CAN STRONG AUTHENTICATION SORT OUT PHISHING AND FRAUD?

Paul Ducklin
Sophos Pty Ltd, Level 11, One Elizabeth Plaza,
North Sydney, NSW 2060, Australia

Email duck@sophos.com.au

ABSTRACT

Authentication, especially two-factor authentication, is seen as an important step against on-line crime, especially for on-line banking and Internet shopping. But authentication alone is not enough to protect computer users against the efforts of organized crime to thieve their credentials, their data and even their identity.

In fact, strong authentication in only one part of a system may even make things worse if users expect to rely entirely on technology to protect them from phishing and related attacks.

Organized criminals have realised (precisely because they are organized) that phishing and identity theft can be carried out over an extended period, by piecing together snippets of information from separate attacks for a final sting. For example, logging on using an authentication token will neutralize password stealers, but the very presence of a token authentication request can make an ideal trigger for spyware, especially if its goal is to build up a pattern of your on-line behaviour by monitoring your financial transactions.

This paper traces the recent evolution of malware techniques in response to technological changes in our security regimes, and proves once again the old cliche that the price of freedom is eternal vigilance. The Bad Guys are out to get us, and if they can turn our defences against us, even in the slightest way, then they surely will.

Q. Can strong authentication sort out phishing and fraud?

A. No.

Q. Hmm. That makes for a rather short paper, don't you think?

A. Yes.

Q. Could you go into a little more detail?

A. These days, a lot of phishing is orchestrated, or at least assisted, by malicious code somewhere in the network. This means that solving the problem of malware is effectively a necessary part of solving the problems of phishing and fraud. (When we say 'fraud' in this paper, we mean on-line fraud against users conducting business via their PCs. We do not mean other sorts of financial fraud such as credit card abuse or kiting.)

But solving the malware problem is hard – indeed, it is undecidable. After all, the Halting Problem tells us that we cannot write a program which will reliably determine the behaviour of all other possible programs:

> 'No program can say what another will do.
>
> Now, I won't just assert that, I'll prove it to you: I will prove that although you might work til you drop, you can't predict whether a program will stop.
>
> [. . .]
>
> You can never discover mechanical means for predicting the acts of computing machines.
>
> It's something that cannot be done. So we users must find our own bugs; our computers are losers!' [1]

This general result can be cast into specific terms to show that a program which will distinguish unfailingly between malware and non-malware cannot be made. Malware authors always get a 'next chance' to circumvent the protection we currently have in place [2].

Q. However, that doesn't mean it is always easy for malware authors, or for phishers, to go to the next level, does it?

A. No. I was just being dramatic. Nothing, whether it is authentication or something else, can actually *solve* the problem of phishing, in a mathematical sense of solving it. But we can make phishing much harder, and authentication is indeed one of the tools we can use.

Q. Staying on the topic of malware detection for a moment, how hard is it to produce malware – a new banking trojan, for instance – which evades detection?

A. On one hand, it is getting harder. On modern PCs, anti-virus software can be much more computationally aggressive than it was in the past. Generic detection techniques mop up a lot of new trojans proactively. On the other hand, it is getting easier. You may even be able to precompute whether your new malware will succeed.

One way to do this is through a targeted attack, where you write a trojan and aim it at a specific part of the Internet, such as a single company, whose defensive posture is known to you. Targeted attacks are not especially difficult to orchestrate, and there is a paper at this conference which investigates this phenomenon [3].

Another way is to use an on-line service to which you can submit malware samples and from which you will receive automated replies telling you which products detected it, and what they called it.

Q. On-line services to help you fine-tune your phishing trojans?

A. That's not how they position themselves, of course. Several such services exist, and some are strongly supported by the security industry. *VirusTotal* [4], for example, has permission to use some 25 different products for scanning incoming files. In return, samples are sent to vendors who miss them, thus helping to improve detection and responsiveness. Unfortunately, *VirusTotal* allows you to withhold submissions from vendors (though this is not the default), which could be said to play into the hands of organized crime and the counterculture.

Q. So let's assume you can create a new phishing trojan and target me and my company with it. How can authentication, or anything else, help me then?

A. When you are carrying out a financial transaction on-line, there are several things that it pays you (literally and figuratively) to check:

- that trustworthy software is orchestrating the transaction,
- that it really is you yourself conducting the transaction,
- that you really are trading with the person or service you expect,
- that the details of the transaction are correct.

Authentication, clearly, can assist you with this.

Q. How? Can you start by giving me an example of the sort of authentication technology which can help with each item above?

A. Of course. Let's ask the questions we want answered one by one.

- Is the right program doing the work? Some endpoint firewalls can help with this, for example by using cryptographic checksums to regulate which applications can make what sorts of connection to which servers.
- Is it really you kicking off the transaction? A hand-held authenticator can ensure that you use a new password every time you connect, which helps to prevent replay attacks where previously-stolen credentials are re-used by someone else.
- Are you connecting to the right service? Digital certificates can help to reassure you that you are not speaking to an imposter at the other end.
- Are you carrying out the transaction you intended? Encryption and digital signatures provide protection against exposing the details of the transaction, and help prevent the transaction being tampered with in transit.

Q. Firewalls, tokens, certificates and encryption. Aren't these old technologies that we've been using for ages? Are they failing us?

A. Yes and no. There are three main ways in which security-related systems fail, and these are mirrored by the main ways in which cryptographic systems fail. This is unsurprising, since computer security relies heavily on cryptography. Things can go wrong because:

- the underlying design is flawed (e.g. a defective cipher),
- the implementation is incorrect (e.g. insufficient key material is used),
- the system is used wrongly (e.g. users write down their PINs).

In a seminal paper about the failure of cryptosystems [5], Ross Anderson shows that problems in implementation and use seem to be the main reasons for failure, rather than weak cryptography.

With hindsight, this is perhaps obvious, since they are the two aspects in which human error is most likely and in which rigorous peer review is hardest. In the last case, human error can effectively be guaranteed by cheating or misleading users.

Of course, what this means is that systems which *can* work correctly to provide us with safe on-line commerce *may* fail in unexpected ways.

Q. But if a system is vulnerable because it doesn't deal well with inadvertent or unexpected use, doesn't that mean the design is wrong?

A. Perhaps it does. But the PC, and its operating system, is designed to be a flexible, general-purpose tool which can be adapted to many tasks, such as word processing, browsing the Internet, watching movies, making art, designing buildings and searching for extraterrestrial life. Users are generally free to add and remove any software they like at any time in order to enjoy this flexibility.

When you carry out commerce on-line, for example when clicking on a [Buy now] link, you need to turn your PC – temporarily, and at short notice – into a secure cryptographic device which acts as an important component of the transaction.

So it is hardly surprising that the design of such a system makes certain assumptions about the state of the PC, and the awareness of the user. And it is hardly surprising that the PC, or the user, or both, sometimes let the system down.

Q. Is this really unsurprising? Don't the banks owe it to us to do better?

A. This paper isn't really about the social contract which banks do or don't have with their customers, so we'll just look very quickly at both sides of the argument.

Critics of the banks say that the banks aren't doing enough. They say it is the banks who have the greater interest in Internet commerce, because it allows them to close branches, lay off tellers and front-of-house staff, and thus to save an awful lot of money. This money, they argue, should already have been used to make Internet banking much safer than it is.

The banks, on the other hand, can argue at least as reasonably that the popularity of on-line commerce is driving the need for Internet banking (*eBay*, *QED*). They can also point out that their younger customers not only much prefer Internet banking but that they expect it to be cheap, and easy, and accessible from anywhere. If the bank cuts off their Internet banking in the interests of safety, and requires them to visit a branch to sort out any possible problems (a reasonable security precaution, you might think), this is viewed as a bug in the system, not a feature.

Uri Rivner of *RSA*, which makes and sells cryptographic solutions including hand-held authenticators, agrees:

> '...[I]n the online consumer authentication market, usability is in many cases of greater importance than security. It's true that some people [would] like to see changes in the banks' security procedures and [would] appreciate it if the financial institution handed them authentication devices or came up with other visible security measures.
>
> But other customers don't really care about all of that; they demand security from the bank, but all they really want is to access their account, pay bills and transfer money without any delay or additional challenge...' [6]

Q. OK, let's go back to the failure points above. Can you give historical examples of each sort of failure, to paint a picture of the sorts of thing that can go wrong? Let's start with the most exciting-sounding one: a cryptosystem which got cracked.

A. An example many people probably know about is Wired Equivalent Privacy (WEP), the authentication and encryption system originally proposed for wireless networking. WEP relies on a secret key, either 40 or 108 bits in length; to access and use the network, you need to know the key. (This, in turn, means you can read all the traffic on the network, just as if you were on a LAN.)

As it happens, the cipher used by WEP has a statistical flaw which affects the randomness of its early output bytes. Interestingly, the cipher, RC4, is also used in SSL (which we will talk about later), but in a way which does not cause the problems seen in WEP. Nevertheless, the flaw exists in the RC4 cryptosystem itself, or at least its key scheduling algorithm (KSA) [7], rather than simply in WEP's implementation.

This statistical flaw allows an attacker to recover a WEP key by capturing and analysing a few million wireless packets. So there is no way to fix WEP without changing it for something different. WEP is irrevocably broken.

Q. How about a system which was based on sound cryptography but implemented dangerously?

A simple example of an implementation flaw – one which was fixed by devising an alternative but compatible approach – is the way early Unix systems stored their password file. All users and programs need read access to this file, as it is (amongst other things) the database which maps usernames, such as 'fp', onto real names, such as 'Ford Prefect'.

However, early Unix implementations also stored each user's hashed password in this file, so anyone could retrieve the hashes and perform a dictionary attack against them off-line. This meant that weak passwords could quickly be recovered without leaving evidence of the dictionary attack on the targeted system.

The backward-compatible solution, used in *Linux* to this day, was to duplicate the password file, to replace the hashes in the world-readable file with a dud entry, such as 'x', and to read-protect the second copy of the file, called the shadow file.

User programs worked exactly as before, except that they saw dud information for the password hash, which they didn't need anyway. Only the login program needed changing to use the shadow file instead.

Q. And what about a case where we used security wrongly and paid the price?

Perhaps understandably, many of us are willing to assume that anyone who is prepared to confirm his identity must, ipso facto, be trustworthy. So when we come across an unknown program which is digitally signed, we sometimes assume that the signature tells us something about the morals and the character of the signatory, rather than simply about his name.

So, for example, in late 2002, many people willingly downloaded and installed software known as FriendGreetings from a company identifying itself as Permissioned Media [8]. These downloads were in response to an email, usually received from a friend or acquaintance, which promised an electronic greetings card.

FriendGreetings displayed two End User Licence Agreements (EULAs), in the second of which it claimed permission to email everyone in your *Outlook* address book. Which, of course, it promptly did.

For system administrators and for those in your address book, the side-effects were little different from a mass-mailing virus such as LoveBug (VBS/LoveLet-A). The signatories, of course, claimed that the virus-like behaviour of their software was entirely legal, as it asked for permission before sending any email.

But who had ever heard of Permissioned Media Inc. of Sun Towers, First Floor Office #39, Ave. Ricardo J. Alfaro, Panama City, El Dorado Zona 6, Panama? And why did they trust this unknown company with their email address book?

Q. That was in 2002. Have users got smarter since then?

A. FriendGreetings was a problem for system administrators, because of the unwanted email it generated. It was an annoyance for users, for the same reason. The application also had the troublesome side effect of preventing programs from appearing in the taskbar, which interfered with the correct use of an affected PC until it was correctly cleaned up. But FriendGreetings didn't set out to steal information that could be used to plunder your bank account or to carry out fraudulent transactions.

Phishing has raised the bar in terms of the risk that each user, and each user's organization, faces from malicious code. This, in turn, has raised both concern and awareness about malware and the importance of preventing it. Whether this counts as a silver lining to the cloud that organized crime has brought into the malware scene is not clear, but an optimist would say that it has.

Q. That's an interesting observation, but I notice you have skirted the question. Have users got smarter since 2002?

A. Security experts are always on a slippery slope when commenting on the knowledge, or lack of it, shown by users. To come down too hard against users sounds arrogant, but to exonerate them from any responsibility for their own PCs is to assume that technology can solve all security problems, which, as we demonstrated light-heartedly at the outset, it cannot.

However, recent research carried out in the USA [9] paints a rather dismal picture of levels of common sense amongst users. (More accurately, it paints a dismal picture of a very small sample of academic staff and students at a prestigious American university. The rest of us might back ourselves to do rather better, but the results are interesting nevertheless.)

In this study, 22 participants were sent to 19 different websites allegedly belonging to a range of well-known banks and other companies associated with on-line financial transactions. Of these, seven were real and 12 were spoofed. The goal was to identify which ones were bogus. Only one site (a real one) was identified correctly by all 22 participants.

All the other sites, real and fake, got a mixture of answers. Eight of the sites (including six spoofed ones) were misidentified by 11 (50%) or more of the participants. In the worst two results, more than 80% of the participants said that a bogus site was real.

The study explains these results quite clearly. It is worth repeating the explanation (or, as the study more conservatively calls it, a hypothesis) because it emphasizes how hard it is for us to be aware of everything we need to take into account when making value judgements on-line, and shows how easy it is for phishers and other on-line fraudsters to exploit this:

> '...Participants made incorrect judg[e]ments because they lacked knowledge of how computer systems worked and did not have an understanding of security systems and indicators. More experienced participants were tripped up by visual deception, e.g. when the address was spoofed or

> when images of the browser [user interface] with security indicators were copied into website content. The study also revealed issues that we did not anticipate [...]:
>
> - Some users don't know that spoofing websites is possible. Without awareness [that] phishing is possible, some users simply do not question website legitimacy.
> - Some users have misconceptions about which website features indicate security. For example, participants assumed that if websites contained professional-looking images, animations, and ads, [then] the sites were legitimate...'

So users may be getting smarter, but there is still a lot that they need to learn and to know.

Q. If we become aware of what this study calls 'security indicators' and can use them reliably, will we be safe? Can the SSL padlock save the day?

A. Secure Sockets Layer (SSL) is very largely the fabric of on-line commerce today. But most people assume that it is simply what it says: secure, which means that too much trust is often placed in the padlock which most browsers display when the SSL protocol is in use. After all, padlock means SSL, and SSL means secure.

In fact, there are a lot of problems with SSL, though fortunately these do not appear to be of the 'flawed cryptography' sort. The problems are a little to do with implementation (or at least with deployment) and a lot to do with use.

Very broadly speaking, SSL provides three main facilities for securing web communications:

- the exchange of digital certificates, permitting each end of the link to establish something about the identity of the other end,
- the secure exchange of session keys allowing for encryption without the need to share key material in advance,
- the encryption of the data in each session, using the keys exchanged above.

When we are banking on-line, the encryption is important, because we do not want others to be able to sniff our account numbers, or to learn how much money we are spending with whom. But the first stage, mutual authentication, is in many ways more important. Without it, we can easily be tricked into engaging in an encrypted conversation with a complete stranger.

Unfortunately, there are many ways in which this authentication can be subverted, or can go wrong. Phishers know this, and so are able to succeed despite, or even because of, the presence of SSL connections and the padlock in your browser.

Q. But if a connection is secure and authenticated, how can it be subverted?

A. There are several different ways in which you can be tricked or misled when making SSL connections, for example:

- By falsified security indicators. A fake website may serve up pages which render in your browser so that they suggest a secure connection. The falsification may range from the trivial, such as displaying a picture of a padlock somewhere on the page, to the sophisticated, where scripts in the page rewrite elements of the browser's user interface to simulate an encrypted site.
- By the use of an illegally acquired certificate. This is uncommon, but not unknown. For instance, in 2001, the world's biggest issuer of SSL certificates, *Verisign*, issued and signed a certificate in the name 'Microsoft' to an individual unassociated with the software giant [10].
- By a worthless certificate. It is easy to produce a self-signed SSL certificate. In this case, you act as your own certifying authority, rather than paying a known third party to do this job for you.
- By a low-quality certificate. Some certification authorities (CAs) issue low-cost certificates, or trial certificates, which make it easy for smaller vendors to enter the market. In some cases the identity checks carried out before issuing these certificates are cursory and almost instantaneous, so the certificates have little value for authentication.
- By malware active on your PC. Malware can suppress security errors, create falsified security indicators, paint over input forms in order to capture or modify your input before it is encrypted by SSL, or otherwise mislead you into how your PC or your browser is behaving.
- By becoming accustomed to starting secure connections from insecure pages. Numerous legitimate on-line financial sites [11] invite you to login from their main (http) page, then take you via some scripting to their secure (https) site. In many cases these insecure pages include padlock imagery, lending credibility to spoofed sites which do the same.

Q. So how can you out-trick such trickery?

A. Fortunately, many phishing tricks are obvious once you know what to look for. In particular, you should familiarize yourself with SSL certificates and how to check them. If you know how your bank usually identifies itself to you, for instance, then you will more easily be able to carry out 'negative authentication' when you need to.

The site http://whichssl.com/, though not as independent as its name might imply (it is run by a certification authority), offers a handy 'test your own site now' link. This takes you to an https site of your choice whilst explaining, in an adjacent browser window, how to use your browser to check the SSL certificate supplied by that site.

Most browsers make an effort to warn you when dubious certificates have been presented, but (as [9] suggests) many users click through these warnings without giving them the attention they deserve. It doesn't help that legitimate sites frequently allow certificates to expire, or publish certificates on one website issued in the name of another, or use certificates which provoke browser warnings which can safely be ignored. This just reinforces risky behaviour.

Q. You mentioned 'negative authentication'. Can't we run community-based databases, like real-time block lists (RBLs) for spam, which help us to identify on-line fraudsters?

A. Several such schemes exist. *Netcraft*, for example [12] offers a browser toolbar add-on through which you can report and identify phishers on-line. *Netcraft* allows ISPs, organizations and the like to utilize its database of known dubious locations on the Internet.

This can be useful in mitigating inbound communications which reference these sites, such as email which tries to persuade you to visit a spoofed website, or to download a piece of malware which the phisher can turn against you later. It is also useful in blocking outbound connections which are aimed at these sites. The blocking can be done by a web filter, an endpoint firewall, a router at the organization's boundary, or in the user's browser.

Microsoft has offered an add-on phishing filter [13] for some time; this has become a built-in feature in *Internet Explorer 7*, currently in its Beta 2 release.

So community-based block lists can help, and it is suggested that they can be very responsive if the community is large and widespread. (If just one person in the entire world reports a phishing site, everyone else can benefit from this knowledge.)

But the phishing criminals can react nimbly, too. For example, using a network of botnet-infected PCs, it would be a simple matter to 'report' that a slew of legitimate sites were bogus. Correcting errors of this sort could take the law-abiding parts of the community a long time, and render the block list unusable until it is sorted out. Alternatively, the community might need to make it tougher to get an Internet site added to the list, to resist false positives. This would render the service less responsive.

Q. You mentioned botnets above, which brings to mind keylogging and other common tricks employed by malware. How are we doing against these threats?

A. A trojan on your PC can succeed without subverting your connection to an on-line service. In fact, many banking-related trojans specifically watch out for you to make a legitimate connection to your bank. (In this case, it may, ironically, be to the trojan's advantage that you check out the bank's SSL certificate closely, thus ensuring that you are connected correctly. If a trojan is intending to manipulate the contents of a transaction, there is no point in doing so when the victim is connecting not to the bank but to a 'service' operated by a rival criminal concern!)

Initially, the most common PC-based attack against banking was indeed the keylogger. The concept is simple: watch for a banking transaction, record the keys typed in (hopefully including account number, password or other personally identifiable information) and later pass those keystrokes to someone outside.

An early response to keyloggers was the so-called virtual keyboard, a script-based or image-based system which requires you to click on pictures of keys using the mouse. Often, the letters or numbers on the virtual keyboard move around randomly each time you visit the site, so that the location of the mouse movements cannot be replayed. Many banks still use this system, believing that it provides additional security.

Malware authors were quick to respond, painting over input forms and popping up virtual keyboard simulators which captured your details before forwarding them to the bank (or, to simplify the programming, before faking an error and forcing you to start again, this time with the trojan allowing your connection to proceed normally).

We can expect this sort of arms race to continue. Unfortunately, the phishers are more nimble than the banks. It might take a bank more than a year to introduce brand new web programming and access control into their on-line systems. After all, change control, correctness and quality are an important part of a bank's IT ethos.

The criminals have no such constraints – and they do not especially care if it is their first, tenth or one hundredth trojan of any new sort which succeeds. The cost of 99 programmatic failures is inconsequential to them; the bank, on the other hand, must succeed at the first attempt.

Q. The malware you describe above relies on capturing information which can be re-used later. Doesn't the hand-held authenticator, or token, make that impossible?

A. No. Or, more accurately, not entirely. What tokens are intended to do is to introduce an unpredictable variable value into the authentication process, instead of a conventional password. This means that any password captured by a trojan cannot be re-used, because each password is designed to be used once, and only once.

This does, indeed, render a lot of current malware impotent. Under some circumstances, however, a trojan can still benefit from capturing a one-time password, for example if it can capture the password before it is used. This may be possible using what is called a man-in-the-middle attack. A handy pictorial summary of a range of such attacks can be found in [14].

Q. Can you give a quick description of how such an attack works?

A. Imagine that you have to play chess against two Grandmasters. (This assumes that you are not a top chess player yourself.) There is a way in which you can guarantee not to get thrashed by both players, provided that you play them both simultaneously, and that you are allowed to play White in one game, and Black in the other.

All you do is wait for your White opponent to move. Then make this move against your Black opponent. When the Black opponent responds, repeat this move against the White player. The two Grandmasters are effectively playing each other. You, the man-in-the-middle, are simply relaying moves between them, although you are turning these moves into what looks like two separate games.

A similar principle applies with a man-in-the-middle trojan. The idea is simple, though the implementation may be complex. The trojan waits for you to begin what you believe to be a transaction with the bank, though you are in fact transacting with the trojan. This means that you mistakenly authenticate against the trojan, and the trojan uses the information you supply – including the one-time password you carefully type in from your token – to authenticate itself with the bank.

The trojan is then free (at least within certain parameters) to alter various aspects of the transaction, such as the amount, the destination account, or any other details of its choosing.

Q. Are there already Trojans which can carry out this sort of attack?

A. Not yet. The main reason is almost certainly that token authentication is not very common in the Internet banking world. This is partly because the expense and complexity of introducing it to every customer is unappealing to the banks, and partly because the need to carry and use a token is still unpopular with many customers. So there has been little need for organized crime to take on the task of writing this more difficult sort of trojan.

Q. When the criminals are forced to confront stronger authentication, how hard will they find it?

The criminals may not need to subvert the authentication process at all. Instead, they may simply come up with new ways of tricking you out of your money. Spammers, for example, already know how to conduct on-line fraud without getting hold of your account number or password. Many spammers operate by persuading you to conduct a transaction willingly and overtly, using your hand-held authenticator if you have one, and then supplying sub-standard goods, or nothing at all, in return.

Now imagine how much easier it would be for criminals to seduce you into bogus transactions if they had a complete picture of your spending habits. For example, if they knew you paid your rent on the seventh of every month, and which agency you paid it to, they could attempt to phish you into paying it into a different account. And before you respond by saying, 'but it's such a big step to start paying bills to a new recipient, so that would simply never work', remember that it sounds just as far fetched to believe that users would willingly go and type in their personal banking credentials into an unknown website on the say-so of an email which could have come from anywhere, and probably did.

The technology to allow outsiders to keep detailed track of your secure on-line activities, including everything you buy, and when, and where, already exists.

One example is the application *Marketscore*, created by the market research company *comScore Networks*, *Inc*. In return for a modest payment for participation, users joined the 'Marketscore Panel' and installed the *Marketscore* application. Amongst other features, *Marketscore* incorporated what is effectively a man-in-the-middle SSL proxy which aimed to crack open and to monitor all your secure on-line transactions, sending data about everything you bought, and how much you paid for it, back to comScore.

Q. Surely a legitimate application wouldn't go quite that far?

A. *ComScore* is no longer distributing *Marketscore*, perhaps due to the publicity it received when some American universities decided to block it outright, despite the strongly held tradition of academic freedom on their networks [15]. But here is what *comScore* themselves [16] have published about its behaviour:

> '...[C]omScore has recruited for the Marketscore Panel over one and a half million opt-in members who have agreed to have their Internet behavio[u]r confidentially monitored and captured on a totally anonymous basis. These members give comScore explicit, opt-in permission to confidentially monitor their online activities in return for valuable benefits [...].
>
> Those individuals who choose to be part of the Marketscore Panel [...] download comScore's technology to their browser where it unobtrusively routes the member's Internet connection through comScore's network of servers [...]. The technology allows comScore to capture the complete detail of all the communication to and from each individual's computer – on a site-specific, individual-specific basis. Information captured on an individual member basis includes every site visited, page viewed, ad seen, promotion used, product or service bought, and price paid.
>
> [...]
>
> It is extremely challenging, even with a consumer's opt-in permission, to capture information communicated to and from a browser in a secure session (e.g. any purchase transaction). In order to do this successfully, technology is required that "securely monitors a secure connection". [C]omScore's patent-pending technology does this at no incremental cost to comScore or risk to the panelists...'

As dubious as this may sound, remember that some security products provide gateway-based tools to open and examine SSL connections out of a network. Whilst this is culturally rather different to placing a market-research-oriented SSL proxy on every PC, it is technically and functionally similar. Like many technologies, whether it is good or evil depends on how it is used, and who is using it.

Q. Let's return to where we started, namely the subversion of the endpoint via malware and potentially unwanted applications. Will improvements in operating system security help prevent users being 'marketscored' by criminals?

A. There is a long answer to that, in which we could look at some of the new features of *Windows Vista*, such as User Access Control, which tries to restrict the subversive use of the administrator account, and at the features of *SELinux*, which does away with the idea of an all-powerful account completely.

The short answer points out that operating systems are becoming more resistant to trivial exploitation, but reminds us all that there are still two important risk vectors:

- Users and administrators who make errors of judgement, and who carry out fully-authenticated installations of risky or inappropriate software. *Vista*'s warning that 'this operation requires elevation', and its careful display of a program's digital certificate (or lack of it), for example, can be undone with a single mouse click to authorize the offending operation.
- Organized crime and the counterculture, who have shown a willingness to invest considerable amounts of time in probing even the most secure systems for tiny cracks into which they can drive a subversive wedge. Additionally, they are nimble enough to respond to technological changes, such as their subversion of virtual keyboards, in weeks or even days, a luxury which security professionals cannot afford.

Q. So can we win? And is authentication the key component to staying ahead of the phishers, even though it cannot solve the whole problem?

A. Some say that we can, and it is. For example, researchers from a Swiss financial institution and IBM [17] have proposed an on-line banking authentication system which sounds very secure.

Briefly summarized, the system relies upon an external smart card reader, with a numeric keypad and a small display. The cryptographic computations for authentication and security between the user's browser and the bank are offloaded to the smart card (which is tamper-resistant and contains an operating system and software of its own); the entry of passwords and one-time codes is offloaded to the card reader's keypad (where they cannot be sniffed or altered); and each transaction is confirmed cryptographically after its details are shown on the card reader's display (where they are not subject to manipulation by malware writing on top of data on the screen).

Of course, this system is complex, which means it will be hard to implement correctly; it is comparatively expensive, which will slow down its adoption by the banks; and it is inconvenient, which will slow down its acceptance by users.

Also, phishers currently target our banking credentials so that they can later masquerade as us in order to raid our accounts. They do this because they can, because it is easy, and because it works. As we have seen, making this harder, or even impossible, is unlikely to stop phishing. The phishers will respond by attacking and subverting other parts of our on-line lifestyle.

This doesn't mean that we should ignore technological advances in computer security, any more than we should throw out the seat belts, the airbags and the crumple zones from the modern automobile. But it does mean that we need to keep ourselves informed and vigilant when we spend money on-line, just as we are encouraged to be safer and more responsible drivers on the road.

REFERENCES AND FURTHER READING

[1] Pullum, G. K. Scooping the loop snooper (an elementary proof of the undecidability of the halting problem). Mathematics Magazine, October 2000.

[2] Cohen, F. Computer viruses: theory and experiments. Computers and Security 6, 1987.

[3] Shipp, A. Targeted trojan attacks and industrial espionage. Proceedings of the Virus Bulletin 2006 conference.

[4] VirusTotal, http://www.virustotal.com/, June 2006.

[5] Anderson, R. Why cryptosystems fail. Communications of the ACM 37(11), November 1994.

[6] Rivner, U. Security and Usability. http://www.rsasecurity.com/, June 2006.

[7] Fluhrer, S., Mantin, I., Shamir, A. Attacks on RC4 and WEP. Cryptobytes, 2002.

[8] Friendly Greetings? You might be spreading unwanted email – Sophos advises on new email nuisance. http://sophos.com/, October 2002.

[9] Dhamija, R., Tygar, D., Hearst, M. Why phishing works. Proceedings of the SIGCHI conference on human factors in computing systems, January, 2006.

[10] Fonseca, B. VeriSign issues false Microsoft digital certificates. Infoworld, March 2001.

[11] Herzberg, A., Gbara, A. TrustBar: protecting (even naive) web users from spoofing and phishing attacks. Cryptology ePrint Archive, 2004.

[12] http://toolbar.netcraft.com/, June 2006.

[13] Microsoft Phishing Filter at a Glance. http://microsoft.com/, June 2006.

[14] Wüest, C. Phishing in the middle of the stream – today's threats to online banking. Proceedings of the AVAR 2005 conference.

[15] Shuster, S. Blocking Marketscore: why Cornell did it. IT Security, Cornell University, 2005.

[16] An ARF methodological review of comScore Networks, Inc. netScore. Advertising Research Foundation, 2001.

[17] Hiltgen, A., Kramp, T., Weigold T. Secure internet banking authentication. IEEE Security & Privacy, March/April 2006.

[18] Online fraud update – beware, don't be caught. June 2006. http://www.absa.co.za.

[19] Biancuzzi, F. Phishing with Rachna Dhamija. http://www.securityfocus.com/, June 2006.

[20] Dhamija, R., Tygar, J. D. The battle against phishing: dynamic security skins. Proceedings of the ACM symposium on usable security and privacy, 2005.

[21] Citibank adds on-screen keyboard to Web banking site. http://finextra.com/, February 2005.

[22] Microsoft security bulletin MS02-050 - certificate validation flaw could enable identity spoofing (Q329115). Microsoft TechNet, November 2003.

[23] Miller, R. Spoofing flaws for Firefox, Mozilla and Opera. http://news.netcraft.com/, August 2004.

[24] Morar, J., Chess, D. M. Can cryptography prevent computer viruses?. Proceedings of the Virus Bulletin 2000 conference.

[25] Stubblefield, A., Ioannidis, J., Rubin, A. D. Using the Fluhrer, Mantin, and Shamir attack to break WEP. Proceedings of the 2002 network and distributed systems security symposium.