Phishing Detection

Sven Strickroth

January 10, 2008

Contents

1 Abstract

Phishing[1], from "password" and "fishing", is an attempt to get sensitive information, like usernames, passwords, credit card information or complete identities from the recipients. This is based on social engineering by masquerading a trustworthy entity in electronic communication and fooling users to submit their data by the use of special techniques, mostly in emails with hyperlinks to forged websites which look nearly like the original ones. Also mails with a faked phone number instead of a hyperlink have been seen to get the personal data. Common targets are banks, auction portals or payment services.

Since it constitutes one of the biggest threats on the internet, phishing causes severe damages. Between May 2004 and May 2005 approximately 1.2 million computer users in the USA lost about $929 million – US business loses about $2 billion annually[2]. In Germany the frequency of phishing frauds rose by 23 per cent from 2006 to 2007 – the same as for the first half of 2007. Officially, the phisher gangs acquired about €13 million, with an average loss of €4700 per victim[3].

In general a phishing attempt pretends that there is some (bigger) problem with the account or some service of the victim and urges the victim to react quickly on the issue (for instance, the phisher uses some deadline for the user. See an example and more details on page 6). Commonly used problems include software updates, data loss or an account fraud and so the customer has to update/check his personal data. The phisher hopes that the recipient will be "shocked", will not check the rest of the message and follow the steps described to send his personal data to the phisher.

[1] based on [14]

[2] see [2]

[3] see [4]

2 Analyzing phishing mails

2.1 Statistics

As indicated by publically available statistics and the author's own experience, phishing mails are among the top 10 for malware detected on mail servers. In correpondence with the mail server and time (number of malware/phishing outbreaks at that time) the phishing/malware ratio shows that phishing mails can be (more than) 40 per cent of the malware caught:

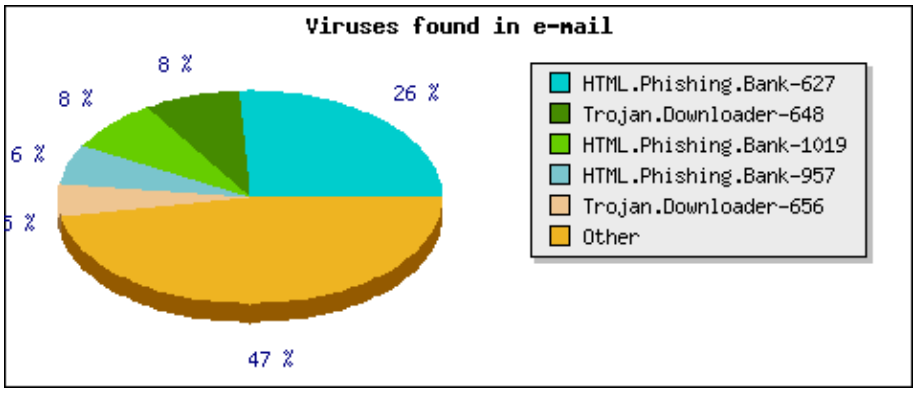


Figure 2.1: Chart: Malware-Phishing ratio

Chart taken from http://noc.bit.nl/mail/virus/virus-list.php?month=2007-01-01.

This report is based on $>$ 15 000 phishing mails, and so the following statistics are not necessarily representative; however, they should give a rough idea of what phishing mails look like in general:

- **99 per cent** of phishing mails included a "http://"-URL
- **95 per cent** of phishing mails used "content-type: text/html"
- **22 per cent** of phishing mails referred to included images with the use of the "cid:"-url[1]
- **13 per cent** of phishing mails used an IP-address in a "http://"-URL
- **3 per cent** of phishing mails included a HTML-script section
- **2 per cent** of phishing mails were addressed to "undisclosed-recipients"
- **2 per cent** of phishing mails tried to overwrite the statusbar on "mouseover" of the link-tag
- **2 per cent** of phishing mails used another port than 80 in an "http://"-URL
- **2 per cent** of phishing mails included a HTML-form

[1] as described in [3]

The average size of a phishing mail is about 7 KiB, the biggest phishing mail I have seen had about 65 KiB (HTML created by Microsoft Word).

- **99 per cent**: phishing mail size $\leqslant$ 20 KiB
- **75 per cent**: phishing mail size $\leqslant$ 10 KiB
- **58 per cent**: phishing mail size $\leqslant$ 5 KiB

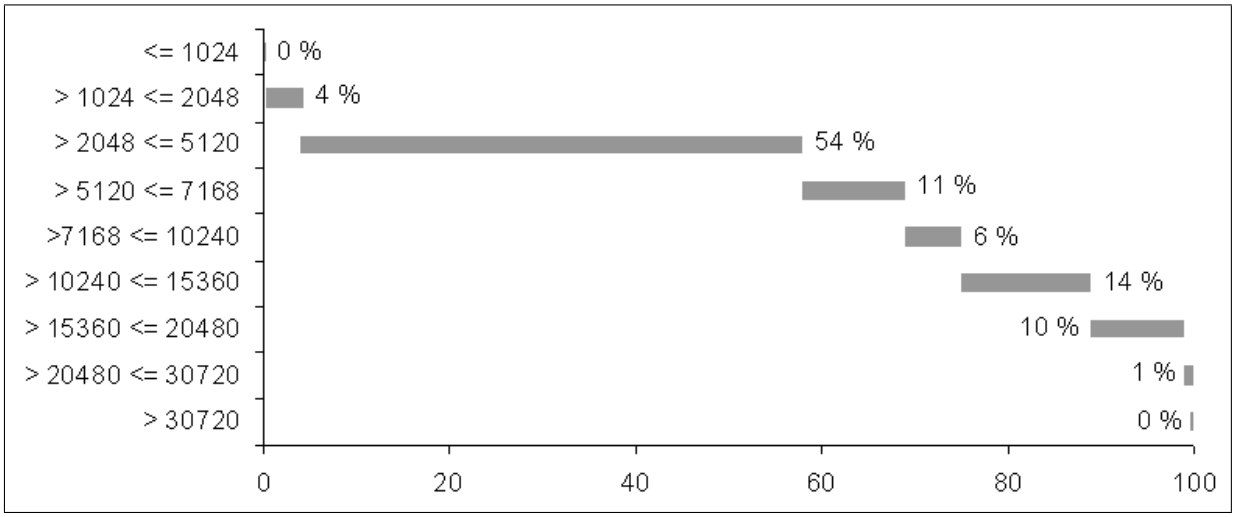


Figure 2.2: Chart: Distribution of mail sizes (in byte)

per cent	claim to be created by
41.6	Microsoft Outlook Express 6
21.3	no User-Agent/X-Mailer header
3.9	The Bat!
3.9	MIME-tools
3.4	Microsoft Outlook
3.0	Internet Mail Service (5.5.2650.21)
2.5	Calypso
1.7	Mozilla Thunderbird
1.6	Microsoft Outlook Express 5
1.4	Mozilla/Netscape
1.4	Pegasus Mail for Win32
1.3	SmartMailer
1.3	Sylpheed
1.3	MailGate
1.2	Microsoft Internet Mail
1.2	PObox II
1.1	Mutt

2.2 Content of phishing mails

In the following text I will disregard analysis of the email header, because in this case phishing-mails are similar to SPAM mails which are sent through open relays, infected computers or bot networks, too. Therefore, I will consider the content in general, focus on the mail body, and then take a more detailed look at the URLs contained in the mails in the next section.

As already mentioned in the abstract (page 3), phishers usually pretend that there is a regular check or a problem with the bank- or user-account/credit-card/personal data and that the user needs to take action in order to correct/check this. For this purpose phishers often urge the victims to act quickly by setting some deadline and threatening the victim with severe consequences, like deleting or closing the account. Also seen were mails containing very high invoices from popular companies. Another way of getting the user's sensitive data is to say that the victim has won a price or other money and that the sender now needs the address or account details for transferring the money. Thus, this knowledge can be used for a human being for roughly rating/evaluating emails, besides the design of the email, whether they might be phishing.

At the beginning of phishing, the readable content, the mail text, was written in bad English (for instance, unusual wording, same for phishing in different languages with bad translations, cryptic characters instead of umlauts, punctuation mistakes) or included a lot of spelling mistakes and did not fit into the corporate identity/design of the vendors (as shown in figures 2.4 and 2.5. In all official HTML-mails paypal uses colors, logos and footers which look like their website). Gradually, the phishers have been improving the quality of their mails, but sometimes they still use spelling mistakes deliberately in order to bypass content or bayesian filters. – However, some phishing mails still do not fit into the corporate design and do not look like an official, legitimate mail.

In order to improve the trustworthiness of their mails, companies include personal data into these, for instance, by including the name of the recipient/customer (see figure 2.13 on page 13). – Phishers have circumvented these steps by replacing some of these "proofs of authenticity": "XX send this message to John Doe" became "XX send this message to you@yourprovider.tld" (which always applies to for any sender and does not say anything about the authenticity of a mail, because the sender always knows the email address of the recipient), "XX sent this message." (without any name/mail address) or "XX send this message from john doe (jdoe)" (figure 2.3 on page 6). – By using bought/captured identities, it is possible for phishers to include the correct real name of the recipient.

Concentrating on the readable content, phishers use the same tricks SPAMMers use: Mails with one picture followed by random text to fool and poison bayesian filters, similar to the one shown in figure 2.7 on page 10.

When comparing the texts of older and new phishing mails, it appears that phishers like to stick

Figure 2.3: Phishing example: "eBay sent this message from ..."

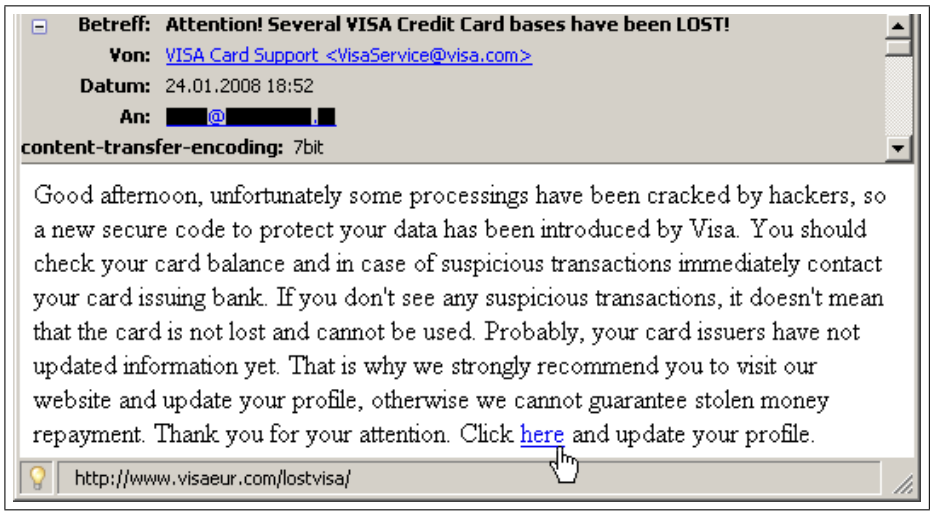


Figure 2.4: Phishing example: early VISA phishing

Dear PayPal customer,

It has come to our attention that when logging into PayPal, you or somebody else have made several login attempts and reached your daily attempt limit. As an additional security measure your access to PayPal will be limited in the next 24 hours if you do not verify your identity.

https://www.paypal.com/cgi-bin/webscr?cmd=_verification

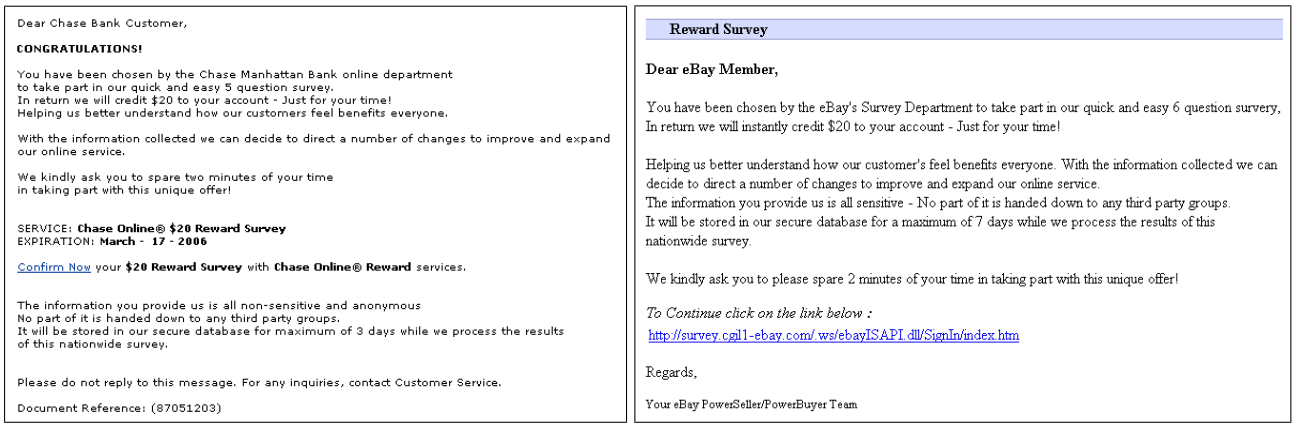
Please verify your details clicking on the link above, before trying to log in again. You will be able to attempt logging into PayPal account immediately after you veryfy your identity without any limitations applying to your account.

Sincerely,
PayPal Account Review Team
PayPal, an eBay Company

Figure 2.5: Phishing example: early PayPal phishing

to specific, proven text fragments like "...if you recently accessed your account while traveling, the unusual log in attempts may have been initiated by you...". At first, this phrase appeared in paypal scam mails, but this text can now be found in all kinds of scam mails. Take figure 2.6 on page 8 as an other example of how specific paragraphs are still used more than one year later.

From the user's point of view, both texts seem to be the same, but looking at the HTML source, you can see, that the two samples are quite different:



Dear Chase Bank Customer,

CONGRATULATIONS!

You have been chosen by the Chase Manhattan Bank online department
to take part in our quick and easy 5 question survey.
In return we will credit $20 to your account - Just for your time!
Helping us better understand how our customers feel benefits everyone.

With the information collected we can decide to direct a number of changes to improve and expand our online service.

We kindly ask you to spare two minutes of your time
in taking part with this unique offer!

SERVICE: **Chase Online® $20 Reward Survey**
EXPIRATION: **March - 17 - 2006**

Confirm Now your **$20 Reward Survey** with **Chase Online® Reward** services.

The information you provide us is all non-sensitive and anonymous
No part of it is handed down to any third party groups.
It will be stored in our secure database for maximum of 3 days while we process the results of this nationwide survey.

Please do not reply to this message. For any inquiries, contact Customer Service.

Document Reference: (87051203)

Reward Survey

Dear eBay Member,

You have been chosen by the eBay's Survey Department to take part in our quick and easy 6 question survery,
In return we will instantly credit $20 to your account - Just for your time!

Helping us better understand how our customer's feel benefits everyone. With the information collected we can decide to direct a number of changes to improve and expand our online service.
The information you provide us is all sensitive - No part of it is handed down to any third party groups.
It will be stored in our secure database for a maximum of 7 days while we process the results of this nationwide survey.

We kindly ask you to please spare 2 minutes of your time in taking part with this unique offer!

To Continue click on the link below :
http://survey.cgi11-ebay.com/.ws/ebayISAPI.dll/SignIn/index.htm

Regards,

Your eBay PowerSeller/PowerBuyer Team

Figure 2.6: Phishing example: Survey-Scam

The example on the left was received in March 2006, the one on the right in July 2007

```
[...]questions survey.</font></p><p
style="margin-top: 0; margin-bottom:
0"><font face="Arial"><b><font
color="#000066" style="font-size:
60%; background-color:#FFFFFF"> 
</font></b><font color="#000066" style=
"font-size: 80%; background-color:
#FFFFFF">In return we will credit<b>
</b></font><b>[...]
```

```
[...]question survery,</SPAN></DIV>
<DIV align=left><SPAN class=style32>
In return we will instantly credit $20
to your account[...]
```

Another extreme example is a mailin which the user always sees the text

> [...]We recently reviewed your account, and we suspect an unauthorized ATM based transaction. Therefore as a preventive measure[...]

but the text is "encoded"/stored differently in every mail. Here, useless, empty HTML tags "" are inserted in randomly chosen places:

```
[...]W<i></i>e rec<i></i>ently revi<i></i>ewed your ac<i></i>count, and
we susp <i></i>ect an unaut<i></i>horized A<i></i>TM ba<i></i>sed
transact<i></i>ion. Theref<i></i>ore as <br>[...]
```

Phishers can also use different methods or combine them: By using non existing HTML tags (but this is not a good choice for the phishers, because content scanners can focus on that) or inserting

null chars (\x00, see [7]), which are ignored by Microsoft Internet Explorer and also by Mozilla Firefox and Thunderbird in the href attribute (not seen in phishing mails so far).

Therefore, to improve phishing detection in e-mail filtering tools, it is very important to have some good normalization, which should carry out the following tasks:

- remove null chars
- lowercase all chars
- remove HTML tags
- remove empty HTML tags and HTML comments
- normalize HTML entities, e.g. replace all   with a whitespace
- reduce/remove useless whitespaces
- (remove punctuation characters, which are sometimes permuted, but there are no statistics on this so far)

Maybe it is advisable to have different types of normalization side by side in order to gain better results on pattern matching. As already implied, it should be possible to use the same detection method for normalised phishing messages as for SPAM: Bayesian filtering.

Hence, it is possible to create a signature on a phishing-specific text fragment after normalization. Take the initial "XX sent this message from"-example. According to the HTML source (all in one line)

```
[...]<td valign=bottom><font face="verdana, sans-serif" color=#666666 size=1>
<b>ebay sent this message from john doe (jdoe).</b></font></td></tr>
</tbody></table>[...]
```

one can take

```
6666 size=1><b>ebay sent this message from {-50}.</b></font></td>
```

as a signature with `{-50}` as a wildcard for 0 up to 50 bytes/chars. Using only "ebay sent this message from" as the signature is not advisable, because in any other mail there could be some other similar text which might lead to false positives. It is also possible to base a signature on conspicuities/mistakes the phisher has made (e.g. mail claims to be from bank A, but contains a link to bank B or also includes images from bank B server) or just on a specific layout. The following generic signature matches the phishing mail shown in figure 2.7 on page 10 (image only, white background and white random text, signature has to be on one line):

```
dy bgcolor="#fffff{-20}"><a href=http://{-20}bank.de.{-100}><img
src="cid:{-150}border={-3}></a></p><p><font color="#fffff
```

In the majority of phishing cases, phishers want the victim to enter a prepared website by clicking on a hyperlink. In order to fool as many people as possible, phishers try to hide the real URL the hyperlink points to (see figure 2.8 on page 10; javascript is sometimes also used to overwrite the statusbar) — I shall take a closer look at this in the next section. There are two more methods which are similar and which are worthy of mention here, even though they are not used all that often: The first one includes a complete HTML form (with text boxes and submit button) in the email: Hence, the victim can enter and submit his personal data directly from the mail client. The second method uses a HTML form, too, but contains only a submit-button (figure 2.9 on page 10), in order to avoid the real address being shown on the statusbar of the mail client.

Figure 2.7: Phishing example: Image followed by random text

Figure 2.8: Phishing example: different URLs

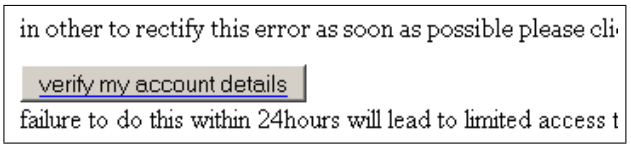


Figure 2.9: Phishing example: no hyperlink, but submit button

2.3 A closer look at the URLs used in phishing mails

The common method of phishing is an HTML mail which tries to hide the real address of the phisher's website (compare with the previous section, beginning on page 6).

There are two feasible solutions for automatic phishing checks, which I discuss in this section:

- Check whether the address to which the hyperlink points is blacklisted (e.g. Google-Safe-Browsing as used in Mozilla Firefox 2.0)
- Compare displayed text and the URL to which the hyperlink points (heuristic method, used in Mozilla Thunderbird, Eudora and Clam AntiVirus)

2.3.1 Blacklisting URLs

Before browsing an URL, check whether the URL is on a blacklist. This can be done online just-in-time by addressing a server which hosts the blacklists and asking for every URL which the user wants to browse to. This can be done by transmitting the whole URL or a hash. – If the URLs are sent plainly over the wire, it is possible for the server provider to record user profiles.

The other method is to periodically download a blacklist[2] from a server and use this for further "offline" checks – this is the way Mozilla Firefox 2.0 fraud detection works. In general, the problem with blacklists is that they need to be up-to-date and must contain all harmful URLs. It is not possible to completely automate the management of the blacklist: On the one hand it might happen that sensible data are published[3] and on the other hand the blacklist can get polluted: It is quite useless to add

```
http://session-id74747.bank.com.somedomain.tld/blablabla
http://session-id9343145.bank.com.someotherdomain.tld/ablalabbla
⋮
http://session-id904749.bank.com.somedomain.tld/lalalala
```

without wildcards since the phisher can use random ids and various (sub)domains. Hence, it is impossible to add all possibilities in order to guarantee proper protection. This issue makes the online hash solution less useful.

Blacklisting the IPs the domains resolve to might be an idea, but if a phishing site is located on a shared web server[4], the whole (maybe hijacked) server with all websites is blocked.

[2] Google-Safe-Browsing blacklist http://sb.google.com/safebrowsing/update?version=goog-black-url:1:1

[3] see [6]

[4] one IP, but hundreds of websites

2.3.2 Checking displayed and real URL

A straightforward way to automatically identify phishing mails would be to compare the displayed URL - this includes the URL for a picture - and the URL to wich the hyperlinks point (figure 2.8 on page 10). This idea has already been implemented and processes at ClamAV[5] and Mozilla Thunderbird[6]:

```
<a href="http://real-destination">https?://displayed-link</a>
and
<a href="http://real-destination"><img src="https?://server/imagefile"></a>
```

Of course the displayed text has to be normalized, because complete URLs are not always available there. A missing "http://", missing "//" of "http:domain.tld", quotes, extra dots, commas instead of dots or spaces are not rare (figures 2.10 and 2.11). Apart from just comparing both URLs (most likely only the domain names) it is advisable to perform further tests to improve the results: Check whether the protocol differs (link uses "http://", but "https://" is displayed, not advisable for images) or the link points to a (numeric) IP-address, which might be masqueraded (there are different encodings for IP-addresses available: "plain"/decimal http://192.168.156.74, "dword" http://3232275530, "octal" http://0300.0250.0234.0112, "hexadecimal" (two ways) http://0xC0.0xA8.0x9C.0x4A and http://0xC0A89C4A or mixed http://0300.0xA8.156.0112). These checks alone lead to an increased false positive rate, because the use of numeric IP-addresses is not forbidden. Especially for newsletters it is a common method to let the hyperlink point to their website but let the images come from a different (image) server.

From the point of view of an anti-virus product, you have the problem of having to say "this file is clean" or "this file is infected". There are no gray values, such as some points given by SPAM-filtering software or the chance to ask the user, as in an email application (see figure 2.15 on page 15). ClamAV has improved these checks by using a periodically updated offline database which contains the domain names for common targets and performs these checks only on included domains.

However, there are negative examples of senders (which are on the common targets list) of legitimate mails who do not care whether the two URLs differ: Use of different (sub)domains or top-level domains (figure 2.12 on page 13) or the use of the secure http-protocol on the displayed URL and normal http-protocol on the real-link side (shown in figure 2.13 on page 13). For this reason and because of the image-server problem, a white list is also needed; this should include the allowed displayed real-link combinations. This is no perfect solution, because online shops are still a big problem. Shops might have a newsletter in which they include their images from their own server; however, they transfer the real link to a reseller address of the enterprise shop which is on the common targets list. In this case, white listing is not feasible/possible.

Phishers monitor the phishing-detection methods as well as anti-phishing-product vendors monitor the (new) tricks of the phishers. As a consequence, phishers are aware of these automatic checks as well as registered new domains like "http://www.pay-pal-billing.com", "http://www.bank-login.com" or those with spelling errors "http://www.barciays.com/". Hence, both URLs are equivalent and not on the common-targets list. A reason why this works might be the marketing of several companies. I expect special services of a company to be located on a specific path under one domain name or to be accessible by using a dedicated subdomain. In this case, however, I suppose the marketing department said "we need a short address for every service". Domains like "bank-online.tld", "bank-ebanking.tld" were registered. There are also banks which

[5] see [9]

[6] see [11] and [12], figure 2.15 on page 15

Figure 2.10: Phishing example: commas in displayed URL instead of dots

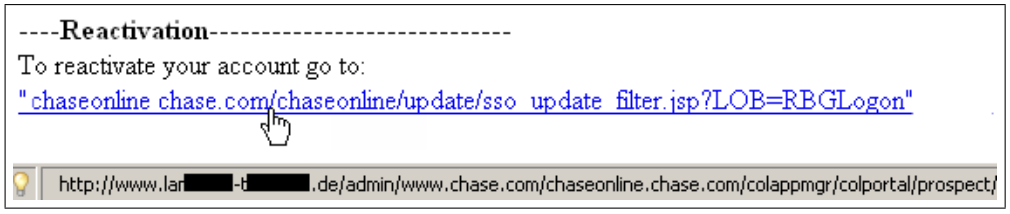


Figure 2.11: Phishing example: spaces and quotes in displayed URL

Figure 2.12: Legit mails: Mismatching URLs

Figure 2.13: Legit mail: https-missmatch

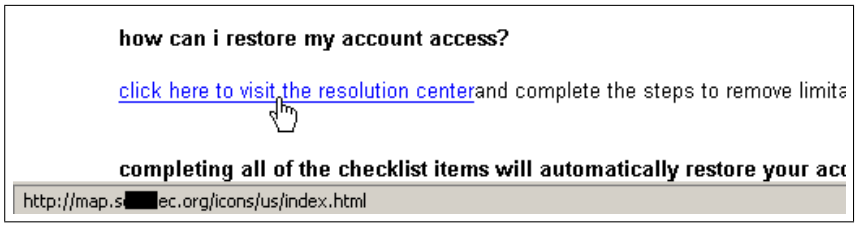


Figure 2.14: Phishing example: text with URL to phishing website

use the subdomain approach, but not many. Even if only more advanced users would notice this in mails, it would be a good start and companies could say "you can validate our web address by checking whether it ends on name.tld or name.tld/ must be on the URL in front of the first slash / (after http://) and no @-sign behind it". The same applies to very long path names and long session ids on the URLs for online banking, which made it possible to use almost endless URLs on phishing mails without making the user think.

Nowadays hyperlinks are often used without a displayed URL but displayed text `<a href="real-URL">Login here</a>` (see figure 2.14 on page 14), "Bank Login", "Update Form", "Activate NOW" or equivalent texts already known from SPAM mails like "Click here". The problem in this case is that the displayed/real URL-combination cannot be used for the described automatic check. However, phishers often include the name or domain name of the company in the real URL:

http://domain.tld/sth/www.bank.com/update.asp
http://domain.tld/sth/bankname/login.php
http://domain.tld/ebayisapidll.htm
http://www.bank.com@domain.tld/check.html

With this knowledge, new checks can be performed on the URL. The only unsolvable problem involves hyperlinks which do not contain any text referring to the target company for these automatic checks (figure 2.16 on page 15) without causing many false positives.

In conclusion, both methods, blacklisting of phishing websites and automatic URL checks are required for detecting phishing mails on the URL basis.

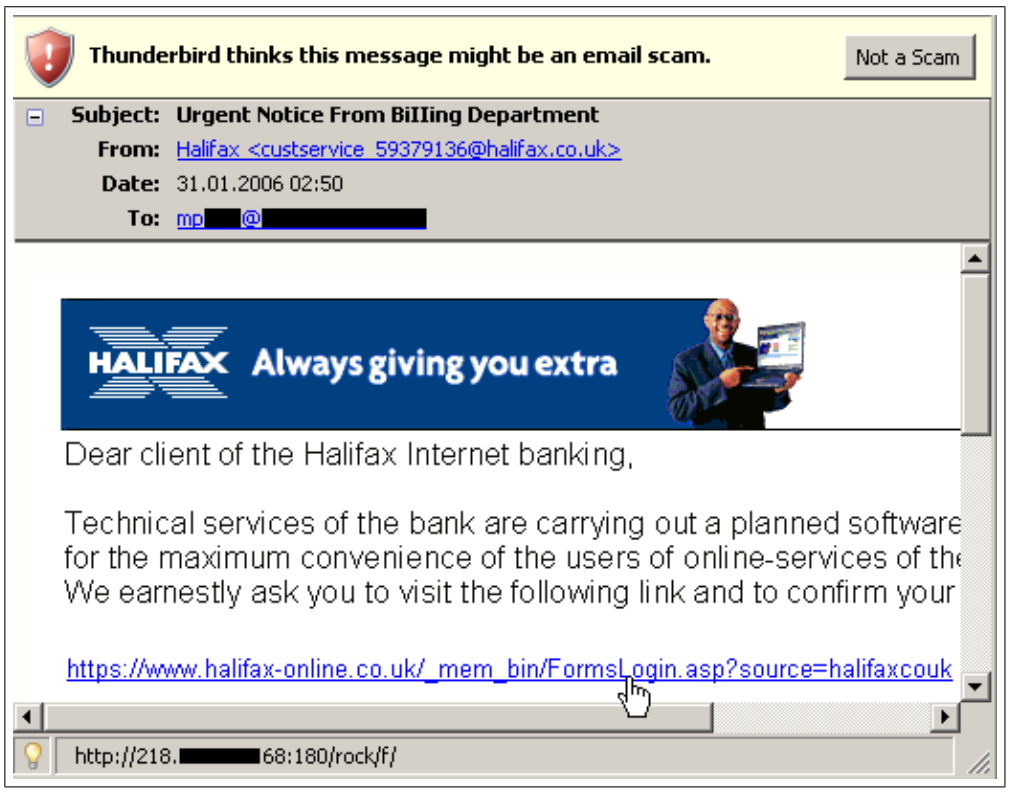


Figure 2.15: Phishing example: Thunderbird scam detection

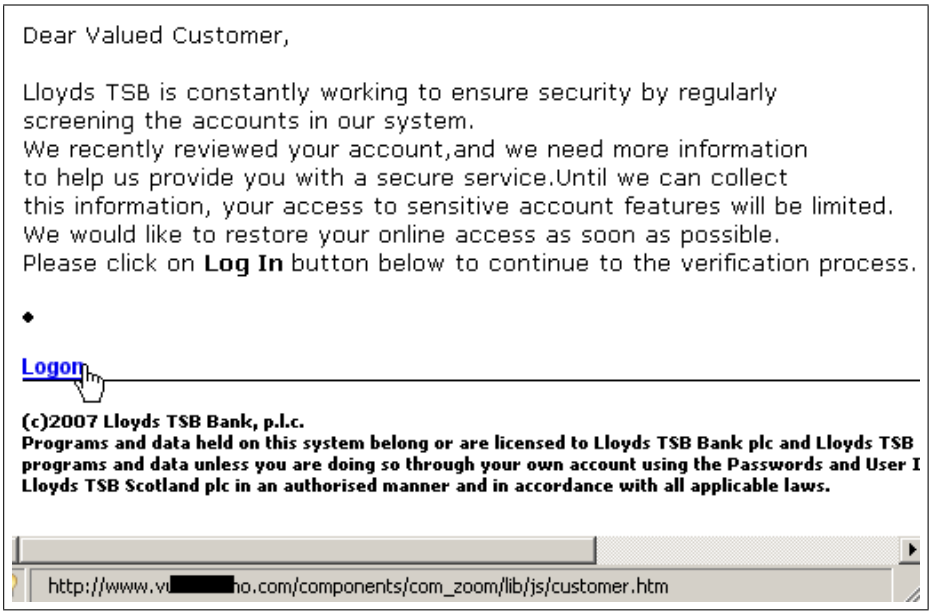


Figure 2.16: Phishing example: displayed text is no URL and real link does not contain any usable data

3 Comment

An often discussed solution should be "'signing all outgoing mail"', but it is not that easy: The simple mail transfer protocol (SMTP[1]) was and is not designed for digitally signed content. Therefore it is the task of the mail user agent (MUA, client software or webmail front end) to deal with and verify signed messages, but not all MUA do this properly. When mail travels through different mail servers with different mail transfer agents, it is not rare that this breaks the digital signature (like S/MIME[2]). Maybe new techniques like Domain Keys Identified Mail (DKIM[3]) perfom better: DKIM performs some normalization before checking the signature and helps to verify if a mail comes from the correct source domain. As one of the first companies eBay and PayPal use DomainKeys (on which DKIM is based) for their domains.

The general use of the secure http-protocol would even help to improve the detection of fraud attempts by not very advanced users: If someone opens a website which is secured by SSL the browser automatically checks the certificate and warns the user if the certificate is not signed by a trusted Certification Authority. But of course the URL has to be checked, because it is still possible with a server certificate for *.domain.tld to run a phishing site with the domain www.some-bank.com.domain.tld (only valid for certain browsers, there is no standard for the wildcard handling, if the wildcard is valid for only one or more levels) or some-bank.domain.tld. Besides this wildcard problem, it is not guaranteed that the domain name is checked before the certificate is issued. Hence, it is possible that someone applies for a certificate for paypal-accounts.com and use this domain for phishing.

A solution for the online banking problem (not identity theft problem) is the use of new techniques apart from the old personal identification number (PIN) and transaction authentication number (TAN) technique, like mTAN: When a customer wants to transfer money from his account to another account, the bank sends a text message to the customer's mobile phone containing the amount, destination account and a TAN which is valid for the next five minutes and this transfer only. In general all techniques which allow the customer to verify and authenticate the transaction with a separate secure computer/device (e.g. for signing this transaction) or via a different channel and allow only this specific transaction can help to protect from phishing, because no phisher can ask the victim to enter TANs for other/future transactions which are not authorized by the victim.

It does not seem as if most banks really care about security: A lot of websites of (even big) banks and other common targets contain or contained well and long known security problems,

- frames ([8] describes vulnerable banking sites and techniques),
- cross-site scripting vulnerabilities (XSS[4], [1] and [5] describe some recent vulnerable banking sites)
- offering unlimited, automatic URL redirecting services (e.g. on ad servers)

[1] see [15], list of RFCs and links included there

[2] RFC1847 and RFC2311

[3] described in [10]

[4] see [13]

and/or help phishers by hindering automatic methods of phishing detection

- inconsequent usage of domain names,
  - in hyperlinks (compare chapter 2.3.2 starting on page 12, Figures 2.12 and 2.13 on page 13),
  - different domains (often one for every service, e.g. www.bank-ebanking.com, www.bank-card.com instead of subdomains of bank.com),
- very long URLs (e.g. very long path names and/or long session ids),
- hyperlinks to login pages in emails.

List of Figures

Bibliography

[1] Ulrich Keil. XSS vulnerability on various german online banking sites (sparkasse), 2007. [Online: http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2007-05/msg00258.html; accessed 30-December-2007].

[2] Paul L. Kerstein. How Can We Stop Phishing and Pharming Scams?, 2005. [Online: http://www.csoonline.com/talkback/071905.html; accessed 17-November-2007].

[3] E. Levinson. RFC 2111: Content-ID and Message-ID Uniform Resource Locators, 1997. [Online: http://www.ietf.org/rfc/rfc2111.txt; accessed 18-November-2007].

[4] Dietmar Müller. Phisher richten immer mehr Schaden in Deutschland an, 2007. [Online: http://www.silicon.de/enid/antivirus/29256; accessed 17-November-2007].

[5] Paul Mutton. Italian Bank's XSS Opportunity Seized by Fraudsters, 2008. [Online: http://news.netcraft.com/archives/2008/01/08/italian_banks_xss_opportunity_seized_by_fraudsters.html; accessed 09-January-2008].

[6] Jeremy Reimer. Google's anti-phishing plugin leaked passwords, 2007. [Online: http://arstechnica.com/news.ars/post/20070122-8677.html; accessed 18-November-2007].

[7] Jürgen Schmid. Null Problemo, 2005. [Online: http://www.heise.de/security/artikel/63411/, http://www.heise-security.co.uk/services/browsercheck/demos/ie/null/default.shtml; accessed 17-November-2007].

[8] Jürgen Schmid and Edward Henning. You can't Bank on Security, 2006. [Online: http://www.heise-security.co.uk/articles/76590/; accessed 30-December-2007].

[9] Edwin Török. PhishingDesign, 2006. [Online: http://wiki.clamav.net/Main/PhishingDesign; accessed 18-November-2007].

[10] Various. RFC 4871: DomainKeys Identified Mail (DKIM) Signatures, 1997. [Online: http://www.ietf.org/rfc/rfc4871.txt; accessed 10-Januar-2008].

[11] Various. Bug 279191 – Add Phishing Detection Support to Thunderbird, 2007. [Online: https://bugzilla.mozilla.org/show_bug.cgi?id=279191; accessed 18-November-2007].

[12] Kelson Vibber. How Thunderbirds Scam Detection Works, 2005. [Online: http://www.hyperborea.org/journal/archives/2005/10/28/thunderbird-scam-detection/; accessed 17-November-2007].

[13] Wikipedia. Cross-site scripting — Wikipedia, The Free Encyclopedia, 2007. [Online: http://en.wikipedia.org/w/index.php?title=Cross-site_scripting&oldid=180821004; accessed 30-December-2007].

[14] Wikipedia. Phishing — Wikipedia, The Free Encyclopedia, 2007. [Online: http://en.wikipedia.org/w/index.php?title=Phishing&oldid=172031049; accessed 17-November-2007].

[15] Wikipedia. Simple Mail Transfer Protocol — Wikipedia, The Free Encyclopedia, 2008. [Online: http://en.wikipedia.org/w/index.php?title=SMTP&oldid=176943033; accessed 10-Januar-2008; list and links to related RFCs included].