

EnterpriseITplanet.com's

AntiOnlinez

Maximum Security for a Connected World

Newsletter #11

INSIGHT

Table of Contents

Editorial
by MsMittensp. 3

Software releases and updates (quick look)
by scriptkiddie18p. 4

The Art of the Traceless RAM Attack
by Galenp. 5

Part 3 of Qod's Snort Tutorial: A look into
IDS/Snort
by Qodp. 8

Gone Phishin'
by MsMittensp. 22

MsMittens' Editorial

Well, this issue is smaller than usual but that's ok. We seem to go through busy periods and slower periods. AO itself seems to be in a bit of a lull but I suspect that will pick up as the "deep freeze" leaves most of North America and people get more into "playing".

During this lull period I've been mucking around some studying for the CISSP (gah! I think this is going to kill me) as well as checking out other "security" websites/forums. It's interesting in someways but also telling insofar as what AO actually has – and I think in a lot of ways, people miss that.

Many of these sites claim to be for "hackers". Quite the tossed around term, in my opinion. Most are filled with "borrowed" tutorials from other locations and have little to no content. Some might say that of AO but there's something else that is far more important: the people.

That is often the most critical thing that is missing from many sites today. The vast diversity and acceptance of diversity. With this diversity many at AO learn new things and discover new ways of dealing with old issues. The big key, in my opinion, is keeping all the doors open and being flexible enough for change. That's definitely one thing you learn at AO, how to deal with change.

Our ability to adapt to new items and things that happen is what can make us stronger (along with our willingness to learn new tricks and trades). This issue talks about various items that deal with change and/or adaptation. Scriptkiddie18 took a brief look at new software that came out at the beginning of the year. Galen takes us through the traceless RAM attack while Qod finishes off his tutorial on Snort and IDS. The closing article by me highlights what I think will be our biggest issue for 2004 and that will probably require some adaptation by administrators on how to deal with it: phishin'.

I hope you enjoy it and learn from this issue. I also hope that you submit something for the next issue. I'm always looking for articles. Don't think you don't know enough. There is always something that can be learned if you present it to the world. Our deadline for Insight #12 will be **March 19, 2004**. Email me (msmittens@msmittens.com) or pop me a PM.

Enjoy!

Software releases and updates

By: Scriptkiddie18

Software releases for January 2004

AIM (AOL Instant Messenger) version 5.5.3501 (beta)

AIM is one of the most used Instant Messengers all over and this beta version has changed this Instant Messenger in many ways. The biggest change would be that it is finally capable of video transmitting. It also has a new feature called linked screen names which I assume will allow one AIM session to sign in as more users. These linked screen names are on the new beta version but they do not work yet and will be fixed when the full version comes out. The new AIM beta version also supports new games. When I load the games, it asks to download the WildTanager Web Driver to play the games. I download the driver but afterwards the games are still not working. This might be just a bug in this beta version and hopefully will be fixed on the full version.

[Download AIM 5.5.3501 beta](#)

Macromedia Director MX 2004

The new updated director MX allows both PC and Mac users to create high-quality DVDs. Macromedia first announced it on the eve of Macworld Expo in San Francisco. Users can work with long video-streams, photo-quality images, audio, animation, 3D models, text, and Flash content in many formats. This update is suited to create educational but also entertainment materials. The company claims that the upgrade also offers a great amount of speed improvement on previous versions. This version of Director MX exploits with the DVD format which means that the producer can embed, control, and play DVD-Video format inside Director. Flash MX 2004 and Fireworks MX 2004 can be launched and edited within Director MX 2004 as long as the others are installed. This product also offers a wide support of media types like QuickTime, Windows Media, RealVideo, AVI and more.

The Art of the Traceless RAM Hack

by Galen

This is not meant to show some grand new way of attacking computers, or some new system that will allow you to defend yourself completely. It's a twist on a lot of old techniques that have been brought slightly up to date, and put together. I'm not talking about any specific attacks that go especially well with this, but rather just a different way of going about old attack and defense methods.

For every computer that anyone tries to hack, there are always two sets of logs, at least. The first is on the computer that was attacked. But where is the second one?

It's on the computer that the hacker used. The various utilities, scripts, logs, and all of that stuff stays there. And for the most part, it's never cleared out. This isn't really important unless they manage to trace the hack back to you and take the computer. But what if they do?

Unfortunately, if they do, you're probably screwed. They now have all of the evidence that they need to convict you of whatever it is you did, and probably a lot of stuff that you didn't do.

That's where it gets fun, isn't it?

It raises an interesting issue, though. What if those files weren't on your computer? What if there were no logs, no scripts, nothing at all like that.

What if the operating system on your machine didn't even match the one that had made the attack, and you'd had XP Installed since the day it came out.

The Attack:

Knoppix(<http://www.Knoppix.net>) would allow you to do just that.

For those of you who don't know, Knoppix is a newer distrobution of Linux that allows you to boot the OS from a CD, which is just plain wonderful. You can load Linux onto any computer, without a problem. (They even have the slogan "From zero to GNU/Linux in Five Minutes.") What's better, it can read FAT and NTFS partitions, so while you're using Knoppix, you can still access any files you have saved on Windows or Mac.

Knoppix is also a read only OS, so there's no risk of leaving anything on your computer when you're done. You just put it in, boot it up, and you're ready to go. Simply marvelous.

You can load whatever tools you think you'll need onto that same CD so that you can read them that way, and have all of your scripts nice and ready to go.

Another good place to load your scripts or utilities onto would be one of those mini-USB Drives, which work wonders. Since the newer ones can easily hold 128 MB, you can store all that you'll need and a bit more on them without a problem.

So the person has loaded up Knoppix and all of their tools onto their computer, and they're ready to go, right? There's one final step, one more thing that should be done, first. Bounce the IP Address, so it can't be traced back to your computer, just in case.

Use SSH or just about anything else to direct everything you do from another computer, and another IP Address. This makes it that much harder to

bring anything back towards your own computer. If the attacker has a lap-top, ideally they would go somewhere, like an air-port or a library, where its easy to get a good connection, and no one will be suspicious about a lap-top hook-up.

After everything is said and done, the CD gets dissolved in some bleach, or a number of other things. No trace is left of the hack on that end. If the hacker was good, no logs at all would survive.

The Defense:

The best defense against this is to be prepared ahead of time. The best, and perhaps only, way to be truly ready for this type of attack.

The best method that I have ever heard of, and this was purely rumor at the time, although I see no reason why it can't be, is to use a honeypot.

In this case, the honeypot would serve two real purposes: To keep the attacker busy (standard) and to gain as much information about them as is possible.

If you implement this, be sure to set it up to do every possible scan. Nmap, Host, everything. Look for any traces of files, UIDs, OSes, DNS strings, anything you possibly can. This is the only thing that MIGHT allow you take legal action against this type of attack in the future.

Aside from that, general security practices are your best hope.

A look into IDS/Snort

By: Q.o.D <QoDwriting@gawab.com>

Introduction:

Over the years IDS has gained popularity amongst organizations, with the rise of security risks, we needed a methods of detecting and possibly stopping intrusions. Last year alone (2002) we were hit with many viruses which could have been avoidable. In this paper we will discuss some of the concepts behind IDS, an infant technology that till recently has saw demand in businesses.

Disclaimer and Copyright:

Copyright © 2004 , 2005 Q.o.D <QoDwriting@gawab.com>

This document is free software; you can redistribute and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

PART III

Introduction:

In part II you saw some of the basics of Snort, what it offers, and why do people prefer it. In this part we are going to talk about some of the challenges of IDS especially NIDS. Techniques described here are general, and most ID systems should have found ways to stop them.

3.1 Architectural problems with NIDS

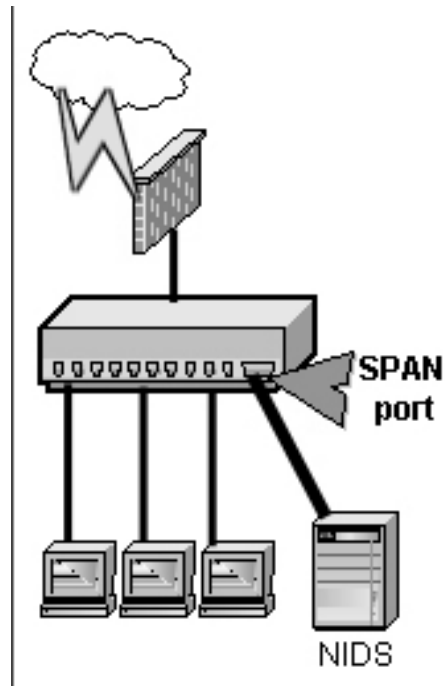
Many problems face IDS, and were mostly discussed in the disadvantages of NIDS in part I. We will be focusing some more on these ways, and discuss ways to fix the problems.

3.1.1 Switches

Switches work by making each port act as its own network segment, so if you send something to PC1 it is not going to send to PC2. This is much different and effective than the hubs, that is why switches are sometimes called intelligent hubs. This is a problem for NIDS that rely on sniffing the network, the reason is that if you put the sniffer on a port on the switch it will only see packets sent to it, and nothing else. There are two main ways to go around this problem:

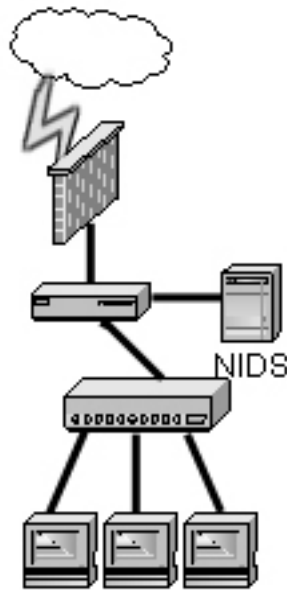
- 1) Using spanning ports

Some of the high end switches currently have SPAN ports on them. Span ports are port that are a mirror of the traffic passed through the switch. As you see in the following exhibit:



2) Installing Taps or Hubs

Similar to the concept of telephone taping implemented by the FBI, taps take all the traffic from the wire before it even goes to the switch. Hubs could be used as taps, as in the following network:



Both solutions have disadvantages. For example using span ports degrades your switch's performance dramatically, and getting the switches could be costly. Hubs on the other hand are half duplex so implementing a hub as you tap will slice your network performance by 50%.

3.1.2 Fast Ethernet

Gigabyte and 10 Gigabyte Ethernet are not a dream now, and many people are starting to use them. Because of the NIDS architecture, and the way it has to pickup each packet and analyze it, Gigabyte or 10 Gigabyte networks could be nearly impossible to Handel by an out of the box IDS. The only way for the NIDS to function at such high speeds is by tweaking it to your network's need, and follow the steps outlined in the reduce false positives section. One thing to note is that a properly configured Snort IDS will be able to Handel 1 Gigabyte Ethernet almost no problems.

3.1.3 Encryption

Since NIDS function at such a low layer (network) they have no means of decrypting or understanding encrypted packets. Therefore encrypted attacks are never detected by the IDS. The best way to combat this is by installing a HIDS on important machines that are commonly attacked such as web servers, or by installing the encryption keys on the NIDS itself so for every encrypted packet it would decrypt it and then check for a signature match.

3.2 Evading the IDS

Evasion is a technique that eludes the IDS from detecting a real attack, and thus rendering a false negative, thus defeating the purpose of implementing an IDS in the first place. Evasion techniques work by making the packet appear not to match an attack signature (while in fact it is the attack), and thus fooling your IDS. Although there are some evasions that work against HIDS we will be mostly focusing on NIDS since they are more popular. Some of the techniques that are described here have been presented in a paper called "A look at whisker's anti-IDS tactics" written by Rain Forrest Puppy(RFP) which is available at his website at www.wiretrip.net/rfp. The techniques are also implemented by a CGI vulnerability scanner program by RFP called whisker (discussed later). Although we are not going to talk about how to use the tools, I would heavily recommend that you test them against your IDS, if your IDS is deceived then it might be time to update or switch to another IDS since these tools have been around a while. We are also not going to discuss each and every commands, but rather some of the common and effective ones.

Throughout the examples we will be showing you multiple ways that an attacker could evade the following rule (signature):

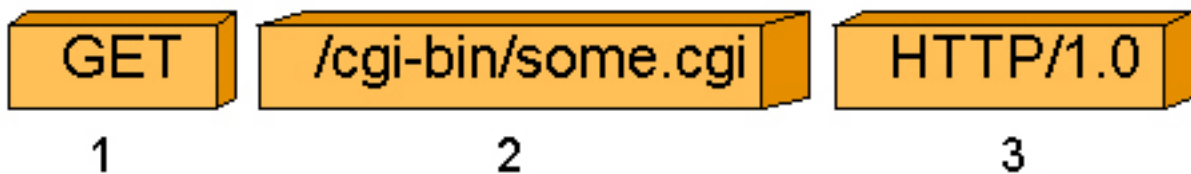
```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"WEB-ATTACKS /etc/shadow access"; flow:to_server,established;
content:"/etc/shadow"; nocase; sid:1372; classtype:web-application-activ-
ity; rev:4;)
```

Which basically says to alert when a request for the /etc/shadow file comes from an external network to the web server on http ports. The reason we chose this rule is because not only it is simple but also is a common file that an attacker will go after.

3.3 Evasion through URL

We have all heard about the traversal attack against IIS, these are common attacks against unpached web servers. These evasion are problematic to NIDS in particular because NIDS operate at the network layer and does not make much sense of the application layer or how the target machine will interpret the attack. If you do not understand how HTTP requests work, here is an overview from "A look at whisker's anti-IDS tactics" by RFP (www.wiretrip.net/rfp):

It is important to understand the components of an HTTP request. As defined by RFC 1945:



1. The method of request; typically GET, HEAD, POST, etc. Indicates what data format the server should expect and provide.

2. The URI, which is otherwise the page you are requesting. It can be relative ('/some/file') or absolute ('http://server/some/file'). Sometimes referred to as URL, but URL typically denotes an absolute URI (includes scheme—ftp:, http:—as well as server name).

3. The version, which is always in the format of "HTTP/x.x". Common versions are 0.9, 1.0, and 1.1. v0.9 is otherwise known as "simple requests".

4. All three components make up a 'request'. Each component is separated by a space.

Most IDS vendors currently are not fooled by these kinds of evasions, Snort for example has a preprocessor called HTTP_decode to deal with these kinds of evasions.

3.3.1 Self referenced directory

This technique is quite simple, in its simplest form it would look something like this:

```
GET ../../etc/../../shadow HTTP/1.0
```

Because the signature will not match the attack `GET /etc/shadow HTTP/1.0` the attack will pass undetected by the IDS.

3.3.2 Reverse transversal

By using the same idea as the transversal attack. Attacker found out that by going into a directory and getting back out they would be able to evade the IDS. The attack evasion would look something like this:

```
GET /etc/kevin/../../john/../../shadow HTTP/1.0
```

The directory could be anything, even none existing directories will work. Keep in mind that this could go on just going into a directory and out forever.

3.3.3 Long URL

Most ID systems have some sort of a maximum limit that a packet could be kept in memory, when that quota goes beyond the limit then the packet is discarded. Long URL functions by making the attack go beyond that maximum limit. For the sake of brevity i will not put an entire evasion but rather a stopped down version on what it would look like:

```
GET /etc/wade/../../museum/../../walkingetc.../shadow HTTP/1.0
```

This attack would be as long as the attacker wants it to be, but usually they will send the attack long enough to evade most ID systems.

3.3.4 Double slashes

Since it is introduction HTTP has undergone many changes, and by far has become on the most popular protocols in use. Because of this popularity, many diversities have occurred. Web server take input that were never defined by any standard. This popes a great challenge for IDS as they will not know if a packet is an attack or just a user not knowing what he is doing. One of the many inputs that web servers accept is double slashes in place of a single slash. So although `//` would look different from `/` , they will be both interpreted the same way by the web server.

```
GET //etc//shadow HTTP/1.0
```

Triple slashes could also be used

```
GET ///etc///shadow HTTP/1.0
```

3.3.5 Extra spaces

Putting an extra space between parameters sometimes would fool the IDS

```
GET /etc/wade/ ../themusium/ ../walkingonthestreetwhenimetyou/ ../shadow  
HTTP/1.0
```

Note: There are two spaces between GET and /etc/shadow

3.3.6 Tabs

Just like IDS are sometimes fooled by extra spaces. They are also sometimes evaded by using tabs instead of spaces.

```
GET /etc//shadow HTTP/1.0
```

Note: There is a tab between GET and /etc/shadow

3.3.7 \ (Backward slashes)

*nix bases OS use forward slashes(/) when representing directories, the Internet also uses /, but Windows does not use forward slashes, instead they use backward slashes(\). Web servers such as Microsoft IIS Server will accept backward slashes instead for forward ones.

```
GET \etc\shadow HTTP/1.0
```

3.3.8 Case sensitivity

Web servers interpret both the words WiLL, WILL, and will the same, while ID systems sometimes do not. You might be able to evade the IDS using this method.

```
GET /ETc/shADOW HTTP/1.0
```

3.3.9 URL encoding

Many people from around the world use the Internet, and most of them do not speak English. The web offerers such language to coexist on the Internet by using a universal language such as Unicode. Attackers found that by changing some of the letters or numbers of the attack to there equivalence of ASCII or Unicode they could successfully evade the IDS.

```
GET %2f%65%74%63%2f%76%68%61%64%6f%77 HTTP/1.0
```

This attack is the same attack as `GET /etc/shadow HTTP/1.0` in ASCII format. And since there are multiple ways to encode each letter (usually 4 or more) an attack would be able to find at least 1000 ways to present the attack to the target system, this is not counting Unicode.

3.3.10 Other evasion techniques

There are many other evasion techniques that are not discussed in this paper, here are just some that you might find interesting for research:

- 1) Padding
- 2) Parameter hiding
- 3) Premature URL ending
- 4) Fake parameters
- 5) Null method
- 6) Session splicing

You could also use any of the previously discussed evasions with each other.

Whisker is one of the very few tools that use evasion techniques in there scans, lets take a moments and ask what is whisker. From the whisker README:

"What is whisker?"

The primary purpose of whisker is to be a URL scanner, which is used to search for known vulnerable CGIs on websites. Whisker does this by both scanning the the CGIs directly as well as crawling the website in order to determine what CGIs are already currently in use."

Whisker is officially listed as deprecated, and RFP suggests that you check out tools like Nikto (www.cirt.net/code/nikto.shtml). Other tools that actually make use of whisker is Nessus (www.nessus.org) which has an option (through the libwhisker) to try to evade the IDS in its scan by using some of the ways described before.

3.5 Denial of Service (DoS)

Denial of service attacks serve mainly focus on two weaknesses in IDS.

1) The human factor

Some tools designed specifically to DoS IDS do that by generating tremendous amounts of false positives that the administrator cannot handle. Think about getting 20K of alerts each day, you will not be able to go through each one of them and analyze them carefully. They also work by burring the real attack between those false positives and thus masquerading the real attack. The disadvantage in this method is that the attack is somewhere in the logs, but it might take some time to find it.

2) The IDS it self

DoSing the IDS it self on the other tries to take the IDS off line by flooding it with too much false positives. This would cause the IDS to run out of memory, or run out of disk space all that would ultimately lead to an IDS crash. Some also try to keep the IDS busy with too much false packets that the IDS will start dropping any new ones and thus making the attack pass by undetected.

There are two tools that currently do just that.

3.5.1 Stick

From the <http://www.eurocompton.net/stick/projects8.html>:

"Stick - This is an IDS stress tool used to evaluate the bottle neck point in an IDS in an operational environment. Stick will not be released anytime soon for the exception of IDS vendors.

During testing it was discovered that ISS Real Secure v5.5 would turn itself off via error. I have sent the code to ISS (Chris Rollard) via a friend to insure that it went to a knowledgeable group in ISS. I have on my own accord contacted other vendors, some of which are not affected by this technique.

Over the last couple months I've been finishing up work on stick. I was planning to release a paper in the coming week and the tool in a month or two from now when IDS vendors have had time to make modifications to handle it.

The tool uses the Snort rule set and produces a C program via lex that when compiled will produce an IP packet capable of triggering that rule from a spoofed IP range (or all possible IP addresses) into a target IP range. A function is produced for each rule and a loop then executes these rules in a random order. The tool currently produces these at about 250 alarms per second.

A Linux based snort will hit 100% CPU and start dropping packets. The stress on recording and disk IO is another problem.

ISS Real Secure dies two seconds after the attack begins. This was tested numerous times. Other IDS and even sniffers (especially with DNS lookups) had problems of

their own."

3.5.2 Snot

from the Snot README:

"Snot is an arbitrary packet generator, that uses snort rules files as its source of packet information. It attempts at all times to randomize information that is not contained in the rule, to hamper the generation of 'snot detection' snort rules.

It can be used as an IDS evasion tool, by using specific decoy hosts, or just something to keep your friendly IDS monitoring staff busy.

*It has been tested to run on *BSD, Linux, Win2k, NT4.0 and Win98.*

Snort 1.8 has stream4 tcp state code. This will defeat TCP attacks with single packets. Use UDP or ICMP rules instead for the moment :)"

So is Snort vulnerable to these type of attacks, yes and no. From the Snort FAQ

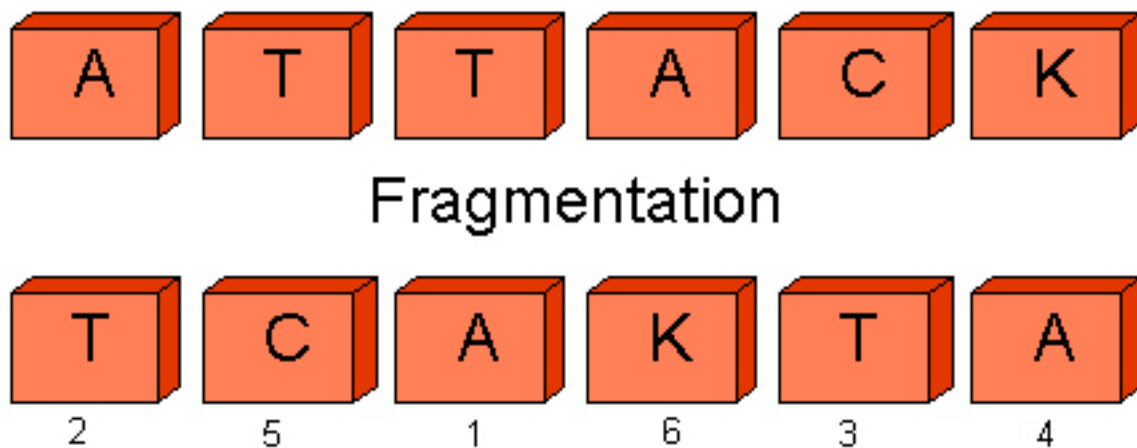
"1.9 Is snort vulnerable to IDS noise generators like "Stick" and "Snot"?"

It is now possible to defeat these kinds of noise generators with the (see FAQ 3.17) preprocessor. Even without the stream4 preprocessor enabled, snort will weather the alert storm without falling over or losing a lot of alerts due to its highly optimized nature. Using tools that generate huge amounts of alerts will warn a good analyst that someone is trying to sneak by their defenses."

3.6 Fragmentation

Although we are not going to talk about this type of evasion in this paper we will just introduce the idea. Fragmentation is when a big packet gets chopped up (fragmented) into smaller ones. Each Fragmented packet contains a sequence number to be assembled in the correct order. Fragmentation occurs naturally on every network, but sometimes they are used to hide the attack. Take this for example.

When this attack is sent the IDS will see **RCAKTA** which does not match the real attack **ATTACK**. There are some tools that automate that for you, the best and well known fragroute is one of them. Also note that in Snort the frag2



preprocessor handles fragments.

Looking at the above examples and the many ways that IDS is challenged you might have a second thought about implementing IDS on your network. If your IDS is implemented correctly you should not worry or run in any of these problems.

Gone Phishin'

by MsMittens

The Internet has pretty much been a wild and crazy place for many of us. We know of the inherent dangers because we started when the internet was smaller and when you didn't have fancy browsers. You did just fine – thank you very much – with a seperate email program (usually pine) and a seperate newsreader program (nn or tin). Heck, you probably also had gopher and archie running around looking for applications from a dozen universities or information storage sites. It was simpler back then.

But someone decided that money needed to be made on "them thar hills" of the internet so we got a browser instead of WAIS, gopher or nn. And it needed pretty pictures because who wants to look at words? And email is boring. I want to add more things to it. This sentiment of making things pretty, less boring, etc., has brought us to our present and most alarming issue. The "phishing".

The term, "leetisized" as it were, refers to the practise of "fishing" for information from users, usually in the form of creditials (user/pass), credit card information, personal information, etc. Whatever information an attacker can use. It dates back to the mid-1990s (around 1996 specifically) when attackers would "phish" for passwords from users of larger mainframes and university systems.

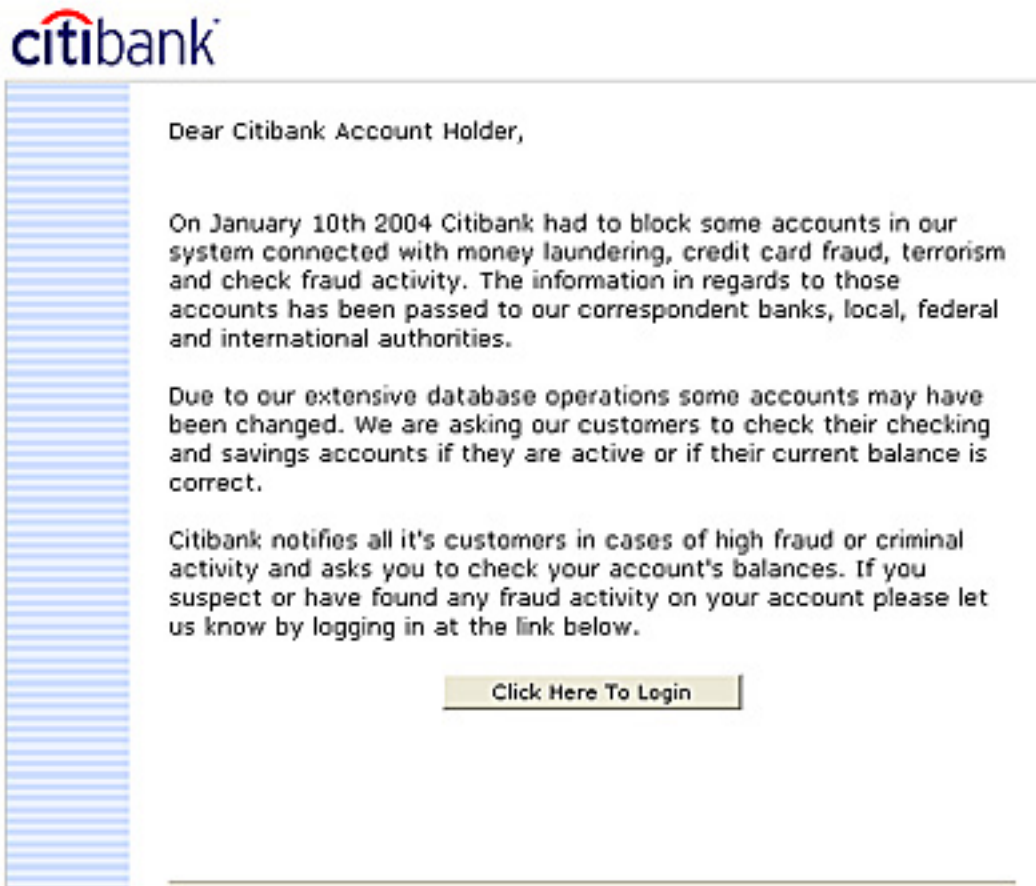
The target today is usually a financial institution or provider. By far, PayPal is the #1 target to date but other institutions are starting to be targetted. A recent "phish" (January 23, 2004) uses the FDIC, the Homeland Security Act and FUD (Fear, Uncertainty, Doubt) as a method to get the attention of the "victim".

The email sent out includes the following comment: "As a result Department Of Homeland Security Director Tom Ridge has advised the Federal Deposit Insur-

ance Corporation to suspend all deposit insurance on your account until such time as we can verify your identity and your account information. Please verify through our IDVerify below.”

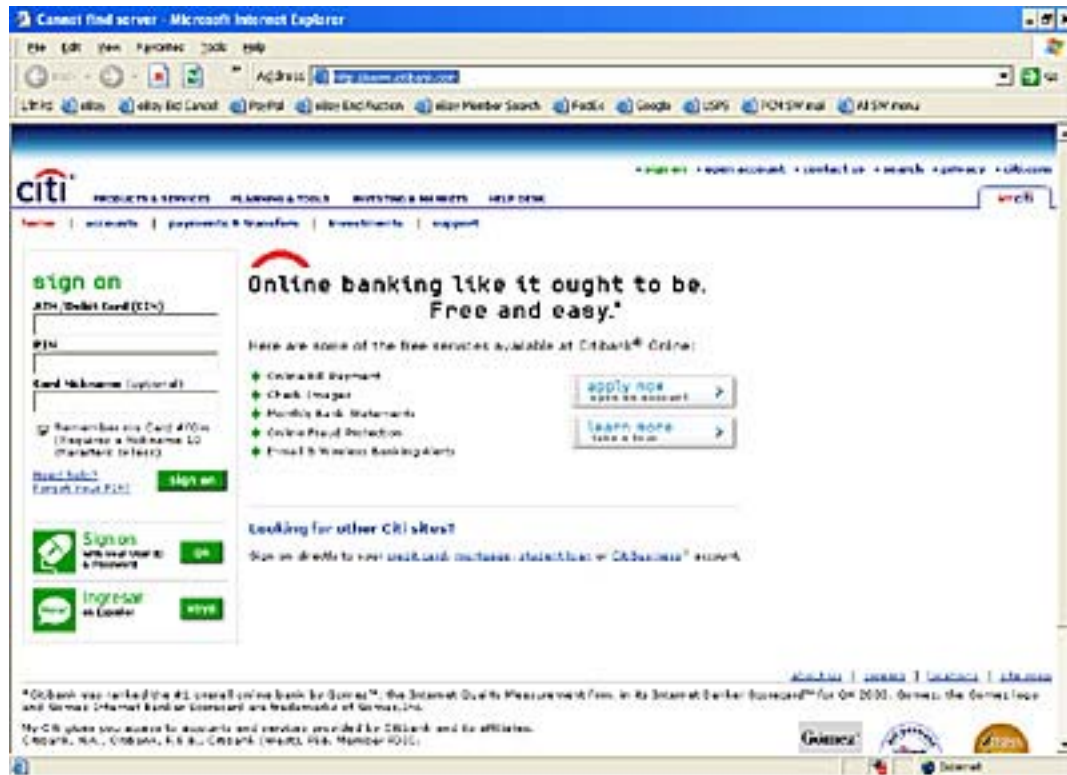
Gasp! Well I guess I better click the link and... Well, you get the point. The use of FUD helps the attacker a lot. Another phish that generated a fair amount of attention (heck, the media actually took notice) was against Citibank. You can see from the screen shot below what a user would have received.

From: Citibank [mailto:citibankxxxxx@yyyyyy.com]
Sent: Monday, January 12, 2004 9:53 AM
To: User@home.somewhere.com
Subject: Important Fraud Alert from Citibank



Most users look at something like this and figure it's got to be real. It looks

real. And it came from Citibank, right? They'll click the link and end up at a site like this:



Attackers are clueing in that spelling and grammar do make a difference. Earlier phishes were ripe with spelling errors and grammatical mistakes like this recent Citibank phish:



Just user gullibility right? Sort of. Users fall into a false sense of security. Look at many of the replies here at Antionline on how to solve things: "Do you have a firewall?", "Do you have anti-virus software up and running?", etc. Thing is, these won't stop phishing. What would stop phishing is raising the awareness and education of users online. We need to stop encouraging the Microsoft generation of users and push them towards exploring their computers again. It's not a bad thing to understand even the basics of the internet and it's not really that hard to use specialized programs for specific tasks.

So, some steps in stopping phishing at your company or home:

- 1 Phishing today mostly targets the Windows user. Main reason? Well, the reality is that many of the programs are integrated into the OS itself and because Microsoft has determined that users need scripts and html to make their lives easier. Internet Explorer and Outlook are two of the main culprits. The desire of Microsoft to hide information from users so they aren't bothered by large email headers or extra "stuff" in URLs is actually opening up users to being "phished".
- 2 Users wanting HTML emails is another problem. As an administrator, you should enforce rules that strip or prevent the reception and distribution of HTML email. Certainly it looks nice and pretty but people seem to think that if it was received via email then it must be true and valid.
- 3 Use browsers that cannot obfuscated URLs as much as Internet Explorer does (reality check on IE). The use of Unicode in Browser address bars shouldn't be allowed (why on earth...?). While not all obfuscated URLs will be taken care of using browsers that eliminate some of them will help.
- 4 Educate your users on the risks and what they should expect. If they receive an email claiming to be from Ebay or somewhere else, they should be suspicious. They should check the URLs for any unusual addressing (e.g., <http://www.ebay.com@1.2.3.4/login/login.html> or if they see a lot of Unicode). If they are unsure, have them forward it to the company security team. If it's a home user, send it back to the company in question and ask them to verify. Call the company if need be to check if they have, in fact, sent out such an email.
- 5 Emails don't ask for things like credit cards, user names and passwords, etc. Users should be leery of any email that asks for personal information or is suspect.

- 6 Users should be educated on using a single special credit card or chequing account so as to mitigate any damages if they still get caught in a phishing scheme.

- 7 Daily checks should be done to see what new phish attacks have come out. The AntiPhishing Group (see link in References) is a good place to start since they are now a central location to learn about such attacks. Keeping up-to-date in genearl on security through subscriptions to security lists is another. Full Disclosure is one that I recommend since it's unmoderated (this is also a downside).

Certainly not the be-all-end-all but some ways to deal with the problem. Phishing will probably be this year's big annoyance and problem along with some nice virus and worms (last year was one of the worst apparently since Melissa and I Love You in 2001). Our only real defense is awareness.

References/Help:

<http://www.antiphishing.org/> : Anti-Phishing.org is a website dedicated to keeping people informed about the latest scams as well as provide solutions to companies.

<http://www.rain.org/~mkummel/stumpers/08dec00a.html> : Treebeard's Stumpers's Answer December 2000: Obfuscating URLs. This gives a fairly indepth, albeit a bit old, look at obfuscated urls. Many of these techniques still work today.