VIRUS ANALYSIS 2

Melissa – The Little Virus That Could…

Ian Whalley
Sophos Plc

[*After this analysis* VB *gauges* IVPC's *reaction to Melissa. Sarah Gordon's feature also mentions its author. Ed.*]

Saturday 27 March was going to be a quiet day – or at least, that was what I thought when I got up at around 8.30am. After a quick breakfast, I dialled my ISP to retrieve my email and read some news. Shortly afterwards, I was in the car on the way to the office.

Newsgroups, mailing lists, on-line news services – all were talking about one thing; a macro virus called Melissa that was (apparently) causing havoc in North America. Companies were reported as being effectively forced to stop all internal and external email in an effort to halt its spread. Consequently, after the initial creation of a patch for our product to detect and remove the virus, a more detailed analysis followed.

The Nitty-gritty

In and of itself, Melissa is almost entirely uninteresting – it is a perfectly standard *Word 97* Class-style infector. The first time an infected document is opened on a given machine, the virus receives control via the standard Document_Open() macro.

The first thing it attempts to do is deactivate macro security. It checks for the value Level in the registry key: HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security. If this value is found, Melissa assumes that it is running inside *Word 2000*. Subsequently, it disables the Security… option on the Macro menu (this causes that option to appear greyed out on the menu), and then resets the Level value mentioned above to 1.

If the Level value is not found, Melissa assumes that it is running under *Word 97*. It greys out the Macro option on the Tools menu, disables format conversion warnings, *Word's* own virus protection, and prompts to save the global template. Instead of setting these options to False or 0, it sets them to (1 – 1) in an attempt to fool macro heuristics. Following this initial work, Melissa moves on to trigger the payload – more on this later.

Infection

This is fairly standard – it copies itself from the source document to the destination one using the InsertLines method on a CodeModule object. It takes care to change the first line of the macro appropriately. This is dependent upon whether it is copying itself into the global template from a document, or into a document from the global template. This is necessary because the macro has two different names – in a document, it is called Document_Open() (as mentioned above), and in the global template, it is called Document_Close().

It is worth noting at this point that Melissa has a little-noticed side effect – it will overwrite the first item in the components collection of documents and global templates which it infects. For most documents, this will not be an issue, of course – however, for global templates, it might be more of a problem.

Payloads

Melissa has two payloads. Not surprisingly, the least significant of the two is also the simplest to explain. Whether or not the virus has had to copy its body from one place to another, at the end of its execution it checks the time. If the minutes of the hour are the same as the day of the month (for example, 11.15 on 15 December, or 10.04 on 4 July), it will insert the following text into the active document, wherever the cursor happens to be:

```
Twenty-two points, plus triple-word-score,
plus fifty points for using all my letters.
Game's over. I'm outta here.
```

At this point in the virus, the following text appears in comments:

```
WORD/Melissa written by Kwyjibo
Works in both Word 2000 and Word 97
Worm? Macro Virus? Word 97 Virus? Word 2000
Virus? You Decide!
Word -> Email | Word 97 <–> Word 2000 ...
it's a new age!
```

A quick session with *Altavista* reveals that Kwyjibo and the text that the virus inserts into the current document derive from an episode of *The Simpsons* called 'Bart the Genius'. The family are playing Scrabble, and Bart says: 'K-W-Y-J-I-B-O… Kwyjibo. 22 points… plus 50 points for using all my letters! Game's over, I'm outta here… '. When asked, he defines Kwyjibo as 'a big, dumb, balding, North American ape with no chin…'.

That Other Payload

The reason for Melissa's sudden infamy is contained within the other payload, referred to at the start of this analysis. Immediately after the virus attempts to disable *Word*'s security features, it uses the CreateObject() function to intialize an instance of *Microsoft Outlook*. This will, of course, fail if *Outlook* is not installed (in fact, it only works with *Outlook 98* or later).

This is not a problem. The virus has installed the now-traditional 'On Error Resume Next' handler, so that if and when all the following commands fail, it will blunder on regardless, without telling the user that anything is wrong.

Once Melissa has obtained a running instance of *Outlook*, it asks it for a MAPI (Messaging API) namespace. In this context, 'namespace' represents 'an abstract root object for any data source', which translates into English as 'something you have to log on to and which you can retrieve information from and do stuff with'. Following this, it checks for the existence of a value 'Melissa?' in the registry key: HKEY_CURRENT_USER\Software\Microsoft\Office.

If this value is set to '... by Kwyjibo', then it skips the next set of instructions – after the payload has been executed, the virus will set that value to that string, preventing the payload from being executed more than once. Administrators should note that a system with a write-protected registry would allow the payload to execute each and every time an infected document is opened. In this case, security works against the prepared.

Then Melissa logs on to *Outlook*. I have been unable to find documentation to describe the code it is using, but when the code is run, it logs on to *Outlook* as the default user on that machine. I suspect, in many environments, *Outlook* attempts to connect to the server using the current network username and password, which would obviously work well in *Exchange*-based environments.

Melissa now iterates across all the 'members' of the MAPI session's AddressLists 'collection' – MAPI (and *Outlook*) allow the user to have multiple address books in which to store names and email addresses of both individuals and groups of individuals for easy access. Once again, in *Exchange*-based environments, one or more of these address books can be held on the server – these address books are shared between multiple users.

The impact of this type of set-up on Melissa's spread should not be underestimated. This is because it seems that in such environments, a large number of addresses in server-based address books are for groups of people.

For each list in the collection, Melissa constructs a message to the first fifty entries, with the subject line 'Important Message From <username>', where <username> is set to the name used to register the currently-running copy of *Word*. The body text is set to 'Here is that document you asked for ... don't show anyone else ;-)', and (here comes the problem), Melissa attaches the current document (which is, of course, infected) to the message, and sends it.

Melissa's Initial Spread

Melissa was distributed on Friday 26 March via a posting to the Usenet group ALT.SEX, in an infected document containing what was claimed to be a list of passwords for porn sites (LIST.DOC, contained within LIST.ZIP).

Unsurprisingly, therefore, the first document to be widely emailed by the virus was LIST.DOC itself. This has led to several stories about the virus mentioning LIST.DOC explicitly (no pun intended). However, whilst initially the mail messages generated by the virus did indeed predominantly contain LIST.DOC, as the virus naturally infected other files, other documents (often confidential ones) were transmitted as well.

The initial impact of Melissa was considerable – news stories quoted *Microsoft* officials as saying that they had been forced to shut down their outbound and inbound email servers. During the weekend of 27/28 March, only two of *Microsoft*'s five inbound mail servers were in operation. One large organization reports that between four hundred thousand and half a million email messages were generated by the virus in under three hours – after which time they also shut down their servers.

So unusual was the spread that *CERT* (an organization not normally noted for its interest in viruses) issued an alert on Saturday 27 March concerning Melissa. This gave, amongst other things, a link to an irrelevant security warning about the '*Word 97* Template Vulnerability' on *Microsoft*'s web site; information on how to update some anti-virus products to detect the virus, and an example of how to configure sendmail to reject all messages the subject lines of which start with 'Important Information From'.

While this type of patch may have been acceptable in the short term, it clearly has significant problems as a long-term anti-virus measure. As it happens, however, the problem has been magnified somewhat by the discovery that, under certain fairly unusual circumstances, the virus can mail *uninfected* documents!

Conclusion

Melissa is undoubtedly the fastest spreading virus we have ever seen. As is now documented, its speed of spread attracted the attention of US law enforcement services, who have since made an arrest, giving David Smith worldwide notoriety. *VB* will, of course, follow the case.

Melissa	
Aliases:	Maillissa.
Type:	*Word 97/2000* macro infector.
Trigger:	(1) Upon initial infection; (2) when the day and the minute are the same.
Payload:	(1) Mails the first fifty addresses in all *Outlook* address books; (2) inserts text into the current document.
Disinfection:	In a clean *Word* environment, delete the virus' module with the *Visual Basic* editor.