Network Operational Security

Thanks to J.F Kurose and K.W. Ross, “Computer Networking: A Top-Down Approach, 8th edition, Pearson, 2020

University of Minho, 2025, pmc

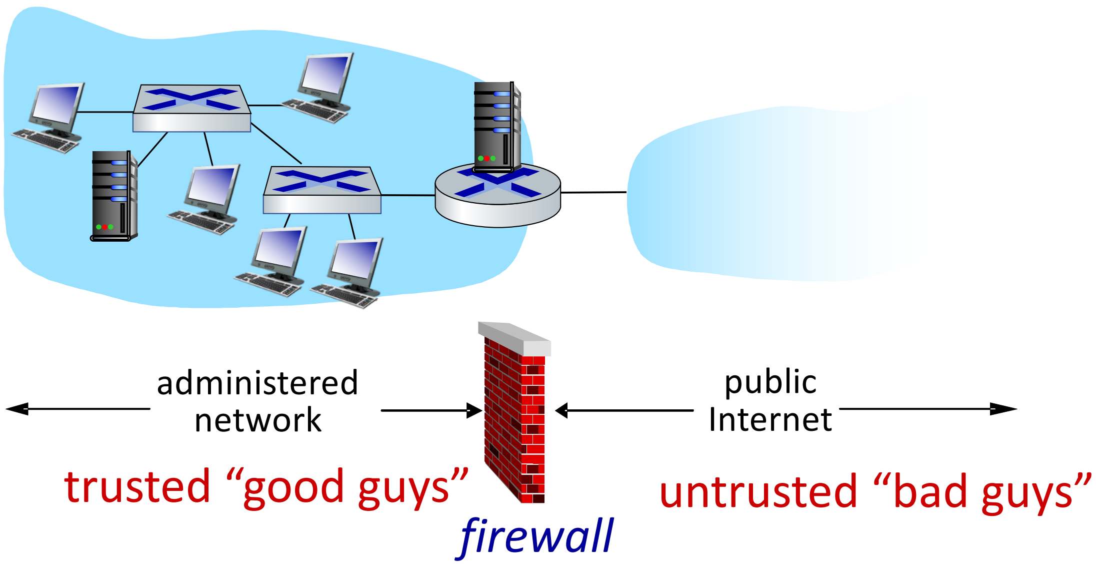
Solutions

- **Firewalls**
- **Intrusion Detection System**
- **Network Access Control (NAC):** technologies and strategies for controlling access to network resources, including 802.1X and role-based access control (RBAC)
- **SIEM (Security Information and Event Management):** how SIEM systems aggregate and analyse security data from various sources, including firewalls and IDSs, for centralized incident detection and response
- **Log Management and Active Monitoring:** logging network activity, ensuring compliance, and identifying potential threats
- **Incident Response and Forensics:** how organizations respond to security incidents, including containment strategies and post-incident analysis
- **Zero Trust Architecture:** from perimeter-based security models to zero trust principles, emphasizing continuous authentication and least privilege principle
- **DDoS Mitigation Techniques:** defending against DDoS attacks, as common operational threats
- **Endpoint Detection and Response (EDR):** tools and strategies for monitoring and securing endpoints as part of the operational security

Firewalls

firewall

isolates organization's internal network from Internet, allowing some packets to pass, blocking others

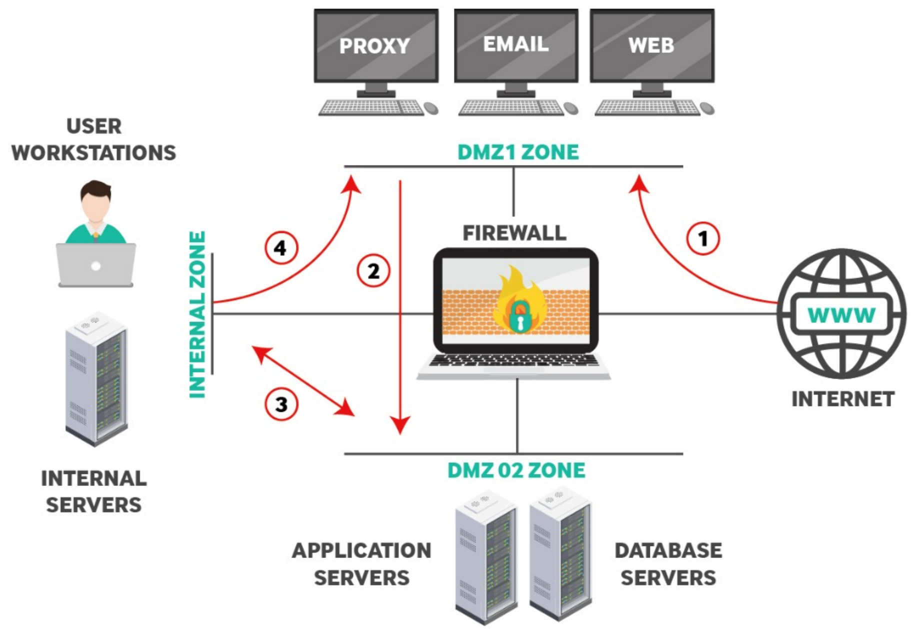


Firewalls: why

- Traffic Control
  - firewall acts as a gatekeeper to control and monitor inbound and outbound traffic based on predefined rules
- Threat Mitigation
  - helps blocking malicious traffic (malware, ransomware, etc.) and unauthorized access attempts
    - set of authenticated users/hosts
  - preventing denial of service attacks
    - SYN flooding: attacker establishes many bogus TCP connections, leaving no resources left for legitimate connections
  - preventing illegal access/modification of internal data
    - e.g., attacker replaces some homepage with fake data

Firewalls: why

- Segmentation and Isolation
  - firewalls can segment networks to contain threats and protect sensitive areas
- Compliance
  - several regulations (e.g., GDPR, HIPAA) mandate the use of firewalls to secure data
- Visibility
  - modern firewalls offer insights into traffic patterns, helping organizations to identify and respond to potential threats

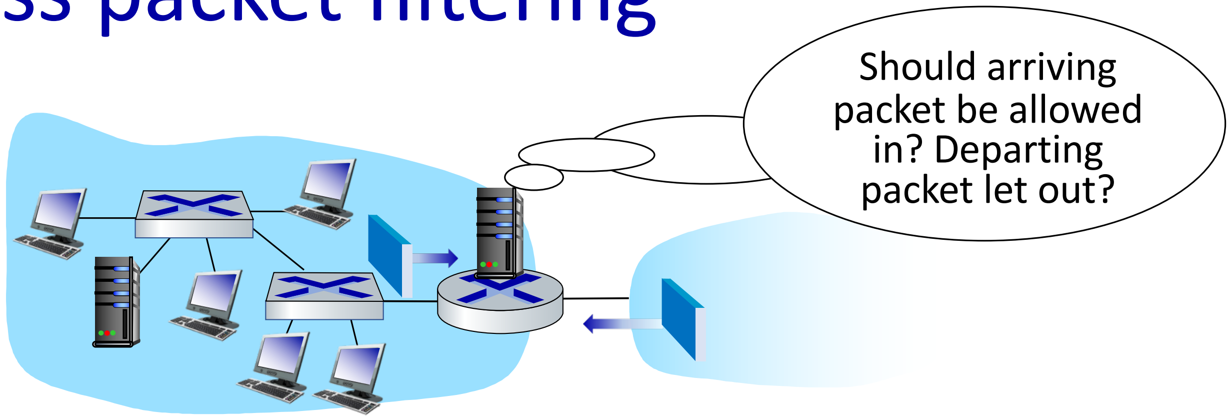


[Source: https://www.titanhq.com/blog/best-firewall-security-zone-segmentation-for-optimal-network-security/]

Firewalls: types

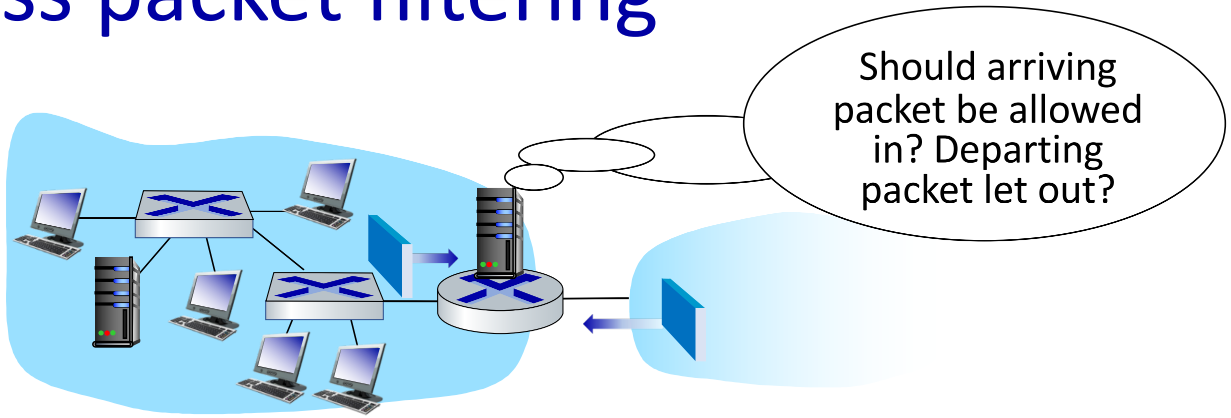
- Stateless packet filtering
  - layer 3 and 4 operation over individual packets; limited context-aware
- Stateful packet filtering
  - keep state for active connections within a session
- Application gateways (proxy firewall)
  - layer 7 operation, acting as intermediates between clients and servers
- Next Generation Firewalls
  - combine traditional firewall capabilities with advanced features such as intrusion prevention, deep packet inspection, and threat intelligence
- Virtual Firewalls
  - software-based designed to secure virtualized environments and containers
- Cloud Firewalls

Stateless packet filtering



- internal network connected to Internet via router firewall
- filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source, destination port numbers
  - ICMP message type
  - TCP SYN, ACK bits

Stateless packet filtering



- example 1: block incoming and outgoing datagrams with IP protocol field = 17 (UDP) and with either source or dest port = 23 (telnet)
  - result: all incoming, outgoing UDP flows and telnet connections are blocked
- example 2: block inbound TCP segments with ACK=0
  - result: prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside

Stateless packet filtering: more examples

Policy	Firewall Setting
no outside Web access	drop all outgoing packets to any IP address, port 80
no incoming TCP connections, except those for institution's public Web server only	drop all incoming TCP SYN packets to any IP except 130.207.244.203/16, port 80
prevent Web-radios from eating up the available bandwidth	drop all incoming UDP packets - except DNS and router broadcasts
prevent your network from being used for a smurf DoS attack	drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255/16)
prevent your network from being tracerouted	drop all outgoing ICMP TTL expired traffic

Access Control Lists

ACL: table of rules, applied top to bottom to incoming packets:
(action, condition) pairs

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Stateful packet filtering

- *stateless packet filter:* heavy handed tool
  - admits packets that “make no sense,” e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets “make sense”
  - timeout inactive connections at firewall: no longer admit packets

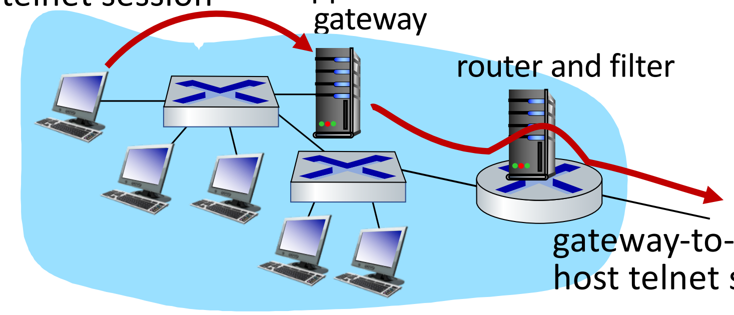
Stateful packet filtering

ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Application gateways

- filter packets on application data as well as on IP/TCP/UDP fields.
- *example:* allow select internal users to telnet outside



1. require all telnet users to telnet through gateway
2. for authorized users, gateway sets up telnet connection to dest host
    - gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway

Limitations of firewalls, gateways

- IP spoofing: router can't know if data "really" comes from claimed source
- if multiple apps need special treatment, each has own app gateway
- client software must know how to contact gateway
  - e.g., must set IP address of proxy in Web browser
- filters often use all or nothing policy for UDP
- *tradeoff:* degree of communication with outside world vs. level of security
- many highly protected sites still suffer from attacks

Firewalls: additional considerations

- Firewall Deployment
  - Network Perimeter: secures the boundary between internal and external networks
  - Internal Segmentation: isolates different segments within a network (see Fig. in slide 5)
  - Host-Based: protects individual devices
- Rule Design and Management
  - crafting effective policies to minimize security risks and prevent misconfigurations
  - regular auditing of firewall rules for relevance and effectiveness
- Firewall Limitations
  - cannot inspect encrypted traffic without TLS inspection
  - ineffective against insider threats or social engineering (e.g. targeting admin)
- Firewall Performance
  - impact on latency and throughput; scalability for high-traffic environments
- Emerging Trends
  - integration with AI/ML for predictive threat detection
  - important role in Zero Trust Network Access

Comparison of firewall types

Characteristic	Stateless Packet Filtering	Stateful Packet Filtering	Application Layer Filtering
Principle	Inspects individual packets based on static rules (IP, ports)	Tracks and evaluates traffic based on connection state	Inspects traffic at the application layer (Layer 7)
Requirements	Simple access control rules (ACLs)	Maintains a connection table to track active sessions	Deep understanding of application protocols (e.g., HTTP)
Complexity	Low – simple rule matching	Moderate – connection state tracking increases complexity	High – requires protocol-level inspection and parsing
Security Level	Basic – vulnerable to spoofing and fragmented attacks	Medium – protects against spoofing, supports session awareness	High – can block application-specific threats and malware
Resource Requirements	Low – minimal CPU and memory usage	Moderate – state tracking requires additional resources	High – resource-intensive due to deep packet inspection
Performance	High – processes packets quickly due to simplicity	Moderate – state tracking adds overhead	Low – deep inspection impacts throughput and latency
Configuration	Straightforward – simple rules for allowed/denied traffic	More complex – requires defining stateful policies	Complex – involves detailed application-level policies
Encrypted Traffic Handling	Cannot inspect encrypted traffic	Limited – needs SSL inspection for encrypted traffic	Supports SSL inspection but adds significant overhead
Examples of Threats Addressed	IP spoofing, port scanning (limited)	SYN floods, session hijacking	SQL injection, cross-site scripting, protocol-based attacks
Limitations	No session awareness; cannot block sophisticated threats	Ineffective against application-layer threats	High resource consumption; prone to false positives
Examples of Use	Legacy systems, low-security environments	Modern perimeter firewalls for small to medium businesses	Protecting critical systems (e.g., web servers).

Firewalls: security

- Do Firewalls themselves face any security concerns?

- Yes. Even with proper policies, configurations, and updates, firewalls can still face various types of threats and attacks

Firewalls: security threats

- 1. DoS and DDoS Attacks
  - target: firewall's processing, bandwidth and memory resources
  - impact: overwhelm the firewall, causing it to slow down or fail, potentially allowing attackers to bypass security

    Examples:
    - flooding firewall with excessive SYN packets (SYN flood)
    - sending fragmented packets that require resource-intensive reassembly
- 2. Evasion Techniques
  - target: inspection mechanisms of firewall
  - impact: attackers craft packets or payloads to bypass detection and filtering

    Examples:
    - fragmented packet attacks to slip malicious payloads through
    - encoding or obfuscating payloads to bypass application-layer filtering

Firewalls: security threats

- 3. Zero-Day Exploits
  - target: unpatched vulnerabilities in firewall software or firmware
  - impact: exploitation of unknown flaws to gain unauthorized access, disable the firewall, or control it

    Examples:
    - exploiting an unreported bug in the firewall's VPN or DPI module
    - triggering a buffer overflow in the firewall's management interface
- 4. Insider Misuse of Access
  - target: firewall’s administrative interface or rule set
  - impact: an insider with legitimate access intentionally or inadvertently weakens firewall configurations

    Examples:
    - adding permissive rules to allow unauthorized access
    - creating backdoors for external attackers

Firewalls: security threats

- 5. Man-in-the-Middle (MitM) Attacks
  - target: firewall management sessions or encrypted traffic passing through the firewall
  - impact: interception or tampering with sensitive communications or config

    Examples:
    - exploiting weak encryption protocols for administrative sessions
    - attacking VPN tunnels that terminate at the firewall
- 6. Supply Chain Attacks
  - target: hardware or software supply chains for the firewall
  - impact: introducing malicious firmware or backdoors before the firewall is deployed

    Examples:
    - malware preinstalled on firewall appliance
    - tampered software updates delivered via compromised channels

Firewalls: security threats

- 7. DNS-Based Attacks
  - target: firewalls that provide DNS services or handle DNS traffic
  - impact: exploitation of DNS resolution to redirect or bypass firewall

    Examples:
    - DNS tunneling to exfiltrate data through the firewall
    - cache poisoning attacks to mislead the firewall's DNS filtering
- 8. Malware Implantation
  - target: firewall’s underlying operating system or management interface
  - impact: implant malware to establish persistence and secret control

    Examples:
    - installing a rootkit to intercept traffic or modify rules
    - compromising the web-based management console

Firewalls: security threats

- 9. API Exploitation
  - target: APIs exposed by modern firewalls for management or integration
  - impact: exploiting misconfig or vulnerable APIs to gain unauthorized control

    Examples:
    - using weakly secured API endpoints to bypass authentication
    - injecting malicious payloads through API calls
- 10. Traffic Saturation (Resource Exhaustion, subset of DoS)
  - target: firewall session table or resources limits
  - impact: legitimate traffic is disrupted as firewall becomes overwhelmed

    Examples:
    - sending high-volume legitimate traffic (e.g., large file downloads) to exhaust resources
    - exploiting session table limits by opening many connections

Firewalls: security threats

- 11. Side-Channel Attacks
  - target: physical characteristics or observable data from firewall
  - impact: extract sensitive information, such as cryptographic keys

    Examples:
    - timing attacks to deduce firewall processing patterns
    - electromagnetic emissions analysis
- 12. Configuration Drift Over Time
  - target: gradual misalignment of firewall settings due to manual or automated changes
  - impact: reduced effectiveness of rules, leading to unintended vulnerabilities

    Example:
    - overlapping rules or outdated deny lists weakening security

Firewalls: security threats

- Mitigation
  - Robust Resource Management
    - deploy high-capacity firewalls and monitor performance to handle DoS/DDoS attacks
  - Regular Vulnerability Scanning
    - continuously scan for vulnerabilities and apply patches promptly
  - Rule Auditing
    - regularly review and update firewall rules to address configuration drift
  - Strong Encryption
    - use strong encryption for management sessions and VPNs
  - Secure Supply Chain
    - verify hardware/software suppliers and verify the integrity of updates
  - Behavioral Analytics
    - employ tools to detect abnormal traffic patterns or access attempts

Intrusion Detection Systems

- packet filtering:
  - operates on TCP/IP headers only
  - no correlation check among sessions
- IDS: Intrusion Detection System
  - deep packet inspection: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - examine correlation among multiple packets
    - port scanning
    - network mapping
    - DoS attack

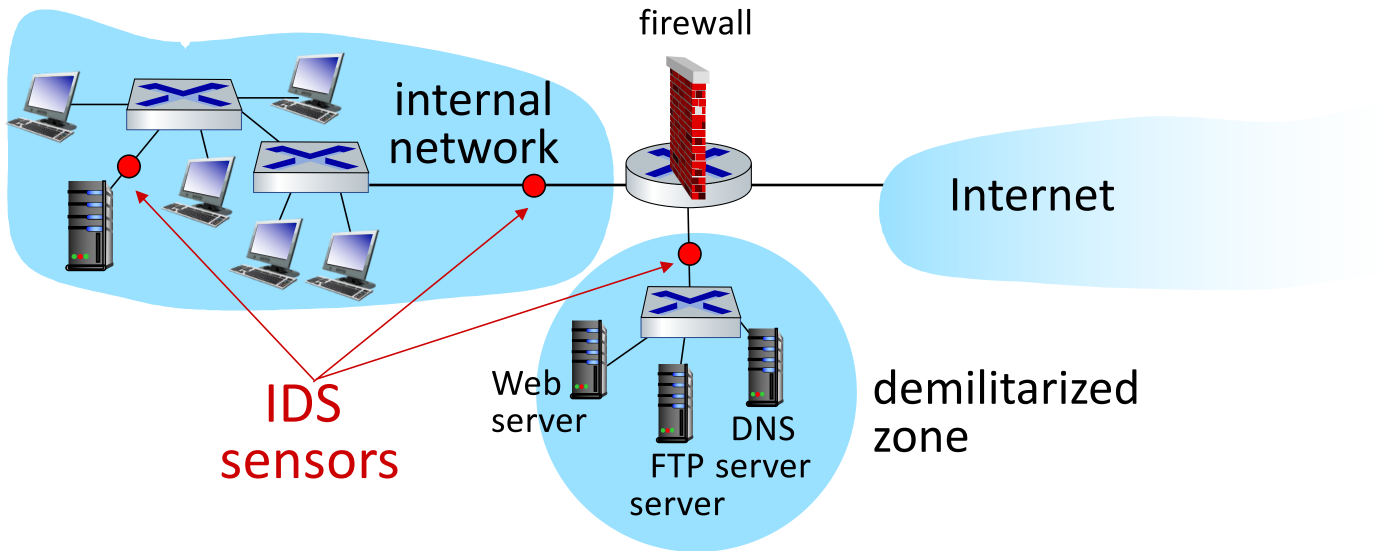
  - Types: Network-based (NIDS), Host-based (HIDS), Hybrid

Network Intrusion Detection Systems (NIDS)

- Why focusing on NIDS?
  - Network-wide Visibility
    - unlike HIDS, which operates on individual endpoints, NIDS monitors traffic across the entire network, detecting threats that firewalls may miss
  - Complementary to Firewalls
    - firewalls filter traffic based on rules, while NIDS analyses traffic for anomalies, policy violations, or signatures of known attacks
  - Attack Detection Beyond Policy Violations
    - firewalls enforce policies but cannot detect sophisticated threats like zero-day exploits, insider threats, or encrypted malicious payloads—NIDS can help fill this gap
  - Scalability
    - NIDS can be deployed at critical network points (e.g., perimeter, DMZ, internal network) to detect attacks before they reach endpoints
  - Forensics & Incident Response
    - NIDS logs attack attempts, suspicious patterns, and traffic anomalies, helping security teams investigate threats

Network Intrusion Detection Systems

multiple IDSs: different types of checking at different locations

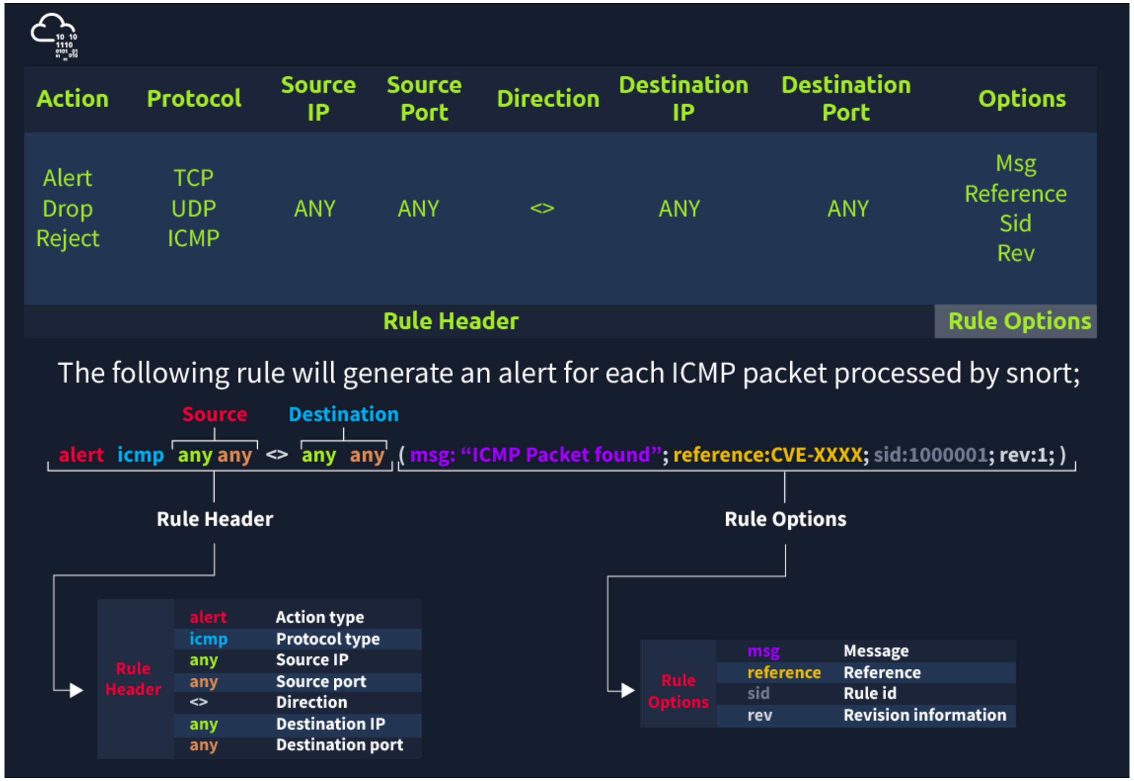


NIDS – detection approaches (1)

- Signature-based
  - compares network traffic against a database of known attack patterns (signatures)
  - if a packet (or sequence) matches a signature, an alert is triggered
  - works similarly to antivirus software but at the network level
  - requires continuous update of signature DB (e.g., Snort, Suricata rules)
  - advantages:
    - highly accurate for well-known attacks; efficient as only checks known patterns
  - disadvantages:
    - may not be effective for new, emerging attacks (zero-day exploits)
    - latency of signature updates behind emerging threads
    - attacks may use evasion techniques (e.g., obfuscation, fragmentation)

NIDS – detection approaches (1)

- Example use cases (signature-based)
  - detecting SQL injection, buffer overflows, and known malware command-and-control (C2) traffic*
  - e.g., snort rule



* communication with an external attacker-controlled C2 server to receive commands, e.g., exfiltrate data, execute code, etc.

NIDS – detection approaches (2)

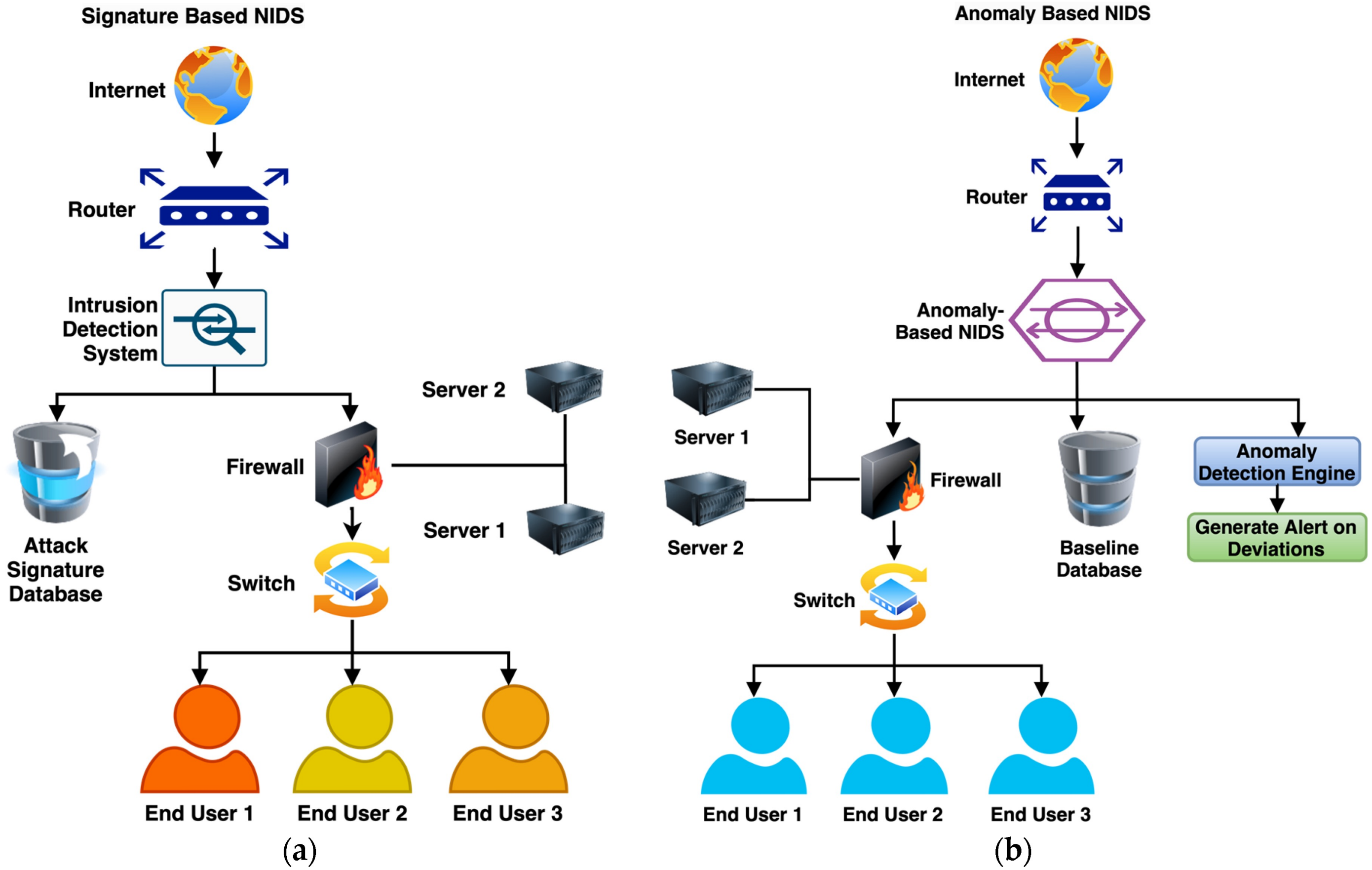
- Anomaly-based
  - establishes a baseline of "normal" network behavior using statistical models, machine learning, or heuristics
  - detects deviations from this baseline that may indicate malicious activity
  - requires initial training period to build a baseline, and continuous monitoring to detect deviations
  - advantages:
    - can detect zero-day attacks and novel threats
    - identifies behavioral anomalies, not just known attack patterns
    - useful for detecting insider threats and slow, stealthy attacks (e.g., slow port scans)
  - disadvantages:
    - high false positives due to deviations caused by legitimate but unexpected activity
    - training complexity; requires fine-tuning to reduce noise

NIDS – detection approaches (2)

- Example use cases (anomaly-based)
  - detecting data exfiltration by monitoring unusual outbound traffic patterns
  - identifying DDoS attacks based on sudden traffic spikes

- Alert scenario:
  - a web server normally handles 500 connections per minute
  - a sudden spike to 5000 connections per minute triggers an anomaly alert

NIDS – detection approaches (1) and (2)



Source: *Mathematics* 2024, *12*(24), 3909; https://doi.org/10.3390/math12243909

NIDS – detection approaches (3)

- Hybrid detection
  - combines signature and anomaly-based approaches for improved detection
  - signatures for known threats and anomaly detection for unknown threats
  - some systems also incorporate AI/ML models for behavior-based analysis
  - requires signature DB for known threats and ML or heuristic-based models for anomaly detection
  - advantages:
    - more comprehensive coverage of threats
    - lower false positives than pure anomaly-based systems
    - can adapt to evolving attack patterns better than signature-only system
  - disadvantages:
    - more complex to configure and maintain
    - requires higher computational resources

NIDS – detection approaches (3)

- Example use cases for Hybrid NIDS
  - detecting both known malware C2 traffic (signature-based) and unusual user login patterns (anomaly-based)
  - identifying multi-stage attacks, where the attacker first probes the system (anomaly) and then exploits a known vulnerability (signature)
- e.g.,
  - a hybrid NIDS might use Snort for signature-based detection and Zeek (Bro) for behavioral monitoring

\+

NIDS – comparison summary

Detection Approach	Principle	Strengths	Weaknesses	Best For
Signature-Based	- Matches traffic against known attack patterns	- Low false positives - Efficient for well-known threats	- Misses zero-days - Requires frequent signature updates	- Detecting known attacks (e.g., SQL injection, malware)
Anomaly-Based	- Learns normal network behaviour and flags deviations	- Detects unknown threats & zero-days - Identifies insider threats	- High false positives - Requires fine-tuning	- Behavioural monitoring, DDoS detection, insider threats
Hybrid	- Combines both signature-based and anomaly-based techniques	- Comprehensive detection - Adaptive & lower false positives than anomaly-based	- Computationally expensive - Complex to configure	- High-security environments, multi-stage attack detection

Other network operational security topics

- Network Access Control (NAC)
  - technologies and strategies for controlling access to network resources, including 802.1X and role-based access control (RBAC)
- SIEM (Security Information and Event Management)
  - SIEM systems aggregate and analyze security data from various sources, including firewalls and IDSs, for centralized incident detection and response, e.g.,
    - e.g., Wazuh, ELK Stack, Security Onion (open source, cost-effective)
    - e.g., TheHive + Cortex, AlienVault OSSIM (threat intelligence and incident response)
- Log Management and Monitoring
  - importance of logging network activity, ensuring compliance, and identifying potential threats through active monitoring

Other network operational security topics

- Incident Response and Forensics
  - to provide insights into how organizations respond to security incidents, including containment strategies and post-incident analysis
- Zero Trust Architecture
  - shift trend from perimeter-based security models to zero trust principles; need for continuous authentication and least privilege
- DDoS Mitigation Techniques
  - major importance as DDoS are common operational threats
- Endpoint Detection and Response (EDR)
  - need for tools and strategies for monitoring and securing endpoints as part of the operational security posture (*fall within host security*)

Network Operational Security (summary)

- Introduction to Firewalls
- Firewalls security threats
- Network Intrusion Detection Systems
- Additional operational security topics