IPSec & Secure Email

Problem – one server emulating many hosts

Q. How does a server hosting many different pages differentiate between them in http?

A. Host headers. Host headers have the host name inside of them. Http GET (host www.google.com). Server uses this to respond.

How does this work in https? SSL/TLS is broken

<table>
<thead>
<tr>
<th>Browser</th>
<th>Server (Cert 1, Cert 2, etc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL Handshake</td>
<td>------------------------</td>
</tr>
</tbody>
</table>

The SSL handshake requires a cert, but the hostname isn't given until HTTP, so the server doesn't know which cert to send back. This is solved by having different ip addresses on the same computer. Appropriate IP addresses are handed out at dns time.

SSL/TLS Stack:

1. http – browser
2. SSL/TLS – browser
3. TCP – OS
4. IP – OS
5. MAC – OS
6. Physical Layer (line card)

IPSec Stack:

1. http – browser
2. TCP – OS
3. IPSec – OS
4. IP – OS
5. MAC – OS
6. Physical Layer (line card)

VPN (Virtual Private Network)

Allows computers to send packets as if on a local area network.

VPNs can be based on IPSec or SSL/TLS

IPSec requires a change to the operating system, whereas SSL/TLS requires applications to have
SSL/TLS support.

SSL/TLS packet: IP Header (src IP, dst IP, TTL, Protocol:TCP, TCP seqnum), Payload
IPSec packet: IP Header (src IP, dst ip, TTL), IPSec (encrypted header), Encrypted Payload

Because of the TCP SeqNum, SSL/TLS is vulnerable to “Rogue Packet DoS”. If an attacker imitates a TCP packet with the next SeqNum, the SSL/TLS layer will throw away the byte-sequence, of which the rogue packet is a part, because it isn't encrypted properly and so will fail integrity checks, however when the real packet arrives TCP will throw it away since it had already processed a packet with the same sequence number.

This wouldn't work in IPSec, because the rogue packet would fail integrity checks in the IPSec layer and so would get thrown away at that layer, and never make it to the TCP layer, and so wouldn't affect valid conversations.

This attack is rather rare due to it being complicated, and not maliciously harmful (it only is a DoS, it doesn't get any private information, or change anything).

<table>
<thead>
<tr>
<th><strong>IPSec vs. SSL/TLS</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Below TCP</td>
<td>Above TCP</td>
</tr>
<tr>
<td>Requires OS Modification</td>
<td>Only App modification</td>
</tr>
<tr>
<td>Checks integrity on per-packet basis</td>
<td>Vulnerable to rogue packet DoS</td>
</tr>
<tr>
<td>Current API broken – only passes applications IP information, not user information.</td>
<td>App Level auth certs/login</td>
</tr>
<tr>
<td>Cheap outboard device</td>
<td>Expensive outboard device</td>
</tr>
</tbody>
</table>

IPSec -> IKE (Internet key exchange) – used for handshaking

Advantages of IPSec:
- IKE supports a much richer suite
- IPSec supports puzzles

Modes:

Transport mode (end to end mode), used by road warriors.

    [ IP header ] [ IPSec header ] [ Encrypted payload]

Tunnel mode (VPNs between branch offices).

    [ new IP header ] [ IPSec header ] [ Encapsulation: ( IP header | Payload ) ]
Secure Email

~ 80 billion emails a day. 60% of which is spam

Mechanics:

1. Person composes on some client & hits send
2. Email queued up at some outgoing server. Mail server looks up dns & gets MX record
3. Contacts that IP address on port 25 using SMTP. On failure, it retries. If rejected, bounces back.

MTA (Mail Transfer Agents) Relays: (Open Relays)

- provides fault tolerance
- provide anonymity
- provide translation services
- provide trust chains

Mail lists:

Local explosion: sender contacts mailserver, gets list & sends to each one on the list.

Remote explosion: sender sends mail to mail server, and then mail server sends to everyone in the list.

Advantages:

<table>
<thead>
<tr>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less load on server</td>
<td>Can have private/anonymous emails</td>
</tr>
<tr>
<td>Don't need to rely on server to do right thing</td>
<td>Maintains fresh list</td>
</tr>
<tr>
<td>Can avoid cyclic lists</td>
<td>Saves bandwidth</td>
</tr>
</tbody>
</table>

Secure Email:

A is sending to B

{ } - encryption
[ ] = signing

\{M [M]_A\}_B (public key way)

Efficient way:

\{S\}_B [h(M)]_A \{M\}_S

But this can be used (against A) by B in a court of law.
Puzzle: how can A get plausible deniability?

Consider \( \{S, [S]_A\} \{M\}_S \), B has the secret key S, and could insert his own message and encrypt it using S, thus giving A plausible deniability.

**Models of key distribution:**

<table>
<thead>
<tr>
<th></th>
<th>PGP (Pretty good privacy)</th>
<th>SMIME (Secure Multimedia)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anarchy model. Keys/roots installed by user</td>
<td>Web model</td>
<td></td>
</tr>
<tr>
<td>Trust chaining</td>
<td>CA (Verisign)'s key embedded in client &amp; everyone uses certs</td>
<td></td>
</tr>
</tbody>
</table>