

# Untrusted Root Certificates Considered Harmful\*

Executive Summary .....	2
Root Certificates .....	3
Trusted root certificates .....	3
Why trusted root certificates are valuable .....	3
How are trusted root certificates included in browser certificate stores? .....	4
Untrusted certificates as alternatives to trusted root certificates .....	5
When untrusted root certificates may be acceptable .....	5
Browsers treat all installed certificates equally .....	5
People do not understand computer issues .....	6
Dealing with adversaries .....	6
Untrusted root certificates are difficult to verify .....	7
Dangers of installing untrusted root certificates .....	10
Hackers can exploit untrusted root certificates .....	10
Getting a root certificate installed .....	10
Taking advantage of an installed certificate .....	11
Legitimate businesses can run into trouble using untrusted SSL certificates too .....	11
Can this really happen? .....	12
Conclusion .....	13
Endnotes .....	14

Copyright © 2005 by Sericon Technology Inc.

\* The title is inspired by Edsger Dijkstra's seminal Computer Science paper "*Go To Statement Considered Harmful.*"



# Executive Summary

An untrusted root certificate is one that is not shipped inside the certificate stores of common Web browsers. Verifying the authenticity of untrusted root certificates is well beyond the capabilities of most computer users.

Most computer users can be persuaded to ignore messages about untrusted certificates and to install these certificates. There are even programmatic ways to install untrusted certificates without the user's consent.

If legitimate businesses encourage users to install untrusted certificates:

- Businesses assume the responsibilities of an SSL certificate authority (CA) and storing the root certificate securely, which they may not be able to do effectively.
- Users are more likely to be persuaded by criminal elements to install untrusted certificates.

Once a certificate from a criminal element is installed, the user is in jeopardy for the following reasons:

- If an untrusted root certificate is installed into the certificate store of a browser, the browser treats it the same way as any other root certificate. This is dangerous, because unsuspecting users will not receive any security warnings.
- Any harm resulting from this security breach may not be immediate (the computer will continue to operate normally) but there may be theft of personal information, such as credit card numbers and identity theft.
- Any attack made on a computer by this means is very hard to detect, even for very experienced users. Furthermore, there is no evidence left on the victim's computer to help prove that an attack really occurred.

Untrusted root certificates are harmful to legitimate businesses and to end-users. Therefore, legitimate businesses should avoid using them anywhere accessible by the public.

# Root Certificates

A root certificate is a certificate that a certificate authority (CA) can use to sign SSL certificates. Before signing a certificate, the CA *vet*s the prospective certificate to positively confirm:

- the identity of the applicant through outside means (such as examining business documents)
- the applicant's authority to own this certificate

If the vetting process confirms the applicant's identity, the CA *signs* the certificate and adds its identity to the Issuer field. By signing a certificate, the CA uses its private key to encrypt information into the certificate so that someone who examines it will be assured that that CA validated the certificate's information. Because the signing process requires possession of the CA's private key, which is highly guarded, it is not possible for someone to forge this signature. If a certificate is signed by one of these CAs in error, or becomes compromised after signing, these CAs have the necessary infrastructure to revoke it in a timely fashion. For additional information on the SSL process, please refer to the Sericon Technology white paper, "Introduction to SSL."<sup>1</sup>

## Trusted root certificates

A *trusted* root certificate is a root certificate that comes from a known, well-respected and trusted CA. Well-known CA companies that control trusted root certificates include:

- VeriSign
- Thawte
- GeoTrust
- RapidSSL
- Comodo
- Starfield Technologies

An SSL certificate that can be traced back to a root certificate from one of these CAs is typically considered trustworthy, i.e. very safe and secure, because these companies know that the integrity of their root certificate is crucial to their business, and they act accordingly. If a CA's trusted root certificate were compromised (i.e. have its private key fall into untrusted hands), then seemingly trusted certificates for any domain could be signed at will by the holder of the key. This would cause a crisis on the Internet that would certainly reach the front pages of newspapers, and create (at least temporarily) concern for any resource protected by SSL certificates (e.g. e-commerce, e-banking). Such an event would make it very difficult for the CA that was breached to carry on business in the future.

## Why trusted root certificates are valuable

Trusted root certificates are shipped with browsers, inside their certificate stores. When a browser encounters a site secured by an SSL certificate that is traceable back to a trusted root certificate, it implicitly accepts the certificate's authenticity. In contrast, if the browser encounters a root certificate it does not recognize, it displays a security alert such as the following, stating that the certificate was issued by an authority that is not trusted.

**Figure 1: Microsoft Internet Explorer displays this Security Alert upon encountering an untrusted certificate**



## How are trusted root certificates included in browser certificate stores?

Companies with a root certificate included in the certificate stores of popular browsers clearly have a very valuable asset. These trusted CA companies can charge e-commerce sites, which want to have their SSL certificates verified and signed by them, at least \$100 for this service. Therefore, we might expect that Microsoft would charge a lot of money to include a certificate in its Internet Explorer browser. However, this service is actually provided without cost<sup>2</sup> to companies that meet its requirements: Microsoft requires a CA to comply with the WebTrust for Certification Authorities program sponsored by the American Institute for Certified Public Accountants (AICPA)<sup>3</sup> before including the CA's root certificate in its Internet Explorer browser.

The WebTrust program is a very comprehensive and demanding set of rules and procedures that define how a "trusted" CA should act. It governs issues such as how root certificates are generated, stored and accessed, how the physical premises are guarded, how checks are done on internal employees who must access secure information, and a long list of other issues. A company that wants to enter the CA business must first create the necessary infrastructure, including defining and implementing procedures for all these aspects of its business, and it must prove that it follows these procedures when the AICPA audits it on an ongoing basis.

Passing the Web Trust program and having a root certificate included in the certificate stores of major browsers is not immediately meaningful to a company, since users are usually very slow to install new software unless they have a compelling reason to do so. As a result, a significant segment of the market still runs old operating systems such as Windows 98 or even Windows 95, and these people are probably not motivated to install a new browser, especially if the one they have already works. This means any certificate signed by a relatively new root is seen as untrusted by older browsers, which display security warnings when they encounter sites protected by it. Therefore, most companies do not want to do business with a new and relatively untested CA until the browsers whose certificate stores contain this new root certificate are installed in nearly all computers used to browse the Internet. This means that the CA business is somewhat static, as entrance into it is a long and costly process.

## Untrusted certificates as alternatives to trusted root certificates

Any root certificate that is not trusted is *untrusted*. Anyone who uses the appropriate software, such as OpenSSL<sup>4</sup> and OpenCA<sup>5</sup> (both freely downloadable), can be a certificate authority, create a root certificate, and generate and sign additional certificates. However, being a CA is not the same as being a trusted CA, any more than having a pair of pilot's wings makes you a jet pilot. It is relatively easy to obtain a pair of pilot's wings (you can buy toy wings for less than a dollar), just as it is easy to acquire CA software. However, a *licensed* pilot has the knowledge and experience required to fly an airplane safely. The pilot's training process creates public trust, and the public expects that only licensed pilots will fly airplanes: few people would choose to fly in an airplane flown by an uncertified pilot. Similarly, browsers, and by extension their users, trust CAs who have demonstrated that they are capable of performing the required tasks responsibly, and users should be wary of anything less.

### When untrusted root certificates may be acceptable

Some CAs may be trusted, but in only a very limited way. For example, a company with employees in diverse locations can make internal documents available to all its employees by setting up a Web site on an intranet that is only accessible from inside the corporate LAN (i.e. people on the Internet cannot see it). If there are documents on this site that should have limited access within the company (such as strategic plans or personnel documents), then these can be protected with SSL.

Since both the servers as well as the browsers are on corporate-controlled equipment, it is well within the company's interests to act as its own CA. This means that the company can generate its own root certificate with which it can sign as many SSL certificates as required for the servers deployed in its intranet. Once this is done, this certificate should be installed into the certificate stores of all the browsers used in the company. Since the computers these browsers run on are controlled by the company, this is easy to do: the corporate IT department can have a policy that the company's root certificate is installed in the browser's certificate store whenever a new computer is set up. This prevents security warnings from being displayed whenever an employee accesses an SSL-secured site on the company intranet.

The advantage to the company is that it can deploy secured sites anywhere on its intranet without purchasing certificates from an external CA. Note that if the company also runs an e-commerce site, then it should purchase its SSL certificate from a trusted CA and not use an internal one for sites accessible to the public, who will not have the certificate installed by the corporate IT department, and thus would receive a security warning.

In such an environment, an unscrupulous employee (most likely a member of the IT team) who has access to the private key could launch very successful MITM attacks against employees who visit SSL-protected e-commerce and e-banking sites at work. This will be discussed later in this document. However, the company can easily protect itself by warning employees not to visit such sites on company time or equipment, since they are not "business related activities."

### Browsers treat all installed certificates equally

Once a root certificate is installed into the certificate store in the browser, the browser treats it the same way that it treats the certificates shipped with it. This means the trusted/untrusted distinction of root certificates described above is not important to the browser. For example, suppose a user installs a certificate signed by an untrusted root certificate into its browser's certificate store. When the browser attempts to connect to a site protected by this certificate, then as long as the dates and names for the certificate are valid, the browser does not display any security messages. The browser displays the site in the same way that it displays a site secured by a certificate signed by VeriSign: for example, in Microsoft Internet Explorer, the padlock in the bottom right corner closes, and the address bar contains the true URL of the resource that is requested.

Displaying sites protected by installed untrusted certificates and by trusted certificates in the same way is dangerous, because it undermines the user's belief that the padlock indicates an encrypted and private session with a remote Web server. As we will see later, an SSL-encrypted session based on an untrusted certificate may not always be encrypted and private.

# People do not understand computer issues

It is widely accepted that the human element is the weakest link in the field of computer security: according to a humorous BBC study, more than 70% of people interviewed were willing to reveal their computer's password in exchange for a chocolate bar.<sup>6</sup> While this is funny, it is also disturbing: theoretically, these are the same people we are protecting. Therefore, it is not surprising that a study by Gartner Research found that almost two million users gave personal information to spoofed Web sites, which cost U.S. banks and credit card issuers an estimated \$1.2B in direct losses in 2003.<sup>7</sup>

There has been much discussion about why the average user is so naïve and incompetent. In general, many users find computers so difficult to use because computers *are* hard to use. Both hardware and software are generally designed by people who would rather work on advanced features than on usability. Often, a product's scheduling and marketing constraints prioritize "getting it out the door" over "getting it right."

Nowhere is this truer than computer security: as users, we are expected to know how to evaluate the security of our own computers, how to install and monitor antivirus software, set up a firewall to protect our computer, and apply the latest security patches to operating systems that fix the latest security scare reported in the media. We do this because we are warned that hackers are lurking around every corner, waiting to destroy our computers or steal our identities. On top of this, we must remember a long list of passwords for each computer account we have, and we are warned to avoid both easily guessed words and writing down those passwords. It is no wonder that many people shut off their computers and give up.

The state of SSL is just as bad. Herzberg and Gbara report on a study they did to determine the average user's understanding of SSL, and what should and should not be considered safe.<sup>8</sup> The study was conducted among participants of the Mexican International Conference in Computer Science 2004 (ENC-04) – people who should understand computer security concepts better than the average computer user. They showed 42 people a series of browser images – some legitimate, and some "spoofed". They asked people whether they could differentiate between the secure and the fake sites by looking at elements such as the browser lock status and the address bar. While the participants clearly were aware of how to determine whether a site was secure, more than two thirds of them were effectively unable to do this.

With this in mind, it is crucial to make computer security as user-friendly as possible. If people attending a computer conference cannot easily distinguish between secure and faked sites, then how can the typical user?

## Dealing with adversaries

Hackers, phishers, and other adversaries make it even more difficult for computer users. In his book, "The Art of Deception: Controlling the Human Element of Security,"<sup>9</sup> notorious hacker Kevin Mitnick describes many techniques that he pioneered or used to trick people out of enough information to allow him to access sensitive information. Mitnick popularized the idea of *social engineering*, which is the process of deceiving people into sharing valuable information without their knowledge. For example, an attacker posing as a system administrator at a company could contact an employee there on the pretext of upgrading software or fixing computer problems, and ask for a password. This is easier than it sounds as most people are painfully unaware of the value of small pieces of information that they control, and how a hacker can use these to gain access to more valuable information.

Today, adversaries continuously use many of these techniques and others, such as spam, phishing attempts, telemarketing phone calls, and viruses to try to gain access to valuable secrets. Never before have people had to be as vigilant as they must be today to protect themselves from these attacks.

Responsible companies are well aware that typical users are under pressure to differentiate between fraudulent attacks and legitimate messages from authentic businesses. For example, many phishers send out e-mail messages that purport to be from real banks or e-commerce sites and try to persuade users to click on links in the e-mail message to update personal information. Legitimate banks are then compelled to periodically send reminders to their customers that any such messages are always fraudulent, because they would never send out such messages.

Now suppose that a legitimate bank began sending e-mail messages to customers asking them to click on links in the message to update personal information. This would result in bank customers becoming very confused about what is appropriate behavior from a legitimate business, which might result in them perceiving similar fraudulent messages as legitimate, and falling prey to them. Therefore, at all costs, legitimate businesses must avoid acting like phishers and hackers. Similarly, legitimate businesses should avoid using untrusted root certificates in the public domain, which could confuse their customers.

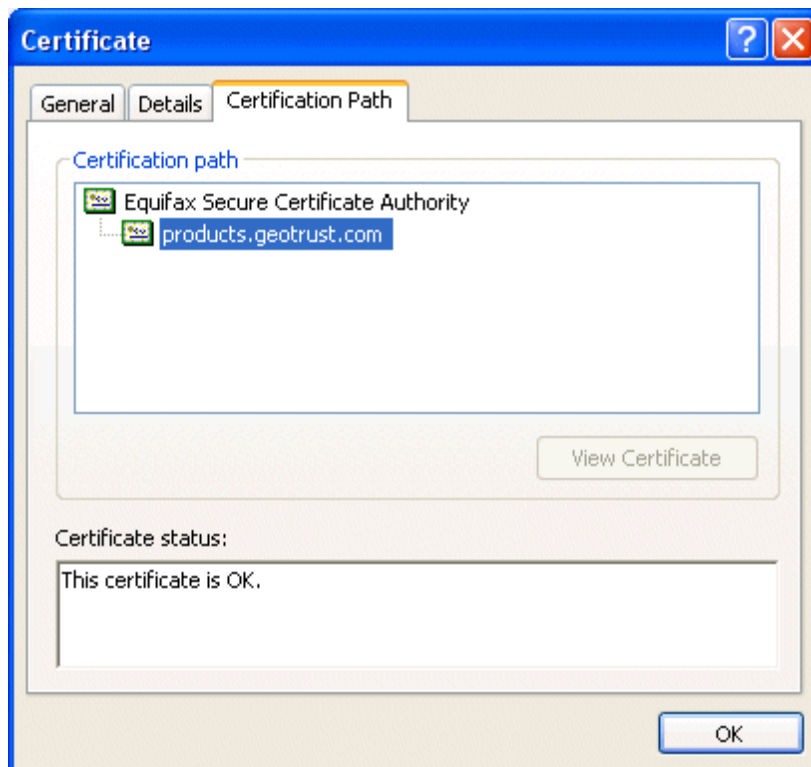
## Untrusted root certificates are difficult to verify

This leads us to the issue of exactly how users should behave when confronted with the security alert displayed in Figure 1, while using the Microsoft Internet Explorer browser, which states that the security certificate was issued by an untrusted company.

Unfortunately, the appropriate behavior is rather complex. Because it is so easy to create certificates with any name on them, it is crucial to be able to verify that a certificate is genuine and has not been created by a hacker. A user *should* follow the steps below when confronted with an untrusted root certificate. (Note that these instructions are for Microsoft Internet Explorer. Other browsers have similar functionality.)

1. To view a certificate from an untrusted CA, click the **View Certificate** button from the dialog box in Figure 1, above.
2. Click the **Certification Path** tab to see the entire certificate chain to which this certificate belongs. In this case, there are only two certificates in the chain: the SSL certificate belonging to `products.geotrust.com`, and the root certificate that signed it, belonging to *Equifax Secure Certificate Authority*.

**Figure 2: Viewing the certificate chain for a certificate**

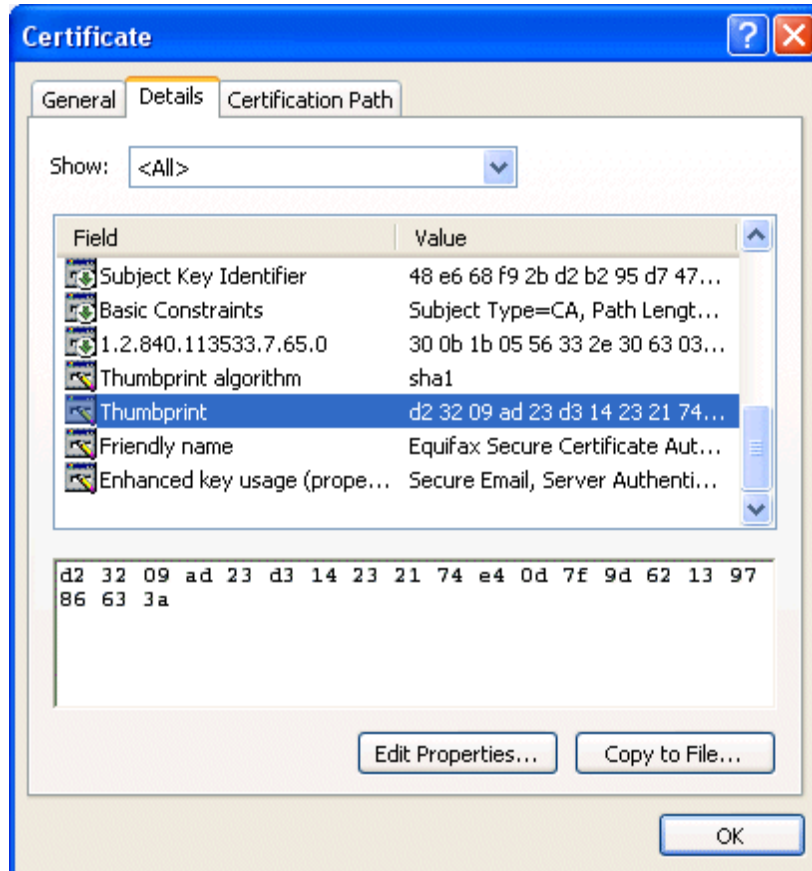


The information in this dialog box describes the `products.geotrust.com` SSL certificate.

**Note:** This example actually uses a trusted certificate, which is installed in the Microsoft Internet Explorer certificate store by default. Therefore, the browser does not really display a security alert for this root certificate; we are simply using this certificate to illustrate the verification process.

- To determine whether the root certificate is authentic, select the root certificate (in this case, Equifax) and click **View Certificate**. Next click the **Details** tab.

**Figure 3: Viewing a certificate's thumbprint**

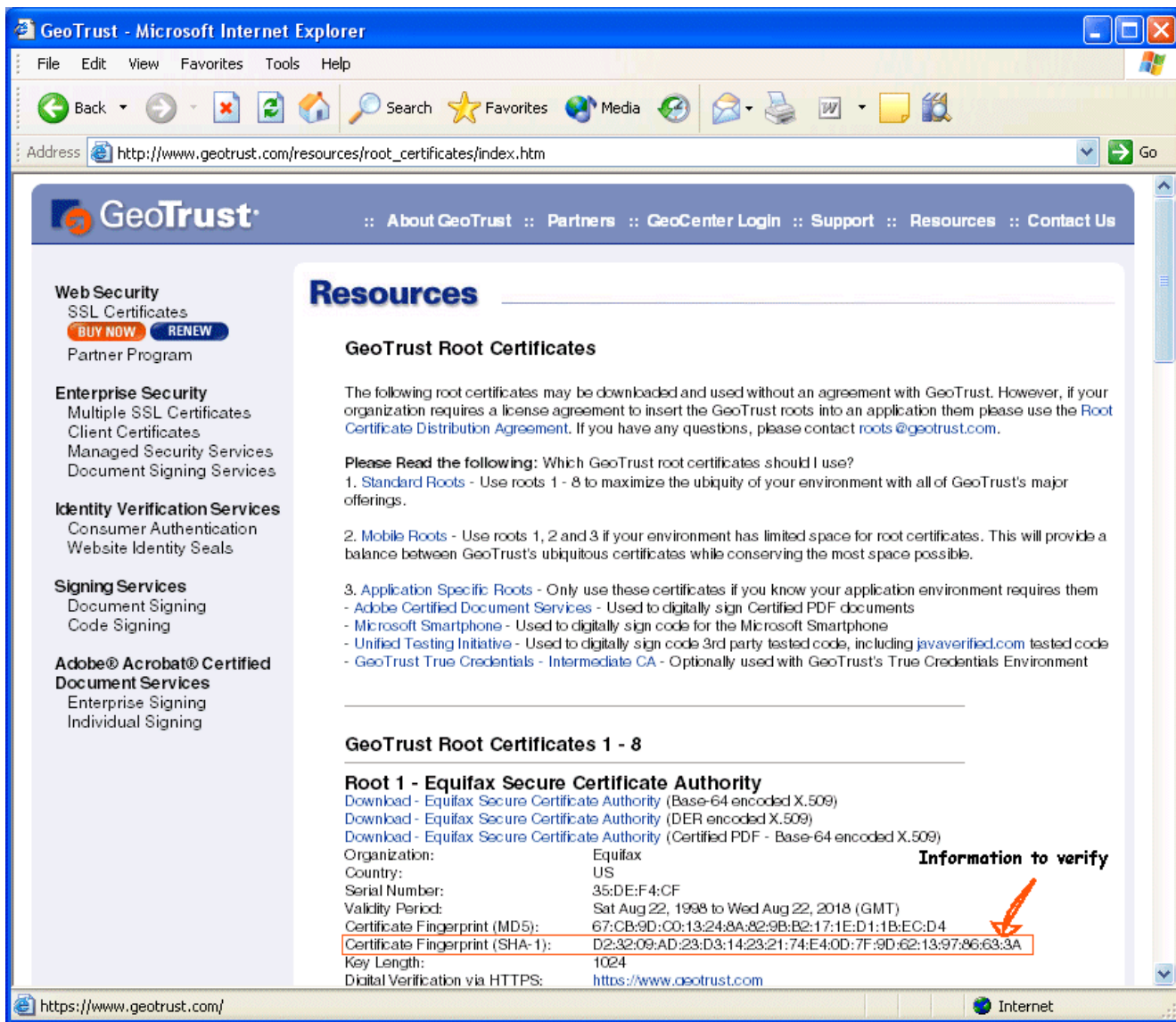


The details of the root certificate are now displayed. In particular, we want to examine the certificate's thumbprint, which is a characteristic of the certificate based on a calculation on the contents of the certificate. It is almost impossible to forge. If the thumbprint in the certificate exactly matches one we can obtain externally, we know that this certificate is authentic.

- Now that we know the thumbprint, we must verify that it is correct. To do so, we must be able to see the CA's "official" thumbprint, which is generally on its Web site. In this case, we also must somehow know, without resorting to any kind of magic, that in September 2001, GeoTrust bought Equifax's root certificate, and thus now controls it. This fact is important to know since information about the genuine Equifax root certificate is now found on GeoTrust's Web site.
- Go to the GeoTrust Web site at [http://www.geotrust.com/resources/root\\_certificates/index.htm](http://www.geotrust.com/resources/root_certificates/index.htm) (again, you must magically know where on the GeoTrust Web site to find information about their root certificates). Fortunately, GeoTrust devoted an entire Web page to help people validate the integrity of its root certificates. If there is no way to positively ascertain non-forgeable information about a certificate, then it cannot be verified, and users must take their chances.

**Note:** Creators of untrusted root certificates may not be as helpful as GeoTrust. For obvious reasons, in general there will not be a clear-cut place to authenticate an untrusted root certificate.

Figure 4: Finding the Equifax thumbprint on the GeoTrust Web site



6. Verify that the certificate's thumbnail is indeed valid. In this case, we must check that the string `D2:32:09:AD:23:D3:14:23:21:74:E4:0D:7F:9D:62:13:97:86:63:3A` in the certificate exactly matches the one from the Web site. Also note that due to inconsistencies in terminology between GeoTrust and Internet Explorer, you should know:
  - A *Certificate Fingerprint* is equivalent to a *Thumbprint*.
  - The *sha1* algorithm is equivalent to the *SHA-1* algorithm.

In this case, the certificate is indeed valid, since the thumbprint exactly matches the one that GeoTrust claims is authentic. If the thumbprint had not matched, then someone may have forged the Equifax root certificate in an attempt to cheat people with it.

# Dangers of installing untrusted root certificates

The process outlined above is clearly extremely complex and well beyond the capabilities of all but the most advanced computer users. It is unreasonable to expect that any but the most knowledgeable users could perform them when required. The implication is that it is unwise to create a system in which users are asked or expected to do this.

When confronted with such a security alert, naïve users will probably do what is easiest for them, which is simply dismissing the message box and assuming that everything is safe. If there are instructions on the Web site or elsewhere to simply dismiss the message box, then most typical users will certainly do so.

This ultimately has a negative effect. If users are taught, either explicitly or implicitly, to ignore the dangers that their browsers present to them, then they will continue to do this in other situations. If they do not see any immediate negative impact on their computer, this reinforces the belief that these message boxes “simply get in the way” and they will continue this behavior and teach others to do the same. In an extreme situation, they may even simply turn off the warnings. Ultimately, this behavior undermines the entire basis for SSL, since companies could believe that users are not concerned with real security. It is also a clear indication to hackers that it is easy to fool people.

We will examine the dangers of installing an untrusted root certificate into the certificate store of a browser for two similar but related cases, where the untrusted root certificate is created by:

- a hacker
- a legitimate business

## Hackers can exploit untrusted root certificates

It is easiest for hackers who want to steal personal information without being detected to first somehow install a root certificate that they control into someone's trusted certificate store. This is difficult, since most browsers usually guard against modifying this sensitive resource, and only allow changes approved by the user. However, some browsers do allow programmatic changes to their certificate store without user intervention; for instance, Mozilla still supports the `certutil` program for certificate store modification.<sup>10</sup> This means that a hacker simply needs to build this functionality into a Trojan horse program in order to achieve the desired results. Otherwise, a hacker can resort to other attacks, as described below.

### Getting a root certificate installed

Hackers can persuade naïve users to unintentionally install something that enables them to take advantage of others' computers. Hackers can simply resort to the tried and tested methods that make phishing so successful, such as:

- directing users to pages the hackers control that are protected by their SSL certificates
- sending e-mails to many users and including the hackers' SSL certificates as e-mail attachments

In both cases, hackers can send clear and simple instructions explaining how to install their SSL certificates along with a list of “benefits” to those that install it.

Recall that when a user installs a hacker's root certificate, no bells are ringing and no red lights are flashing. The user's antivirus software does not complain, because there is no virus to detect. And unlike viruses, which may be operating-system specific, this scam works on all computers, to the hacker's benefit. Users who think they are immune to hackers because they are running Macintosh must still be vigilant against these types of attacks. While an attack like this will leave no viruses per se on the computer, it will still be open to further attack, and hence is “infected”.

## Taking advantage of an installed certificate

Once hackers can widely distribute their root certificates, and these certificates are installed into the browsers' certificate stores in the computers of unsuspecting users, the hackers can begin exploiting them. Because browsers treat all root certificates in their certificate stores equally, hackers now have the same privileges that the browser developers (such as Microsoft and Mozilla) grant only to reputable organizations such as VeriSign and GeoTrust. This means that **any** certificate presented to the browser that is signed by the hacker's certificate does not trigger any security warnings. This creates many new problems.

For example, the hacker can easily use a root certificate to sign a certificate for [www.amazon.com](http://www.amazon.com) that he created himself. While he could never convince VeriSign or GeoTrust to do this for him, he can do it himself with the same results!

You may wonder what the danger is of a hacker creating an Amazon SSL certificate that appears valid. After all, Amazon controls its URL. However, if the hacker can position himself between the victim and amazon.com he could certainly launch a very effective man-in-the-middle (MITM) attack. In fact, the MITM attack can be SSL-enabled with a certificate that looks perfectly legitimate. Unless the victim investigated the certificate (which there is no apparent reason to do and which most users are incapable of doing in any case), there is no way to know that someone is listening in (and possibly modifying) the supposedly secure connection. Even worse, once the connection is complete, there are no clues left around indicating that anything untoward occurred. By tricking someone into installing an untrusted root certificate, the hacker can commit the *almost* perfect crime, simply.

If the hacker cannot position himself for a proper MITM attack, there are other possible scams. For example, to deceive with the phony [www.amazon.com](http://www.amazon.com) certificate, all he must do is divert traffic from Amazon to a host he controls. While this may be very difficult to do for a large site like Amazon, it is certainly easier with a smaller site by simply attacking the resolution of the site's name in the appropriate DNS server. DNS<sup>11</sup> is a system that serves an important purpose, but it is far from secure. There are many well-known ways to subvert the DNS system, so hackers have many options to use their phony certificates to deceive.

Since it is nearly impossible to determine that someone is listening in (and tracking him), the hacker can now freely observe all "secure" connections coming out of the victim's computer, such as banking, e-commerce, or any other service the victim uses that is protected by SSL.

## Legitimate businesses can run into trouble using untrusted SSL certificates too

In addition to all the problems users can have with untrusted certificates created and deployed by hackers, they can have problems with untrusted certificates from legitimate businesses that do not understand the consequences of their actions and put innocent people at risk.

Suppose a Web site offers valuable information to its members who subscribe to its service. For the user names and passwords to be secured during login, and to protect its own assets (the information that it serves) during transmission, the Web site chooses to implement an SSL strategy. Out of sheer hubris (and a bit of ignorance) the Web site's managers choose to act as their own CA in developing this SSL strategy in an effort to save some money on the certificates. This means:

1. The Web site generates its own root certificate, and uses it to sign its own SSL certificate, which it also generates.
2. It uses the SSL certificate to provide SSL service for its Web site.
3. It securely stores the private key of its root, which is known to be sensitive information. At least it believes that the key is secure. Properly protecting the key from both internal as well as external threats requires specialized hardware and software. Locking a diskette in a file cabinet is simply asking for trouble.

When members connect to this site, they are confronted with a security warning from the browser, such as the one in Figure 1, indicating that the certificate is not trusted. If enough members complain, the Web site may send instructions to its members about how to install its root certificate into the certificate stores of their browsers.

Once enough members install this certificate into the certificate stores of their browsers, the security of the certificate's private key becomes critical. However, the managers of the Web site are probably ill-equipped to understand the issues involved to properly secure this key from both internal as well as external threats. Someone with criminal intent who can access this key can launch the same attacks against innocent users that the hackers can, as described earlier.

Suppose that the legitimate business is an ISP and the attacker is an employee of the ISP from which the users connect to the Internet. In this case, the attacker can launch a MITM attack very easily and without detection. The ease of attack and the difficulty of detection make this scenario particularly attractive to unscrupulous employees.

## Can this really happen?

Companies are not merely "theoretically" vulnerable to a hacker's attack or theft from internal employees; some companies have already experienced such attacks. All companies that hold our vital information (such as ISP, cable company, cell phone company, etc.) are possible targets of people trying to obtain their confidential information for personal gain. While these organizations spend large amounts of money and effort to thwart these attacks, some of them do succeed. Unfortunately, when these organizations are successfully attacked, it is generally the innocent users who are the real victims.

## The saga of Paris Hilton

In February, 2005, the story of Paris Hilton's hacked cell phone was widely reported in the news.<sup>12</sup> The story of how hackers accessed Hilton's T-Mobile Sidekick II, stole her phone book, and posted it on the Internet, caused her great distress and problems with her friends, whose personal information and phone numbers were posted for all to see. The media had a field day as it reported on yet another blunder by the heiress who is popularly portrayed as a ditzy blonde.

But there's another way to view the Paris Hilton cell phone saga:

*Paris Hilton used a complex piece of electronics properly – and was victimized because of it.*

This perspective is less entertaining, so it was not reported. However, it **is** a more truthful version of the story. In fact, no one stole any data from Paris Hilton's Sidekick II: the data was stolen from the T-Mobile server where it was stored. Hilton used the service correctly and did nothing wrong to encourage her information to be stolen. Her only shortcoming in this story was trusting too much in T-Mobile. There is no reason to suspect that she did anything wrong that made her vulnerable, but there is every reason to believe that she followed the instructions given to her by T-Mobile on how to properly use its service.

Paris Hilton was not the first victim of this sort of security compromise and she likely will not be the last. Whether this security compromise was due to outside hackers breaking in to poorly secured servers, or employees of the company taking advantage of their access is not relevant. The point is that anyone who believes that the servers of their cell phone provider, ISP, or any other carrier of data that they use is immune to compromise, both internal and external, is badly mistaken.

## Wachovia and Bank of America

While the Paris Hilton incident was most likely caused by external hackers infiltrating T-Mobile's servers, the recent incident involving Wachovia and Bank of America was clearly an inside job. In May, 2005, it was reported that employees of Wachovia and Bank of America had used their privileged access to steal the financial records of over 100,000 customers of the bank and sell them for their own gain.<sup>13</sup> While the banks certainly had precautions built into their systems to try to prevent this from happening, it is very difficult to stop a committed thief from gaining access while still providing a usable system for the honest employees.

## Lessons to learn

The lesson to learn from both the Paris Hilton incident as well as the theft from the banks is that no system is immune to attack, whether internal or external. An attacker's success depends on how well the target is guarded along with how determined the attacker is to gain access. The very high potential value of a well-distributed untrusted root certificate to an attacker increases his determination. For organizations not well-versed in the proper CA procedures, it is reasonable to believe that the keys are poorly secured.

With this in mind, someone who blindly installs a root certificate from organizations without understanding whether or not they are really capable of acting as a CA, is acting much more foolishly and irresponsibly than Paris Hilton.

## Conclusion

The Bible instructs us: "You shall not curse the deaf nor place a stumbling block before the blind."<sup>14</sup> This is generally interpreted as an instruction not to exploit any shortcoming or disability of a person or group, such as naïveté, blind trust, or ignorance. In the context of this discussion, it means: for the sake of a few extra dollars, do not teach bad computer habits to the unsuspecting public, who do not understand the issues.

We have described the dangers of creating untrusted root certificates and encouraging naïve users to install them into their trusted key stores:

- It teaches bad habits to users, who generally do not understand the issues.
- The lack of immediate feedback that something went terribly wrong (such as when a virus is caught by an antivirus program) positively reinforces this unsafe behavior, which encourages its continuation and proliferation.
- It enables unscrupulous employees to exploit private customer information to their advantage.

When hackers begin to take advantage of this behavior, everybody loses:

- individual consumers lose privacy, personal information, and money
- legitimate businesses lose credibility and future business

Therefore, teaching and encouraging users to engage in this unsafe behavior undermines the entire SSL foundation upon which e-commerce is built.

Finally, a company that uses untrusted certificates in the public domain ultimately pays the heaviest price for its decision. For example, suppose a company creates its own root certificate and encourages users to install it. If those users are exploited by someone who gains access to that root certificate's private key, who is ultimately responsible? While legal liability may be unclear, it is clear that the value of the resultant negative publicity would far exceed any amount of money the company has saved by using an untrusted root certificate instead of a trusted one.

---

Founded in 2004, Sericon Technology is an independent software vendor committed to making the Internet more useful and easier to use for both corporate and consumer users.

Its patent-pending technology enables photo-sharing software to transmit personal information securely using SSL.

For more information, visit <http://www.sericontech.com>.

# Endnotes

---

- <sup>1</sup> [http://www.sericontech.com/WhitePapers/Introduction\\_to\\_SSL.pdf](http://www.sericontech.com/WhitePapers/Introduction_to_SSL.pdf)
- <sup>2</sup> <http://www.microsoft.com/technet/archive/security/news/rootcert.msp>
- <sup>3</sup> <http://www.webtrust.org>
- <sup>4</sup> <http://www.openssl.org>
- <sup>5</sup> <http://www.openca.org>
- <sup>6</sup> *Passwords revealed by sweet deal*, BBC News, online at <http://news.bbc.co.uk/1/hi/technology/3639679.stm>, Tuesday, 20 April, 2004.
- <sup>7</sup> A. Litan, *Phishing Attack Victims Likely Targets for Identity Theft*, Gartner FirstTake, FT-22-8873, Gartner Research, 4 May 2004.
- <sup>8</sup> A. Herzberg, A. Gbara. *TrustBar: Protecting (even Naïve) Web Users from Spoofing and Phishing Attacks*. <http://www.cs.biu.ac.il/~herzbea/Papers/ecommerce/spoofing.htm>.
- <sup>9</sup> K. Mitnick, *The Art of Deception: Controlling the Human Element of Security*, Wiley, 2002.
- <sup>10</sup> <http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html>
- <sup>11</sup> P. Albitz, C. Liu. *DNS and BIND*. O'Reilly, April, 2001.
- <sup>12</sup> *World wild Web of Paris hacker*, The New York Daily News, February 23, 2005.
- <sup>13</sup> *Banks Notify Customers of Data Theft*, The New York Times, May 23, 2005.
- <sup>14</sup> Leviticus 19:14