NDSS 2016 Presentation, Feb. 22, 2016

Killed by Proxy: Analyzing Client-end TLS Interception Software

Xavier de Carné de Carnavalet Mohammad Mannan

Madiba Security Research Group at Concordia University, Montreal, Canada

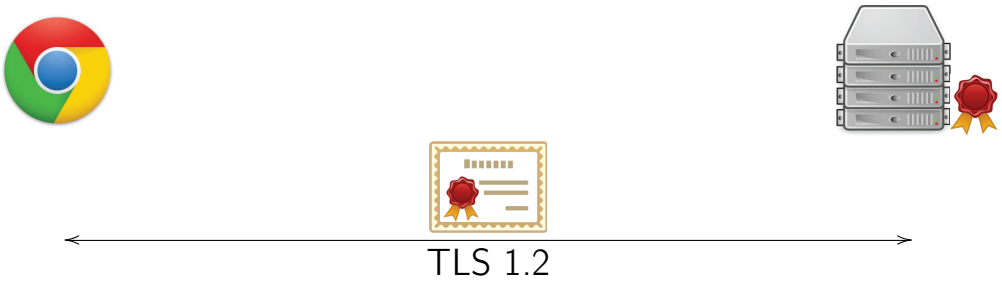
Bourses d'études supérieures du Canada
Vanier
Canada Graduate Scholarships

What is this talk about?

- Strong movement by browsers to improve secure connections
- TLS 1.3 soon?
- Reports about tools undermining this effort, e.g., SuperFish
- What about antiviruses? Parental control applications?

How to intercept/filter TLS traffic?

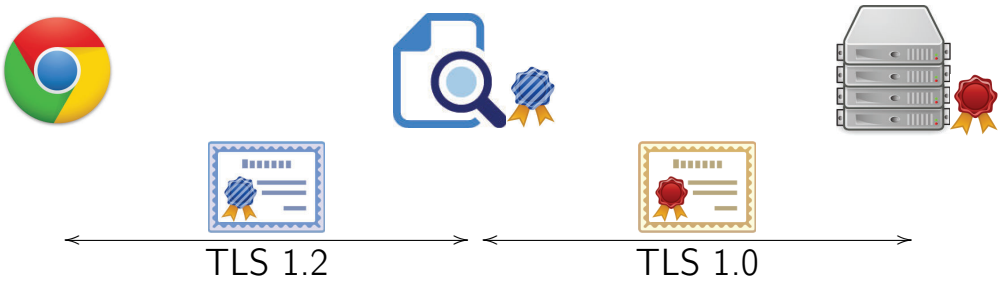
Regular server-authenticated TLS connection:



- Client trusts  or one of its issuers

How to intercept/filter TLS traffic?

Intercepted TLS connection by client-end proxy:



- Client trusts  but, where is the private key?

How to intercept/filter TLS traffic?

Intercepted TLS connection by client-end proxy:

- Client trusts  but, where is the private key? Same system!

Network appliance vs. client-end software (1/2)

TLS filtering by network appliance:

1. Not new, in enterprises
2. Appliances found to be vulnerable by Dell SecureWorks (2012) and CMU CERT (2015)
3. List of "common mistakes"

Network appliance vs. client-end software (2/2)

TLS filtering by client-end software:

1. Relatively new, e.g., advertisement products
2. Scandal early 2015 because of **SuperFish/PrivDog/Komodia**
3. Problems: root certificate reuse, no site certificate validation

Motivations

1. Antivirus and parental control apps filter TLS connections, shown to be significant
2. Existing TLS test suites not adapted for these proxies
3. Bigger attack surface
4. Pre-installed by OEMs $\Rightarrow$ millions of users
5. Antivirus = more security?

Cannot just uninstall antiviruses

Banks sometimes require antiviruses:

12. **Use an Anti-Virus Program, Anti-Spyware Program and Firewall; Signing Off:** The Electronic Device you use may be vulnerable to viruses or online attacks that seek to intercept or alter information including sensitive information that you provide through the Internet. To reduce the chances of harm, you should take all reasonable precautions, including ensuring that any Electronic Device you use to access Online Banking or Wealth Management Online has an up-to-date anti-virus program, anti-spyware program and a firewall, if such security measures are available for your Electronic Device. To prevent unauthorized access to your Accounts, you must sign off of Online Banking or Wealth Management Online, close your browser, or sign-off of the mobile applications used by you for Mobile Banking, as soon as you finish using them.

Contributions

Contributions

1. Design a general hybrid framework: adapt existing + custom tests
    1. Private key protection
    2. Certificate validation
    3. Cipher suites & protocols
    4. Transparency

Contributions

1. Design a general hybrid framework: adapt existing + custom tests
    1. Private key protection
    2. Certificate validation
    3. Cipher suites & protocols
    4. Transparency
2. Review **14** {AntiVirus + Parental Control} apps for Windows

Contributions

1. Design a general hybrid framework: adapt existing + custom tests
    1. Private key protection
    2. Certificate validation
    3. Cipher suites & protocols
    4. Transparency
2. Review **14** {AntiVirus + Parental Control} apps for Windows
3. Found —sometimes major— flaws in *all* 14 products

Analysis

1. Initial list from Wikipedia, AV-comparatives.org, other ad-hoc comparatives: 55 products
2. 14 products, 12 proxies

3. Analyzed in March and August 2015: up to 2 versions/product

Framework

Threat model

Attacker is an active Man-in-the-Middle (MitM). Motivations:

- Impersonate the server to the client
- Extract authentication cookies

Two types of attacks:

1. Generic MitM: no additional per-user effort
2. Targeted MitM: can launch unprivileged code on the target

(1/4) Root certificate and private key

(1/4) Root certificate and private key

1. Is the root certificate install-time generated or pre-generated?

(1/4) Root certificate and private key

1. Is the root certificate install-time generated or pre-generated?
2. Imported in the OS/browser trusted stores?

(1/4) Root certificate and private key

1. Is the root certificate install-time generated or pre-generated?
2. Imported in the OS/browser trusted stores?
3. Period of validity? Removed upon uninstallation?

(1/4) Root certificate and private key

1. Is the root certificate install-time generated or pre-generated?
2. Imported in the OS/browser trusted stores?
3. Period of validity? Removed upon uninstallation?
4. Where/how is the private key stored?

(1/4) Root certificate and private key

1. Is the root certificate install-time generated or pre-generated?
2. Imported in the OS/browser trusted stores?
3. Period of validity? Removed upon uninstallation?
4. Where/how is the private key stored?
5. Does the proxy accept site certificates signed by its own root cert?

(2/4) Site certificate validation

(2/4) Site certificate validation

1. Tests with a corpus of "tricky" certificates
   - 9 invalid certificates/broken chain of trust
   - MD5, SHA1, RSA512, RSA1024

(2/4) Site certificate validation

1. Tests with a corpus of "tricky" certificates
   - 9 invalid certificates/broken chain of trust
   - MD5, SHA1, RSA512, RSA1024
2. How are errors propagated?

(2/4) Site certificate validation

1. Tests with a corpus of "tricky" certificates
   - 9 invalid certificates/broken chain of trust
   - MD5, SHA1, RSA512, RSA1024
2. How are errors propagated?
3. How to make the proxy trust our test root certificate?

(2/4) Site certificate validation

1. Tests with a corpus of "tricky" certificates
   - 9 invalid certificates/broken chain of trust
   - MD5, SHA1, RSA512, RSA1024
2. How are errors propagated?
3. How to make the proxy trust our test root certificate?
4. Which CAs does the proxy trust? OS or custom trusted store?

(2/4) Site certificate validation

1. Tests with a corpus of "tricky" certificates
   - 9 invalid certificates/broken chain of trust
   - MD5, SHA1, RSA512, RSA1024
2. How are errors propagated?
3. How to make the proxy trust our test root certificate?
4. Which CAs does the proxy trust? OS or custom trusted store?
5. Used some network tricks to avoid caching of certificate
   - Other proposals can extend our tests (e.g., Frankencert)

(3/4) Protocol, cipher suites and attacks

(3/4) Protocol, cipher suites and attacks

1. Are all domains filtered? All clients (browsers)? All ports?

(3/4) Protocol, cipher suites and attacks

1. Are all domains filtered? All clients (browsers)? All ports?
2. What library is the proxy using?

(3/4) Protocol, cipher suites and attacks

1. Are all domains filtered? All clients (browsers)? All ports?
2. What library is the proxy using?
3. What are the TLS versions/cipher suites supported by the proxy?

(3/4) Protocol, cipher suites and attacks

1. Are all domains filtered? All clients (browsers)? All ports?
2. What library is the proxy using?
3. What are the TLS versions/cipher suites supported by the proxy?
4. Is the proxy vulnerable to known attacks?

(4/4) Proxy transparency between client/server

1. Does the proxy map TLS parameters between both connections?
2. Does it map certificate's key size and signature hashing algorithm?
3. EV certificates?

Results

Results summary: Part 1

1. Pre-generated certificates (2/14)
2. Proxies accept certificates issued by their root CA (11/12)
3. Root cert. not removed after uninstallation (8/14)
   - Certificates are valid, on average, for 10 years, from 1 to 20
   - CA/Browser forum limits CAs to 3.25 years (5 max)
4. User-readable private keys (9/14)

Results summary: Part 2

1. No certificate validation (3–4/12)
2. Improper signature verification (1/12)
3. MD5 accepted (9/12), RSA512 (7/12)
   - Old such certificates could be revived by NTP attacks
4. Custom CA store (3/12):
   1. DigiNotar+CNNIC
   2. Mozilla Trusted CAs from May 2009
   3. One RSA512 still-valid root CA
5. No revocation check (9/12)

Results summary: Part 3

1. OpenSSL and/or Schannel
2. SSL 3.0 support (6/12), no support for TLS 1.1+ (6/12)
3. RC4 and MD5-based cipher suites (10/12)
   - 1 anonymous Diffie-Hellman
   - 1 export-grade ciphers
4. Proxies vulnerable to Insecure Renegotiation (1), BEAST (7), CRIME (1), FREAK (5), Logjam (3), but not POODLE

Results summary: Part 4

1. Virtual upgrade of TLS version as seen by the client (7/12)
   - SSL 3.0 $\rightarrow$ TLS 1.0 or 1.2
   - TLS 1.0 $\rightarrow$ TLS 1.2
2. Cipher-suites are never transparent, client's choice ignored
3. Fixed-size 1024 or 2048-bit RSA certificates (10/12)
4. Fixed-hash SHA1 or SHA256 certificates (10/12)
5. EV certificates filtered, replaced by DV (11/12, but whitelists)

Practical attacks

Altogether, possible attacks by increasing order of effort (mostly untested):

1. 4 generic MitM out-of-the-box
2. 2 more generic MitM if TLS filtering is activated
3. 1 more generic MitM for an old version
4. 1 CRIME attack (depends on the server)
5. 3 BEAST attacks (depends on the server)
6. 6 targeted MitM (1 still valid post-uninstallation)

Companies contacted, some products are fixed

Recommendations for safer TLS proxying

1. TLS key-logging
2. Private keys: Use OS-provided storage APIs
3. Certificate validation: Rely on the TLS library, handle revocation checks, communicate errors to users, block obvious tampering
4. Transparency: TLS library up-to-date + respect client's choice
5. Browsers: More pro-active, warn users when proxied
6. Servers: Bad proxy fingerprinting by cipher suites?

To recap

1. Designed a framework to test client-end TLS proxies
2. Tested 14 products and uncovered several flaws
3. Provided some guidelines for safer proxying

Contact: x_decarn@ciise.concordia.ca

Questions?

Additional slides

Results, part 1

	Certificate generation time	Reject own root certificate	Removal during uninstallation	Validity (years)	Key protection	Access right
Avast	Installation	×	✓	10	OS API	Admin
AVG	Installation	✓*	✓	10	Obfuscation	Unknown
BitDefender	Installation	×	✓	10	Hardcoded pwd	User
BullGuard AV	Installation	—	×	10	Hardcoded pwd	User
BullGuard IS	Installation	✓	×	10	Hardcoded pwd	User
CYBERsitter	Pre-generated*	×	×	20	Plaintext	User
Dr. Web	Installation	×	×	1	OS API	Admin
ESET	Installation*	×	×	10	OS API	Admin
G DATA	Installation	×	✓	10	Obf. encryption	User
Kaspersky	Installation	×	×	10	Plaintext	User
KinderGate	Installation	×	×	5	Plaintext	User
Net Nanny	Installation	×	✓	10	Modified SQLCipher	User
PC Pandora	Pre-generated	×	✓	10	OS API	Admin
ZoneAlarm	Installation	—	×	10	Plaintext	User