VIRUS CASE STUDY

A Portrait of Jini

Dr Igor Muttik
AVERT, Europe

The virus called X97M/Jini.a was first discovered in February 2000. When run, it drops an SHN.XLS file in the XLSTART folder. Despite being considered an intended, for a while all major AV scanners were able to detect and clean the virus shortly after it was first found. In such a situation, any outbreak is unlikely and we would normally only receive samples from users who have not updated scanning engines and DATs.

However, on 10 August our *AVERT* research unit in the UK received an XLS sample by email. The sample was from a customer – a big UK superstore chain – who claimed that an XLS was demonstrating unusual, virus-like behaviour. Still, neither our scanner nor any of our internal tools could find anything suspicious and even VBA heuristics were silent. The user, however, reported that the file triggered a macro protection message box in *Excel* and that once the spreadsheet was loaded a 'funny' SHN.XLS file had appeared in the XLSTART folder. This prompted me to take a closer look.

Initially, we thought the sample contained an unusual, nonviable corruption of the X97M/Jini.a virus. It took a great deal of time to establish that this is really a new virus and that it does replicate recursively. And it was great timing – within a week we received similar virus samples from two other companies, one of them heavily relying on *Excel* spreadsheets in their business.

But what is this new virus? Why did our tools let us down? The research carried out within *AVERT* revealed that we were dealing with a unique, viable corruption of the known X97M/Jini.a virus that was missing parts normally present in any VBA project (i.e. present in any VBA files with macros – be they .XLS, .DOC or .PPT).

Normal files with VBA macros have three components in the VBA project: compressed VBA source, compiled p-code for each VBA module, and executable codes (execodes) for all VBA modules. However, *Office* applications (and *Excel* is no exception) have the ability to use one of these three components if another is missing, corrupt or unrecognizable (e.g. created by another version of *Excel*).

That is what had happened – the mother X97M/Jini.a virus lost both compressed sources and p-code but had ready-to-use execodes. The new virus was later given the name X97M/Jini.a1 to denote its relation to the parent virus. X97M/Jini.a1 is a crippled but viable form of X97M/Jini.a and its VBA_PROJECT has only VBA execodes in so-called _SRP_n streams (n=0,1,2,3 etc).

When X97M/Jini.a1 replicates it is unable to return to its X97M/Jini.a state. And because VBA execodes are *Excel*-version specific the virus can replicate only under *Excel 97*. Other *Excel* versions would not understand the execodes and would not run the virus.

X97M/Jini.a1 got lucky – not one scanner used execodes to detect VBA viruses because compressed sources and p-code were easier targets. There was never any reason to scan execodes! Fortunately, our latest scanning engines have so-called ActiveDAT technology which makes it possible to implement in the DATs algorithms of any complexity. So, in a couple of days the problem was solved – scanning and cleaning of VBA execodes were implemented.

Now we knew how to solve the customer's problem and an EXTRA.DAT to detect and clean the X97M/Jini.a1 virus was sent to all the users who had the virus. The detection was also included in the regular, weekly DATs. The whole exercise took about 10 days (!) while usually the reply (and an EXTRA.DAT) for any new virus goes back in several hours. The X97M/Jini.a1 virus was described and announced to other AV experts on 22 August 2000.

At that point we encountered an unexpected problem. Some AV researchers could not replicate the new form of the virus and were arguing against the very existence of it. As time passed the mistake was, of course, rectified (subsequently, many confirmations of X97M/Jini.a1 virality were received from both AV researchers and from the field) and it was recognized that X97M/Jini.a and X97M/Jini.a1 are two different, viable forms of the same virus.

We still do not know how and where the very first sample of X97M/Jini.a1 was created. It could have been the result of incomplete cleaning by some AV product, the sample could have been manually handcrafted, or it could have been the result of experiments with a live virus. In any case, it caused a lot of trouble for both users and AV developers. We received yet another confirmation that playing with viruses is not a good idea. X97M/Jini.a1 is currently the only known case of a virus consisting of only VBA execodes. It carries the text:

```
'Hye. You have just got me.
It's shani a little jini. You may call me a
virus in your termenology
It's a good idea taking backup of you files.
I am freindly but get wild sometimes'
```

Please, in future let us be cautious, because if we do not many more viruses will 'get wild sometimes'!

And I would like to thank our customers – if it were not for their vigilance, this particular virus would have been discovered much later and could have caused a great deal more trouble.