

'For most people even a single piece of malware is too much – especially if they are currently affected by it.'

Robert Sandilands Authentium

HOW MUCH MALWARE IS REALLY OUT THERE?

Since the beginning of the AV industry more than two decades ago, the amount of malware in existence has been an often-debated point. Answers range from none to an infinite amount.

If you use a custom operating system on custom hardware that is running applications that are of no importance to anybody but yourself, then you are probably right to assert that there is no malware. There are probably no financial (or other) incentives to attack such a system.

To get closer to a real answer we need to focus a bit broader than this contrived example – although there may be some truth in the observation that there is only as much malware as people want to know about. Whether that leaves you with no malware or with an infinite amount depends on the perspective. For most people even a single piece of malware is too much – especially if they are currently affected by it.

Let us assume that any platform that is somewhat accessible and has either a large enough userbase or great enough value will eventually be attacked by malware. We can see this with the recent growth of *Mac* malware and also with something like Stuxnet that (probably) attacked a single, but very high-value target.

In the mid 90s we were in a position where we could accurately count the number of viruses that had been seen. This was possible for several reasons:

1. The number of new viruses was small enough for each sample to be identified and analysed in detail.
2. It was easy to determine which part was virus and which part was the infected application.
3. The size and complexity of the malware was quite limited.

If you took one of the polymorphic file infectors from the 1990s and infected 100 million clean files then you could get 100 million unique infected files. If you then counted that like most people count malware today you could say that you had 100 million pieces of malware. This would be incorrect, but it is how malware tends to be counted these days.

There are several reasons for this. The first is that modern malware is probably several orders of magnitude larger and more complex than the malware that was around in the mid 90s. The second major reason is the use of packers to obfuscate the malware. The last, and probably most important reason is the location of the polymorphic engine. This has moved from being inside the 1990s virus to being on the server today, where analysts generally cannot access it.

In the old days we could carefully replicate most pieces of malware in a protected and isolated environment and gain a good understanding of how each morphs and we could therefore use a small number of very efficient signatures to detect those pieces of malware.

These days most pieces of malware won't work without what appears to be a real Internet connection. They generally also won't replicate. To get a 'replicated' copy you either have to be reinfected or download a new copy of the malware.

Not only that, but analysing a specific piece of malware in detail can take weeks to months. For example, people are still busy analysing Stuxnet more than six months after the initial samples were found and we don't yet have a complete picture of the malware. Given the flood of malicious files we receive, we are rarely, if ever, able to spend such amounts of time on any specific malware or malware family.

We have a Catch-22 situation. If we don't have the time to analyse the malware and understand that we are actually working with a limited set of malware families then we are dealing with a virtually infinite amount. If we take the time to understand each malware family, then proper detection for the family will take significantly longer than our customers will accept. In the end it is all about doing it fast or doing it well. You can rarely do both.

Editor: Helen Martin
Technical Editor: Morton Swimmer
Test Team Director: John Hawes
Anti-Spam Test Director: Martijn Grooten
Security Test Engineer: Simon Bates
Sales Executive: Allison Sketchley
Web Developer: Paul Hettler
Consulting Editors:
Nick FitzGerald, *Independent consultant, NZ*
Ian Whalley, *IBM Research, USA*
Richard Ford, *Florida Institute of Technology, USA*