

VIRUS ANALYSIS 1

Bad Transfer

Peter Ferrie and Péter Ször
Symantec Security Response, USA

In December 2001 *Symantec* received its one millionth customer sample submission. We waited for the big moment, but the millionth submission arrived sooner than we expected. This is because a new worm, Win32/Badtrans.B@mm, was making its way quickly around the Internet. The worm was released in late October 2001 and during December we received 30,000 submissions of this worm alone. This is an extremely high number for a single month and at least twice as many as we have experienced before.

So what happened to Badtrans? Why did it become so widespread all of a sudden? The original variant was in the wild from April 2001 and did not attract much attention, even though it was reported to the WildList.

First of all there are a number of new features in this worm that would easily cause it to be considered a new variant. The '.B' letter was fairly arbitrary, given that there were at least three variants of the worm known at the time. Badtrans uses techniques picked up from the Nimda virus, and it is clear that these techniques contributed highly to its success.

Configuration Bits

This worm arrives as an email with one of several attachment names and a combination of two appended extensions. The attachment contains the worm code and an appended block of configuration data which controls its behaviour. Clearly the author wanted to change the behaviour of the worm without recompiling the code. Thus the configuration data at the end of worm work much like a .ini file. The worm's author probably had a patch tool to create different behaviours during testing.

The worm also has the ability to replace existing copies of itself with newer versions. Badtrans.B is written in Visual C++ and packed with UPX, which is a common runtime compressor. With such a widespread worm, we might have expected to have seen new variants by now, however the configuration data contain file offsets which change if the file is altered in any way (such as repacking) and such alterations will prevent the worm from running correctly.

The configuration data contain various things, such as the name of the registry key, the registry value and data to use, the names of the files to create (for the worm itself, the key logger, and the data files), and the texts that will cause the key logging to begin. Additionally, there are control bits that are checked by the worm code, and a unique identifying value for controlling the overall execution.

The control bits control the logging, the encryption, the directories in which files are created, and what is stolen (keystrokes and/or cached passwords). Thus many things can change based on these values.

The unique value is used as a parameter to verify the requests to run a new copy and delete the old one. When the worm is executed for the first time, it will search for and terminate all other running copies of itself. Then it will append the unique value to the word 'Restart_' and run again with this parameter. If this parameter has been specified already, then the worm will run itself yet again, but with an additional parameter which is the unique value appended to the word 'Kill_'. The purpose of the Kill command is to delete the file that was used to launch the worm initially.

Auto Launcher

The worm uses the malformed MIME exploit to execute automatically. The emails are HTML format combined with a malformed MIME header that causes *Microsoft Outlook* to execute the attachment immediately and without prompting. More on the exploit, including the necessary patches to protect the system against such an attack can be found at <http://www.microsoft.com/technet/security/bulletin/MS01-020.asp>.

When *Badtrans.B* is first executed, it copies itself to %System% or %Windows%, depending on the control bits, using the filename contained in the configuration data (currently 'kernel32.exe'). Then it registers itself as a service process (*Windows 9x/ME* only) to hide its presence from the Task List in *Windows*. It creates the key logging files in %System%, whose names are specified in the configuration data (currently 'kdll.dll' and 'cp_25389.nls').

The log file is encrypted with a simple algorithm, whose keys come from a string in the configuration data (currently 'uckyjw@hotmail.com'). Andreas Marx at *AV-Test.org* developed a *Windows* application to decrypt the encrypted log file. This may be of use to anybody who wants to know what has been logged and possibly distributed to the hacker.

The DLL is loaded and several functions are accessed dynamically from it. The code of the DLL is stored in the resource section of the worm's code. That explains the choice of UPX, a packer that does not pack resource sections. There are some buggy packers that pack resources and therefore cause problems for certain applications that use *Windows* resource APIs.

%Windows% and %System% are variables. The worm locates the \Windows folder (by default this is C:\Windows or C:\Winnt) or the \System folder (by default this is C:\Windows\System or C:\Winnt\System32) and copies itself to that location.

Password Stealing

A timer is used to examine the currently open window once every second and to check for a window title that contains

particular characters. Currently, these are 'LOG', 'PAS', 'REM', 'CON', 'TER' and 'NET'. These texts form the start of the words LOGon, PASsword, REMote, CONnection, TERminal and NETwork, respectively. There are also Russian versions of the same words in the list. If any of these words are found, then the key logging is enabled for 60 seconds. When the logging delay expires (currently every 30 seconds), the log file and the cached passwords are sent to one of several addresses (some of which are currently not operational, some contain obscene words and are not listed here), using one of several SMTP servers. The addresses are:

ZVDOHYIK@yahoo.com	rmxqpey@latemodels.com
DTCELACB@yahoo.com	muwripa@faieresuivre.com
WPADJQ12@yahoo.com	cxkawog@krovatka.net
I1MCH2TH@yahoo.com	ssdn@myrealbox.com
udtzqcc@yahoo.com	bgnd2@canada.com
YJPFJTGZ@excite.com	smr@eurosport.com
JGQZCD@excite.com	tsnlqd@excite.com
OZUNYLRL@excite.com	eccles@balls.net
XHZJ3@excite.com	fjshd@rambler.ru
S_Mentis@mail-x-change.com	

The SMTP servers are:

mx2.mail.yahoo.com	mail5.rambler.ru
mail.ifrance.com	mail.canada.com
fs.cpio.com	smtp.myrealbox.com
mail.monkeybrains.net	mail.ukr.net
usa-com.mr.outblaze.com	mail-fwd.rapidsite.net
mta.excite.com	imap.front.ru
inbound.latemodels.com.criticalpath.net	
inbound.balls.net.criticalpath.net	

The email addresses as well as the server names, are encrypted in the worm's code. Since there are Russian server names and Russian words in the password stealing routine we can safely assume that this worm has a Russian origin.

After 20 seconds, the worm shuts down if the appropriate control bit is set.

Sending Mail

If RAS support is present on the computer, then the worm waits for an active RAS connection. When such a connection is made, with a 33 percent chance, the worm searches for email addresses in *.htm* and *.asp in %Personal% and Internet Explorer %Cache%. If it finds addresses in these files, then it sends mail to those addresses using the victim's SMTP server. The attachment name will be one of the following:

Pics	images
README	New_Napster_Site
news_doc	HAMSTER
YOU_are_FAT!	Stuff
SETUP	Card
Me_nude	Sorry_about_yesterday,
info	docs
Humor	fun.

In all cases, MAPI will also be used to find mail to which the worm will reply. The subject will be 'Re:'. In that case, the attachment name will be one of the following:

PICS	IMAGES
README	New_Napster_Site
NEWS_DOC	HAMSTER
YOU_ARE_FAT!	SEARCHURL
SETUP	CARD
ME_NUDE	Sorry_about_yesterday
S3MSONG	DOCS
HUMOR	FUN

The worm appends two extensions. The first should be one of the following: .mp3, .zip, .doc, but because of a bug, .zip cannot be chosen. The second extension that is appended to the file name is .pif or .scr. The resulting file name would be something like, for example, CARD.doc.pif or ME_NUDE.mp3.scr and so on.

If SMTP information can be found on the computer, then it will be used for the From: field. Otherwise, the From: field should be one of the following:

"Mary L. Adams" mary@c-com.net
 "Monika Prado" monika@telia.com
 "Support" support@cyberramp.net
 "Admin" admin@gte.net
 "Administrator" <administrator@border.net>
 "JESSICA BENAVIDES" <jessica@aol.com>
 "Joanna" <joanna@mail.utexas.edu>
 "Mon S" <spiderroll@hotmail.com>
 "Linda" <lgonzal@hotmail.com>
 "Andy" <andy@hweb-media.com>
 "Kelly Andersen" Gravity49@aol.com
 "Tina" <tina0828@yahoo.com>
 "Rita Tulliani" <powerpuff@videotron.ca>
 "JUDY" <JUJUB271@AOL.COM>
 "Anna" <aizzo@home.com>.

However, due to a bug, only every second name in the list

can be chosen. In order to prevent multiple emails to the same person, Badtrans.B writes email addresses to the %System%\Protocol.dll file. Additionally, the underscore (_) character is prepended to the sender's email address, which interferes with replying to infected mails to warn the sender of infection (for example, user@website.com becomes _user@website.com).

Before sending email, the worm will look in the registry for the name of a DNS server. If one cannot be found, then it will use a default server whose IP address is stored in the worm code.

The DNS server is used to verify that the domain specified in an email address is truly valid. This idea exists in Nimda, too, however in the case of Badtrans.B the result is ignored and email is sent even if the domain cannot be verified as valid.

After sending the mail, the worm adds itself to the registry key specified in the configuration block, using the specified value and data (currently, these are 'HKLM\Microsoft\Windows\CurrentVersion\RunOnce', 'Kernel32', and 'kernel32.exe'). This causes the worm to run the next time *Windows* is started. These values can differ based on the control bits mentioned previously.

Conclusion

Evidently the use of exploits in computer viruses is becoming increasingly common. Additionally, many malicious hackers are creating mass-mailing worms by combining backdoors and other password-stealing applications to make them more successful. Just like Sircam, W32.HLLW.GOP@mm started up as a password-stealing Trojan and became more successful once the MIME encoding was added to the SMTP mass mailing that was already present.

Hackers are becoming increasingly interested in your personal information! We should all be sure to keep aware of recent exploits and apply all security patches to protect ourselves from a lot of trouble. Apparently the Nimda virus spread this message well enough, yet Badtrans.B became successful with a very similar strategy.

Win32/Badtrans.B mm

Alias:	I-worm.Badtrans.B, W32/Badtrans.B mm.
Type:	SMTP mass mailer that uses malformed MIME exploit to automatically execute itself if <i>Microsoft Outlook</i> is not patched.
Size:	29,020 bytes.
Removal:	Stop and delete worm process, fix registry values.