

1 Crypto Recap

1.1 Objectives:

- Confidentiality
- Integrity
- Authenticity
- Availability
- Authorization
- Non-Repudiation, Accountability
- Freshness
- Anonymity, Unlinkability
- Intervenableity, Contro
- Transparency

1.2 Confidentiality-Encryption

1.2.1 Symmetric Ciphers

- Secret key for en- and decryption
- Much more efficient
- Block cipher:** encryps a plaintext block of fixed len e.g.: *Advanced Encryption Standard (AES)*
- Stream cipher:** encryps a bitstream e.g.: *ChaCha20*

1.2.2 Asymmetric Ciphers

- Public key for encryption
- Private key for decryption
- Ex.: RSA-based encryption

1.3 Integrity, Authenticity-Signatures, MACs

1.3.1 MACs

- Symmetric cryptography
- Protects data integrity & authenticity
- Ex.: Hash-based MAC

1.3.2 Digital Signatures

- Asymmetric cryptography
 - Signing with private key
- Protects data integrity & authenticity
- Provides non-repudation

1.4 Block Cipher Modes of Operation

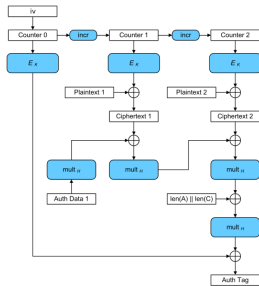
1.4.1 Electronic Code Book (ECB)

- Each plaintext block is encrypted seperately
- Inherintly insecure! -> Smae block = Same cipher

1.4.2 Cipher Block Chaining (CBC)

- Plaintext is chained to previous ciphertext by XOR and encrypted afterwards
- Difficult to apply securely -> implementations often vulnerable

1.4.3 Galois Counter Mode (GCM)



2 Tranport Layer Security (TLS)

2.1 TLS handshake protocol

- Parameter Negotiation
- Key exchange
- Authentication

2.2 TLS record protocol

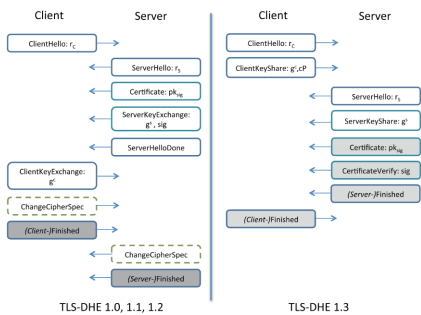
- Protection of integrity, authenticity and confidentiality
- Symmetric Cryptography:** e.g., block cipher, usually AES

2.3 Attacks

- Attacks on the record layer
- Attacks on the session key
- Attacks on the private server key
 - Attacks on implementations: Timing Attacks, Heratbleed, Invalid Point Attacks
- Attacks on TLS eco system

- Attacks on certificates and the PKI
- Attacks on the browser GUI

2.4 Overview TLS 1.0-1.2 & TLS 1.3



2.5 HKDF Key Derivation Function

2.6 Forward Secrecy

- TLS 1.3 using certificate-based Authentication: **forward secrecy**
- TLS 1.3 using pre-shared keys (PSK):
 - PSK using *elliptic-curve* Diffie-Hellman: **forward secrecy**
 - PSK without EC-DHE (symmetric-only) and zero-round-trip data: **no forward secrecy**

2.7 Datagramm TLS

- DTLS is identical to TLS where possible
- DTLS has to introcure new mechanisms
 - Explicit sequence numbers
 - Retransmission timer for handshake message
 - Message re-ordering for the handshake
 - Fragmentation of large handshake messages into severall DTLS records
 - Optional replay detection
 - Invalid records are discarded (silently)
 - Denial-of-Service countermeasures: statless cookies, HelloRetryRequest

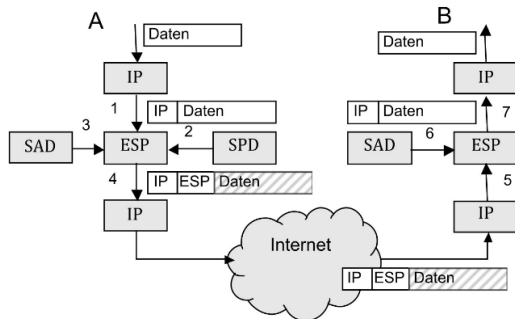
3 Network Layer Security: IPsec

3.1 TLS limitations

- Does not protect all traffic between hosts
 - Protects only a single connection; new connection => new key agreement or session resumption necessary
 - Non-TCP traffic (resp. non-UDP traffic for DTLS) cannot be protected
- Does not protect the TCP layer (RST attacks on TCP are possible which terminate TLS connection)
- Application specific: applications have to be modified to use TLS/DTLS
- Using a TLS-based tunel for VPNs has disadvantages (TCP) (DTLS good option tho)

3.2 Overview IPsec

- Goal: Protect IP packets cryptographically
 - Confidentiality
 - Integrity & Authenticity
- Seperation of packet protection from key exchange
 - Protocols for packet encryption and authentication: symmetric crypto, implemented in the TCP/IP stack
 - Key Exchange completely unrelated: asymmetric (+symmetric) crypto, implemented as network daemon



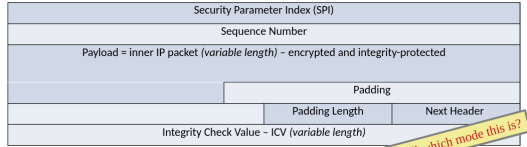
3.3 Packet Formats and Operational Modes

3.3.1 Authentication Header (AH)

- Integrity & Authenticity only
- Partially protects outerIP header
 - Variable fields cannot be included
 - Flags, Fragment Offset, TTL, Header Checksume
 - Other fields are included in MAC computation
 - Version, IP header length, total length, identification, Protocol (must be AH), source/dest Address

3.3.2 Encapsulating Security Payload (ESP)

- Confidentiality, Integrity & Authenticity
- Does not protect outerIP header



- SPI: SA-Identifier
- Next Header: type of payload data (e.g. IPV4)
- Integrity Check Value: data for authentication / integrity protection
 - Message Authentication Code (MAC)
- SAs are negotiated in the key exchange
- Sender side: Packet must be categorized to determine which SA applies (By parameters such as destination=)
- Receiver side: Determines SA from IPsec header of received packet
- SA Parameters:
 - IPsec protocol (AH or ESP)
 - Authentication algorithm and key
 - Encryption algorithm and key
 - IPsec mode (transport or tunnel)
 - SA lifetime
 - Sequence number: current counter
 - On reciver side: SSliding Window"for replay protection

3.4 Scenarios

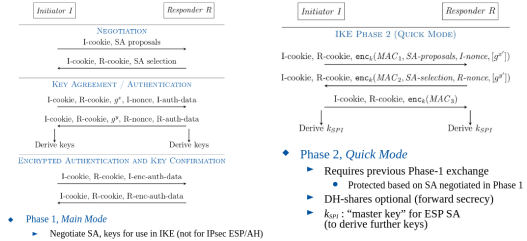
- Host to host
- Host to gateway
- Gateway to gateway

3.5 The Internet Key Exchange (IKE)

- So far, keys are already exchanges, SAs negotiated... but how?
- Authentication of the communication endpoints
- Dynamic negotiation fo algorithms and parameters
- Key establishment
 - Forward Secrecy
- Resistance to DoW attacks
- Efficiency (computation, messages, round trips)

3.6 IKEv1

- Phase 1: main mode vs. aggressive mode
- Authentication modes:
 - Digital Signatures
 - Public key encryption(PKE): two variants (rarely used)
 - Pre-shared keys(PSK)
- Phase 2 (quick mode)
 - With DH (=> forward secrecy)
 - Without DH (=> no forward secrecy)



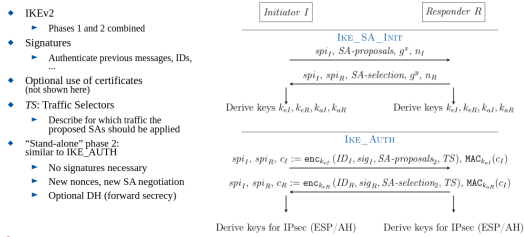
3.6.1 Discussion

- Complicated
 - Information from severall (complex) RFCs required
 - Many options variants
- Generic
 - Clean separation of different phases and functionalities
 - Can be advantageous in theory, but leads to inefficiencies
 - Intended as general handshake and key agreement protocol, not only for IPsec
 - In practice: only used for IPsec

- Inefficient
 - Requires too many round trips
- Inadequate DoS protection
 - IKEv1 uses statelful cookies

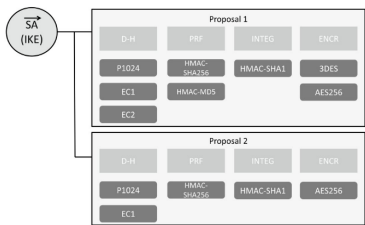
3.7 IKEv2

- Phase 2 (partially) combines with Phase 1
 - More efficient than IKEv1
 - Initial negotiation of IPsec SA included
- Additional SAs can be negotiated in further Phase 2 protocol runs
- Simplified specification
 - Essential information in one RFC
 - No distinction main mode vs. aggressive mode
- DoS protection using statless cookies: optional



3.8 SA Proposals

- Serverl proposals possible
- Proposal contains transforms
- Different options for each transform possible
 - Exactly one transform has to be selected



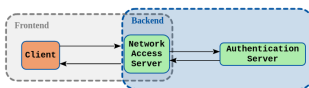
4 Security on Layer 2 (MAC Layer, Ethernet)

4.1 Attacks

- MAC address spoofing
- ARP spoofing
- ARP cache poisoning

4.2 Network Access Control

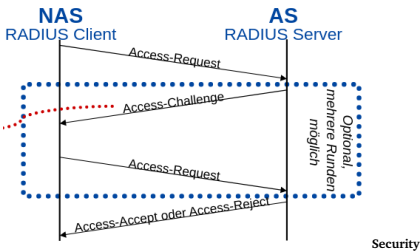
- Frontend protocols: Client -> Network Access Server
 - Point-to-Point connections: PPP
 - LAN: PPPoE, EAPoL
 - WLAN: WPA/WPA2/WPA3, EAPoL
- Backend protocols: Network Access Server -> Authentication Server AAA protocols (Authentication, Authorization, Accounting)
 - Radius, Diameter
 - TACACS+



4.2.1 Radius - Remote Authentication Dial-In User Service

- Backend protocol
- Dial-In Server
 - E.g. DSL
 - Company infrastructure

Packet Format



- Old protocol
 - Tries to protect sensitive data only
 - Insecure cryptography
- Uses only over secure protocols
 - Usually TLS
- RADIUS successor: Diameter
 - More flexible, extensible, application-aware
 - SSecurity features removed, no more insecure crypto of RADIUS => To be used over secure protocols TLS/DTLS/IPsec

4.2.2 Point-to-Point Protocol (PPP)

- PPP: Frontend protocol to connect two devices
 - Typically used for dial-up connections (e.g. DSL)
 - Transported over Layer 2 protocol (e.g. ethernet)
- Authentication mechanisms
 - PAP: username/password
 - CHAP: Challenge Handshake Authentication Protocol
 - EAP: Extensible Authentication Protocol
- PPPoE: PPP over Ethernet (e.g. used for DSL connections)
 - Discovery: Client negotiates session with network access server
 - PPP Session: PPP frames are encapsulated in ethernet frames; potentially several hops

4.2.3 Extensible Authentication Protocol (EAP)

- Extensible authentication framework
 - Fronted protocol for client authentication
 - Server send request (e.g. challenge, ident. request, ...)
 - Client responds
 - Server sends success or failure message
 - EAP messages can be transported in RADIUS messages (as RADIUS attribute) for backend communication
- EAPoL: EAP over LAN

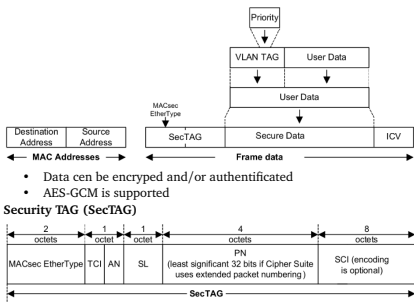
4.2.4 Port Based Network Access Control (PNAC)

- Authentication of clients before they can use the network (LAN)
 - Client connect to the network
 - * Authentication traffic is allowed
 - Switch port can only be used for other purposes after successful authentication
 - Re-authentication after timeouts, link-down, etc.
- PNAC uses EAPoL
- Terminology: Port Access Entities (PAE)
 - Client: supplicant
 - Network access server: authenticator
 - Attention: "port"

4.3 MACsec

- cryptographically protect the ethernet layer
- Based on MACsec Key Agreement (MKA)
 - After successful authentication (typically EAP)
 - Derives key from CAK - connectivity association key (pre-shared)
 - Key can be set up by EAP (e.g. EAP-TLS)

4.3.1 Frame



- TCI: TAG Control Information

- AN: Association Number
- SL: Short Length (payload length; for short frames only)
- PN: Packet Number
- SCI: Secure Channel Identifier (identifies SA, when CA needs more than 4)
 - SA is identified by SCI (optional) and AN

4.4 Summary

- Security on Layer 2 can protect all higher Layer
 - But: only in same network
 - Does not work across Layer 3 routers
- Network Access Control protects access to the network
 - Client Authentication
 - RADIUS/DIAMETER
 - EAP
 - Port-Based Access Control
- MACsec protects traffic
 - Encryption & authentication

5 Wireless Security

5.1 WiFi Security

5.1.1 Historic Overview

- 1999: WEP (Wired Equivalent Privacy)
 - Goal: as secure as a wired LAN
 - Insecure, various attacks known
- 2003: WPA (WiFi Protected Access)
 - Improved protocols; most known attacks on WEP prevented
 - Enterprise Mode
 - But: Requirement of hardware-compatibility with WEP devices => Encryption improved, but still based on obsolete stream cipher
- 2004: WPA2 (still used)
 - Similar to WPA, but AES-based encryption: AES-CCMP
- 2018: WPA3 (supported by new devices)
 - Several improvements: Prevention of offline-attacks on pre-shared keys, forward secrecy, encryption for open WLANs

5.1.2 WPA/WPA2 Security

- Personal Mode
 - Pre-Shared Keys
- Enterprise Mode
 - EAP-TLS, PEAP, EAP-TTLS
- AES-CCMP

- Authenticated Encryption with Associated Data (AEAD)
- AES-CCM
 - Authentication: CBC-MAC
 - Encryption: Counter Mode (CTR)
 - MAC and encryption: computed simultaneously

4-Way Handshake

- Based on Pairwise Master Key (PMK)
 - Personal Mode (WPA-PSK): Computed from passphrase and SSID (as fSalt) using PBKDF2 (password-based key derivation function)
 - Enterprise Mode: Established by key exchange protocol (e.g. EAP-TLS, PEAP)
- 4-Way HS to derive Pairwise Transient Key (PTK)
 - Exchange nonces
 - PTK is derived by hashing PMK, nonces, MAC address
 - Further key (for different purposes) derived from PTK
 - Client gets Group Temporary Key from AP (encrypted)
 - Message Integrity Code (MIC): MACs for integrity protection, key confirmation
- KRACK (2018): Key reinstallation attacks => meanwhile prevented by software/firmware updates
- Problem: Offline attacks against passphrase

5.1.3 WPA 3 Improvements

- Mandatory Protected Management Frames
 - Prevents deauthentication attacks (DoS)
- Replace PSK Authentication with SAW protocol
 - Simultaneous Authentication among Equals (SAE): "Dragonfly" handshake
 - Prevents offline attacks on passphrase
 - Based on elliptic curve cryptography by default
- Forward Secrecy based on Diffie-Hellman
- 192-bit Security Mode (optional)
 - AES-256 (GCM)
 - SHA-384
 - 284-bit elliptic curves or RSA with at least 3K bits

5.1.4 Simultaneous Authentication among Equals

- SAE "Dragonfly" authenticates participants and establishes PMK
 - Based on passphrase and (EC-)Diffie-Hellman
 - Can be initiated simultaneously by both parties (useful for mesh networking)
- 4-Way Handshake
 - Establishes PTK based on PMK
 - Same as in WPA2
 - But now: PMK with much higher entropy => Offline attacks not practical
- Hash-to-Group: "Hunting and Pecking"
 - Generate point on elliptic curve from passphrase (and MAC addresses, etc.)
 - Cryptographic hash function generates pseudo random numbers (by including a counter in the input)

- * Both parties must use the exact same inputs in the same order
 - Fixed procedure to derive x-coordinate
 - * Check if point on curve can be generated
 - * If check fails: increase counter and try again

- Auth-Commit messages
 - Exchange ECDH shares
- Auth-Confirm messages
 - Key confirmation, authentication of messages

5.2 Bluetooth Security

- Authentication: device authentication, no user authentication
- Pairing/bonding: create shared keys; used in connections later on
- Confidentiality: encryption of BT communication
- Message Integrity: MACs (authenticated encryption) to protect BT communication
- Authorization: control access to resources (based on devices, not users)
- Security Modes
 - Mode 1: no security
 - Mode 2: service level (only for backward compatibility)
 - Mode 3: link-level enforces security (only for backward compatibility)
 - Mode 4: authenticated link key using SSecure Connections*, based on device pairing
- Eavesdropping not trivial: Bluetooth uses frequency hopping (not a security feature)

5.2.1 Device Pairing

- Authentication and generation of link key / long term key
- PIN/Legacy Pairing: enter PIN on both devices
 - Key generation based on PIN, device address, and random values
- Secure Simple Pairing (SSP): since Bluetooth 2.1
 - Numeric Comparison
 - * Compare 6-digit numbers
 - Passkey Entry
 - * Read 6-digit from one device, enter on the other one
 - Just Works
 - * User accepts connection without verification
 - Out of Band (OOB)
 - * Transmit data using other communication channels (e.g. NFC)

5.2.2 Simple Secure Pairing (SSP)

- Unauthenticated ECDH
- 2-Stage Authentication
 - Stage 2: depends on pairing method
 - Stage 2: Cryptographic authentication based on Stage 1 values and ECDH secret
- Key derivation to generate link key / long term key

5.2.3 Secure Authentication

- Paired (bonded) devices authenticate each other
- Challenge-Response scheme
 - 128-bit random challenges
 - Response: HMAC of BT addresses and challenges (using link key from pairing)
 - * Before Bluetooth 4.1: based on Bluetooth-specific algorithm E1
- Authentication failure: introduce delay (exponential back-off)

5.2.4 Confidentiality

- Bluetooth-specific stream cipher E0
 - Designed for efficiency
 - Serious attacks have been published
 - "Practical theory (but complex, hard to apply in practice)"
- AES-CCM
 - Used since Bluetooth 4.1
 - Key derived from link key (pairing) and the authentication step

5.2.5 Privacy

- Privacy problem: Devices (users) can be identified by Bluetooth MAC addresses
- Mitigation: BLE private device addresses
 - Resolvable Private Address (RPA) is changed periodically
 - Identity Address remains constant (but is not transmitted over the air)
 - Identity Resolving Key to map RPA to Identity Address
 - Especially important to discoverable devices (which advertise identity info)

5.2.6 5.x Security

- No major changes to security protocols and algorithms
- Bluetooth 5.0
 - PHY improvements, no relevant security changes
- Bluetooth 5.1
 - HCI support for debug keys (should not be relevant in production systems)
- Bluetooth 5.2: adds new features (Extended Attributes, Isochronous Communication, ...)
 - Isochronous communication: connection-oriented or connection-less
 - * Group communication: group keys need to be established
 - * Broadcast Authentication
- Bluetooth 5.3: Key Size Negotiation
 - Enables host to define minimum key size

5.2.7 BLUFS

- BLUFS: New attacks against bluetooth
 - Breaks Forward Secrecy and Future Secrecy
 - Enables man-in-the-middle attacks, impersonation if one session key compromised
 - Forces weak key: spec allows minimum of 7 Bytes entropy (56 Bits)
 - * Brute-force attack: offline, parallelizable

- * Forces reuse of compromised key
 - Attack against bluetooth spec (BR/EDR: "Bluetooth Classic") versions 4.2 to 5.4): All compliant devices are affected
 - Published and presented at ACM CCS 2023

5.2.8 implementations Vulnerabilities

- BlueBorn (2017): Collection of implementation Vulnerabilities
 - On Windows, iOS, Linux, Android
 - Buffer overflow, integer overflows, ...
- Android (2018): implementation flaws in L2CAP and SMP
 - Remote Memory Disclosure
- BleedingTooth (2020): several bugs in Linux
 - Can even lead to arbitrary code execution in kernel mode
- Windows (2021): BT Driver Elevation of Privilege
- BrakTooth (2021)
 - Bluetooth controllers: SoC firmware Vulnerabilities (Link Manager)
 - Estimation 1400 bluetooth chips/modules affected

5.2.9 Summary

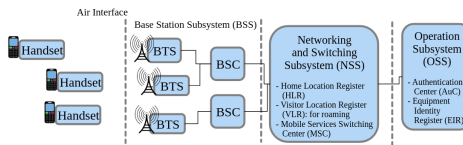
- Complex protocol stack, not easy to implement
- Many attacks in the past
 - on cryptography algorithms
- Bluetooth versions before 2.1 are basically completely insecure
- Bluetooth versions since 4.2 are relatively secure (...but: "BLUFS")
 - But the implementations not necessarily!
- Bluetooth 5.2 architecture similar to 4.x
 - Introduces new features and minor security improvements

6 Security Mobile Networks

6.1 Overview

- **1st generation: analog telephony networks** (voice only)
- **2nd generation (2G): GSM 1990s**
 - Digital, circuit-switched, SIM-based; introduced **cryptographic protection**
 - Generation 2.5: **GPRS** (packet-switched data traffic)
 - Generation 2.75: **EDGE** (increased bandwidth)
- **3rd generation (3G): UMTS 2000s**
 - Much higher bandwidth
 - Improved security: **mutual authentication**, better crypto algorithms
 - Generation 3.5: **HSDPA/HSUPA** (increased bandwidth)
- **4th generation (4G): LTE 2010s**
 - Higher bandwidth
 - Completely new **IP-based** core network
 - Changes to security
- **5th generation (5G): 2020-?**
 - ... even better, of course ;) ...

6.2 GSM Network



BTS: Base Transceiver Station

BSC: Base Station Controller

- Circuit-switched network, designed for voice communication
 - Stable channel between communication partners is reserved
 - Separate channel for signaling (different time slices)
 - Slow, expensive
 - GSM introduces SMS: small texts can be transferred in signaling channel

6.2.1 Security

- Goal: At least as secure as a landline"
- Security functions
 - Authentication and Key Agreement (AKA)
 - * Shared secret in AuC and SIM
 - * Authentication via challenge-response protocol
 - * Agreement on short term session keys for encryption
 - Confidentiality Protection
 - * Encryption of user data ("user plane")
 - * Encryption of signalling ("control plane")
 - Integrity Protection
 - * Only for signalling
 - * Often implicitly by encryption

6.2.2 (U)SIM

- SIM card introduced with GSM
 - **SIM: Subscriber Identity Module**
 - **International mobile subscriber identity (IMSI)**
 - * Temporary mobile subscriber identity (TMSI): derived from IMSI
 - **Security token (Smartcard)** of the network operator
 - * Typically implemented on a JavaCard

- **Network Operators and 3rd Parties Offering SServices[™]:**
 - **Network Functions (NF):** Core network features exposed as services.
 - **Other Services:** Additional functionalities provided by the network.
 - **TLS Support:** Ensures secure communication between entities.
 - **OAuth 2.0-based Authorization:** Allows secure access control using access tokens.
- **Network Slices:**
 - Provides virtualized and isolated network partitions with specific **QoS (Quality of Service)** properties.
 - Uses **EAP-based Authorization** for access control.

7 Firewalls, Intrusion Detection and Prevention

7.1 Firewalls

- Firewalls separate networks
 - Restricts traffic: enforces policy
 - Logging
 - Implemented in SW and/or HW

7.1.1 Packet Filtering

- Packet filtering firewalls:**
 - Policies defined by filtering rules (source, destination, port, protocol, etc.).
 - Rules decide actions: accept, block, log, or modify packets.
 - Advanced filters can modify packets (e.g., NAT or port translation).
- Advantages:**
 - Cheap, widely available, and efficient.
 - Examples: Linux netfilter, Windows firewall.
 - Simple implementation; no modification to applications required.
- Limitations:**
 - Susceptible to IP and port spoofing attacks.
 - Rules limited to individual packets and lack comprehensive state awareness.
 - Cannot enforce application-specific filtering; lacks deeper protocol knowledge.
 - Usually not capable of rejecting specific packets within a particular application session.
- Niche Aspects:**
 - Filters can reject IP spoofing by blocking packets from internal addresses arriving at external interfaces.
 - Access control by ports can restrict services to specific source addresses (e.g., SMTP limited to internal IPs).
 - Can filter based on packet type (TCP/ UDP) or flags (e.g., SYN for connection establishment).

7.1.2 Connection Tracking and Stateful Firewalls

- Connection Tracking:**
 - Tracks packets belonging to the same logical "connection"(not limited to TCP).
 - Examples:
 - UDP-based connections (e.g., DTLS).
 - DNS requests/responses and ICMP echo/replies.
 - Supports protocols with separate control and data connections (e.g., active FTP).
 - Enables NAT functionality by rewriting addresses for complex protocols.
- Stateful Firewalls:**
 - Maintain state information about active connections.
 - Rules can dynamically adapt based on connection states (e.g., allow replies to outgoing requests).

7.1.3 Linux Firewall: Netfilter

Netfilter, part of the Linux kernel, is a framework for packet filtering and network address translation (NAT). It operates using a series of hooks in the network stack where rules, organized into **tables** and **chains**, are applied.

Chains and Their Functions:

- Input Chain:** Handles packets destined for the local system (e.g., filtering SSH traffic).
- Output Chain:** Manages packets generated locally and leaving the system (e.g., limiting outgoing ICMP traffic).
- Forward Chain:** Processes packets passing through the system but not destined for it (e.g., traffic routing in a gateway).
- Prerouting Chain:** Alters incoming packets before routing decisions (e.g., DNAT for port redirection).
- Postrouting Chain:** Alters outgoing packets after routing decisions (e.g., SNAT for source IP modification).

Key Components:

- Tables:** Contain chains for specific tasks:
 - Filter Table:** Default table for packet filtering (Input, Output, Forward chains).
 - NAT Table:** Used for network address translation (Prerouting, Postrouting chains).
 - Mangle Table:** For modifying packet headers.
- Tools:**
 - iptables: Legacy tool for managing rules.
 - nftables: Modern replacement offering simplified syntax and better performance.

Use Case Example: A packet enters the system through the **Prerouting Chain**, is routed to the appropriate interface (via **Forward Chain** if being forwarded or **Input Chain** if destined locally), and exits through the **Postrouting Chain** after necessary modifications. Netfilters modularity and flexibility make it a powerful tool for managing network traffic.

7.1.4 Application-Level Gateways

An **Application-Level Gateway (ALG)** acts as an intermediary between clients and applications, offering fine-grained control and monitoring of traffic specific to application protocols. It provides a specialized interface tailored for applications, such as SMTP or HTTP, allowing for filtering, logging, and user profiling at the application layer.

Functionality:

- Connection Handling:** Clients connect to the gateway, which filters and analyzes the traffic before forwarding it to the application server.
- Protocol Awareness:** Equipped with detailed knowledge of application-specific protocols, enabling in-depth traffic inspection and filtering.
- State Management:** Tracks the state of connections or application data to enforce security or logging policies effectively.
- In-depth analysis and traffic filtering based on application logic.
- Enables user profiling, attack detection, and intrusion prevention.

- Provides application-specific caching and potential for secure communication.
- App extensions for additional features, such as authentication, without modifying the original application.

Limitations:

- Performance Overhead:** The inspection and filtering process can reduce throughput.
- Maintenance Effort:** Requires frequent updates to stay compatible with evolving application protocols.
- Limited Scope:** Only supports predefined applications, limiting flexibility in heterogeneous environments.

Use Case: ALGs are frequently employed in enterprise environments for managing email servers (e.g., SMTP gateways) or for secure access to critical applications by acting as a filter and logging interface for external traffic.

7.1.5 Proxy Firewalls

Proxy firewalls act as a generic proxy on the transport layer, enabling advanced traffic control and enhancing communication security.

Key Features:

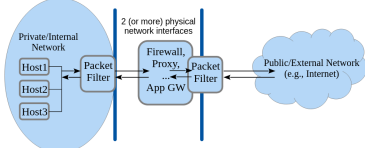
- Connection State Tracking:** Monitors and manages the state of connections.
- Authentication and Security:** Adds user authentication and communication security.
 - Frequently used in corporate environments:
 - Outgoing traffic is routed only via the proxy.
 - User authentication is required before access.
 - Only specific ports or applications are allowed.
- Flexibility:** Can extend functionality toward application-level filtering (e.g., web proxies or caches).

Comparison to Application-Level Gateways:

- Proxy firewalls are more general than application-level gateways.
- The distinction between proxy firewalls and application gateways is not always clear.

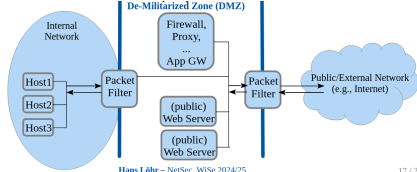
Firewall Architectures: Dual-Homed Firewall vs. DMZ

- Dual-Homed Firewall:**



- Concept:** Firewall with two interfaces separating internal and external networks.
- Operation:** Traffic flows through the firewall, using packet filters or proxies for mediation.
- Advantages:**
 - Strong physical isolation.
 - Centralized traffic control.
 - Flexible integration of security mechanisms.
- Disadvantages:**
 - Requires additional hardware resources.
 - Limited scalability for complex networks.
- Use Cases:** Suitable for small/medium networks needing strict isolation.

- Demilitarized Zone (DMZ):**



- Concept:** Segregated network zone for public-facing services (e.g., web servers).
- Operation:** Packet filters manage traffic to/from DMZ, preventing direct internal access.
- Advantages:**
 - Protects internal networks from compromised public-facing services.
 - Granular traffic control for specific services.
 - Limits exposure of internal systems.
- Disadvantages:**
 - Adds configuration and maintenance complexity.
- Use Cases:** Ideal for hosting public services (e.g., web hosting, e-commerce).
- Key Differences:**
 - Dual-Homed focuses on isolation, while DMZ separates public services from internal systems.
 - DMZ adds a dedicated network segment for external-facing servers.

7.2 Intrusion Detection and Prevention System (IDPS)

- Intrusion Detection Systems (IDSs):**
 - Detect attempts to attack or intrusions.
 - Focused on monitoring and logging suspicious activities.
- Prevention:**
 - React to detected intrusions to prevent successful attacks.
 - Goal: Stop attacks in progress by blocking or responding.

- IDPS = IDS + Prevention:** Combines detection and prevention to ensure proactive and reactive defense mechanisms.
- Historical Work:**
 - James Anderson: *Computer Security Threat Monitoring and Surveillance* (1980).
 - Dorothy Denning: *An Intrusion Detection Model* (1987).

Why Use IDPS?

- Identify incidents effectively.
- Support incident response to minimize damage.
- Identify gaps in security policy and practices.
- Deter insiders from violating policies.

7.2.1 IDPS Functionality (High-Level)

- Record Events:**
 - Logging and accounting for detected activities.
 - Observations useful for detecting and analyzing incidents.
 - Integration with Security Information and Event Management (SIEM) systems.
- Notify Administrators:**
 - Alert administrators to take appropriate action.
- Produce Reports:**
 - Summarize activities and detected threats for analysis.
- Automated First-Level Reaction:**
 - Stop attacks or intrusion attempts automatically.
 - Dynamically change the security environment, e.g., reconfigure firewalls.

7.2.2 Intrusion Detection Model (Denning, 1987)

- Components:**
 - Subjects:** Entities initiating actions.
 - Objects:** Resources being accessed or affected.
 - Audit Records:** Logs of subject actions for monitoring.
 - Profiles:** Behavioral characterizations based on metrics.
 - Anomaly Records:** Highlight abnormal activities.
 - Activity Rules:** Define responses to anomalies (e.g., profile updates, intrusion reporting).
- Metrics:**
 - Event Counters
 - Interval Timers
 - Resource Measures
- Models:**
 - Statistical Models:** Identify statistical anomalies.
 - Mean and Standard Deviation Model
 - Multivariate Correlation Model
 - Markov Process Model
 - Time Series Model
 - Operational Models:** Compare metrics against predefined thresholds.
 - Modern Approaches:** Incorporate Machine Learning/AI for advanced anomaly detection.

7.2.3 Types of IDPS

- Network-Based Intrusion Detection System (NIDS):**
 - Monitors and analyzes network traffic.
 - Often integrated with firewalls (tightly or loosely).
 - Includes specialized Wireless IDPS for wireless networks.
 - Incorporates Network Behavior Analysis (NBA) for deeper traffic insights.
- Host-Based Intrusion Detection System (HIDS):**
 - Monitors activity on a specific host (server or end-user device).
 - Analyzes system and application logs, network activity, running processes, and system calls.
 - Tracks file accesses and configuration changes for anomaly detection.

7.2.4 IDPS Architecture

- Sensors (Agents):** Monitor and analyze activity.
- IDPS Management Server:** Collects information from sensors and manages them.
- Database:** Serves as a repository for audit records.
- Admin Console:** Provides a user interface for administrators.

7.2.5 Detection Methodologies: Categorization

- Signature-Based Intrusion Detection**
 - Uses fixed patterns to identify *known threats* (e.g., hash values of malware).
 - Limited to known threats and can be evaded by intelligent attackers.
- Anomaly-Based Intrusion Detection**
 - Detects previously unknown threats by modeling normal/abnormal behavior.
 - Requires a *training/learning phase* for data collection and profiling.
 - Often uses Machine Learning or AI; main focus of current research.
- Specification-Based Intrusion Detection (Stateful Protocol Analysis)**
 - Compares observations to models/specifications of legitimate behavior.
 - Can be integrated into application-level gateways.

7.3 Summary

- Firewalls**
 - Essential building block for network security.
 - Alone cannot effectively prevent attacks.
 - Various types:
 - Packet filters.
 - Application-level gateways/proxies.
 - Example: Linux firewall (netfilter, iptables/nftables).
- Intrusion Detection and Prevention Systems (IDPS)**
 - Detect and prevent intrusion attempts.
 - Types:
 - Host-based vs. Network-based IDPS.

- Detection methodologies:
 - Signature-based.
 - Behavior-based (e.g., anomaly detection).
- Specialized IDPS for specific systems (e.g., industrial automation, vehicle networks).