Global Information Assurance Certification Paper

License to surf?

SANS Security Essentials

GSEC Practical Assignment Version 1.2e

Eddy Vanlerberghe

Introduction

Some time ago I read an article about car accidents in the early days of the automobile era. Although there were not nearly as many cars on the roads as there are today, the number of accidents was higher and caused more casualties then today's traffic.

There are a number of reasons for that high death toll:

- traffic laws still had to be invented
- no certification required (that is: no drivers license necessary)
- cars were built with far less security in mind than today's cars
- accidents were accepted as a natural phenomenon (much like hurricanes, floods, ...)
- no infrastructure was available for emergency situations (no ambulances, no trained medics on the spot etc.)
- ...

If one goes even further back in time, cars were only available to those who were sufficiently technically skilled to build them, and to those who paid someone else to do it for them. Operating a car in those days required considerable technical knowledge about those devices.

While reading the article, I realized the striking similarity with the evolution in computer usage. At first computers were only available to those who could build them: nowadays, everyone with enough money can buy one. With the general public, lacking sufficient education, starting to use computers, incidents are starting to occur more and more frequently (virus threads, accidental removal of important data etc.)

In this document, the similarity between car and computer evolutions will be used to highlight security shortcomings in today's personal computer usage, as well as hint at possible remedies.

Although Mr. Cailliau, often named together with Mr. Tim Berners-Lee for inventing the World Wide Web, already suggested a license for web-surfing in 1999[1], in that interview Mr. Cailliau primarily stressed accountability as his primary reason for wanting a licensing scheme. This document will also discuss other topics, such as responsibility of ISP's etc.

Basic operation of personal computer systems

In the current personal computer market, the majority of the system runs some version of a Microsoft operating system, frequently completed with an office production suite of the same company. Although such a setup is usually perceived as easy to install and maintain, the truth is that lots of people make a pretty good living just from installing and maintaining such software. In fact, Microsoft itself acknowledges the need for professional assistance by creating an entire certification track[2]

Unfortunately, computers are treated as a commodity, just like a radio or refrigerator: you buy one, have it delivered at your house, switch the power button and all is well. There is nothing wrong with that, except that current day computers (and their software) are not ready yet for a technologically challenged public. Compare this with the time when automobiles were first built in large quantities on the first assembly lines. Anyone with the money could buy one and take it on the roads ... until they ran out of oil ("What do you mean, check the oil level? Oh, is that what this little flashing light is for?")

Things become worse when those same users hook up their equipment to the Internet: they not only have the potential to hurt themselves (smash a car into the wall of their own garage), but they can affect other Internet users as well (cause a head-on collision)

The security related domains could be roughly categorized as follows:

- user awareness
- insecure software (e.g. buffer overflows)
- poorly configured systems
- lack of professional support
- lack of international rules
- lack of rule enforcement agencies

The combination of the above results in the lawless environment of today's Internet. Clueless users are not only in danger of hurting their own computer experience, but, without them even knowing it, could also be part of an attack directed against third parties (e.g. Zombie systems used in distributed denial of service attacks)

User awareness

Compared with the automobile story, many users are unaware of the problems caused by drinking and driving, not driving on the right side of the road etc. The same applies to safe computing guidelines: many users are simply unaware of the issues involved.

Many documents about computer security stress the fact that users need to be educated in security issues. One such example is provided by SANS itself[3]. Other documents have already been written for GSEC certification. One of them has a rather catching title[4], but there are other examples available[5]

Probably one of the fundamental issues with respect to user awareness is that most of them do not realize that when they connect to the Internet, they actually enter a hostile environment: there are people out there to get at them. If one goes out into the more shady areas of a big city, one has visible clues that not everything might be nice and cozy about the neighborhood. The sight of a cable going from the computer on your desk to a wall-mounted socket is not nearly as threatening as a couple of guys wearing ski masks

and automatic assault rifles. However, people have been arrested in raids at dawn for the things they did with such connections.[6] [7]

As a result of this lack of understanding, most users act as if there is no security issue at all. If an email arrives from an acquaintance, they will click happily on the attachment (and thus launch a virus) After all, they know the sender and trust her, so what could be wrong with that? If a virus hoax arrives, they will be more than happy to forward it to as many people as possible: they are saving the world from a malicious virus ... but they will clog up valuable resources in doing so.

Reactions on message dialog boxes can be predicted pretty accurately: the user will click on those buttons that look as if they will allow to get rid of the message box and to get on with whatever action prompted the message box in the first place. The original idea of a message box was that users could read the contents and bail out of a potentially dangerous operation if necessary (e.g. if they accidentally clicked on a "Format Disk" button). Unfortunately, contemporary software tends to use message boxes gratuitously, with the result that most users don't even bother to read the contents of the message. The potential risks become clear if a web browser displays a security alert. For example, if a browser asks for confirmation to download and execute an unsigned ActiveX component, most users will hit the "Yes" button automatically, even after the famous Chaos Computer Club demonstrated the risks[8].

Nowadays people know that there are thieves out there to steal their car stereos, so they would not dream of leaving their car unlocked in the more shady places in a city: perhaps with time, computer users will start behaving more responsibly.

Insecure software

Not all car accidents are due to driver errors: cars can break down, wind may blow branches off trees and on top of oncoming cars, animals cross the road unexpectedly etc. The same principle applies to security incidents: not all breaches are caused by user misconduct.

Sometimes a software error can cause loss of data or loss of service, which in turn, could lead to other problems. Two interesting collections of real-life problems caused by software errors can be found in the references list.[9] [10]

Most software related security issues do involve a malicious party (in contrast to software bugs causing problems all by themselves) This type of incidents require two things:

- vulnerable software
- a malicious party willing to exploit the vulnerability

Two examples of programs that are known to have security related problems are

Microsoft Internet Explorer and Microsoft Outlook[11]. Because so many people use these programs, a security glitch in one of them has potentially great consequences. There is not much a user can do to prevent exploits using bugs that are not yet known today. The only thing a user can do, is to monitor if a new problem has been found and then download and install a fix as soon as possible.

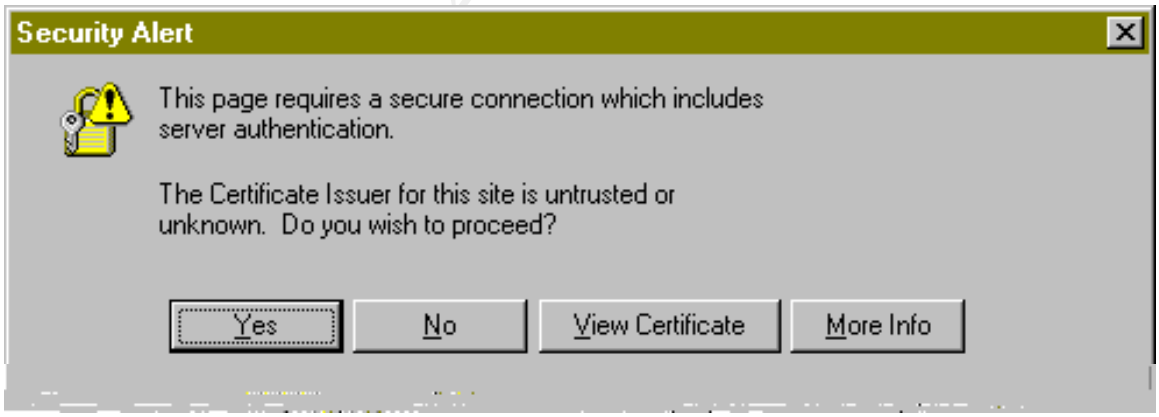
The automobile analogy might be that the car manufacturer recalls a number of cars that contain potentially dangerous components (e.g. a batch of airbags known to go off for no good reason) One difference with the software version, is that car manufacturers usually take the initiative of notifying their customers. Software users are on their own and have to hunt continuously for information about new fixes.

Other types of insecure software involve poorly designed user interfaces. Just like the fact that the brake pedal is in the same location in every car, a consistent and clear user interface helps security.

Probably the software that is used most in e-business is SSL (Secure Socket Layer), yet many of the client implementations are awful from a security point of view.

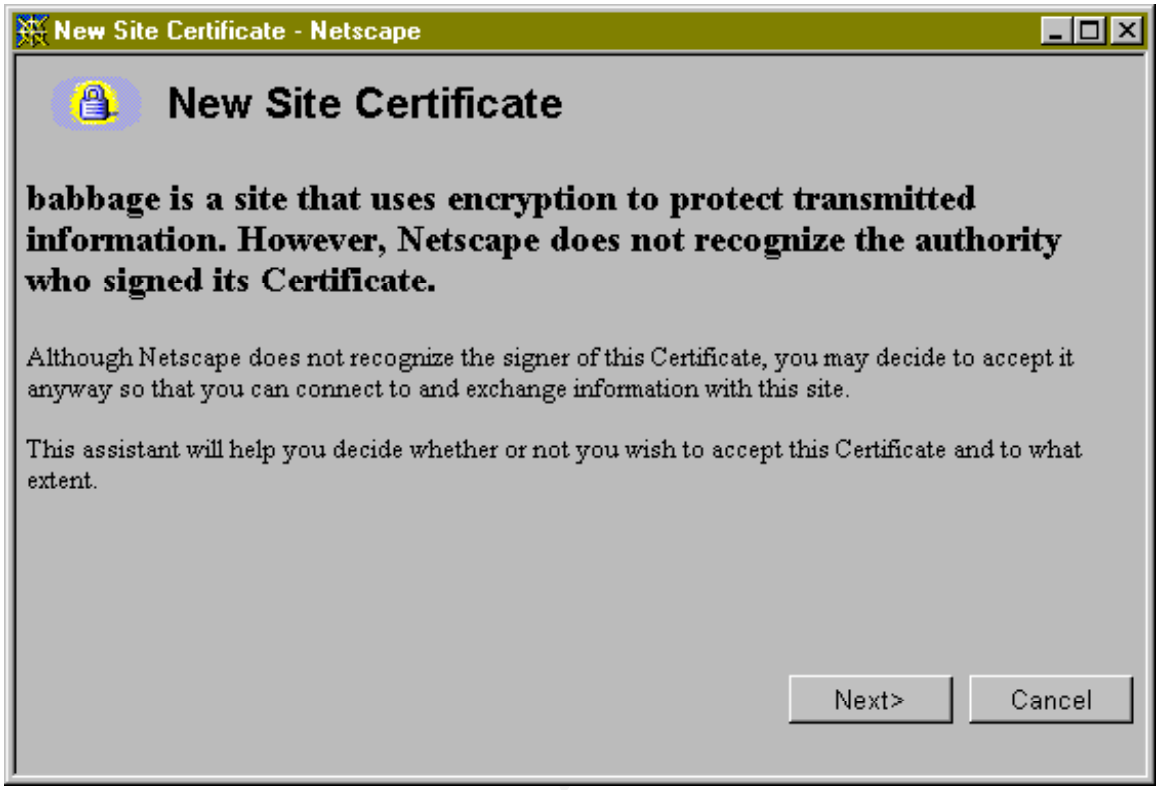
To illustrate this issue, here are the dialogs displayed by Microsoft's Internet Explorer (version 4.0) and Netscape Navigator (version 4.76) when a client connects to an SSL server that uses a certificate that was **not** signed by a trusted CA. In fact, both the CA certificate and the server certificate were generated using OpenSSL just for this single test.

The message presented by IE is this:



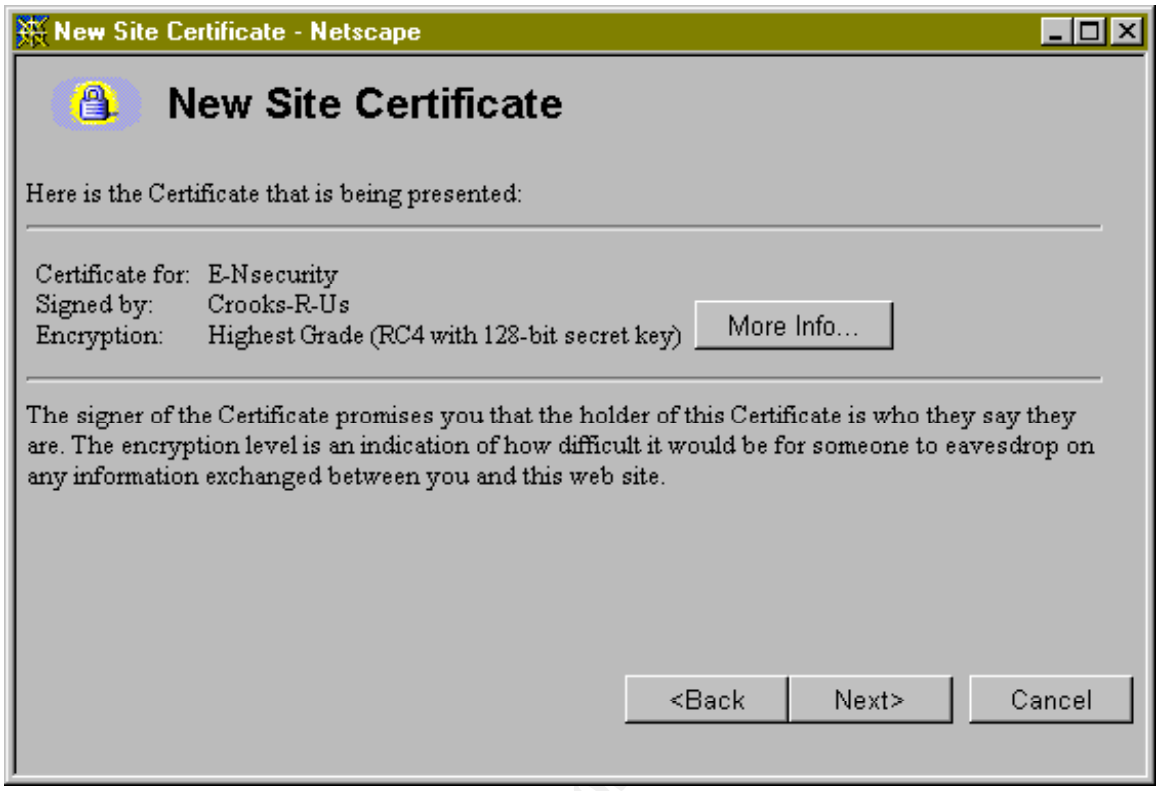
As indicated above, most users are conditioned, like Pavlov's dogs, to click blindly on the "Yes" button, such that they will receive the desired page. One could argue that this is more of a user awareness issue, but the fact remains that a server certificate signed by an untrusted CA is worthless: it does **not** guarantee the identity of the server, which was the whole point of the exercise. A better approach by the software developer might be an alert that someone tries to steal the users creditcard data.

Netscape users are presented more dialogs before they finally reach the requested page:



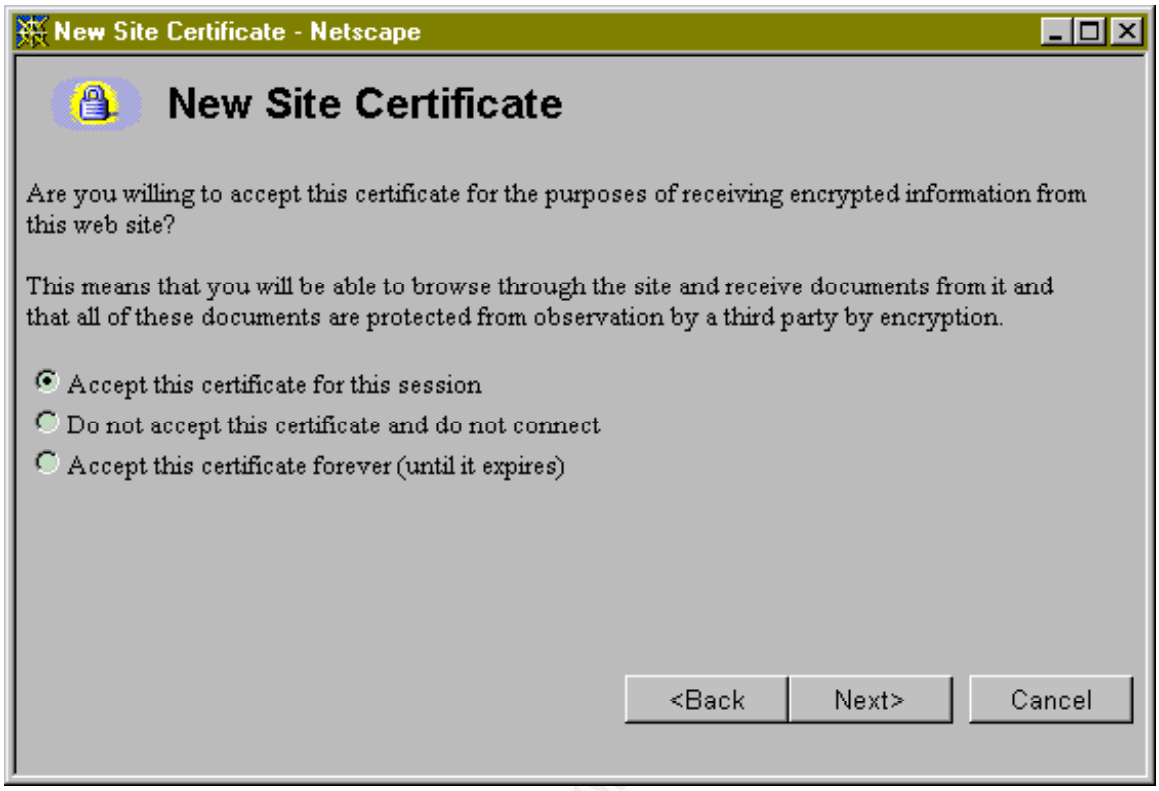
This message is still far too vague for most users. The part "does not recognize the authority" might just as well refer to a transmission hiccup. Slightly better is that this dialog does not contain a default button, so the user is forced to make a choice. Also, the name of the suspicious webserver (in this case the system "babbage") is displayed. This might help detecting unexpected detours by malicious scripts (if that name is not the one the user requested)

After clicking the "Next" button, this message appears:



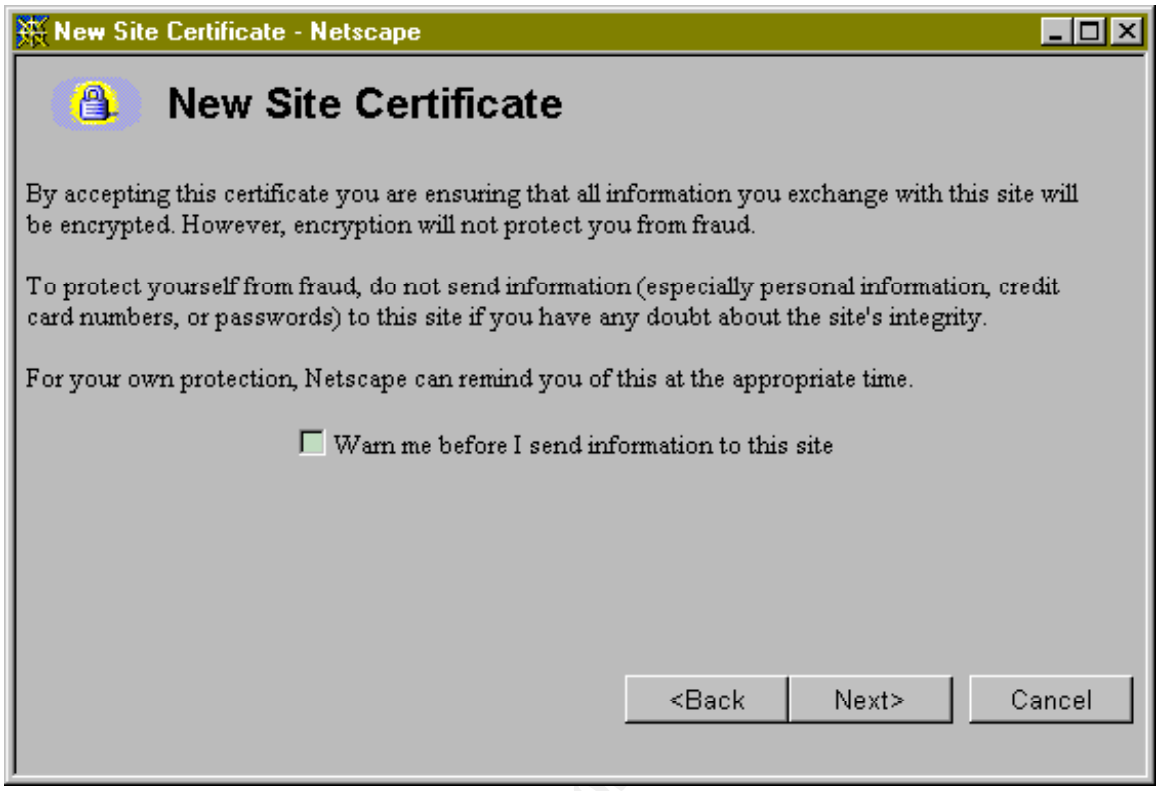
Again, this dialog does not contain a default button, so again the user is forced to make a choice. What is troubling, however, is the soothing text below the certificate information. To most users, this text appears to confirm that the certificate is to be trusted (note that the texts "E-Nsecurity" and "Crooks-R-Us" are arbitrary texts that were entered during generation of the certificate, so a more malicious user could easily replace them with "Amazon" and "Verisign"). The remark about encryption strength is irrelevant because we are communicating with a malicious entity, anyway. Its only effect will most likely be to ensure the user that all is going well ("Highest Grade")

After clicking the "Next" button, another annoying message appears:



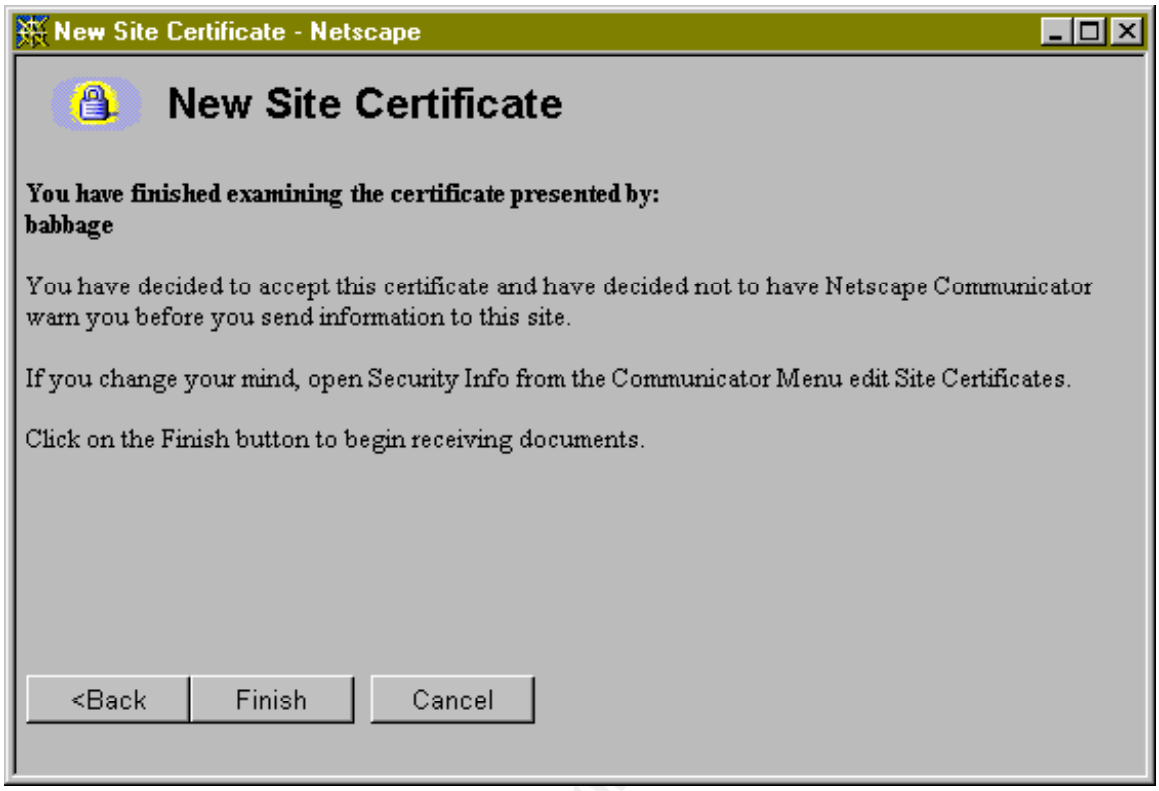
The only good thing, from a safety point of view, in this dialog is, again, the fact that there is no default button. A better design would use the middle radio button as default. The two sentences at the top are meaningless to many users: of course they want to receive encrypted information from that site and receive documents from it. That's why they selected that site in the first place.

Clicking on the "Next" button gives this dialog:



This is already the fourth message box presented to the user, and still she cannot see the requested page. In addition to that, this page comes pretty close the standard disclaimers found in just about every installation program. If the program authors would be really concerned about the safety of the SSL session, a better default would be to select the "Warn me..." checkbox.

When clicking on the "Next" button, again, the user receives yet another dialog:



There is not much new in this message box: it looks like a confirmation of the choice made in the previous boxes (the disclaimer). Only after the user clicks on the "Finish" button, is the requested page shown to the user.

Netscape forces the user to go through five message dialogs before the page is actually displayed, IE shows just one, yet none of them warns the user for what is most likely to be going on: a Man-In-The-Middle attack[12].

A better approach for a webbrowser, would be to simply refuse to connect to a webserver that presents a certificate signed by an untrusted CA. Most users do not understand the nature of SSL or PKI infrastructures, so it makes sense that developers make a safe choice that works fine for most users. Developers might feel the need for using test certificates nonetheless, but one may reasonably expect that developers do understand the issues and are capable enough to reconfigure their browsers to accept test certificates.

A similar test of two webbrowsers on the Macintosh platform was even less convincing, from a security point of view. Internet Explorer displayed similar behavior, but the OmniWeb browser gave no indication at all about a fishy certificate! This means that users of that program have **no** warning that they may be doing business with a less than trustworthy party.

Configuration glitches

Driving a convertible, with the roof open, through a car wash is usually not a pleasant experience. The same principle applies to computer security: if properly configured, a computer can be much more secure.

When installed using all defaults, most operating systems are not secure. This is especially true for the so-called "consumer" operating systems produced by Microsoft, as is indicated by number 7 (Global File Sharing) on the SANS Top Ten Most Critical Internet Security Threats[13]. Unfortunately, due to cable and DSL availability, the number of those vulnerable systems increases and most of their owners/users are unaware of the security holes in a default installation. So, most of those connected systems remain vulnerable. The Honeynet project has an interesting illustration regarding vulnerable home systems[14]. What is so interesting about this article, is not so much the fact that a machine was broken into, but the apparent ease with which the attacks were performed (and automated)

Although numerous consultancy companies make a nice living from securing systems for their clients, it is not an art or black magic. Unfortunately, their existence **is** an indication that this sort of work is well beyond the reach of an ordinary home user. However, there is no reason why an operating system company should not be able to produce a reasonably secure default installation.

From a hardware point of view, most modern PC systems are incomplete. A modern personal computer system usually has a harddisk capacity of many GB of data. The only backup medium in said systems is mostly limited to a single floppy drive. This means that most home systems are not properly equipped for making regular backups (1 GB of data requires over 700 floppy disks!) A growing number of home systems come with CD writers, and that could be a good evolution, from a security point of view.

Ideally, home systems should come with sufficient hardware for automatic backups, as well as software that enforces transparent backups. Modern cars come with an array of safety features, but under normal circumstances, they remain invisible for the driver (ABS, safety belts, airbags, spare tire...) Personal computer systems could be build with, for example, multiple disks, such that the operating system can maintain a few generations of backup data. RAID systems do not offer protection against unintentional deletion of files, or virus infections, so a system where multiple generations of data can be maintained would be better for backup purposes.

In addition to a backup system, a couple of other software tools should be present on every personal system: virus scanning software and a personal firewall.

Virus software has matured in pretty effective, yet unobtrusive, components. Most of them can catch viruses when an infected file is opened, so they do not rely on scheduled scans. In addition, they come with automatic update features, so that the list of known viruses remains up to date.

Personal firewalls are a pretty recent development and they are not nearly as unobtrusive as virus packages. Ideally, they should be effective, yet invisible to the user. Unfortunately, today's versions still require a bit of user configuration. The Securityportal site contains an analysis of personal firewalls[15].

Lack of professional support

For car owners, there are several layers of professional service available:

1. first line support, often in the form of automobile association services, for assistance when a car breaks down during a trip
2. dealers provide second line support for regular maintenance and, if necessary, repairs

For most PC owners, too, a layered set of services is available:

1. the eleven year old son of the neighbors
2. the shop where the system was bought
3. the computer manufacturer

Unfortunately, the first option hardly qualifies as professional, and, all too often, the second option is limited to packaging the faulty hardware and shipping it to layer 3: the manufacturer.

Today's situation on personal systems is quite a bit more complicated than maintaining cars. The computer scene, where everybody installs tons of software himself, may be compared to a DIY person who installs lots of electrical gadgets (TV, cell phone, radar detector, artistic interior lights etc.) in the car and afterwards wonders why the electrical ignition is dead. So, there is something to be said in favor of a mechanism that prevents that random software installations can interfere with one another. Microsoft is already working in that direction with Windows2000 enforcing rules on certified software such that an installer program does not interfere with system files. Signed drivers too are a step in that direction.

However, sandbox technology would allow running several applications on one system without them being aware of each other. One particular personal firewall vendor[16] uses this technique to protect the entire system. The idea is to monitor system calls for suspicious activity, but this idea could be extended such that every application runs in a virtual environment.

Such measurements would not prevent a user from installing suspect software, but from a maintenance point of view, it would improve (and encourage development of) professional assistance: when the system does not behave as expected, the sandboxes could be disabled one by one, so that the culprit can more easily be detected.

Lack of international rules

If a driver is caught for speeding in one country, the chances of him getting prosecuted in his own country are usually very small. If, however, that same user kills somebody in a hit and run accident, the chances of a successful prosecution in the country of origin increase dramatically (in some cases the culprit may be extradited to the country where the accident occurred)

The same is not true for crimes on the Internet. Laws regarding computer crimes tend to differ greatly between countries and, in some cases, they can be completely absent (as was demonstrated by the "I Love You" virus and, more recently, by the Sklyarov case[17])

Matters may even get more complicated. A server with a name that puts it squarely in a certain country top-level domain (e.g. *www.nice-english-webserver.uk*) is not guaranteed to actually be **in** that country. The domain owner might have her own reasons to put the webserver somewhere in San Francisco (perhaps because of cheaper hosting) So, if a UK resident hacker breaks into that system, which laws apply? Is it a strictly UK internal matter, or are the US laws involved as well? Things can become even more complicated if an attacker uses a route over systems that are physically in different countries of which at least a couple have no laws regarding computer crimes (or are not too keen on cooperation with law enforcement officers in the other countries)

It is clear that national laws fail completely in such situations and that therefor international rules have to be established.

As indicated by Mr. Cailliau, the individual web user should have responsibility for whatever happens on her system. In many countries, the owner of a car (license plate) is responsible for whatever happens with that car (unless he can prove otherwise) Holding computer owners responsible for software running on their system could be a strong encouragement for user awareness training. A typical Distributed Denial of Service (DDOS) attack uses a large number of compromised systems to attack one other target system. One report of such an attack (by a thirteen-year-old attacker!) can be found on the Gibson Research pages[18].

Normally, it is pretty difficult to find out the real attacker of a DDOS attack, but is usually pretty straightforward to find who is actually bombarding the target: the Zombie systems. Such systems are typically compromised computers where the owner does not even know that her computer is trying to bring another computer down.

Compare this with a youngster taking your car for a ride (after you let the keys in the ignition) and causing several serious accidents, before escaping from the police. Even if you manage to escape conviction for the accidents, you most likely will not get away with the fact that you recklessly left the key in the ignition and thus invited everyone passing by to "borrow" the car. Nowadays, most home users simply cannot be bothered to ensure that their system is not compromised and turned into a DDOS Zombie.

ISP's too have a responsibility to establish and enforce acceptable usage policies. In order to help establish criminal evidence, the very least they can do is to keep track of which user used which IP address in which time frame. This is similar to provide personal coordinates of the owner of a license plate, in case this information is required by law enforcement agencies.

Finally, software vendors should take responsibility for their products: customers pay good money for the software, so they are entitled to a decent product. The, now disappeared site, *http://www.l0pht.com* had an interesting text on top of the home page. The first sentence was a quotation from Microsoft: "That vulnerability is purely theoretical". Below that was a reaction from the L0pht crew: "Making the theoretical practical since..." This piece illustrates that many vendors are often unwilling to acknowledge that there might be a security risk in their product and that they only start working on it if the public opinion forces them to. Such hesitation may prove fatal for their vulnerable clients.

Lack of rule enforcement agencies

In the real world, there is probably no lack of law enforcement agencies. In cyberspace, however, there is no single organization that has the authority to enforce anything at all for the entire Internet. A French judge has done an interesting attempt at such regulation[19], but it is doubtful if that the decision will ever be enforceable.

Given the international nature of the Internet, only a supranational organization will ever be able to enforce international rules on the Internet. Such an organization does not have to be supranational if it consists of a set of tightly cooperating national agencies (a malicious user might be arrested by the law enforcement agencies of his own country), but swift and open communication channels between the national departments are a requirement. Once an attack is detected, tracing the tracks back to their origin may require immediate action in several different countries across the globe.

Conclusions

More and more people are using computers on the Internet and many of those new users are both computer and network security illiterates. They are treating computers just like any other appliance, but unfortunately, from a security point of view, the current state of technology is not yet ready for such public.

Internet users should have a minimum form of education such that they understand why and how to protect themselves and the rest of the Internet from abuse. With this education comes a responsibility for software running on systems they own and operate.

Hardware and software vendors could improve the security of home users computer systems:

1. provide the means for decent backups
2. provide safer defaults for software installations
3. provide software with less security problems (ActiveX does not use a sandbox, buffer overflows, automatic execution of code received over the Internet,...)
4. improve the overall user interface such that important issues are communicated clearly (for the user)
5. act promptly when security problems are reported

Improving the stability of home computers, will also ease the task of support people if something does go wrong: they will have a better chance of isolating the problem. Such diagnostics improvement may encourage support people to take computer support more seriously and thus a better, professional assistance community may develop.

Ideally an international set of laws would regulate the Internet, enforced by a supranational agency. A workable compromise may be to have similar laws in all Internet connected countries across the globe and have local law enforcement agencies cooperate tightly with their colleagues in other countries.

---

[1] The original interview with Mr. Cailliau seems to be lost in cyberspace, extensive quotation of said interview can be found at:
<http://www.fitug.de/debate/9911/msg00415.html>

[2] Microsoft Training and Services
<http://www.microsoft.com/trainingandservices/>

[3] "Consensus Information Security Awareness Draft Papers"
<http://www.sans.org/newlook/projects/cap_draft.htm>

[4] Simon Cope, "Your Naked Wife, Anna, in the Stables with a Trojan Horse!"
<http://www.sans.org/infosecFAQ/malicious/naked_wife.htm>

[5] William Rybcynzki, "Information Systems Security User Awareness: Social Engineering and Malware"
<http://www.sans.org/infosecFAQ/securitybasics/awareness.htm>

[6] "Police raid teen hacker's home"
< http://www.aftenposten.no/english/local/d121152.htm>

[7] "Four Israeli hackers suspected of planning New Year's Eve attack "
<http://www.globes.co.il/serveEN/globes/docView.asp?did=461297&fid=947 >

[8] "CCC: Microsoft Security Alert"
<http://www.ccc.de/radioactivex.html>

[9] "Collection of Software Bugs"

<http://wwwzenger.informatik.tu-muenchen.de/persons/huckle/bugse.html>

[10] "SOFTWARE HORROR STORIES"
<http://www.cs.tau.ac.il/~nachumd/verify/horror.html>

[11] "Internet Explorer security - Georgi Guninski Security Research"
<http://www.guninski.com/browsers.html>

[12] Kurt Seifried, "The End of SSL and SSH?"
< http://www.securityportal.com/cover/coverstory20001218.html >

[13] " SANS Top Ten Most Critical Internet Security Threats"
< http://www.sans.org/topten.htm >

[14] "Know Your Enemy: Worms at War"
< http://project.honeynet.org/papers/worm/>

[15] " Personal Firewalls/Intrusion Detection Systems "
< http://www.securityportal.com/articles/pf_main20001023.html >

[16] " Anti-vandal Sandbox"
< http://www.esafe.com/esafe/enterprise/sandbox.asp?cf=tl>

[17] " Free Dmitry Sklyarov !"
< http://freesklyarov.org/ >

[18] "Denial Of Service: Investigation & Exploration Pages"
< http://grc.com/dos/intro.htm >

[19] "French judge tells Yahoo! to block Nazi auctions"
<
http://www.zipple.com/newsandpolitics/internationalnews/20001123_yahoo_nazi_auctions.shtml >