SSL/TLS Vulnerabilities

BİL 448/548
Internet Security Protocols
Ali Aydın Selçuk

SSL/TLS

- The main workhorse of secure Internet communication.
- Everyday, billions of web packets, and much more, are encrypted by SSL/TLS.
- There is a viable commercial model for the trust infrastructure behind it.

Success of SSL

- Trust infrastructure has an autonomous and self-governing structure, consisting of
  - browser vendors
  - audit firms and standards bodies
  - certificate authorities
  - SSL servers
- Has been remarkably successful, especially compared to previous efforts such as PEM to secure Internet communications.

Certification Authorities

- CAs' public key should be distributed in a trusted way to all the parties in the system in advance.
- In SSL, root CAs are approved by the browser (or the OS) makers, and distributed with the browser/OS code.
- CAs must satisfy certain criteria for this:
  - https://wiki.mozilla.org/CA:How_to_apply
  - http://www.chromium.org/Home/chromium-security/root-ca-policy
  - http://technet.microsoft.com/en-us/library/cc751157.aspx

Certification Authorities

- Browser makers require CA firms to be audited and accredited according to some standards:
  - WebTrust
  - ETSI TS 101/102
  - ISO 21188:2006
- Public key infrastructure of SSL:
  - Oligarchy model: A number of trusted root CAs,
  - which issue certificates to intermediate CAs, or to end users (SSL servers)

Certificates & Validation

- Valid SSL/TLS certificates are issued to web servers by root or intermediate CAs.
  - E.g., Google's certificate: GeoTrust (root) → Google Internet Authority → accounts.google.com
- Client (browser) authenticates this chain of certificates beginning from the root CA.

http://en.wikipedia.org/wiki/Certification_path_validation_algorithm

SSL/TLS in Practice

SSL/TLS:

- A reasonably secure protocol
- with a reasonable trust model
- and commercially viable operation

What may go wrong?

- "Man in the browser" attacks
- Cert. validation software may get it wrong
- Compromised CAs, fake certificates
- and more…

MitB Attacks

- "Man in the browser"
- Trojan is used to manipulate calls between the browser and its security mechanisms & libraries.
- Utilizes facilities provided to enhance browsers capabilities:
  - browser extensions, user scripts, etc.
- SSL is useless in this context.
- Attacks mostly target financial transactions.
- Out-of-band transaction verification can be used for protection (e.g., an SMS with detailed info).

MitM by the Browser

- Many mobile browsers use remote rendering of webpages for performance (caching, compression, etc.)
- Opera Mini, Kindle Fire Silk, Nokia browser…
- HTTPS traffic is routed through a “trusted” proxy, which decrypts the pages and then does rendering, caching, compression, etc.
- “Trust us, we're not looking at your data.”
- Usually considered ok if not done secretly.

MitM by Corporation

- Many corporations install their computers with a trusted root key, and route the traffic through a proxy.
- Data is monitored to make sure that no sensitive info is leaked, no porn is surfed, etc.
- Users’ “secure” connection to their bank's website, etc. is fully readable by the company's IT department.
- Care must be taken to prevent any leakage of cached data, logs, etc.

Certificate Validation Errors

Certificate validation at the browser may not be as easy as it seems. For instance:

- Erroneous string comparisons
- Not fully inspecting the certificates
- Disregarding the warning flags

Certificate Validation Problem – 1

Null Prefix Attacks:

- Subject names containing the NULL character are allowed in ASN.1 strings. E.g. certificate for www.paypal.com\0.hackersrus.com can be issued to hackersrus.com.
- C string comparison libraries process a string till the NULL character! E.g., paypal.com\0.hackersrus.com == paypal.com
- https://www.blackhat.com/presentations/bh-usa-09/MARLINSPIKE/BHUSA09-Marlinspike-DefeatSSL-PAPER1.pdf

Certificate Validation Problem – 2

Non-verification of certificate constraints:

- Client (browser) software may fail to check the "Basic Constraints" and "Key Usage" fields in a certificate.
- In that case, any leaf certificate holder can act like a CA!
  - http://www.thoughtcrime.org/ie-ssl-chain.txt
  - https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf

Mishandling of Warning Flags

- Some certificate validation errors are signaled through warning flags rather than errors. E.g.,
  - certificate expired
  - name mismatch (e.g., m.xyz.com vs. www.xyz.com)
  - certificate issued by an unknown CA (useful for self-signed certificates)
- Browsers display warning messages to the user.
- But what do non-interactive SSL software do?
  - payment gateway SDK
  - mobile apps
  - cloud client API
  - …

Non-Interactive SSL/TLS Software

- Many non-interactive SSL clients just disregard the warning flags!
- SSL certificate validation is completely broken in many security-critical applications and libraries.
  - https://crypto.stanford.edu/~dabo/pubs/abstracts/ssl-client-bugs.html
  - Certificates issued to completely different names are accepted.
  - Certificates issued by completely unknown root CAs (by anybody!) are accepted.

Problems in the Trust Chain

- Compromised CAs issuing fraudulent certificates
- Uncompromised CAs issuing fraudulent certificates (by mistake or otherwise)

Compromised CAs

- DigiNotar, a Dutch CA company, was hacked by Iranian hackers in July 2011.
- Fraudulent certificates were observed for Google services in Iran, August 2011.
- DigiNotar was removed from the list of trusted CAs in browsers, August-September 2011.
- DigiNotar went bankrupt, September 2011.

A relatively easy problem to handle.

Uncompromised CAs

"Uncompromised" CAs issuing fraudulent certificates (by mistake or otherwise)

- Comodo, 2011 (auxiliary RA hacked?)
- Trustwave, 2011 (sub-CA cert. sold to customer!)
- Turktrust, 2011-2012 (sub-CA cert. issued by mistake?)
- and more…
- targeting google.com, yahoo.com, skype.com…

Unlike DigiNotar, almost nothing happened to any of these CAs.

Proposed Solutions

- Using DNSSEC for domain name authentication ("DANE")
- Pinning certificates
- Distributing trust, avoiding CAs:
  - "Trust agility"
  - Perspectives (CMU)
  - Convergence (Moxie)
- And more…
- A very active area of research

Problems – Certificate Revocation

- Discovered fraudulent certificates are added to certificate revocation lists (CRLs).
- These can be queried by the Online Certificate Status Protocol (OCSP).
- Not good enough: MitM can easily disable OCSP.
- Response message 3: "Try again later"
  - https://www.blackhat.com/presentations/bh-usa-09/MARLINSPIKE/BHUSA09-Marlinspike-DefeatOCSP-PAPER2.pdf
- Chrome's way: Don't use OCSP; update the CRL-set in the browser periodically.

Stripping of SSL

- Better alternative for the MitM attacker, with no fingerprints (i.e., fake certificates) left: Change HTTPS connection to HTTP!
- Hardly anybody notices.
  - https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf
  - http://www.thoughtcrime.org/software/sslstrip/
- Proposed solution: HSTS
  - http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security

Conclusions (before Snowden)

- Although SSL/TLS is a reasonably secure protocol on paper, there are many things that may go wrong in practice.
  - Malware infections may render SSL useless.
  - Browser makers and IT departments must be trusted with the certificates installed.
  - Buggy software may fail to do the checks properly.
  - Trust chain may be broken due to different reasons.
  - Protocols can be downgraded to insecure alternatives, without anybody noticing.
- Caution is needed for a secure use of SSL/TLS.

SSL after Snowden

- Who is Edward Snowden?
- What happened? What is going on?

Brief Bio of Edward Snowden

- Born in 1983. Didn't receive much formal education. Dropped out of high school. (interesting?)
- Studied computers. Became a self-taught "computer wizard".
- First worked at NSA as a security guard (2006), then at CIA as an IT security specialist (2007), and later at private contractors (Dell, Booz Allen) for NSA as an "infrastructure analyst".

Bio of Edward Snowden (cont.)

- Snowden grew increasingly uncomfortable with what he saw at NSA; in particular unlawful surveillance of US citizens, and more.
- He said he believes in the Nuremberg principles,

"Individuals have international duties which transcend the national obligations of obedience. Therefore individual citizens have the duty to violate domestic laws to prevent crimes against peace and humanity from occurring."

Snowden Incident (2013)

- Working as a contractor, he compiled a large store of top-secret NSA documents.
- While stationed in Hawaii, he took a leave of absence for health reasons; first flew to Hong Kong and then to Moscow.
- He passed the documents to journalist Glenn Greenwald. They are going over them and publishing selectively (still observing the US and UK national security).

Snowden Revelations

- Mostly about post-9/11 excesses of NSA under the G.W.Bush administration, but still continuing.
  - bulk collection of data on US citizens
  - spying on foreign leaders (inc. friendly nations), UN, journalists, etc.
  - infiltrating the global telecommunications industry
  - …
- Stuff that is somewhat expectable if not acceptable.

Snowden Revelations (cont.)

Most shocking (for us, IT security people):

- Working to deliberately weaken international cryptographic standards (over NIST)
- Working with h/w and s/w vendors to weaken encryption and random number generators
- Decrypting encrypted Internet traffic (SSL) somehow!
- "For the past decade, NSA has led an aggressive effort to break widely used Internet encryption technologies. Cryptanalytic capabilities are now coming online. Vast amounts of encrypted Internet data are now exploitable." NSA briefing to GCHQ, 2010

Some Guesses on Breaking SSL

How can that happen?

- Some fundamental cryptosystem is broken (RSA-1024?)
- Using fake certificates issued by cooperating CAs (possible but not very suitable; subject to detection)
- Obtaining/stealing private keys from large SSL servers (gmail, yahoo, etc.?) (possible, but doesn't help to break other SSL connections)
- Weakening RNGs by working with vendors (MS, Intel, etc.?)
- Weakening open-source libraries (OpenSSL, etc.?) (doesn't look easy, but definitely possible)

What can be Done?

- Widespread use of encryption all over the Internet
  - IETF-88 (Vancouver, November 2013)
  - This will make things much harder for the NSA, although not impossible.
- Preferring open-source alternatives
  - They can be compromised too; but that is not as easy as the closed-source alternatives.
- Hiding your communications
  - Using anonymization services like Tor makes surveillance much harder.

Conclusion

- SSL/TLS is a reasonably well-designed, reasonably secure protocol, with a quite successful operation (until recently).
- If used properly, it is mostly secure against ordinary adversaries.
- Against the biggest brother, we don't really know.