1. Certificate Exception Attack on Firefox

Trust is an important concept in establishing the public key infrastructure. Generally, a site wishing to identify itself as secure under the HTTPS protocol is required to use a digital certificate which is issued by a trusted certificate authority, such as Verisign. The certificate authority will ensure that the site's identity is tied to a specific owner, and issue a digital certificate corresponding to the site's details, containing an expiry date and other details relating to the site. Within Firefox it is possible for a user to manually specify a certificate exception via the user interface provided, if a site uses an untrusted or erroneous certificate.

URL of the site, including the port	Hash Algorithm OID (Commonly SHA1-256)
[redacted]	OID.2.16.840.1.101.3.4.2.1
Certificate Fingerprint using Hash Algorithm	
58:65:35:27:54:1E:23:91:C7:F1:78:97:A2:A8:FE:90:2B:E8:1E:7A:9F:58:64:72:BC:97:64:F0:C5:85:1B:EE	
Exception Flags M - allow mismatches in the hostname U - allow untrusted certificates T - allow expired certificates	**Base 64 Encoded Description**
MUT	Not needed in attack

Figure 1: *cert_override.txt* entry format, along with example entries. (The URL of the site is removed to preserve anonymity in this submission.)

Our proposed attack removes this element of interaction from the victim, and makes it so that certificate exceptions can be added silently without the victim being privy to this knowledge. Firefox insecurely stores certificate exceptions in a plain-text file located in the user profile known as *cert_override.txt*. The user profile can be trivially read to and written to by an extension, with no additional permissions needed. The file is organised in a known format illustrated in Figure 1, with each item delimited by a space character.

Each entry block is delimited by a linefeed character, and multiple entries can be appended to the file. It was found possible for an attacker to remotely send the certificate, in a format such as base64, to the extension along with the URL of the target site and the extension would silently add the exception as a line in *cert_override.txt*. After the victim closed and re-opened their browser, they could navigate to the site without receiving the untrusted certificate warning, and use it as a regular trusted HTTPS website. It was found that expired and untrusted (such as self-signed) certificates, along with certificates containing URL mismatches could be silently marked as trusted using this attack. Depicted in Figure2 is the result of a victim browsing a site that uses an untrusted certificate.

This attack has several implications. Primarily, if a certificate exception is added, then the attack basis can be

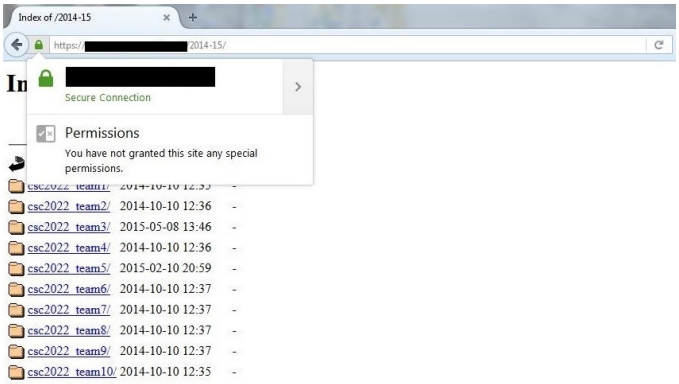

Figure 2: The connection now appears as secure when browsing a site using an untrusted certificate. (The web address is removed to preserve anonymity in this submission.)

leveraged to perform man-in-the-middle attacks by utilising a spoofed server [1] using techniques such as DNS poisoning to make it seem to the target that they are connecting to a legitimate authority. This is severe as the man-in-the-middle attack results in the target's secure channel being compromised and decrypted. This is because the self-signed certificate added as a certificate exception contains the attacker's public key, rather than the legitimate authority that they wish to connect to. Other implications include allowing certificates using outdated certificates vulnerable to known attacks, such as [2] and allowing sub-domain mismatches to occur, which can be a problem in shared domains.

References

[1] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the https protocol," *IEEE Security and Privacy*, vol. 7, no. 1, pp. 78–81, 2009.

[2] A. K. Lenstra, X. Wang, and B. de Weger, "Colliding x. 509 certificates," Tech. Rep., 2005.