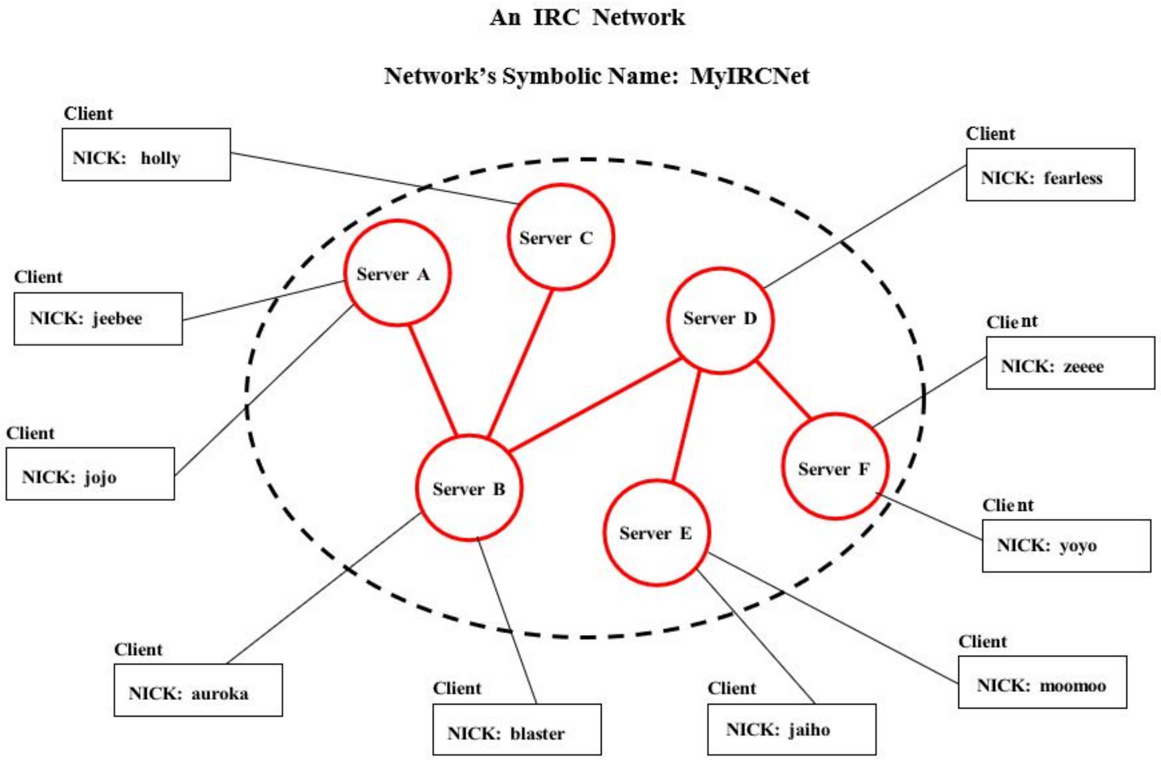
- In pull mode, HTTPD servers are preferred.

- A botnet exploit is more likely to go undetected if the communication between the bots and the C&C server uses standard protocols as opposed to some custom designed protocol.

- The bot master only has to communicate his/her intentions to the C&C server in order for those intentions to be sent to all the bots. The communications between the human and the C&C server to be infrequent, making it that much harder to discover the human handler.

IRC Protocol

- A chat server is a server socket that listens for incoming requests from new clients. When a new request is received, the server socket forks a client socket to a new child process.
- The individual chat clients could be plugged into different machines in different parts of the world and all servers in the same IRC network scattered in different machines in different parts of the world would appear as a single logical chat server to all the clients.



- Each server can communicate using TCP/IP messages.
- The path (Overlay network) shown in the figure is for exchanging IRC messages.
- The overlay network does not contain any loop for easy exchange of state information among the servers. This is also known as real time server-to-server synchronization.

IRC Protocol

- Each user in an IRC network is identified by a unique nickname.
- A channel is simply a set of users that ensures grouping of users.
- Two kinds of channels in an IRC network: 1) channels that are local to each specific server and 2) channels that are global to all the servers. The former are denoted with the ‘&’ prefix and the latter with the ‘#’ prefix.

#movie = {holly, zeee, moomoo}

&localSchool = {jeebee, jojo}

- When a message is sent to a channel, it is sent to all the users (servers) that are in the set corresponding to the channel.
- The IRC protocol considers the first person to start a new channel as the operator of that channel. An operator has certain privileges, such as the privilege to “kick” a troublesome user off a channel.
- A user needs to register a nick using NickServ or NS command and the command for registering a channel is ChanServ or CS.

IRC Protocol

- The command and control messages in IRC network following the syntax:

1. an optional “:”-prefixed string, followed by

1. a valid IRC command in ASCII (or the corresponding 3-digit number), followed by

1. the arguments to the command.

Example: :botBoss MODE #botnetUnderground +k abracadabra

- An IRC message is always terminated in the internet line terminator, which is CR+LF.

- The MODE command that is included in the list shown above is used to set the properties of servers, channels, and users.

IRC Protocol

- A channel ***botnetUnderground*** can be made secret by

  MODE #botnetUnderground +s

A channel becomes invisible to the non-members of the channel.

- A user *botboss* can be made invisible by

  MODE botBoss +I

The user becomes invisible to the non-members of its channel.

- Suppose a client ***botBoss*** sends the following message to the connected server:

  MODE #botnetUnderground +k abracadabra

- When the same message is forwarded to the other servers in the IRC network, it becomes

  :botBoss MODE #botnetUnderground +k abracadabra

- The command for sending text message to the server is PRIVMSG

  PRIVMSG #botnetUnderground   :Hello Bots! Are you ready to wage war?

Resources

- **Chapter 7**, Computer Security Principles and Practice --- William Stalling and Lawrie Brown
- Lecture 29 (section 29.1, 29.2, 29.3, 29.7, 29.7.1, 29.7.2) from https://engineering.purdue.edu/kak/compsec/.

Acknowledgment

Lecture slides 40-52 are adapted from lecture 29 of Avinash Kak from https://engineering.purdue.edu/kak/compsec/.

Resources

- **Multi Layer Switches**: https://www.fiber-optic-transceiver-module.com/what-is-a-multilayer-switch-and-how-to-use-it.html
- **Internet Backbone**: https://www.networkworld.com/article/3532318/what-is-the-internet-backbone-and-how-it-works.htm