

PA197 Secure Network Design

4. Security Architectures II

Eva Hladká, Luděk Matyska

Faculty of Informatics

March 8, 2023

Content

- 1 End2end principles
- 2 Secure channels
 - IPv4
 - IPv6
 - IPsec
- 3 Virtual private networks
- 4 Transport layer security
- 5 Authentication mechanisms
 - 802.1x and derivatives
- 6 Trusted network access
 - Basic authorization principles
 - Trusted Network Connect (TNC)

End2end principle

- Network transparency
 - legacy from the telco network
 - basic two-point communication
- Application architecture → end2end principle
 - the properties of the communication defined at its end points
 - network properties not accounted for
- Security implications
 - communication through channels
 - adding security to these channels
 - encryption at different layers
 - privacy threat: trail between source and destination within network

Secure channels

- Encryption of the communication between concrete layers of the network protocol
 - Explicit encryption by the application before pushing data to the transport stack
 - Secure transport layer: SSL/TLS
 - Secure internet layer: IPsec
- The last two transparent to the application
- Concept of Virtual Private Network (VPN)
 - applications sits on top of secure communication channels
 - mobility of one end-point
 - potential for multi-point communication

IPv4

- IPv4 was not build with the security in mind
 - conceived in times of pure academic (i.e. restricted) use
 - small number of nodes and small number of users
- All information exposed to any eavesdropper
 - destination and source address
 - type of the message (meta-info)
 - content of the message
 - unless explicitly encrypted before transmission
- Security through organizational and legal barriers
 - physical access to the network and attached computers
 - legal restrictions on eavesdropping old telecommunication lines

IPv6

- The security drawbacks of IPv4 recognized
- Full security incorporated as a mandatory requirement
- Integrity
 - not possible to modify the control data
 - source and destination addresses
 - type of the messages
- Content hidden from eavesdropper
 - content encryption
 - possibility to also encrypt most of the metadata
 - e.g. type of message

However, source and destination address always visible
- The principles transformed into separate protocol description:
IPsec
- The mandatory security for IPv6 was dropped in RFC 6434

IPsec

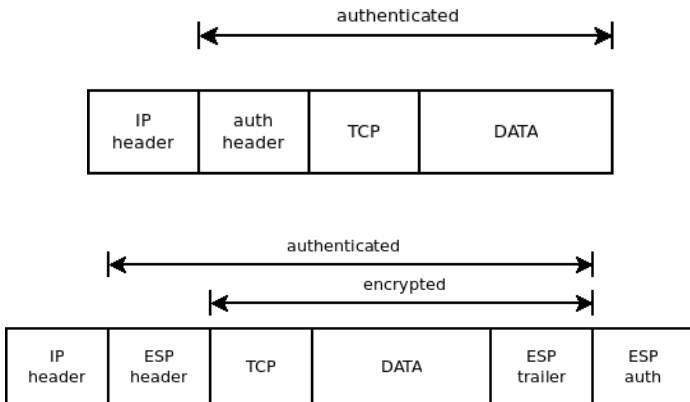
- IP security protocol suite
 - first developed for IPv6
 - backpropagated to IPv4 as **IPsec**
- Addresses security problems of the IP layer
 - Eavesdropping, hijacking, spoofing, . . .
- Implemented at the IP layer
- Provides specific protocols/mechanisms
 - confidentiality (no eavesdropping)
 - data origin authentication (no spoofing)
 - message integrity (no data modification)
 - access control
 - replay detection

IPsec Architecture

- Authentication header (AH)
 - data integrity
 - source authentication
- Encapsulating security protocol (ESP)
 - confidentiality (authentication just optional)
- Security association (SA)
 - one directional relationship between sender and receiver
 - establishment of security parameters
 - security association database (SADB)
 - security parameter index (SPI)
 - a unique index for each entry in the SADB
 - associates SA with a packet
 - security policy database (SPD)—“IP sec firewall”
- Transport mode
 - protection of higher-level protocols
- Tunnel mode

• protection at the IP layer

IPsec Diagrams



IPsec—Transport Mode

- End-to-end transmission
 - internal active elements not necessarily involved
- Original packet not encapsulated
 - IPsec specific header **inserted** between original IP and TCP/UDP headers

IPsec—Tunnel Mode

- Tunneling between active elements
 - needs support inside the network
 - at least edge routers must be involved
- Encapsulates original packet
 - **prepends** new IP header
 - identifies the source and destination addresses of the tunnel
 - IPsec header immediately follows the new (tunnel) IP header
 - the original packet is thus fully encapsulated
 - can be **fully encrypted**, including the original source/destination addresses

IPsec—Definitions

- A collection of protocols
 - basic: RFC 2401
- Authentication Header (AH)
 - RFC 2402
- Encapsulating Security Payload (ESP)
 - RFC 2406
- Internet Key Exchange (IKE)
 - RFC 2409
- IP Payload Compression (IPcomp)
 - RFC 3137

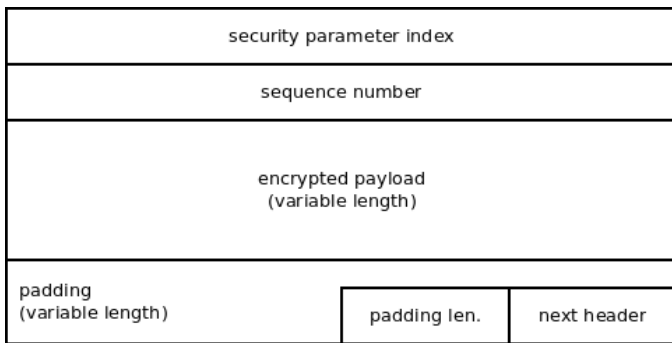
AH Details

- Protection against **replay** attacks
 - 32-bit monotonically increasing sequence numbers
- Protecting data **integrity**
 - cryptographically strong hash algorithms (96 bits)
 - symmetric key cryptography
 - HMAC-SHA-96, HMAC-MD5-96

next header	payload length	reserved
security parameter index		
sequence number		
authentication data (variable length)		

ESP Details

- On top of AH provides data confidentiality
 - symmetric key encryption to encrypt full packets



Internet Key Exchange

- Essential part of IPsec
 - however usable also outside IPsec
- Exchange and negotiate security policies
- Establish **Security Associations**
- Key exchange
- Key management

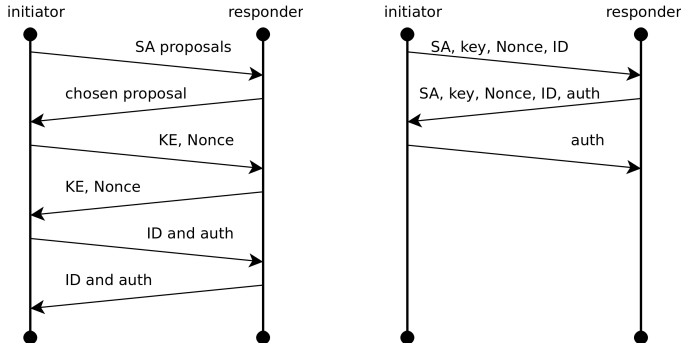
IKE Phases

- Phase 1: negotiate and establish an auxiliary e2e secure channel
 - medium for subsequent phase 2
 - only once between any two endpoints
- Phase 2: negotiate and establish custom secure channels
 - for each separate flow
 - occurs many times
- Both phases use Diffie-Hellman key exchange to establish a shared key

IKE Phase 1

- Secure channel between two end-points
- Security features
 - source authentication
 - data integrity and confidentiality
 - protection against replay attacks
- Purpose
 - to provide basic security environment
 - to support secure negotiations for the applications
 - different policies
 - different keys
- Two modes:
 - **main** mode
 - six messages in three round trips, more options
 - protects the identity of the peers
 - **aggressive** mode
 - three messages in two round trips, less options

Main vs Aggressive mode



Phase 1 Authentication

- Different ways
 - digital signature
 - two forms of authentication with public key
 - pre-shared key
- Uses public-key based cryptography for encryption

IKE Phase 2

- Custom secure channels
 - using the secure channel established in Phase 1 for setup
 - support heterogeneity
- End-point identification:
 - $\langle \text{IP, Port} \rangle$ (as in transport layer)
 - packet end address (network/range)
 - e.g. all packets for 147.251.11.0/24
- Only **quick** mode
 - multiplexes multiple quick mode exchanges
 - generates SAs for both end-points

Virtual Private Networks (VPN)

- A network that uses **public** infrastructure (e.g. Internet) to connect remote offices or users with secure access to their organization's internal network
 - it **extends** the internal organization's network to remote users in a secure way
- Through the VPN, users are able to use **internal** applications as if they are within the organization boundaries
- “Classical” VPNs work at Layer 3
 - point to point connection
 - no (limited) support to broadcast domains
- VPNs at Layer 2
 - layer 2 tunneling protocols
 - VPLS
 - extend organizational network also at broadcast domains

Security

- VPNs are (usually) not anonymous
 - some kind of authentication is mandatory
- VPNs are (usually) encrypted
 - protection against eavesdropper over public network
 - confidentiality is provided
- Message integrity is also provided

Basic protocols

- IPsec tunnels
 - standards-based security protocol
- TLS/SSL tunnels
 - used in OpenVPN project
 - can tunnel entire network or just a single user connection
 - alternative to IPsec in NATed and firewalled environment
- Secure Socket Tunneling Protocol (SSTP)
 - Microsoft
 - using SSL3 3.0 to tunnel Point-to-Point Protocol (PPP) or L2 Tunneling Protocol
 - Poodle attack sensitive
- Secure Shell (ssh) VPN
 - OpenSSH
 - VPN tunneling
 - do not confuse with port forwarding

Layer 2 VPN

- Virtual LAN
 - IEEE 802.1Q trunking protocol
 - packet tagging
 - single trunk/single LAN
- Virtual private LAN service (VPLS)
 - multiple tagged LANs share a common trunk
 - it is a provider provisioned VPN, not a private line
 - available for connecting two or more LANs over a public network at L2
 - all connected LANs behave as a single LAN from users' point of view
 - works with **frames**, not packets
- Ethernet over IP
 - EtherIP (RFC 3378)
 - only packet encapsulation
 - no confidentiality nor message integrity

Layer 3 VPN

- Provider-provisioned services discussed here
 - multiple customers
 - private IP address space disambiguation at the edge device
- BGP/MPLS
 - defined in RFC 2547
 - BGP extensions advertise IPv4 VPN address' routes
 - Route Distinguisher (8 bytes)
 - IPv4 address
 - providers edge routers "know" VPNs' topology
 - MPLS used to tunnel between these edge routers
- Virtual router
 - customer is responsible for the VPN's address space
 - no extensions to the routing
 - MPLS tunnels, different VPNs disambiguated by their label

Mobile VPN

- VPN for mobile devices (mVPN)
 - power (battery) sensitive
 - allows gaps in connections
- Roaming support
 - no single IP address assigned by the network to the mobile end-point
- Uses permanent IP address of the device
 - tunneling VPN
 - software layers take care of tunnels re-connection
 - the end-point IP visible to organization's network does not change

Transport Layer Security

- An attempt to guarantee a transport protocol to prevent **eavesdropping** and **tampering**
- A statefull connection
 - a handshake to establish connection security that leads to a secure (encrypted) communication channel
- Needs a reliable end-to-end communication channel (TCP)
- Predecessor is the **Secure Socket Layer (SSL)**
 - the last version 3 (1996, see RFC 6101)
 - insecure, vulnerable to the POODLE attack
- Evolution through **Transport Layer Security (TLS)** protocol
 - similar but not compatible with SSL
 - version 1.2 (RFC 5246 in 2008)
 - refined in 2011 (RFC 6167)
 - removed backward compatibility with SSL
 - version 1.3 in draft (October 2014)

TLS protocol

- **TLS session**
 - association between peers (client/server)
 - established by the **TLS handshake**
 - specifies cryptographic parameters
 - to work over expensive public-key cryptography
 - shared across several connections
- **TLS connection**
 - mechanisms to transport data
 - type of service
 - how data are sent/received
 - every connection is associated with one TLS session

Basic TLS Handshake

- Always one-way
 - server and client must authenticate independently
- Negotiation phase (server is authenticated)
 - client sends **ClientHello** message
 - highest TLS protocol it supports; random number; suggested cipher suites; suggested compression methods (not v1.3)
 - server responds with **ServerHello** handshake
 - chosen protocol version; random number; selected cipher suite and
 - includes also **session ID** compression method
 - client sends its **Certificate**
 - server sends its **ServerKeyExchange** and **ServerHelloDone**
 - client responds with **ClientKeyExchange**
 - could include **PreMasterSecret** key, encrypted with server public key
 - client and server now compute **master secret** (from PreMasterKey and random numbers)

TLS Handshake II

- Cipher confirmation
 - client sends **ChangeCipherSpec** record
 - client sends **Finished** messages containing hash and MAC over previous conversation
 - server checks Finished message and tears down the connection if check fails
 - server does the same towards client (with its own **ChangeCipherSpec**)
- Application phase
 - handshake is complete
 - all messages are authenticated and encrypted as the Finished message
- Optionally no encryption can be negotiated during the handshake
 - in such case no PreMasterSecret is exchanged and messages are not encrypted

Client-authenticated TLS handshake

- Adding client authentication to the negotiation phase
- Modifications
 - after ServerKeyExchange, server sends **CertificateRequest** to ask for client authentication
 - after ServerHelloDone, client responds with **Certificate** message with its own certificate
 - after ClientKeyExchange, clients sends **CertificateVerify**
 - signature over previous handshake messages
 - signed by client private key
 - server verifies the signature

Resumed TLS handshake

- Uses session ID sent by server during the original full handshake
 - client keeps a triple <session ID; server IP address; TCP port>
 - server keeps the session ID together with the cryptographic parameters negotiated (the master secret)
- Negotiation phase
 - client sends the **ClientHello**
 - it includes also the session ID from the previous handshake
 - server responds with **ServerHello**
 - send the same session ID if it recognizes it
 - a different session ID means new full handshake is requested
- Cipher confirmation
 - same as for the full handshake, using the previously stored master secret
- Much shorter, does not need public key cryptography (if the negotiated cipher suite does not need it)

802.1X Protocol

- IEEE standard for Port based Network Access Control
 - part of IEEE 802.1 group of standards
 - authentication framework
 - the actual algorithm how to do it

802.1X—Authentication framework

- Based on **Extensible Authentication Protocol** (EAP)
 - original RFC 2284 made obsolete by RFC 3748 updated in RFC 5247
- **EAP encapsulation over LAN** (EAPOL) protocol
 - Ethernet, including 802.11 wireless
 - token rings, including FDDI
- A **supplicant** request access to an access point (**authenticator**)
- AP allows only EAP message to be sent by supplicant
- Authenticator sends “EAP-Request/Identity”
- Client returns “EAP-Response/Identity” that is forwarded to the **authentication server**
 - it either accepts or rejects the authentication request
 - the decision is sent back to access point

it could (but is not required to) use the Remote Authentication Dial-In User Service (RADIUS)

LEAP and PEAP

- LEAP: Lightweight Extensible Authentication Protocol
 - CISCO-developed 802.1x derivative
 - targets CISCO Aironet equipment
- Uses TKIP and dynamic WEP keys
 - frequent WEP key alteration reduces risks in using this protocol
- PEAP: Protected EAP
 - developed by RSA, Microsoft, and CISCO
 - more advanced than LEAP
- Uses server-side PKI to create an encrypted EAP-TLS tunnels
- This tunnel is used to transmit user's credentials

PANA

- Protocol for Carrying Authentication for Network Access
 - IETF-backed
 - RFC 5191
- IP-based protocol
 - device authentication (to get access)
 - uses EAP
 - PANA carries EAP payload
 - no need for EAPOL or the likes

PANA—Elements

- **PANA Client (PaC)**
- **PANA Authentication Agent (PAA)**
 - message exchange with PaC for authentication and authorization
- **Authentication Server (AS)**
 - stores the info needed to check PaC credentials
 - affirmative reply could contain also some data what is allowed
 - bandwidth parameters
 - IP configuration etc.
 - always time constrained (session time)
- **Enforcement Point (EP)**
 - filters data from PaC according to the policy
 - a key is established between PaC and EP
 - valid during the session time only

Basic authorization principles

- **Least privilege**
 - default is **no access**
 - all privileges must be explicitly defined/assigned
- **Separation of duties/privileges**
 - no combination of responsibilities in one person/entity
- **Need to know**
 - access only to the information (infrastructure) needed to perform the work
- **Complete mediation**
 - All accesses must be checked

Access Control

- The criteria used to decide on access usually include one or more from the following:
 - roles
 - groups
 - location
 - time
 - type of access

Trusted Network Connect (TNC)

- An activity to define an open solution architecture for access control to the network endpoints
 - TNC-Working Group: companies, government, academia
 - first introduced in 2005
- TNC reference architecture
 - federated TNC protocol (IF-FTNC) which enables communication of
 - IF-M attributes
 - IF TNCCS Access recommendations
 - IF-MAP metadata from one security domain to another
- To support network administrators in protecting networks
 - impose enterprise security policies
 - audit endpoint configurations

TNC Key Elements

- **Network Access Requester (NAR)**
 - a client software on endpoint that initiates the network access attempt
 - VPN client, 802.1x supplicants, web browser with initiating TLS handshake etc.
- **Policy Enforcement Point (PEP)**
 - network infrastructure device
 - 802.1x compliant
 - forwards information about NAR to PDP
- **Policy Decision Point (PDP)**
 - a device that hosts NEA
- **Network Access Authority (NEA)**
 - determines the fate of the NAR request

Summary

- Two major concepts
 - secure end-to-end communication
 - access control to the network
- Different ways for secure channels
 - IPsec in details (including IKE)
 - animation of IPsec functionality:
<http://frakira.fi.muni.cz/~jeronimo/vyuka/IPSec>
- VPNs
 - for end users and between sites
 - L2 and L3 protocols
- 802.1x protocol for authentication
- Access control
 - Basic authorization principles
 - Trusted Network Connection
- Next session: Advanced architectures