Real-Time Communication

Security: SSL, IPSEC

Ahmet Burak Can
Hacettepe University
abc@hacettepe.edu.tr

The Internet

A packet-switched network:
- Data to be transmitted is divided into “packets”
- Each packet is forwarded by “routers” towards the destination

TCP/IP Reference Model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer (HTTP, FTP, SMTP, etc.)</td>
<td></td>
</tr>
<tr>
<td>Transport Layer (TCP, UDP)</td>
<td></td>
</tr>
<tr>
<td>Network Layer (IP)</td>
<td></td>
</tr>
<tr>
<td>Data Link Layer (PPP, Ethernet, etc.)</td>
<td></td>
</tr>
<tr>
<td>Physical Layer</td>
<td></td>
</tr>
</tbody>
</table>

- IP: delivery of packets to the destination
- TCP: reliability of the communication
  - ordering the packets
  - error detection & recovery
  - congestion control
- UDP: basic transport protocol

Securing TCP/IP Communications

Layer 3 (IPsec):
- can secure all IP communication transparent to applications
- must be built into the OS

Layer 4:
- doesn’t require OS modification; deployment easy

Different Security Models in TCP/IP

Real-Time Protocol Security Issues

- Interactive session security (unlike e-mail)
- Layer 4 (SSL)
  - Implemented on top of layer 4, between TCP & application
  - Doesn’t require any modifications to OS (deployment made easy!)
- Layer 3 (IPsec)
  - Implemented between IP & TCP
  - Each IP packet authenticated separately
  - Built in the OS
  - Can secure all IP communication
  - Host-to-host application is common.
  - Process-to-process also possible
Perfect Forward Secrecy
- PFS: Compromise of long-term secrets doesn’t compromise session keys
- Example: Diffie-Hellman with RSA authentication
- Non-PFS examples:
  - Kerberos
  - Session key transport with RSA encryption
- By-product: Escrow foilage
  Conversations can’t be decrypted by authorities holding copies of long-term private keys

A PFS Protocol Example: Diffie-Hellman with RSA signature

SSL/TLS
- SSLv2
  - Released in 1995 with Netscape 1.1
  - Key generation algorithm kept secret
  - Reverse engineered & broken by Wagner & Goldberg
- SSLv3
  - Fixed and improved, released in 1996
  - Public design process
- PCT: Microsoft’s version of SSL
- TLS: IETF’s version

SSL Architecture

Handshake Protocol
- Negotiate Cipher-Suite Algorithms
  - Symmetric cipher to use
  - Key exchange method
  - Message digest function
- Establish the shared master secret
- Optionally authenticate server and/or client
**Basic SSL/TLS Handshake Protocol**

- Hello, crypto offered, $R_a$
- Certificate, crypto selected, $R_b$
- $\{S\}_K$ [keyed hash of messages]
- $(K = f(S, R_a, R_b))$
- [keyed hash of messages]
- Session keys derived from $K$

**Key Computation**

- “pre-master key”: $S$
- “master key”: $K = f(S, R_a, R_b)$
- For each connection, 6 keys are generated from $K$ and the nonces. (3 keys for each direction: encryption, authentication/integrity, IV)

**Session and Connection**

- **Session:**
  - association between a client and a server;
  - created by the Handshake Protocol;
  - defines secure cryptographic parameters that can be shared by multiple connections.
- **Connection:**
  - end-to-end reliable secure communication;
  - every connection is associated with a session.

**SSL Session Establishment**

- Client authentication: Bob can optionally send “certificate request” in message 2.
- Session vs. Connection: “Sessions” are relatively long-lived. Multiple “connections” (TCP) can be supported under the same SSL session. (designed for HTTP 1.0)
- To start a connection, Alice can send an existing session ID.
- If Bob doesn’t remember the session ID Alice sent, he responds with a different value.

**Negotiating Crypto Suites**

- Crypto suite: A complete package specifying the crypto to be used. (encryption algorithm, key length, integrity algorithm, etc.)
  - ~30 predefined standard cipher suites.
  - **Confidentiality**: Achieved by encryption using DES, 3DES, RC2, RC4, IDEA.
  - **Integrity**: Achieved by computing a MAC and send it with the message: MD5, SHA1.
  - **Key exchange**: relies on public key encryption.
- Selection:
  - v2: Alice proposes a set of suites; Bob returns a subset of them; Alice selects one. (which doesn’t make much sense)
  - v3: Alice proposes a set of suites; Bob selects one.

**SSL Record Protocol**

- Provides confidentiality and message integrity using shared keys established by the Handshake Protocol
IPsec

- Cryptographic protection of the IP traffic, transparent to the user
- Main components:
  - Internet Key Exchange (IKE): IPsec key exchange protocol
  - Authentication Header (AH): Authentication of the IP packet
  - Encapsulating Security Payload (ESP): Encryption/authentication of the IP packet

Uses of IPsec

- Can be used to provide user-, host-, or network-level protection (the granularity)
- Protocol modes:
  - Transport mode: Host applies IPsec to transport layer packet
  - Tunnel mode: Gateway applies IPsec to the IP packet of a host from the network (IP in IP tunnel)
- Typical uses:
  - Remote access to network (host-to-gateway)
  - Virtual private networks (gateway-to-gateway)

Security Association & Policy

- Security Policy Database
  - Specifies what kind of protection should be applied to packets (according to source-destination address, port numbers, User ID, data sensitivity level, etc.)

- Security Association (SA)
  - An IPsec-protected connection (one-way)
  - Specifies the encryption/auth. algorithm, key, etc.
  - Identified by
    - Security parameter index (SPI)
    - DESTINATION IP address
    - Protocol identifier (AH or ESP)
  - SAs are stored in SA databases
  - AH information (auth. algorithm, key, key lifetime, etc.)
  - ESP information (auth/encryption algorithm, key, key lifetime, etc.)
  - Lifetime of the SA

IPsec Packet Processing

Outbound packets:

- The proper SA is chosen from the security policy database
- From the SA database, the SPI and SA parameters are retrieved
- The IPsec protection is performed; packet passed to IP

Inbound packets:

- By the SPI, the SA is found
- IPsec auth/decryption is performed
- Packet passed to upper layer protocol

History of IKE

- Early contenders:
  - Photuris: Authenticated DH with cookies & identity hiding
  - SKIP: Authenticated DH with long-term exponents
- ISAKMP:
  - A protocol specifying only payload formats & exchanges (i.e., an empty protocol)
  - Adopted by the IPsec working group
- Oakley: Modified Photuris; can work with ISAKMP
- IKE: A particular Oakley-ISAKMP combination

Authentication Header (AH)

- Auth. alg.: HMAC (with MD5, SHA1, etc.)
  - CBC-MAC (3DES, RC5, AES, etc.)
- Typically, IV is included in the payload
- Authentication covers immutable fields of IP header as well as the payload.
AH with IPv4

BEFORE APPLYING AH

IPv4 | (any options) | AH | TCP | Data |

AFTER APPLYING AH

IPv4 | orig IP hdr | (any options) | AH | TCP | Data |

| -------- authenticated -------- |

except for mutable fields

Encapsulating Security Payload (ESP)

IPv4 | orig IP hdr | ESP | ESP | ESP |

| (any options) | Header | TCP | Data | Trailer | Auth |

| | Payload | (0-255 bytes) |

| Authentication data (variable) |

| Encryption: usually a block cipher in CBC mode |

| IV is typically included in the payload (not encrypted) |

ESP with IPv4

BEFORE APPLYING ESP

IPv4 | orig IP hdr | (any options) | TCP | Data |

AFTER APPLYING ESP

IPv4 | orig IP hdr | ESP | ESP | ESP | ESP |

| (any options) | Header | TCP | Data | Trailer | Auth |

| -------- encrypted -------- |

| -------- authenticated -------- |