<https://www.eff.org/observatory>

<https://www.eff.org/files/DefconSSLiverse.pdf>

<https://www.eff.org/files/ccc2010.pdf>

\begin{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. By maintaining a user created list, we can abate the risk of trusting less ubiquitous CAs. After developing the plugin we sample the users using our 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.

\end{abstract}

\section{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.\cite{ellison2000ten} Our current Oligarchy PKI model leverages the use of trust anchors, where a certificate issued by any one of the trust anchors is accepted.\cite{kaufman2002network} 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 browser’s trusted CA list, the website’s 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.\cite{kaufman2002network} 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 browser’s 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, Mozilla’s Firefox browser stores 124 trusted CAs. Microsoft’s built in Windows CA database, used by Google’s 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. \cite{SSLobservatory} 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 Microsoft’s default CA directories.

Allowing for the user to have a more hands-on experience with their own browser security settings is also advantageous. Where most web browser security