

# Shooting phish in a barrel...

A white paper by  
Piers Wilson,  
Siemens Enterprise



One of the latest family of attacks mounted against web-based applications, often those belonging to financial sector organisations, is that of phishing

Phishing is derived from the phrase 'password harvesting fishing' and describes the process of luring sensitive information, such as a password and other personal information, from a victim by masquerading as someone trustworthy.

In practical terms, phishing usually means the establishment of a fake web site, constructed to look similar to the genuine one, which the attacker then attempts to lure genuine users onto in order to harvest their login details.

It's a modern day variant of the older family of attacks where you would write a program to display a copy of a system login screen which is then left running. The next user logs on, the program copies their details to a file and then displays some sort of error message – or, if the developer has been really clever, it exits neatly, passes the credentials into the genuine login program and logs the user in anyway.

For this very reason, this paper will also cover the related issue of Trojan software that is specifically designed to capture passwords as they are entered by a user into a web site.

The problem with phishing, as a threat, is that it can be totally transparent to the genuine system. It is a threat that affects end users – either through their browser or as a result of the above mentioned Trojan installed on their system.

The issue, of course, is that irrespective of how secure your web site is, you potentially have millions of users with little or no security on their systems and scant knowledge of the threats they face.

However, there are measures organisations can take to reduce the risk of these types of attacks and, in the interests of reducing fraud and for your own and your customers' sake, you should at least consider some of them. Let's begin with a bit of threat assessment.

## Phishing – The threat

The actual threat we are trying to address is not really the new threat of phishing, but *unauthorised access or compromise of authentication credentials*. In these attacks – especially within the fairly broad net we have cast for this white paper – the attack is mounted by capturing the username, password and other user credentials.

# SIEMENS

Global network of innovation

Consider all of the ways that this could be manifest...

- Interception of details over the network
- Shoulder surfing
- Accessing credentials directly on the systems itself
- Poor use of cookies
- Establishment of a fake web site
- Attacks on users via a Trojan keystroke sniffer
- Persuading users to disclose their credentials.

As can be seen, there are a number of ways for the problem to arise. The last few methods mentioned – which are what phishing and this paper really refers to – are just one set of problems.

### **Conventional wisdom**

From the list above, the first few types of attack are fairly well known, fairly well understood and, in general, fairly well defended. Most organisations now accept usernames and passwords over the web using SSL to protect information in transit between the browser and server. However, it is probably worth us reiterating some standard advice for the use of SSL:

- Ensure that the appropriate strengths of encryption (cipher suites are used)
- Ensure that information is not cached at the client end
- If possible, utilise hardware security modules (HSM) for both key storage and faster crypto processing – keys stored on web servers themselves can be at risk from some types of attack.

Be aware that if you do use SSL (Secure Sockets Layer), your network-based IDS systems might not be able to detect all attacks – some web based attacks will be encrypted and hidden by the SSL itself. Consider terminating SSL sessions at a proxy server layer in front of web servers.

*Shoulder surfing* is another familiar attack type – and defences against it are also commonly in use. We want to avoid making it too easy to see the passwords typed in (eg. for someone sitting behind, or next to, a genuine user at an Internet cafe).

Preventative measures include:

- Ensure passwords are blanked or asterisked on the screen
- Don't make login processes too easily visible (ie. use normal sized text fonts, not large, easily readable input boxes for usernames)
- Make users or customers aware of the need to avoid letting people see what they type at a keyboard.

Usernames and passwords (or other credentials) are, of course, potentially at risk on the servers themselves. SSL only protects data in transit and, once received by an application, the authentication data is often stored in an unprotected database or within some authentication mechanism such as a directory. It goes without saying that you need to:

- Make sure passwords are not stored unprotected (ie. are hashed or encrypted)
- Prevent access to the usernames and password information – ensure your application is not vulnerable to various application weaknesses such as SQL injection attacks which might allow this data to be accessed
- Make sure web server and application connections to databases are controlled and do not use a high privilege and/or unsecured account
- Use stored procedures to perform validation (and other database accesses and queries), rather than doing it within server scripting code
- Ensure directories or other authentication solutions are protected.

Be careful how cookies are used. It is important to avoid storing credentials or other useful information (to an attacker). Also, if you allow a user the ability to 'keep logged in from this computer' then make them aware that they should not do this on a public or shared system.

The management of authenticated sessions is an issue in its own right and extends beyond cookies into the whole process of keeping a valid state throughout an inherently stateless browser session.

### **Now onto phishing...**

Having discussed the standard advice above, is there anything we can really do

about people establishing a fake web site and luring our users to it? Can we take any steps to avoid the capture of passwords on users' own home systems?

There is no silver bullet to addressing these threats, but there are a number of things that can be done to reduce the likelihood or impact of these type of attacks. We'll describe the ones that are most important or useful below.

### Getting more from passwords

There are a number of ways one can extend the traditional username/password process to make life harder for phishermen. These all typically revolve around ways to emulate one-time passwords or provide simple challenge-response mechanisms.

**Multiple credentials** The easiest way is to have a number of different passwords or phrases that users must supply. This is often done by using a password or PIN they always give, along with a separate piece of memorable information randomly chosen from a list of perhaps four possible questions and answers (typically mother's maiden name, place of birth, memorable date, memorable person, etc) as shown below. Note that italic text would be concealed and bold prompts would change at each login:

Username	wilson
PIN/Password	<i>password</i>
<b>Place of birth</b>	<i>Mytown</i>

This means that information potentially changes every time, although there are only a small number of possible values. It would deter someone who happened to overlook a single login and would also defend against the storing of passwords in the browser (the second password would be different at each login).

**Chosen characters** The next stage of evolution is to take a password or secret word and ask the user to select a subset of characters from it. This is, again, typically used in conjunction with a fixed PIN or password. Essentially, given an eight character secondary password, you then ask the user for the characters at positions x and y as shown below.

This makes it much harder to intercept or overlook credentials. There is a way of working out how many exchanges you'd have to monitor to recover the whole word. An attacker might get lucky and be able to guess it having seen some of the letters (eg. P A \_ S W \_ \_ D) or be unlucky and end up seeing the same letter a number of times.

Username	wilson
PIN/Password	<i>password</i>
Enter the <b>5th</b> character of your memorable word	F
Enter the <b>8th</b> character of your memorable word	K

This method is really a variant of a challenge-response mechanism, albeit one that is simple enough for a human to process.

**Non-keyboard input** Of course neither of the above will protect against a hardware or software keystroke sniffer. These are typically a tiny device that connects between the keyboard cable and the socket on the PC or, much more commonly, a Trojan program that logs key strokes.

One solution, used by at least one major UK financial services organisation, is to force the input of the chosen characters using a pull down menu, rather than a single character text box. They even precede the pull down options (ie. the letters) with a space to prevent users shortcutting the pull down process with the actual key press.

Username	wilson
PIN/Password	<i>password</i>
Enter the <b>5th</b> character of your memorable word	F ▼
Enter the <b>8th</b> character of your memorable word	K ▼

This way the user will never enter the password as keystrokes, simply by mouse movements, which are much harder (if not impossible), to intercept from the client system.

**Graphical prompts** Somewhere in the middle of these techniques is a practice of providing a user with a challenge as a graphic which must be re-entered as text. This is often used when registering for a system – as a mechanism to enforce uniqueness of the authentication exchange and to prevent interception when the accounts are set up.

Graphical information is slightly harder to intercept and replay than textual information, and this also ensures that no two authentication processes are the same, as the graphical challenge is different every time.

Username	wilson
PIN/Password	password
	Help
Enter the word above exactly as it appears	Help

### Site processes

So far we have talked about ways to construct an application login process to reduce the opportunity for authentication compromise. We have also highlighted other controls that should be in place, such as database, host and network security. There are several practices and processes that can also help reduce the risk from this family of threats.

**Domain name registration** Cyber squatting is nothing new, but the attack used to be that someone would register the name and then try and sell it back to you – this has declined now as a result of some fairly high profile court cases.

The phishing threat – in its purest form – relates around the ability to create a web presence that, to a casual observer, is indistinguishable from a genuine site.

Copying the HTML, images and 'look and feel' is easy – your browser downloads all of the necessary data each time a page is retrieved. If an attacker then uploads that content to a similar sounding URL, they are very close to welcoming their first unsuspecting victim.

There is a balance, of course, in where do you draw the line? Do you get just

the .com and .co.uk variants, or try and obtain some of the more exotic suffixes as well? Do you try and get hyphenated versions registered, eg. *my-bank.co.uk* as well as *mybank.co.uk*?

Clearly, you could do this *ad infinitum* with various URL combinations. However, with the ever decreasing cost of domain name registration (.co.uk suffixes can be picked up for as little as £10) it can be a cheap countermeasure. There is also the side benefit that a legitimate customer has an easier time finding your web site, as virtually every combination of letters in the URL will work.

**User awareness** One of the biggest problems is that many phishing attacks (certainly some of the ones that ISPs have suffered from) rely on social engineering.

A customer will receive an email explaining that there is a problem with the accounts database and that the organisation needs the username and password to check whether they have been affected or to repair their account.

These types of attacks work, not just because users are easily fooled, but because the attackers are able to sound genuine – trading off the fear many people have about online trading, Internet access and computers in general.

The solution, as one politician put it, is 'Education, Education, Education'. It is imperative to inform users and customers that they must *never*, under any circumstances, disclose their password or credentials and why this is so important.

Organisations may even need to back this up with limitations on their liability either as a bank, on-line retailer or whatever – the security of ATM PINs operates in much the same way.

Raising awareness for such a potentially large and diverse population can be logistically difficult. If you are raising awareness of your staff – explaining to support personnel why they must never ask a user for their password, for example – you can achieve this face to face, or perhaps encourage them to watch a computer-based training (CBT) presentation on security best practices.

With a user or customer population running into the thousands or millions, and the fact that you cannot risk raising

hurdles that obstruct casual browsers or first time purchasers, the problem becomes far harder.

In such a case, security awareness must be focussed and to the point. Simple, clear phrases such as *'Never under any circumstance give out your password to anyone. No Mybank.co.uk staff will ever ask you for your password'* are to be recommended.

These can, of course, be as friendly or as legalese as you like, although you do still need to take account of consumers' basic rights.

Couple this awareness with [published] limitations on liability and you have gone some way to not only protecting yourself as an organisation, but also to protecting your customers and clients.

Don't forget, too, that if you do intend to limit your liability, you may need to be prepared to defend your position – this means having adequate systems logging, so that if your system is abused in this way, you can clearly prove that the false transactions did occur along with the details about where it originated from.

Be aware, too, that in mass, large scale attacks (the kind that make the news) you may also need to consider the PR issues of taking a harder line on the losses brought about by users.

**Reducing the phishing risk** Essentially, organisations must define clear and firm practices for communications between the site (be it a bank, retail, or any other type of business) and the user population.

Once these principles are established, they must be publicised and repeated as often as possible – even being explicitly reiterated during an attack. Consider establishing the following as a policy statement:

***We will never send an email to a user with a link that allows the user to click to our site, the URL must always be typed in directly.***

You could even use your registration process to place a bookmark or desktop icon on a users system – although depending on their local settings this might not be possible, and there is a risk an attacker would divert this mechanism to their spoof/phishing site.

### **More proactive, positive steps**

There are also many more proactive things that concerned companies can consider. It may be prudent to have test accounts set up, valid email addresses and customer accounts that, to all intents and purposes, are real customers.

If a major phishing attack is launched against your business, there is a chance (although not a certainty) that someone within the administration team or security incident *chain of command* will get at least an early warning through being invited to participate in whatever scam is being perpetrated.

Another thing to consider is the provision of security facilities for the user end of your systems. One organisation in the UK did, for a time, offer a limited number of free copies of an Internet Security suite to its users. This was a worthwhile initiative and, although clearly it didn't cover their entire user base, it did at least give the users something tangible and useful.

Of course, if you have millions of customers this may not be possible, but there is always a possibility of making an arrangement with an Internet security company to allow you to offer the software (possibly even a restricted version) to your customers at a reduced cost. You might even consider modifying your liability clauses for those users who take up the offer (eg. having different levels of loss protection for those users who have taken responsible steps to protect their PC).

There is also the possibility to provide the function as part of the web site itself – there is at least one anti-virus supplier that runs a web-based scanning solution that anyone can use to scan their local system without having to download and install a piece of software (it runs from inside a browser window).

Organisations at high risk could provide either links to these kinds of services or, commercial arrangements notwithstanding, have equivalent functionality offered from within their own site.

This type of service would actually give the organisation a large amount of control over the minimum levels of security at the client end. You could log when a given user last ran a check (through your site) and, if a new Trojan had been released

