Global Information Assurance Certification Paper

Malicious Code: Evolution of a Virus

December 10, 2001

By

Vince E Fogle Jr.

Table of Contents

1. Introduction

Computer viruses have progressed from urban myth to annoyance to major threat; yet, even with all the damage that computer viruses have done. Viruses are computer programs that are designed to spread themselves from one file to another on a *single* computer.

What I will be discussing is the **Life Cycle of a Virus;** computer viruses have a life cycle that starts when they're created and ends when they're completely eradicated. To better understand this cycle a discussion on classification of worms, their evolution, future and containment is required.

2. Creation

Until a few years ago, creating a virus required knowledge of a computer programming language. Today anyone with even a little programming knowledge can create a virus. Usually, though, misguided individuals who wish to cause widespread, random damage to computers create viruses.

Worm Classifications Computer worms can be classified based on two characteristics; the transport mechanism used by the worm to send itself and how the worm is actually launched on a computer system.

2.1 Worm Transport Classifications

The following are known or potential worm transport schemes.

2.1.1 Email Worms

The *email worm* is one that uses email as its primary means of transport. Under this top-level classification, we can create two subcategories; n*ative email worms and parasitic email worms*

A *native email worm* is one that is built in the native scripting language of the host email system. Such a worm is carried in a proprietary form along with its associated email message, as opposed to being carried as a file attachment. The native email worm can only exist within the email platform, and is not viable outside of the host email system. To date, we have seen no native email worms.

A *parasitic email worm* is one, which leverages the transport capabilities of an email system to spread itself. The parasitic email worm, for example, may use the email program to send itself as an attachment to email. The parasitic email worm can exist outside of the email platform, and may actually use other techniques to spread itself. Of the recent worms, Melissa, ExploreZip and Happy99 would be considered parasitic email worms.

2.1.2 Protocol Worms

The *protocol worms* spread themselves using one or more non email based protocols, such as IRC's DCC protocol, the FTP protocol, or using simple TCP/IP sockets. The Internet Worm could be considered an arbitrary protocol worm since it used standard TCP/IP connections (from one UNIX box to another) to spread itself. The *peer-to-peer worm* is another example of an arbitrary protocol worm that spreads itself over peer-to-peer networks.

2.2 Worm Launch Classifications

These classifications are used to describe how the worm actually gains control of a computer system: does it require user interaction (and how much?), or can it spread unaided.

2.2.1 Self-launching Worms

Worms are capable of spreading to a new system and actively running on that system can be called *self-launching worms*. These worms do not require user interaction in order to gain control of a system; instead, they exploit some aspect of the host (operating system, application system, email system) to cause their code to automatically execute upon introduction to a new system. The Internet Worm and the IRC worms (described in later sections) are examples of self-launching worms. A subset of this category is the *back door worm.* The back door worm is one that exploits a back door in a target system to gain entry *and* to ensure that it is launched, again without human intervention.

2.2.2 User-launched Worms

This category of worms requires user intervention in order to execute on a new system. For instance, the Melissa worm/virus is just such a worm. In order for Melissa to infect a system, an infected attachment must be manually opened/viewed by a user. The worm cannot cause itself to launch on a system without user intervention.

2.2.3 Hybrid-launch Worms

These worms are capable of spreading using both mechanisms. An example of a hybrid-launch worm is ExploreZip. This worm, when sent in email, required a user to launch the infected attachment to gain control of the system. On the other hand, once running on a computer system, ExploreZip would automatically spread itself to other computers over the peer-to-peer network. These targeted machines would then become infected on the next reboot (without any known user intervention).

2.3. The Future of Worms

While the majority of computer malware consists merely of knockoffs of older malware strains, there are a core group of virus writers that consider themselves “trail blazers.” This group has brought us threats like Boza, the first 32-bit Windows virus, Strange Brew, the first Java virus, and countless others. It is inevitable that over the next few years we will see viruses and worms attacking and leveraging many heretofore untargeted platforms, as varied as Palm Pilots, Lotus Notes, and personal web servers.

2.3.1 Cable/DSL Brings Worms To The Home

As more users adopt broadband technologies in the home, we expect that the incidence of computer worms targeted at home users (and small businesses) will grow rapidly. Today, home Internet users are assigned dynamic IP addresses each time they login to the Internet. Given that users log in and out frequently, it becomes very difficult for a worm to find such a target machine and spread to it. As users migrate to broadband technology such as cable modems or Digital Subscriber Line (DSL), they will have constant, reasonably static connections to the Internet, making their computer a “sitting goose.” Computer hackers or roving worms will be able to easily cull Internet addresses and use them to attack these machines.

The personal firewall (in conjunction with anti-virus software) will become a must-have application and help to stem at least some of the worms and viruses that will plague the growing number of connected desktops.

2.3.2 MAPI Worms

MAPI is an acronym that stands for Messaging Application Programming Interface. Many common email client applications, such as Outlook, Exchange, etc. support the MAPI system (which is provided via the COM mechanisms discussed in previous sections). This means that other applications, including worms and viruses, can leverage email functionality (sending email, receiving email, examining attachments, etc.) without having to understand the details of email systems or protocols. In fact, Office 97 macro viruses can access the MAPI system using the simple Visual Basic language, making it possible to construct a computer worm in less than 30 lines of programming statements! The Melissa and ExploreZip viruses used this facility to spread themselves, and we expect that this mechanism will be employed more than any other worm replication mechanism in the foreseeable future.

2.3.3 Information Stealers and Remote Control Worms

Over the last few years, we've seen a rash of new virus, worm and Trojan threats that are capable of exporting information from an infiltrated machine or allowing a remote attacker to take control of such a machine. While older malicious code threats would delete files or format hard drives, the new *payloads* of choice are information stealing and remote control since these payloads leverage the power of the Internet.

2.3.4 Peer-to-peer Worms

The Melissa virus/worm spread itself using MAPI email commands and achieved a huge penetration of company email systems, flooding virtually every emailbox in some corporations. However, it is unclear how many actual corporate desktop computers it was able to infiltrate. How many of the thousands of corporate users who received Melissa in email actually viewed the document attachment and launched Melissa?

Worms can spread to other peer-networked machines by using simple Windows programming techniques, so this is likely to be exploited far more in the future. This should serve as a wakeup call to administrators; peer-to-peer networks are large holes waiting to be exploited by computer worms.

2.3.5 Email Scripting Worms

We often tell end-users that they can't get a virus by simply opening an email. While this is true for most consumer email packages, there is a risk with corporate, group-ware email systems. Most of the groupware email programs allow the user to embed programmable scripts in messages. These scripts allow the user to create simple mail-based user interfaces, forms, etc.; unfortunately, on some platforms, they may also be leveraged to produce malicious or self-replicating code.

2.3.6 ActiveX and Java Worms

While worm authors may choose to build ActiveX-based worms, the likelihood of Java worms is extremely
small in the short to medium term.

ActiveX programs are basically fully functional Windows programs that are capable of performing any number of malicious actions (just like ExploreZip), and therefore should be considered a potential threat. ExploreZip, Happy99 and even Windows viruses could easily be built and deployed as ActiveX components instead of standard Windows programs. Java worms are unlikely for two reasons. First, Java security prevents

unsigned Java applets from spreading themselves or accessing the local computer resources. Second, while signed Java applets can access the host computer system and potentially spread themselves, the liability issues associated with digitally signed applets will likely limit the number of wild threats.

3. Replication

The process by which a virus makes copies of itself in order to carry out subsequent infections. Replication is one of major criteria separating viruses from other computer programs. If it infects your hard disk, it may well remain undetected for many months, infecting every floppy disk that you use. Viruses replicate by nature. A well-designed virus will replicate for a long time before it activates, which allows it plenty of time to spread.

4. Activation

Viruses that have damage routines will activate when certain conditions are met, for example, on a certain date or when the user takes a particular action. Viruses without damage routines don't activate, instead causing damage by stealing storage space.

5. Discovery

This phase doesn't always come after activation, but it usually does. When a virus is detected and isolated, it is sent to the International Computer Security Association in Washington, D.C., to be documented and distributed to antivirus developers. Discovery normally takes place at least a year before the virus might have become a threat to the computing community.

6. Containment

How can corporations protect themselves against computer worms? The following sections describe some of the defenses that corporations and governments can use to stem the threat of computer worms.

6.1 Run Anti-virus Software on Servers, Gateways, and Desktops

Enough said.

6.2 Remove "all company" Addresses From Your Lists

Computer users rarely have the need to send emails to the entire company and such a facility is extremely vulnerable to email-based computer worms. Email administrators should limit public email lists to small functional groups and eliminate all company-wide lists. Should users need to send such an email (this should be rare), they can forward the email to an administrator for company-wide posting.

6.3 Lock Down All Peer-to-peer Networking

Peer-to-peer networks are a huge security risk for network-aware worms and viruses. We recommend that administrators lock down peer-to-peer networked drives on all computers where this is not absolutely required. Administrators may also want to establish an official policy against peer-to-peer volumes and distribute this to users. At the very least, the administrator should maintain a special computer grouping or domain in the network management software (or anti-virus console) for all peer-to-peer networked computers. This will enable quick deployment of anti-virus definitions to these particularly vulnerable machines.

6.4 Deploy Internal Firewalls

Corporate firewalls are fairly effective at preventing both hacker and malware attacks from outside sources; however, they provide no benefit once a worm has entered the corporate network. As we have seen with ExploreZip, the vast majority of computers that were actually penetrated by ExploreZip were attacked from within the corporation, from other peer-to-peer networked computers. Deploying internal firewalls could prevent such intra-network infections. Administrators should consider deploying internal firewalls around corporate servers, such as:

1. File servers
2. Email servers
3. Corporate databases/SQL servers.

In addition, personal firewalls are effective at preventing attacks on desktop PCs running Windows 9X or Windows NT. While this may be a more expensive option, it could seriously neuter many back door worms and Trojan horses.

6.5 Disable Email Script Capabilities

If your group-ware product supports email scripting, this should be disabled for all but a few users (most likely those in the IT department). By disabling these facilities, you can protect your corporation from Native email threats.

7. Assimilation

At this point, antivirus developers modify their software so that it can detect the new virus. This can take anywhere from one day to six months, depending on the developer and the virus type.

8. Eradication

If enough users install up-to-date virus protection software, any virus can be wiped out. So far no viruses have disappeared completely, but some have long ceased to be a major threat.

9. Conclusion

Computer worms have grown to become the fastest spreading and most costly malicious code threats of this decade. While a virus might slowly spread from one corporate department to another, the computer worm can often blitzkrieg through an organization in hours or even minutes. This makes worms, especially with destructive payloads or data export capabilities, extremely ruthless attackers.

The malicious code problem will continue to grow as the Internet grows. The constantlyaccelerating
trends of interconnectedness, complexity, and extensibility make addressing the problem more urgent than ever. As extensible information systems become more ubiquitous, moving into everyday devices and playing key roles in life-critical systems, the level of the threat moves out of the technical world and into the real world. We *must* work on this problem. Our best hope in combating malicious code is creating sound policy about software behavior and enforcing that policy through the use of technology.

BIBLIOGRAPHY

1) ‘Mobile Code Threats, Fact or Fiction’, Carey Nachenberg. Proceedings of the 1999 ICSA IVPC Conference.

2) Lackey, R.D., "Penetration of Computer Systems, an Overview", Honeywell Computer Journal, Vol. 8, No. 2, 1974.

3) Saltzer, J. H. and M. D. Schroeder, "The Protection of Information in Computer Systems," Proceedings of the IEEE, Vol. 63, No. 9, September 1975, pp. 1278-1308.

4) Spafford, E. H., "The Internet Worm: Crisis and Aftermath," Communications of the ACM, Vol. 32, No. 6, June, 1989, pp. 678- 688.

Last Updated: July 23rd, 2017

Upcoming Training

SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor