SSL Security in Browser plug-in: Managing the list of CAs

Nikhil Khatu, Yuri Kolesnikov
{ngkhatu,ykolesn}@ncsu.edu

4/2/2013

Abstract

Public Key Infrastructure (PKI) is trusted by web browsers for SSL security. A ubiquitous list of root CAs forms the default basis of trust within the Web Browser of the average consumer. Sub-authorities are allowed to sign for authenticity on behalf of their root CAs. This has the result of producing an overgrown network of seemingly trusted nodes, some of which shouldn't be trusted. We propose maintaining a separate list of personally trusted Certificate Authorities via a developed browser plugin. The list is populated by prompting the user to "accept" a CA that is not already present the list, eventually eliminating the need for acceptance prompts. By maintaining a user created list, we can abate the risk of trusting less ubiquitous CAs. After development we sample the users' CA profiles from the plugin. Each user will have a certain number of trusted CAs in their list from day to day browsing. The list will vary from user to user. The reduced number of CAs in these lists has a correlation with the risk abated. By default browsers trust many more CAs than are actually required by the average user.

1 Introduction

A root certificate authority (CA) list is the basis for remote security in today's web browsers. The PKI requires the list be maintained in a local file by either the operating system or the browser itself. While there are several security flaws inherently present when using CAs with any Public Key Infrastructure one must also be wary of the trust model.[7] Our current Oligarchy PKI model leverages the use of trust anchors, where a certificate issued by any one of the trust anchors is accepted.[14] With the use of CA chains a trust anchor, or root CA, can vouch for other sub-authorities who can then vouch for even more sub-authorities. The web browser confirms the authenticity and trustworthiness of this chain by checking the validity of each signature in the chain up to the root CA. If the root CA matches an entry in the browsers trusted CA list, the websites certificate is considered validated by the browser. The increased presence of sub-authorities combined with the average internet user's inherent trust in the browser's handling of certificate validation increase the probability of compromise in the system. A compromised sub-authority can be used to forge certificates for fraudulent websites claiming to be legitimate. An example would be an attacker using a fraudulent certificate to present a forged bank login page to a user during a man in the middle attack.

Several PKI trust models have been proposed; mostly rooting from monopolistic, oligarchic, anarchic, and constraining CAs to a particular subset.[14] However there has not been much initiative to mold and constrain PKI. The monopolistic model anchors too much trust at a central authority, while an anarchic PKI model is only as great as its weakest node. The anarchic model is bound to degrade once the CA market saturates. Our current PKI model allows CAs to profit and there is little incentive to drive change.

Assuming the PKI model remains relatively static we propose enhancing the browsers CA validation functionality. The solution is implemented as a browser plugin which allows the user to assign the level of trust given to a new sub-authority during initial access of a secured site. If the user decides to trust the sub authority, the entry is stored in a separate sub-authority database maintained by the plugin.

According to the SSL Observatory project, Mozillas Firefox browser stores 124 trusted CAs. Microsofts built in Windows CA database, used by Googles Chrome browser and Internet Explorer, typically has over 300 certificates through updates. This large number of trusted CAs leads to a recorded 1,377,067 valid leaf certificates. [4] The new list of sub-authorities that is derived from using our plugin aims to greatly reduce the number of valid sub-authorities trusted by the average internet user. The list eventually grows large enough to provide a seamless experience for daily web browsing. The final lists collected during our plugin evaluation also allow us to draw comparisons to the default size of Mozilla and Microsofts default CA directories. Our plugin approach is also advantageous in allowing the user to have a more hands-on experience with their browser security settings. In an area where most browser security settings are developed

to be as automated and out of sight as possible, our plugin serves to expose and educate the user on the risks and vulnerabilities behind the simplistic "green lock" icon in their address bar.

2 Related Work

There are many works published on securing SSL in infrastructure. The Related works section is divided into the following sections: Phishing and Browser related User Vulnerabilities, PKI Trust Infrastructure, and SSL protocol Vulnerabilities.

2.1 PKI Trust Infrastructure

Public Key Infrastructures (PKI) depends on a reliable method of authentication. One such method is used in the Secure Sockets Layer (SSL), where Certificate Authorities are used to verify a website's signature, and thus confirm the authenticity of the website providing the signature. The Achilies heel of any remote system is the Man in the Middle Attack. On trusting CA roots; it is possible to pose as a MITM and insert a certificate signed by a different root CA, but still be trusted by the browser due to the browser trusting multiple roots. [8] It is also possible to compromise a root CA, and thus make it sign many fraudulent child certificates. [19] One solution to the multiple CA root CAs can possibly be addressed by short lived certificates at the cost of performance. [6] Roosa et al studied the structural defects and lessons learned overthe lifetime of the CA trust model. [21]

The security of SSL enabled applications is heavily dependent on securing current Public Key Infrastructure and the trust of Certification Authorities (CA). Three basic PKI models include PKIX(x.509), SPKI, and PGP.[11][15] The trust and reliance on a browser's CA database is a major point of failure, and a popular topic for research. [15]In Josang et al Trust Management in PKI is the exclusive focus to benefit from SSL security.[11] Since the PKI is commercialized, the low barriers to entry leave the system vulnerable to new untrusted Certification Authorities entering the market.[7] It is important for any party trusting a CA to analyze various properties of the Certificate; certificate type, security risks of the key holder, certificate and organizaiton of CA. [31] Various models from Anarchical to Hierarchical have been proposed to better manage the infrastructure.[14] While current PKI remains mostly pessimistic in regards to trusting commercial entities Wilson et al remains optimistic and even suggest building a "Superstructure" with existing mature PKI to improve utility and practicallity of Digital Certificates.[28] NIST provides documentation on current implementations.[16] Slagell et al adapt existing PKI protocols to solve scalability issues [25] Current research is also being done on the exchanging of signed certificates without the use of CAs. [23]

2.2 Browser Vulnerabilities and Phishing

Even if CAs are secure, most novice users are susceptible to what is known as a Man In The Middle (MITM) Phishing attack where the attacker can display spoofed login forms to the user. Therefore it is important to consider this style of attack and the presence of novice security users when designing a plugin to improve the browser's CA database. Phishing is generally executed in an unsophisticated manner, many users do not make the effort to check things as simple as being at the correct URL before entering sensitive information. [5] [20] The Yahoo! sign-in seal allows users to personalize the image used during sign in which will be used as a security seal.[1] Several solutions to phishing attacks in the form of browser plugins have been proposed. Most early anti phishing plugins attempt to prevent the user from accessing a spoofed website through the confirmation of a website's certificate. Once the certificate is confirmed, the user is clearly informed that they are on a legitimate website. [13] [2] [17] [27] [18] [30] Other plugins such as SpoofGuard analyze the HTML POST request from the user to determine the authenticity of a website. SpoofGuard then displays a warning message when an invalid website is detected. [3] Most confirmation message related toolbars are ineffective due to the fact that most internet users do not understand phishing attacks, and do not comprehend how complex they can be. Users simply fail to pay attention to even the most clear warnings [29] A complex learning approach, similar to our more simple CA database learning plugin, was proposed by Sharifi et al. [24] The system works by monitoring user habits such as ignoring certificate warnings. The system will then either automatically choose to ignore such warnings, or prompt the user with a more thorough explanation of their risky behavior.

The next generation of phishing attacks will prove to be more sophisticated than current attacks strictly focused on collecting personal information. Due to current SSL technology being a one-way, server to user, system, a MITM attacker can now authenticate credentials in real time and present a seemingly valid phishing site to the user. Hashing of the users password with the websites public key can be used to prevent these types of attacks. [12] A hashing plugin is originally described in PwdHash, where a browser plugin hashes the users password with the websites domain name, thus making the password hashed password useless for other websites. [22]

2.3 SSL Vulnerabilities

PKI infrastructure heavily relies on the security of the SSL Application Layer protocol and handshaking used to trade public and session keys it worth mentioning that there are various studies on strengthing the protocol. SSL by itself is a secure protocol, when used in conjunction with PKI vulnerabilities are exposed. There is a proposal to authenticate the user to avoid MITM attacks. Since SSL security is usually one way (server to user) Herzberg et al. propose a system, TPL, in which a certificate authority authenticates the user. [9] Another is a Scheme to improve the security of SSL - [10][26]

References

[1] N. Agarwal, S. Renfro, and A. Bejar. Phishing forbidden. *Queue*, 5(5):28–32, 2007.

[2] R. Biddle, P. Van Oorschot, A. S. Patrick, J. Sobey, and T. Whalen. Browser interfaces and extended validation ssl certificates: an empirical study. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 19–30. ACM, 2009.

[3] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell. Client-side defense against web-based identity theft. In *11th Annual Network and Distributed System Security Symposium (NDSS04)*. San Diego, USA, 2004.

[4] Defcon. *An Observatory for the SSLiverse*, volume 18, 2010.

[5] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590. ACM, 2006.

[6] L. H. C. J. E. Topalovic, B. Saeta and D. Boneh. Towards short lived certificates. In *IIEEE Oakland Web 2.0 Security and Privacy*, pages 1–6, 2012.

[7] C. Ellison and B. Schneier. Ten risks of pki: What you're not being told about public key infrastructure. *Comput Secur J*, 16(1):1–7, 2000.

[8] J. Hayes. The problem with multiple roots in web browsers-certificate masquerading. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1998. (WET ICE '98) Proceedings., Seventh IEEE International Workshops on*, pages 306–311, Jun 1998.

[9] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 2–14. IEEE, 2000.

[10] Z. Huawei and L. Ruixia. A scheme to improve security of ssl. In *Circuits, Communications and Systems, 2009. PACCS'09. Pacific-Asia Conference on*, pages 401–404. IEEE, 2009.

[11] A. Jøsang, I. Pedersen, and D. Povey. Pki seeks a trusting relationship. In *Information Security and Privacy*, pages 191–205. Springer, 2000.

[12] Y. Joshi, D. Das, and S. Saha. Mitigating man in the middle attack over secure sockets layer. In *Internet Multimedia Services Architecture and Applications*

(IMSAA), 2009 IEEE International Conference on, pages 1–5, Dec.

[13] Y. Joshi, S. Saklikar, D. Das, and S. Saha. Phishguard: A browser plug-in for protection from phishing. In *Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on*, pages 1–6, Dec.

[14] C. Kaufman, R. Perlman, and M. Speciner. *Network security: private communication in a public world*. Prentice Hall Press, 2002.

[15] S. Kent. Evaluating certification authority security. In *Aerospace Conference, 1998 IEEE*, volume 4, pages 319–327. IEEE, 1998.

[16] D. R. Kuhn, V. C. Hu, W. T. Polk, and S.-J. Chang. Introduction to public key technology and the federal pki infrastructure. Technical report, DTIC Document, 2001.

[17] O. MahMood. Custom plugin - a solution to phishing and pharming attacks. In *Proceedings of the 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing*, pages 32–38. World Congress In Computer Science, 2006.

[18] M. Maurer, A. Luca, and T. Stockinger. Shiningchrome: Using web browser personas to enhance ssl certificate visualization. In *Human-Computer Interaction*, pages 44–51, 2011.

[19] H. M. Park. The web accessibility crisis of the korea's electronic government: Fatal consequences of the digital signature law and public key certificate. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 2319–2328. IEEE, 2012.

[20] T. Raffetseder, E. Kirda, and C. Kruegel. Building anti-phishing browser plug-ins: An experience report. In *Proceedings of the Third International Workshop on Software Engineering for Secure Systems*, page 6. IEEE Computer Society, 2007.

[21] S. Roosa and S. Schultze. Trust darknet: Control and compromise in the internet and certificate authority model. volume PP, pages 1–1.

[22] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. Mitchell. Stronger password authentication using browser extensions. page 15. Stanford Crypto, 2005.

[23] M. Sharifi, E. Fink, and J. Carbonell. Verikey: A dynamic certificate verification system for public key exchanges. In *Detections of Intrusions and Malware, and Vulnerability Assessment*, pages 44–63, Jul 2008.

[24] M. Sharifi, E. Fink, and J. Carbonell. Learning of personalized security settings. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 3428–3432, Oct.

[25] A. J. Slagell and R. Bonilla. Pki scalability issues. *arXiv preprint cs/0409018*, 2004.

[26] C. Soghoian and S. Stamm. Certified lies: Detecting and defeating government interception attacks against ssl (short paper). *Financial Cryptography and Data Security*, pages 250–259, 2012.

[27] A. Upadhyaya. Design of a plugin for a browser against phishing and spoofing attacks. *World Journal of Science*, 2:30–33, 2012.

[28] S. Wilson. Public key superstructure. 2008.

[29] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610. ACM, 2006.

[30] Z. E. Ye, S. Smith, and D. Anthony. Trusted paths for browsers. *ACM Transactions on Information and System Security (TISSEC)*, 8(2):153–186, 2005.

[31] M. Zhang, X. Zheng, S. Lv, and Y. Yu. Improved approach on modeling and reasoning about pki/wpki. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1–4. IEEE, 2010.