GIAC Advanced Incident Handling And Hacker Exploits (GCIH)
Level Two Practical Assignment for Capitol SANS
(December 10th - 15th, 2000)

Microsoft Outlook / Outlook Express GMT Field Buffer Overflow Vulnerability

Jeffrey P. Hanson

Table Of Contents

1. Exploit Details

Name: Microsoft Outlook / Outlook Express GMT Field Buffer Overflow Vulnerability
Bugtraq ID: 1481
CVE: CVE-2000-0567
Variants: MIME Attachment
Operating System: Windows 95/98/NT/2000 running Microsoft Outlook 97/98/2000 or Outlook Express 4.0/4.01/5.0/5.01
Protocols/Services: POP3 and IMAP4. This is not a problem with POP3 and IMAP4. It is a problem with the code library used to implement them (inetcomm.dll).
Description: This vulnerability allows a remote user to run arbitrary code or crash the Outlook email client on an email recipient's system by attaching extra data or code to the date/time field of an email. The recipient of this email does not have to open or preview the email; the data or code is processed when the client attempts to retrieve the email from the mail server.

2. Introduction

The Microsoft Outlook / Outlook Express GMT Field Buffer Overflow Vulnerability was discovered by Metatron at GFI Security and was surfaced by Underground Security Systems Research (USSR) on July 18, 2000. USSR attempted to be very responsible in how they handled their findings, as they informed Microsoft about the vulnerability and waited until a patch was available before they publicly announced it. USSR's plan may have been a double-edged sword. While Microsoft was given plenty of lead time to develop the patch, and the patch has been available for some time now, there are still many systems at risk because of the relatively low profile of this vulnerability. Most users simply do not know that there is a problem.

3. Protocol Description

This vulnerability is present when Microsoft Outlook or Outlook Express is configured to retrieve email using the POP3 or IMAP4 protocols.

POP3 – Post Office Protocol (Version 3)

The Post Office Protocol is used for the retrieval of email messages from a mail server. A POP session consists of three states: Authorization, Transaction, and Update.

The way it functions is that a server is set up with a listener on TCP port 110. The client establishes a TCP connection with the server. The POP session then enters its Authorization state. At this time, the client provides its identity to the server using the USER and PASS commands.
Once authorization has been completed, the session enters its Transaction state. In this state, the client receives resources on the server, and commands can be issued. At this time messages can be retrieved using the RETR command. The delete command, DELE, can be issued, though the deletion does not actually take place until the next state. This is very important to the exploitation of this vulnerability.

The final command that the client issues is the QUIT command. This command causes the session to enter its third and final state, the Update state. Messages flagged for deletion are deleted, and the TCP connection is closed.

IMAP4 – Internet Message Access Protocol (Version 4, Revision 1)

The Internet Message Access Protocol is used to manipulate and retrieve email messages from a server. It is different from the Post Office Protocol in that the client can manipulate mailboxes, instead of just retrieving messages. An IMAP session consists of four states: Non-Authenticated, Authenticated, Selected, and Logout.

The IMAP server listens on TCP port 143. The client establishes a TCP connection with the server. At this point, the session is in its Non-Authenticated state. Using the LOGIN or AUTHENTICATE command, the client provides its credentials, and if successful, enters the Authenticated state.

In this state, the client must select a mailbox to view. This is accomplished through the use of the SELECT and EXAMINE commands. Once a mailbox has been selected, the session enters its Selected state.

The Selected state is similar to the Transaction state of the Post Office Protocol, in that it is where the majority of the commands, including the retrieval and deletion of messages, occur. Messages are retrieved by using the FETCH command. The DELETE command flags messages for deletion, though they are not actually deleted until the EXPUNGE or CLOSE command is executed.

Finally, the LOGOUT command is issued, causing the session to enter its Logout state. Here the TCP connection is closed.

4. How The Exploit Works

The header of an email message contains several fields filled with data, including the sender's email address, the route that the message followed to arrive at its destination, a content-type indicator, and the date/time it was sent. This vulnerability is contained in the date/time field, specifically the Greenwich Mean Time (GMT) section.

This is a classic textbook example of an exploitable buffer overflow. Basically, Outlook and Outlook Express do not properly parse the Greenwich Mean Time section of the date/time field, overflowing the buffer intended to store the value of the field when an exceptionally long string of characters is encountered. A malicious user can craft an email with extra data in this field that will perform one of two functions: crash the Outlook client or execute arbitrary code.

Crashing The Outlook Client

In order to crash the Outlook client, all the malicious user has to do is follow the GMT section of the field with a random string of characters. When the recipient of the message connects to the mail server and attempts to use Outlook or Outlook Express to download this message, the buffer will overflow, causing an invalid page fault error. In most cases, this will force the user to close Outlook, though occasionally, depending on the particular random characters chosen and the length of the string, it will cause the entire system to become unstable, requiring a reboot.

Because the Outlook client crashes while the header of the message is still being downloaded, the message is not deleted from the mail server. This is because, in the case of a POP3 connection the session is never issued a QUIT command, and never reaches its Update state where messages are deleted. In the case of an IMAP4 connection, the session is never issued an EXPUNGE or CLOSE command, which commits deletions. Therefore, the next time the recipient of the malicious message attempts to retrieve his or her mail, the crash will happen again. It will keep happening until the message is manually removed from the server.

Executing Arbitrary Code

Executing arbitrary code is more difficult than simply causing a crash. It is also more powerful. The attacker could remotely cause the recipient to run a particular executable, delete a specified file, or any number of other functions. As with the crashing technique, the recipient does not need to open or preview the message in any way; he or she must simply attempt to retrieve it from the server.

In order to execute arbitrary code on the recipient's system, the bytes added to the GMT section need to be very carefully selected, as they are actually the compiled form of the code that will be executed. As with the crashing technique, the string of characters is added to the end of the GMT section. When the buffer overflows, the attached code is executed by the recipient's system.

While so far there have been no reports of this technique being used maliciously, USSR has created a sample utility that generates emails with headers that open a browser window and bring the recipient to their web site. Another example, a link for which is found in the Source Code section of this document, opens the FreeCell game that comes with Windows. Since the source code to both of these examples is available, it is conceivable that some industrious hacker has modified one of these headers and could unleash it at any time.

Limitations

A limitation of this vulnerability is that the code is executed with the permissions that the user has on the system. Therefore, if the user is logged in as a guest on the system, the attacker's code should not have the ability to modify or delete crucial system files, while a user logged in as an administrator would provide the attacker with full permissions on the system.

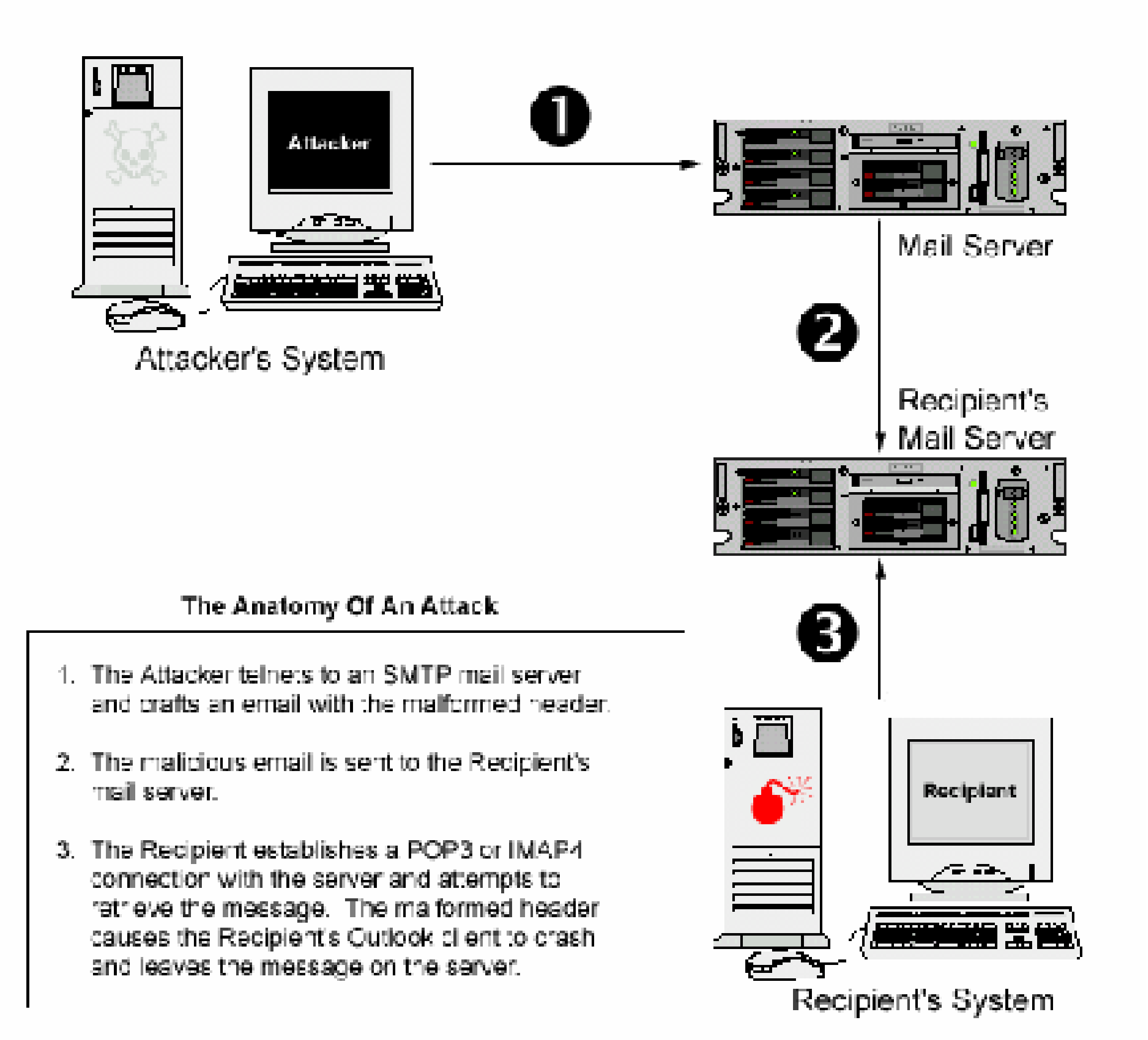
Also, this vulnerability is only present if Outlook is configured to use the POP3 or IMAP4 protocol to retrieve messages. If a system is only using MAPI (Messaging Application Programming Interface), which is only available to Outlook and is commonly used in corporate settings, it is unaffected by this vulnerability.

All Outlook Express users, however, are vulnerable.

5. Description of Variants

The one known variation on this exploit is achieved by encoding a message with a malformed header, and attaching it to another message via Outlook's MIME attached message format. This has the advantage of bypassing any Sendmail filters that may be defending against the exploit. The disadvantage is that it is less powerful than the original version of the exploit in that it must be opened or previewed by the recipient in order to activate it.

6. Diagram Of The Attack



The Anatomy Of An Attack

1. The Attacker telnets to an SMTP mail server and crafts an email with the malformed header.
2. The malicious email is sent to the Recipient's mail server.
3. The Recipient establishes a POP3 or IMAP4 connection with the server and attempts to retrieve the message. The malformed header causes the Recipient's Outlook client to crash and leaves the message on the server.

7. How To Exploit The Malformed Header Vulnerability

This vulnerability is very easy to exploit, either through a utility or manually. While so far no hacker-made utilities have surfaced, there is one utility aptly named Malformed Email Spawner that was created by Underground Security Systems Research, the organization who informed Microsoft about the vulnerability, to test and demonstrate the exploit.

Malformed Email Spawner is available as both a Linux/Unix Perl script and as an executable compiled for the Windows platform. When you run Malformed Email Spawner, it creates and sends an email with a malformed header that causes the recipient's Outlook client to open a browser and go to USSR's web site when it is downloaded from the server.

In order to run the Perl version of Malformed Email Spawner, you must provide the script with the IP address of an SMTP server and the intended recipient's email address. It randomly generates an address to put in the From field.

Example:

```
./outoutlook.pl -h 123.123.123.123 -m victim@domain.com
```

The Windows version is just as easy to use, though it lacks the capability to randomly generate a From address. You must supply the IP address of an SMTP server, a From address, and the intended recipient's email address.

Example:

```
outoutlook –s123.123.123.123 -sender:badguy@domain.com
-to:victim@domain.com
```

Note that the example above is one command, and is meant to be on one line, but wrapped because of length.

Manually crafting a malformed header is nearly as easy. Simply telnet to an SMTP server and execute the following:

```
HELO
MAIL FROM: badguy@domain.com
RCPT TO: victim@domain.com
DATA
Date: Sun,28 Jan 2001 10:56:22
+0000111111111111111111111111111111111111111111111111111111111
.
QUIT
```

Note that the two lines highlighted in purple are meant to be one line, but wrapped because of length. There is one space between the 10:56:22 and the +00001111…

Unlike the Malformed Email Spawner utility, this will create an email that crashes the Outlook client, rather than opening a browser. Changing the 1’s to bytes of compiled code will allow you to craft an email that runs the code.

8. Signature Of The Attack

Since the vulnerability is exploited by overflowing the date field buffer, any email message that has extraneous data in this field could potentially be an attempt at this attack. It will appear as a set of bytes following the standard GMT adjustment factor in the date/time field.

However, if the user can only access his or her email through Outlook, and the attacker has sent a message that causes the email client to crash, the user will not be able to view the email header as it will not finish downloading from the server. An error message similar to the following will occur:

```
OUTLOOK caused an invalid page fault in module <unknown> at
00de:00aedc5a.

Registers:
        EAX=80004005 CS=016f EIP=00aedc5a EFLGS=00010286
        EBX=70bd4899 SS=0177 ESP=0241ef94 EBP=31313131
        ECX=00000000 DS=0177 ESI=0241efc6 FS=2b57
        EDX=81c0500c ES=0177 EDI=0241efc4 GS=0000
Bytes at CS:EIP:

Stack dump:
        0241f360 0241f554 00000000 00000001 00000000 004580d0
        00000054 00000054 0241efc4 0000003b 00000100 00000017
        3131312b 31313131 31313131 31313131
```

Each time the user attempts to connect to the mail server, his or her email client will invariably crash when it attempts to download the malicious message. This will also identify an attack.

If nothing has been done to defend against malformed headers, an attacker could conceivably write code that would execute as a background process, and not crash the recipient’s system. The message would appear normal unless the recipient knew enough to look at the date field in the header. Even if the recipient did figure out what was going on, the damage has already been done.

9. How To Protect Against Malformed Headers

The preferred way of protecting against this vulnerability is by upgrading inetcomm.dll. Two different methods of upgrading the inetcomm.dll are discussed below.

Method #1: The Patch

Microsoft has released a patch that will eliminate this vulnerability in Outlook Express 5.01. In order for this patch to function properly, Internet Explorer 4.01 Service Pack 2 or Internet Explorer 5.01 must be installed. If any other version of Internet Explorer is installed, an error message will be generated, and the patch will fail. The patch can be found at the following URL: http://www.microsoft.com/windows/ie/download/critical/patch9.htm

Method #2: Internet Explorer Upgrade

Installing Internet Explorer 5.01 Service Pack 1 or Internet Explorer 5.5 will eliminate the vulnerability as well. This method will only work if an installation option that installs the upgraded Outlook Express components is chosen. Choosing the default installation will do this.

Note that if the system in question is running Windows 2000, installing Internet Explorer 5.5 will not solve the problem because it will not install the upgraded Outlook Express components. Instead, Windows 2000 users should install Windows 2000 Service Pack 1.

Other Options

Another way to protect against this vulnerability is to configure the mail server to filter out messages with abnormally large date fields. A Sendmail filter that rejects messages with date fields larger than 60 characters is shown below:

```
LOCAL_CONFIG
Klinetoolong regex -a@MATCH ^.{60,}$

LOCAL_RULESETS
HDate: $>+CheckDate

SCheckDate
R$*                     $: $(linetoolong $1 $)
R@MATCH                 $#error $: 553 Date Header too long error
R$*                     $@ OK
```

While using a Sendmail filter such as this will protect against the vulnerability, it is best to upgrade inetcomm.dll. The patches and upgrades listed above address additional vulnerabilities in Microsoft Outlook and Outlook Express.

10. Conclusion

The Microsoft Outlook / Outlook Express GMT Field Buffer Overflow Vulnerability is easy to exploit, potentially powerful, and hard to detect until the damage has already been done. It can be used to temporarily deny service to selected users by crashing their systems, or to run malicious code on a user's machine, possibly downloading and executing trojans, deleting files, or anything else an inventive hacker can imagine. Fortunately, it is also very easy to protect against. Simply upgrading a single DLL can eliminate this vulnerability, forcing the hackers to find another way to get into your system.

11. Source Code

Malformed Email Spawner - Unix/Linux Perl Version:
http://www.ussrback.com/outoutlook.pl

Malformed Email Spawner - Windows Version Executable:
http://www.ussrback.com/outoutlook.exe

Malformed Email Spawner - Windows Version Source Code:
http://www.ussrback.com/outoutlook.zip

Code that opens FreeCell when the message is received:
http://packetstorm.securify.com/0007-exploits/outlook.advisory.txt

12. Additional Information And References

RFC1725: The Internet Official Protocol Standard document for the Post Office Protocol (POP):
http://www.landfield.com/rfcs/rfc1725.html

RFC2060: The Internet Official Protocol Standard document for the Internet Message Access Protocol (IMAP):
http://www.landfield.com/rfcs/rfc2060.html

Underground Security Systems Research homepage. They have information on this and several other vulnerabilities:
http://www.ussrback.com

SecurityFocus – malformed header information:
http://www.securityfocus.com/frames/?content=/vdb/%3Fid%3D1481

Microsoft Security Bulletin. This is the official documentation of the vulnerability:
http://www.microsoft.com/technet/security/bulletin/ms00-043.asp

Microsoft Security Bulletin FAQ. Information on technical support and answers to general questions about the vulnerability are found here:
http://www.microsoft.com/technet/security/bulletin/fq00-043.asp

BugNet article on the vulnerability:
http://www.bugnet.com/alerts/bugalert_000721.html

Internet Explorer 5.01 Service Pack 1:
http://www.microsoft.com/Windows/ie/download/ie501sp1.htm

Internet Explorer 5.5
http://www.microsoft.com/windows/ie/download/ie55.htm