ANALYSIS 2

SOMETHING SMELLS FISHY

Peter Ferrie
Symantec, USA

Multi-platform malware is nothing new. In 1999 we saw the W32/W97M infector Coke and W32/HLP infectors SK and Babylonia. In 2000 we saw W32/HLP infectors Dream and Pluma; in 2001 we saw W32/Linux infector Peelf, followed by Simile in 2002 and Bi in 2006. In 2003 and 2004 we saw W32/W64 infectors MSIL/Impanate and Chiton. Three new multi-platform scripting viruses were seen in 2005 (see *VB*, November 2005, p.4) – and of course, there was the Morris worm in 1988.

These points are apparently lost on Paul Sebastian Ziegler, the author of MSIL/Yakizake. The virus author wanted to call his virus 'Akikaze' (Japanese for 'Autumn wind'), but I went with the Japanese word for grilled salmon. The virus author claims that 'very few implementations of multi-platform malware exist up until now' (despite the dozen that I've just listed), so he went ahead and wrote a 'multi-platform' virus and presented it at the DEFCON 15 conference.

MULTI-WHAT?

It's unclear why Mr Ziegler thinks that his virus is multi-platform, because the platform is the environment in which the application runs. It's not the CPU on which it is running, because it needs to interact with other hardware to survive. It's not the operating system, either, if the environment is a virtual machine of some kind, or the virus exists outside of the operating system itself (for example, a boot sector virus).

In this case, the virus runs on a particular platform that has multiple implementations – which include *Microsoft .NET Framework*, *Novell Mono*, and *DotGNU Portable.NET*. The platform is a hardware-independent virtual machine. The platform has been ported to several CPU architectures, but that's irrelevant, and since it's hardware-independent, the applications running inside it can't see the CPU anyway. So it's really just the one platform. The virus is aware of the operating system, but that's also irrelevant. It's still just the one platform.

There can be exceptions, of course, such as MSIL/Impanate (see *VB*, November 2004, p.6). Impanate is a file infector that understands both the 32-bit and 64-bit MSIL file formats. It's an MSIL virus, so it's not multi-platform, but it is multi-platform-*aware*. Yakizake is neither of these things.

THE VIRUS

The virus begins by looking for the *Thunderbird* address book. There is code to deal with Unix systems, *Macintosh* systems, and *Windows* systems, however due to a bug, only the Unix and *Windows* code works. The bug is that the code to check for the *Macintosh* system is identical to the code to check for all other Unix systems. As a result, the *Macintosh* code can never be reached. This means that the virus cannot replicate from *Macintosh* systems.

In the case of *Windows* systems, the virus will attempt to terminate all instances of the *Thunderbird* executable, in order to gain control over the address book.

The virus creates a list of all addresses that it can find. The first version of the virus accepts addresses in '*@*.*' format, where '*' can be any character. The second version of the virus restricts this to one or more case-insensitive alphanumeric characters before and after the '@', and no longer checks for the '.' character.

The virus also looks inside 'prefs.js' for the SMTP server information, and inside 'signons.txt' for the SMTP server password.

The virus creates different email messages, depending on certain characteristics. If the virus is sending from a German system to a German user, the subject will be 'Programmierung' and the message will be in German (an almost exact translation of the English message), otherwise the subject will be 'Programming' and the message will be in English. The virus chooses an 'advanced' message body if it is running on a Unix system, and the string '/gcc' exists in %path% or if it is running on a non-Unix system, and 'Visual Studio' exists in the %ProgramFiles% directory.

The 'advanced' message body is:

```
Hi,

I wrote this program using a new approach. Please
tell me what you think of it.
```

The 'average' message body is:

```
Hi,

I have recently started to try out programming!

This is one of my first programms. What do you think
of it?
```

On Unix systems, both messages continue with:

```
If the programm should not work instantly on your
non-windows-system you probably need to execute it
using mono. (mono-project.com)
```

After constructing the message, the virus sends it to each recipient in turn, using the host SMTP server and credentials, with the virus executable as an attachment. When the mailing is finished, the virus exits. There is no payload.

DUMB AND DUMBER

To create a virus because one did not exist before is just dumb. To incorrectly call it multi-platform is even dumber.