POSTER: Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties

Italo Dacosta*, Mustaque Ahamad† and Patrick Traynor†
*Student**, *Faculty*†
Converging Infrastructure Security (CISEC) Laboratory
Georgia Institute of Technology
{*idacosta, mustaq, traynor*}*@cc.gatech.edu*

The Secure Sockets Layer (SSL) protocol and its successor, Transport Layer Security (TLS), have become the de facto means of providing strong cryptographic protection for network traffic. Their near universal integration with web browsers arguably makes them the most visible pieces of security infrastructure for average users. While vulnerabilities are occasionally found in specific implementations, SSL/TLS are widely viewed as robust means of providing confidentiality, integrity and server authentication. However, these guarantees are built on tenuous assumptions about the ability to authenticate the server-side of a transaction by using digital certificates signed by a *trusted* third-party certification authority (CA).

The security community has long been critical of the Public Key Infrastructure for X.509 (PKIX) and its CA-based trust model [5], [9], [1]. Much of the concern has focused on the role of the CAs and their ability and motivation to not only correctly verify and attest the coupling between an identity and a public key, but also to protect their own resources. Browsers and operating systems determine what CAs users should trust by default (i.e., trust anchors). However, this model has resulted in *hundreds of CAs, all equally trusted and from more than 50 different countries* [19], [3]. Due to this excessive trust, CAs can forge certificates for any domain that will be accepted as valid by most browsers. Thus, adversaries can obtain forged certificates by coercing or compromising any CA and use them to execute man-in-the-middle (MITM) attacks against SSL/TLS connections. The number of reported attacks against CAs increased considerably last year [18], [11], [6], [8], [12], [2], [17]. In some cases, adversaries were able to forge certificates for important web domains (e.g., google.com, yahoo.com and live.com). Even worse, it has been estimated that a forged certificate was used to intercept close to 300,000 Gmail sessions in Iran [14]. Furthermore, there is evidence that governments and private organizations are using forged certificates as part of their surveillance and censorship efforts [20], [7], [21], [15]. The frequency of these incidents is likely to increase in the future, as more and more web applications rely on SSL/TLS to protect all their communications. Unfortunately, sufficient mechanisms for detecting and preventing this problem are currently lacking.

Multiple solutions have been proposed to deal with the threat imposed by forged certificates and MITM attacks. The most popular approach is the use of additional third-parties to extend or replace the rigid CA trust model (e.g., network notaries [22], [16], public audit logs [4], [13] and secure DNS (DNSSEC) [10]). In this approach, users can select one or more third-parties to vouch for the authenticity of a certificate, improving the chances of detecting a MITM attack. However, depending only on third-parties for certificate validation has several shortcomings such as: significant deployment and operational costs (e.g., additional infrastructure with high availability requirements), more complex trust model for users, privacy concerns and more complex revocation procedures. Therefore, *the inherent complexity and costs associated with third-party solutions have prevented their widespread deployment*. As a result, most users still rely on weak certificate validation checks to detect MITM attacks.

In this poster we present Direct Validation of Certificates (DVCert), an efficient and easy to deploy protocol that provides stronger certificate validation and effective detection of MITM attacks without using third-parties. Our mechanism comes from a simple observation – users have already established secrets (e.g., passwords) with their most important web applications. *DVCert allows web applications to use these secrets to directly and securely attest for the authenticity of their certificates without exposing those secrets to offline attacks*. After a single round-trip DVCert transaction, a browser receives the information required to validate all the certificates that could be used during a session with the web application, including certificates from other domains. As a result, to execute a MITM attack, an adversary not only needs to compromise a CA but also each targeted web domain. A DVCert transaction uses a modified Password Authenticated Key Exchange (PAKE) protocol known as PAK. However, we are not simply applying a known protocol; rather, we modified PAK to provide *only* server authentication and integrity protection instead of mutual authentication and generation of encryption keys (i.e., traditional use of PAKE protocols). These changes

allow better performance and simplify deployment without affecting PAK's formal security proofs. Our experimental evaluation shows that an optimized DVCert transaction requires little computation time on the server (e.g., < 1 ms) and on the browser. More importantly, DVCert transactions are executed at most once per session; thus, their impact on server performance or user experience is negligible. DVCert's design also provides multiple advantages over third-party solutions: simpler trust model, lower deployment and operational costs (e.g., no additional infrastructure is required) and no privacy risks. Finally, DVCert is a readily available mechanism designed to improve the current CA trust model and be compatible with third-party solutions such as DNSSEC, once these solutions are deployed in the future. In so doing, we make the following contributions:

- **Designing and implementing an efficient and easy to deploy mechanism to detect MITM attacks against SSL/TLS without third-parties:** We identify key properties required to achieve a robust and practical defense against MITM attacks. Based on these properties, we develop a protocol that provides more robust certificate validation and detects MITM attacks, even if the adversary uses forged certificates. By allowing web applications to attest directly for their certificates, our mechanism avoids many of the challenges hindering the deployment of third-party solutions. We implemented a proof-of-concept extension for Firefox and Firefox for mobile browsers and a PHP-based server component to demonstrate the deployability of our solution.
- **Conducting an extensive performance analysis in multiple platforms:** We characterize DVCert's performance using our prototype implementation in both desktop and mobile browsers. Our results show that an optimized DVCert transaction requires 0.54 ms of computation time on the server and 12.03 and 97.70 ms on a laptop and on a smartphone respectively. Compared to a naive implementation, these results represent a 94.96%, 55.07% and 77.82% improvement on the server, laptop and smartphone correspondingly. Moreover, our experimental evaluation demonstrates that DVCert transactions are as efficient as existing server operations (e.g., processing HTTPS requests). Thus, given their low frequency, DVCert transactions are unlikely to degrade server performance or scalability.
- **Making our DVCert implementation available to the community:** The DVCert extension for Firefox and Firefox for mobile as well as the server PHP code are available for evaluation at: http://www.cc.gatech.edu/~idacosta/dvcert/index.html.

References

[1] C. Adams and M. Just. PKI: Ten Years Later. In *PKI R&D Workshop*, 2004.

[2] P. Eckersley. How secure is HTTPS today? How often is it attacked?, 2011.

[3] P. Eckersley and J. Burns. The (Decentralized) SSL Observatory. In *USENIX Security Symposium (Invited Talk)*, 2011.

[4] Electronic Frontier Foundation (EFF). The Sovereign Keys Project, 2011.

[5] C. Ellison and B. Schneier. Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure. *Computer Security Journal*, 16(1):1–7, 2000.

[6] GlobalSign. Security Incident Report, 2011.

[7] D. Goodin. Tunisia Plants Country-Wide Keystroke Logger on Facebook, 2011.

[8] D. Goodin. Web Authentication Authority Suffers Security Breach, 2011.

[9] P. Gutman. PKI: It's Not Dead, Just Resting. *Computer*, 35(8):41–49, 2002.

[10] P. Hoffman and J. Schlyter. IETF Internet-Draft: Using Secure DNS to Associate Certificates with Domain Names For TLS (draft-ietf-dane-protocol-06), 2011.

[11] G. Keizer. Hackers May Have Stolen Over 200 SSL Certificates, 2011.

[12] J. Kirk. KPN Stops Issuing SSL Certificates After Possible Breach, 2011.

[13] B. Laurie and A. Langley. Certificate Authority Transparency and Auditability, 2011.

[14] J. Leyden. Inside 'Operation Black Tulip': DigiNotar Hack Analysed, 2011.

[15] J. Leyden. Trustwave Admits Crafting SSL Snooping Certificate, 2012.

[16] M. Marlinspike. Convergence, 2011.

[17] J. Menn. Key Internet Operator VeriSign Hit by Hackers, 2012.

[18] R. Richmond. An Attack Sheds Light on Internet Security Holes, 2011.

[19] I. Ristic. Internet SSL Survey 2010, 2010.

[20] R. Singel. Law Enforcement Appliance Subverts SSL, 2010.

[21] C. Soghoian and S. Stamm. Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL. In *Proceedings of Financial Cryptography and Data Security*, 2011.

[22] D. Wendlandt, D. G. Andersen, and A. Perrig. Perspectives: Improving SSH-style Host Authentication with Multi-path Probing. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2008.