HTTPS and the Lock Icon

Dan Boneh
Goals for this lecture

• Brief overview of HTTPS:
  • How the SSL/TLS protocol works (very briefly)
  • How to use HTTPS

• Integrating HTTPS into the browser
  • Lots of user interface problems to watch for
Threat Model: Network Attacker

Network Attacker:

- Controls network infrastructure: Routers, DNS
  - Passive attacker: only eavesdrops on net traffic
  - Active attacker: eavesdrops, injects, blocks, and modifies packets

Examples:

- Wireless network at Internet Café
- Internet access at hotels (untrusted ISP)
SSL/TLS overview

Public-key encryption:

Bob generates \((SK_{Bob}, PK_{Bob})\)

Alice: using \(PK_{Bob}\) encrypts messages

and only Bob can decrypt
How does Alice (browser) obtain $\text{PK}_{\text{Bob}}$?

Bob uses Cert for an extended period (e.g. one year)
Certificates: example

Important fields:

Certificate Signature Algorithm
Issuer
• Validity
  • Not Before
  • Not After
Subject
• Subject Public Key Info
  • Subject Public Key Algorithm
    • Subject's Public Key
• Extensions

Field Value

Certificate

Modulus (1024 bits):
ac731497b410a3aa41c15edcf92f39a
97269acf1b1e41bdcd2c92f7dd2e6071d
ad2b23ef78c2faa1b79ee35440343fb9
e2128a306b0cfaf306a0161e97cb198
d20dc63803b455337f104045c5c3e4d6
6b9cdf0dd08e4f390d2bd2e988cb2d21a3
f184613caaa8c1827e67ef7b86a0a75
e1bba1472950b6478068481eb7b078d49

This certificate has been verified for the following uses:
SSL Server Certificate

Issued To
Common Name (CN): *.gmail.com
Organization (O): Google Inc
Organizational Unit (OU): <Not Part Of Certificate>

Issued By
Common Name (CN): Thawte Premium Server CA
Organization (O): Thawte Consulting cc
Organizational Unit (OU): Certification Services Division

Validity
Issued On: 9/25/2008
Expires On: 9/25/2010

Fingerprints
Certificates on the web

Subject’s CommonName can be:

- An explicit name, e.g. \textit{cs.stanford.edu}, or
- A name with a wildcard character, e.g. \textit{*.stanford.edu} or \textit{cs*.stanford.edu}

matching rules:

IE7: “*” must occur in leftmost component, does not match “.”
    example: \textit{*.a.com} matches \textit{x.a.com} but not \textit{y.x.a.com}

FF3: “*” matches anything
Certificate Authorities

Browsers accept certificates from a large number of CAs.
Brief overview of SSL/TLS

Browser

- client-hello
- server-hello + server-cert (PK)

Server

- key exchange (several options)
  - client-key-exchange: $E(PK, k)$
- $k$

HTTP data encrypted with KDF(k)

Finished

Most common: server authentication only
Integrating SSL/TLS with HTTP ⇒ HTTPS

Two complications

• **Web proxies**
  solution: browser sends
  CONNECT domain-name
  before client-hello (dropped by proxy)

• **Virtual hosting:**
  two sites hosted at same IP address.
  solution in TLS 1.1 (RFC 4366)
  client_hello_extension: server_name=cnn.com
  implemented in FF2 and IE7 (vista)
Why is HTTPS not used for all web traffic?

- Slows down web servers
- Breaks Internet caching
  - ISPs cannot cache HTTPS traffic
  - Results in increased traffic at web site
- Incompatible with virtual hosting (older browsers)
HTTPS in the Browser
The lock icon: SSL indicator

**Intended goal:**
- Provide user with identity of page origin
- Indicate to user that page contents were not viewed or modified by a *network attacker*

**In reality:**
- Origin ID is not always helpful
  - example: Stanford HR is hosted at BenefitsCenter.com
- Many other problems (next few slides)
When is the (basic) lock icon displayed

• All elements on the page fetched using HTTPS (with some exceptions)

• For all elements:
  • HTTPS cert issued by a CA trusted by browser
  • HTTPS cert is valid (e.g. not expired)
  • CommonName in cert matches domain in URL
The lock UI: help users authenticate site
The lock UI: help users authenticate site

Firefox 3: (no SSL)

(SSL)
The lock UI: help users authenticate site

Firefox 3: clicking on bottom lock icon gives

<table>
<thead>
<tr>
<th>Web Site Identity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Web site: crypto.stanford.edu</td>
<td></td>
</tr>
<tr>
<td>Owner: This web site does not supply identity information.</td>
<td></td>
</tr>
<tr>
<td>Verified by: Comodo CA Limited</td>
<td></td>
</tr>
</tbody>
</table>

This web site provides a certificate to verify its identity.

<table>
<thead>
<tr>
<th>Privacy &amp; History</th>
<th>Yes, 8 times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have I visited this web site before today?</td>
<td>Yes, 8 times</td>
</tr>
<tr>
<td>Is this web site storing information (cookies) on my computer?</td>
<td>Yes</td>
</tr>
<tr>
<td>Have I saved any passwords for this web site?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technical Details</th>
<th>Connection Encrypted: High-grade Encryption (AES-256 256 bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The page you are viewing was encrypted before being transmitted over the Internet.</td>
<td></td>
</tr>
<tr>
<td>Encryption makes it very difficult for unauthorized people to view information traveling between computers. It is therefore very unlikely that anyone read this page as it traveled across the network.</td>
<td></td>
</tr>
</tbody>
</table>
The lock UI: Extended Validation (EV) Certs

- Harder to obtain than regular certs
  - requires human lawyer at CA to approve cert request

- Designed for banks and large e-commerce sites

- Helps block “semantic attacks”: www.bankofthevvest.com
A general UI attack: picture-in-picture

Trained users are more likely to fall victim to this [JSTB'07]
HTTPS and login pages: incorrect version

Users often land on login page over HTTP:

- Type site’s HTTP URL into address bar, or
- Google links to the HTTP page

View source:

```html
<form method="post" action="https://onlineservices.wachovia.com/...">
```

`https://onlineservices.wachovia.com/...`
HTTPS and login pages: guidelines

General guideline:

• Response to http://login.site.com should be Redirect: https://login.site.com
Problems with HTTPS and the Lock Icon
Problems with HTTPS and the Lock Icon

1. Upgrade from HTTP to HTTPS

2. Semantic attacks on certs

3. Invalid certs

4. Mixed content
   - HTTP and HTTPS on the same page

5. Origin contamination
   - Weak HTTPS page contaminates stronger HTTPS page
1. HTTP → HTTPS upgrade

Common use pattern:
- browse site over HTTP; move to HTTPS for checkout
- connect to bank over HTTP; move to HTTPS for login

Easy attack: prevent the upgrade (ssl_strip) [Moxie’08]

```
<a href=https://…> ⇒ <a href=http://…>
Location: https://… ⇒ Location: http://… (redirect)
<form action=https://… > ⇒ <form action=http://…>
```
Tricks and Details

**Tricks:** drop-in a clever fav icon

![Image of Firefox bookmarks](image)

**Details:**
- Erase existing session and force user to login:
  
  `ssl_strip` injects “Set-cookie” headers to delete existing session cookies in browser.

Number of users who detected HTTP downgrade: 0
2. Semantic attacks on certs

International domains: xyz.cn
• Rendered using international character set
• Observation: chinese character set contains chars that look like “/” and “?” and “.” and “=”

Attack: buy domain cert for *.badguy.cn
setup domain called:
www.bank.com/accounts/login.php?q=me.badguy.cn

note: single cert *.badguy.cn works for all sites

Extended validation (EV) certs may help defeat this
3. Invalid certs

Examples of invalid certificates:

• expired: current-date > date-in-cert
• CommonName in cert does not match domain in URL
• unknown CA (e.g. self signed certs)
  • Small sites may not want to pay for cert

Users often ignore warning:

Is it a misconfiguration or an attack? User can’t tell.

Accepting invalid cert enables man-in-middle attacks
(see http://crypto.stanford.edu/ssl-mitm)
Man in the middle attack using invalid certs

GET https://bank.com

bad cert warning!

SSL key exchange

k₁

HTTP data enc with k₁

Attacker proxies data between user and bank.
Sees all traffic and can modify data at will.
Firefox: Invalid cert dialog

Firefox 3.0: Four clicks to get firefox to accept cert
- page is displayed with full HTTPS indicators
IE: invalid cert URL bar
PacketForensis: SSL MiTM for law enforcement

Scenario [SS’10]: (browsers on windows trust 264 root CAs)
- User in country X wishes to access web site in country Y
- Country X compels its local CA to issue cert for web site

→ country X can eavesdrop on all traffic to web site
   (no cert warning in user’s browser)

SS’10 solution: browser extension that rejects certs where
issuing country ≠ web-site country
4. Mixed Content: HTTP and HTTPS

Page loads over HTTPS, but contains content over HTTP
(e.g.  <script src="http://.../script.js>  )

IE7: displays mixed-content dialog and no SSL lock
Firefox 3.0: displays `!` over lock icon (no dialog by default)

Both browsers:
• Flash swf file over HTTP does not trigger warning !!
• note: Flash can script the embedding page

Safari: does not attempt to detect mixed content
Mixed Content: HTTP and HTTPS

IE7: No SSL lock in address bar:

silly dialogs
Mixed Content: HTTP and HTTPS

Firefox 3.0:

- No SSL indicator in address bar
- Clicking on bottom lock gives:

![Technical Details]

**Connection Partially Encrypted**
Parts of the page you are viewing were not encrypted before being transmitted over the Internet.
Information sent over the Internet without encryption can be seen by other people while it is in transit.
Mixed content and network attacks

banks: after login all content served over HTTPS

Developer error: somewhere on bank site write

```
<embed src="http://www.site.com/flash.swf">
```

Active network attacker can now hijack session

Better way to include content:

```
<embed src="/www.site.com/flash.swf">
```

served over the same protocol as embedding page
An Example From an Online Bank

network attacker can modify SWF file and hijack session

(the site has been fixed)
5. Origin Contamination: an example

**safeLock:** removes lock from top page after loading bottom page
Final note: the status Bar

Trivially spoofable

<a href="http://www.paypal.com/">
  onclick="this.href = ‘http://www.evil.com/’;”>
  PayPal
</a>
THE END