

VIRUS ANALYSIS 3

Sircamstantial Evidence

Peter Ferrie and Péter Ször
Symantec Corporation, USA

Although Sircam made a name for itself sending out random files and personal documents from infected PCs, not all of the information that spread with Win32/Sircam was actually spread by the worm itself. Almost as soon as updated descriptions of Sircam were posted to Web sites, selected texts from these descriptions appeared on other sites, complete with identical spelling errors and other inaccuracies.

Evidently the emerging complexity of new 32-bit worms is proving a tough challenge for every one of us in this business: if ExploreZip was boring and difficult to analyse, Sircam was a major pain. Sircam's author tried to make sure that the analysis would not be straightforward. The worm is written in a high-level language, but all the string constants (including its email message) are encrypted in such a way that it took a little while to decrypt completely (at least for some of us).

Start Your Engines

Win32/Sircam usually arrives as an attachment to an email. This attachment is special, because it contains not only Sircam itself, but an additional file (appended to the end of Sircam), which has been 'stolen' from the Personal or Desktop directory of the sender's computer.

When this attachment is run, Sircam will detach the stolen file and display it. The way in which the file is displayed depends on its suffix. If the suffix is .doc, Sircam will attempt to run *WinWord*. If this fails, then *WordPad* will be used instead. If the suffix is .xls, Sircam will run *Excel*. If the suffix is .zip, Sircam will run *WinZip*. If the suffix matches none of these, Sircam will run *dll32*. Even in the event that no suitable application can be found to display the file, Sircam will install itself in the system. There is the additional risk that the stolen file might contain confidential information, or even macro viruses, in the case of *WinWord* and *Excel* documents, which Sircam will help to spread further.

Sircam begins installation by attempting to copy itself into the Recycle Bin. It is assumed that this is called 'Recycled', and that it is located on the drive that contains *Windows*. (The hard-coded directory name is the one thing that prevents Sircam from functioning correctly in *Windows NT/2000/XP*, in which the Recycle Bin is named 'Recycler'.)

Once Sircam has placed itself in the Recycle Bin, where it is hidden from the view of programs such as *Explorer*, Sircam will copy itself to the System directory, using the

name 'SCam32.exe'. A new value, *Driver32*, is placed in the *RunServices* key in the registry, which refers to the *SCam32.exe* file. Thus, the worm will run whenever *Windows* is booted.

Additionally, *SirCam.exe* is specified to be the application that handles requests to run other .exe files, by changing the *exefile Open* key (HKCR\exefile\shell\open\command) in the registry. In this way, Sircam gains control whenever an application is run. This is not a new technique. In fact, the *PrettyPark* worm was one of the first viruses to utilize this technique, more than two years ago.

Not content with such control, Sircam will also watch for requests to run applications in the Desktop directory (referred to by ...\Explorer\Shell Folders\Desktop in the registry). When such a request is made, Sircam will prepend itself to the specified file, before running the application! Thus, even if the registry is restored and the files are removed from the Recycle Bin, infected files could remain in the Desktop directory.

Spread the Word

After installation is complete, Sircam will search the local network for computers which allow unrestricted access. Sircam will copy itself to the Recycled directory on each unprotected computer that is found and append a line to the *Autoexec.bat* file. The line will run the Sircam file from the Recycle Bin whenever the computer is booted. Then Sircam will rename *rundll32.exe* to *run32.exe* in the *Windows* directory on the remote computer, and create another copy of Sircam in its place. Neither the copying of the Sircam files to remote computers nor the emailing to other users occurs in *Windows NT/2000/XP*, however each of the other effects can be observed.

Randamn

The date-activated trigger is checked at this point, however two factors prevent it from working. The least significant of these factors is the dependency on the date format used by the computer, which Sircam requires to be dd/mm/yy (as opposed to mm/dd/yy, for example). However, the more significant factor is that the trigger contains a random component, but the random number generator is never initialized, resulting in there being no chance of producing the required condition.

Unfortunately, there are two other ways in which the payload can be activated. One is by renaming one of the three files, *SirC32.exe*, *SCam32.exe*, or *rundll32.exe*, to another name and running that file. The other is to run an attachment whose stolen file contains the characters 'FA2' not followed immediately by the characters 'sc'. The

payload deletes all files in all directories on the drive that contains *Windows*.

The missing randomiser initialization prevents Sircam from copying itself to the Windows directory as ScMx32.exe, and copying itself to the Startup directory (referred to by ...\\Explorer\\Shell Folders\\Startup in the registry) as Microsoft Internet Office.exe. It also prevents Sircam from creating, on October 16, a file that fills the remaining disk space.

I'm Sending You a Letter

When Sircam is run for the first time, it will change *Internet Explorer's* Download directory (referred to by HKCU\\Software\\Microsoft\\Internet Explorer\\Download Directory in the registry) to point to the Desktop directory, in order to maximize the use of the prepending routine mentioned earlier.

During the second execution, Sircam will gather email addresses into files stored in the System directory. Sircam searches for email addresses in *Internet Explorer's* Cache directory (referred to by HKCU\\Software\\Microsoft\\WindowsCurrentVersion\\Explorer\\Shell Folders\\Cache in the registry), the user's Personal directory (referred to by HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders\\Personal in the registry), and the directory that contains the *Windows* Address Books (referred to by HKCU\\Software\\Microsoft\\WAB\\WAB4\\Wab File Name in the registry), in files whose name begins with 'sho', 'get', or 'hot', or whose suffix is 'htm' or 'wab'.

Thus, Sircam creates a file called scy1.dll, which contains the addresses from %cache%\\sho* files, sch1.dll contains the addresses from %cache%\\get* and %cache%\\hot* files, sci1.dll contains the addresses from %cache%*.htm files, sct1.dll contains the addresses from %personal%*.htm files, and scw1.dll contains the addresses found in *.wab files.

If the Address Book registry key is not found, Sircam will search for WAB files in the System directory instead. After creating the lists of email addresses, Sircam will search for files to attach to the emails that it will send. The list that is created consists of the name of every .doc, .xls, and .zip file in the user's Personal and Desktop directory and is called scd.dll. An apparent oversight on the part of Sircam's author prevents the inclusion of .exe files in the list.

On the third and subsequent runs, and if an active connection to the Internet exists, Sircam will retrieve the information required to send email using SMTP. Sending mail using SMTP avoids relying on an email program such as *Outlook*. The SMTP information consists of the current user's email address (HKCU\\Software\\Microsoft\\Internet Account Manager\\Default Mail Account\\Accounts\\SMTP Email Address in the registry), the address of the email server (HKCU\\Software\\Microsoft\\Internet Account Manager\\Default Mail Account\\Accounts\\SMTP Server in

the registry) and the user's display name (HKCU\\Software\\Microsoft\\Internet Account Manager\\Default Mail Account\\Accounts\\SMTP Display Name in the registry).

If, for some reason, this information does not exist, Sircam will use prodigy.net.mx as the email server, and the user's logon name as the email address and display name. Then Sircam will attempt to connect to an email server. First, it will try the user's own email server (or prodigy.net.mx). If this fails, Sircam will attempt to connect to the email server of the person who sent the infected email. This is possible because Sircam carries within it the email information of the previously infected person. If this connection fails, then Sircam will attempt to connect to goeke.net, then enlace.net, then dobleclick.com.mx.

Compositions

If one of the connections to an email server is successful, an email is constructed in the following way: if the language used on the current user's computer is Spanish, then Sircam will send email in Spanish, otherwise it will use English.

The email body consists of three lines. The first line of the email body is always 'Hola como estas?' in Spanish, and 'Hi! How are you?' in English; the third line is always 'Nos vemos pronto, gracias.' in Spanish, and 'See you later. Thanks' in English. The second line is chosen from the following list, in Spanish:

- 'Te mando este archivo para que me des tu punto de vista'
- 'Espero me puedas ayudar con el archivo que te mando'
- 'Espero te guste este archivo que te mando'
- 'Este es el archivo con la informacion que me pediste'

and, in English:

- 'I send you this file in order to have your advice'
- 'I hope you can help me with this file that I send'
- 'I hope you like the file that I sendo [*sic*] you'
- 'This is the file with the information that you ask for'

However, since the randomiser is not initialized, the choice is reduced to the first line alone, until October 16, or until Sircam has been run at least 8000 times, at which point the last line can be chosen, too.

As long as an active connection to the Internet exists, Sircam will send email to every address in each of the email lists that it created. It will send an email three times to each address in the scw1.dll list, then once each to all the other addresses, in the order: scy1.dll, sch1.dll, shi1.dll, and sht1.dll, before starting again with scw1.dll.

Sircam keeps the current mailing position in the registry, so if the connection is broken and restored later, Sircam can continue to send mail as though it were never interrupted.

Interestingly, Sircam ensures that the current user never receives an email from Sircam. In the case that the recipient is the current user, Sircam will send the mail instead to email address `otrorollo@esmas.com`.

I'm Sending You a File ...

For each email it sends, Sircam will select a file randomly from the `scd.dll` list, prepend itself to that file, attach an additional extension, chosen randomly from 'pif', 'lnk', 'bat', or 'com', and send the email. The randomiser has no impact on the emailing routine. If an Internet connection exists for long enough, every recipient will, eventually, receive every file in the list, each with a different extension. To avoid overloading email servers, Sircam remains idle for one minute between sending each email.

In some ways, Sircam's success has had much to do with luck: the emails Sircam constructs are unintentionally malformed such that it appears, to some email scanning products, that the mail contains no attachment. This has allowed the worm to slip past some gateway scanners, though this is far from the sole reason for Sircam's wide-spread distribution.

Conclusion

Evidently SMTP propagation is the hot topic of the year. Even the first Win32 mass-mailer, Parvo (see *VB*, January 1999) used an SMTP engine. However, most of the worms that have utilized SMTP mailing so far have got a few things wrong. Thanks to the implementation mistakes and bugs, it was a little while before SMTP worms could take their real place. Most of the previous worms have lacked some important detail in their spreading mechanism. For instance, Magistr often sends clean files or files that will not run on the recipients' computers because of some missing DLLs.

As VBS creations are controlled with proactive technologies, so virus writers turn their attention to the creation of more dangerous binary worms. One thing is for sure: there is more to come!

Name: W32/Sircam.worm

Aliases:	W32.Sircam.Worm@mm, Win32/SirCam@mm, Backdoor.SirCam.
Type:	Win32 SMTP mass-mailer worm, preponder.
Payload:	Propagates confidential files, attempts to delete all files on disk, attempts to eat up free space on disk.
Removal:	Fix registry and modified files, delete standalone worm copies, restore infected once from backups.