

# Know your Enemy: Phishing

## *Behind the Scenes of Phishing Attacks*

The Honeynet Project & Research Alliance

<http://www.honeynet.org>

Last Modified: 16th May 2005

Phishing is the practice of sending out fake emails, or spam, written to appear as if they have been sent by banks or other reputable organisations, with the intent of luring the recipient into revealing sensitive information such as usernames, passwords, account IDs, ATM PINs or credit card details. Typically, phishing attacks will direct the recipient to a web page designed to mimic a target organisation's own visual identity and to harvest the user's personal information, often leaving the victim unaware of the attack. Obtaining this type of personal data is attractive to blackhats because it allows an attacker to impersonate their victims and make fraudulent financial transactions. Victims often suffer significant financial losses or have their entire identity stolen, usually for criminal purposes. This KYE white paper aims to provide practical information on the practice of phishing and draws on data collected by the [German Honeynet Project](#) and [UK Honeynet Project](#). This paper focuses on real world incidents that the Honeynet Project has observed in the wild, but does not cover all possible phishing methods or techniques. Attackers are constantly innovating and advancing, and there are likely to be new phishing techniques already under development or in use today.

After a brief introduction and background, we will review the actual techniques and tools used by phishers, providing three examples of empirical research where real-world phishing attacks were captured using honeynets. These incidents will be described in detail and include system intrusion, phishing web site preparation, message propagation and data collection. Common techniques and trends are then analysed, including the growing integration of phishing, spamming, and botnets. Examples of the malware used by phishers to automate harvesting of email addresses and sending of spam email are reviewed, and we also present our observations on network scanning techniques and how compromised machines are used to spread phishing emails and other spam. Finally, we conclude this paper with an overview of the lessons learned in the last six months and suggest further research topics.

This white paper includes extensive amounts of supporting information, with many hyperlinks to more detailed data on specific attacks available inline. Lastly, no confidential personal data was collected in the process of this research. In some cases, organizations involved in phishing attacks were contacted directly, or the incident data was forward to local CERTs.

## **Introduction**

Tricking others into giving out passwords or other sensitive information has a long tradition in the attacker community. Traditionally this activity has been performed through the process of social engineering. In the 1990s, with the increasing growth in interconnected systems and the popularity of the Internet, attackers started to automate this process and attack the mass consumer market. The first systematic research to cover such activity was published in 1998 by Gordon and Chess (Sarah Gordon, David M. Chess: [Where There's Smoke, There's Mirrors: The Truth about Trojan Horses on the Internet](#), presented at the Virus Bulletin Conference in Munich, Germany, October 1998). Gordon and Chess were researching malware on AOL, but they were faced with phishing attempts instead of the expected trojan horse attacks. The term phishing ("password harvesting fishing") describes the fraudulent acquisition, through deception, of sensitive personal information such as passwords and

credit card details by masquerading as someone trustworthy with a real need for such information. A phishing message described by Gordon and Chess is shown below:

```
Sector 4G9E of our data base has lost all I/O functions. When your account
logged onto our system, we were temporarily able to verify it as a
registered user. Approximately 94 seconds ago, your verification was made
void by loss of data in the Sector 4G9E. Now, due to AOL verification
protocol, it is mandatory for us to re-verify you. Please click 'Respond' and
re-state your password. Failure to comply will result in immediate account
deletion.
```

Early phishing attacks were primarily aimed at gaining access to the victim's AOL accounts, or occasionally at obtaining credit card data for fraudulent purposes (e.g. to make illegal purchases with this information). Often the phishing messages contained a simple ruse to trick unskilled computer users and relied heavily upon the victim's innate sense of trust in "automated" system functions or (apparent) figures of authority. As demonstrated in the previous example, this could be a story about a broken hardware device or the failure of a database, and most normal system users would take at face value any reasonably official-looking or highly urgent technical request that appeared to offer them assistance. Users were usually prompted to enter sensitive information quickly to avoid a serious problem, for example via the phrase "[...] and re-state your password. Failure to comply will result in immediate account deletion". To avoid potentially dire consequences the victims often complied immediately, unknowingly providing the social engineer with the credentials they required. Anecdotal evidence suggested that the culprits usually were acting alone or in small, unsophisticated groups. Literature often portrays early phishers as adolescents desiring account data for causing mischief and to make long distance phone calls, usually with little high level organisation or malice.

Today, the preferred strategy chosen by phishers is to bulk email their lures to as many end users as possible whilst masquerading as a trusted brand - usually one with whom the phisher hopes there is a chance that the victim already trusts. A request for urgent action is sent, often ironically to protect the user's confidential data from malicious activities, and this spoof email will contain an obscured link to a remote web-page that masquerades as the public web site of the target brand. The phisher hopes that victims will be tricked into submitting their credentials into a fake, but apparently legitimate looking "official" web interface for the trusted brand. Examples of the organisations being targeted by phishers include many well-known banks, credit card companies or well known Internet traders requiring regular payments (e.g. eBay and PayPal). Numerous examples of phishing emails targeting customers can be found at the [Anti-Phishing Working Group](#) web site, which has a [archive of phishing emails](#), many of which illustrate the high degree of accuracy with which phishers can trick innocent users into believing they are accessing a legitimate web interface.

Following this brief introduction to the concepts of phishing, we will now review the actual techniques and tools we have captured during phishing attacks observed in the wild. If you are interested in further background on phishing, we have prepared this page of [detailed background information](#).

## Tools and Tactics

Phishing attacks generally rely on a number of simple tools and techniques to trick unsuspecting users. The underlying infrastructure to support a phishing scam may be as basic as a simple copied HTML page uploaded to a freshly compromised web server and a server side script to process any user input data, or it may involve more complex web sites and content redirection, but generally the objectives are the same - to set up a fake web presence for a trusted brand with the necessary back end capabilities to process user input data and make it available to the attacker. Using modern HTML editing tools it is very easy to produce a web site mimicking a target organisation, and poorly secured web servers can easily be located and compromised if an attacker is not adverse to scanning entire portions of Internet IP address space in the search for vulnerable target hosts. Once compromised, even home PCs can make effective hosts for phishing web sites, so not only well known corporate or academic systems are targeted. Attackers are often indiscriminate in their choice of target computers, purely selecting large IP address blocks to scan at random for a particular exploitable security vulnerability.

Once a phisher has established a realistic and convincing fake web site that mimics a trusted brand, their main challenge is how to divert users of a legitimate web site to the fake web site instead. Unless the phisher has the ability to alter the DNS for a target web site ([DNS poisoning](#)) or somehow otherwise redirect network traffic (a technique sometimes referred to as [pharming](#)), they must instead rely on some form of content level trickery to lure unfortunate users to the fake web site. The better the quality of the lure, and the wider the net that can be thrown, the greater the chance of an innocent user mistakenly accessing the fake web site (and in the process potentially providing the phisher with the victim's credentials or other personal data).

Unfortunately for the attacker, when they target an individual organisation (such as a bank or trusted retailer), the phisher probably does not have any information about who on the Internet is a customer of the target organisation and therefore who might be most receptive to a particular lure. Although the attacker could post hyperlinks pointing to the fake web site on chat rooms and forums related to the target brand (such as a technical support web site or community discussion group), it is likely that the target organisation would be notified reasonably quickly and the offending hyperlinks removed or discredited before many victims had accessed the content and submitted their personal details. There would also be a significant risk that the target organisation or law enforcement agencies might trace and potentially shut down the fake web site. The phisher therefore requires a method of reaching the maximum number of potential victims with the minimum amount of risk, and they have found their ideal partner in crime in the form of spam email.

Spammers have databases containing many millions of active email addresses, so the latest mass emailing techniques can be employed to allow a phisher to distribute their lure to a very wide audience with very low risk. Spam emails are often sent via compromised servers hosted in foreign countries, or via global networks of zombie PCs ([botnets](#)), so the likelihood of an individual sender being traced is low. If an unsuspecting user receives an officially branded email that appears to have been sent by their bank which asks them to go to what appears to be the bank's usual branded web site to change their online banking password for security reasons, they are much more likely to consider doing so than when confronted with standard spam emails about novelty products and links to unknown web sites. To increase the likelihood that a user will believe that an email is genuine, the phisher can employ a number of techniques to further improve the quality of their attempted deception:

- Using IP addresses instead of domain names in hyperlinks that address the fake web site. Many innocent users will not check (or know how to check) that an IP address is registered and assigned to the target organisation that the branded fake web site claims to represent.
- Registering similar sounding DNS domains and setting up fake web sites that closely mimic the domain name of the target web site (i.e. b1gbank.com or bigbnk.com instead of bigbank.com), in the hope that users will mistake the fake domain name for the real domain name.
- Embedding hyperlinks from the real target web site into the HTML contents of an email about the fake phishing web site, so that the user's web browser makes most of the HTTP connections to the real web server and only a small number of connections to the fake web server. If the user's email client software supports auto-rendering of the content, their client may attempt to connect automatically to the fake web server as soon as the email is read, and manual browsers may not notice the small number of connections to a malicious server amongst the normal network activity to the real web site.
- Encoding or obfuscating the fake web site URL. Depending on the method employed, many users will not notice or understand what has been done to a hyperlink and may assume it is benign. One variant of this technique ([IDN spoofing](#)) is to use [Unicode](#) URLs that render in browsers in a way that looks like the original web site address but actually link to a fake web site with a different address.
- Attempting to exploit weaknesses in the user's web browser to mask the true nature of the message content. Microsoft's Internet Explorer and Outlook applications have been particularly vulnerable to such techniques (such as the [address bar spoofing](#) or [IFrame element](#) bugs).
- Configuring the fake phishing web site to record any input data that the user submits (such as usernames and passwords), silently log them and then forward the user to the real web site. This might cause a "password incorrect, please retry" error or even be totally transparent, but

in either situation many users will not be overly worried and put this event down to their own poor typing, rather than intervention by a malicious third party.

- Set up a fake web site to act as a proxy for the real web site of the target brand, covertly logging credentials that are not encrypted using SSL (or even registering valid SSL certificates for spoof domains).
- Redirect victims to a phishing web site by first using malware to install a malicious [Browser Helper Object](#) on their local PC. BHOs are DLLs designed to customize and control the Internet Explorer web browser, and if successful, victims can be tricked into believing they are accessing legitimate content when in fact they are accessing a fake web site.
- Use malware to manipulate the *hosts* file on a victim's PC that is used to maintain local mappings between DNS names and IP addresses. By inserting a fake DNS entry into a user's *hosts* file, it will appear that their web browser is connecting to a legitimate web site when in fact it is connecting to a completely different web server hosting the fake phishing web site.

Due to the relatively complex nature of many e-commerce or online banking applications, which often employ HTML frames and sub-frames or other complex page structures, it may be difficult for an end user to easily determine if a particular web page is legitimate or not. A combination of the techniques listed above may mask the true source of a rendered web page and an unsuspecting user might be tricked into mistakenly accessing the phisher's fake web site, unknowingly divulging their authentication credentials or other personal data. At this point the phisher will be free to make use of the user's accounts or electronic identity as required, and the user becomes another victim of a successful phishing attack.

## Real World Phishing Techniques

Very often Internet users become aware of phishing attacks by receiving spoof emails themselves or viewing a recorded copy of a malicious web site below the headlines on a technology news site, long after the server temporarily hosting the phishing content has been taken down. These events tend to be viewed in isolation and purely from the perspective of the victim. One of the major benefits that honeynet technology can offer is the capability to capture all activity from the perspective of the attacker, allowing security analysts to build up a more complete understanding of the entire life span of a phishing attack. Members of the Honeynet Project's [Research Alliance](#) are fortunate enough to have captured a number of rich data sets that can help to illustrate the stages of such an attack, from initial compromise and phishing web site set up through to mass emailing and victim data capture. Three different examples of typical real world phishing techniques are presented and reviewed below.

### Phishing Technique One - Phishing Through Compromised Web Servers

Most phishing attacks that we have observed in the wild involve attackers breaking in to vulnerable servers and installing malicious web content. Honeynet technology allows us to capture in detail the typical life cycle of a phishing attack, and in general terms the flow of events we have observed during such incidents are as follows:

- Attackers scan for vulnerable servers
- Server is compromised and a rootkit or password protected backdoor installed
- Phishers gain access to the server through this encrypted back door
- If the compromised server is a web server, pre-built phishing web sites are downloaded
- Some limited content configuration and web site testing is performed (potentially revealing the phisher's true IP address when they first access the web server)
- Mass emailing tools are downloaded and used to advertise the fake web site via spam email
- Web traffic begins to arrive at the phishing web site and potential victims access the malicious content

Often the time taken for this incident life cycle is only a matter of hours or days from when the system is first connected to the Internet, and our research suggests that such activity is taking place on many

servers and targeting many organisations at once. We will illustrate these theories using data recorded during two incidents that are typical of common phishing attacks, using one incident observed by the German Honeynet Project and one incident observed by the UK Honeynet Project. In each case, vulnerable Linux honeypots were deployed by Honeynet Research Alliance members. The subsequent compromise of both honeypots shared a similar modus operandi: the vulnerable honeypots were scanned and compromised in quick succession, with pre-built phishing web sites and mass emailing tools for sending spam emails being uploaded and used by the attackers. Rootkits and IRC servers were also installed during these attacks, something we commonly observed in other similar incidents. The compromised honeypots were also used for several different purposes in addition to phishing: as an IRC bot by Romanian attackers and also as a scanner to locate and attack additional vulnerable computers (although the honeynet architecture prevented the attackers from successfully exploiting other servers from the compromised honeypots). Some interesting differences were also apparent, not least in the case of the UK incident, where several different groups accessed the compromised honeypot at the same time, making forensic analysis more complicated. For the sake of brevity, we have not included the details of these specific attacks in this paper and have only covered the lessons learned and how they apply to phishing. If you would like to review more details about the specific attacks, the following information is available:

- [Overview of Honeynet configurations](#)
- [Details of German honeypot compromise](#)
- [Details of UK honeypot compromise \(timeline\)](#)
- [Details of UK honeypot compromise \(content analysis\)](#)

The table below shows a summary of the key factors and differences between the incidents:

Data	DE Incident	UK Incident
Compromised honeypot	Redhat Linux 7.1 x86.	Redhat Linux 7.3 x86.
Location	German corporate network.	UK ISP data centre.
Attack method	"Superwu" autorooter.	"Mole" mass scanner.
Vulnerability exploited	Wu-Ftpd File globbing heap corruption vulnerability ( <a href="#">CVE-2001-0550</a> ).	NETBIOS SMB trans2open buffer overflow ( <a href="#">CAN-2003-0201</a> ).
Level of access gained	Root.	Root.
Rootkit installed	<a href="#">Simple rootkit</a> that backdoors several binaries.	<a href="#">SHV4 rootkit</a> .
Probable attackers	Unknown.	Multiple groups from cable modem IP ranges in Constanta region of Romania.
Web site activity	Multiple pre-built phishing web sites downloaded targeting eBay and major US banks.	Pre-built phishing web site downloaded targeting a major US bank.
Server side processing	<a href="#">PHP script</a> to validate user input.	<a href="#">PHP script</a> with more advanced user input validation and data categorisation.
Email activity	Tried to send spam ( <a href="#">example 1</a> , <a href="#">example 2</a> ), but blocked by <a href="#">Honeywall</a> .	Only test emails sent, potentially to fellow phishers. Improved syntax and presentation.
Mass emailing method	<a href="#">Basic PHP script</a> from medium sized input list of email addresses.	Basic PHP script from small input list of email addresses - possibly just a test.
Victim traffic reached honeypot	No, spam advertisement and access to phishing web site blocked.	Yes. 265 HTTP requests in 4 days, not due to spam sent from this server (no customer details were compromised).

From observing the phisher's keystrokes in both incidents (captured using [Sebek](#)), it is clear that the attackers connected to pre-existing back doors and immediately went to work deploying their phishing

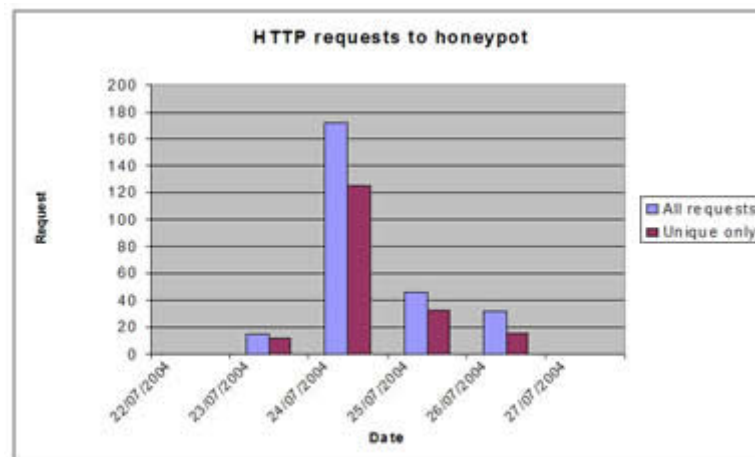
web sites. The attackers appeared to be familiar with the server environment, suggesting they were part of the group who originally compromised the honeypots, and that the phishing attempt was fairly well organised. As the uploaded web content often referred to other web servers and IP addresses, it is also likely that such activity was probably occurring on multiple servers at once.

Analysis of the phishing web site content downloaded by attackers during these incidents makes it clear that phishers are simultaneously targeting many well known online brands. Well constructed and officially branded pre-built fake web sites are routinely being deployed onto compromised servers - often targeting multiple organisations via separate "micro sites", with separate web server document roots, along with the necessary tools to propagate spam emails to potential phishing victims. Directory listings observed during FTP sessions also confirm that the attackers were heavily involved in spam and phishing activities, revealing pre-built web content and message delivery tools stored on a central server and appearing to target at least eBay, AOL, and several well known US banks in the case of the [UK incident](#). These individual phishing attacks are unlikely to be isolated events, as the spam emails sent during the incidents often directed victims to a different web server than the compromised honeypot. This indicates that phishers are running multiple fake web servers and sending spam from multiple systems at once. Parallel phishing operations are also indicated by the timing of the first inbound HTTP request for phishing content after the UK honeypot was compromised:

```
2004-07-23 21:23:14.118902 XXX.XXX.XXX.XXX -> 10.2.2.120 HTTP GET  
/.internetBankingLogon HTTP/1.1
```

This inbound HTTP request to the honeypot occurred before the attackers had finished setting up the fake online banking content on the honeypot, and confirms the hypothesis that the attacker knew in advance that this server was available for use as a phishing web site. Spam messages advertising the new phishing web site were already being emailed to victims from another host, even whilst the attacker was setting up the new phishing web site.

We were surprised by the number and range of source IP addresses making inbound HTTP requests to the compromised honeypot for the fake online banking content. The graph below shows the number of unique and repeat HTTP requests from individual IP addresses to the UK phishing web site before the honeypot was disconnected to protect end users (and the incident details logged with the targeted bank):



A breakdown of the source top level DNS domains, countries and host operating systems accessing the phishing content on the UK honeypot can be found [here](#). Note that before the honeypot was taken offline for forensic analysis, although web traffic for the phishing web site did arrive at the UK honeypot, no HTTP POST requests were made to the PHP script that processes users' data and therefore no user data was compromised during this phishing attack. In all the incidents discussed in this white paper, either the target organisation was notified of the incident and any relevant data was made available to them on request, or the local CERT was notified of any malicious activity. In all cases no compromised victim personal data was captured by HoneyNet Project or Research Alliance members.

Data from these two example incidents suggests that phishers are active and organised, moving quickly between compromised computer systems and simultaneously targeting multiple well known brands. It also appears that a number of email users are regularly being tricked into accessing fake web interfaces for organisations such as online banks or retailers, and risk becoming victim's of phishing attacks.

## Phishing Technique Two - Phishing Through Port Redirection

In November 2004, the German HoneyNet Project deployed a classic [GenII honeynet](#) with a Redhat Linux 7.3 honeypot. Although this is a relatively old operating system release and an easy target for attackers, it surprisingly took around 2.5 months before the honeypot was successfully compromised - a marked contrast with the relatively quick compromise of the honeypots discussed in the incidents above. More information on this trend can be found in a previous KYE white paper "[Know your Enemy: Trends](#)".

On January 11th 2005, an attacker did successfully compromise the honeypot, using an exploit for the [OpenSSL SSLv2 Malformed Client Key Remote Buffer Overflow Vulnerability](#) present in the default Redhat Linux 7.3 distribution. This incident was unusual in that once the attacker had gained access to the compromised system, no phishing content was uploaded directly. Instead, the attacker installed and configured a port redirection service on the honeypot.

This port redirection service was designed to re-route HTTP requests sent to the honeypot web server to another remote web server in a transparent manner, potentially making the location of the content source harder to trace. The attacker downloaded and installed a tool called [redir](#) on the honeypot, which was a port redirector utility designed to transparently forward incoming TCP connections to a remote destination host. In this incident the attacker configured the tool to redirect all incoming traffic on TCP port 80 (HTTP) of the honeypot to TCP Port 80 (HTTP) on a remote web server in China. Interestingly, the attacker did not bother to install a rootkit to hide their presence on the honeypot, which suggests that the attacker did not value the compromised server too highly and that they were not particularly worried about being detected.

The command used by the attacker to establish port redirection was:

```
redir --lport=80 --laddr=<IP address of honeypot> --cport=80 --caddr=221.4.XXX.XXX
```

In addition, the attacker modified the Linux system start up file [/etc/rc.d/rc.local](#) to ensure that the redir port redirector service would be restarted if the honeypot system was rebooted, improving the chance of survival for their port redirection service. They then began to send out spam phishing emails which advertised the honeypot, an example of which can be found [here](#) (note that relevant sensitive information has been obfuscated).

To further investigate the activities for the phisher, members of the German HoneyNet Project intervened and covertly modified the configuration of the attacker's redir tool installed on the honeypot, enabling logging within the redir application itself, to more easily observe how many people received a spam email advertising the honeypot and then clicked on a hyperlink to access the transparently redirected phishing content. Within a period of about 36 hours, 721 unique IP addresses were redirected, and once again we were surprised by how many users were apparently being tricked into accessing such content through phishing emails. An analysis of the IP addresses accessing the port redirector honeypot can be found [here](#) (note that this information has been sanitized to protect the users who accessed the phishing content, and again only IP data was logged during this research. No confidential user data was captured).

A summary timeline of the incident is provided below:

Date / Time	Event
1st Nov 2004	First network probe data of honeypot
11th Jan 2005 - 19:13	Honeypot <a href="#">OpenSSL</a> service compromised, port redirector installed and <a href="#">phishing spam</a> sent

11th Jan 2005 - 20:07	<a href="#">Web requests</a> for phishing content begins to arrive at honeypot
13th Jan 2005 - 8:15	Honeypot taken offline for forensic analysis

## Phishing Technique Three - Phishing Using Botnets

The recent white paper by the HoneyNet Project called "[KYE: Tracking Botnets](#)" introduced a method to track botnets. A botnet is a network of compromised computers that can be remotely controlled by an attacker. Due to their immense size (tens of thousands of systems can be linked together), botnets can pose a severe threat to the community when used for Denial-of-Service (DoS) attacks. Initial research in this area demonstrated that botnets are sometimes used to send out spam emails and can also be used for phishing attacks. During a study in October 2004, email security company [CipherTrust](#) suggested that 70% of monitored phishing spam was sent through one of five active botnets, but our own observations suggest that many more botnets are in use for spam operations. Although not the analysis of one single incident, in this section we present our observations on the tools and techniques used by attackers engaged in phishing via botnets.

### Incident Timeline

During the period between September 2004 and January 2005, the German HoneyNet Project deployed a series of un-patched [Microsoft Windows](#) based honeypots to observe botnet activity. An automated process was developed to allow honeypots to be repeatedly deployed, compromised and shutdown for forensic analysis. During this period over 100 separate botnets were observed and thousands of files were captured for offline analysis.

### Analysis

Some versions of bot software captured during this research project provided the capability to remotely start a SOCKS v4/v5 proxy on a compromised host. SOCKS provides a generic proxy mechanism for TCP/IP-based networking applications ([RFC 1928](#)) and can be used to proxy most popular Internet traffic, such as HTTP or SMTP email. If an attacker with access to a botnet enables the SOCKS proxy functionality on a remote bot, this machine can then be used to send bulk spam email. If the botnet contains many thousands of compromised hosts, an attacker is then able to send massive amounts of bulk email very easily, often from a wide range of IP addresses owned by unsuspecting home PC users.

The lack of a central point of contact and the range of international boundaries crossed could make it very difficult to trace and stop such activity, making it of low risk, but potentially high reward to spammers and phishers. Perhaps unsurprisingly, resourceful botnet owners have begun to target criminal activity and it is now possible to [rent a botnet](#). For a fee, the botnet operator will provide a customer with a list of SOCKS v4 capable server IP addresses and ports. There are documented cases where botnets were sold to spammers as spam-relays: "[Uncovered: Trojans as Spam Robots](#)". Some captured bot software also implemented a special function to harvest email-addresses or to send spam via bots. The following listing shows some of the commands related to sending spam/phishing emails implemented in Agobot, a popular bot used by attackers and a variant regularly captured during our research:

- **harvest.emails** - "makes the bot get a list of emails"
- **harvest.emailshttp** - "makes the bot get a list of emails via HTTP"
- **spam.setlist** - "downloads an email list"
- **spam.settemplate** - "downloads an email template"
- **spam.start** - "starts spamming"
- **spam.stop** - "stops spamming"
- **aolspam.setlist** - "AOL - downloads an email list"
- **aolspam.settemplate** - "AOL - downloads an email template"
- **aolspam.setuser** - "AOL - sets a username"
- **aolspam.setpass** - "AOL - sets a password"

- **aolspam.start** - "AOL - starts spamming"
- **aolspam.stop** - "AOL - stops spamming"

Further information about how these commands are implemented can be [found here](#) in a side note about the source code of bots. With the help of *drone*, a customised IRC client developed by the German HoneyNet Project, we were able to learn more about how bots are used for spam/phishing email attacks by smuggling a fake client into a botnet using the connection data collected through the attacks against our honeynets. A number of typical examples of observed activity are shown below.

## Example 1

Within one particular botnet we observed an attacker who issued the following command (please note that the URLs have been obfuscated):

```
<St0n3y> .mm http://www.example.com/email/fetch.php?4a005aec5d7dbe3b01c75aab2b1c9991
http://www.foobar.net/pay.html Joe did_u_send_me_this
```

The command `.mm` ("mass emailing") is a customized version of the generic `spam.start` command. This command accepts four parameters:

1. A URL for a file that contains several email addresses.
2. The web page to target within the spam email - this could be a normal spam web-page or a phishing web site.
3. The name of the sender.
4. The subject of the email.

In this case, the *fetch.php* script returned 30 different email addresses every time it was invoked. To each of these recipients, an email message was constructed that advertised the second parameter of the command. In this example, it pointed to a web-page which attempted to install an [ActiveX](#) component on the victim's computer.

## Example 2

In another botnet we observed the installation of Browser Helper Objects on a victim's PC:

```
[TOPIC] #spam9 :.open http://amateur.example.com/l33tag3/beta.html -s
```

The `.open` command tells each bot to open the requested web-page and display it to the victim. In this case the web-page contained a [Browser Helper Object \(BHO\)](#) that would attempt to install itself on the victim's computer. As the channel name indicates, this botnet was also used for sending spam.

## Example 3

In another botnet we observed examples of spyware propagation:

```
http://public.example.com/prompt.php?h=6d799fbeeef3a9b386587f5f7b37f[...]
```

This link was found during analysis of captured malware. It directs the victim to the web-page of a company that offers "*free ad delivery software which provides targeted advertising offers*". This web site contains several pages that try to install ActiveX components on visiting clients, presumably aware or spyware.

# Common Themes

A number of common themes were observed during our research into phishing attacks, and it is clear that attackers are employing a blend of tools and techniques to improve their chances of success. We will now briefly review two such techniques - mass scanning and combination attacks.

## Mass Scanning

Analysis of a number of compromised honeypots suggests that the systems were being attacked using automated attack scripts or exploits, often known as autorooters. In both the incidents described in phishing technique one above, once the attackers had compromised the honeypots, autorooter toolkits were uploaded to the server. The attackers then attempted to scan large ranges of IP addresses for similarly vulnerable servers (using scanners called "*superwu*" in the German incident and "*mole*" in the UK incident). Captured attacker keystrokes from the UK incident are show below, showing examples of the types of mass scanning activity attempted from compromised honeypots. Note that due to the honeynet configuration, hostile outbound traffic was blocked and these attacks did not succeed.

Attacker extracts scanner and attempts to scan class B network blocks:

```
[2004-07-18 15:23:31 bash 0]tar zxvf mole.tgz
[2004-07-18 15:23:33 bash 0]cd mole
[2004-07-18 15:23:38 bash 0]./mazz 63.2
[2004-07-18 15:24:04 bash 0]./mazz 207.55
[2004-07-18 15:25:13 bash 0]./scan 80.82
```

Attacker attempts to exploit potentially vulnerable servers:

```
[2004-07-19 11:56:46 bash 0]cd mole
[2004-07-19 11:56:50 bash 0]./root -b 0 -v ns1.victim.net
[2004-07-19 11:57:26 bash 0]./root -b 0 -v 66.90.NNN.NNN
```

Attacker returns later to check list of successfully compromised servers (the list was empty, due to the honeynet configuration):

```
[2004-07-23 08:13:18 bash 0]cd mole
[2004-07-23 08:13:20 bash 0]ls
[2004-07-23 08:13:25 bash 0]cat hacked.servers
```

Attacker attempts to scan multiple class B network blocks and then test an exploit against a selection of targets:

```
[2004-07-24 10:24:17 bash 0]cd mole
[2004-07-24 10:24:19 bash 0]./scan 140.130
[2004-07-24 10:24:27 bash 0]./scan 166.80
[2004-07-24 10:25:36 bash 0]./scan 166.4
[2004-07-24 10:26:23 bash 0]./scan 139.93
[2004-07-24 10:27:18 bash 0]./scan 133.200
[2004-07-24 10:36:37 bash 0]./try 202.98.XXX.XXX
[2004-07-24 10:38:17 bash 0]./try 202.98.YYY.YYY
[2004-07-24 10:38:27 bash 0]./try 202.98.YYY.YYY
```

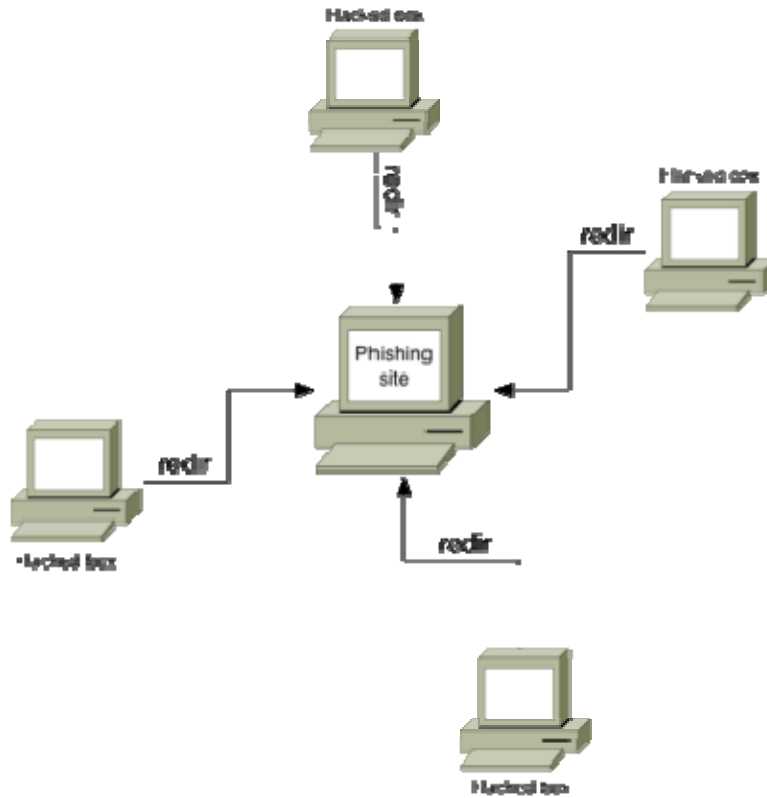
In the final example above, note that the hosts that the attacker attempts to compromise are not part of the IP address ranges scanned from this honeypot, which again provides evidence of well coordinated and parallel mass scanning activity.

Further investigation of the *mole.tgz* file downloaded by UK attackers revealed a number of text files in the root directory of the unpacked autorooter toolkit. These files included scan configurations and logs of previous scanning activity for the "*grabb2.x and samba2.2.8 vulnerability*". 42 cases of attacks against other hosts were present in these files, along with evidence of mass scanning of many class B network blocks, confirming that the observed incident was part of larger and more organised attack against similar systems. An example of the output from the mole scanning tool, viewed from an attacker's perspective, can be found [here](#).

Finally, some of the mass scanning tools recovered from compromised honeypots do not appear to be in popular circulation, which suggests that the attackers had some level of development and tool smith capabilities beyond basic script kiddy activity, or were part of a closed community that did not share their tools in public forums. Again, this suggests more organised attackers.

## Combination Attacks

In our research, we also observed that phishers are frequently combining the three attacking techniques we have observed and documented in this white paper, sometimes combining multiple methods to provide redundancy and protect their phishing infrastructure through implementation of a two-stage networking configuration. The following diagram depicts a possible phishing network topology:



In this example a central web server hosts the physical phishing content, often serving more than one web site (e.g. an eBay phishing-site in /ebay and a PayPal phishing-site in /paypal). Several compromised remote computers redirect incoming HTTP traffic on TCP port 80 to the central web server with the help of the `redir` port redirector. This has several advantages from an attacker's point of view when compared to a single phishing web site:

- If the compromise of one of the remote `redir` hosts is detected, the victim will probably take the system offline and re-install it. This does not represent a major loss for the phisher because the main phishing web site is still online and several other `redir` hosts continue to deliver HTTP traffic to the central web server.
- If the compromise of the central phishing server is detected, this system will also be taken offline. Now the phisher can simply set up a new phishing site on a freshly compromised system and then re-adjust the existing network of `redir` hosts to redirect traffic to the replacement central host. Using this technique, the whole network can be made available very quickly and the phishing attacks can soon recommence.
- A `redir` host is very flexible, since it can be easily reconfigured to point to another phishing web site. This decreases the time between initial system compromise and phishing web site availability, and increases the length of the attack window in which the phishing attacks can be performed.

The use of such techniques again suggests more organised and capable attackers, rather than the work of simple script kiddies. Similar operational models are often used by major web hosting companies and high volume content providers.

## Further Observations - Fund Transfer

Our research has also shed light on how phishers use captured information about bank accounts, for example, an account number with associated TAN (transaction number used in electronic banking). Since foreign currency transfers are monitored by most banks, phishers cannot simply transfer large amounts of money from one country to another without alerting the financial authorities. Phishers therefore have to use intermediaries to transfer money for them - in a two stage process the phisher transfers money from the victim's bank account to a bank account of an intermediary in the same country. The intermediary then withdraws the money from their bank account (less a percentage remuneration for providing the service) and sends it to the phisher, for example by surface mail. Of course, the intermediary could be caught, but as the phisher's money is already in transit they do not face too much risk and can easily change to channel their funds through a replacement intermediary. An example email demonstrating some of the financial structures behind phishing attacks is show below:

Hello!  
We finding Europe persons, who can Send/Receive bank wires from our sellings, from our European clients. To not pay TAXES from international transfers in Russia. We offer 10% percent from amount u receive and pay all fees, for sending funds back.Amount from 1000 euro per day. All this activity are legal in Europe.  
Fill this form: <http://XXX.info/index.php> (before filling install yahoo! messenger please or msn), you will recieve full details very quickly.

---

Wir, europäische Personen findend, die Bankleitungen davon Senden/erhalten können unsere Verkäufe, von unseren Kunden von Deutschland. STEUERN von internationalen Übertragungen in Russland nicht zu bezahlen. Wir erhält das Prozent des Angebots 10 % vom Betrag und bezahlt alle Schulgelder, um Kapital zurück zu senden. Betrag von 1000 Euro pro Tag. Diese ganze Tätigkeit ist in Europa gesetzlich.  
Füllen Sie diese Form: <http://XXX.info/index.php> (bevor die Füllung Yahoo installiert! Bote bitte oder msn), Sie recieve volle Details sehr.

Thank you, FINANCIE LTD.

This is a poor translation from English to German, probably computer-generated, and it suggests that the attackers are not native English speakers. Since the money will be transferred to Russia, the attacker probably originated from this country. This behaviour is becoming increasingly common as phishing activities become more organised.

## Honeysnap - An Incident Analysis Assistant

One conclusion was immediately obvious when we started to analyse data from the UK honeynet compromise in phishing technique one above - due to multiple simultaneous attacks by different blackhat groups, a significant amount of time would be required to extract and prepare the data from the network streams before more detailed analysis could take place. This data extraction process is repetitive and tedious, and if carried out manually represents an inefficient use of valuable analysis time. An automated solution was required.

The *honeysnap* script, written by [David Watson](#) of the UK Honeynet Project, grew out of this idea and was designed to process honeynet data feeds on a daily basis and produce a simple summary output to direct later manual analysis. The *honeysnap* script breaks down the data for each honeypot and provides lists of outbound HTTP and FTP GETs, IRC messages and Sebek keystroke logs. TCP stream re-assembly for interesting connections is automated, as is extraction, identification and storage of files downloaded by FTP or HTTP, meaning that much of the time-consuming preparatory work of incident analysis is removed, leaving the analysts free to concentrate on manually investigating key elements of

an incident. *Honeysnap* also provides an automated method for screening IRC traffic for interesting keywords (e.g. bank, account, password) and providing daily summary reports by email.

Currently *honeysnap* is a basic proof of concept UNIX shell script and the alpha release can be found [here](#), whilst a set of sample *honeysnap* output can be found [here](#). A modular and fully expandable version written in Python is currently under development by members of the HoneyNet Project and will be beta released to the community in June 2005.

## Further Research

The information presented in this white paper suggests a number of potential avenues for future research in the area of phishing attacks and we would recommend further investigation of the following subjects:

We would like to investigate if honeypots can be used to help in the fight against spammers and phishers. One possible research project would be to deploy additional honeypots of a type regularly used in previously observed phishing attacks or tuned to present attractive targets to spammers (such as SMTP open relays). Analysis of further attacks against these systems would help us to learn more about the anatomy of phishing attacks, particularly in the area of phishing using botnets, and to track the evolution of phishing techniques. Another research possibility would be to further develop the idea of honeypots and produce "client-side honeypots". This type of next-generation honeypot would actively participate in communication networks, for example, by automatically follow links in spam emails and accessing the target content. Client-side honeypots could idle in IRC-channels or share/download files via [peer-to-peer](#) networks, further improving our knowledge about the type of threats present in these communication networks.

In addition, we would like to investigate potential methods of countering or stopping phishing attacks. Since the time window between the start and end of a phishing attack is likely to be limited to a matter of only hours or days, and the source hosts are widely distributed, this is a difficult task. Current research efforts in this area (for example [The AntiPhishing Group](#) and [PhishReport](#)) concentrate on collecting phishing emails received by end users. Whilst this is a viable approach, capture occurs at the final stage in the incident lifecycle. An automated approach to capturing and responding to phishing attacks would be more desirable.

We suspect that accounts and passwords are being traded between blackhat groups, probably via IRC. HoneyNet technology could be used to capture such communication and further understand phishing activities. In addition, phishing tools often appear to be downloaded from a number of regularly updated central web or FTP servers. Although more contentious, monitoring of such activity or contacting the system owners would help to prevent some phishing activity, and a framework for operating such research and potential countermeasures could be established.

Further work is required to improve the automation of incident analysis, particularly in automatic profiling of data captured during such attacks. Automation of traffic and IP address extraction, reverse DNS and IP block ownership lookups, per IP address or per domain traffic summaries and en-masse passive operating system fingerprinting would all be particularly useful when analysing large data sets, as would a local forensic database of known hosts, attackers, attack signatures, message contents, etc. In the longer term, agreed standards for sharing such information and a global forensic database to support analysis of distributed blackhat activities would be highly desirable and of significant benefit to the community.

## Conclusions

In this paper we have presented a number of real world examples of phishing attacks and the typical activities performed by attackers during the full lifecycle of such incidents. All the information provided was captured using high interaction research honeypots, once again proving that honeyNet technology can be a powerful tool in the areas of information assurance and forensic analysis. We analysed multiple attacks against honeypots deployed by the German and UK HoneyNet Projects. In

each incident phishers attacked and compromised the honeypot systems, but after the initial compromise their actions differed and a number of techniques for staging phishing attacks were observed:

1. Setting up phishing web sites targeting well known online brands.
2. Sending spam emails advertising phishing web sites.
3. Installing redirection services to deliver web traffic to existing phishing web sites.
4. Propagation of spam and phishing messages via botnets

This data has helped us to understand how phishers typically behave and some of the methods they employ to lure and trick their victims. We have learned that phishing attacks can occur very rapidly, with only limited elapsed time between the initial system intrusion and a phishing web site going online with supporting spam messages to advertise the web site, and that this speed can make such attacks hard to track and prevent. IP address blocks hosting home or small business DSL addresses appear to be particularly popular for phishing attacks, presumably because the systems are often less well managed and not always up to date with current security patches, and also because the attackers are less likely to be traced than when targeting major corporate systems. Simultaneously attacking many smaller organisations also makes incident response harder. We have observed that end users regularly access phishing content, presumably through receiving spam messages, and a surprisingly large number appear to be at risk from becoming victims of such attacks.

Our research also suggests that phishing attacks are becoming more widespread and well organised. We have observed pre-built archives of phishing web sites targeting major online brands being stored, ready for deployment at short notice, suggesting the work of organised phishing groups. Such content can be further propagated very quickly through established networks of port redirectors or botnets. When coupled with evidence of mass scanning and hard coded IP addresses in web content and scripts, this suggests that many instances of a particular phishing site may be active at any one time. Web traffic has been observed arriving at a newly compromised server before the uploaded phishing content was completed, and phishing spam sent from one compromised host does not always appear to advertise the sending host, which again suggest it is likely that distributed and parallel phishing operations are being performed by organised groups.

Our research demonstrates a clear connection between spamming, botnets and phishing attacks, as well as the use of intermediaries to conceal financial transfers. These observations, when combined with quantitative data on mass vulnerability scanning and combined two-stage phishing networks, demonstrate that the threat posed by phishers is real, their activities are organised, and the methods they employ can sometimes be quite advanced. As the stakes become higher and the potential rewards become greater, it is likely that further advancements in phishing techniques and an increase in the number of phishing attacks will continue in the coming year. Reducing the number of vulnerable PCs contributing to botnets, countering the increasing volume of spam email, preventing organised criminal activity and educating Internet users about the potential risks from social engineering all remain significant security challenges.

A summary of all the linked sub-sections of this paper that provide supporting detail can be found below:

- [More detailed background information about phishing](#)
- [Details of UK compromise \(timeline\)](#)
- [Details of UK compromise \(content analysis\)](#)
- [UK attacker mole session](#)
- [Analysis of UK phishing victim source IP addresses](#)
  
- [Overview of German and UK honeynet configurations](#)
- [Details of German honeypot compromise](#)
- [German attacker sessions](#)
- [German PHP script analysis](#)
- [Analysis of German phishing victim source IP addresses](#)

- [Learning about phishing from bot source code](#)
- [Honeysnap sample output](#)

Questions and comments should be directed to the [German HoneyNet Project](#) and the [UK HoneyNet Project](#).