The geneology of malware

Fernando de la Cuadra, international technical editor, Panda Software

Fernando de la Cuadra

A long time ago, viruses were very simple. Today's features, such as hiding the virus or giving it a very fast propagation system, were not considered necessary. In the early days viruses were spread by transferring a floppy disc from one computer to another, and at that time virus creators were really experts, creating incredible infection systems in just a bunch of kilobytes.

Today, being a virus creator is more about being a good internet searcher. The code for much malware is posted in websites, and it's not too difficult to find it. This has meant that much of today's malware is a mixture of source code from different authors. The first version of a new virus may be more or less well programmed, but as the code goes from one developer to another and new variants emerge, it becomes more and more complex. In some cases, it will carry useless parts and duplicated modules.

The practice of malware writing

Ideally, new variants of original malware should be clearly understood by the new coder, but in practice this does not always happen. Much modern malware is adapted by script kiddies who make a habit of tweaking original virus codes rather than inventing their own designs. In many cases they do not have enough knowledge to

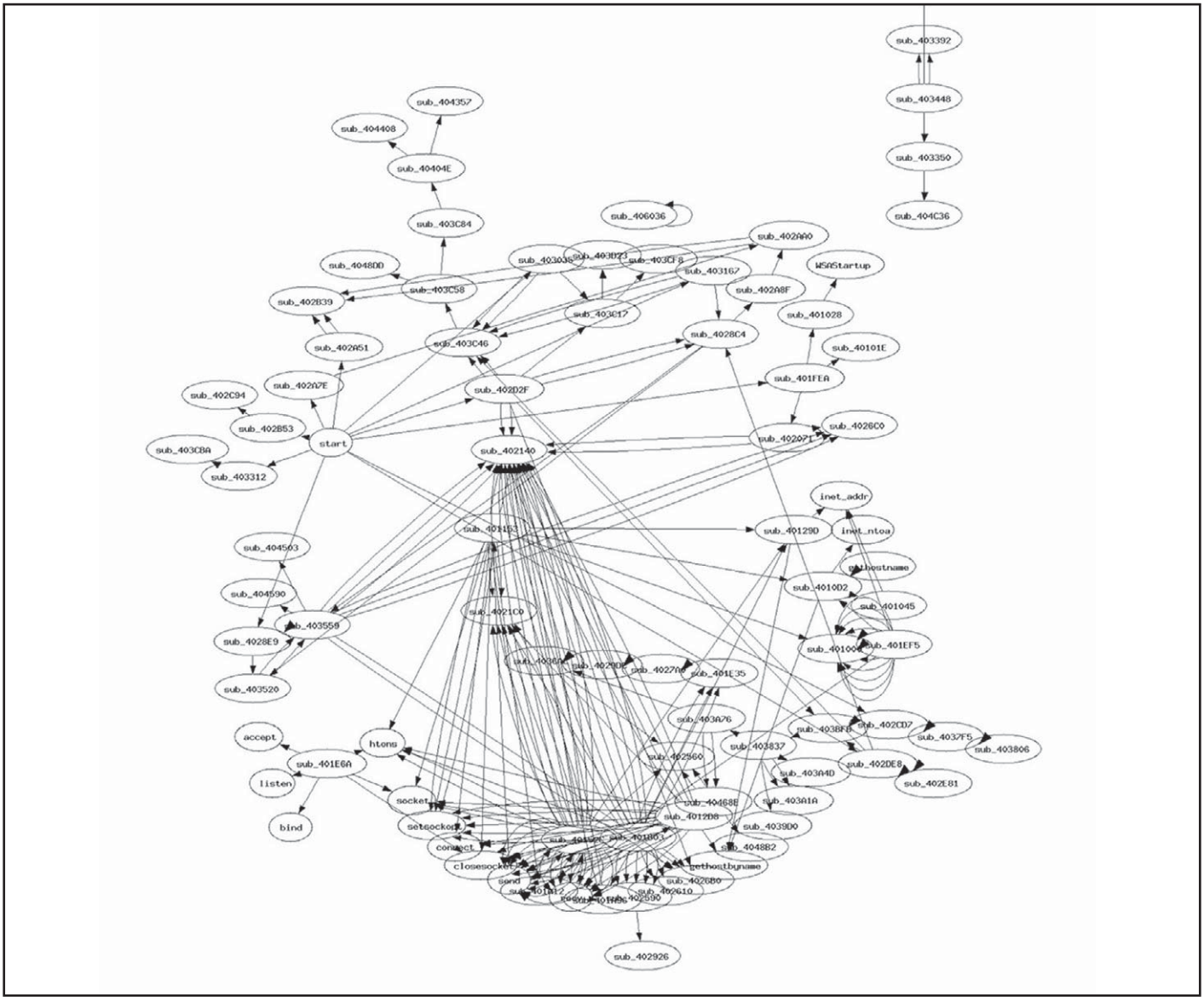

Figure 1: A graph of the Sasser.. A Malware's structure

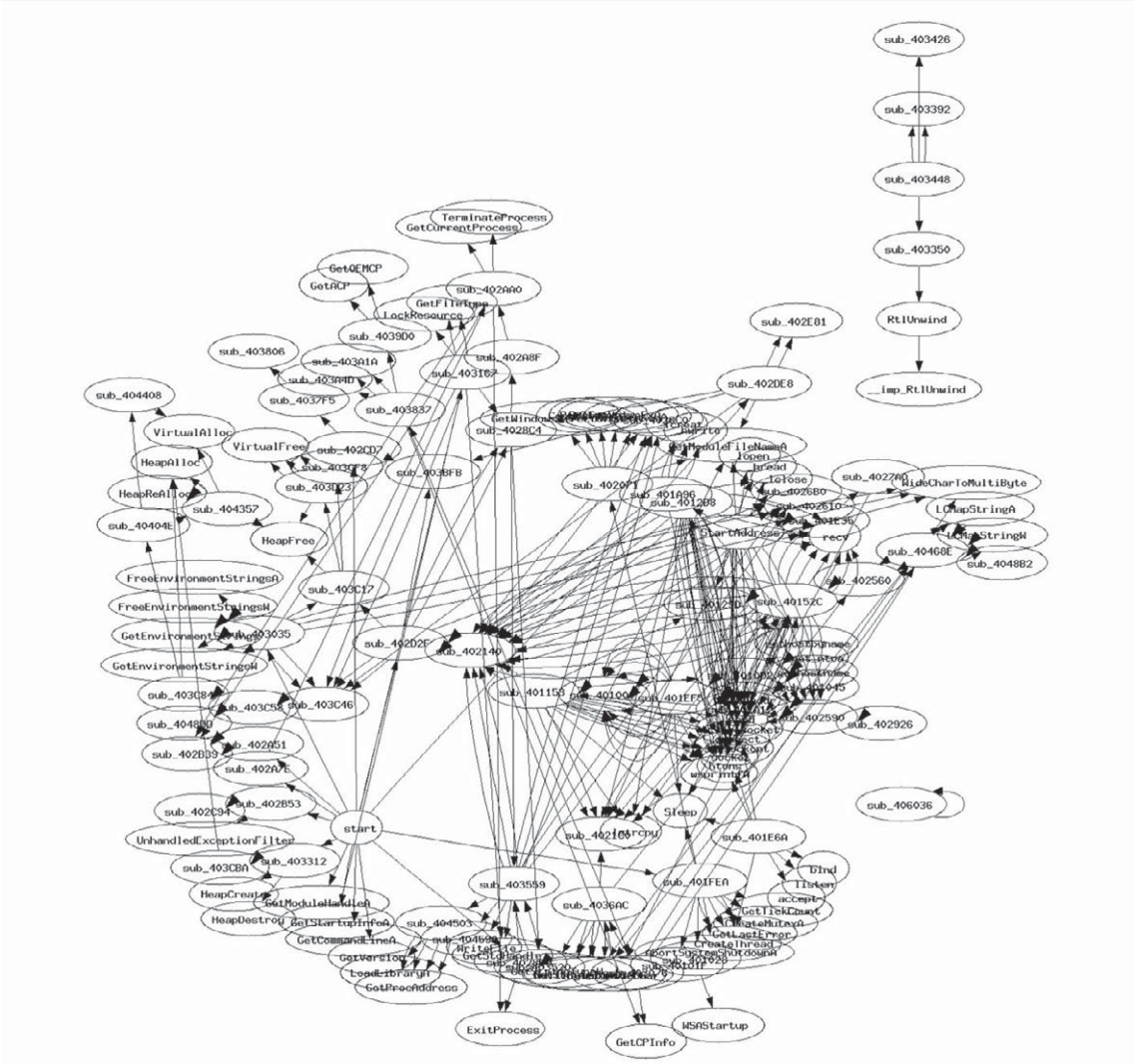

Figure 2: Sasser. F

understand the program's structure, and they create new sections of code that simply duplicate existing functions. They may then fail to remove parts that go unused.

Although stemming from a lack of expertise, this adaptation sometimes means that new malware variants may not be detected by antivirus tools. By removing or adding a part, hobbyist malware 'developers' may inadvertently change the code enough to require a whole new virus recognition signature.

The figures in this article contain some graphic examples. The graphs are representations of the relationships between modules in the code. An oval shape contains the names of each module, and the structure is represented by lines joining each oval. The least number of lines indicates the least complex code.

> "Much modern malware is adapted by script kiddies who make a habit of tweaking original virus codes rather than inventing their own designs"

Take a look at the Sasser.A graph in figure 1. This is the original code for the Sasser worm created on May 1, 2004. It contains approximately 100 modules, connected by a reasonable number of lines. The relationships representing the structure are relatively clear and simple, and yet as we know from the time of its release, it was very effective. The author clearly understood what they were doing.

After this, other versions appeared. On May 11, Sasser.F surfaced, with a very different graph. It now has over 200 different modules, and the relationships between them are extremely complex (see figure 2).

Throwing bad code after good

We don't know how much time the author of the original version needed to create the code, but, during the course of the next 10 days the author of Sasser.F made plenty of changes. The differences between both versions are important, but the effects and the way it replicates and carries out actions are very similar. Sasser.A had a size of 15,872 bytes. Sasser.F was 74,752 in size, but once decompressed this grew to 132,155 bytes!

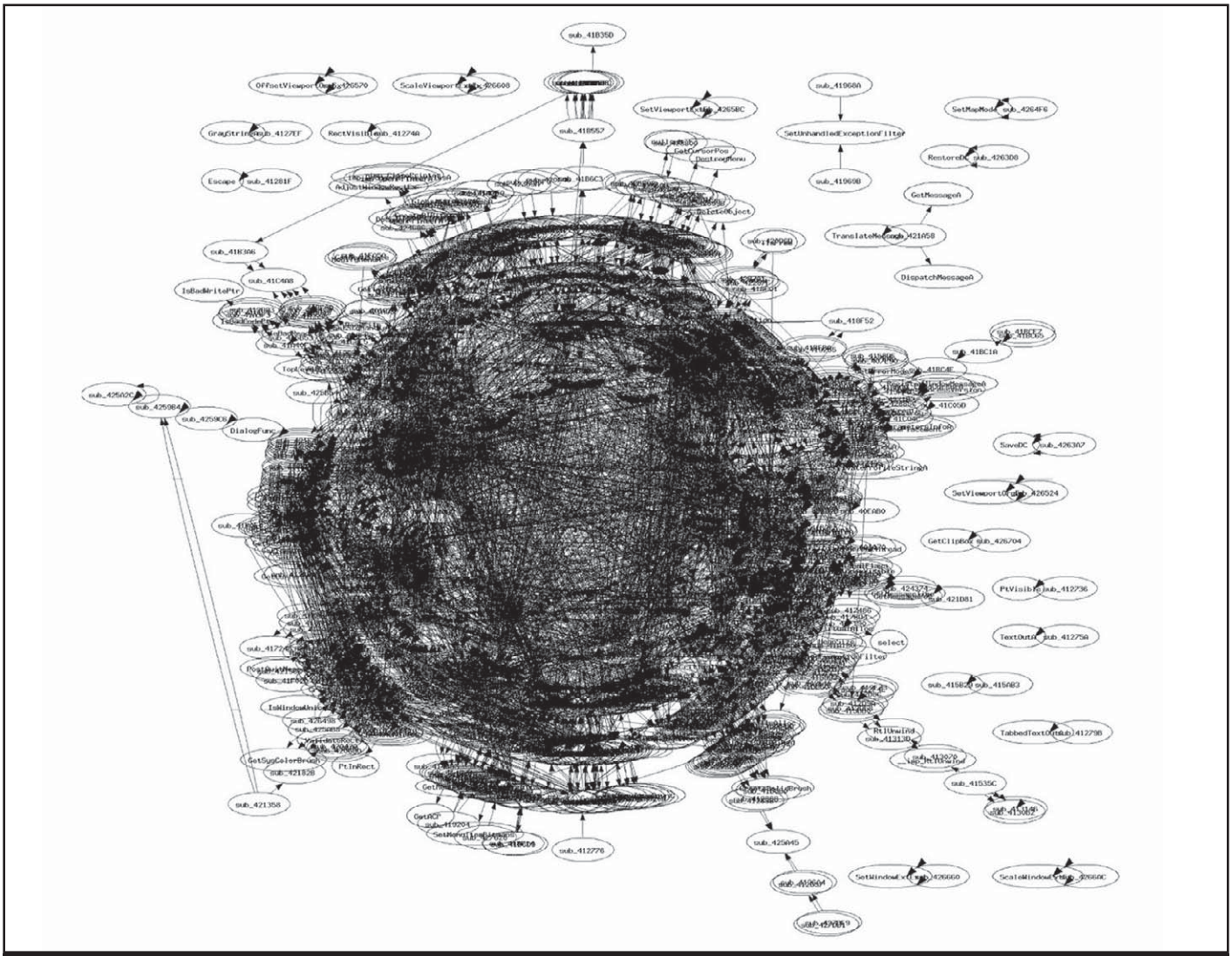

Figure 3: Gaobot. A

The debugging process was clearly of poor quality. There were other priorities for the creators. The Sasser coders were involved in a code war with other virus creators, the Netsky hackers. So, they needed a winning code to spread on the internet rather than a program following best practices in software development. The emphasis would have been on speed, rather than elegance.

The other interesting point is not just the complexity of a certain variant of the code, but the additions that make subsequent versions look altogether different. The bases are the same but the script kiddies change it, adding so many modules in addition to the original so that it looks like a new piece of malware.

For example, take a look at the original Gaobot code in figure 3, corresponding to the first version named "A".

> "The other interesting point is not just the complexity of a certain variant of the code, but the additions that make subsequent versions look altogether different."

Waiting for Gaobot

As the graph shows, it's more complex than the previous Sasser code. And this was just the first! Many versions later the version OKO appeared. Bear in mind that versions are named with letters running from A to Z, then AA, AB, through to AZ. Then from BA through to BZ and so on, until we reach ZZ. Then, naming starts with three letters, running from AAA to ZZZ. So, the OKO was over 15,000 versions later.

Gaobot.OKO works curiously. The original code with the modifications was becoming more and more complex, so that it had become almost a black spot in the graph, due the large number of modules and the interactions between them. But close by is another area with a clearer graph, which indicates that somebody decided to trash the original code and then start with a clearer and more effective graph. But the original part remains there! There now appears to be a XVII

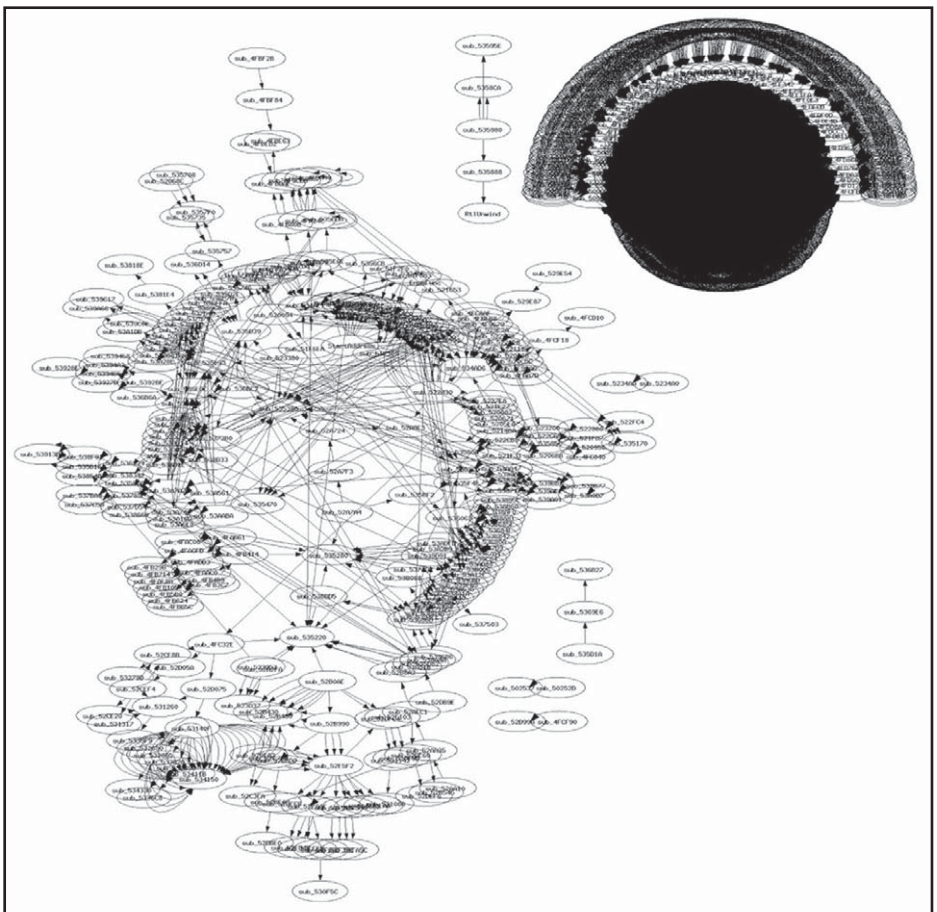

Figure 4:Gaobot. OKO

century lady (with the then typical hair style) looking at the graph!

Malware writers: more common, less talented

As we can see, the virus scene is getting worse. Not only is there more malware, causing silent epidemics (in 2006 there was more malware detected than in the previous 15 years), but the creators are not as good as they were during the early days of virus writing.

Nowadays, the situation can be summed up in three points. First, the hackers are less well trained because the codes are easily found on the Internet and programming knowledge need not be as sophisticated. Second, the number of malicious codes is growing quickly. Finally, with the exception of the occasional innovation such as the recent Storm.Peacomm trojan, the epidemics are not so notorious.

In the future we will see far more malware variants of poorer quality, causing silent epidemics. The solution is to not have a signature for each code, but to have solutions that can detect the codes as a result of their behaviour, as the final goal of each variant will be very similar to other previous versions. This should help avoid the constant game of whack-a-mole that chaotic and fragmented malware development practices have forced anti-virus companies to play.

About the author

Fernando de la Cuadra has worked at Panda Software since 1997, and is responsible for its global editorial content. His professional career began in IB2 consultores, providing training and support to key accounts and the entire sales area. He later moved to IBM's Helpware department, before working freelance on a wide variety of IT projects. He studied Pedagogy at the Faculty of Philosophy (Education Branch) in Madrid's Complutense University (Spain).

EVENTS CALENDAR

1 - 4 May
17th Conference on Computers, Freedom and Privacy
Location: Montreal, Canada
Website: www.cfp2007.org

14 - 15 May
Cyber Security and Information Infrastructure Research Workshop
Location: Oak Ridge, TN, USA
Website: www.ioc.ornl.gov/csiirw07/

20 - 23 May
2007 IEEE Symposium on Security and Privacy
Location: Oakland, CA, USA
Website: www.ieee-security.org/TC/SP-Index.html

20 - 24 May
Eurocrypt
Location: Barcelona, Spain
Website: www.iacr.org/conferences/eurocrypt2007/

21 - 25 May
Integrated Network Management 2007
Location: Munich, Germany
Website: www.ieee-im.org/2007/

3 - 6 June
Techno Security Conference 2007
Location: Myrtle Beach, South Carolina, USA
Website: www.techsec.com/html/Techno2007.html

5 - 8 June
ACNS 2007
Location: Zhuhai, China
Website: http://icsd.i2r.a-star.edu.sg/acns2007/