



## **A matter of trust, or is it?**

Please fill in my web form – I only need a few personal details and your credit card number. It won't hurt a bit. But how do you know who I really am?

How do you know who you are really dealing with when disclosing your personal details over the Internet? How can you ensure the credit card details you submit are going to the site you expected? How can you be sure the latest anti-virus, firewall, or operating system software upgrade you install is bona fide? How do you know the link you click on is genuine?

This white paper looks at current methods available for proving the identity of a web site and explains why they fail. It offers an alternative solution to the problem of web site authentication.

Several schemes are already available for authenticating web sites but what do they actually achieve? At best, most offer a false sense of security.

## **SSL – secure communications but with whom?**

SSL ensures you have a secure means of communication between the client PC and the server. It is by no means impossible to redirect communications to a spoof site. Sure, your communications stay secure but not between you and the party you thought you were communicating with...

How many of us have actually bothered to check the link by double-clicking on the SSL icon? Why should we have to, and what does the information provided actually mean? To ensure you really are talking to the correct site it is important that you double-click the padlock icon and check carefully the information it shows you. Hardly dynamic, as you have to do this every time the padlock item appears to ensure you are communicating with the correct site. If you click and nothing happens leave the site, it's a fake.

So what stops a hacker placing a fake padlock item on your taskbar via some java script in a web page? Well not a lot actually, but chances are they would probably just redirect you to a fake site. You see the padlock item and presume all is OK as the communication is secure. Fact is, it is secure, just with the wrong site! You have placed your trust in a mechanism that has one huge weakness and you are not even alerted as to anything being wrong because as far as the client PC is concerned, a secure connection is taking place with a server. SSL harps on that its strength is ensuring the communications between the client PC and the server is secure. Fat lot of use that is if it is the wrong server!

Worst still, the padlock item normally only appears on web pages where you are making a purchase (involving credit card details). What about all of those sites that ask you for other personal information (name, address, phone number, etc.)? How can you verify whom this information is being sent to? It is all very well a site having a privacy policy that says they do not divulge your information to any one else, but what if you are unknowingly supplying that information to a totally different organization?



## **Web Security Logos**

There are several schemes that have sprung up that guarantee a site's identity via a logo that appears on the web site's home page. The idea is that you click on the logo to bring up a secondary window that verifies the site is who it claims to be. Other schemes direct you to information held on yet another site. So what stops me putting that logo on my site and producing my own pop-up window to verify I am who I say I am? Well nothing really.

Most schemes say their logos cannot be copied (often they prevent right-clicking on the image and doing a 'Save As') but their weakness is that anything displayed on screen can be duplicated by the 'ALT-PRINTSCREEN' command and pasted into a Paint package. Then it is a simple matter to select the desired image and save it as a file for inclusion on a bogus site. A pop-up window can be readily produced or a diversion to a spoof page, and the user would not be aware that anything was wrong. It is that simple.

Many logo schemes go beyond just checking that the company exists. They check that the company obeys rules such as privacy policies, trading standards, etc, etc. It goes without saying that the hacker, however, can still put these logos on their own web sites and the user would be none the wiser. A hacker isn't bothered by trading standards, they will be gone before the authorities can locate their computers. How much checking would you actually bother to do to make sure all is kosher? Unfortunately, all of the current schemes are static solutions that require interaction from the user. Yet another thing for the user to do and remember!

## **Scumware and bogus links**

In addition, a new web evil called scumware – please see [www.scumware.com](http://www.scumware.com) for more information – places bogus links on web sites (in the manner of a virus) diverting you to other locations without the owner's knowledge. A visitor to the site would not suspect anything was wrong – why should they? It is extremely easy in this manner to fool users into giving personal information away to the wrong site without either the original (and true) site or the users knowledge.

## **Server protection software**

Some organizations provide software that alerts the web administrator if content has been modified on their web site and automatically replaces it within a certain time. Other software actually prevents web pages that have been modified from leaving the server. The problem with both of these approaches is that web pages are held in caches on servers all round the world. There is nothing to protect pages once they have left the domain of the organizations web server. As a result, everyone is lured into a false sense of security thinking that the content they are viewing is always genuine. Also, these approaches do not stop DNS spoofing where a hacker diverts a user to a spoof site. There is nothing the user can do to check whether the site or the content they are viewing is genuine or not.



### **So why do web sites use these schemes?**

The bottom line is that so far nothing has been produced that is any better. Web site owners have implemented what they could to try and protect their web customers. There has been no better alternative solution to the problem – they are stuck with using what there is.

### **So what can be done to ensure trust on the web?**

What is needed is a dynamic rather than a static solution. This can be achieved by placing software on the client that transparently checks web pages as you browse and alerts you if anything is wrong. How about a solution that alerts you if you are dealing with a site that cannot be authenticated, if a site has some items on it that cannot be authenticated, or just if any items on that site fail authentication? The choice would be down to the user to decide how they wished to be alerted and of what – no other user interaction would be required. Users could then surf safely, knowing whom exactly they were communicating with, whether they were giving their personal information, credit card details, etc., to the right site and whether anything on a web site was bogus or had been modified by a hacker.

It should be every web site owner's responsibility to provide this service to anyone that visits his or her site. After all, you walk into a high street store, knowing exactly whom you are dealing with. Shouldn't this be the same on the Internet?