Day

Date

CHAPTER - 09

• THE NEED FOR FIREWALLS

↳ EVOLUTION OF ORGANIZATIONAL NETWORKS

↳ Centralized Systems : One mainframe with connected terminals.

↳ LANs : PCs & terminals connected to each other & the mainframe

↳ Premises Networks : Multiple LANs with PCs, servers and mainframes.

↳ Enterprise Networks : Geographically distributed premises linked via pvt WANs.

↳ Internet Connectivity : Premises networks connected to the internet

↳ Enterprise Cloud Computing : Virtualized servers in data centers providing internal & public services

→ Why Internet Access is a Security Risk

① Internet access is essential for organizations & users

② Users may by pass restrictions using wireless broadband if LAN access is blocked

③ Internet connectivity allows external entities to

Date _____ Day _____

interact with internal sys, creating security threats.

→ **Limitations of Host Based Security Alone**

↳ Large networks have hundreds or thousands of systems

↳ Multiple OS increases complexity

↳ Security flaws require patching every affected systems

↳ Requires scalable configuration management & aggressive patching

↳ Host-only security is costly & difficult to manage at scale

→ **Role & Purpose of A Firewall**

↳ A firewall complements host based security

↳ Placed b/w the premises networks & the Internet

↳ Creates a controlled access point between internal & external networks

↳ Act as a security perimeter (outer-wall)

↳ Protects internal sys from Internet-based Attacks

↳ Provides a single choke point for:

↳ Security enforcement

↳ Traffic control

↳ Auditing & monitoring

Date Day

↳ A firewall can be:
  ↳ A single system, or multiple cooperating sys
↳ Provides an additional layers of defense

- **FIREWALL CHARACTERISTICS AND ACCESS POLICY**

→ **Design Goals**

(1) All traffic must pass through the firewall
  ↳ No direct access to the internal network
  ↳ Firewall ⇒ only entry/exit point

(2) Only Authorized traffic is allowed
  ↳ Traffic is permitted based on the orgs security policy
  ↳ Different firewall types enforce diff policies

(3) Firewall must be secure itself
  ↳ Hardened system with a secured OS
  ↳ Must resist attacks & penetration

Date _____ Day _____

→ Firewall Access Policy

↳ Defines what traffic is allowed or denied

↳ Based on:

- Organization's risk assessment
- Info security policy

↳ Sepecifies

- IP Adresses
- Protocols
- Applications
- Content Types

↳ Starts with a broad set of rules → refined into detailed filtering rules

→ Traffic Filtering Criteria

① IP Address & Protocol Filtering

- Based:
  - ↳ Src / dest IP
  - ↳ Port #
  - ↳ Inbound or Outbound traffic
- Used by
  - ↳ Packet-filtering firewalls
  - ↳ Stateful inspection firewalls
- Commonly limits access to specific services

Date _____ Day _____

② Application Protocol Filtering
- Filters based on application-level ~~gateways~~ data
- Used by application-level gateways
- Examples:
  - ↳ Checking SMTP email for spam
  - ↳ Allowing HTTP access to provide approved websites only

③ User Identity Filtering
- Access based on user authentication
- Mostly for internal users
- Uses secure authentication

④ Network Activity Filtering
- Based on traffic behaviour
  - Time of access (business hrs only)
  - Request rate
  - Traffic patterns

→ Capabilities
1. Single choke point
   - ↳ Blocks unauthorized access
   - ↳ Prevents unsafe services
   - ↳ Protect against spoofing & routing attacks
   - ↳ Simplifies security management.

Date _____ Day _____

2. Security Monitoring
   ↳ Logs event
   ↳ Generates alerts & audits
3. Support non-security services
   ↳ Network Address Translation
   ↳ Internet usage logging & management
4. Support VPNs
   ↳ Act as a platform for IPSec

→ **Limitations**

① Cannot protect traffic that bypass it
   ↳ Direct ISP Access
   ↳ Peer-to-Peer network links
② Limited protection against insider threats
   ↳ Malicious or careless employees
③ Wireless risks
   ↳ Poorly secured Wi-Fi can bypass internal firewalls
④ Mobile & removeable devices
   ↳ Devices infected outside of pvt network can bring malware inside

Date _____ Day _____

• Types Of Firewalls

① Packet Filtering Firewalls

↳ A packet filtering firewall checks each incoming and outgoing IP packet

↳ Based on predefined rules, the packet is either "forwarded" or "discarded"

↳ Filtering is usually applied in both directions
  ↳ pkts going in & out of the internal net.

↳ Packet Filtering Rule Criteria:

↳ Rules are based on info in IP/TCP header

- Source IP Addr.
- Dest IP Addr
- Src & Dest Port #
- IP Protocol Field (TCP, UDP)
- Interface.
  ↳ which firewall interface the pkt came from or is going to

↳ Rule Processing

↳ Rules are checked top → bottom

↳ If a pkt matches a rule → action is applied
  ↳ (permit / deny)

↳ If no rule matches → default policy is applied

Day _____

Date _____

↳ Default Firewall Policies

↳ Default Discard

- "Request which is not expressly permitted is prohibited"
- More secure & conservative
- Initially blocks all traffic
- Services are allowed one by one
- Preferred by businesses / gov organizations

↳ Default Permit

- "Request which is not explicitly prohibited is permitted"
- Easier for users
- Less secure
- Administrator reacts to threats as they arise
- Often used by open orgs (eg: universities)

↳ SMTP Packet Filtering Ruleset

1. Inbound mail from an external src is allowed
2. ~~This rule is intended~~ Allow responses to inbound SMTP connections
3. Allow outbound SMTP traffic
4. Allow responses to outbound SMTP connection
5. Explicit deny rule (default policy)

Date _____ Day _____

Rule	Direction	Src Address	Dest Address	Protocol	Dest Port	Action
1	In	External	Internal	TCP	25	Permit
2	Out	Internal	External	TCP	>1023	Permit
3	Out	Internal	External	TCP	25	Permit
4	In	External	Internal	TCP	>1023	Permit
5	Either	Any	Any	Any	Any	Deny

! Rule #04 allows external traffic to any dest port of internal network above 1023
An attacker can exploit this to access internal services (eg: Web proxy on port 8080)

- First Fix:
  - ↳ Add source port field
    - Rule 2 & 4 → src port = 25
    - Rule 1 & 3 → src port = 71023

andar se ya bahir se koi req aaye to wo smtp ki taraf se hi ho, kisi unknown port se hoga to reject

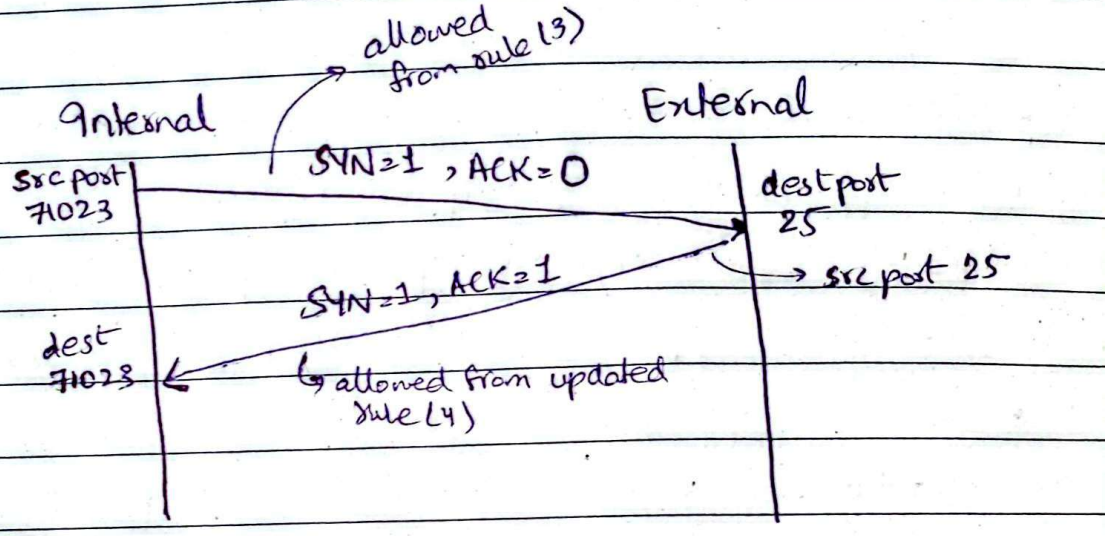
Gor agr andar se ya bahir se SMTP ki taraf traffic jaarha hai to wo bhi ek predefined port se jaye.

- Remaining Vulnerability:
  - ↳ Bahir se jo request port 25 se aashi hai, to zaroori nhi k 25 par SMTP ~~port~~ hi run horaha ho.
  - ↳ An attacker can send pkt using src port 25 to bypass rule.

Date _____ Day _____

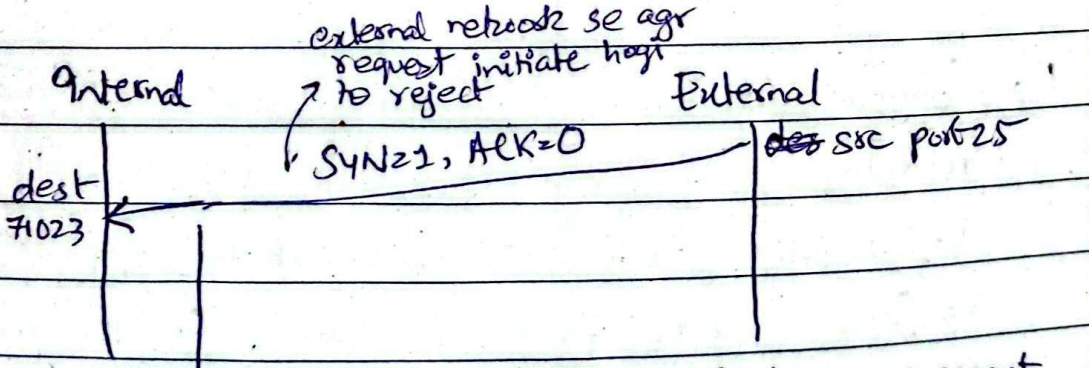
- Final Fix :

↳ Baahir se sirf wohi packet accept honge jo k hamari apni request k response k liye aashe ho.



↳ To ab iske liye hum ACK flag check krenge k agr baahir se jo request aarhi hai or uska ACK=1 ho to hi PERMIT else DISCARD

Rule	Direc	Src Addr	Src Port	Dest Addr	Protocol	DestPort	Flag	Action
4	IN	External	25	Internal	TCP	>1023	ACK	Permit



Agr hum prev rule (4) ko lein to wo ye accept krleta bcz ACK wala flag hum check nhi krrhe the.
Ab update rule (4) checks ACK so it will not be allowed

Date _____ Day _____

↳ Advantages
- Simple to design and configure
- Fast performance (low processing overhead)
- Transparent to users

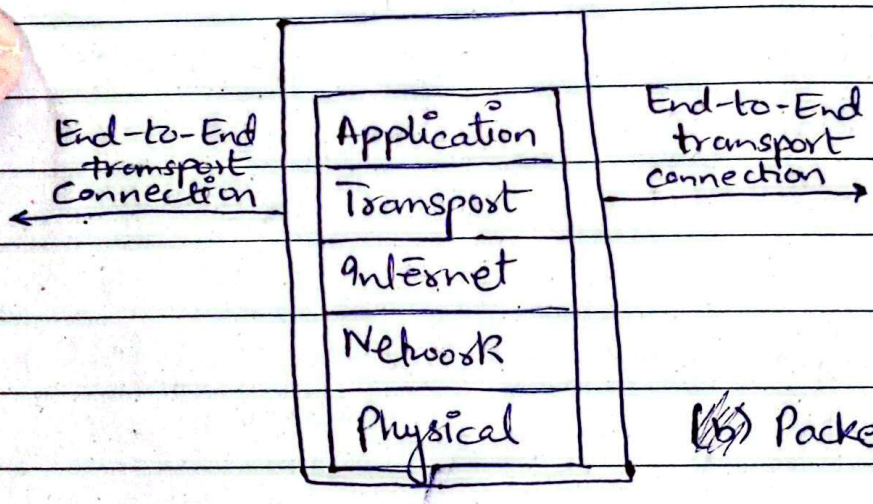
↳ Weakness
- No application-level inspection
  ↳ Cannot see or block specific application cmds
    ↳ If an app is allowed, all funcs are allowed
- Limited Logging
  ↳ Logs only basic info (src/des IP, port --)
- Weak user authentication
  ↳ Cannot support advance auth due to lack of upper-layer data
- Vulnerable to TCP/IP Exploits
  ↳ Cannot reliably detect spooped or altered IP
- Configuration errors are common

↳ Common Attacks & Countermeasure
- IP Address Spoofing
  ↳ Attacker fakes an internal IP to appear trusted
  ↳ Counter: Drop pkts with internal src IPs arriving from external interfaces
- Source Routing Attack
  ↳ Attacker specifies pkt route to bypass security checks

Date _____ Day _____

↳ Counter: Discard all pkts that use src routing

- Tiny Fragment Attack
  ↳ Packet is split so TCP header is hidden in later fragments
  ↳ Counter: Require first fragment to contain enough TCP header data; if rejected, drop all related fragments



(1) Packet Filtering Firewall

(2) Stateful Inspection Firewall

- Limitations of Packet Filtering firewalls
  ↳ Makes decision per packet only
  ↳ Donot understand connection context

Example:

↳ TCP applications uses a client-server model
↳ Server uses a well known port (<1024) eg: SMTP=25
↳ Client uses a temporary high numbered port (1024-65535)

packet filter firewall

Date _____ Day _____

↳ "Now since client can send request from any port (1024-65535) so a server must be allowed inbound traffic to high ports which is a security risk"

- Functionality of Stateful Inspection Firewall

↳ "When a request from client (internal network) is sent from a port (1028), to jab external network se response ayega (1028) par to we bhi accept krna hoga"

↳ ye same ACK wala problem hai.

Internal PC (port ~~5000~~ 1028) ⟶ External Mail (25) Server

↳ Now Reply comes back

External Mail server Port (25) ⟶ Internal PC (1028)

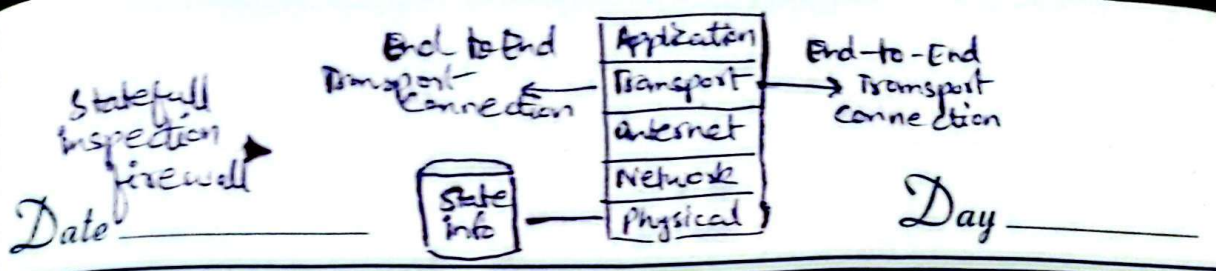
↳ This is legit so accepted by prt filter firewall

→ But, But, But, an attacker can do this

Attacker (port 25) ⟶ Internal PC (1028)

↳ this will bypass since port is same

↳ Attacker can also fake "ACK"



Date _____ Day _____

→ The "ACK" wali logic itself is no self sufficent or strong.

→ Countermeasure:

↳ Mantains a state table (connection directory) of all active TCP connections.

↳ There is an entry for each currently established connection

↳ Incomming packets to high-numbered ports are allowed only if they match an existing entry

Src Addr	Src Port	Dest Addr	Dest Port	Connection State
192.168.1.100	1030	210.9.88.29	80	ESTABLISHED
192.168.1.102	1031	216.32.42.123	80	ESTABLISHED
192.168.1.101	1033	173.66.32.122	25	ESTABLISHED

↳ Jese hi internal network se request jaati hai to entry create hojaati hai.

③ Application-Level Gateway

↳ Also called Application Proxy

↳ Act as a middleman (relay) for application level traffic

↳ User connects to the gateway, not directly to the remote server

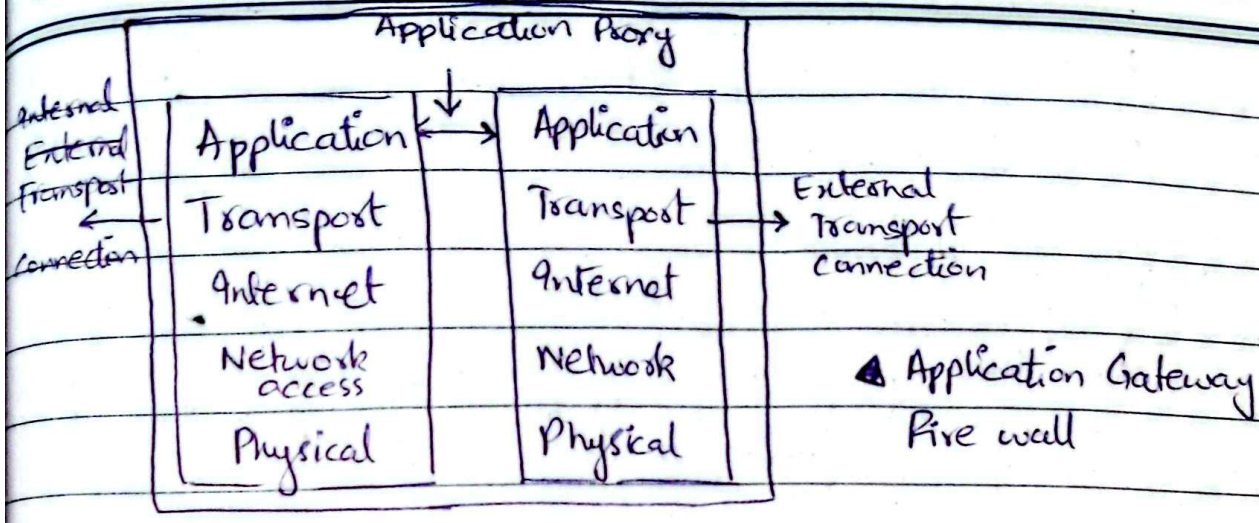
Date _____ Day _____

↳ Gateway then,
  ↳ Authenticates the user
  ↳ Connects to the remote application on behalf of the user
  ↳ Forwards application data b/w both sides
↳ Only supported applications (eg: FTP, Telnet, HTTP) can pass through
↳ Can allow or block specific feature of an application

↳ ADVANTAGES
  ↳ More secure than pkt filtering firewall
  ↳ Inspect traffic at the application level
  ↳ Easy to log & audit user actions & traffic

↳ DISADVANTAGE
  ↳ Higher processing overhead
  ↳ Two connections / session (client → gateway, gateway → server)
  ↳ Slow compared to packet filtering & stateful inspection

Date _____ Day _____



④ Circuit-Level Gateway

↳ Operates at the TCP session (circuit ~~ter~~) level, not application data level

↳ Doesnot allow end-to-end TCP connection

↳ Creates two separate TCP connection

- Internal Client ⟷ Gateway
- Gateway ⟷ External Server

↳ After connection setup, TCP segments are relayed without inspecting payload

↳ Security decision is based on whether to allow or deny a connection, not its content

↳ Key Characteristics

- Examines connection setup (handshake), not application commands
- Hides internal network structure from external hosts
- Faster than application-level gateways (no payload inspection)

- Simple to implement
- within firewall, it doesnot offer protection against data leakage from devices.
- Frequent updates are required

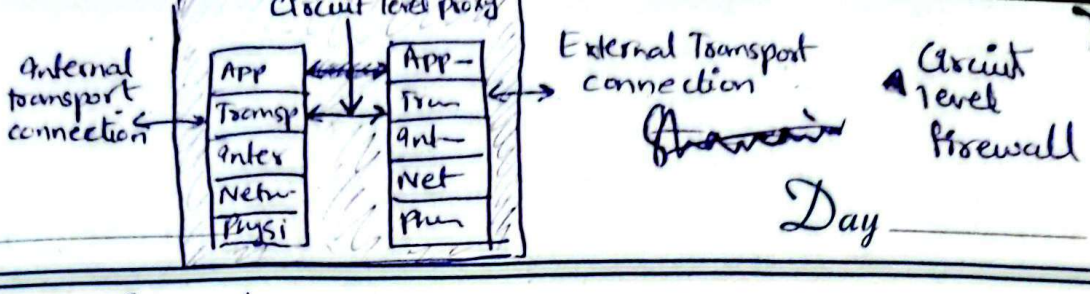
Date Day

- Less secure than application proxies
- Doesnot filter individual packets

↳ Typical Use Case

- If internal users are trusted
  - ↳ then for inbound traffic → application-gateway
  - ↳ " " outbound traffic → circuit level "

- SOCKS (Example Implementation)
  - ↳ Standard circuit-level gateway protocol
  - ↳ Act as a "shim layer" blw app. & transport layer
  - ↳ SOCKS Components
    - ↳ SOCKS Server — Runs on firewall (port 1080)
    - ↳ SOCKS Client Library — Installed on internal hosts
    - ↳ SOCKS-enable Applications = FTP, Telnet etc (modified or linked with SOCKS)
  - ↳ SOCKS Connection Flow
    1. Client opens TCP connection to SOCKS server
    2. Client & server negotiates authentication method
    3. client authenticates
    4. Client sends relay request
    5. SOCKS Server Either
       - Allows → establishes external connection
       - Denies → blocks request
    6. Gateway relays traffic transparently



Date _____ Day _____

↳ UDP Support

↳ Uses a TCP connection for auth

↳ UDP packets are relayed only while TCP session remains active

- **INTRUSION PREVENTION SYSTEM (IPS)**

→ Also called IDPS, extends IDS by actively blocking or preventing attacks, not just detecting

→ Can be:

↳ Host-based IPS (HIPS)

↳ Network-based IPS (NIPS)

↳ Distributed / Hybrid IPS

→ Detection Technique

- Anomaly-based detection → identifies abnormal behavior
- Signature / heuristic-based detection → identifies known attacks

→ Detects malacious activity AND takes action immediately

→ Networks IPS behaves similar to Firewall but uses IDS-style detection algorithms

- **Host Based IPS - (HIPS)**

↳ HIPS run on individual hosts

↳ Uses Signature or anomaly based detetion

Date _____ Day _____

↳ Types of Attacks Detected

① Modification of system resources
  ↳ (Rootkits, Trojans, backdoors modifying libraries, directories, registry, users)

② Privilege Escalation attacks
  ↳ Normal user gaining root/admin access

③ Buffer Overflow Attacks — overwriting memory of an appli.

④ Access to email contact list

⑤ Directory traversal attacks
  ↳ Accessing files outside allowed directories

↳ Platform Specific Protection
  ↳ Can be general purpose (desktop, servers)
  ↳ Can be application specific (web servers, db servers)

↳ Sandboxing
  ↳ Suspicious code is run in an isolated env
  ↳ If behaviour violates policies or matches malicious patterns → execution is stopped

↳ Areas Monitored by HIPS

① System Calls — checks kernel level operations
② File Sys Access — enforces file access policies
③ System Registry — prevents malicious registry changes
④ Host I/p O/p — Monitors local & network I/O activity

Date _____ Day _____

↳ Role Of HIPS

↳ As the endpoints (desktop, laptops..) are primary attack targets

↳ HIPS combine multiple tools (antivirus, antispyware, firewall) int one integrated sdn.

↳ Benefits

↳ Better coordination of security funcs

↳ More comprehensive protection

- **Network Based IPS — (NIPS)**

↳ NIPS works inline on the network & can block, modify or drop pkts

↳ Similar to NIDS, but with prevention capability

↳ Uses signature / Heuristic and anomaly detection

→ Not commonly found in firewall

↳ Flow Data Protection

↳ Reassembles application payload accross multiple packets

↳ Inspects the entire traffic flow, not individual pkts

↳ If a flow is malicious, current & future pkts of that flow are dropped

→

Date _____ Day _____

↳ Malicious Packet Detection Methods

- pattern matching — matches packet data with known attack signatures
- Stateful Matching — Analyzes attacks in the context of a traffic stream
- Protocol Anomaly Detection — detects deviations from RFC standards
- Traffic Anomaly Detection — Identifies abnormal traffic behaviour (eg: floods)
- Statistical Anomaly " — compares traffic to normal baseline pattern

- **Distributed / Hybrid IPS**
  - ↳ Combines host-based & network-based IPS
  - ↳ Sensors collect security data & send it to a central analysis sys
  - ↳ Central syst correlates and analyzes events, then distributes updated signatures & behaviour rules to all sensors
  - ↳ Enables coordinated, sys-wide detection & response
  - ↳ Core Functions
    - ↳ Automatically captures new malware
    - ↳ Analyzes it in a protected env
    - ↳ Generates detection signatures & shielding mechanisms

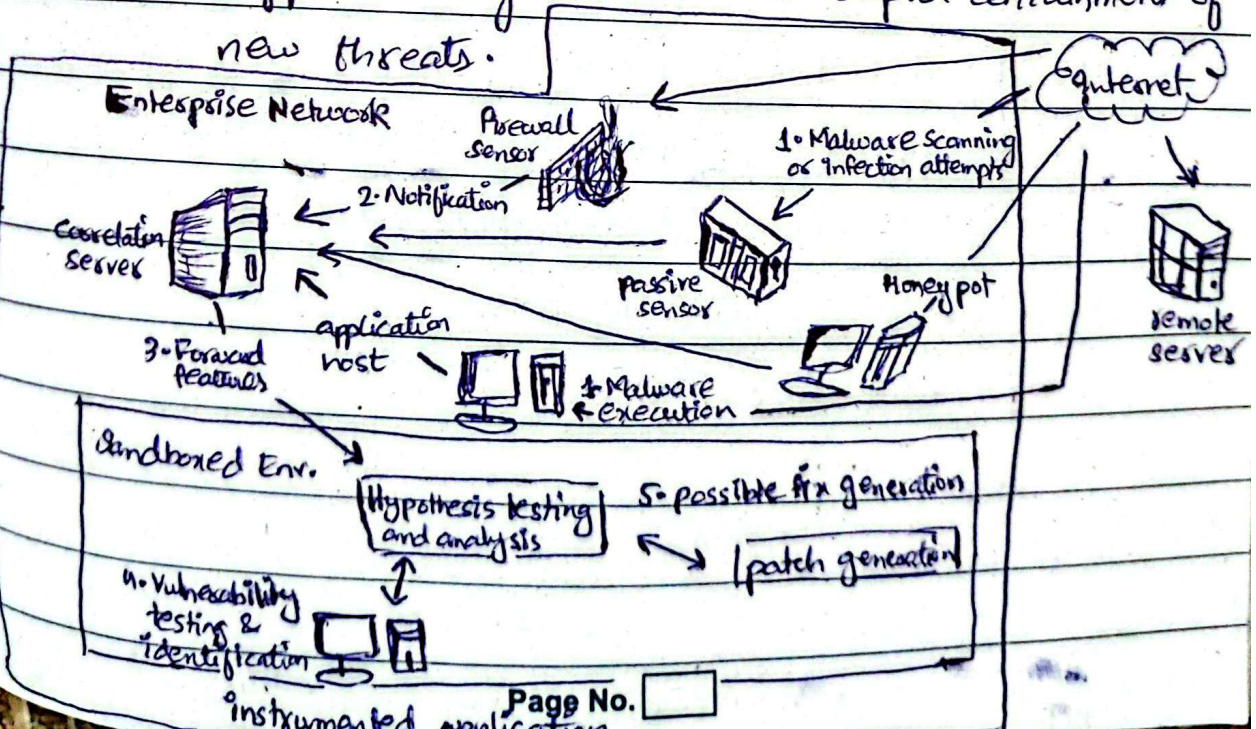
Date _____ Day _____

↳ Removes malware from infected systems
↳ Distributes updates to clients to prevent future execution.

↳ Hybrid IPS Architecture

1. Sensors detect scanning, infection or execution steps
2. Sensors send alert & malware samples to a central correlation server
3. Malware is analyzed in a sandboxed env
4. Vulnerabilities are identified using instrumental applications
5. Patches are generated & tested
6. Verified patches are deployed to affected systems

→ Effectiveness depends on continuous malware analysis & frequent updates
→ Supports early detection & rapid containment of new threats.



Date _____ Day _____

- **Snort Inline**

  ↳ Snort inline is a modified version of Snort that works as an IPS (an IDS)

  ↳ It can block / modify malicious traffic, not just detect it.

  ↳ New Rule Types in Snort Inline
  - Drop — Blocks the packet & logs the event
  - Reject — Blocks " " " " " , and sends an err msg.
    - ↳ TCP → sends TCP reset
    - ↳ UDP → sends ICMP port unreachable
  - SDROP — Blocks pkt without logging

  ↳ Replace Option
  - ↳ Allows modifying pkt contents instead of dropping them
  - ↳ Useful in honeypot setups
  - ↳ Attacks are disabled silently, so attackers see failure but don't know why
  - ↳ Enables continued monitoring of attackers while preventing real damage.