Certified Lies: Detecting and Defeating Government Interception Attacks against SSL* (Short Paper)

Christopher Soghoian and Sid Stamm

Center for Applied Cybersecurity Research, Indiana University
chris@soghoian.net, sid@sidstamm.com

Abstract. This paper introduces the *compelled certificate creation attack*, in which government agencies may compel a certificate authority to issue false SSL certificates that can be used by intelligence agencies to covertly intercept and hijack individuals' secure Web-based communications.

1 Introduction

Consider a hypothetical situation where an executive is in a foreign country for a series of trade negotiations. After a day of meetings, she logs in to her corporate webmail account using her company-provided laptop and the hotel wireless network. Relying on the training she received from her company's IT department, she makes certain to look for the SSL encryption lock icon in her web browser, and only after determining that the connection is secure does she enter her login credentials and then begin to upload materials to be shared with her colleagues. However, unknown to the executive, the foreign government has engaged in a sophisticated man-in-the-middle attack, and is able to covertly intercept the executive's SSL encrypted connections. Agents from the state security apparatus leak details of her communications to the foreign company with whom she is negotiating, who use the information to gain an upperhand in the negotiations. While this scenario is fictitious, the vulnerability is not.

The security and confidentiality of millions of Internet transactions per day depend upon the Secure Socket Layer (SSL)/Transport Layer Security (TLS) protocol. At the core of this system are a number of *Certificate Authorities* (CAs), each of which is responsible for verifying the identity of the entities to whom they grant SSL certificates. It is because of the confidentiality and authenticity provided by the CA based *public key infrastructure* that users around the world can bank online, engage in electronic commerce and communicate with their friends and loved ones about the most sensitive of subjects without having to worry about malicious third parties intercepting and deciphering their communications.

* The full length version of this paper is available at www.dubfire.net. The authors hereby permit the use of this paper under the terms of the Creative Commons Attribution 3.0 United States license.

G. Danezis (Ed.): FC 2011, LNCS 7035, pp. 250–259, 2012.

While not completely obvious, the CAs are all trusted equally in the SSL public key infrastructure, a problem amplified by the fact that the major web browsers trust hundreds of different firms to issue certificates for any site. Each of these firms can be compelled by their national government to issue a certificate for any particular website that all web browsers will trust without warning. Thus, users around the world are put in a position where their browser entrusts their private data, indirectly, to a large number of governments (both foreign and domestic) whom these individuals may not ordinarily trust.

In this paper, we introduces a new attack, the *compelled certificate creation attack*, in which government agencies compel (via a court order or some other legal process) a CA to issue false certificates that are then used by law enforcement and intelligence agencies to covertly intercept and hijack individuals' secure communications. In order to protect users from these powerful government adversaries, we introduce a lightweight defensive browser add-on that detects and thwarts such attacks.

2 Certificate Authorities and the Browser Vendors

> *"[Browser vendors] and users must be careful when deciding which certificates and certificate authorities are acceptable; a dishonest certificate authority can do tremendous damage."*
> — RFC 2246, The TLS Protocol 1.0 [1]

CAs play a vital role in the SSL *public key infrastructure* (PKI). Each CA's main responsibility is to verify the identity of the entity to which it issues a certificate. Thus, when a user visits `https://www.bankofamerica.com`, her browser will inform her that the bank's certificate is valid, was issued by VeriSign, and that the website is run by Bank of America. It is because of the authenticity and confidentiality guaranteed by SSL that the user can continue with her transaction without having to worry that she is being phished by cyber-criminals.

CAs generally fall into one of three categories: Those trusted by the browsers ("root CAs"), those trusted by one of the root CAs ("intermediate CAs" or "subordinate CAs"), and those neither trusted by the browsers nor any intermediate CA ("untrusted CAs"). Furthermore, intermediate CAs do not necessarily have to be directly verified by a root CA — but can be verified by another intermediate CA, as long as the *chain of trust* eventually ends with a root CA. The problem, however, is that each of the hundreds of different root CAs are equally trusted by the browsers to issue certificates for any site.

From the end users' perspective, root CAs and intermediate CAs are functionally equivalent. A website that presents a certificate signed by either form of CA will cause the users' browser to display a lock icon and to change the color of the location bar. Whereas certificates verified by an untrusted CA and those self-signed by the website owner will result in the display of a security warning, which for many non-technical users can be scary [2], confusing, and difficult to bypass in order to continue navigating the site [3].

It is important to note that there are no technical restrictions in place that prohibit a CA from issuing a certificate to a malicious third party. Thus, both the integrity of the CA based public key infrastructure and the security users' communications depend upon hundreds of CAs around the world choosing to do the right thing. Unfortunately, as will soon be clear, any one of those CAs can become the weakest link in the chain.

3 Compelled Assistance

Many governments routinely compel companies to assist them with surveillance. Telecommunications carriers and Internet service providers are frequently required to violate their customers' privacy — providing the government with email communications, telephone calls, search engine records, financial transactions and geo-location information.

In the United States, the legal statutes defining the range of entities that can be compelled to assist in electronic surveillance by law enforcement and foreign intelligence investigators are remarkably broad [4]. Examples of compelled assistance using these statutes include a secure email provider that was required to place a covert back door in its product in order to steal users' encryption keys [5], and a consumer electronics company that was forced to remotely enable the microphones in a suspect's auto-mobile dashboard GPS navigation unit in order to covertly record their conversations [6].

Outside of the United States and other democratic countries, specific statutory authority may be even less important. The Chinese government, for example, has repeatedly compelled the assistance of telecommunications and technology companies in assisting it with its surveillance efforts [7, 8].

Just as phone companies and email providers can be forced to assist governments in their surveillance efforts, so too can SSL certificate authorities. The *compelled certificate creation attack* is thus one in which a government agency requires a domestic certificate authority to provide it with false SSL certificates for use in surveillance.

The technical details of this attack are simple, and do not require extensive explanation. Each CA already has an infrastructure in place with which it is able to issue SSL certificates. In this compelled assistance scenario, the CA is merely required to skip the identity verification step in its own SSL certificate issuance process.

When compelling the assistance of a CA, the government agency can either require the CA to issue it a specific certificate for each website to be spoofed, or, more likely, the CA can be forced to issue a intermediate CA certificate that can then be re-used an infinite number of times by that government agency, without the knowledge or further assistance of the CA. Furthermore, such an intermediate issuing CA can be installed into surveillance appliances already available on the market and quickly deployed to intercept any traffic [9].

4 Protecting Users

The major web browsers are currently vulnerable to the compelled certificate creation attack, and we do not believe that any of the existing privacy enhancing browser add-ons sufficiently protect users without significantly impacting browser usability.

In an effort to significantly reduce the impact of this attack upon end-users, we have created *Certlock*, a lightweight add-on for the Firefox browser. Our solution employs a Trust-On-First-Use (TOFU) policy (this is also known as 'leap-of-faith' authentication) [10, 11], reinforced with a policy that the country of origin for certificate issuing does not change in the future. Specifically, our solution relies upon caching CA information, that is then used to empower users to leverage country-level information in order to make common-sense trust evaluations.

In this section, we will outline the motivations that impacted the design of our solution, discuss our belief in the potential for users to make wise country-level trust decisions, and then explore the technical implementation details of our prototype add-on.

Design Motivations. The compelled certificate creation attack is a classic example of a low probability, high impact event [12]. The vast majority of users are extremely unlikely to experience it, but for those who do, very bad things are afoot. As such, it is vital that any defensive technique have an extremely low false positive rate, yet be able to get the attention of users when an attempted SSL session hijacking is detected.

Country-Based Trust. We believe that many consumers are quite capable of making basic trust decisions based on country-level information. We are not alone in this belief. Since March 2010, Google has been providing country-level warnings to users of its Google Mail service when it detects that their account has been accessed from a potentially suspect IP address in a different country [13].

Thus, a consumer whose banking sessions are normally encrypted by a server presenting a certificates signed by a US based CA might become suspicious if told that her US based bank is now using a certificate signed by a Tunisian, Latvian or Serbian CA.

To make this trust evaluation, she doesn't have to study the detailed business policies of the foreign CA, she can instead rely on geographical prejudice, and ask herself why her Iowa based bank is suddenly doing business in Eastern Europe. In order to empower users to make such country-level evaluations of trust, CertLock leverages the wealth of historical browsing data kept by the browser.

Likewise, individuals located in countries with oppressive governments may wish to know if their communications with servers located in foreign democracies are suddenly being facilitated by a domestic (or state controlled) CA.

Avoiding False Positives. A simplistic defensive add-on aimed at protecting users from compelled certificate creation attacks could simply cache all certificates

encountered during browsing sessions, and then warn the user any time they encounter a certificate that has changed. In fact, such an add-on, Certificate Patrol, already exists [14].

Unfortunately, this approach is likely to generate too many false positives. Each time a website intentionally changes its certificate, the browser displays a warning that will needlessly scare and soon desensitize users. There are many legitimate scenarios where certificates change. For example: Old certificates expire; certificates are abandoned and or revoked after a data breach that exposed the server private key; and many large enterprises that have multiple SSL accelerator appliances serving content for the same domain use a different certificate for each device [15].

By adopting a Trust-On-First-Use policy, we assume that if a website starts using a different certificate issued by the same CA that issued its previous certificate, there is no reason to warn the user. This approach enables us to significantly reduce the false positive rate, while having little impact on our ability to protect users from a variety of threats.

We also believe that there is little reason to warn users if a website switches CAs within the same country. As our threat model is focused on a government adversary with the power to compel any domestic CA into issuing certificates at will, we consider CAs within a country to be equals. That is, a government agency able to compel a new CA into issuing a certificate could just as easily compel the original CA into issuing a new certificate for the same site. Since we have already opted to not warn users in that scenario (described above), there is no need to warn users in the event of a same-country CA change.

Implementation Details. Our Certlock solution is currently implemented as an add-on to the Firefox browser. Because the Firefox browser already retains history data for all visited websites, we have simply modified the browser to cause it to retain slightly more information. Thus, for each new SSL protected website that the user visits, a Certlock enabled browser also caches the following additional certificate information: (a) A hash of the certificate, (b) the country of the issuing CA, (c) the name of the CA, (d) the country of the website, (e) the name of the website and (f) the entire chain of trust up to the root CA.

When a user re-visits a SSL protected website, Certlock first calculates the hash of the site's certificate and compares it to the stored hash from previous visits. If it hasn't changed, the page is loaded without warning. If the certificate has changed, the CAs that issued the old and new certificates are compared. If the CAs are the same, or from the same country, the page is loaded without any warning. If, on the other hand, the CAs' countries differ, then the user will see a warning.

5 Related Work

Over the past decade, many people in the security community have commented on the state of the SSL public key infrastructure, and the significant trust placed in the CAs [16–18].

In 1998, James Hayes of the US National Security Agency published a paper that focused specifically on the threat of rogue insiders within a Certificiate Authority [19]. Although the technical details of the threat outlined by Hayes are largely the same as the scenario on which we have focused (albeit with vasty different legal and policy consequences), Hayes did not address the threat of government compelled certificate creation. It is unclear if he was simply unaware of this scenario, or if the topic was too sensitive for him to discuss, given his employer. In his paper, Hayes proposed a technical solution to address the insider threat, which relied on users configuring various per-site attributes within their browser that would be used to evaluate each new site's certificate.

Crispo and Lomas also proposed a certification scheme designed to detect rogue CAs [20], while the Monkeysphere project has created a system that replaces the CA architecture with the OpenPGP web of trust [21].

Ian Grigg has repeatedly sought to draw attention to both the potential conflict of interest that some CAs have due to their involvement in other forms of surveillance, and the power of a court order to further compel these entities to assist government investigations [22–24]. In particular, in 2005, Grigg and Shostack filed a formal complaint with ICANN over the proposal to award VeriSign control of .net domain name registration, arguing that the firm's surveillance products created a conflict of interest [25].

In recent years, several browser-based tools have been created to help protect users against SSL related attacks. Kai Engert created Conspiracy, a Firefox add-on that provides country-level CA information to end-users in order to protect them from compelled certificate creation attacks. The Conspiracy tool displays the flag of the country of each CA in the chain of trust in the browser's status bar [26]. Thus, users must themselves remember the country of the CAs that issue each certificate, and detect when the countries have changed. We believe, like Herley [27], that this is an unreasonable burden to place upon end-users, considering how rarely the compelled certificate creation attack is likely to occur.

Wendlandt *et al.* created Perspectives, a Firefox add-on that improves the Trust-On-First-Use model used for websites that supply self-signed SSL certificates [28]. In their system, the user's browser securely contacts one of several notary servers, who in turn independently contact the webserver and obtain its certificate. In the event that an attacker is attempting to perform a man in the middle attack upon the user, the fact that the attacker-supplied SSL certificate, and those supplied by the Perspectives notary servers differ will be a strong indicator that something bad has happened. Unfortunately, the Perspectives system requires that users provide the Perspectives notaries with a real-time list of the secure sites they visit.

Alicherry and Keromytis have improved upon the Perspectives design with their DoubleCheck system [29], substituting Tor exit nodes for special notary servers. Because the Tor network anonymizes the individual user's IP address, there is no way for the Tor exit nodes to know who is requesting the certificate for a particular SSL website. While the authors solved the major privacy issues that plague the Perspectives scheme, their choice of Tor carries its own cost:

Latency. Their system adds an additional second of latency to every new SSL connection, and up to 15 seconds for visits to new self-signed servers. We believe that this additional latency is too much to ask most users to bear, particularly if the chance of them encountering a rogue CA is so low.

Herzberg and Jbara created TrustBar, a Firefox add-on designed to help users detect spoofed websites. The browser tool works by prominently displaying the name of the CA that provided the site's certificate, as well as allowing the user to assign a per-site name or logo, to be displayed when they revisit to each site [30].

Tyler Close created Petname Tool, a Firefox add-on that caches SSL certificates, and allows users to assign a per-site phrase that is displayed each time they revisie the site in the future. In the event that a user visits a spoofed website, or a site with the same URL that presents a certificate from a different CA, the user's specified phrase will not be displayed [31].

In May 2008, a security researcher discovered that the OpenSSL library used by several popular Linux distributions was generating weak cryptographic keys. While the two-year old flaw was soon fixed, SSL certificates created on computers running the flawed code were themselves open to attack [32, 33]. Responding to this flaw, German technology magazine Heise released the Heise SSL Guardian for the Windows operating system, which warns users of Internet Explorer and Chrome when they encounter a weak SSL certificate [34].

In December 2008, Stevens *et al.* demonstrated that flaws in the MD5 algorithm could be used to create rogue SSL certificates (without the knowledge or assistance of the CA). In response, CAs soon accelerated their planned transition to certificates using the SHA family of hash functions [35]. As an additional protective measure, Márton Anka developed an add-on for the Firefox browser to detect and warn users about certificate chains that use the MD5 algorithm for RSA signatures [36].

Jackson and Barth devised the ForceHTTPS system to protect users who visit HTTPS protected websites, but who are vulnerable to man in the middle attacks due to the fact that they do not type in the `https://` component of the URL [37]. This system has since been formalized into the HTTP Strict Transport Security (HSTS) standard proposal [38], to which multiple browsers are in the process of adding support. While this system is designed to enable a website to hint to the browser that future visits should always occur via a HTTPS connection, this mechanism could be extended to enable a website to lock a website to a particular CA, or CAs of a specific country.

6 Conclusion and Future Work

In this paper, we introduced the compelled certificate creation attack and presented evidence that suggests that governments may be subverting the CA based public key infrastructure. In an effort to protect users from these powerful adversaries, we introduced a lightweight defensive browser based add-on that detects

and thwarts such attacks. Finally, we use reductive analysis of governments' legal capabilities to perform an adversarial threat model analysis of the attack and our proposed defensive technology.

Our browser add-on is currently just a prototype, and we plan to improve it in the future. We plan to explore the possibility of expanding the country-level trust model to regions, such as the European Union, where, for example, residents of the Netherlands may be willing to trust Belgian CAs. We are also considering adding a feature that will enable users to voluntarily submit potentially suspect certificates to a central server, so that they can be studied by experts. Such a feature, as long as it is opt-in, does not collect any identifiable data on the user, and only occurs when potentially rogue certificates are discovered, would have few if any privacy issues.

Ultimately, the threats posed by the compelled certificate creation attack cannot be completely eliminated via our simple browser add-on. The CA system is fundamentally broken, and must be overhauled. DNSSEC may play a significant role in solving this problem, or at least reducing the number of entities who can be compelled to violate users' trust. No matter what system eventually replaces the current one, the security community must consider compelled government assistance as a realistic threat, and ensure that any solution be resistant to such attacks.

Acknowledgements. Thanks to Kevin Bankston, Matt Blaze, Kelly Caine, Jon Callas, Allan Friedman, Jennifer Granick, Markus Jakobsson, Dan Kaminsky, Moxie Marlinspike, Eddy Nigg, Eli O and Adam Shostack for their useful feedback.

References

1. Dierks, T., Allen, C.: The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), Obsoleted by RFC 4346, updated by RFCs 3546, 5746 (January 1999)
2. Nightingale, J.: SSL Question Corner. meandering wildly (blog) (August 5, 2008), http://blog.johnath.com/2008/08/05/ssl-question-corner/
3. Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., Cranor, L.F.: Crying wolf: An empirical study of SSL warning effectiveness. In: Proceedings of the 18th Usenix Security Symposium (August 2009)
4. Soghoian, C.: Caught in the cloud: Privacy, encryption, and government back doors in the web 2.0 era. Journal on Telecommunications and High Technology Law (forthcoming)
5. Singel, R.: PGP Creator Defends Hushmail. Wired News Threat Level Blog (November 19, 2007), http://www.wired.com/threatlevel/2007/11/pgp-creator-def
6. McCullagh, D.: Court to FBI: No spying on in-car computers. CNET News (November 19, 2003), http://news.cnet.com/2100-1029_3-5109435.html
7. Markoff, J.: Surveillance of skype messages found in china. The New York Times (October 1, 2008), http://www.nytimes.com/2008/10/02/technology/internet/02skype.html
8. Jacobs, A.: China requires censorship software on new pcs. The New York Times (June 8, 2009), http://www.nytimes.com/2009/06/09/world/asia/09china.html

9. Singel, R.: Law Enforcement Appliance Subverts SSL. Wired News Threat Level Blog (March 24, 2010), http://www.wired.com/threatlevel/2010/03/packet-forensics/
10. Stajano, F., Anderson, R.J.: The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks. In: Malcolm, J.A., Christianson, B., Crispo, B., Roe, M., et al. (eds.) Security Protocols 1999. LNCS, vol. 1796, pp. 172–182. Springer, Heidelberg (2000)
11. Arkko, J., Nikander, P.: Weak Authentication: How to Authenticate Unknown Principals without Trusted Parties. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2002. LNCS, vol. 2845, pp. 5–19. Springer, Heidelberg (2004)
12. Bussiere, M., Fratzscher, M.: Low probability, high impact: Policy making and extreme events. Journal of Policy Modeling 30(1), 111–121 (2008)
13. Diwanji, P.: Detecting suspicious account activity. The Official Gmail Blog (March 24, 2010), http://gmailblog.blogspot.com/2010/03/detecting-suspicious-account-activity.html
14. Certificate patrol (2010), http://patrol.psyced.org/
15. Kaminsky, D.: Email conversation with author (February 28, 2010)
16. Gillmor, D.K.: Technical Architecture shapes Social Structure: an example from the real world (February 21, 2007), http://lair.fifthhorseman.net/~dkg/tls-centralization/
17. Peter SJF Bance. Ssl: Whom do you trust? (April 20, 2005), http://www.minstrel.org.uk/papers/2005.04.20-ssl-trust.pdf
18. Ed Gerck. First published online by the MCWG at http://mcwg.org/cert.htm (April 1997). Invited talk at the Black Hat Briefings 1999, Las Vegas, NV, July 7-8 (1999). Published by The Bell, ISSN 1530-048X, Vol. 1, No. 3, p. 8 (July 2000), http://www.thebell.net/papers/certover.pdf
19. Hayes, J.M.: The problem with multiple roots in web browsers - certificate masquerading. In: WETICE 1998: Proceedings of the 7th Workshop on Enabling Technologies, pp. 306–313. IEEE Computer Society, Washington, DC (1998)
20. Crispo, B., Lomas, M.: A Certification Scheme for Electronic Commerce. In: Lomas, M. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 19–32. Springer, Heidelberg (1997)
21. Monkeysphere (2010), http://web.monkeysphere.info/
22. Grigg, I.: VeriSign's conflict of interest creates new threat. Financial Cryptography (blog) (September 1, 2004), http://financialcryptography.com/mt/archives/000206.html
23. Grigg, I.: PKI considered harmful (October 14, 2008), http://iang.org/ssl/pki_considered_harmful.html
24. Grigg, I.: Why the browsers must change their old SSL security (?) model. In: Financial Cryptography (blog) (March 24, 2010), financialcryptography.com/mt/archives/001232.html
25. Grigg, I., Shostack, A.: VeriSign and Conflicts of Interest (February 2, 2005), http://forum.icann.org/lists/net-rfp-verisign/msg00008.html
26. Engert, K.: Conspiracy — A Mozilla Firefox Extension (March 18, 2010), http://kuix.de/conspiracy/
27. Herley, C.: So long, and no thanks for the externalities: the rational rejection of security advice by users. In: NSPW 2009: Proceedings of the 2009 Workshop on New Security Paradigms Workshop, pp. 133–144 (September 2009)

28. Wendlandt, D., Andersen, D.G., Perrig, A.: Perspectives: improving ssh-style host authentication with multi-path probing. In: ATC 2008: USENIX 2008 Annual Technical Conference on Annual Technical Conference, pp. 321–334. USENIX Association, Berkeley (2008)
29. Alicherry, M., Keromytis, A.D.: Doublecheck: Multi-path verification against man-in-the-middle attacks. In: ISCC 2009: IEEE Symposium on Computers and Communications, pp. 557–563. IEEE, Piscataway (2009)
30. Herzberg, A., Jbara, A.: Security and identification indicators for browsers against spoofing and phishing attacks. ACM Trans. Internet Technol. 8(4), 1–36 (2008)
31. Close, T.: Petname tool (2005), http://www.waterken.com/user/PetnameTool/
32. Ahmad, D.: Two Years of Broken Crypto: Debian's Dress Rehearsal for a Global PKI Compromise. IEEE Security and Privacy 6, 70–73 (2008)
33. Yilek, S., Rescorla, E., Shacham, H., Enright, B., Savage, S.: When private keys are public: results from the 2008 Debian OpenSSL vulnerability. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, pp. 15–27. ACM, New York (2009)
34. The H Security. heise SSL Guardian: Protection against unsafe SSL certificates (July 4, 2008), www.h-online.com/security/features/Heise-SSL-Guardian-746213.html
35. Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D.A., de Weger, B.: Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 55–69. Springer, Heidelberg (2009)
36. Anka, M.: SSL Blacklist 4.0 (January 31, 2010), http://www.codefromthe70s.org/sslblacklist.aspx
37. Jackson, C., Barth, A.: Forcehttps: protecting high-security web sites from network attacks. In: WWW 2008: Proceeding of the 17th International Conference on World Wide Web, pp. 525–534. ACM, New York (2008)
38. Hodges, J., Jackson, C., Barth, A.: Strict Transport Security (December 18, 2009), lists.w3.org/Archives/Public/www-archive/2009Dec/att-0048/draft-hodges-strict-transport-sec-06.plain.html