

## **One's Complement: On "Professional Malware"**

**infectionvectors.com**

**June 2005**

### **Overview<sup>0</sup>**

The definition of malware (and related terms) has been a problem for the anti-virus research industry for years. With the increasing use of "professional virus" and "professional virus writer," the problem has the potential to grow; now incorporating what a "professional virus" means to the community as a whole and how both the media and law enforcement<sup>1</sup> interprets this issue. This report examines whether and how the term "professional" can be applied to malware and malware authors.

### **The Community Center**

Malware's opponent is the anti-virus, anti-spyware, anti-phishing, and law enforcement communities, generically the anti-malware forces of the world. Often, the workers and organizations that build these communities are profit-motivated, something that is both a strength and weakness in many cases. Malware can be just as profitable as "legitimate" software, and is managed just as well in some cases. This in turn can be a strength and weakness for the criminal. What the anti-malware world witnessed in many cases was the unorganized, unprofitable nature of virus writers – mistakes and carelessness that lead to some arrests and the attraction of those willing to risk being caught for nothing more than bragging rights. Whether that is turning around in a global sense is for another paper, this report looks at the division of malware as professional or non-professional and how that may affect virus research.

Certainly the idea of a "professional criminal" has been around for some time, even applied to computer crimes<sup>2</sup>. The professional criminal derives his/her income from illegal activities, generally something in which they have particular skill. That is not unlike what one would imagine for the professional virus writer. The notion of a professional malware author, however, has developed over the last few years. The idea of a professional category for malware is something that has yet to be formalized, but could be important to law enforcement agencies hoping to deal with these criminals effectively.

### **Not So Bad**

Defining what is and what is not a virus is quite a challenge, one that has yet to be conquered<sup>3</sup>. Although the idea of "spyware" and what has been termed "grayware"<sup>4</sup>

muddies the waters a bit more, there has always been a concern over what precisely identifies an application as nefarious.

Could the division of all software as evil or not-evil be a false dichotomy? If there is malware, then “beneware” (beneficial software, how about just “anti-malware”)<sup>5</sup> may need a category of its own. That produces at least three categories: malware, anti-malware, and innocuous software. These groups can be divided further, as has been the case with adware vs. spyware, however, the exercise of this infinite regression is avoided by this paper.

Taxonomies are difficult even after broad categories are constructed. Although there would be virtually no one arguing against placing Blaster in the “bad” group and F-PROT in the “good” group (whatever terms are actually used), there is certainly room for debate with regards to well-known, seemingly innocuous software. Many people would argue that Easter eggs in software place the proverbial checkmark next to tests such as that for a Trojan: “Does this software have hidden routines that act in a way a user does not know about?” Given the nebulous nature of many software functions and the opinions of each person attempting to organize code, there is no doubt that a formal, conclusive definition of malware is still some time in coming. In either case, the idea of dividing software into more groups would not improve the debate over what code is malicious. If instead we attempt to add dimensions to our understanding of malware, then there may be a way to improve the discourse for researchers.<sup>6</sup>

One area that may be of interest to malware researchers is the distinction of professional malware. There are numerous stories in the media over a wave of for-profit virus-writing outfits. Whether or not the virus writing community as a whole is shifting towards professionalism is not for this report, the next section will look at how we may easily classify a virus as professional or amateur<sup>7</sup>.

### **Not So Good**

If the software itself is difficult to define as malware/non-malware, then it stands to reason that the authors of that software are equally difficult to place. That is probably true; however, their wares are less difficult to place. The professional virus writer has been around for years in the virus research literature. Recently the professional malware coder idea has appeared in popular media with greater regularity. The discussion here is not intended to be about professional people that write viruses (that is, a person that writes a virus and also has an unrelated vocation, or even somewhat related job as a coder)<sup>8</sup>, but those people that make money directly as a result of their malware production and/or distribution.

What is meant by a professional piece of malware? Generically, it is a piece of malicious code that creates a direct profit-making infrastructure for its controller/author. More specifically, professional malware works for its author; it generates a product, one that requires a crime to be committed to obtain or to liquidate. That does presuppose one can identify a program as malicious, possibly taking us back to the problem in the previous

section. However, in the overwhelming majority of cases, the use of a custom application to commit a crime greatly helps to define that application as malicious or not.

Taking this idea further, to be a “professional” piece of malware, there needs to be a level of manageability to the endeavor; there needs to be a “maturity” (to borrow a term from the SEI’s CMM) to the product and the production. Whether the creation and distribution of the code is profitable (that is, the revenue is greater than the investment) is immaterial to the effort’s professionalism, and would be impossible to measure accurately considering the nature of the coder’s business. Simply putting money in an author’s pocket is not enough to be professional, although it certainly makes the code a revenue generator (and possibly even profitable).

A well-developed rootkit or backdoor is not necessarily a professional revenue producer. To meet the standard proposed here, there needs to be some manageability to the application, in much the same way engineers talk about “enterprise level” software in the systems administration world.

MyDoom, one of the most successful mass mailers of all time, was a rampant problem in 2004, especially its early incarnations. Although the worm did open a backdoor, there was no central management solution, making the use of the compromised machines for a growing enterprise difficult. The addition of a well-defined, self-reporting function (the SdBot code which allowed for registration and control via IRC) that made MyDoom, however, could be considered a move that took a successful worm to professional status. The MyDoom iterations form a very scalable, manageable set of clients. In its own right, MyDoom’s release patterns have the characteristics of a strategic effort, not one simply designed to wreak havoc (except for its enemies, the malware researchers<sup>9</sup> of the world).

Beagle is often reported as one of the professional worms, and with good reason. This malware shows some of the most professional practices in software testing and distribution of any non-commercial product (and many commercial packages). Its ability to also generate income for its author places it prominently in the scope of professional malware products. Some of the functions of the worm that help define it this way: reporting function to organize infected hosts, self-updating routine allowing new versions of code on compromised clients, and the specialized profit-generators that lift email accounts, banking information, passwords, etc. from Beagle victims.

On the other end of the spectrum are worms designed without profit-generation, without production in mind. SQL Slammer (Sapphire)<sup>10</sup> was one of the most disruptive worms in Internet history. It can, in that way, be considered very successful as far as malware goes. It did not attempt to generate revenue, however, and could not be considered professional. In the same way, MSBlaster, another successful worm, would not be placed in the professional category.

As mentioned above, there are at least two pieces of the professional puzzle, one being revenue/product creation, the other being a set of professional practices employed by the code’s creators. Profit motivation alone, as seen in the case of many backdoor

applications, is not in itself an indicator of professionalism. These types of malicious applications often do not employ effective (or well-tested) mechanisms for managing the products, do not define their product well, and suffer from poor rollout strategies. One specific example, Webber/Berbew is a piece of malware that installs a backdoor and attempts to steal passwords (even attempting to send these possibly valuable items to an external host). This application, however, also attempts to delete directories with the string "system" in the name, destroying the client in many cases, and thereby destroying the malware's ability to produce.

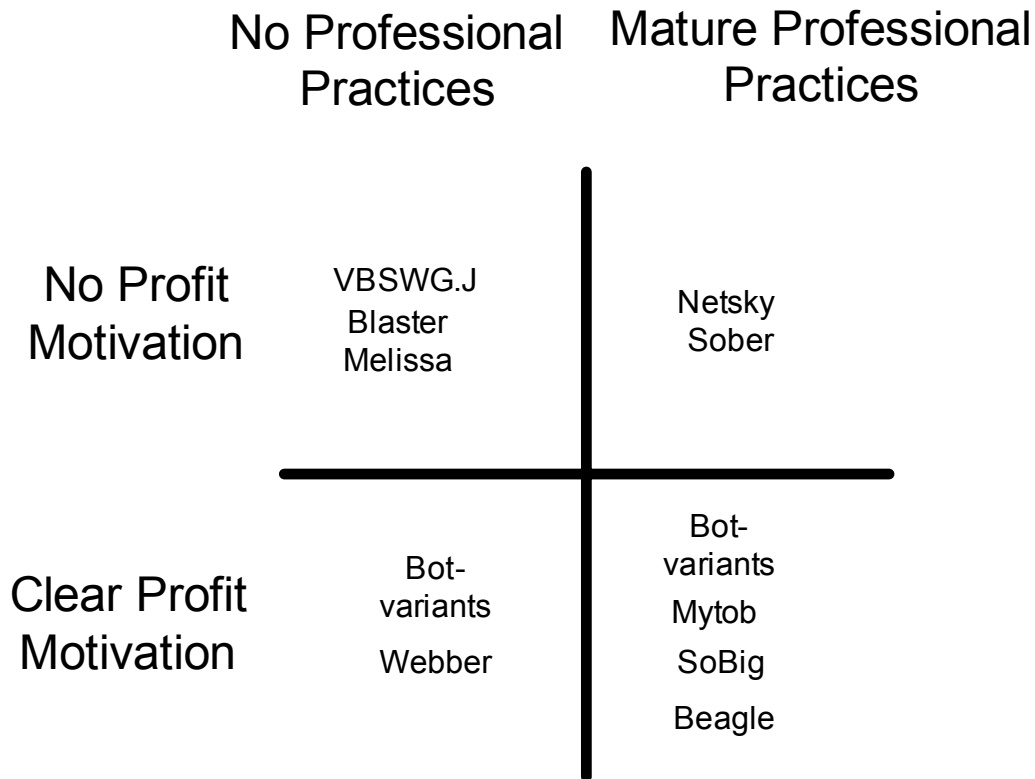
On the other end of this spectrum are well-crafted, mature pieces of malware that have no apparent profit motive. The Sober variants are prime examples of this. Although some analysts classify the later variants as sending spam, the worm does not establish a profit-motivation, nothing is harvested from the machine; Sober appears to be simply interested in propagation. This worm is one of the most successful mass mailers because of well-defined routines and sound release tactics.

Many very similar pieces of code may be categorized in different groups. Good examples of this are the multitude of bot variants released each day, some of which are part of very successful strategies to lease zombie-nets of victim boxes; some are clearly the work of hobbyists who have acquired source code and want to build their own malware, even though their work may have routines to steal information that conceivably could be sold.

The limited number of examples presented does point to a few delimiters of professional practices, which, admittedly, is a fairly nebulous concept at first glance.<sup>11</sup>

- Well-defined scope (efficient routines that specify and generate a product)
- Preservation/extension of the client (to ensure/continue revenue creation)
- Function development (sound testing/release tactics – evident planning & change management)
- Manageable production (clients and/or lifted data is organized)
- Repeatable success (no defined level of success, only learning from past lessons)

With this set of examples as a guide, some well-known malware can be divided into groups in the following graphic, showing profit-motivation and mature practices:



One means of refining the above that was considered was to add a scale to each axis, making it a graph of how well developed each trait was. This type of placement would be possibly the most subjective addition one could make, and the ensuing debate would surely overwhelm the issue of whether malware can be organized as professional or not.

A possible third dimension to this table would be “success” (and its cousin distribution of malware). Success, however, is not easy to build into the equation, as it has no predetermined meaning that cuts across all existing categories of malware. For example, Blaster and Slammer can be called successes (as can Beagle) because of the number of infections reported. However, a very profitable Trojan may be directed to only a few devices, making the amount of money it generates important in success.

A very telling “Z axis” for the graphic would be the “return on investment” (ROI) of the malware. That could consider the number of infections as a function of the total number attempts, and then the cost of each attempt. Self-propagating code would have the advantage in sheer volume; however, anti-virus companies notice worms much faster than a Trojan directed at a few selected machines (which may never be detected, begging the question as to how it would ever be included in such a scheme). In any case, the ROI for a specific piece of malware can only be estimated until a virus writer that keeps very meticulous records is arrested.

The mutation of malware, taking it from one category to another can be influenced by the author or by another coder. One case of this would be a “pirated” infection, as in the case of bot variants that use and then close the backdoors left by MyDoom. An author could just as easily change the profitability of any worm or also begin to adopt mature coding practices.

### **Complementary**

If malicious code can be described, even catalogued as a “professional” creation, what impact will that have on virus research? The simple idea that any viruses classified as “professional” should be watched more closely and given a higher priority is not far off. A professional malware author is much more beholden to his/her craft than an amateur. From the examples that have presented themselves up to this point (Mytob, Beagle, etc.) we have seen a very motivated group of authors distribute malware with worldwide reach and in relentless iterations.

Professional virus writers are in some ways handicapped by the functionality built into their code. Someone profiting from stolen goods has to endure more exposure to the public world in two ways: they have to somehow market/cash in on their wares, and they often have to expand the number of people involved in a crime (the “consumers” at the very least). This in turn increases the chances that law enforcement will cross paths with the coder, who will be facing penalties for “real” crimes that other (non-profit-minded) authors will not. Professional virus writers must establish the same types of financial distribution channels/infrastructure that more traditional criminals must. This is where law enforcement already has a strong foothold.

The above has two potential impacts for law enforcement. The first is that these criminals may receive much more attention as they are committing crimes that fall outside of the often-fuzzy cyber-law. Given that they receive the scrutiny of police forces, those forces will need the help of anti-virus researchers and their records, as will prosecutors. Second, the percentage of overall attacks initiated by the professional malware author has yet to be determined. It may be found that a small portion of malware coders is responsible for the majority of attacks.<sup>12</sup>

The inclusion of “professionalism” is not meant to muddy the waters of malware research, but to foster additional dialogue that may help align research with the goals of law enforcement. If the shift to professionalism in virus writing is picking up momentum, and there are signs that it is, then the researchers that document, catalog, and reverse engineer the products of this shift will be some of the greatest assets of the law enforcement community. The ability to recognize and label code as not only malicious, but also of a professional nature will be an important step to that end.

## Notes/References

0. A quick note on "One's Complement:" Very briefly for anyone not familiar (and if this is the case I recommend finding a more detailed account of the method), computer processors need to have a means of identifying negative numbers. To do this, the "sign" of the number is built into the binary representation. There are a few mechanisms for doing this: establishing a "sign bit," one's complement, and the system used by virtually all computer systems today: two's complement. In one's complement, the processor takes the bitwise complement of each "0" and "1" to produce the original number's opposite/negative. For example, the number 5, represented as a byte would be 0000 0101, -5, using one's complement would be 1111 1010.

Modern machines do not use this scheme as it allows for two different representations of zero. As there are only 2 states to each bit for a machine, "0" and "1" trying to define at least 3 states of number (positive, negative, zero) is a somewhat artificial exercise. There is a similar problem with malware taxonomy; positive software, software with a negative impact on the Internet, and innocuous software are difficult to define given the structure we are forced to use to define each type of code. The answer to this problem, of course, for a processor-based system is the two's complement notation, in the malware world it has been to attempt to classify everything as only "negative" or innocuous. The use of the term here is simply to imply that researchers need to break free of the existing schema and add the notion of "professional" to the proverbial mix.

1. The subject of law enforcement is far from the author's area of expertise and is used here as the logical audience of any information detailing how a crime is committed. The reader is encouraged to delve deeper into the issue for information on cyber crime/forensics.

2. The professional malware author has been mentioned in many sources, a few are provided here for reference:

As far back as 2002, Kaspersky Lab used the term. See, "History of Malware – 2002" Viruslist.com  
<http://www.viruslist.com/en/viruses/encyclopedia?chapter=153311186>

"Who Writes Malicious Programs and Why?" Viruslist.com  
<http://www.viruslist.com/en/viruses/encyclopedia?chapter=153280553>

"F-Secure Corporation Data Security Summary for 2004: The year of phishing, professional virus-writing, and arrests"  
<http://www.f-secure.com/2004/>

Furthermore, some analysts noted MyDoom as the introduction of the "professional" era in virus release: "MyDoom turns 1, Impact Grows." Gregg Keizer, TechWeb News. 28 January 2005.  
<http://informationweek.com/story/showArticle.jhtml?articleID=59100629>

And others noted that 2005 would be known as "the year of the professional virus."  
"Year in review: Seeking to squelch spam" CNET News.com. Stefanie Olsen and Rob Lemos, December 2004.  
[http://news.com.com/Year+in+review+Seeking+to+squelch+spam/2009-1024\\_3-5498173.html](http://news.com.com/Year+in+review+Seeking+to+squelch+spam/2009-1024_3-5498173.html)

For historical reference, the idea that a virus writer would be professional had not been introduced to the mainstream public until recently. This article from 2003 details an interview with one of the greatest virus writer researchers, Sarah Gordon. Her pioneering work on the profile of a virus coder is well worth investigating for anyone unfamiliar with her research.

"Who creates viruses?" Timofey Saytarly, 30 May 2003. Computer Crime Research Center.  
<http://www.crime-research.org/news/2003/05/Mess3004.html>

3. Recently the US Senate had trouble taking action on spyware issues because of confusion over the software's definition.

"Senate panel mulls action on spyware" Consumer Security – MSNBC.com. Bob Sullivan, 11 May 2005.  
<http://www.msnbc.msn.com/id/7818285/>

4. Grayware Page at Trend Micro, for example:

<http://www.trendmicro.com/vinfo/grayware/default.asp>

5. "Beneware" started off as shorthand when jotting down ideas for this report. The term, although more of a joke than anything, has remained as it seems to capture both the idea of a piece of software that acts on behalf of its host, but also the silly nature of such a notion.

The idea of defining "beneware" is a fruitless task. If it has been impossible to sort programs into two categories, then sorting them into three would likely only boost the frustration of researchers. What it does introduce, however, is the possibility of an anti-virus program that runs a true "white list" for executables, much the same way application proxies work. This type of technology already exists in many applications, from sentinels that protect servers/firewalls from rogue applications to white lists for many antivirus products. The list would not prevent the application from being scanned; simply allow it to run if it otherwise "checks out."

6. What I mean is that instead of extending a group of tags we may not be happy with, maybe the best way to open up the debate is by working on whole new layers to the taxonomy – instead of bad versus good, we'll adopt that and also discuss professional versus non-professional. That leads into the section of the paper following this note that attempts to outline how "professional" malware (presupposing we are not interested in professional software that is not considered malicious) may be defined.

7. If you are a malicious coder, prior to taking offense at "amateur" please note that I mean it simply as the opposite of something written to generate profit, not as an indication of how well the code was written.

8. Nor is it about virus writers that turn to more legitimate vocations:

"Sasser author gets IT security job." The Register, John Leyden, 20 September 2004.  
<http://www.viruslist.com/en/viruses/encyclopedia?chapter=153280553>

9. If malware is hard to define, how exactly do you know you are a malware researcher? Although this was intended as a joke, we could say that at the lowest level, a malware researcher is studying a broad range of code in search of what is malicious.

10. Now, it's certainly possible that the author of Slammer/Sapphire was in fact paid by someone to release a worm that would disrupt the normal operation of Internet-connected systems. However, that is a motivation that could never be measured. Furthermore, in those cases, the worm does not generate profit; it is the product in and of itself.

11. For some time, I liked the idea of "dangerous code" versus innocuous code as a means for dividing software. The intriguing part of this taxonomy was that code wouldn't necessarily have to be "harmful" to a machine to be dangerous. So, publishing the code for a virus with one JMP missing (so that the code doesn't hurt anything when run, as it won't function) is still dangerous because the author would have given the blueprints for a virus to the world at large. Unfortunately, virus scanners themselves would fall into this category, because with slight adjustment, a program that opens every file on a machine and runs with low-level system privileges could be pretty dangerous.

12. Although, in the banking fraud game, professionals are responsible for the bulk of the crimes, making their capture very important.

Clint Swift and Karen Epper Hoffman. "Fraud Looms" BAI Banking Strategies, Jul/August 2004.  
<http://www.bai.org/bankingstrategies/2004-jul-aug/fraud/print.asp>

Additional Information on Releases Mentioned in the Paper

Agobot information:

<http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=37776>

Berbew/Webber information:

<http://securityresponse.symantec.com/avcenter/venc/data/backdoor.berbew.q.html>

The latest release of Sober, not catalogued as a Sober release at Symantec's Security Response site incidentally), is a possible exception, if the email it sends out were directed by a customer. In this case, the worm may have some profit capability. Other variants, like Sober.O can be researched:

<http://securityresponse.symantec.com/avcenter/venc/data/w32.sober.o@mm.html>