Virus Writer Liability
infectionvectors.com
June 2004

Overview

Most readers will likely agree that viruses cause work disruptions, data loss, and resource use that cost businesses money. If the attack costs money, there are real damages that could be cited in a court case. But when an automated virus (worm) breaks into a number of machines and wreaks havoc, what price should the virus author pay? With the growing number of arrests this year, that question will probably be asked quite a bit. This article looks at a few viruses and alleged authors, and asks whether there is any liability for virus writers beyond the code they compile and actively distribute?

Arrests in 2004

German authorities scored two high profile arrests this year when they arrested both the Netsky/Sasser author (at least one of them, with more to follow) and the Agobot code developer. Civil cases are pending against the Sasser writer, although it is unclear how many are going to be heard. These cases would appear to at least have a direct path to the author: it is alleged that he wrote the code, knew what it would do, and then released it in such a way as to compromise as many machines as possible. In the case of the Agobot coder, this path is a little more crooked: the claim is that he wrote the program, knew what it was capable of, and then released the source code to the world. If there is no evidence that either of these men actually distributed the compiled virus with the intention of compromising boxes, has any crime been committed? This question is, of course, answered differently based on the laws of the jurisdiction where the crime took place (which opens a different discussion completely).

Consider Wang Ping-an, a 30 year-old man in Taiwan. He was arrested at the end of May for allegedly writing the Trojan known as Peep. This particular program plagued a number of Taiwanese and Korean machines in May and June of 2004. However, reports do not indicate that Ping-an is charged with actually using the Trojan, just writing and releasing it to a hacker website.

Last year, when the University of Calgary decided to offer a virus-writing course, the reaction from the security community was swift. Arguments included the notion that teaching people to write viruses will lead to accidents, not to mention provide skills to some nefarious individuals. If someone is liable for what others do with source code/applications (Agobot and Peep), wouldn't U of Calgary also be responsible for outbreaks linked to their course?

Use at Your Own Risk

Agobot is published under the GPL. GPL, based on its contents, relinquishes producers from liability. It has never been challenged and tested in court, however, for anything such as malicious software. The code is clearly created for a single purpose though, compromising boxes (or does it "test" vulnerabilities and act as a scanner/remote control tool?). Agobot contains additional license clarifications, known as the "Agobot Private License" and a separate disclaimer:

```
"Attention Users: This product was meant for TESTING & EDUCATIONAL
purposes only. I am not held responsible for any misuse of this
product. You are held responsible for EVERYTHING & ANYTHING you do with
this product. Not intended For Kids Under The Age Of 13. Enjoy &
Remember Im Not Responsible For What You Do!"
```

One strain of Agobot, known as Nortonbot also carries warnings against disassembly and malicious use:

"***ATTENTION*** NortonBot is protected under international copyright laws. Any attempt to dissassemble or alter this file is a violation of international copyright law. NortonBot is NOT intended to be a virus or trojan."

Does either of these disclaimers hold up? It would be a very interesting legal test, by any set of cyber laws. Virus writers may use a number of defenses if caught, including this type of reference to the programs intent. There will undoubtedly be a number of attempts to cite a worm as a "vulnerability assessment tool" that uses the actual exploit to determine compliance with the fix. This is consistent with some vulnerability scanners, such as Nessus, that come equipped with some plugins that actually attempt to use the exploit code to determine patch levels (as is found with some of the DoS checks). In addition, there could be a claim that bots such as Agobot are only remote management tools (combined with security scanners). Programs such as DameWare's remote manager offer remote control to client systems, why not these freely available "tools?"

The idea of a "good" virus is eschewed by most security professionals, however, would a jury find that the author of Welchia/Nachi is deserving of a sentence equal to the Blaster writer? If not, then the intent of the writer must be weighed in the equation. It is an increasingly difficult calculus to get a hold on, one that must be worked out sooner than later by the courts.

Examples specific to teaching virus writing include the "ELF Virus Writing HOWTO" which has the following in its first page:
"No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, the author does not take any responsibility."

MyDoom.C carried the source code for .A with it and dropped it onto every infected PC. Was this an attempt to widely distribute the code so that the author would have plausible deniability if/when the most damaging evidence was found on machines he owned?

Variant/Variable Sentencing?

In 2003, the plight of teekid was introduced to the world. Minnesota teenager Jeffrey Lee Parson, aka teekid, was arrested for allegedly making changes to the original Blaster code and then releasing it into the wild. Does making a minor change to a virus and releasing it garner the same sentence as does writing the initial worm? The answer often comes down to how much money a worm costs in cleanup, which is the only measure of damages when it comes to court cases. The difficulty arises in determining how much any particular variant costs a business/country. Names for viruses are still not universally agreed upon. Furthermore, when a variant is released, it is often using the same infection mechanisms as its parent, making it likely that a company that was hit by the variant was also hit by the original.

This author would posit that the damage is done via the release of the virus, not in the writing. The implications of this are that classes on virus writing are not a bad thing in themselves, nor is writing potentially catastrophic worms. Making a change to a virus is not punishable, however, distributing it is. For cases such as Blaster, releasing the virus to propagate carries with it the intent to infect as many computers as possible. Variant writers are just as responsible, provided the variant is released into the wild.

References

Wang Ping-an
http://www.sophos.com/virusinfo/articles/taiwanarrest.html

Peep in Korea
http://www.sophos.com/virusinfo/articles/koreanpeep.html

TruSecure against U of Calgary class
http://www.trusecure.com/company/press/release467.shtml

Arrested Sven
http://msnbc.msn.com/id/4946173/

MyDoom.C spreading source
http://www.techweb.com/wire/story/TWB20040210S0015

ELF Virus Writing HOWTO
http://www.lwfug.org/~abartoli/virus-writing-HOWTO/_html/intro.html.

Teekid
http://www.internet-security.ca/internet-security-news-003/security-experts-expressing-caution-over-fbi.html