Recent Worms: A Survey and Trends

Darrell M. Kienzle *
Symantec Corporation
12801 Worldgate Dr. Suite 800
Herndon, VA 20170
703-668-8872
Darrell_Kienzle@symantec.com

Matthew C. Elder
Network Associates
1145 Herndon Pkwy. Suite 500
Herndon, VA 20170
703-885-4814
Matthew_Elder@nai.com

ABSTRACT

In this paper, we present a broad overview of recent worm activity. Virus information repositories, such as the Network Associates' Virus Information Library, contain over 4500 different entries (through the first quarter of 2003). While many of these entries are interesting, a great number of them are now simply historical and a large percentage of them are completely derivative in nature. However, these virus information repositories are the best source of material on the *breadth* of malicious code, including worms.

This paper is meant to provide worm researchers with a high-level roadmap to the vast body of virus and worm information. After sifting through hundreds of entries, we present only those that we considered breakthrough or novel, primarily from a technical perspective. As a result, we found ourselves omitting some of the most notorious worms simply because they lacked any original aspects. It is our hope that others in the community who need to get up to speed in the worm literature can benefit from this survey. While this study does not contain any original research, it provides an overview of worms using a truly breadth-first approach, which has been lacking in the existing worm literature.

From this raw data, we have also extracted a number of broad quantitative and qualitative trends that we have found to be interesting. We believe that a workshop discussion of these, and other thoughts, will be engaging and informative.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *Invasive software (e.g., viruses, worms, Trojan horses).*

General Terms

Security.

Keywords

Malicious code, survey.

* *The author was with Network Associates while performing this work.*

WORM '03, October 27, 2003, Washington, DC, USA.
Copyright 2003 ACM 1-58113-785-0/03/0010…$5.00.

1. INTRODUCTION

In March 2001, c|net declared that 2001 would be “The Year of the Worm” [6]. They predicted that fast-moving, self-replicating code would become the weapon of choice for those wanting to inflict widespread damage on the Internet. As it turns out, 2001 saw a renaissance in worm creation. This culminated in the release of Nimda, an incredibly sophisticated worm that made headlines worldwide.

As part of a larger research project on detecting worm-like behavior, we conducted a study of recent worm activity. The goal of this study was to better understand recent trends in worm development and attempt to extrapolate future worm developments. In this paper, we present our findings about recent worms. We do not make any predictions about future worm developments, if for no other reason than we would rather not give anyone any ideas.

We found conducting this exercise to be a very useful and insight-generating activity. While there are a number of excellent, detailed research papers describing specific, significant worms, we were unable to find a broad survey of worms in the literature. Using a breadth-first approach, we sorted through the thousands of malicious code descriptions to determine the ones that could be considered worms, then examined these worm descriptions to classify them and determine the ones that are truly interesting. The purpose of this paper is to aid others in the community by sharing this (tedious) legwork. We present here a roadmap to this vast library of virus and worm information, identifying those strains that we consider to be interesting to the worm researcher. Following the introduction that this paper provides, the worm researcher can then examine the many well-written, depth-first explorations of particular worms (e.g., Code Red [7] and Slammer [8]).

In this paper, we discuss the past and present of worms and related malicious code (through the first quarter of 2003). The paper is structured as follows:

- Section II presents some of the varying definitions for malicious code categories: worms, viruses, Trojan horses, remote access Trojans, and backdoors. We outline the significant distinctions that we are making to determine the worms that we include in this study. Then, we divide worms into three broad categories for detailed discussion of their key innovations and impact in the next three sections.
- Section III reviews important e-mail worms.
- Section IV reviews Windows file sharing worms.
- Section V reviews traditional worms.

- Section VI presents some high-level quantitative trends extracted from Network Associates' Virus Information Library.
- Section VII proposes some of the qualitative trends we have observed and would be interested in discussing at the workshop.
- Section VIII provides a brief summary.

2. DEFINITIONS AND CATEGORIES

One of the impediments that we encountered in our study of worms was the different definitions and categorizations that various people use. In this section we outline the definitions of a worm and other types of malicious code, and we propose a definition for worm that we use for the remainder of the paper. We also describe our division of worms into three broad categories that we find useful for discussing recent trends.

2.1 What Constitutes a Worm?

The scope of this survey is "recent worms." For that reason, it is necessary to determine precisely what constitutes a worm. Unfortunately, a quick look at primary sources indicates that there is no consensus as to what that definition should be. For a good cross-section of the definitional landscape, please refer to the web sites of the various anti-virus companies and other security organizations, such as F-Secure [3], Network Associates [10], the SysAdmin, Audit, Network, Security (SANS) Institute [11], and Symantec [17]. We considered the following aspects of a definition for a worm:

1) Malicious Code.

There appears to be a general consensus that worms are malicious in nature. While some have talked about "good worms" that break into systems in order to repair them (in fact, the first computer worm was a benign propagating maintenance program created at XEROX PARC [12]), the connotation of worm now is generally one of an uninvited program. When mobile code is used for legitimate purposes, the term "agent" is generally applied.

Network Associates' Virus Glossary defines malware (also known as malicious code or malicious software) as "programs that are intentionally designed to perform some unauthorized (and often harmful or undesirable) act," and worms are typically considered to be one category of malicious code.

2) Network Propagation.

Another generally agreed upon aspect of worms is that they actively propagate over a network. Whereas earlier viruses often relied on humans to carry floppy disks from one system to another, a worm attacks another computer directly over some network interface. File-infecting viruses that infect local files that happen to be remotely mounted from another machine are generally not considered worms because they are not actively aware of the network.

3) Human Intervention.

One other relevant characteristic of worms is the degree to which user intervention is required for propagation – this characteristic is sometimes part of the distinction that is made between viruses and worms. Worms are sometimes thought of as requiring little or no human assistance in order to spread, whereas a virus traditionally has required user intervention to spread from one machine to another (e.g., copying via floppy disk).

However, there are others that define two categories of worm: one that requires user intervention to propagate (e.g., opening an e-mail message or attachment) and one that does not require user intervention [6]. There are, of course, varying degrees of user intervention: even some worms that do not require the user to actively execute or open malicious code files require the user to take other seemingly unrelated actions, such as rebooting or running a mail program (e.g., Klez).

4) Standalone or File-Infecting.

The anti-virus community has traditionally defined worms by contrasting them with viruses: a virus infects a file (its host), whereas a worm does not require a host file [17]. In other words (the words of F-Secure's Virus Glossary), a virus is "a computer program that replicates by attaching itself to another object" and a worm is "a computer program that replicates independently by sending itself to other systems" [3].

However, there are many outside the anti-virus community who do not make this distinction. To them, any program that replicates over the network could be considered a worm, regardless of whether it infects files or acts as a standalone program.

Because the nature of malicious code is constantly changing, any sort of classification will certainly be quickly found to be incomplete. Consider the authors of the seminal paper on the Morris Internet worm [1] who argued to great lengths (and no avail) that the Morris worm was actually a virus because the computers that it infected acted as hosts!

Again, the purpose of our study is to consider as much of the raw data as is feasible and extract trends that might provide insight. In order to ensure that we not blind ourselves to important trends, we assumed a very broad a definition of a worm:

> A worm is malicious code (standalone or file-infecting) that propagates over a network, with or without human assistance.

The most important characteristic of this class of malicious code, from our perspective, is the active use of network interfaces for propagation, whether that be e-mail, shared network drives, direct network connections, or some other interface. Any malicious code that can propagate over a network interface is included in our overview of past worms. While some of the examples of worms that we will present have traditionally been classified as viruses and/or Trojan horses, their worm-like spread across varying network interfaces warrants inclusion in this paper if for no other reason than to illustrate the evolution of malicious code with network capabilities.

2.2 Other Types of Malicious Code

There are other relevant categories of malicious code in addition to worms and viruses. The other most widespread type of malicious code is a Trojan horse, defined by the SANS Institute as "a computer program that appears to have a useful function, but also has a hidden and potentially malicious function" [11]. Trojan horses require user intervention in order to perform their malicious or unauthorized activities; in fact, a primary purpose of Trojan horses is to trick the user into executing the program or opening the file containing the Trojan horse. Typically, Trojan horses are contrasted with viruses and worms in that Trojan horses do not replicate [10].

Two other types of malicious code that are becoming more common are remote access Trojans and backdoors. A remote access Trojan is a Trojan horse that, when executed, enables some form of remote access and control to the now compromised system by unauthorized persons [4]. The intent is similar to that of a backdoor – "a feature built into a program by its designer, which allows them to gain full or partial access to your system" [10].

Depending on the definitions that one settles upon, all of these categories of malicious code are not mutually exclusive. Many of today's viruses do more than just infect files on their local machine: they also spread over the network like a worm. In addition, many viruses and worms trick the user into opening or executing the malicious program just like a Trojan horse. Many recent worms also open backdoors or drop remote access Trojans on the systems that they compromise. Throughout this paper we will note when a worm possesses characteristics of more than one malicious code category.

2.3 Categories of Worm

After studying a great number of worms, we found it useful to consider the various specimens and strains as three very broad categories into which worms can be grouped:

- E-mail (and other client application) worms
- Windows file sharing worms
- Traditional worms

This is not a strict classification scheme. A number of worms appear in two categories, and one (Nimda) appears in all three. Nevertheless, these categories represent distinct branches of worm development, in which meaningful trends can be detected. For that reason, we present them as three separate sections:

E-mail (and Other Client Application) Worms. The recent explosion in network-aware malicious code started with the invention of e-mail worms. Since the first few concept worms appeared in late 1998, there have been hundreds of e-mail worms. In Section 3, we provide an overview of the most significant developments in this area.

Although not e-mail, there are a number of other file-sharing applications that have been used to spread worms in a manner very similar to e-mail worms. These include client-to-client protocols (applications where end users can address each other and initiate communication, typically through an exchange medium), such as Internet Relay Chat (IRC), AOL Instant Messenger (AIM), and a variety of peer-to-peer file sharing systems. The worms that attack using these other protocols are genealogically very close descendents of e-mail worms, in that they all focus on tricking users into executing untrusted files. Because these other application worms are so similar to e-mail worms, and none have demonstrated any significant advancements distancing themselves from e-mail worms, they are currently not deserving of a separate category.

Windows File Sharing Worms. In 1999, the ExploreZip e-mail worm exploited the ubiquitous Windows file sharing (SMB/CIFS) protocol to further its spread. Since then, there have been a number of significant advancements in this area. At the time of writing, SMB worms appear to be the area in which the virus/worm writing community is focusing its attention and its creativity. As such, we present Windows file sharing worms in Section 4 as a separate strain worthy of further study.

Traditional Worms. The final category of worm development that we consider is the "traditional" worm, which is modeled largely on the Morris worm of 1988. These are worms that attack across the Internet using primarily direct connections over TCP/IP-based protocols, exploit vulnerabilities in operating systems and applications, typically do not require user intervention, and use other propagation vectors besides e-mail and Windows file sharing. This strain of worm development is presented in Section 5.

Each of the worm descriptions in the next three sections draws primarily from Network Associates' Virus Information Library [9], where the full worm descriptions can be found (on-line). Other extensive repositories of virus and worm descriptions are also provided by F-Secure [2] and Symantec [16]. Information from other sources is cited appropriately.

3. E-MAIL (AND OTHER CLIENT APPLICATION) WORMS

E-mail worms are programs that, when executed on a local system, take advantage of the user's e-mail capabilities to send themselves to others. E-mail has been used to propagate malicious code from as early as 1987, with the Christmas Tree Trojan horse. Mailers, as e-mail worms and viruses are sometimes called, have been incredibly popular among writers of malicious code in the past five years. They are extremely simple to write, and there are a number of toolkits and tutorials to help the aspiring virus/worm author readily available on the Web.

Some have argued that most mailers should not be considered worms, as they do not use direct network connections and most rely on some degree of human intervention to spread. However, we cannot ignore the fact that malicious code utilizing e-mail has proven to be the most effective means of infecting a sizable percentage of hosts on the Internet. In addition, e-mail worms are by far the most common form of network-aware malicious code, as we will show in the trends section (Section 6). We leave it to the individual reader to decide whether each of these examples of malicious code should be classified as worms, viruses, Trojan horses, or some combination. We will try to be precise as to correct terminology when describing each malicious code instance, but we will refer to malicious code that uses e-mail to propagate collectively as e-mail worms.

There are simply too many e-mail worms to consider them all individually. Table 1 provides an overview of the most significant e-mail worms, those which we feel represented a technological advance from previous worms.

Having surveyed a large number of e-mail worms, overall we have observed an increasing sophistication with respect to the e-mail capabilities that these worms employ. The first worms simply used local mail programs and/or mail APIs on a compromised

Table 1. E-mail Worms of Note

Worm	Discovery Date	Distinction
Christmas Tree	Dec. 1987	Was the first malicious code to use e-mail to propagate (though it did trick the user into opening a fake Christmas card, like a Trojan horse).
ShareFun	Feb. 1997	Was the first known virus to use e-mail to spread.
Antimarc	Sep. 1998	Was the first known virus to use both e-mail and IRC to propagate.
Ska/Happy99	Jan. 1999	Piggybacked on e-mail communication, sending out one copy of itself after each legitimate e-mail.
Melissa	Mar. 1999	Was the first *mass mailer*. E-mailed itself to the first 50 entries in the user's address book, causing a widespread epidemic.
ExploreZip	Jun. 1999	Added the ability to search the network neighborhood and infect open shares. Generated a plausible response to messages that had been received by the user, including itself as a Trojan horse attachment that looked like a self-extracting ZIP archive.
BubbleBoy	Nov. 1999	Exploited a vulnerability in popular Microsoft mail software, that allowed attachment to automatically execute. Utilized an obscure executable extension (.WSH) to trick users on mail clients that were not vulnerable.
LoveLetter	Mar. 2000	Introduced hidden double extensions (two file type designations under Windows operating systems, where the real file type is hidden). Used the prospect of a secret admirer to entice users to execute it.
Stages	May 2000	Used the Shell Scrap extension (.SHS), an executable type that was given special exemption by Microsoft to ignore the "always show extension" configuration option. Used 12 different permutations of subject line. Propagated via e-mail, 2 IRC clients, and shared network drives.
VBSWG Toolkit	June 2000	Is a sophisticated worm generator toolkit. Produces worms with many features, such as e-mail and mIRC propagation, random naming, and encryption/decryption routines. Produced the Anna Kournikova virus, among many others.
Magistr	Mar. 2001	Contained its own SMTP engine for mailing itself. Randomly attached private user files to its outgoing mail messages.
Sircam	Jul. 2001	Contained its own SMTP engine and communicated with open relays to send itself via e-mail. Was also bilingual (English and Spanish).
PeachyPDF	Aug. 2001	Was the first PDF worm (though it required a full version of Adobe Acrobat, version 5 and higher, to propagate).
Nimda	Sep. 2001	Was a significant traditional worm that combined multiple attack vectors, including e-mail..
Klez	Oct. 2001	Used its own SMTP engine to spoof the From: field. Exploited a different Microsoft e-mail client vulnerability to automatically execute. Infected the host with a symbiotic virus (Elkern.cav).
Goner	Dec. 2001	Used a screensaver extension (.SCR), which was not previously known to be executable. Propagated via ICQ (another client application). Disabled common security software, which was copied by later variants of Klez.
Bibrog	Jan. 2003	Spread via many different peer-to-peer systems, such as Kazaa, Grokster, Morpheus, and ICQ, in addition to using e-mail and IRC.

machine to send out copies of themselves to one or more addresses. Later e-mail worms contained their own SMTP engines so that they were not (as) dependent on the mail capabilities of the compromised machine (e.g., Magistr). Soon after, e-mail worms took advantage of the prevalence of open mail relays on the Internet (e.g., Sircam), then utilized these capabilities to spoof mail headers (e.g., Klez).

However, there has been a general lack of innovation in e-mail worms recently. In 2002, there were no e-mail worms with any notable technological advances. Bugbear was the second most prolific virus/worm of 2002 according to Sophos [13], but it was highly derivative: it exploited the same vulnerability as Klez, and like previous worms it generated random subject headers and spoofed e-mail addresses. The only interesting e-mail worm of the first quarter of 2003, Bibrog, is notable only for its use of many different peer-to-peer file sharing systems – hardly a revolutionary step forward.

4. WINDOWS FILE SHARING WORMS

Windows file sharing worms take advantage of the Microsoft Windows peer-to-peer service that is enabled whenever Windows determines networking hardware is present in a system. The underlying protocol is sometimes referred to as Server Message Block (SMB) and sometimes the Common Internet File System (CIFS). It was originally designed to allow small workgroups to

Table 2. Windows File Sharing Worms of Note

Worm	Discovery Date	Distinction
Many file infecting viruses	Pre-1999	Capable of infecting files that happen to reside on a shared network drive and are subsequently executed by others sharing that drive.
ExploreZip	Jun. 1999	Searched network neighborhood for drives that were not mapped and had no password protection. Dropped files into startup directories on unprotected drives.
NetLog	Feb. 2000	Used randomly generated IP addresses to infect unprotected shared drives across the Internet.
Shorm	Jan. 2001	Attacked local and remote systems. Provided a “hit list” of IP addresses to scan.
Nimda	Sep. 2001	Was a significant traditional worm that combined multiple attack vectors, including file sharing.
Ladex	Jul. 2002	Used hidden Admin$ share to attack systems that weren’t actively sharing any drives.
Opaserv	Sep. 2002	Exploited a vulnerability in file sharing protocol to infect password-protected shares.
Gaobot	Oct. 2002	Used password guessing to exploit weak passwords. Used protocol’s capability to enumerate user names.
Lioten	Dec. 2002	Used password guessing to exploit weak administrator accounts. Exploited remote job scheduling capability to run dropped program immediately.
Netspree	Jan. 2003	Assumed user identity.

share files in a trusted environment. Although security features have been added, it makes end users responsible for how these features will be used. As a result, this most ubiquitous of services is often configured in a very insecure manner. It has recently become a favorite target of worm authors.

DOS and Windows viruses have always had the potential to spread over shared network drives, if the network was configured in a fortuitous manner. However, over the past three years worm authors have become much more aware of the Windows file sharing capability and have begun to actively exploit it.

File-sharing propagation is rarely seen in isolation – most worms use attacks on Windows file sharing in addition to other attack vectors. Well-configured firewalls block all file-sharing connections from outside the organization, so it would be difficult for a worm that relied entirely on file sharing to gain much traction. But coupled with some other attack vector, file-sharing attacks can be very effective. As Nimda demonstrated, e-mail and HTTP are the best mechanisms for penetrating the enterprise perimeter, but file sharing can be a very effective mechanism for spreading inside the firewall.

While there are far fewer Windows file sharing worms than e-mail worms, there are still too many to describe each of them individually. Table 2 outlines the key features of the Windows file sharing worms that we believe represent significant advances in the state of the art.

The evolution of Windows file sharing worms has demonstrated an increasing sophistication in their exploitation of file sharing capabilities. The first viruses and worms that actively spread using Windows file sharing simply searched the Network Neighborhood for machines with unprotected shares (e.g., ExploreZip). After that, NetLog demonstrated that Windows file shares could be attacked remotely using random IP addresses. Later worms took advantage of default administrative shares that many users were not even aware existed (e.g., Ladex). Most recently, Windows file sharing has incorporated password-guessing attacks on shares that are not left open but rather protected too weakly.

It is interesting to note that Window file sharing worms are a fairly recent phenomenon, arising largely in the past two years. It is likely that the evolution of this type of worm will continue and more advances will be made as vulnerabilities are discovered and exploited.

5. TRADITIONAL WORMS

Traditional worms are worms that do not require user intervention (as opposed to many of the e-mail worms) and/or worms that use other methods of propagation (besides e-mail and/or network shares). Most often the propagation uses direct connections over TCP/IP-based protocols to exploit vulnerabilities in operating systems and applications.

In contrast to e-mail and Windows file sharing worms, there are relatively few traditional worms. Many of the worms that do exist showed very little novelty: simply changing platform or exploiting a different but already disclosed vulnerability is hardly groundbreaking. When a study of worms is developed in 20 years, it is likely that only Morris, Nimda, Code Red, and Slammer will be deemed worthy of discussion. However, we will briefly present some of the more interesting features of a number of other traditional worms in the table below (Table 3), if only because they are the only data points in existence.

The Morris worm was the first truly significant worm, introducing many of the key features still found in worms today, including password-guessing attacks, exploitation of vulnerabilities, and multi-phase attacks. Furthermore, it utilized a previously undisclosed vulnerability as one of its attacks, the only significant day-zero attack by any worm. (There is some debate as to how publicized the fingerd vulnerability was prior to the Morris worm [14], but the patches were definitely not available before the outbreak.) The Morris worm was the most sophisticated worm

Table 3. Traditional Worms of Note

Worm	Discovery Date	Distinction
Morris/Internet	Nov. 1988	Was the first significant worm. Was multi-platform. Exploited multiple vulnerabilities. Included a zero-day attack. Attacked only neighboring systems.
ADM	May 1998	Introduced random scanning of IP address space.
Ramen	Jan. 2001	Exploited three vulnerabilities.
Lion	Mar. 2001	Was a stealthy, rootkit worm.
BoxPoison	May 2001	Was a multi-platform worm. Exploited multiple vulnerabilities.
Cheese	Jun. 2001	Was a vigilante worm that secured vulnerable systems.
Code Red	Jul. 2001	Was the first significant traditional Windows worm. Was completely memory resident.
Walk	Aug. 2001	Recompiled source code locally.
Nimda	Sep. 2001	Was a hybrid Windows worm – attacked client-to-client, server-to-server, client-to-server, and server-to-client.
Scalper	Jun. 2002	Was a near zero-day worm (released 11 days after announcement of vulnerability). Built a peer-to-peer network of compromised systems.
Slammer	Jan. 2003	Used a single UDP packet for explosive growth.

of any kind, and it was another ten years before any other significant traditional worm appeared.

Most of the traditional worms have exploited Unix-based operating systems such as Linux (e.g., ADM, Ramen, and Lion). It has only been in the past couple of years that Microsoft operating systems have been targeted by traditional worms, starting with Code Red and Nimda in 2001 and continuing with Slammer in 2003.

One thing to note in the traditional worms is that most of them exploit vulnerabilities to propagate, and the time between the announcement of a vulnerability and its exploitation by a worm has been shrinking (in general). In 2001, there was almost two months between the announcement of a DNS vulnerability and the release of the Lion worm that took advantage of it. Code Red I was released only one month after the vulnerability that it exploited was confirmed by Microsoft. In 2002, Scalper took advantage of an Apache vulnerability that was only 11 days old. Of course, worms have still been able to exploit older vulnerabilities successfully – the vulnerability utilized by Slammer in 2003 was six months old when that worm created widespread denials of service.

6. QUANTITATIVE TRENDS

In order to determine approximate statistical trends in worms, we searched the Network Associates' Virus Information Library (VIL) for the number of entries that could be classified according to our different categories. Leveraging our access to the raw XML files for the VIL database, our methodology at a high level was to perform searches for specific keywords on the body of each VIL entry, then to manually inspect each entry that contained the keywords in question. (Access to the database files enabled better searching and processing than can be conducted using the standard web interface to virus information repositories – for example, there is standard boilerplate text in each entry that makes on-line text searches of the entire entry often fruitless.)

There are two caveats with approximating trends using this method. First, the actual number of worms and their variants is much larger than the number of VIL entries. For example, the Deborm worm that was first discovered in March 2003 already has 39 known variants, but it is represented by a single VIL entry. Even new viruses and worms that are not necessarily direct variants of existing ones can sometimes be detected by somewhat "generic" signatures, in which case a new VIL description may not be generated.

We assert that the number of VIL entries represents the number of significant, sufficiently distinct worm instances. In other words, each VIL entry can be thought of as corresponding to the "least publishable unit" of worm authorship. So while the number of VIL entries does not correspond precisely to the number of worms discovered, it does provide a measurable indication of where the virus-writing community is focusing its creative energies.

The second caveat is that performing text searches for keywords is an inexact method of classification. Although there are primary and secondary classifiers for each VIL entry (such as worm and mass mailer), these are not used consistently enough to be relied upon for our purposes. The VIL is like most, if not all, virus description databases, in that its purpose is to inform the reader regarding individual viruses and worms, rather than to be looked at holistically. Virus descriptions are intended primarily for customers, in order to aid them in identifying malicious code and recovering from outbreaks. While much of this information is still useful for this study, the lack of consistency between entries complicates our efforts. Furthermore, VIL entries have been written over a number of years by a variety of researchers and each one has a slightly different way of using the different fields, thus contributing to some inconsistency.

One final note about the data: as mentioned previously, our categories are not mutually exclusive. There are worms that are listed under multiple categories (e.g., Nimda, which is listed under e-mail, Windows file sharing, and traditional). Therefore, the numbers for individual categories below cannot be added together to calculate the total number of worms.

In spite of these caveats, searching the VIL provides a rough but valid approximation of the number of different worm instances for the purpose of observing relative trends. We present the data from 1998 through the first quarter of 2003. There were too few worms prior to 1998 to merit inclusion. We extrapolate the first quarter data of 2003 for the entire year in order to include that year as part of the comparison. The data is shown below in the following table (Table 4) and graph (Figure 1).

Table 4. Virus Information Library Entries by Category

Category of Worm	1998	1999	2000	2001	2002	2003*
Traditional	1	1	0	10	3	4
Windows File Sharing	0	7	14	20	28	80
E-mail	1	18	44	93	159	192
IRC	1	16	42	23	45	84
Peer-to-Peer	0	0	1	1	44	128

(2003 figures are projected from actual 1st quarter totals)*

As one can see from the data in Table 4, the number of worms in all categories has increased over the years. This can be explained intuitively by a number of factors. One is that the virus writing community as a whole seems to learn from other previous, well-publicized malicious code outbreaks. As worms are seen to have significant impact, more prospective authors learn to write worms. A second related factor is that worm-writing knowledge becomes encapsulated in virus and worm generation toolkits. These toolkits have increased in number and become more sophisticated over the years. These generators make the job of creating a new worm almost trivial for malcode authors, though there must be some note of originality in order to warrant new VIL entries.

A second statistic of note is the significantly larger number of e-mail worms over the years. There were a total of 363 e-mail worms in the VIL from 1998 through Q1 2003, while there were only 89 Windows file sharing worms and 16 traditional worms. This underscores two primary points: the effectiveness of e-mail worms in the past and the ease with which e-mail worms can be generated in the present.

On the other hand, it does seem possible that the popularity of e-mail worms is not increasing as rapidly as it has in the past, or as rapidly as for the other types of worms. While we do not want to read too much into the extrapolated data for 2003, the slope of the e-mail line seems to be decreasing. It is possible that a lack of originality or creativity, rather than a lack of popularity, could

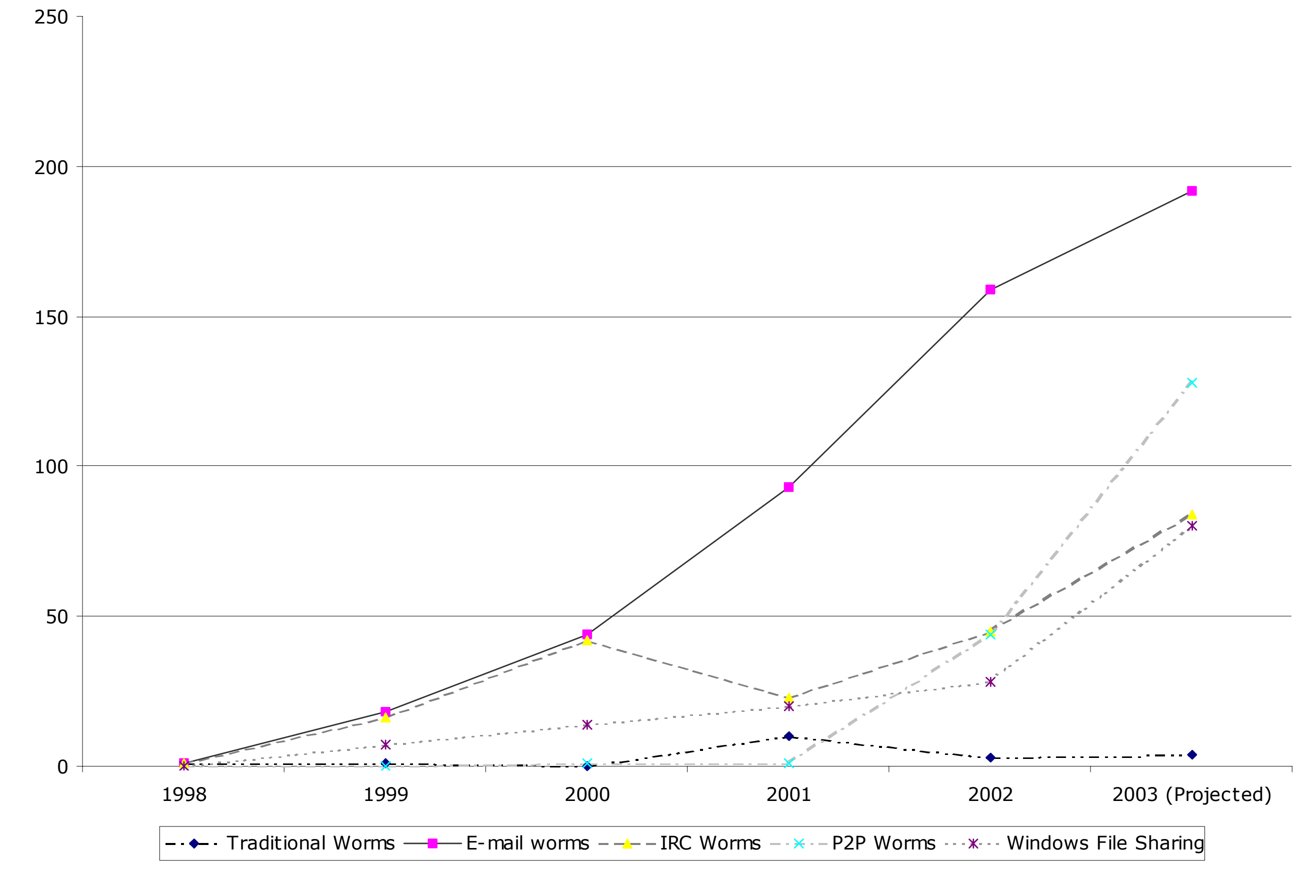

Figure 1. Virus Information Library Entries by Category

account for any decrease in e-mail worm prevalence. As defenses against e-mail worms become better and users become smarter about safe e-mail practices, it will take more innovation on the part of the worm writer to create a new e-mail worm that is significant enough to merit a new description.

Also of note is the increasing popularity of worms that utilize Internet Relay Chat (IRC) and various peer-to-peer programs to propagate. These worms grew out of e-mail worms, as authors learned how easy it was to add these propagation vectors to their worms. If a compromised machine is running an IRC client, the worm only needs to change a script file to ensure that the worm will be delivered to anyone connecting to the same IRC channel as the machine's user. If a compromised client is running one of many different peer-to-peer file sharing programs, the worm only needs to place copies of itself into the appropriate shared directories. All of the e-mail worm strategies for tricking or enticing a user into executing the worm program are just as applicable to these propagation methods.

Windows file sharing worms are definitely increasing rapidly in popularity. In the past year especially, there have been a number of significant worms that have taken advantage of Windows file sharing to propagate. Successful e-mail worms such as Sircam, Klez, and Bugbear have also included the capability of spreading via network shares. Windows file sharing worms are a relatively recent phenomenon and still maturing – it will be interesting to watch for new advances in this area and see if the trend in popularity continues.

Finally, there are too few traditional worms to be able to discern any real trends. In 2001 there were a number of vulnerabilities that were exploited in traditional worms, but aside from that year the prevalence of traditional worms has paled in comparison to worms that rely on e-mail and other propagation vectors.

7. QUALITATIVE TRENDS

We can extract a number of trends subjectively from our study of past and present worms. These are not based on empirical statistics. Rather, these are our subjective impressions taken from studying hundreds of first-hand worm descriptions. While others might have found different trends to be interesting, we feel that taking all of the raw data together supports these observations.

1) Commoditization

Advances in worms seem to become commodities very soon after they appear in their first significant worm. In other words, the sophisticated becomes routine very quickly: what is interesting or innovative becomes assimilated into the common body of knowledge and then can appear in routinely in later unrelated worms. Especially with respect to e-mail capabilities, complex worms have become trivial to write. Whether building atop a base of existing code or using a point-and-click toolkit, virtually anyone can create an e-mail worm. Increasingly, other features such as IRC capabilities and peer-to-peer protocols are becoming commonplace as well and it seems likely that Windows file sharing is the next area of innovation that could become commoditized.

2) Convergence

Related to the commodization of worm capabilities is a trend towards the convergence of worms and other malicious code types. It seems likely that malcode authors do not think of themselves as exclusively “virus authors” or “worm authors” or “Trojan horse specialists”, etc. Advances in viruses, remote access Trojans, and backdoors get incorporated into the general body of knowledge along with worm innovations. As such, we have seen many file-infecting viruses that spread like worms over the past few years. Recently, worms that drop remote access Trojans have become more common. Categories of malicious code are arguably becoming less important as specific instances of malware become more sophisticated and incorporate more features from the different categories.

Consequently, attempting to classify worms becomes more difficult and less useful. While many e-mail worms in the VIL have the term @M or @MM in their name (denoting mailer or mass mailer, using a standardized malicious code naming convention), it is not really fair to label them as simply e-mail viruses or worms. Today's e-mail worms spread by both e-mail and a variety of other protocols, and even traditional worms (e.g. Nimda) have used e-mail as one of their many propagation vectors.

3) Social Engineering

Worm authors continue to come up with new ways to pique the curiosity users, in particular recipients of e-mail. Malcode authors understand that you can indeed fool some of the people some of the time, and especially with the speed and ubiquity of e-mail, you only have to fool a few of the people to have a successful worm. It appears to be an axiom that worm writers will always come up with some new insidious method to cause a user's curiosity to overcome his/her skepticism for the moment needed to compromise his/her computer.

E-mail has been the most popular propagation vector for worms, and e-mail users have been tricked in a number of different ways over the years. But even as many users have learned to be more wary, the malcode authors have adapted by improving the mail messages generated by e-mail worms to look more legitimate. They have experimented with generic message contents, simple well wishes, standard replies to legitimate messages, bilingual messages, complex schemes for generating convincing text, and even warnings or fixes concerning viruses promised in attachments.

4) Additional Propagation Vectors

Although e-mail has been the most widespread protocol for propagating malicious code, the basic model of tricking users into opening infected files is applicable to any protocol that supports file sharing. Worm authors have attempted to exploit every popular client-to-client file sharing protocol, including IRC, peer-to-peer systems, AOL Instant Messenger, MSN Messenger, and ICQ, among others.

5) Technology /Vulnerabilities

While social engineering has played a large role in many of the most widespread worm outbreaks, recently it has been the worms that exploit vulnerabilities that have spread the fastest and done the most immediate damage. Code Red and Slammer demonstrated how a common, unpatched vulnerability could be utilized to spread a worm so quickly that denials of service ensue. Nimda demonstrated how effective a worm that exploits a vulnerability can be when combined with e-mail and other propagation vectors.

Beyond standard social engineering, worm authors have also discovered new ways of tricking the user without relying on clever content. Examples include using double extensions, new “dangerous” extensions (file types that were not previously known to possibly contain malicious code), and vulnerabilities in “innocuous” data types (data that was not previously known to be executable and therefore capable of performing malicious actions).

It is worth noting that, other than the Morris Worm, not one other worm used a novel (i.e., zero-day) vulnerability. In every case, the worm authors were able to use a published vulnerability to significant effect. The shortest window between a published vulnerability and the appearance of an exploiting worm was 11 days, but one or two months appears to be the most common interval.

6) Speed of Propagation

Worm authors seem to have recognized that they are in a race against the anti-virus vendors to infect as many systems as possible before new signature files can be developed and deployed. While early e-mail worms showed some restraint in their propagation, it was Melissa that demonstrated the effectiveness of spreading quickly before defenses can be put in place. Since then, worms have gotten steadily faster, with Code Red and Nimda setting new marks for speed.

The ultimate high-speed propagation may have finally been reached, with Slammer. While it is technically possible for a worm to propagate even faster than Slammer (especially with some form of a pre-attack phase) [15], it is certainly not necessary: Slammer demonstrated that it was fast enough to infect just about every potential target before any meaningful human-mediated response was possible.

7) Countermeasure Awareness

Worm authors are more than aware of the countermeasures being used against them, primarily anti-virus software. Monitoring virus discussion boards, it is evident that they routinely test their viruses against the most popular anti-virus clients in order to ensure that they will go undetected, for at least the first few critical hours. Malcode authors continue to uncover dangerous extensions and new executable types – from macro-enabled documents to screensavers to CGI scripts to compiled help files to “shell scraps” and on – that anti-virus products did not previously know had to be scanned.

The more recent successful worms have used active countermeasures to thwart detection by anti-virus products:

- Some viruses and worms actively undermine existing security products, by disabling major anti-virus software packages for example (e.g., Klez).
- A few viruses and worms check if they are being run in a simulated environment and behave differently in order to evade detection and analysis (e.g., Nomis).

8) Common Platforms and Software

Worm authors are actively looking to exploit flaws in, or trick users of, ubiquitous software applications. As we have seen in e-mail worms, Microsoft Outlook and Outlook Express are particularly attractive targets, as a flaw in or clever use of one of these mail clients can guarantee a very widespread outbreak. The sheer number of Microsoft Windows server operating systems deployed ensured that some percentage would contain a default, unpatched installation of IIS for the Code Red worms to exploit. As Linux has become more popular, the first worms have surfaced for that platform.

In contrast, platforms that are less common (in part because of their greater expense), such as Sun OS and HP Virtual Vault, have not been vulnerable to almost all of the previous virus and worm outbreaks. There are a number of contributing factors for this lack of outbreaks, of course, but certainly the fact that they are not common platforms contributes. They are not widespread enough for most common malware authors to have access to them, and there are fewer deployed systems (and users) to take advantage of for an outbreak.

8. SUMMARY

Having studied literally hundreds of worms, the only thing that we are confident in predicting is that we will be surprised and amazed by the next truly innovative worm.

However, it is clear that the vast majority of worms are derivative in nature. These worms have little or no originality and can often be prevented simply by protecting against the last worm. If this study has revealed anything in terms of trends, it is that the history of malicious code is, for the most part, evolutionary. By defending against yesterday's attacks, you can effectively protect against the vast majority of tomorrow's threats. This is not simply a case of closing the barn door after the horse has bolted – it is closing the barn door after one horse has bolted but before the other 99 get the idea to follow him.

The other major point of interest is that best security practices do work against these worms. Systems that have been kept up-to-date with patches have been largely invulnerable to worms. Demilitarized zones that strictly limit incoming and outgoing communications prevent worms from spreading. And strong user education can seriously dampen the effect of social engineering attacks.

Admittedly, we can never fully eliminate the risk that a truly novel worm will surprise us all and cause a great deal of damage. Worms like Morris, Nimda, and Slammer demonstrate a certain genius that can't be predicted or anticipated. But the use of best practices significantly reduced the risk from even those worms for those organizations and individuals that were prepared.

Finally, it is worth noting that the most novel, inventive worms have not had malicious payloads. This might not be entirely fortuitous – perhaps thus far the creative genius required to author a truly great worm has precluded the wanton destruction that some of the copycats have attempted? Whether this trend has occurred by coincidence or not, we cannot count on it always being true.

9. ACKNOWLEDGEMENTS

This work was performed by Network Associates, under contract with and supported by the U.S. Defense Advanced Research Projects Agency Advanced Technology Office under Contract No. MDA972-03-C-0023, ARPA Order No. P193/00.

10. REFERENCES

[1] Eichin, M. and J. Rochlis. “With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988”. Proceedings of the 1989 IEEE Symposium on Security and Privacy (Oakland CA, May 1989), IEEE Computer Society, 326-344.

[2] F-Secure. F-Secure Computer Virus Information Center. http://www.f-secure.com/v-descs, 2003.

[3] F-Secure. “F-Secure Corporation Virus Glossary”. http://www.f-secure.com/virus-info/glossary.shtml, May 2003.

[4] Grimes, R. “Danger: Remote Access Trojans”. Security Administrator, http://www.microsoft.com/technet/security/ virus/VirusRAT.asp, September 2002.

[5] Kaspersky, E. Computer Viruses. Kaspersky Lab, http://www.viruslist.com/eng/viruslistbooks.html, 2000.

[6] Lemos, R. “Year of the Worm: Fast-spreading code is weapon of choice for Net vandals”. CNET News.com, http://news.com.com/2009-1001-254061.html, March 2001.

[7] Moore, D., V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. “Slammer Worm Dissection: Inside the Slammer Worm”. IEEE Security & Privacy, Vol. 1 No. 4 (July-August 2003), 33-39.

[8] Moore, D., C. Shannon, and J. Brown. “Code-Red: a case study on the spread and victims of an internet worm”. Proceedings of the Internet Measurement Workshop 2002 (Marseille France, November 2002).

[9] Network Associates. Virus Information Library. http://vil.nai.com, 2003.

[10] Network Associates. “Virus Glossary”. http://mcafeeb2b.com/naicommon/avert/avert-research-center/virus-glossary.asp, 2003.

[11] SANS Institute. “SANS Glossary of Terms Used in Security and Intrusion Detection”. http://www.sans.org/resources/ glossary.php, May 2003.

[12] Shoch, J. and J. Hupp. “The Worm Programs: Early Experience with a Distributed Computation”. Communications of the ACM, Vol. 25 No. 3 (March 1982), 172-180.

[13] Sophos. “Klez worm is most prolific virus of the year”. Sophos Press Releases, http://www.sophos.com/pressrel/uk/ 20021204yeartopten.html, December 2002.

[14] Spafford, E. “The Internet Worm Program: An Analysis”. Purdue Technical Report CSD-TR-823, http://www.cerias.purdue.edu/homes/spaf/tech-reps/823.pdf, December 1988.

[15] Staniford, S., V. Paxson, and N. Weaver. “How to 0wn the Internet in Your Spare Time”. Proceedings of the 11th USENIX Security Symposium (San Francisco CA, August 2002).

[16] Symantec. Symantec Security Response – Search and Expanded Threats Page. http://securityresponse.symantec.com/avcenter/ vinfodb.html, 2003.

[17] Symantec. “What is the difference between viruses, worms, and Trojans?”. http://service1.symantec.com/SUPPORT/ nav.nsf/pfdocs/1999041209131106, November 2002.