Module 8
Network Security

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University, Jackson, MS 39232
E-mail: natarajan.meghanathan@jsums.edu

Module Topics

- 8.1 Classical Denial of Service (Spoofing-based) Attacks
- 8.2 Defense using Cryptography
- 8.3 IPSec
- 8.4 Firewalls and IDS

8.1 Classical Network Security Attacks

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

Port Scanning

- Port scan: is a program that when run for a particular IP address, reports the following:
  - Which standard ports or services are running and responding on the target system.
  - What operating system is installed on the target system
  - What applications (and their versions) are present at the target system
- All of the above information can be collected quietly, anonymously, without identification or authentication, drawing little attention to the scan.
- After knowing details like the OS, the application programs and versions, an attacker can explore the weaknesses of these software and potential loopholes to get into the target system using these services.
  - Sometimes sending messages from an application to another application running at the target host may help us to obtain the version details of that application at the target.
- Port scanning tools: nmap, netcat – free tools, several commercial tools are also available.
  - nmap: Given an IP address, nmap reports all open ports, the service they support, and the user ID of the daemon providing the service.
  - The user ID is important because it helps us to further explore the gains that could be obtained if his/ her service running on the target host is compromised.

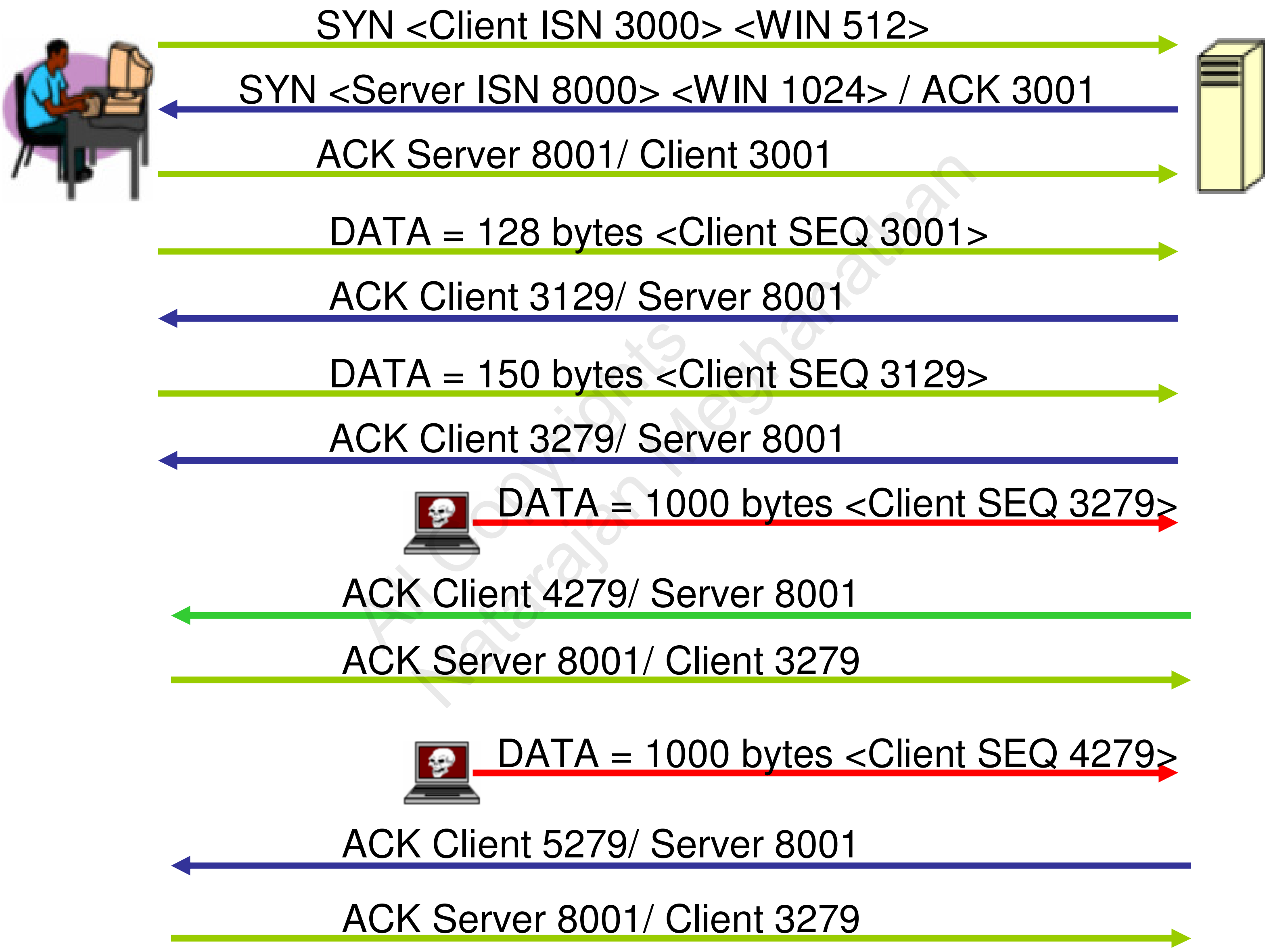
Syn Flood Attack

- When a client attempts to start a TCP connection to a server, the client and serve exchange a series of messages as follows:
  - The client requests a connection by sending a SYN message to the server.
  - The server acknowledges the request by sending SYN-ACK back to the client
  - The client responds with an ACK and the connection is established.
- Occasionally, packets get lost or damaged in transmission. The destination maintains a queue called the SYN_RECV connections, tracking those connection requests for which a SYN-ACK has been sent but the corresponding ACK has not yet been received.
- If the SYN-ACK or the ACK packet is lost, eventually the destination host will timeout the incomplete connection and discard it from its queue.
- The attacker sends many SYN requests from spoofed non-existing IP addresses and never responds back with ACKs, thereby filling up the SYN-RECV queue at the server.
- The server waits with the connection requests in the SYN-RECV queue and denies permitting any legitimate client connection request arriving in the mean time.

TCP Session Hijacking

- Session hijacking is the act of taking over an already established TCP session and injecting your own packets into that stream so that your commands are processed as the authentic owner of the session.
- A TCP session is merely identified by the quadruple: Client IP address, Client Port number, Server IP address and Server Port number.
- Any packets that reach either machine with the above identifiers are assumed to be part of the existing session.
- Hence, if an attacker can spoof these items, they can pass TCP packets to the client or server and have those packets processed as coming from the other machine.
- Two steps: Desynchronize the session and Inject own commands
- Desynchronizing the session: Predict the sequence number to be used by a client (or server) and use that sequence number before the client (or server) gets a chance to.
  - How to do it? Use Local Session Hijacking or Blind Session Hijacking
  - Local Session Hijacking: If we have access to the network and can sniff the TCP session, we can tell the next expected sequence number from the ACK packets exchanged
  - Blind Session Hijacking: If we do not have the ability to sniff the TCP session between the client and server, then we have try all options and guess the expected sequence number.

Desynchronizing a Session



SYN <Client ISN 3000> <WIN 512>

SYN <Server ISN 8000> <WIN 1024> / ACK 3001

ACK Server 8001/ Client 3001

DATA = 128 bytes <Client SEQ 3001>

ACK Client 3129/ Server 8001

DATA = 150 bytes <Client SEQ 3129>

ACK Client 3279/ Server 8001

DATA = 1000 bytes <Client SEQ 3279>

ACK Client 4279/ Server 8001

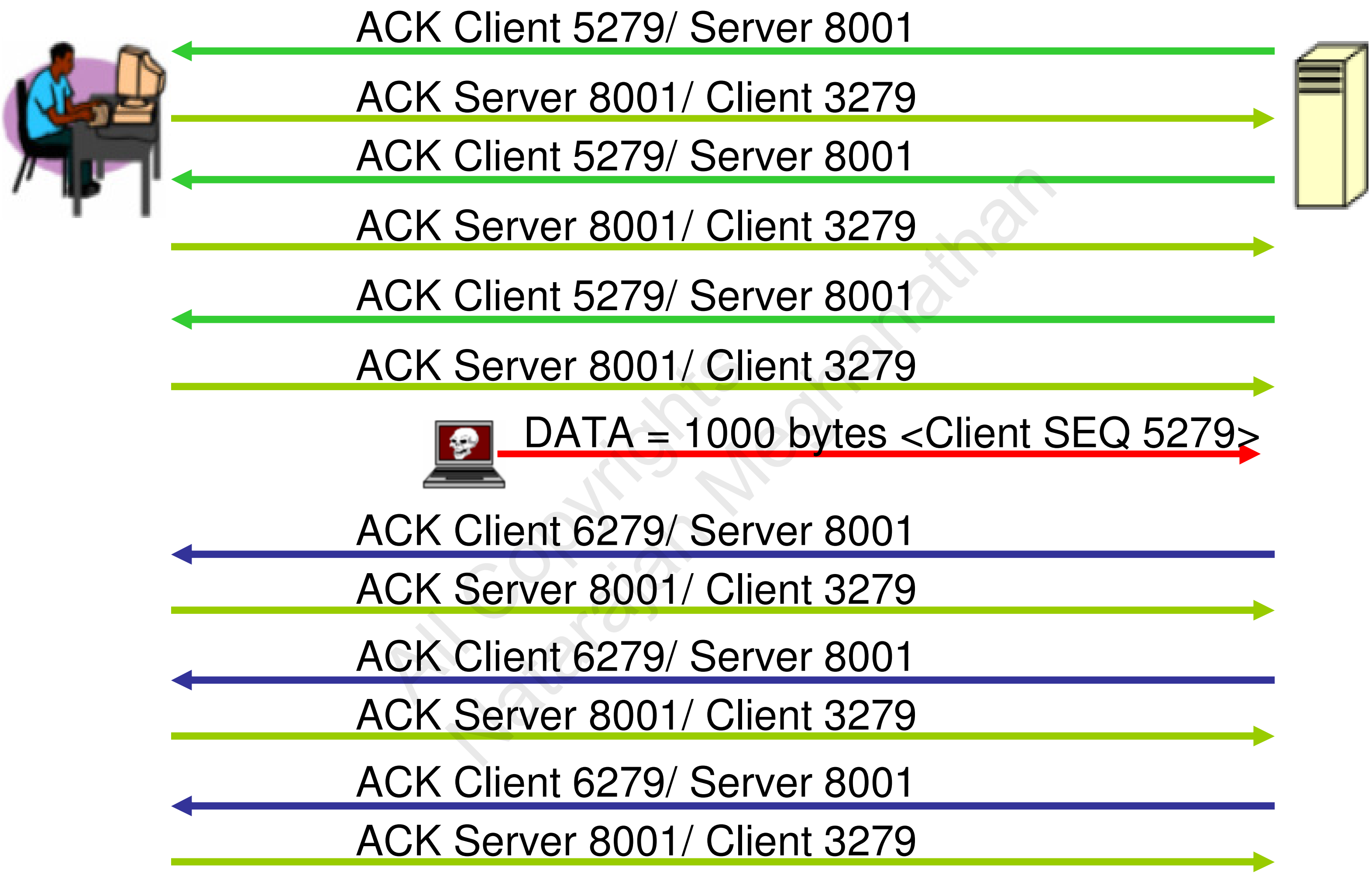
ACK Server 8001/ Client 3279

DATA = 1000 bytes <Client SEQ 4279>

ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

TCP ACK Storm



ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

DATA = 1000 bytes <Client SEQ 5279>

ACK Client 6279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 6279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 6279/ Server 8001

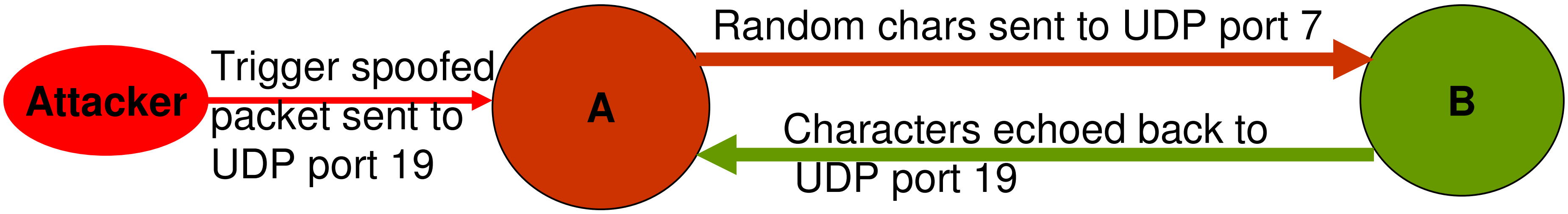
ACK Server 8001/ Client 3279

TCP Session Hijacking

- When the attacker successfully hijacks the TCP session and injects its own data (and spoofs the server as if the data is coming from the original client), the server will acknowledge the receipt of the data by sending to the original client an ACK packet.
- This ACK packet will most likely contain a sequence number that the client is not expecting, so when the client receives this packet, it will try to resynchronize the TCP Session with the server by sending it an ACK packet with the sequence number that it is expecting.
- This ACK packet will in turn contain a sequence number that the server is not expecting and so the server will resend its last ACK packet.
- This cycle will go on and on, and this rapid passing back and forth of the ACK packets creates the TCP ACK Storm.
- The attacker could keep injecting more and more data, the size of the ACK storm increases and can quickly degrade the network performance.
- The original client will have to eventually get exhausted after a certain number of resynchronization attempts and close the connection with the server.

Echo-Chargen Attack

- The Chargen (Character Generator) service is an internet-layer protocol and is intended for testing and measurement purposes.
- A host may connect to a server that supports the Chargen protocol, on either TCP or UDP port 19.
- Upon opening a TCP connection, the server starts sending arbitrary characters to the connecting host and continues until the host closes the connection.
- In the UDP version of the protocol, the server sends an UDP packet containing a random number (between 0 and 512) of characters every time it receives an UDP packet from the connecting host.
- <u>Attack Scenario:</u> The attacker sends a packet to machine A's UDP port 19 with machine B as the forged source address and UDP port 7 as the source port.
  - The attacker spoofs a conversation between the two services and redirects the output of each service to the other, creating a rapidly expanding spiral of traffic.
  - Eventually, the attack begins to consume memory and processor power at the targeted devices A and B, causing them to become non-responsive to user commands.



Attacker

Trigger spoofed packet sent to UDP port 19

A

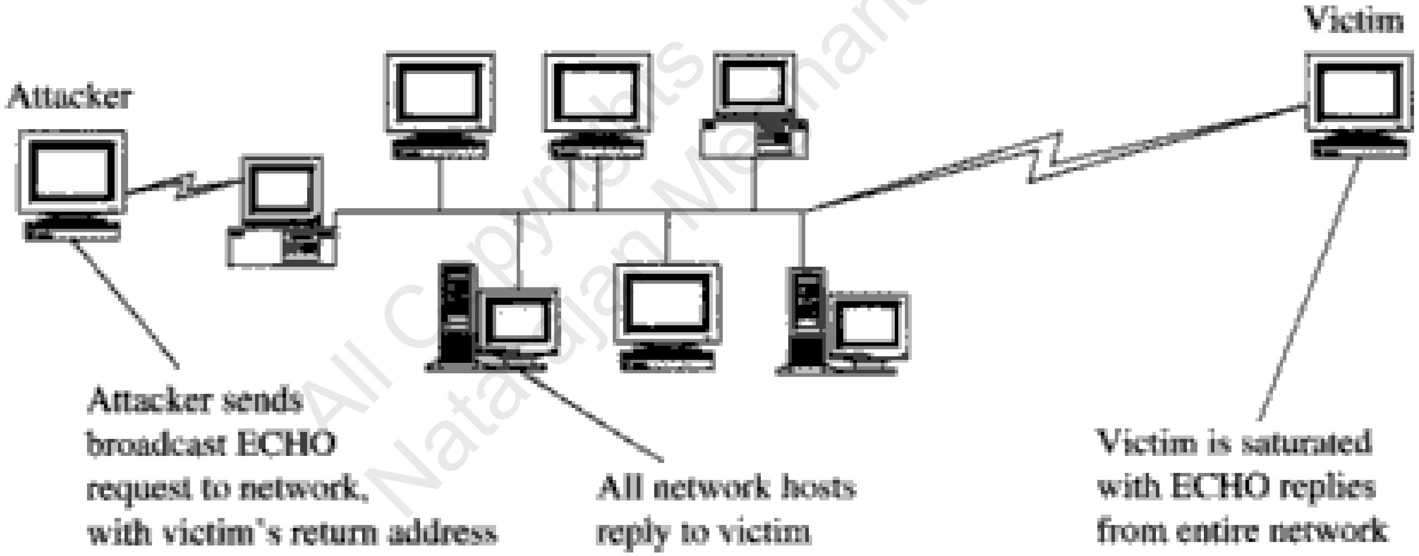
Random chars sent to UDP port 7

B

Characters echoed back to UDP port 19

Smurf Attack

- The smurf attack, a kind of denial-of-service attack, floods a target system via spoofed broadcast ping Echo-Reply messages.
- A perpetrator sends a ping Echo-Request message (having a spoofed address of the intended victim) to the broadcast IP address of a network.
- All the hosts in that network on receiving the ping message, send a reply to the source of the ping, which is the victim machine.



Attacker sends broadcast ECHO request to network, with victim's return address

All network hosts reply to victim

Victim is saturated with ECHO replies from entire network

- Solution: After the incidence of several Smurf attacks, routers in the Internet were configured not to forward packets having a broadcast IP address as the destination address. Hosts are also configured not to respond to ping requests that were sent to them as a broadcast message.

Teardrop Attack and Traffic Redirection

- The Teardrop attack involves sending IP fragments with overlapping, over-sized, payloads to the target machine.
- The attacker sends a series of datagrams to the target machine, such that the fragments cannot fit together properly.
- Example:
  - One datagram might say it is position 0 for length 60 bytes, another position 30 for length 90 bytes and another position 41 for length 173 bytes.
  - As the above three pieces overlap, they cannot be reassembled properly.
  - The OS locks up with these partial data units it cannot reassemble, thus leading to denial-of-service attacks.
- Modern operating systems are configured to discard reassembly when overlapping fragments arrive and a simple reboot could bring the system back to normality.

- Traffic Redirection: If a router is compromised, then it could advertise to all its neighboring routers that it lies on the shortest path to every other destination network in the Internet. Soon, all the traffic will be redirected to this router, the router is flooded, drops all the packets and they do not make it to their intended destination.

DNS Attacks

- A Domain name server (DNS) is a machine that holds a table mapping domain names to IP addresses.
- A DNS server queries other DNS servers to resolve domain names it does not know and updates its table with the mapping learnt.
- <u>DNS Cache Poisoning:</u> Is a technique that tricks a DNS server into believing it has received authentic information when, in reality, it has not.
- Once the DNS server has been poisoned, the information is generally cached for a while, spreading effect of the attack to users of the server.
- Example:
  - An attacker poisons the IP address DNS entries for a target website on a given DNS server, replacing them with the IP address of the server the attacker controls.
  - The attacker then creates fake entries on the server he/she controls with names matching those on the target server.
  - These files could contain malicious content, such as a worm or a virus.
  - A user whose computer has referenced the poisoned DNS server would be tricked into thinking that the content comes from the target server and unknowingly download malicious content.

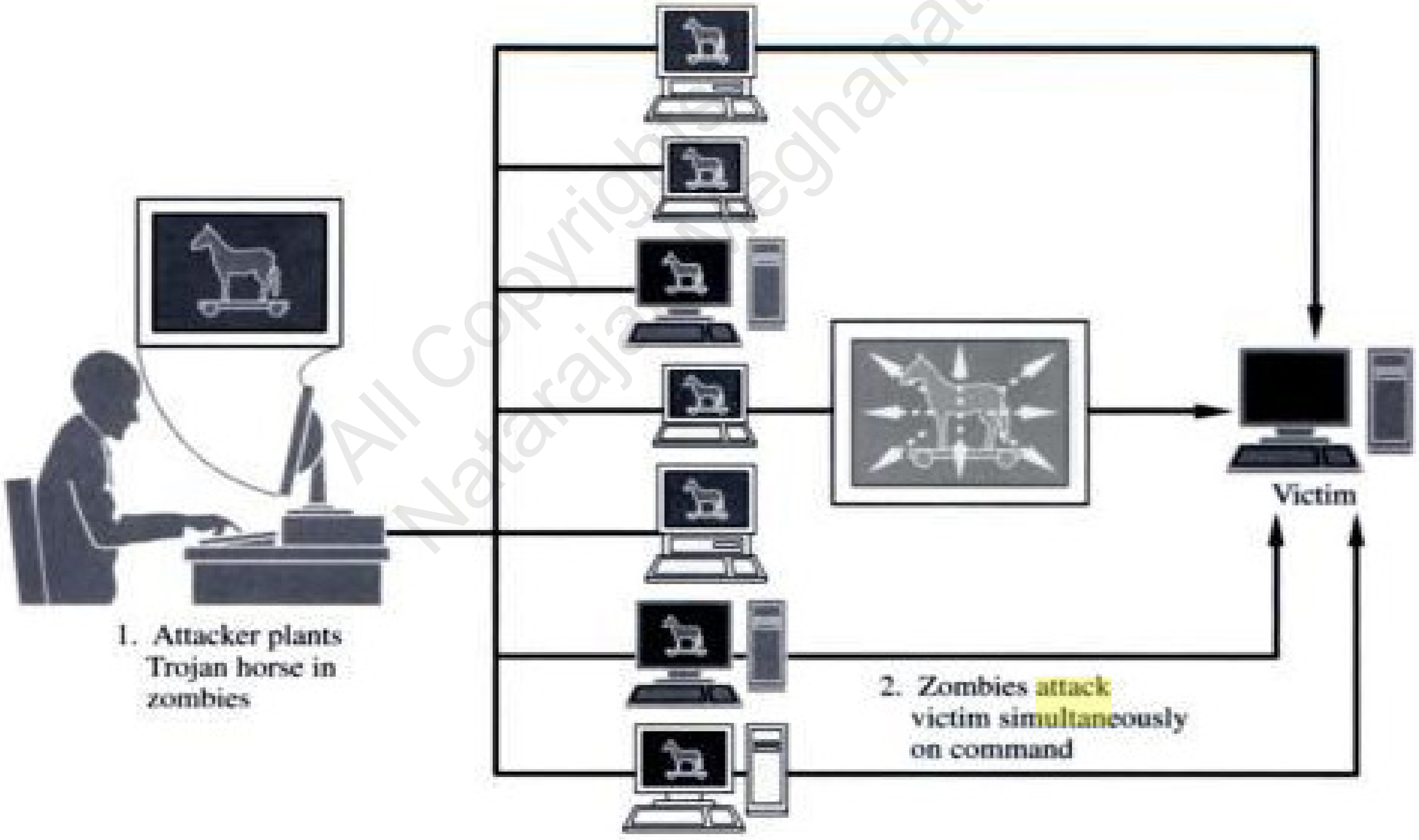
Distributed Denial-of-Service Attacks

- DDoS attacks involve breaking into hundreds or thousands of machines all over the Internet. Then, the attacker installs DDoS software on these burgled machines and controls them to launch coordinated attacks on victim sites.
- DDoS attacks typically exhaust bandwidth, router processing capacity and break network connectivity to the victims.

- First stage: Forming Zombies
  - The attacker uses any convenient attack (such as exploiting buffer overflow or tricking the victim to open and install unknown code from an email attachment) to plant a Trojan Horse on a target machine.
  - The attacker also installs software such as “rootkit” on these compromised machines. The rootkit helps to conceal the fact of the break-in, hide traces of subsequent malicious activities and also replaces the standard commands for displaying running processes with versions that fail to replace the attacker’s processes. The compromised victim machine is referred to as a “zombie”
  - A zombie could also be used to break into some more machines, install the Trojan Horse and rootkits and convert them to be a zombie. A network of zombies is called a botnet.

Distributed Denial-of-Service Attacks

- Second stage: Launch the DDoS attack through the zombies.
  - The attacker choose a victim and sends a signal to all the zombies to launch the attack. Each zombie could launch a different type of attack on the victim.
  - A victim of the DDoS attack will thus have to counter multiple zombies, launching the same or different types of attacks.

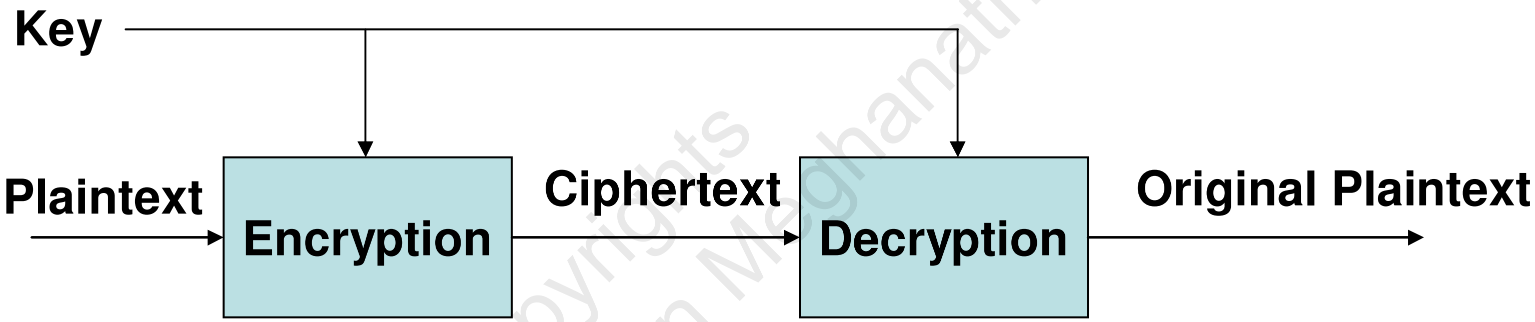


8.2: Defense using Cryptography

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

Types of Encryption

- Symmetric encryption: The same key performs, both encryption and decryption.
  - P = D(K, E(K, P))



Key

Plaintext

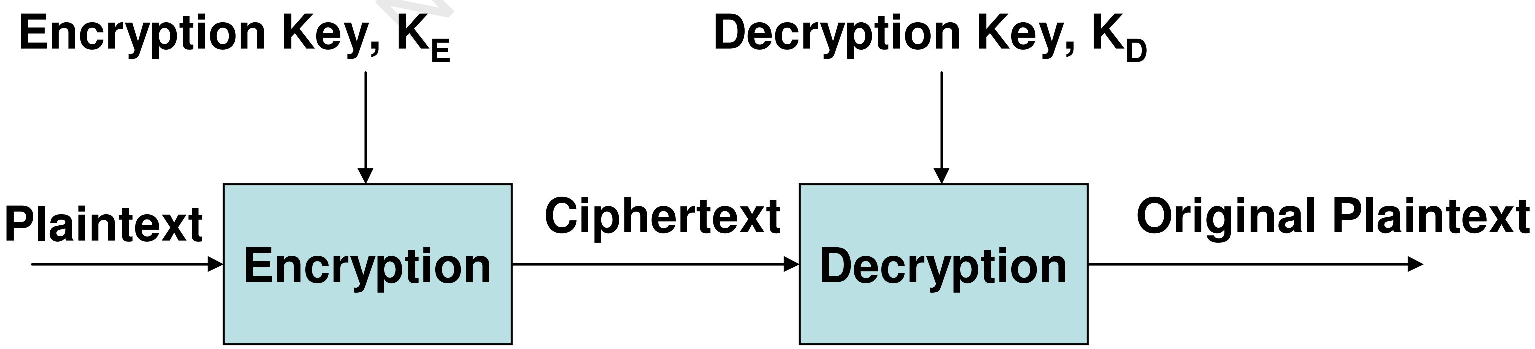
Encryption

Ciphertext

Decryption

Original Plaintext

- Asymmetric encryption: distinct, very different keys, one for encryption and the other for decryption only



Encryption Key, K_E

Decryption Key, K_D

Plaintext

Encryption

Ciphertext

Decryption

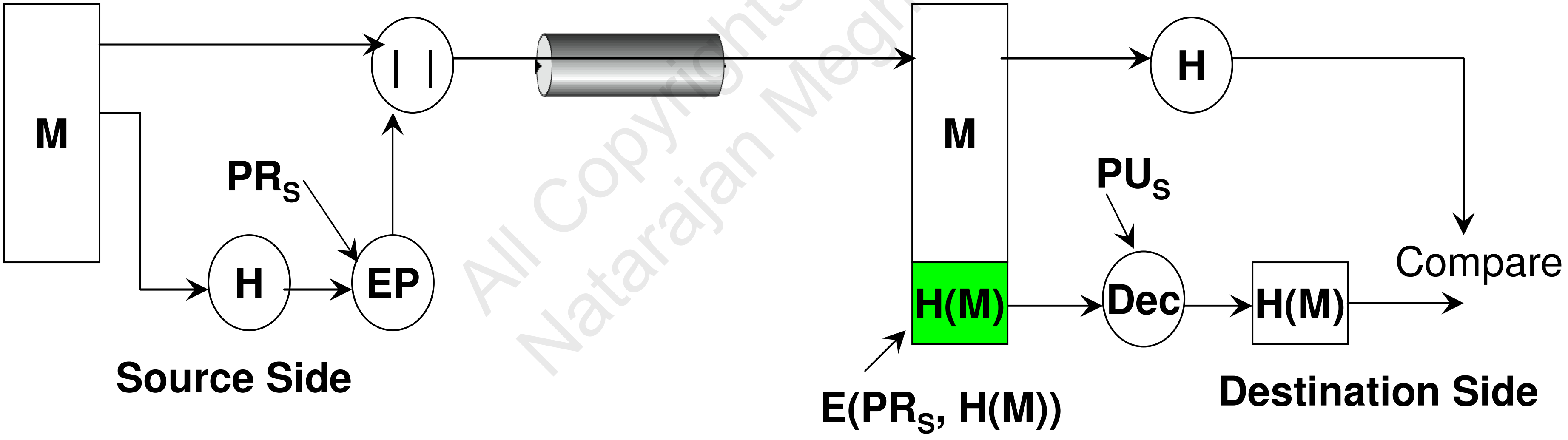
Original Plaintext

Public-Key Encryption

- Let K_{PRIV} and K_{PUB} be the private key and public key of a user. Then,
  - P = D(K_{PRIV}, E(K_{PUB}, P))
  - P = D(K_{PUB}, E(K_{PRIV}, P))
- <u>Exchange of Secret Message using Asymmetric Encryption</u>
  - Let K_{PUB-S}, K_{PRI-S} denote the public and private keys of Sender S. Similarly, let K_{PUB-R} and K_{PRI-R} be the public and private key of Receiver R. Let M be the secret message to be sent from S to R.
  - S sends to R the following:
    - E (K_{PUB-R} E(K_{PRI-S}, M))
  - The inner encryption guarantees that the secret message M came from S and the outer encryption guarantees that only the receiver R could open the outer encryption of the message and get access to the inner encryption.

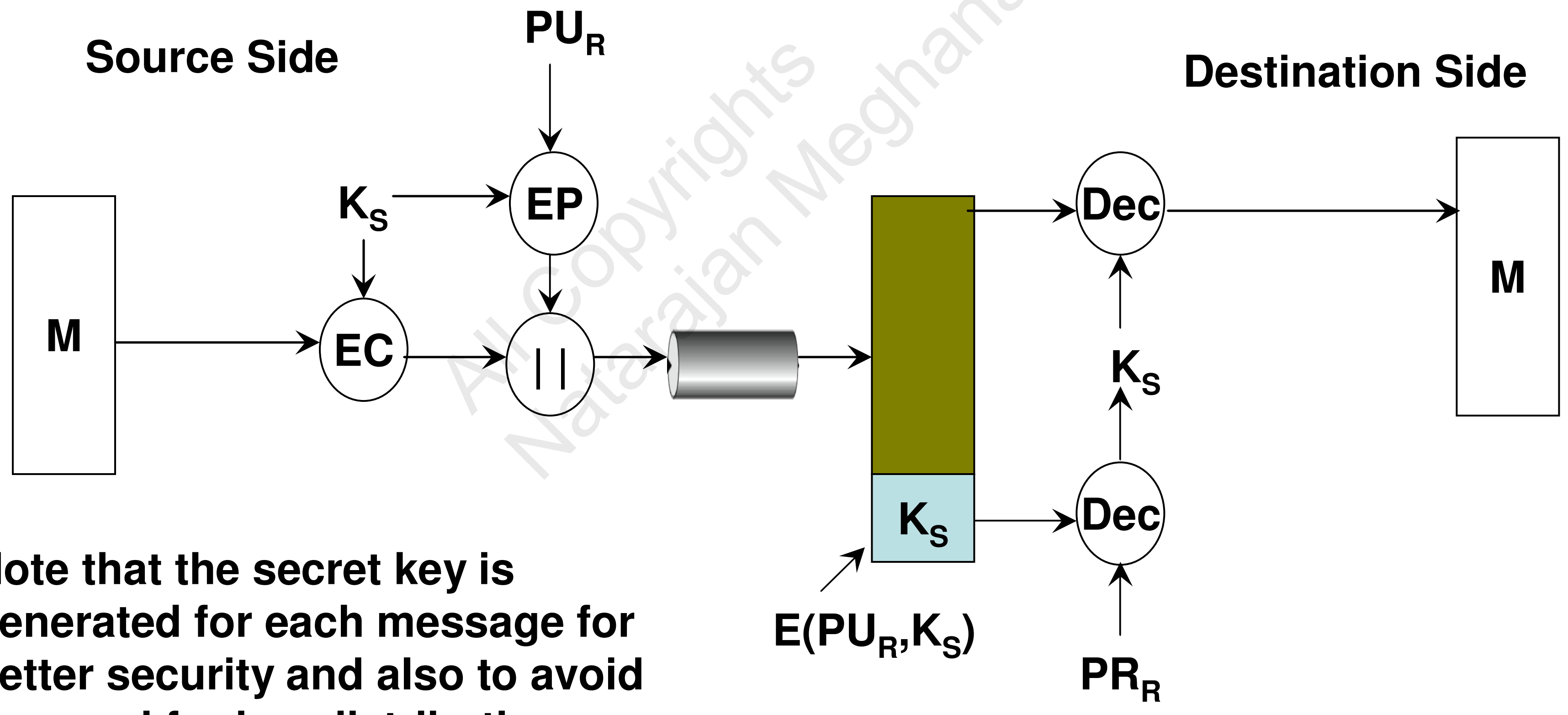
Use of Public-Key Encryption to Provide Integrity and Authentication

- $M \mid\mid E_{Pri\text{-}S}(\ Hash(M)\ )$



Use of Public-Key Encryption to Provide Confidentiality

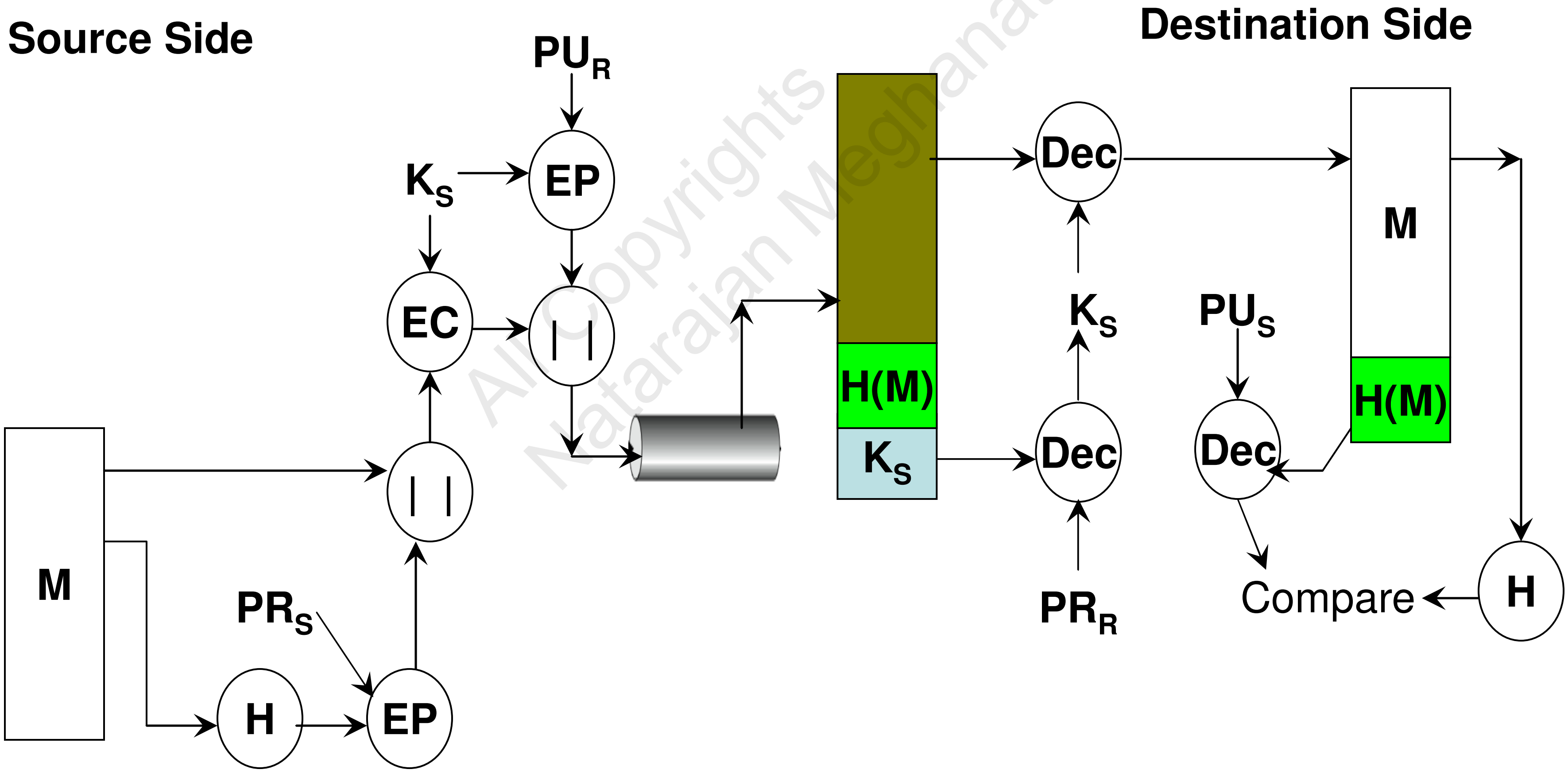
- $E_{\text{Secret-Key}}(M) \,||\, E_{\text{Pub-R}}(\text{Secret-key})$



Note that the secret key is generated for each message for better security and also to avoid the need for key distribution.

Use of Public-Key Encryption to Provide Confidentiality, Integrity and Authentication

$E_{Secret-Key}(M \,||\, E_{Pri-S}(Hash\ (M)\ )\ ) \,||\, E_{Pub-R}(Secret-key)$



Link Encryption vs. End-to-End Encryption

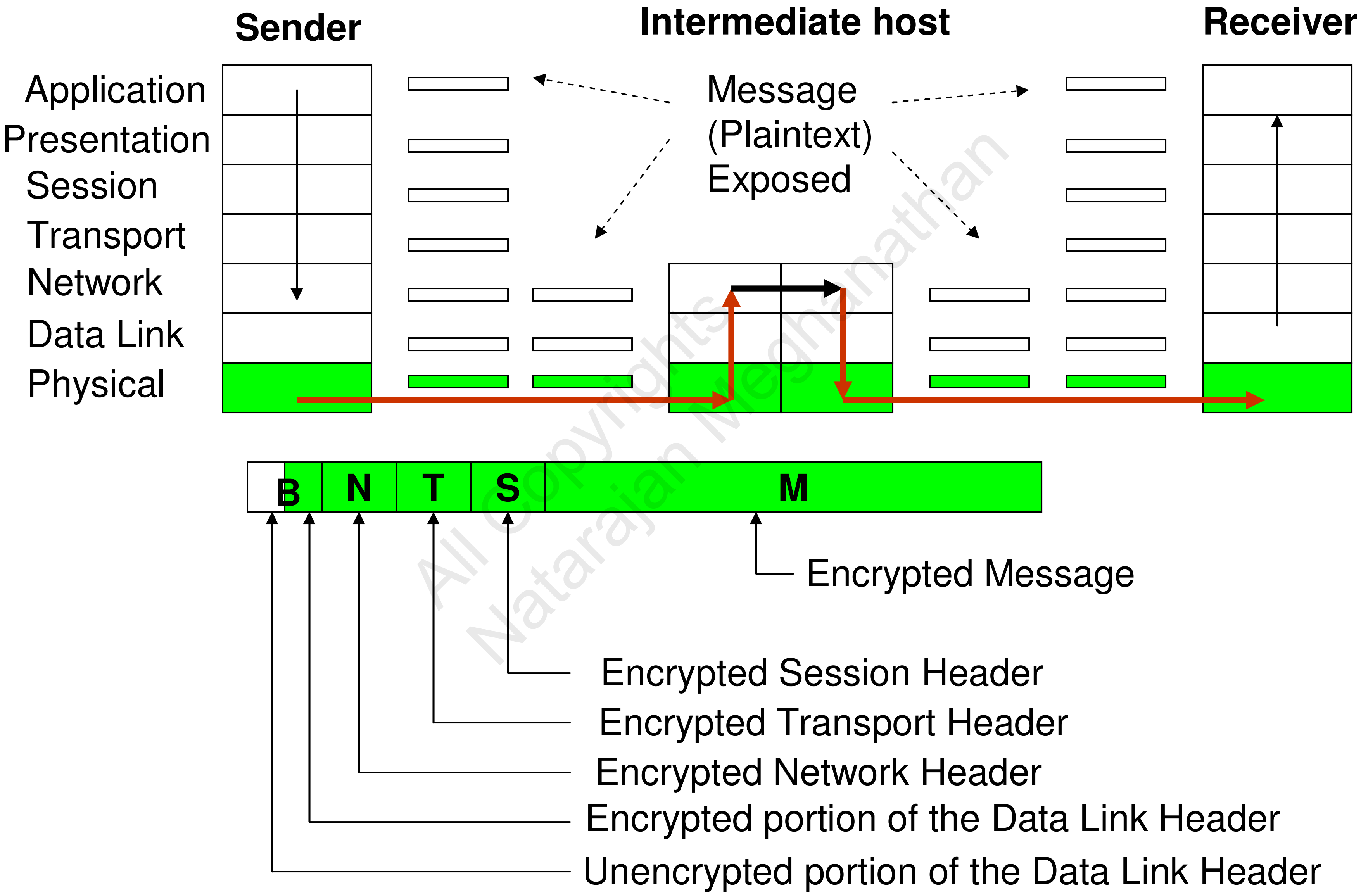
- Link Encryption: Encrypt a packet at every network on its way from the source network to the destination network.
  - Encryption handled at the network routers
  - Data is exposed at the intermediate routers
  - Every network on the source-destination path needs to support encryption; otherwise no use
  - Encryption keys are to be maintained for every pair of network routers.

- End-to-End Encryption: The application user encrypts the data
  - No need for the networks to support encryption
  - Data is not exposed at intermediate routers
  - Encryption keys need to be maintained on per-user basis

🟩 Message encrypted

□ Message in plaintext: Exposed

Link Encryption



Encrypted Message

Encrypted Session Header

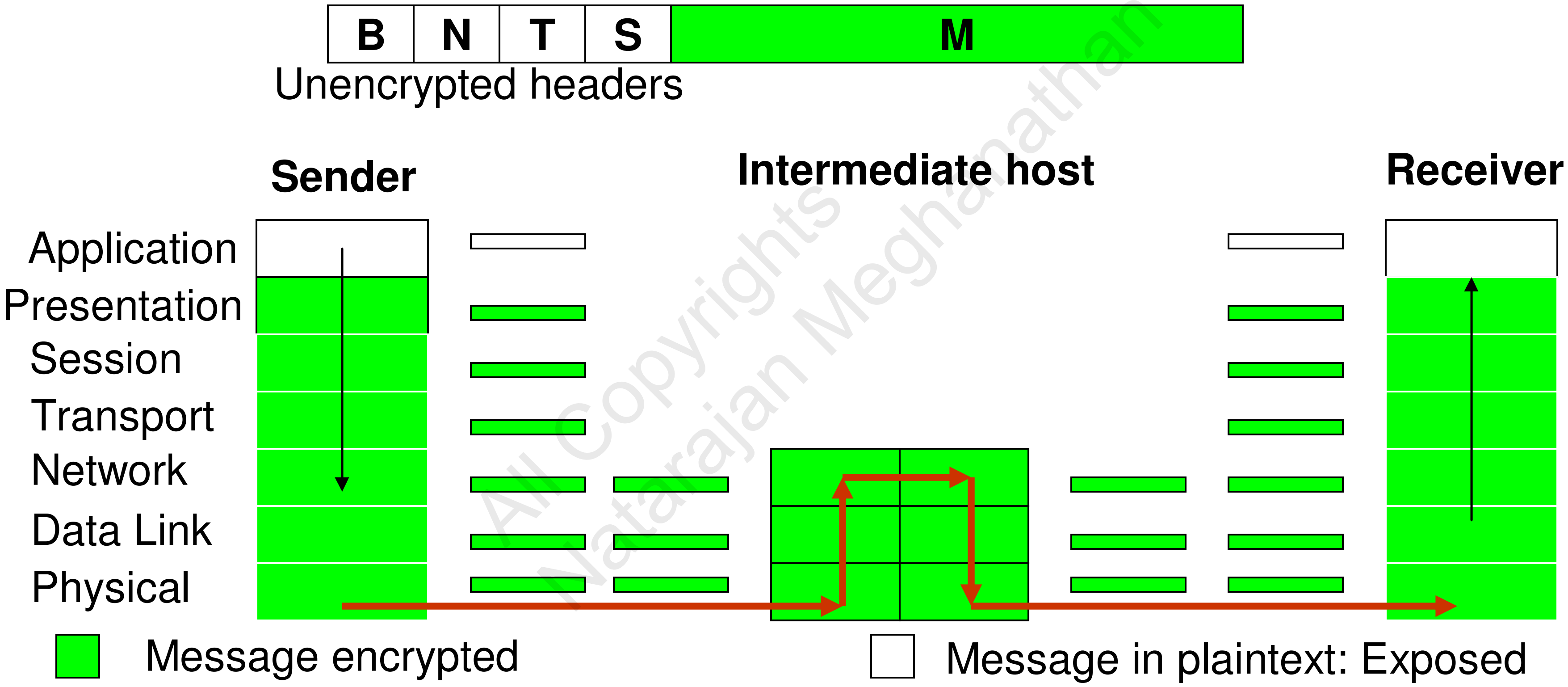
Encrypted Transport Header

Encrypted Network Header

Encrypted portion of the Data Link Header

Unencrypted portion of the Data Link Header

End-to-End Encryption



Man-in-the-Middle Attack

- Man-in-the-middle (MITM) attack is an attack in which an attacker is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised.
- The attacker must be able to observe and intercept messages going between the two victims.
- Example: (MITM attack on public-key cryptography)
  - Suppose Alice wishes to communicate with Bob.
  - Mallory wants to eavesdrop their conversation or also possibly deliver a false message to Bob.
  - First, Alice must ask Bob for his public key.
  - If Bob sends his public key to Alice, but Mallory is able to intercept it, a MITM attack can begin.
  - Mallory sends a forged message to Alice that claims to have come from Bob, but contains Mallory's public key
  - Alice believes the public key received to be that of Bob's. So, Alice encrypts the message she wishes to send to Bob using the public key received and transmits on the link to Bob.
  - Mallory could now intercept the message, decrypt it with his private key and get the actual contents of the message.
  - Mallory now again encrypts the message (could be even altered too) with Bob's public key and transmits the message to Bob.
  - Bob on receiving the message, decrypts the message with his private key and reads the contents of the message assuming it came from Alice

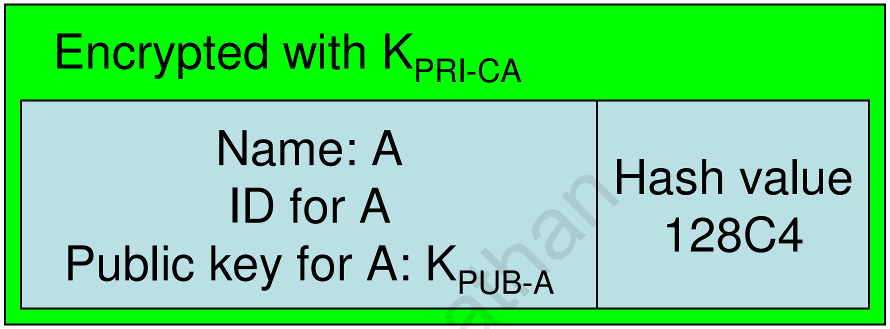
Public-Key Certificates

- Each of us adopt a “trust threshold” – a degree to which we are willing to believe an unidentified individual.
- We will use the concept of “vouching for” by a third party as the basis of thrust in settings where two parties do not know about each other.

- <u>Certification Authority (CA):</u> Is an entity that issues digital certificates that contain a public key and the identity of the owner.
- The CA attests that the public key contained in the digital certificate belongs to the person (CA is a sort of digital notary).

Certificates

Digital Certificate for the Public Key of A

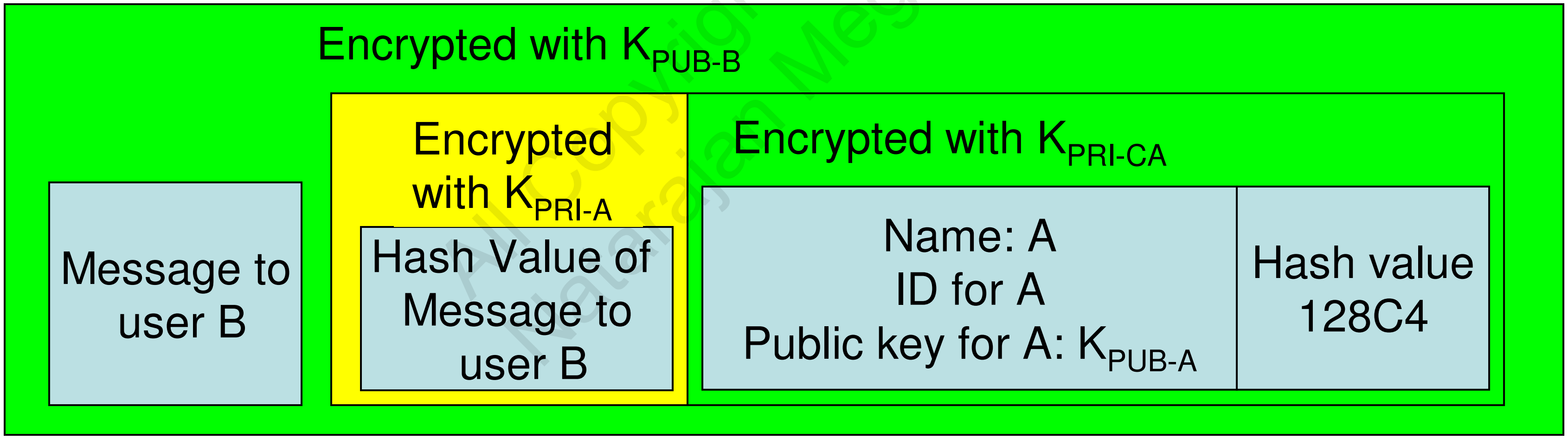


Encrypted with K_{PRI-CA}

Name: A
ID for A
Public key for A: K_{PUB-A}

Hash value
128C4

User A sending to user B



Encrypted with K_{PUB-B}

Message to user B

Encrypted with K_{PRI-A}

Hash Value of Message to user B

Encrypted with K_{PRI-CA}

Name: A
ID for A
Public key for A: K_{PUB-A}

Hash value
128C4

Note: The certificates are created and formatted based on the X.509 standard, which outlines the necessary fields of a certificate and the possible values that can be inserted into these fields. The latest X.509 version is v.3.

8.3: IPSec

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

IPSec

- Before two machines send the messages using their IP addresses, they have to establish an IPSec SA (Security Association)
- IPSec SA
  - The two hosts A and B exchange their public-key certificates (that has their IP address and public-key certified).
  - All further communication are encrypted with the public key of the receiver (so that it can be decrypted only by the receiver with its private key)
  - The two hosts A and B negotiate on the encryption and keyed-hashing algorithms to use for confidentiality and integrity + authentication respectively.
  - The two hosts establish a session key (for integrity and authentication) using the public-key encryption based Diffie-Hellman key exchange mechanism.
  - Using the session key, the two hosts can then establish a secret key for confidentiality in communication.
  - IPSec SA is unidirectional: If machines A and B want to send messages back and forth, they have to establish an IPSec SA in each direction.
  - An IPSec SA from A to B is said to be outbound at A and inbound at B.
  - An IPSec SA from A to B is identified by the tuple <SPI, IPaddress of A> where SPI is the Security Parameter Index value, locally unique at A. The combination of the SPI with the IP address of the host makes the tuple globally unique.

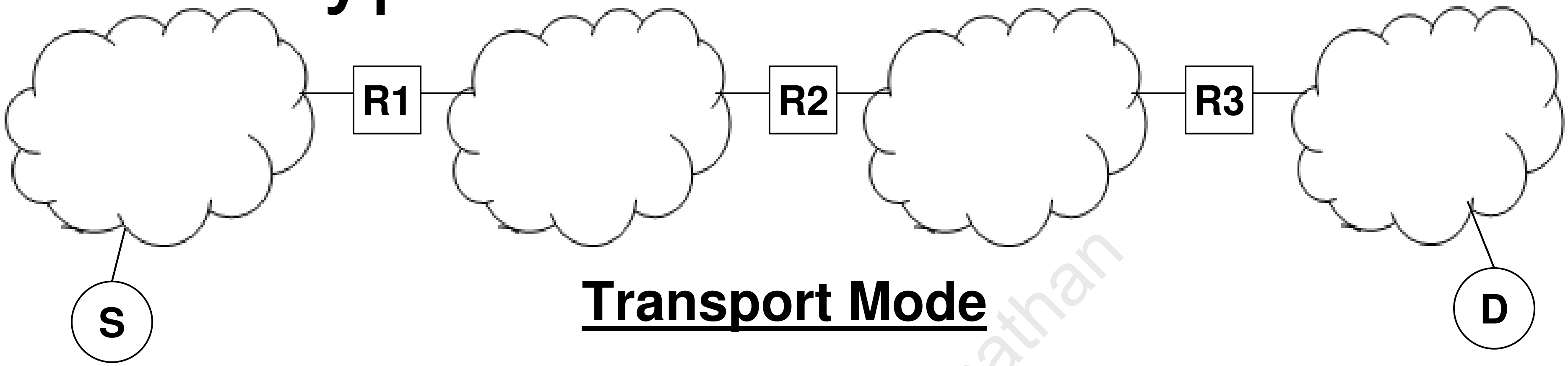
IPSec

- The IPSec header is inserted in between the IP header and transport layer header. There is no need for support from any higher layers.

IP Header	**IPSec Header**	**TCP/ UDP Header**	**Data**

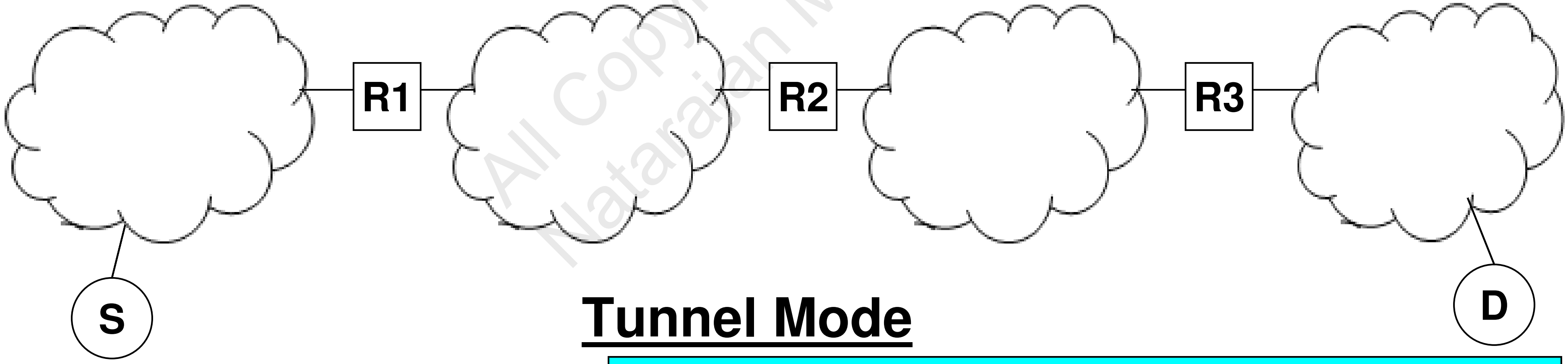
- IPSec headers:
  - Authentication Header (AH): used for integrity + authentication
  - Encapsulated Security Payload Header (ESP): used for confidentiality, integrity + authentication.
- IPSec Modes:
  - Transport Mode: When IPSec SA is directly established between the two end hosts. Message is secure all the way from the source host to the destination host
  - Tunnel Mode: When IPSec SA is established between the gateway routers of the two end hosts. Message is not secure in the source and destination networks. Need to use IP-in-IP encapsulation to encapsulate the IP datagram with the IP addresses of the two ultimate end hosts.

Typical IPSec Scenarios



Transport Mode

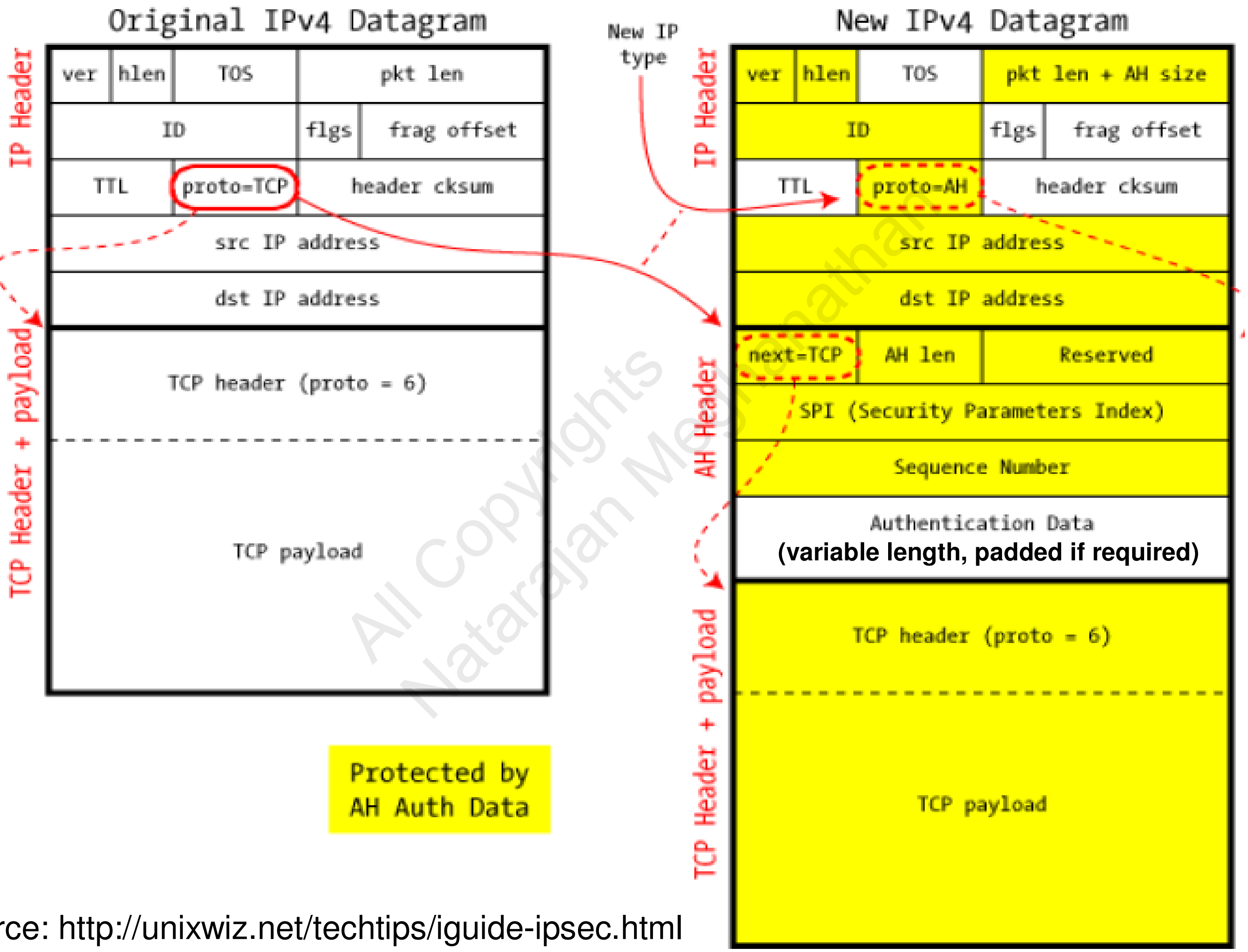
IP Header IP$_S$ IP$_D$	IPSec Header SA: S -- D	TCP Header	Data



Tunnel Mode

IP Header IP$_{R1}$ IP$_{R3}$	IPSec Header SA: R1 – R3	IP Header IP$_S$ IP$_D$	TCP Header	Data

IP4 Datagram with Authentication Header



Original IPv4 Datagram

IP Header

ver | hlen | TOS | pkt len

ID | flgs | frag offset

TTL | proto=TCP | header cksum

src IP address

dst IP address

TCP Header + payload

TCP header (proto = 6)

TCP payload

New IP type

New IPv4 Datagram

IP Header

ver | hlen | TOS | pkt len + AH size

ID | flgs | frag offset

TTL | proto=AH | header cksum

src IP address

dst IP address

AH Header

next=TCP | AH len | Reserved

SPI (Security Parameters Index)

Sequence Number

Authentication Data
(variable length, padded if required)

TCP Header + payload

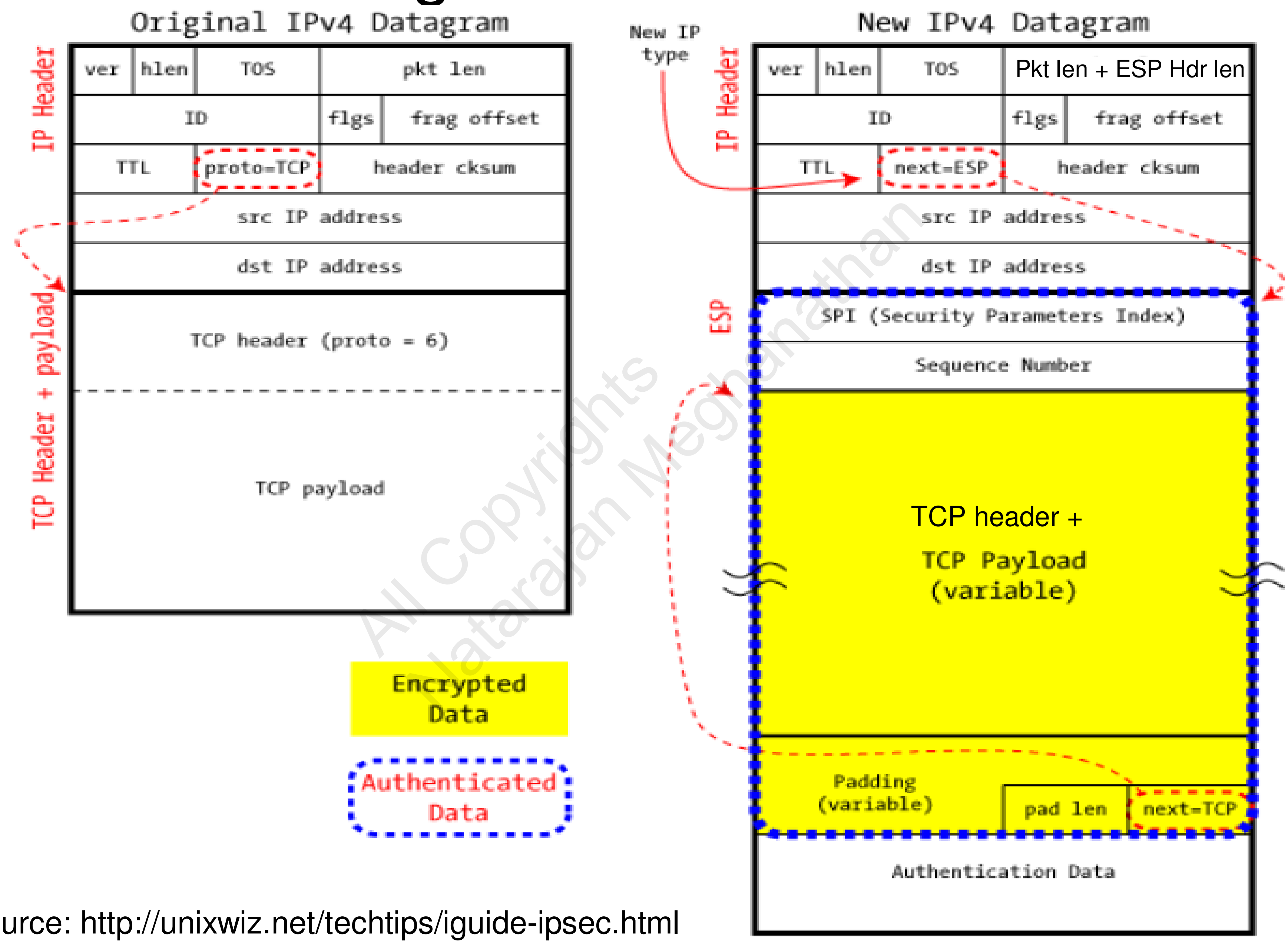
TCP header (proto = 6)

TCP payload

Protected by
AH Auth Data

Source: http://unixwiz.net/techtips/iguide-ipsec.html

IP4 Datagram with ESP Header



Source: http://unixwiz.net/techtips/iguide-ipsec.html

8.4 Firewalls and IDS

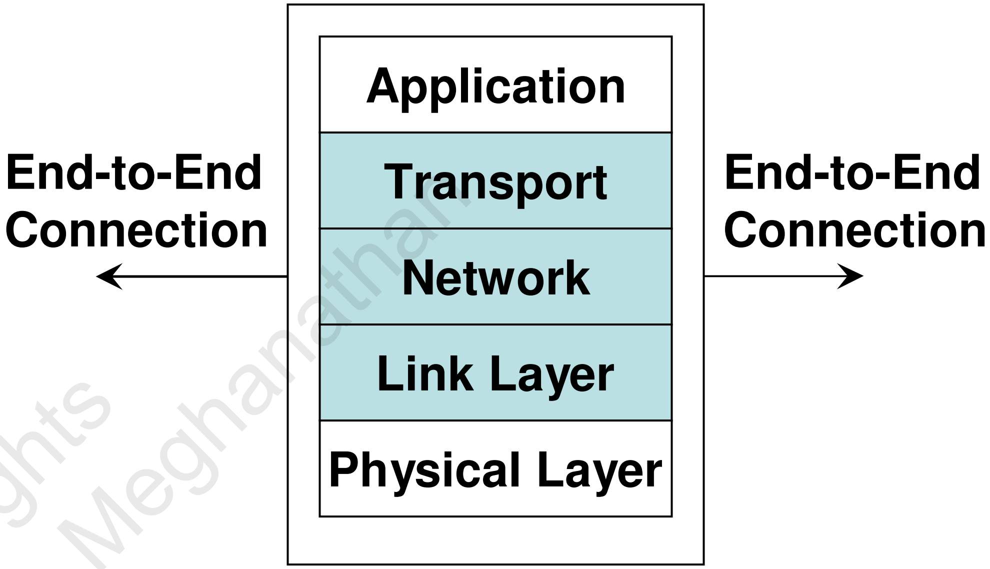
Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

Firewalls

- A firewall is a software running on a dedicated host computer on which no other application is run.
  - To prevent someone from changing the firewall rules by exploiting the vulnerabilities of the other applications that may be run on the host.
- A firewall is as good as it is configured with, depending on the needs of the admin.
- Allowable or non-allowable traffic are typically identified with source/destination IP/network addresses and ports.
- **Filtering:** Egress filtering (filter outgoing traffic); Ingress filtering (filter incoming traffic)
- **Typical Firewall designs:**
  - **Default-deny approach (white-list):** Have a list of allowable traffic and block the rest.
  - **Default-allow approach (black-list):** Have a list of non-allowable traffic and allow the rest.
  - A good design needs to have a hybrid of these two approaches

Packet Filters

- A packet filter firewall is a stateless firewall that looks at only the packet headers to decide whether or not to drop a packet.
  - Stateless: Does not keep track of the decisions taken on any packet.
    - The decision taken on a packet is independent of the decision taken on the preceding packets.



End-to-End Connection

Application

Transport

Network

Link Layer

Physical Layer

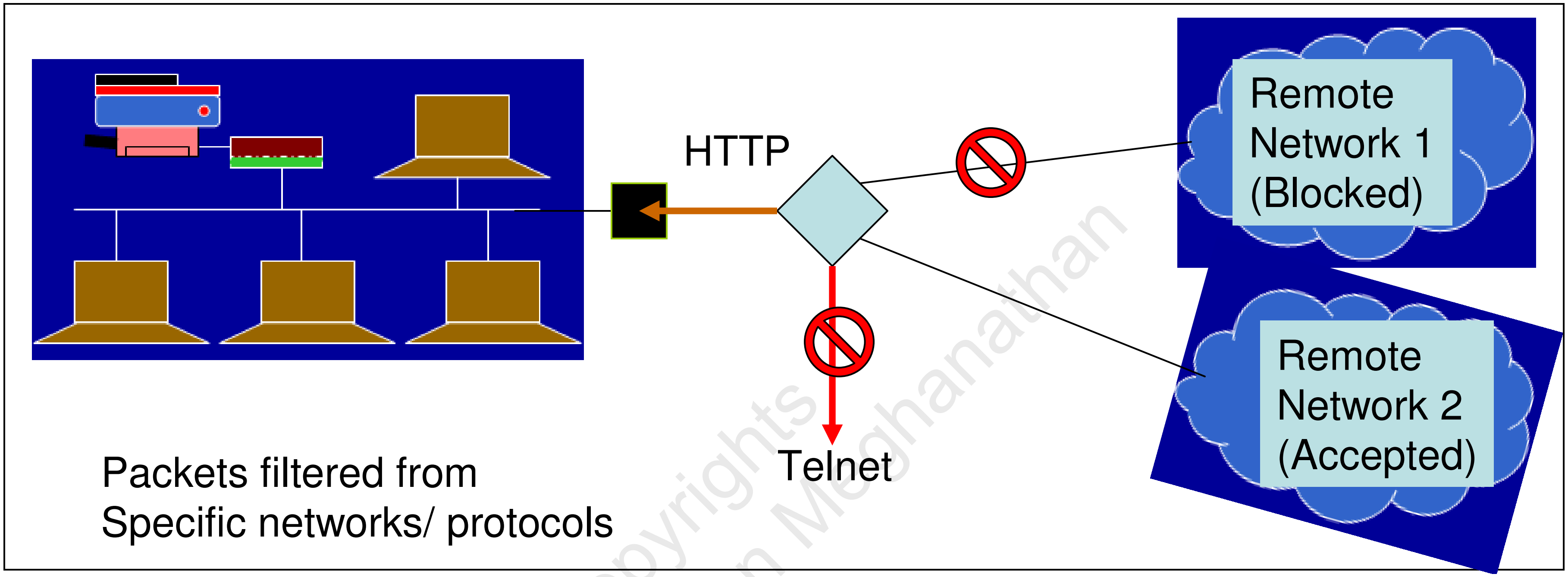
End-to-End Connection

Layers supported by Packet Filter and Stateful Firewalls

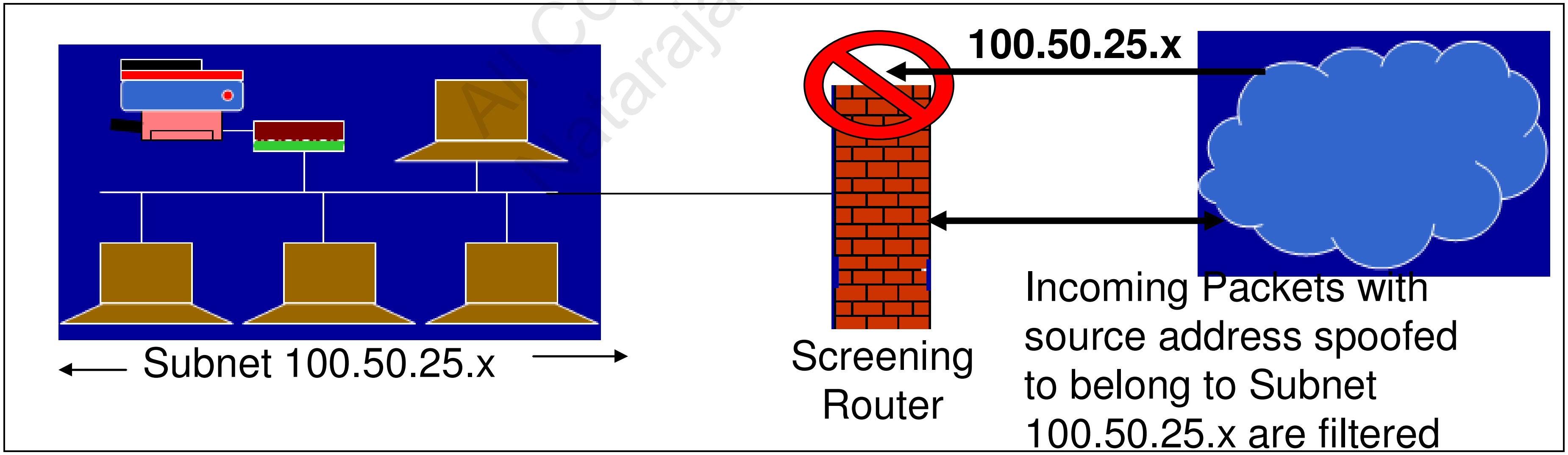
The code for packet filters will become lengthy as we want to block traffic belonging to specific networks, IP addresses and transport layer protocols.

Need efficient filtering algorithms

Packet Filters



Packets filtered from
Specific networks/ protocols



Incoming Packets with
source address spoofed
to belong to Subnet
100.50.25.x are filtered

Attacks Detected by Packet Filters

- **IP Spoofing Attacks:** Have the packet filter configured not to let in packets having a source address that corresponds to the internal network.
  - For example, the attacker has spoofed the source IP address to be the IP address of a machine belonging to the network being protected by the firewall.

- **Source routing attacks:** where source specifies the route that a packet should take to bypass security measures, should discard all source routed packets

- **Tiny fragment attacks:** intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into fewer separate fragments to circumvent filtering rules needing full header info; can enforce minimum fragment size to include full header.

Stateful Inspection Firewalls

- Stateful firewalls (also called circuit firewalls) examine the contents of each packet with regards to their placement within the packet series belonging to a specific session/connection and maintain records of all connections passing through the firewall.

- **Attacks and actions prevented with Stateful Inspection Firewalls**
- **Session Hijacking Attack**: Stateful firewalls can remember the sequence numbers expected on both sides as part of a TCP session and can block attempts to hijack the session, when an intruder sends several TCP segments with different sequence numbers (trial-and-error).
- **SYN Flood Attack**: Stateful firewalls can remember the number of connection requests that have been let through for an IP address/TCP port and block connection requests beyond a threshold.
  - Also, do not let more than a certain number of simultaneous TCP connections to originate per (source) IP address.
- **Bandwidth Exhaustion**: Do not let more than a specific amount of data to be transferred per day from the inside network to any outside IP address.
- **ICMP-based Attacks:** The volume of ICMP packets (like Echo Reply/Request, Destination not reachable, etc) that are transmitted in and out of the networks should be within a threshold.

Application Firewall

- The packet filter and stateful firewall look at only the packet headers. The application proxy firewall scans through the entire packet (including the application data) and makes sure if it could be forwarded in/out.
- An application firewall protecting an internal network of clients from being attacked by an external server/user is called a Proxy Firewall.
  - Example: An application firewall that protects an internal network of desktop/ office machines from users attempting to connect after office hours.
- An application firewall protecting an internal network of servers from being attacked by an external client is called a Reverse Proxy Firewall
  - Example: A reverse proxy firewall hosted to protect a sales network (comprising of various servers – database server, file server, etc) monitors every incoming packet to make sure it does not have any malicious scripts to cause any command injection attacks (XSS, XSRF or SQL-injection) or buffer overflow attacks.

Personal Firewalls

- Motivation: Home users, individual workers, and small businesses use cable modems or DSL connections with unlimited, always-on access.
- These people need a firewall, but a separate firewall computer to protect a single workstation can seem too complex and expensive.
- A workstation could be vulnerable to malicious code or malicious active agents (ActiveX controls or Java applets), leakage of personal data stored in the workstation, and vulnerability scans (like nmap) to identify potential weaknesses.
- A personal firewall is an application program that runs on a workstation to screen traffic on the workstation and block unwanted traffic leaving or entering the workstation to the network to which it is connected.
- A user could configure the personal firewall to accept traffic only from certain sites, and not from specific sites, and to generate logs of activities happened in the past
- A personal firewall could be also configured with a virus scanner which would be then automatically invoked to scan any incoming data to the workstation.

What Firewalls Can and Cannot Block

- Firewalls cannot alone secure an environment.
- A firewall protects only the perimeter of its environment against attacks from outsiders who want to execute code or access data on the machines in the protected environment.
- Firewalls cannot protect from internal threats (through disgruntled employees).
- Firewalls cannot protect against malware imported via laptop, PDA, or portable storage device infected outside the network, then attached and used internally.
- Firewalls can be held responsible for any security breach in if they are the only means to control the entire network perimeter.
  - If a host in the inside network has a connection to the outside network through a modem, the whole of the inside network is exposed to the outside network through the modem and the host. A firewall cannot be responsible for any attack
- A firewall is often a single point of failure for a network.
  - A more layered approach like a screening router, followed by a proxy firewall, followed by a personal firewall may be more helpful.
- Firewalls must be frequently configured and updated to take into account the changes in the internal and external environment and based on the review of the firewall activity reports that may indicate intrusion attempts.
- The machine hosting the firewall code will not have any other software like an editor, compiler, etc. in order to reduce the chances of an attack.

Demilitarized Zone (DMZ) Networks

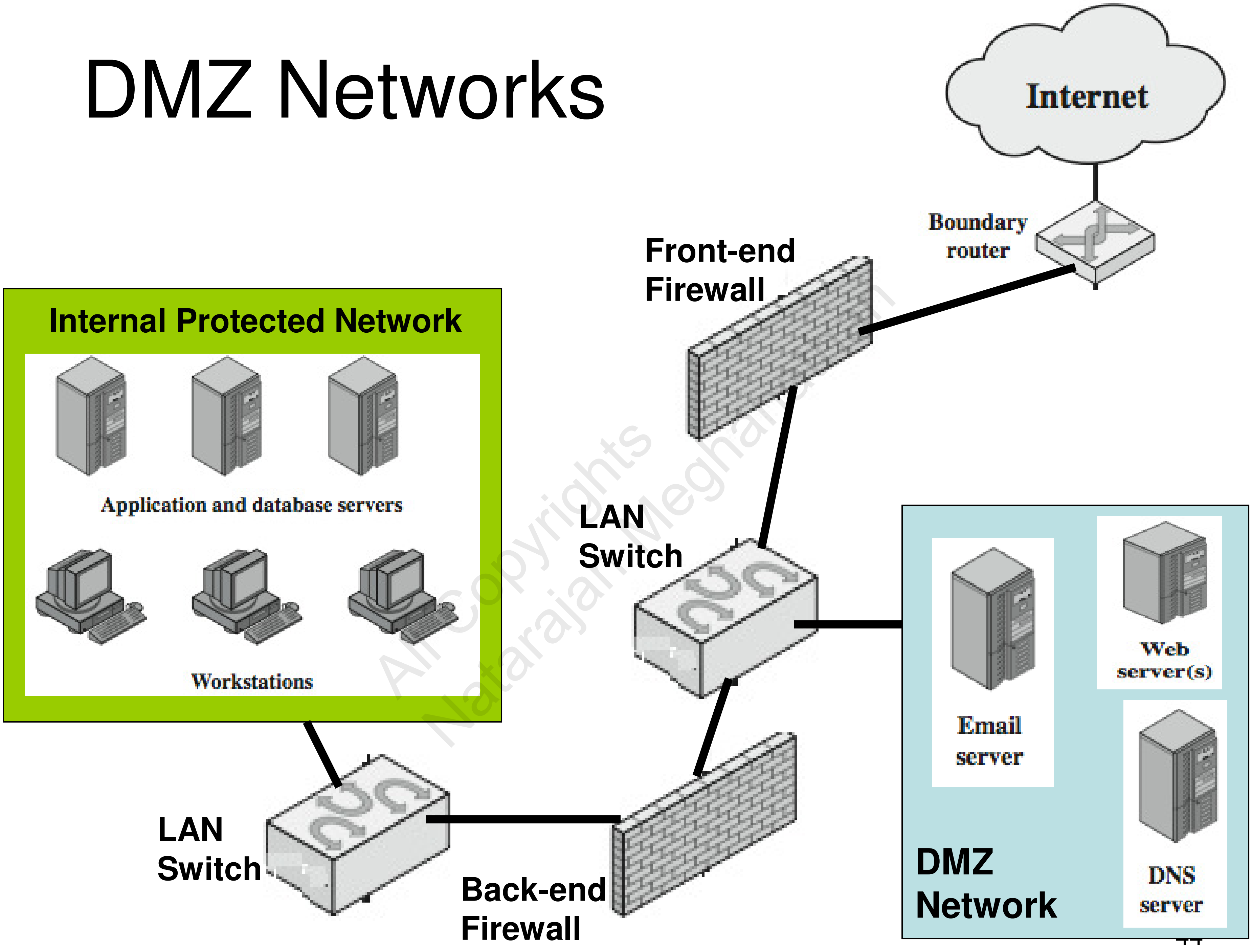
- A DMZ network (also called perimeter network) is a subnet that contains an organization’s services that are exposed to a larger untrusted network (like the Internet).

- In other words, the DMZ comprises of hosts that provide services to users outside the internal LANs, such as e-mail, web, DNS servers.
- Because of the higher chances of these hosts being compromised, they are placed into their own sub-network in order to protect the rest of the network if an intruder were to succeed in attacking them.

- Thus, a DMZ network adds an additional layer of security to an organization’s LAN – an external attacker only has access to the hosts in the DMZ and not to any other internal networks.

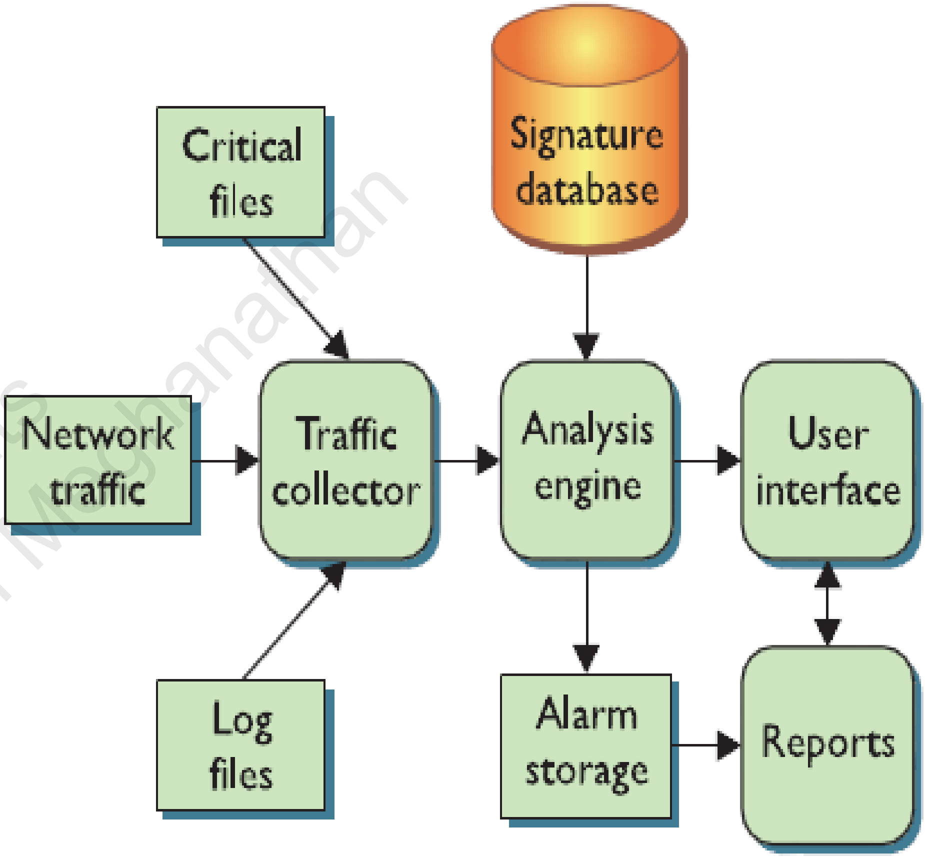
- Hosts in the DMZ provide services to both the internal and external networks – an external (“front-end”) firewall monitors the traffic between the DMZ network and the external Internet; while, an internal (“back-end”) firewall monitors the traffic between the DMZ hosts and the internal network clients.

DMZ Networks



Intrusion Detection Systems (IDS)

- An IDS to the networking world is like a burglar alarm to the physical world.
- The main purpose of an IDS is to identify suspicious or malicious activity, note activities that deviate from normal behavior, catalog and classify the activity, and, if possible, respond to the activity.
- Host-based IDS (HIDS): It examines activities on an individual system and not concerned with other systems or the network.
- Network-based IDS (NIDS): It examines activity (traffic) crossing the network it is monitoring and not concerned about individual systems.



Logical Depiction of IDS Components

Source: Figure 13.2 from Conklin and White – Principles of Computer Security, 2nd Edition

Signature and Anomaly-based IDS

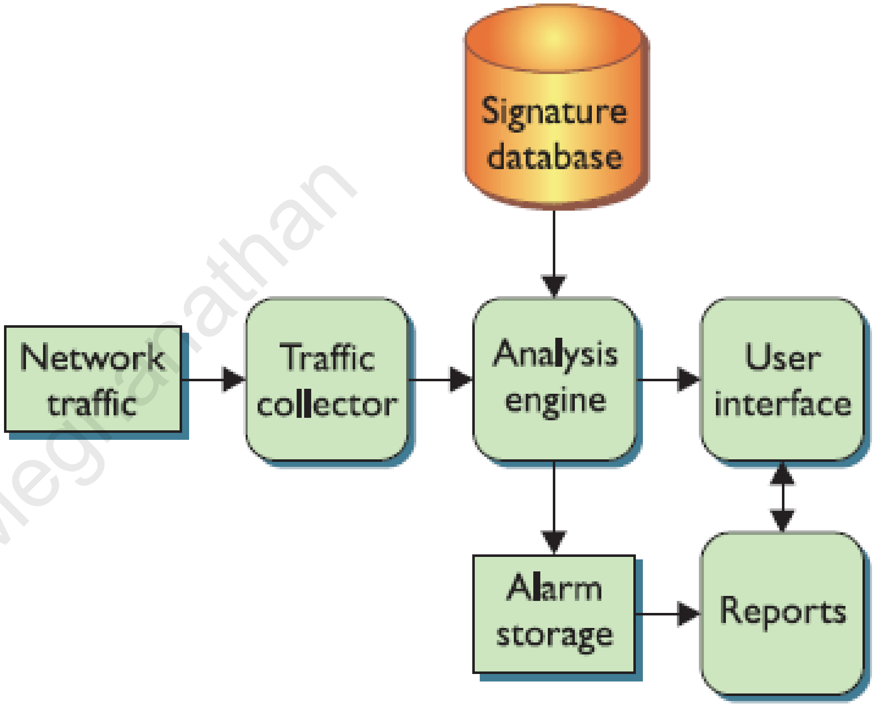
- Based on the approach adopted to detect suspicious or malicious traffic, IDS could be categorized into Signature-based and Anomaly-based IDS.
- *Signature-based IDS*: Relies heavily on a pre-defined set of attack and traffic patterns called signatures.
- A signature-based IDS (like an anti-virus software) can only match against known patterns – if a new attack comes in that the signature-based IDS has never seen before, it would not be able to identify it as suspicious or malicious – a primary weakness of signature-based IDS.
- *Anomaly-based IDS*: Monitors activities and attempts to classify them as either “normal” or “anomalous” (suspicious and unknown) based on self-created rule sets.
- An anomaly-based IDS uses heuristic techniques to categorize and classify traffic while developing and refining their internal rule sets.
- An advantage with anomaly-based IDS is that it can potentially detect new attacks or variant of old attacks.
- A drawback of anomaly-based IDS is that it could generate a potentially high number of false positives while the system is learning what “normal” is. Hence, such IDS should be programmed to dynamically adapt to changes.

False Positives and False Negatives

- When an IDS matches an activity to a specific pattern and generates an alarm for a non-malicious traffic that is not a threat, it is called a false positive.
- Technically, the IDS is functioning correctly by matching the pattern and has no ability to determine the intent behind the activity; but, from a human standpoint, this is not an information the analyst needed to see, as it does not constitute a threat and does not require intervention.
- Hostile activity that does not match an IDS signature and goes undetected is called a false negative.
- Note that if an IDS is limited by its signature set, it can match only activity for which it has stored patterns.

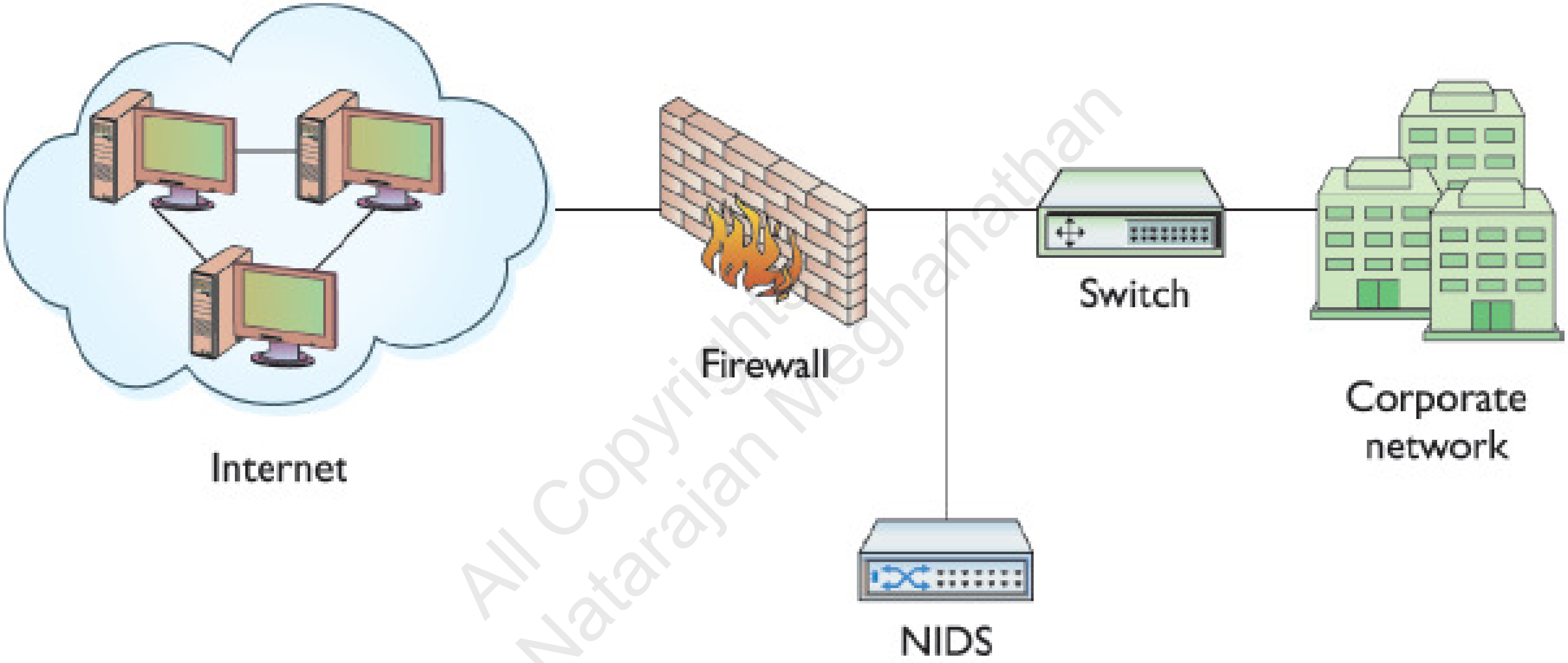
Network-based IDS (NIDS)

- NIDS are placed next to the firewall on the network perimeter and analyze the traffic as it passes by for the protocols, source, destination, content, traffic already seen and etc.
- A NIDS looks for traffic that typify hostile actions or misuse, such as the following:
  - Denial-of-service attacks, Port scans or sweeps, Malicious content in the data payload of a packet or packets, Vulnerability scanning, Trojans, Viruses, Worms, Tunneling and Brute-force attacks.
- The traffic collector of a NIDS logically attaches itself to a Network Interface Card (NIC) that operates in promiscuous mode (stealth mode) and sniffs the passing traffic.



Source: Figure 13.4 from Conklin and White – Principles of Computer Security, 2nd Edition

NIDS Placed behind Firewall



Source: Figure 13.7 from Conklin and White – Principles of Computer Security, 2nd Edition

NIDS: Advantages and Disadvantages

- Advantages of a NIDS
- Less Overhead: With a few well-placed NIDSs, one can monitor the entire network traffic going in and out of the organization. Also, upgrading and maintaining a fewer number of NIDSs is usually much cheaper than upgrading and maintaining hundreds of host-based IDSs.
- Big Picture: The collection of the few NIDSs can have visibility into all the network traffic and can correlate attacks (whether they are widespread or concentrated, unorganized or focused) among multiple systems.
- Disadvantages of a NIDS
- A NIDS is ineffective when traffic is encrypted.
- A NIDS cannot see traffic that does not cross it – If a NIDS is placed only in the perimeter, chances are that it could miss traffic traversing the internal network.
- A NIDS must be able to handle high volumes of traffic (even 1-Gbps is common nowadays) with the availability of networks with larger bandwidth.
- A NIDS does not know about activities on the hosts themselves.

Active vs. Passive NIDS

- Passive NIDS:
- A passive NIDS simply watches the traffic, analyzes it and generates alarms.
- It does not interact with the traffic itself in any way, and it does not modify the defensive posture of the system to react to the traffic.
- Active NIDS:
- An active NIDS contains all the same components and capabilities of the passive NIDS with one critical addition – the active NIDS can react to the traffic it is analyzing.
- The reactions of an active NIDS could range from something simple, such as sending a TCP reset message to interrupt a potential attack and disconnect a session, to something complex, such as dynamically modifying firewall rules to reject all traffic from specific source IP addresses for the next few hours or days.
- Active NIDS are also referred to as Intrusion Prevention Systems (IPSs). When configured with the private keys of the servers in the internal network, IPSs would be able to decrypt the SSH connection establishment messages between a client and server and extract the session keys that would be used during the complete session. This gives an added advantage for the IDS/IPS to handle encrypted traffic.

Host-based IDS (HIDS)

- A host-based IDS (HIDS) examines log files, audit trails (both generated by the local operating system), and network traffic coming into or leaving a specific host.
  - On UNIX systems, the examined logs are those created by syslog, kernel logs and error logs; On Windows systems, the examined logs are the event logs – Application, System and Security.
- Critical files are those that are vital to the system’s operation or overall functionality. They may be program (or binary) files, files containing user accounts and passwords, or even scripts to start or stop system processes.
- Any unexpected modifications (for e.g., could be detected using checksum) to the critical files could mean the system has been compromised or modified by an attacker. By monitoring these critical files, the HIDS can warn users of potentially malicious activity.
- Within the log files, the HIDS is looking for certain activities that typify hostile actions or misuse, such as the following:
  - Logins at odd hours, Login authentication failures, Additions of new user accounts, Modification or access of critical system files, Modification or removal of binary files (executables), Privilege escalation

Packet Sniffer (Protocol Analyzer)

- Packet Sniffer: Is a computer software (or even a computer hardware programmed to) intercept and log traffic passing over a LAN.
- A packet sniffer may be used for both beneficial and malicious purposes:
  - Analyze network problems and monitor network usage
  - Gather and report network statistics
  - Filter suspect content from network traffic
  - Spy on other network users and collect sensitive information such as passwords
  - Reverse engineer (study using the structure of the different packet headers) the protocols used over the network
  - Detect network intrusion attempts
  - Gather information for effecting a network intrusion
- In order to capture all the network traffic, the Network Interface Card (NIC) on the IDS hosting the packet sniffer should run in promiscuous mode and analyze every packet crossing the wire.
- Most switches come with SPAN (Switched Port Analyzer) port – a mirrored port that will see all the traffic passing through the switch or through specific virtual LANs. Packet sniffers can be run on the SPAN port of a switch.

Honeypot

- A honeypot is a trap to detect, deflect or in some manner counteract attempts at unauthorized use of information systems. –
- A honeypot is usually a computer, and sometimes data or an unused IP address space that appears to be part of a network but which is actually isolated, unprotected and monitored, and which seems to contain information or a resource that would be of value to attackers.
- Honeypots have no production value and hence should not see any legitimate traffic or activity. Whatever they capture can be surmised as malicious or unauthorized.
- A honeynet is a network of honeypots. A honeynet is used for monitoring a larger and/or more diverse network in which one honeypot may not be sufficient.
- A honeypot/ honeynet is more of a preventative approach of detecting potential attackers existing in the Internet who may target the organization network in the near future.
- Honeypots could be used to fake as open relays to attract spam emails and determine the source e-mail address and destination e-mail addresses used by the spammers.
  - An open relay is an e-mail server that allows anyone on the Internet to send email through it.
  - Once they find an open relay, spammers keep sending the span email to the open relay and expect it to spread the spam.
- Note that no ordinary e-mail will come to a honeypot. All it receives could be categorized as spam.