POLITECNICO DI TORINO

Fundamentals of Information Systems Security

Student:

Gianmarco Michelini

Academic Year 2024/2025

All notes are derived from oral presentations and written papers.

Contents

List of Figures

List of Tables

Chapter 1

Firewall and IDS/IPS

[1]

What is a Firewall?

Literally, a firewall is a 'wall to protect against fire propagation' (a safety feature designed to compartmentalize fire and limit damage). In the context of computer networks, it guarantees a controlled connection between networks at different security levels, serving as boundary protection and a network filter.

1.1 Ingress vs. Egress Firewall

Beware

Bidirectional protection is essential. This concept involves protecting both incoming (ingress) and outgoing (egress) network traffic to ensure comprehensive security.

The Ingress Firewall:

- Is intended for **incoming** connections.
- Typically, **controls access** to the (public) services offered by your network or system.

The Egress Firewall:

- Is intended for **outgoing** connections.
- Typically, used to **monitor and control** the activity of internal personnel or devices (to prevent unauthorized traffic, and also for privacy and data protection).

Classification of Traffic

It's straightforward to classify traffic for **channel-based services** (e.g., TCP applications), but more challenging for **message-based stateless services** (e.g., ICMP, UDP applications), due to their lack of a consistent connection state.

1.2 Three Commandments of Firewall

1. The firewall (FW) must be the only contact point between the internal network and the external network.

2. Only “authorized” traffic should be allowed to pass through the firewall.

3. The firewall must be a highly secure system.

– *D. Cheswick and S. Bellovin*

Referring to each rule:

1. The behavior of employees poses a risk.
2. The technician must understand what they are configuring, especially which rules are necessary.
3. Dedicated security elements should be used to avoid cross-vulnerabilities.

1.3 Authorization Policies

We have two possible choices:

- **Permitlist** (AKA allowlist): "All that is not explicitly permitted, is forbidden."
  - Higher security (gatekeeper).
  - More complex to manage.
- **Blocklist** (AKA denylist): "All that is not explicitly forbidden, is permitted."
  - Lower security (open gates).
  - Easier to manage.

FW: Basic Components

Beware

The Firewall is a system! With several components.

- **Packet filter / screening router / choke**: A component that filters traffic at the network level.
- **Bastion host**: A secure system with auditing.
- **Application gateway (proxy)**: A service that works on behalf of an application, with access control.
- **Dual-homed gateway**: A system with two network cards and routing disabled (ip-forwarding off).

What is a Proxy

A proxy is a system or service that sits between the client and the application server. It intercepts and controls the traffic between the two, often for purposes such as filtering, caching, security, or access control. A proxy is not inherently a firewall, but it can function in a way similar to a firewall depending on its role and how it's configured.

1.4 Control Mechanisms for each Level

To provide a clear and structured explanation of the different controls at various network levels, here's an overview of each control type along with a comparison of how they differ in terms of:

- Controls to be performed (i.e., threats detected).
- Performance.
- Protection of the firewall OS.
- Keeping or breaking the client-server model (where breaking means no direct communication between client and server).

Different controls at various network levels:

- (Static) packet filter.
- Stateful/stateless (dynamic) packet filter.
- Cutoff proxy.
- Circuit-level gateway / proxy.
- Application-level gateway / proxy.
- Stateful inspection.

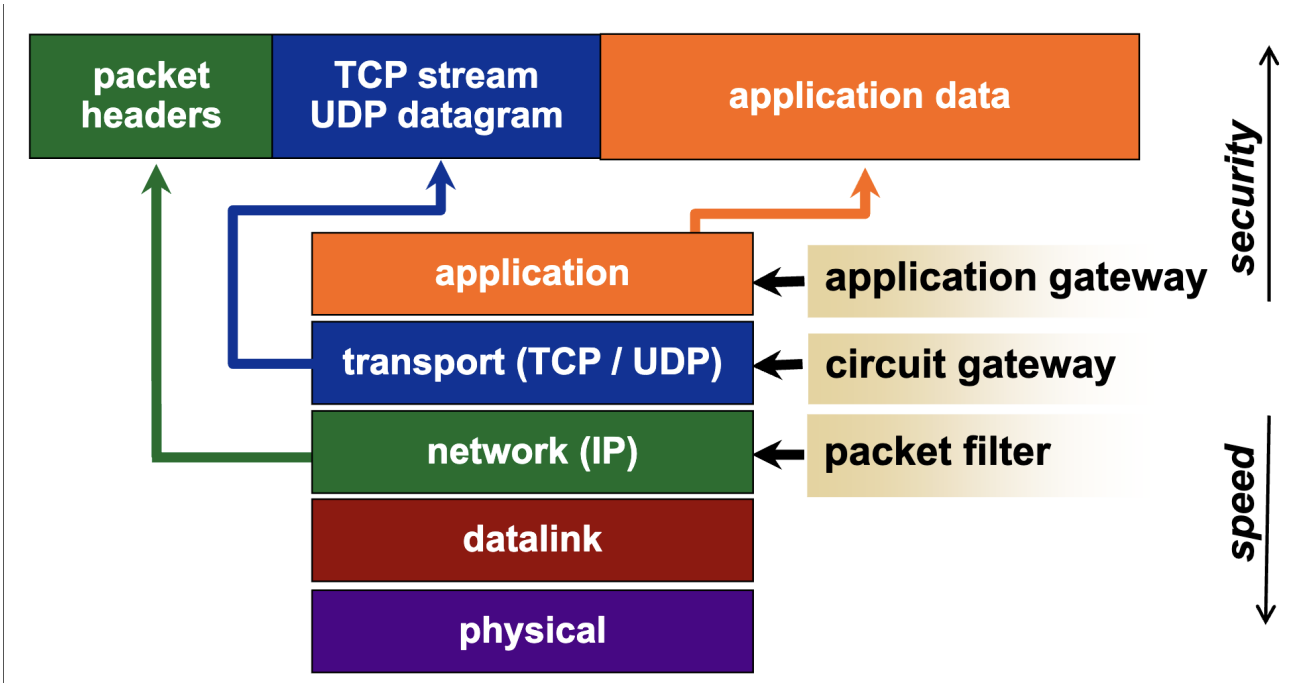

Figure 1.1: Controls for each level.

1.4.1 Packet Filter

Beware

Order is important (first match principle).

- Historically available on routers, nowadays found in almost every OS.
- Performs packet inspection at the network level.
- Inspects the IP header.
- Inspects the transport header.
- Rule examples:
  - Permit incoming connections to our web server:
    ```
    src any dst 10.1.2.3/0.0.0.0 tcp 80 allow
    ```
  - Only our internal DNS server can query external DNS servers:
    ```
    src 10.1.2.1/0.0.0.0 dst any udp 53 allow
    ```
- **Pros:**
  - Independent of applications.
  - Good scalability.
  - Good performance.
  - Low cost (available on routers and in many OS).
- **Cons:**

  - Approximate controls: easy to "fool" (e.g., IP spoofing, fragmented packets).
  - Difficult to support services with dynamically allocated ports (e.g., FTP).
  - Complex to configure (and understand the configuration sometimes).
  - Difficult to perform user authentication.

1.4.2 Circuit-level Gateway

Generic Proxy (i.e., not "Application-Aware")

- Creates a transport-level circuit between the client and server.
- Does not understand or manipulate the payload data in any way.
- Simply copies TCP segments or UDP datagrams between its two interfaces, provided they match the access control rules.
- Re-assembles the IP packets, which helps provide protection against some Layer 3 (L3) and Layer 4 (L4) attacks.
- Breaks the TCP/UDP-level client/server model during the connection.
- Provides more protection for the server.
  - Isolated from attacks related to the TCP handshake.
  - Isolated from attacks related to IP fragmentation.
- May authenticate the client, but this requires modification to the application.
- Exhibits many limitations of a packet filter.
- **SOCKS** is one of the most well-known examples of a generic proxy.

1.4.3 Application-level Gateway

Composed of a set of proxies (collection of elements) inspecting the packet payload at the application level:

- Often requires modifications to the client application.
- May optionally mask or renumber the internal IP addresses.
- When used as part of a firewall, usually performs peer authentication.
- Provides top security (e.g., protects against buffer overflow vulnerabilities in the target application).
- Difference between forward proxy (egress) and reverse proxy (ingress).
- Rule example:

```
deny dangerous HTTP methods "PUT, DELETE deny"
```

- SMP (Symmetric Multiprocessing) may improve performance.

- **Pros:**
  - Rules are more fine-grained and simpler compared to those of a packet filter.
  - Provides more protection for the server.
  - May authenticate the client.
- **Cons:**
  - Every application requires a specific proxy.
  - Delay in supporting new applications.
  - Heavy on resources (many processes).
  - Low performance (due to user-mode processes).
  - Completely breaks the client/server model.
  - Not transparent to the client.
  - The proxy's OS may be vulnerable to attacks.
  - Problems with Application-Level Security Techniques that Do Not Permit Traffic Inspection (e.g., TLS)

Variants of Application-level Gateway:

- **Transparent Proxy:**
  - Less intrusive for the client.
  - Requires additional work (packet rerouting and destination extraction).
- **Strong Application Proxy:**
  - Checks semantics, not just syntax.
  - Only some commands/data are forwarded, based on deeper inspection.
  - This is the only correct configuration for a proxy in cases requiring high security.

HTTP Proxy

Forward Proxy

- HTTP Server Acting as a Front-End:
- Acts as an egress control, passing requests to the real (external) server.
- **Benefits** (in addition to network ACLs):
  - Shared cache of external pages for all internal users.
  - Authentication and authorization of internal users.
  - Various controls, such as allowed sites, transfer direction, data types, etc.

Figure 1.2: Forward proxy example.

Reverse Proxy

- Acts as a front-end for the real server(s), forwarding requests to them.
- Implements network ACL and content inspection.
- **Additional Benefits:**
  - Obfuscation: Hides information about the real server(s).
  - TLS Accelerator: Handles TLS encryption, leaving unprotected connections between the proxy and the backend servers.
  - Load Balancer.
  - Web Accelerator: Caches static content.
  - Compression.
  - Spoon Feeding: Retrieves a full dynamic page from the backend server and delivers it to the client based on its speed, offloading the application server.

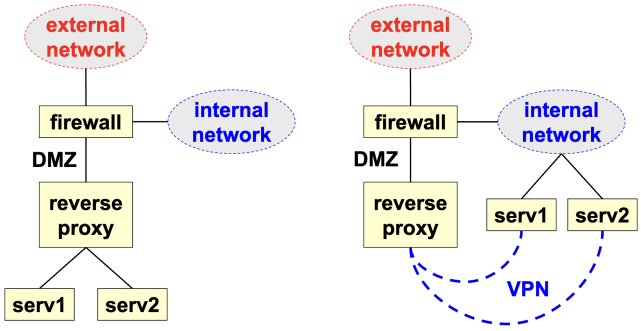

Figure 1.3: Reverse proxy example.

1.4.4 WAF

(Web Application Firewall)

The implementation of this module enables a proxy to function as an application firewall.

The large use of web applications leads to an increase in threats targeting them.

- A WAF is a module installed at a proxy (either forward and/or reverse) to filter application traffic.
- Filters the following types of traffic:

  - HTTP commands.
  - HTTP request/response headers.
  - HTTP request/response content.

- **ModSecurity:**
  - A popular plugin for Apache and NGINX, which power about 50% and 30% of world-wide HTTP servers, respectively.
  - Includes the **OWASP ModSecurity Core Rule Set (CRS)** to protect against a wide range of attacks.

1.5 Firewall Architectures

1.5.1 Packet Filter

Control the flow of traffic between networks or network segments. It operates at the network layer (Layer 3) inspecting each packet's header and making decisions based on predefined filtering rules. The architecture of a packet filter is designed to act as a firewall that filters traffic based on IP addresses, protocols, ports, and other packet header attributes, without inspecting the payload data.

Beware

Simple, cost-effective, but... insecure!

- Exploits the packet filter to screen traffic at both the IP and upper layers.
- The Packet Filter element represents a single point of failure.
- If implemented with a router, it becomes a "screening router," eliminating the need for additional dedicated hardware.
- No need for a proxy, thus no modification of applications is required.

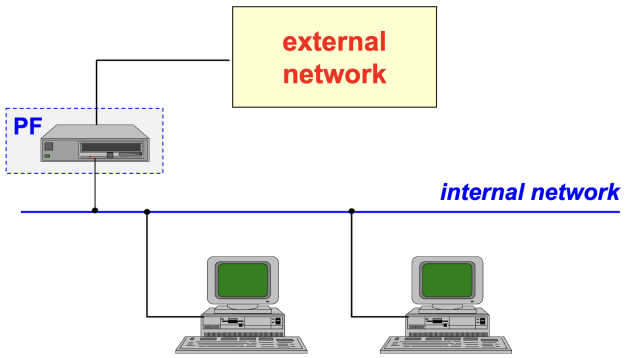

Figure 1.4: Packet filter architecture.

1.5.2 Dual-homed Gateway

A Dual-Homed Gateway uses a bastion host with two NICs (Network Interface Cards) to connect to two different networks, typically a trusted internal network and an untrusted external network (such as the internet). This configuration provides a layer of security by isolating and controlling the communication between the two networks.

- Easy to implement.
- Small additional hardware requirements.
- The internal network can be masqueraded.
- Inflexible: The packet filter cannot easily adapt to changing network requirements or policies, and it does not provide much flexibility in managing traffic.
- High work overhead.
- The border router performs an initial screening of the traffic.
- The bastion host (gateway) could become a bottleneck, reducing the overall performance of the system.

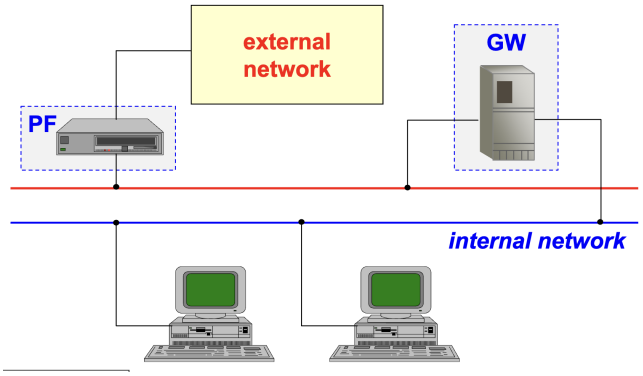

Figure 1.5: Dual-homed gateway architecture.

1.5.3 Screened Host

Uses two primary components: a router and a bastion host. This setup aims to provide enhanced security by controlling the flow of traffic between the internal and external networks. Below are the characteristics and key features of this architecture:

- Router:
  - Acts as a filtering device between the internal (INT) and external (EXT) networks.
  - Blocks traffic from internal to external networks (INT > EXT) unless the traffic is coming from the bastion host.
  - Blocks traffic from external to internal networks (EXT > INT) unless it is directed towards the bastion host.

  - Exceptions: The router allows traffic for directly enabled services, meaning services that are explicitly allowed (e.g., HTTP, DNS) can pass through based on predefined rules.

- Bastion Host:
  - Serves as a secure intermediary and runs either a circuit gateway or an application gateway.
  - Controls access to the authorized services (e.g., web servers, FTP servers) by inspecting and filtering traffic at a deeper level.
  - Ensures that only legitimate services are accessible to the external network, protecting the internal network from unauthorized access.

- Pros and Cons:
  - More Expensive and Complex to Manage: The setup is more complex because it involves managing two systems (router + bastion host), which increases both the cost and administrative overhead.
  - More Flexible: The architecture offers flexibility because it allows skipping control over some services or hosts. For example, some services may bypass the bastion host if specifically configured to do so, making it easier to manage certain use cases.
  - Limited Masking: Only the hosts and protocols that go through the bastion host can be masked for security (such as hiding internal IP addresses or data). However, if the packet filter (PF) uses NAT (Network Address Translation), it can mask additional traffic and hide internal network details.

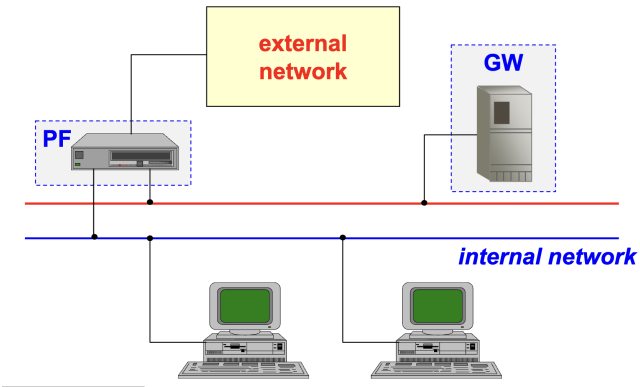

Figure 1.6: Screened host architecture.

1.5.4 Screened Subnet

The most secure solution, however very complex and expensive. Creates a buffer zone between an internal network and an external network (such as the internet). The screened subnet typically involves two firewalls and a DMZ (Demilitarized Zone) to filter and control the flow of data between the internal network and external network. This setup ensures that external entities can access certain resources without directly exposing the internal network, providing an additional layer of security.

Beware

The bastion host must not be in the internal network.

The two packet filters should not have the same implementation (i.e., they should not be from the same vendor).

DMZ

The De-Militarized Zone (DMZ) is home not only to the gateway but also to other hosts, typically the public-facing servers.

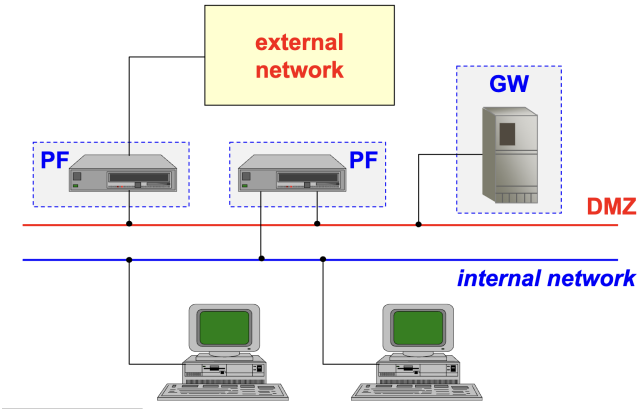

Figure 1.7: Screened subnet architecture.

1.5.5 Screened Subnet V2

(AKA Three-Legged Firewall)

In this version of the **Screened Subnet** architecture, the **packet filter (PF)** and **gateway (GW)** functions are often combined into a single device. This version is cheaper but still presents a single point of failure.

Three-Legged Design: This setup involves a firewall device with three network interfaces.

- One interface connecting to the **external network** (untrusted),
- One interface connecting to the **internal network** (trusted),
- One interface connecting to the **DMZ** (where public-facing services reside).

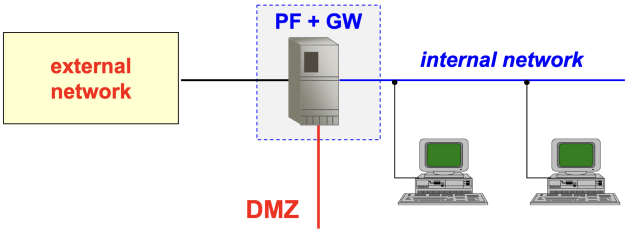

Figure 1.8: Three-legged architecture.

1.6 Local and Personal Firewall

Is a firewall (typically, a packet filter) installed directly on the device or server to be protected.

$$Local \rightarrowtail Server$$

$$Personal \rightarrowtail User\ Device$$

Key points:

- Differs from network firewalls, which are typically installed at the network perimeter. In fact, my limit the processes that are permitted:
  - It can limit processes opening network channels (acting as a client).
  - It can limit processes answering network requests (acting as a server).
- It is usually implemented as a packet filter, controlling the data packets sent to and from the protected device.
- Security Goals:
  - Helps limit malware and Trojan diffusion within the device or network.
  - Reduces the risk of configuration mistakes that might expose vulnerabilities.

Beware

For the firewall to remain effective, its management must be separate from system management. This separation prevents unauthorized users or processes from tampering with firewall settings.

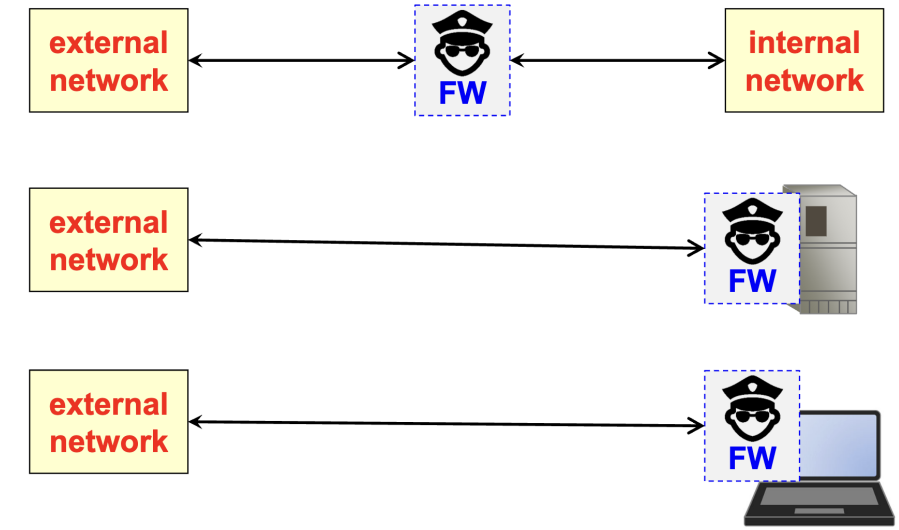

Figure 1.9: Network, Local and Personal firewall.

1.7 Firewall Security Features

This section explains the protection scope and limitations of a firewall:

- Effectiveness of a Firewall:
  - A firewall is fully effective only against attacks targeting blocked channels (e.g., ports or IP ranges that it is configured to deny).
  - Be aware! It cannot protect against threats coming through allowed channels, as those are intentionally left open for communication.
- For the channels left open, other security measures are required to handle potential threats:
  - VPN (Virtual Private Network)
  - Semantic Firewalls or IDS (Intrusion Detection Systems): Analyzes the meaning or behavior of traffic to detect and block malicious activity.
  - Application-Level Security: Protects specific applications by adding layers of security (e.g., input validation, authentication mechanisms).

1.7.1 IDS

(Intrusion Detection System)

An IDS is a security system designed to detect:

- Unauthorized users attempting to access a system or network.
- The actions of unauthorized users within the system.

It can also be extended to identify authorized users who violate their privileges, such as accessing data or performing actions beyond their granted permissions.

Core assumption: **Unauthorized users exhibit different behavioral patterns compared to authorized ones.**

Functional features:

- Passive IDS:
  - Focuses on detection without direct interaction or intervention.
  - Effect Detection: Identifies the impact of an attack, such as file changes detected through tools like cryptographic checksums or Tripwire.
  - Pattern Matching: Compares traffic or payloads against known attack or malware signatures to identify threats.
- Active IDS:
  - Takes a more dynamic and proactive approach to intrusion detection through the following processes.

    * Learning: Uses statistical analysis to understand and establish baseline system behavior over time.
    * Monitoring: Actively collects statistical information about traffic, data flows, sequences, and user actions.
    * Reaction: Compares real-time activity against the established statistical parameters. Initiates a reaction or alert when certain thresholds (e.g., anomalous traffic or behavior) are exceeded.

Usually, we have a hybrid strategy, combining both passive and active approaches, and we need a bit of tolerance for false positives to ensure potential threats are not overlooked.

Topological features:

- HIDS (Host-Based IDS):
  - Operates on individual hosts (e.g., servers, workstations).
  - Log Analysis: Examines logs generated by the operating system (OS), services, or applications to detect suspicious activities or patterns.
  - Internal OS Monitoring Tools: Uses tools within the OS to track file integrity, process activity, and other host-level indicators of compromise.
- NIDS (Network-Based IDS)

1.7.1.1 NIDS

(Network-Based IDS)

Monitors and analyzes network traffic across a network segment or the entire network. Uses Network Traffic Monitoring Tools, which capture and examine packets flowing through the network to detect unusual patterns, malicious payloads, or unauthorized access attempts.

Components:

- Sensor: The sensor is responsible for monitoring network traffic and logs to detect potential security threats.
  - Traffic and Log Analysis: It scans the network traffic and logs for suspect patterns that might indicate an attack or malicious activity.
  - Security Event Generation: When suspicious patterns are detected, the sensor generates security events to report them.
  - Interaction with the System: The sensor can take actions based on detected threats, such as modifying Access Control Lists (ACLs) or triggering a TCP reset to disrupt a malicious connection.
- Director: The director is the central component that coordinates and manages the NIDS system.
  - Sensor Coordination: It manages the interaction and operation of multiple sensors distributed across the network.
  - Security Database Management: The director manages a database that stores security events, logs, and configurations for future analysis and reporting.

- IDS Message System: This component ensures secure and reliable communication between the various NIDS components (sensors, director, etc.).
  - It ensures that data and alerts are transmitted between components securely, preventing tampering or loss of critical information.

Architecture:
The workflow of a Network-Based Intrusion Detection System (NIDS) architecture involves several steps (fig.1.10):

- Sensors:
  - Capture network traffic (packets) and log information from various points within the network.
  - Compare the network traffic and logs to a set of predefined attack signatures or anomaly patterns.
- Anomaly Detection: If the traffic deviates from normal behavior, the sensor may identify it as an anomaly (e.g., a DDoS attack, port scanning, or other unauthorized access attempts).
- If suspicious activity is detected, the sensor generates security events and sends them to the IDS director.
- The director analyzes and correlates incoming events from multiple sensors, looking for patterns or sequences of activities that indicate a more significant attack (e.g., a coordinated attack across multiple sensors).
- Immediate Actions: Based on predefined rules, the director can initiate automatic responses.

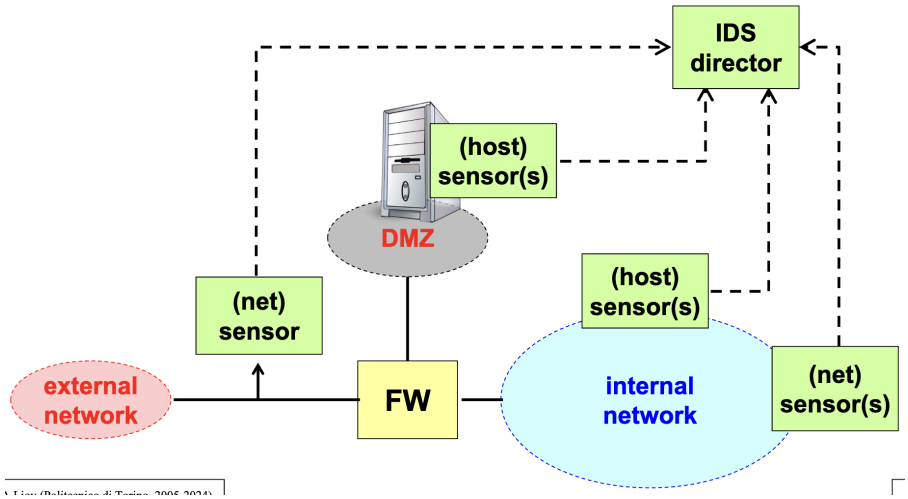

Figure 1.10: Network-based IDS architecture.

1.8 Other Firewall Implementations

1.8.1 IPS

(Intrusion Prevention System)

An IPS combines Intrusion Detection with the capability to actively prevent detected threats. The goal is to speed up and automate the response to intrusions by integrating an IDS (Intrusion Detection System) with a distributed dynamic firewall, which can take immediate actions to block malicious traffic.

Beware

Dangerous! IPS systems can erroneously block or allow traffic, potentially causing harm to the network or users if the wrong decision is made.

- An IPS is not just a standalone product; it's a technology that can be integrated into the overall security infrastructure to provide real-time detection and mitigation of attacks.
- Many modern security systems integrate IDS and IPS into a single product known as IDPS (Intrusion Detection and Prevention System).

1.8.2 NGFW

(Next-Generation Firewall)

Extends traditional firewall functionality by integrating advanced features that provide deeper inspection and more granular control over network traffic.

Key features:

- Application Identification:
  - Can identify and control applications regardless of the port or protocol used (w.r.t. classic firewalls an NGFW analyzes also traffic patterns on the application layer).
  - Deciphering/Re-ciphering Traffic: NGFWs can perform SSL/TLS decryption to inspect encrypted traffic, allowing them to detect hidden threats in secure communications.
- User Identification:
  - NGFWs can integrate with user authentication mechanisms, enabling them to enforce policies based on the user rather than just IP addresses.
  - Integration with: Captive Portals (e.g., Wi-Fi), 802.1x and end-point authentication (e.g., Kerberos, Active Directory, and LDAP)
- Per-User and Per-Application Policies:allow the creation of security policies that are user-specific and application-specific (For example, an NGFW could block social media applications for some users while allowing them for others).
- Filtering based also upon known vulnerabilities, threats, malware

1.8.3 UTM

(Unified Threat Management)

Integrates multiple security functionalities into a single device. This approach simplifies network security by combining various protections into one solution, making it easier to manage and often more cost-effective.

Key features:

- Common capabilities: firewall, VPN, anti-malware, content-inspection, IDPS (Intrusion Detection and Prevention System).
- Actual capabilities depend upon the manufacturer.
- Mainly targeted to reduce the number of different systems, hence the management complexity and the cost

1.8.4 Honey Pot / Honey Net

Security tools designed to deceive attackers by creating artificial environments that simulate vulnerable systems or networks. Their main purpose is to attract and observe malicious activity, helping security professionals gather information on attack methods, identify vulnerabilities, and develop better defense strategies.

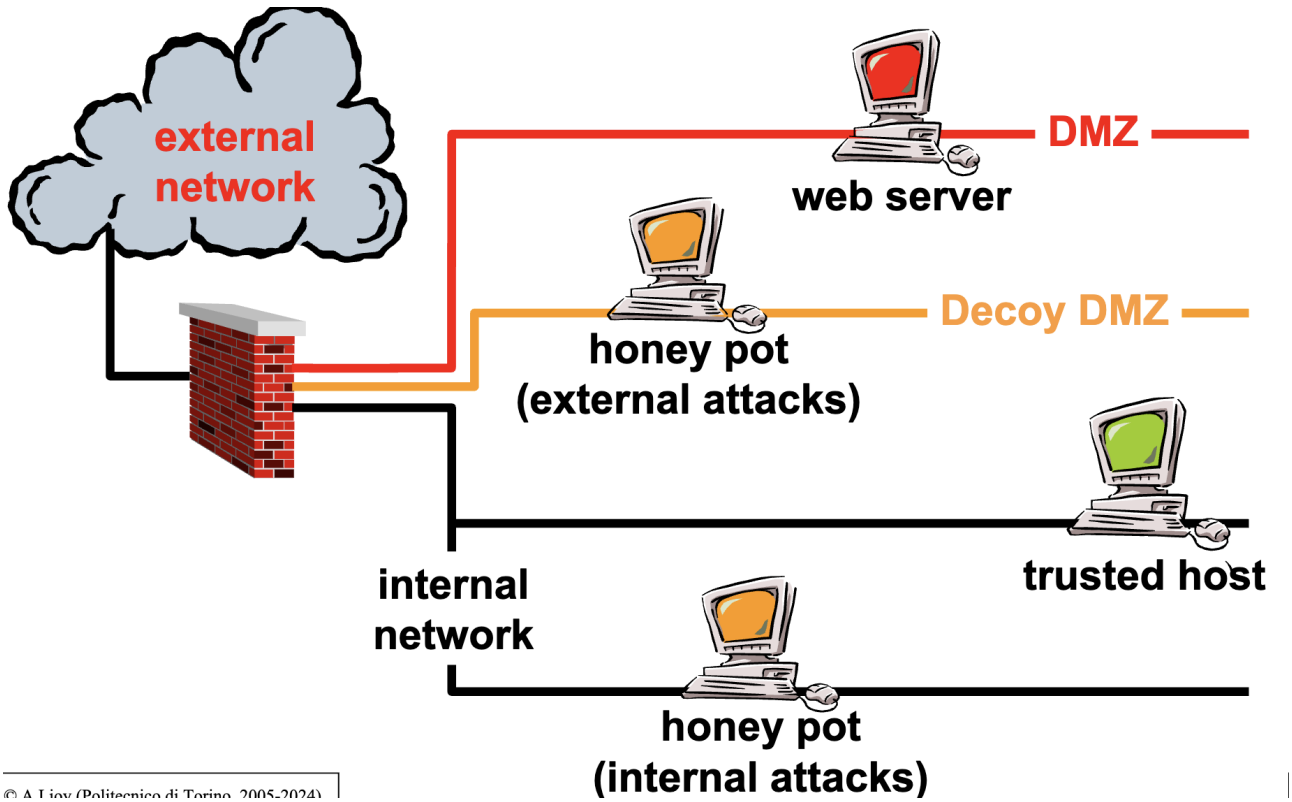

Figure 1.11: Honey pot architecture.

Bibliography

[1] Antonio Lioy. Firewall and ids/ips, 2024.