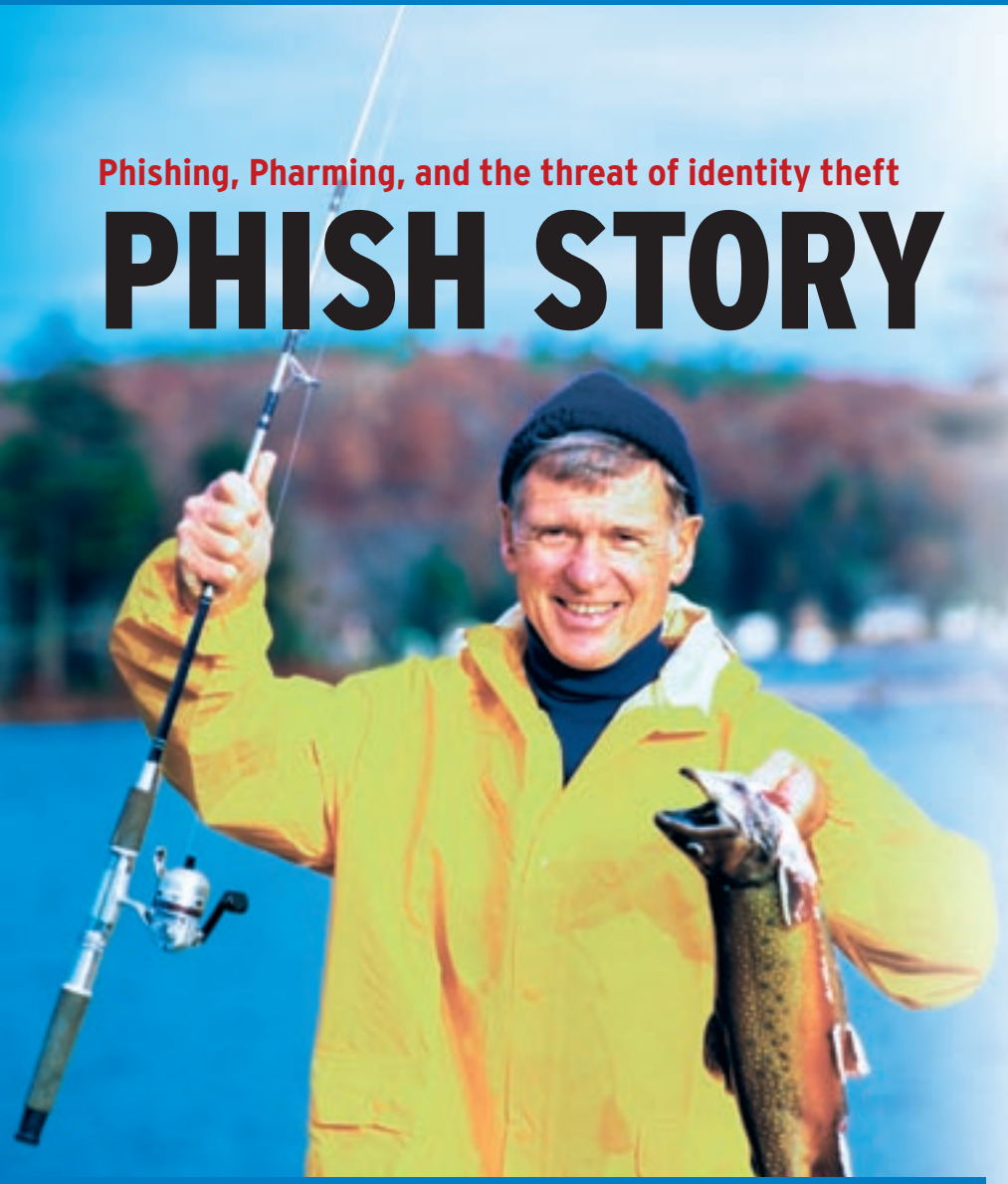
Phishing, Pharming, and the threat of identity theft

PHISH STORY



The pharmers and phishers are after your precious financial information. We'll show you how to protect your interests.

BY CHRISTOPH WEGENER AND RALF SPENNEBERG

Phishing is the art of tricking unsuspecting Internet users into giving up personal and financial information. This nefarious game has brought big rewards to a new generation of cyber con men. But phishing is only one of several tools in a bigger bag of tricks. This article examines some popular techniques for identity theft and shows what you can do to stay ahead of the threat.

Phishing and Pharming

Phishing uses tricks like spoofed email to tempt unsuspecting users into visiting rogue sites, where they are asked to enter personal data such as passwords or PIN numbers. Phishing attacks occur in two phases: in Phase One, the user is tricked into visiting the attacker's web server. Attackers use various initial vectors to attract victims. In Phase Two, the user is prompted to enter the personal data. This part of the attack is often referred to as visual spoofing. The best known initial vectors are email spoofing and cross site scripting (XSS), where the attacker misuses the website of a trusted third party to initiate the attack. In all of these cases, the attack relies on the victim playing along and would fail without the victim's compliance.

In contrast to phishing, a pharming attack involves the attacker poisoning the DNS cache entries on a vulnerable DNS server, and then redirecting users that rely on this server to a rogue server that hosts the pharming site. This attack typically relies on errors in the DNS server implementation to inject fake IP/hostname pairs into the server's domain cache. The important thing is that the pharming attack is purely technical. This considerably improves a pharming attack's chances of succeeding, as it removes the uncertainty caused by relying on human compliance.

Although the issue of identity theft on the Internet is not new, there has been a definite increase in recent months. Over a year has now passed since the market research experts, Gartner [1], dedicated a survey to this spectacular topic. More recent research from June last year clearly shows the economic impact of phishing and pharming. The researchers noted a clear loss of consumer trust in online business, leading to increased cost and investment risks both for corporations and for consumers. And, last but not least, the Anti Phishing Working Group (APWG) has noted a constantly high level of phishing attacks and a considerable increase in crimeware since April 2005: twice as much malevolent code was detected between April and July 2005 [2].

These dangers are not just hypothetical; they cause genuine damage. Ten percent of those participating in a recent survey stated that they had suffered financial loss due to phishing emails. Although these figures may not be authoritative, they definitely show how real the dangers are.

In the Net

As we mentioned earlier on, phishers spoof email messages from well-known online service providers to trick victims into visiting rogue websites. Once the victim goes to the rogue site, the phisher attempts to trick the victim into revealing login data, PIN/TAN numbers, or credit card data. Phishing attacks mainly target major banks and eCommerce websites such as Citibank, PayPal, or Ebay. A typical phishing message is shown in Figure 1.

In contrast to many less polished phishing mails, the user is not warned by obvious spelling or grammar mistakes. Mistakes of this kind often give you a clue, as many phishing messages are created by non-native speakers.

A victim who clicks on the link is taken to the attacker's website, which typically looks very much like a real site (Figure 2), except that phishing sites


Dear eBay User,
During our regular update and verification of the accounts,
we couldn't verify your current information.
Either your information has changed or it is incomplete.
If the account information is not updated to current information
within 5 days then, your access to bid or buy on eBay will be suspended.
go to the link below,
and re-enter your account information.

Click here to update your account.
Please Do Not Reply To This E-Mail As You Will Not Receive A Response
Thank you
Accounts Management

Copyright© 1995-2005 eBay Inc.

Figure 1: A phishing email message asks the user to update account information.

often contain spelling or grammatical errors, since these sites are often created by international criminals who use automatic translation tools. If the victim really does enter the data the site asks for, the attacker can use this data to transfer funds or buy and sell on Ebay. The attacker steals the victim's identity.

Your best protection against phishing is common sense in combination with Spam Assassin. Potential victims can identify spoofed emails and websites just by checking the URL in their browsers, and by the fact that the connection is not typically encrypted – the padlock icon in the bottom right corner of the browser window is open. But some rogue websites have actually started using SSL. The browser checks the certificate and warns the user that the selected site does not match the site for which the certificate was issued, or does not come from a trustworthy organization, but many users can't cope with these warnings and just ignore them. This just leaves the URL as a means of identifying the scam.

Insecure, Old MSIE

A vulnerability in older versions of the Microsoft Internet Explorer additionally lets attackers spoof the URL bar, the SSL padlock icon, and the certificate view. In this case, the padlock is closed, the certificate data appears to be okay, and the URL bar shows you the right URL – a perfect spoof. The only protection against this is to update your version of MSIE, or personalize the existing version of the browser to make visual spoofing more or less impossible.

Spam Assassin can be useful in detecting spoofed emails. A recent version of Spam Assassin will typically detect many typical phishing mails. Filtering incoming messages with Spam Assassin can thus go a long way to protecting you against known phishing attacks. Of course, not even Spam Assassin can protect you against previously unknown phishing messages.

The bad guys earn considerable amounts of money with phishing attacks. The perpetrators are often organized groups of criminals. InternetNews.com estimates the financial damage caused in the USA in the year 2003 at around $US 1.2 billion.

Phar More Promising

As phishing relies on compliance, and as the human factor can endanger the success of the attack, malevolent hackers have devised methods that tilt the odds in their favor. Pharming does not rely on an email to lure the victim to a rogue site; instead, a comprised DNS server takes the victim right there. Users don't typically type their bank's IP addresses in their browsers; instead they type the DNS hostname. The Internet-based DNS service then resolves the address, and it is the process of address resolution that the pharmer exploits.

The attack is fairly straightforward on Microsoft Windows systems using a virus that modifies the *System32/drivers/etc/hosts* file. If users enter DNS hostname/IP address pairs in the file, the browser goes straight to the IP address when a user attempts to surf to the hostname. Of course, the IP address points the victim to the attacker's rogue website. The victim typically feels quite secure about this, after all, they weren't taken to the site by an email, but entered the address themselves.

To prevent this attack, security experts recommend write-protecting the hosts file. But as Trojans commonly assume admin privileges, this kind of protection is typically useless. Also, more recent variants use a different vector, entering a rogue DNS server, run by the pharmer, in the machine's network settings. In this case, write-protecting the hosts file will not help at all.

DNS Spoofing

An alternative vector involves the attacker logging the DNS lookup request and responding instead of the bona fide DNS server. To do so, the attacker must be somewhere in the victim's vicinity to actually sniff the name resolution request. Of course, there are many tools capable of this. Linux has the legendary Dsniff tool by Dug Song, for example, with its *dnsspoof* command. Windows even has a GUI-based tool dubbed Cain&Abel [3]. But there are any number of tools that support this kind of attack. Attackers can also use ARP spoofing to sniff DNS responses or compromise SSL sessions using Man-in-the-Middle (MitM) techniques.

Poisoned Cache

However, the most effective approach for a pharmer is to poison the DNS server's cache. Although this vulnerability has been known for years, you would be surprised how many many vulnerable DNS servers there still are. Dan Kaminsky investigated around 2.5 million servers in July 2005, and ascertained that about 10 percent of current DNS servers are open to poisoning attacks.

In a cache poisoning scenario (Figure 3), the attacker first looks up a name for which their own name server is authoritative (such as *www.hackingexample.com*) on the vulnerable, caching DNS server (1). The caching DNS server will not have this information, and so it turns to the pharmer's DNS server (2) for help. The pharmer's DNS server responds to the request, at the same time passing its

own address entries for the DNS names *www.anybigbank.com* and *www.yetanotherbigbank.com* (3). The caching DNS server adds this information to its own cache, along with the IP address for *www.hackingexample.com*. The name server then responds to the request for *www.hackingexample.com* (4).

Now, when a victim queries the compromised DNS server for *www.anybigbank.com* or *www.yetanotherbigbank.com*, the DNS server will find the requested data in its own cache and not turn to the authoritative name server for the address. The server thus passes the fake IP address, provided to it by the attacker, on to the victim. All the pharmer needs now is a web server with a rogue site running at the IP address they have slipped to the victim.

Name chaining attacks are a more advanced version of cache poisoning. They involve the attacker linking the DNS lookup itself and the additional section in an unusual way. The response will not contain an IP address, but a pointer to the hostname of the redirecting server. In our previous example, the answer would be something like "*www.hackingexample.com* is an alias for *www.anybigbank.com*," and as an additional section, "www.anybigbank.com resolves to 192.0.2.1." There is no way of preventing the attacker from injecting spoofed information by checking if the *Additional Section* entry matches the lookup request.

Whenever a new security hole is revealed in a DNS server software, firewall, or similar product, pharmers step up to exploit it. And this is exactly what happened in March 2005 with the Symantec Firewall. Details of a DNS cache error in this product were revealed in June 2004 [4], but not all installations had been patched by March the next year. This in turn made a large-scale DNS cache poisoning attack [5] possible.

A variant on this attack attempts to manipulate the client-side cache on a Windows or Linux machine, rather than targeting the server cache. To achieve this, the attacker uses the following approach: first, the attacker sends a harmless looking email to the victim; the mail contains an image that points to the *www.example-attacker.com* domain. The client has to query the authoritative name server to discover the IP address, and this is where the cache poisoning attack starts. The DNS server not only provides an entry for *www.example-attacker.com*, at the same time it poisons the client PC's cache directly, slipping a fake IP address for *www.anymajorbank.com* to the client. The next time the client attempts to access the bank website, the unsuspecting user is beamed to the rogue website.

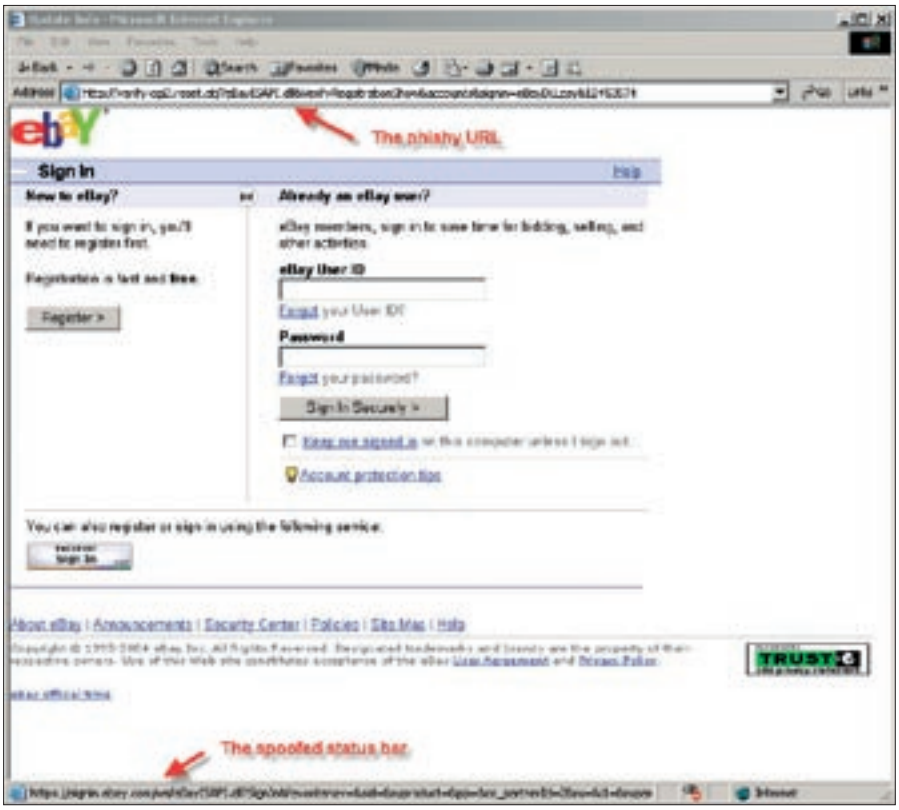


Figure 2: A user who clicks on the link in Figure 1 goes to a webpage that closely resembles an eBay sign-in page – but what is it really?

mTAn, iTAN, eTAN

After many years with the PIN/TAN method for electronic transations, banks are now trying to improve the security of online banking with a new approach to TANs (transaction numbers). With the iTAN method, the bank customer is given an indexed TAN list. When the customer attempts to transfer funds, the bank requests a TAN with a specific number. If the customer ends up on a pharming page, the pharmer will not know which TAN to request. The iTAN approach is now used by many banks and building societies, and more are planning to introduce it.

The eTAN method involves giving the customer an electronic device. When the customer wants to transfer funds, the bank sends the customer a random number; the customer has to enter the number into the device, which looks something like a calculator. The device then calculates a customer-specific response which the customer sends back to the bank to authorize the transfer. The eTAN method is used by the GE Money Bank, for example.

The mTAN method uses a customer's cellphone to authorize transactions. When a customer wants to transfer funds, the bank sends a short message with a TAN number and some additional information to the customer's cellphone. The customer checks the details of the transfer, and enters the TAN to authorize the transaction. The TAN is only valid for one transfer.

The iTAN method is insecure if the attacker does not store the data on a rogue website but forwards this data to the right bank, like some kind of transparent proxy. All the attacker has to do is to modify the amount and account details before passing the data on to the bank. The bank then requests the iTAN from the attacker; the attacker passes the request straight to the victim, who kindly provides the correct iTAN.

The mTAN method seems to be the only genuinely secure method; at pres-

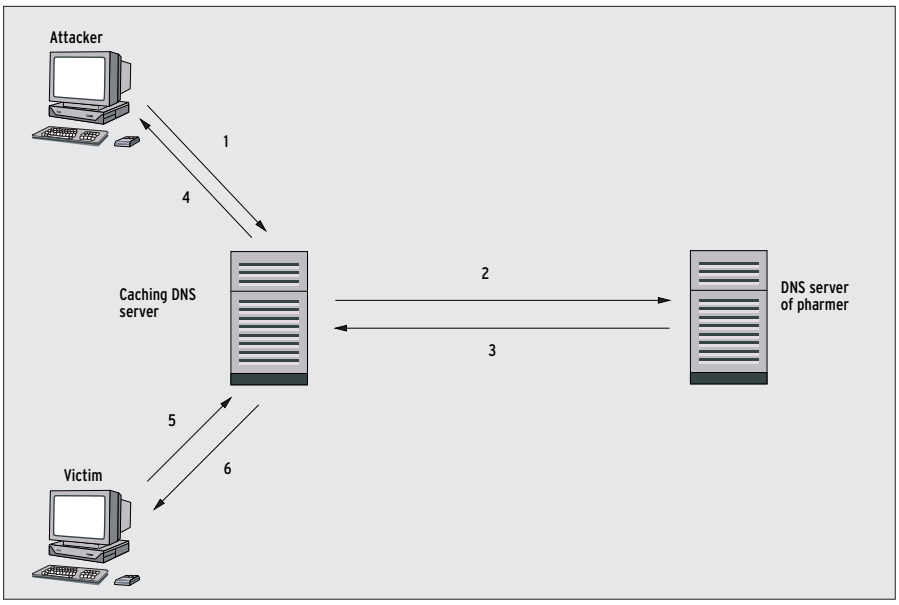


Figure 3: To poison the DNS cache, attackers need to set up their own name servers.

ent you need an iTAN to enable the method. The customer selects the mTAN approach, and the bank sends a registration key to the customer's cellphone for the customer to check. The customer then has to enter the registration key in a form, and is prompted to enter an iTAN to enable the mTAN method. Existing mTAN cellphone numbers can be deleted without additional verification.

Unfortunately, it seems unlikely that this system will survive, as mTANs add cost to the transaction. Add to this the fact that the future of the mTAN seems insecure in an age characterized by Bluetooth hacking, cellphone viruses, and Tojans. And let's not forget that there is always the danger of losing your phone.

HBCI Is Secure

HBCI is the only approach to provide genuine security at this time of writing. However, as HBCI is quite complex and implies financial overhead for additional hardware in the form of smartcards and card readers, not many banks support it, and fewer customers use it. Customers do not use PINs and xTANs to authenticate here, but X.509 certificates, and the private keys to match; at the other end of the connection, the bank's web server needs the same credentials to authenticate against the user. To prevent an attacker from just breaking into a victim's home and stealing the victim's private key, the key is stored on a smartcard, which additionally protects the key against unauthorized read access.

To be able to use the private key to sign transactions, a user always needs a smartcard. To prevent an attacker who gains possession of the smartcard from misusing the smartcard, the key on the smartcard is additionally PIN protected. Customers are required to enter the PIN to authenticate with the smartcard. To thwart keyboard loggers and Trojans that attempt to sniff the PIN off the customer's machine, Class 2 smartcard readers have integrated pinpads that let users type in the PIN directly.

However, the message to be signed is often displayed on the PC at this point, to allow the customer to verify it. And this can open up an attack vector to a Trojan sitting between the card reader and the PC display. In this case, a user might sign transactions they did not intend to sign. Class 3 smart card devices provide genuine security. Again the text to be signed should never be displayed on the PC (because it is too long for the card reader to display, for example), as this would again compromise security.

In online banking applications, critical data, such as the target account number, the bank identification code, and the amount, could be displayed on the card reader, thus giving customers a solution where they could be sure of what they signed.

Protection via SSL Client Certificates

All of these approaches ignore one problem: in legacy applications, the bank server authenticates against the customer. Customers have to make sure that they are talking to the right server, and to do so, they need to be able to distinguish a spoofed SSL connection from a legitimate connection.

Day to day experience suggests that depending on the customer does not always solve the problem: for one thing, users tend to click to remove error messages without reading them. An error message is no surprise to a customer. For another thing, customers often lack knowledge of the way a SSL connection works. Various attempts by banks to educate their customers in this respect have not always come up with the expected results. Some policies actually give the customer a false sense of security, such as displaying the SSL certificate fingerprint on the unprotected bank homepage.

One interesting approach to providing more protection against phishing and pharming attacks is to reverse the authentication direction and to force the customer (or the customer's PC or browser to be more precise) to authenticate against the bank. Banks could use client-side SSL certificates to do this. Banks would issue a certificate to a customer after checking the customer's legitimation (this could take place at a local branch of the bank), and the customer would install the certificate on their own PC. Customers would need to present this certificate in order to talk to the bank server. The certificate itself could be PIN-protected against unauthorized use.

To avoid attackers visually spoofing the bank server identity, visual authentication techniques could be introduced. The bank homepage would expect some kind of visual identification from the customer, which would be transmitted via a secure channel; attackers would find it extremely difficult to fake the visual customer ID.

One issue with this approach is that the typical customer's online behavior would collide with this type of procedure. Bank statistics reveal that most bank transactions take place on customers' lunch breaks using their office desktops. Some environments would not allow this to happen, as users typically have restricted privileges for their office desktops. A USB token might solve the

problem, but again this would involve some financial overhead.

Legal Situation

Phishing and pharming raise legal issues in various fields. One important question is, who is actually responsible for the phishing transaction? The bank may try to claim compensation from the phishing victim for funds transferred by the phisher. A bank transfer works like this: the bank asked to transfer funds (the phishing victim's bank in this case) transfers a sum of money to the attacker. The bank will then want to take money from the phishing victim's account as reimbursement for this transaction. Even if it is clear that the transaction does not have the victim's approval, the bank may claim damages in the amount of the disbursement. The bank may then have to prove its case in court; in other words, the bank must prove that the customer ordered the transfer, and typically it will be unable to do this.

Of course, all such proceedings depend on the local laws and regulations. At a minimum, even if the victim succeeds in avoiding having to pay for the money stolen by the phisher, the event ends in a long flurry of letters, threats, and bureaucratic wrangling.

Conclusions

It remains to be seen when the first big wave of pharming attacks will sweep over the country, but considering the fact that a large proportion of name servers world wide are vulnerable to cache poisoning, the danger is very real and hard to predict. The important thing for users is to think before you leap on the Internet, especially if you are asked to provide personal data. Banks, online auction platforms, and web shops will all need to work on promoting more security consciousness among their customers. This is probably the only way to stop large-scale misuse of online payment and banking systems. ■

INFO

[1] Gartner Group: *http://www.gartner.com*

[2] Anti Phishing Working Group (APWG): *http://www.antiphishing.org*

[3] Cain&Abel: *http://www.oxid.it*

[4] Vulnerability in the Symantec Firewall: *http://securityresponse.symantec.com/avcenter/security/Content/2004.06.21.html*

[5] DNS caching attack: *http://isc.sans.org/diary.php?date=2005-03-04*

THE AUTHORS

Ralf Spenneberg works as a freelance Unix/Linux trainer, consultant, and author. Ralf's company, OpenSource Training offers coaching and consultancy services. Ralf has published various books on topics such as Intrusion Detection and Virtual private networks. His new book "Linux Firewalls with Iptables & Co" will be published in just a few weeks.

Christoph Wegener has a Ph.D. in physics, and is Head of Business Development with gits AG. He has been a freelance Linux and IT security consultant for many years, and is the author of various publications.