Public Key Protocols
Epilogue

Simon Foley

December 4, 2012

Hacking the Secure Web

Early browsers (for US export) permitted only a 40-bit session key, making a brute force attack on the key possible.

An SSL-authenticated connection from your web-browser to a web-server does not secure the transaction, it only secures the connection/session;

It is typically the credit card that secures a web-purchase.

While the browser will authenticate an https connection to a website, the user must confirm by manual inspection that the website/certificate is indeed the business/etc that they expect.

- Earlier certificates/browsers did not necessarily include/validate the website address.
- It is a user's responsibility to confirm that the URL is as expected. An attacker (with website `www.amason.com`) could obtain an X509 certificate for their public key. Any connection to this (malicious) website will be authentic and secure.

Who decides who should be the trustworthy CAs? Microsoft, Mozilla?

What's this website?

Semantic Attacks on URL by subtly changing it meaning without an (easily) apparent change in syntax.
Plain text attacks

https://*www.amason.com*
http://*www.amazon.com@143.231.211.12/malicious*
https://*www.amazon.com@143.231.211.12/malicious*

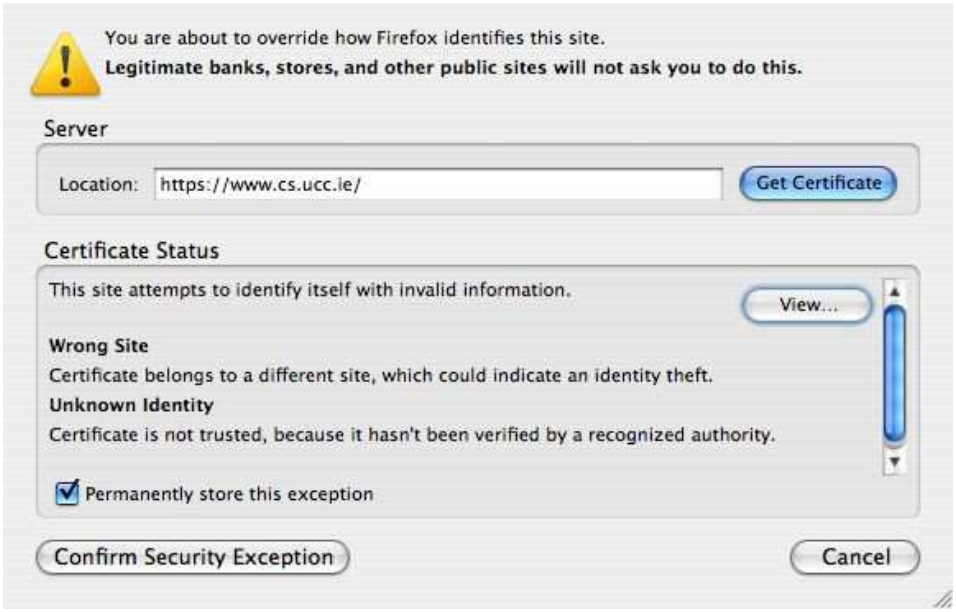
Some browsers will warn you that you are attempting to login to a website.
Simple HTML attacks

<A HREF=https://*www.malicious.com*> *https://www.amazon.com* </A>
<A HREF=http://*www.amazon.com*> *https://www.amazon.com* </A>

It is the user's responsibility to know the website to which they are connecting.

It is the user's responsibility to know whether the connection they are making is secure or not.

Yes, Yes, Yes, I accept!



MD5 Considered Harmful Today

Recall that it is possible to find collisions in MD5, that is, it is possible to find values x and x' such that $md5(x) = md5(x')$.

This has a serious implication for public key certificates that use MD5 to implement their signatures. Recall that

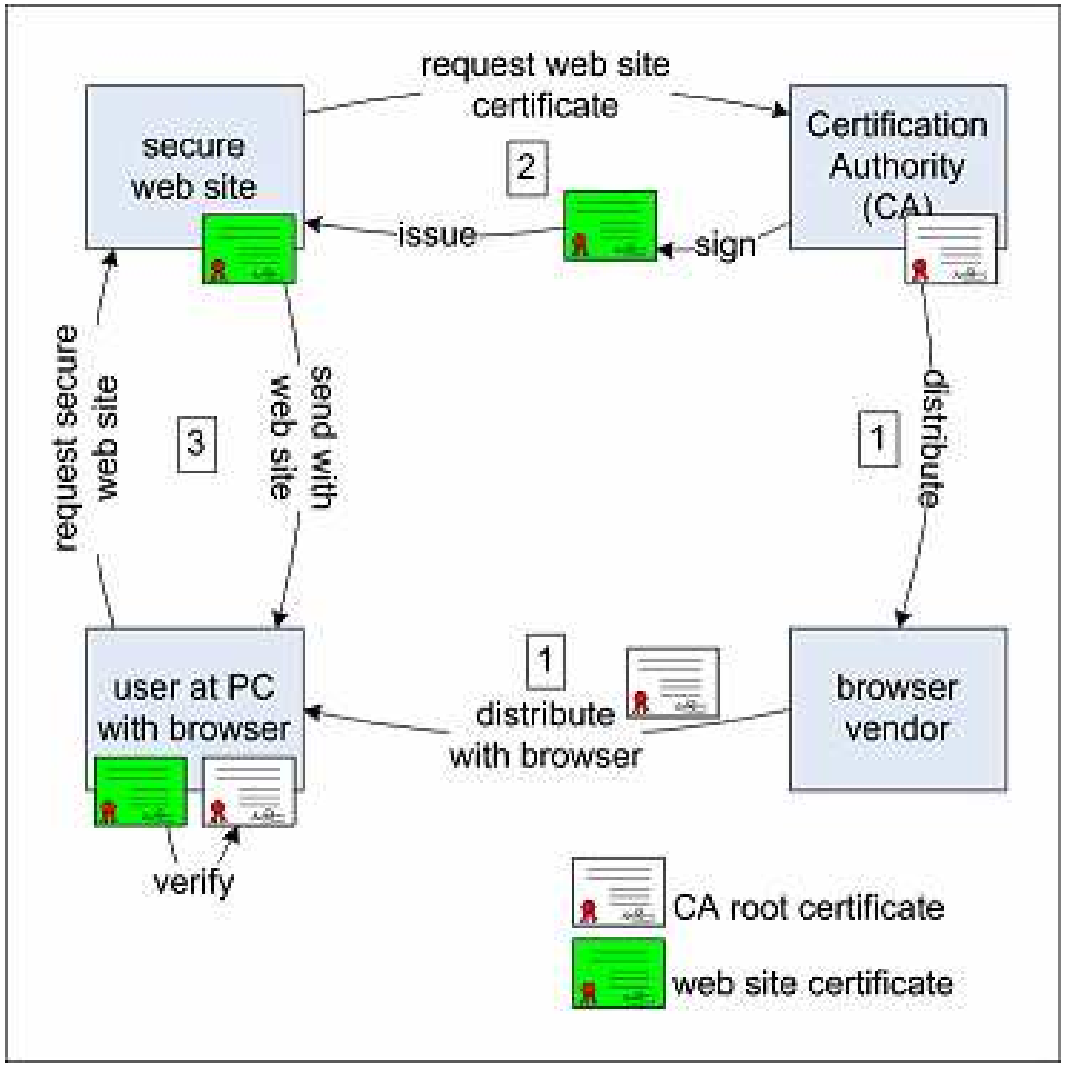
$$\{K_A, Alice, ...\}_{sK_B} \approx (K_A, Alice...), \{h(K_A, Alice, ...)\}_{K_B^{-1}}$$

If MD5 is used to implement $h()$ then it means that we could have two different certificates with the same signature!

Furthermore, with MD5 it is possible, given $\{K_A, Alice, ...\}_{sK_B}$ to forge another certificate $\{K'_A, Alice, ...\}_{sK_B}$ that has the same signature value as the original certificate!

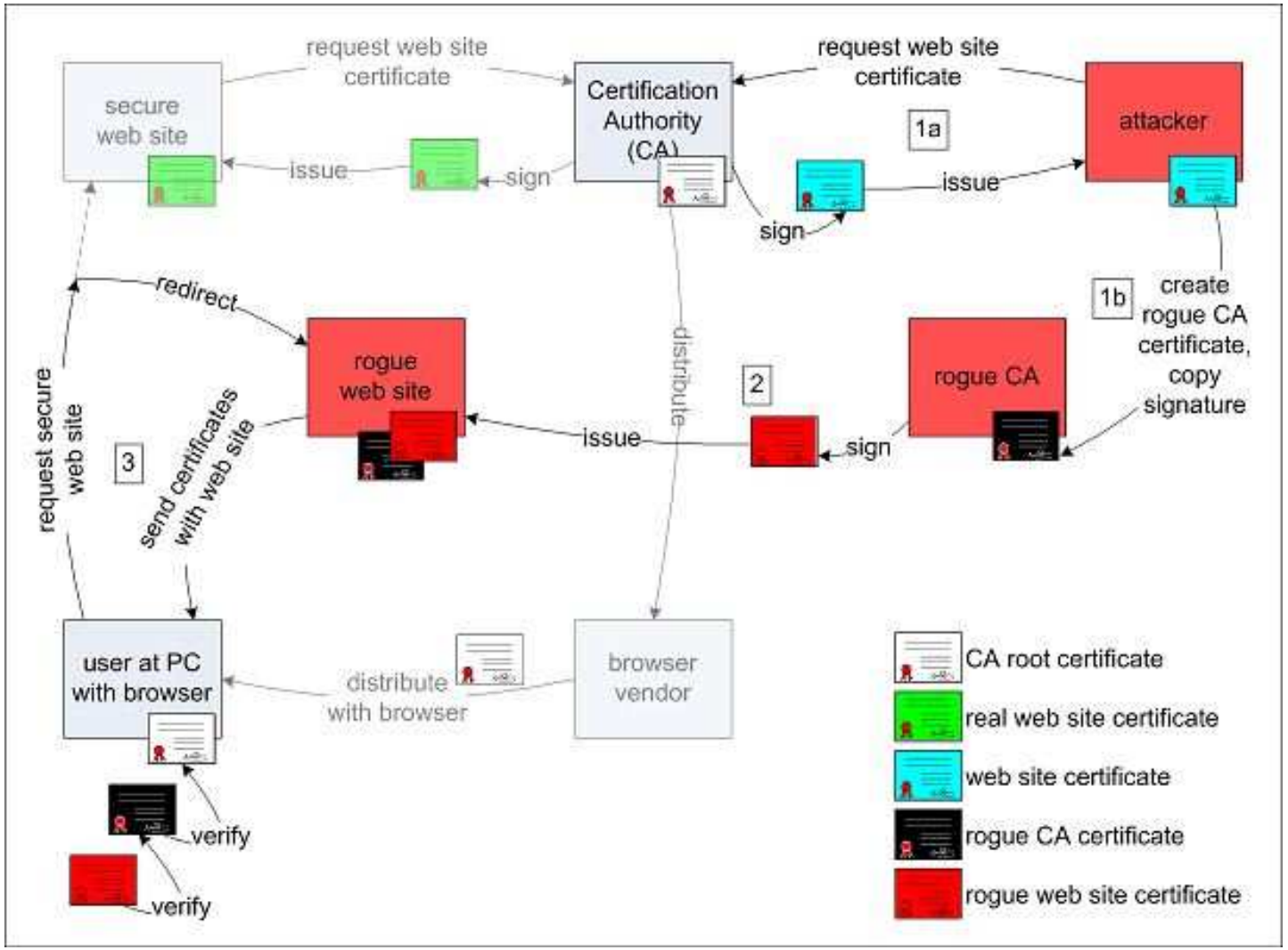
This vulnerability was used to demonstrate a weakness in securing WWW when MD5-based certificates are available.

MD5 Considered Harmful Today



From `http://www.win.tue.nl/hashclash/rogue-ca/`

MD5 Considered Harmful Today (2008)



From `http://www.win.tue.nl/hashclash/rogue-ca/`

MD5 Considered Harmful Today

1a A legitimate website certificate is obtained from a commercial CA.

1b A rogue CA certificate is constructed. It bears exactly the same signature as the website certificate. Thus it appears as being issued by the CA, whereas in fact the CA has never even seen it.

2 Then a website certificate bearing the genuine website's identity but another public key is created and signed by the rogue CA. A copy of the genuine website is built, put on another web server, and equipped with the rogue website certificate.

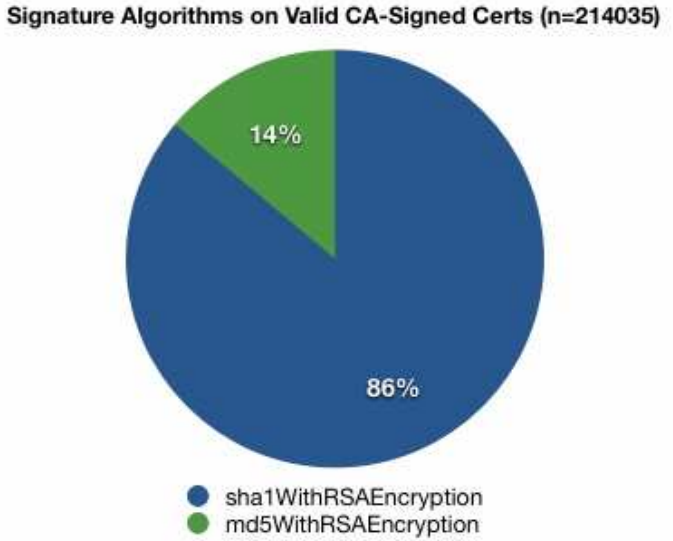
3 When a user wants to visit the secure website, the web browser will look on the Internet for the genuine web server. There exist "redirection attacks", by which the communication from the browser can be redirected to the rogue website. This rogue website presents its certificate to the user, together with the rogue CA certificate. The signature in the rogue website certificate can be verified with the rogue CA certificate, and this rogue CA certificate in turn will be accepted by the browser, as its signature can be verified with the CA root certificate in the trust list. The user will not notice anything.

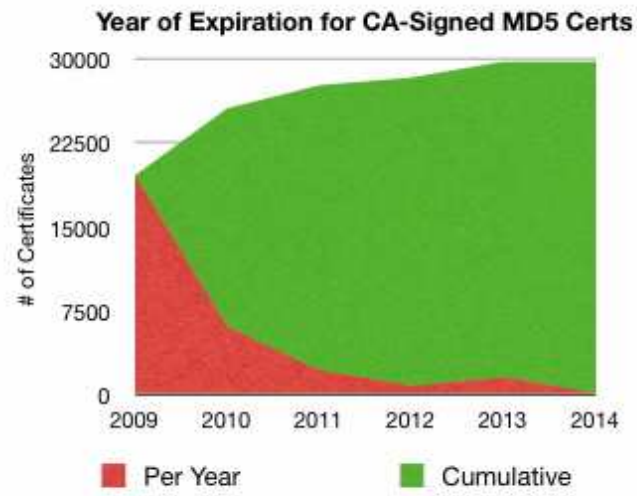
Numbers of MD5 based certificates

From

`http://blog.johnath.com/2009/01/21/ssl-information-wants-to-be-free/`

Crawled the top 1 million HTTPS sites [Jan 2009]





Ignoring Security Warnings

Browser	Understood	Expired Certificate		Ignored		Unknown CA		Ignored		Domain Mismatch		Ignored	
FF2	Y	48	50%	71%		37	39%	43%		57	59%	19%	$\chi^2_2 = 9.40$
	N	48	50%	56%		59	61%	49%		39	41%	49%	$p < 0.009$
FF3	Y	55	47%	64%	$\chi^2_2 = 21.05$	35	30%	31%		46	39%	15%	$\chi^2_2 = 8.65$
	N	62	53%	34%	$p < 0.0005$	82	70%	34%		71	61%	41%	$p < 0.013$
IE7	Y	45	23%	53%	$\chi^2_2 = 11.81$	44	22%	27%		62	32%	16%	$\chi^2_2 = 7.50$
	N	151	77%	32%	$p < 0.003$	152	78%	32%		134	68%	35%	$p < 0.024$

Table 1: Participants from each condition who could correctly identify each warning, and of those, how many said they would continue to the website. Differences in comprehension within each browser condition were statistically significant (FF2: $Q_2 = 10.945$, $p < 0.004$; FF3: $Q_2 = 11.358$, $p < 0.003$; IE7: $Q_2 = 9.903$, $p < 0.007$). For each browser condition, the first line depicts the respondents who could correctly define the warnings, while the second depicts those who could not. There were no statistically significant differences between correctly understanding the unknown CA warning and whether they chose to ignore it.

[From J. Sunshine et al, *Crying Wolf: An Empirical Study of SSL Warning Effectiveness*, 18th USENIX Security Symposium, 2008]

Trust on First Use (TOFU)

Its not unusual for a website to have a self-signed certificate, rather than having to pay a CA to sign their public key.

The user-strategy is that the first they visit the site they decide how likely is there to be a man-in-the-middle attacker between the website and their browser. If they decide the likelihood is low then they accept the certificate and trust on its first use that it comes from the legitimate website (and not a man in the middle). All subsequent use is as secure as the first use, regardless of the presence of an man-in-the-middle.

For example, for a wired connection on the UCC backbone there is a low-chance of a man in the middle attacker and the user accepts (and stores) the https://www.cs.ucc.ie certificate. This (stored) certificate is used when the user visits https://www.cs.ucc.ie from outside the ucc domain, where the chance of an attacker is higher.

ssh can rely on a similar TOFU strategy.

Example: ssh TOFU

machine:~/.ssh simon$ ssh cosmos.ucc.ie
The authenticity of host 'cosmos.ucc.ie (143.239.159.63)' can't be established.
RSA key fingerprint is 10:2c:a2:1f:5d:9e:c4:b1:1d:12:e1:66:dc:ac:49:e1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'cosmos.ucc.ie,143.239.159.63' (RSA)
to the list of known hosts.
Password:
Last login : Mon Dec 1 14:21:10 2008 from machine.ucc.ie
##########################
Welcome to cosmos
Note: sftp is available NOT ftp

##########################
Directory : /home/simon
Mon Dec 1 14:22:49 GMT 2008
cosmos /home/simon>

Example: ssh TOFU, saved public key

machine:~ simon$ **more .ssh/known_hosts"|**
cosmos.ucc.ie,143.239.159.63␣ssh−rsa␣AAAAB3NzaC1yc2EAAAABIwAAAIEAsZ1
PWZELHoW2qEX6WraLPxzZVrN/iHlfz/zKkvRdoFy+UYXsI0OccmMvOevcAbFm
aDFWufU1OtrtvC81E8IQIHsQEjQ/5lvY08rnvQ7Vt/kwbfPym00DlikhsfP6LIRa2Qn

Example: ssh TOFU, next login


```
machine:~ simon$ ssh cosmos.ucc.ie
Password:
Last login: Mon Dec 1 14:22:49 2008 from machine.ucc.ie
##############################################
### Welcome to cosmos #########
Note: sftp is available NOT ftp
----------------------------------------------
##############################################
Directory: /home/simon
Wed Dec 3 12:39:40 GMT 2008
cosmos /home/simon>
```

Example: ssh TOFU, change cosmos PK


```
machine:~ simon$ emacs .ssh/known_hosts
 ...
machine:~ simon$ ssh cosmos.ucc.ie
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
10:2c:a2:1f:5d:9e:c4:b1:1d:12:e1:66:dc:ac:49:e1.
Please contact your system administrator.
Add correct host key in /Users/simon/.ssh/known_hosts to get rid of this message.
Offending key in /Users/simon/.ssh/known_hosts:1
RSA host key for cosmos.ucc.ie has changed and you have requested strict checking
Host key verification failed.
machine:~ simon$
```

Example: ssh TOFU, strong client authentication


```
machine:~ simon$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/simon/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/simon/.ssh/id_rsa.
Your public key has been saved in /Users/simon/.ssh/id_rsa.pub.
The key fingerprint is:
79:c6:49:24:9b:6a:d7:31:b6:ef:67:0e:e2:bc:c7:b9 simon@machine.ucc.ie
machine:~ simon$
 ..... then copy d_rsa.pub to .ssh/authorized_keys on cosmos ....
machine:~ ssh cosmos.ucc.ie
Last login: Wed Jan 14 12:32:17 2009 from machine.ucc.ie
### Welcome to cosmos #########
Note: sftp is available NOT ftp
------------------------------
##############################
Directory: /home/simon
```

Perspectives: Strengthening TOFU

A plug-in for Firefox that strengthens TOFU of self-signed certificates.

Network Notary servers maintain databases of self-signed keys that have been accepted by users in the past.

On presentation of a self-signed certificate, the plug-in consults with the network servers in order to confirm the certificate and whether the public key has appeared to have changed recently.

See `http://perspectives-project.org/` for overview and plugin.