SOURCE CODE ANALYSIS AND PERFORMANCE MODELING OF MALWARE

Anand Mylavarapu, Anil Chukkapalli
Computer Science Department
St. Cloud State University
St. Cloud, MN-56301
myan0301@stcloudstate.edu

Abstract

The exponential growth of malware is degrading the performance of the machines, networks and various wireless devices. The objective of this research is to study those virus' source codes that primarily affected machines running versions of Microsoft Windows Operating System and analyze them. Our goal is to harden the internet against attacks. All the viruses analyzed in this paper were released in 2003 or years preceding it. Striking similarities were found in the patterns they used to compromise the systems. Also we took a look at the network performance modeling techniques that are being developed and their significance in developing a hardened automated defense system which is able to contain new threats. As will be clear from discussion further, the terms viruses and worms would be used to describe a malicious program, depending on the context, often interchangeably. Such studies will help identify weaknesses in software, develop antivirus system software and harden the internet. The analysis will also aid in computer forensics to determine the sequence of files compromised.

1. Introduction

From the moment a computer is connected to the Internet or local network, it becomes an object for attack, misuse or other undesirable actions. The threat of worms started from as far back as the Morris worm released in 1988. Every year there have been more releases and variants like Melissa in 1999, Code Red in 2001 and Blaster, Slammer and Sapphire worm, in 2003. Of all these, Sapphire spread the fastest [1]. Code-Red spread faster than Melissa and Sapphire spread faster than Code-Red, proving the authors are learning from past mistakes. With the availability of more information about the source codes, writing new viruses/worms also doesn't need extra-ordinary programming skills. Clearly, we need to analyze the patterns to be able to be better prepared.

A *Virus* is a malicious program that replicates and propagates by attaching itself to another program. An analogy to its behavior can be found in a biological virus and hence the name "virus". Viruses have the potential to damage our files, any software that's on the machine and also hardware. *Worms* can be classified as a subclass of virus, the main difference being that it can propagate itself without active user involvement. Normally, a worm would distribute itself across the systems, in the process consuming network bandwidth, CPU time, memory and often slowing down the system to an extent where it will stop responding and this will lead to a Denial of Service (DoS) attack. *Trojan horses* are seemingly harmless programs that cause a lot of damage by performing actions in contrast to our intentions. They can be concealed in an application. They end up getting access to a system, when they are downloaded, as part of freeware/shareware software. They have a client – server architecture. [2]

Viruses can be broadly classified into seven categories namely, companion virus, executable program, memory, boot sector, device driver, macro and source code viruses[2]. Each of the viruses could be defined by its characteristic behavior. *Malicious code*, a broad term for unfriendly codes, has been an issue in computer security for many years, dating back to Ken Thompson's self replicating code [3] [20].

This paper will survey and analyze the source codes to understand the working of worms and viruses and identify the common patterns in their behavior. Code for most of the viruses is available on the open internet and this presents its own challenges, depending, on how we look at it. Section 2 of this paper analyses *source codes*. Section 3 deals with *Network Performance modeling*. Section 4 describes with the ways to harden the system, on a corporate level and personal level. Section 5 concludes the paper.

2. Source Code Analysis

This section analyzes the source codes and presents them in the form of a pattern, to better understand their behavior. Block Arrows in a pattern indicate a YES and Bold arrows indicate a NO, for decision boxes.

2.1 Code Red

Figure 1 illustrates the pattern of Code-Red attack. Code Red is a malicious program that can scan the web by itself, looking for computers it can copy itself to. On certain dates, all infected machines, with code red, are supposed to bombard the White House website with bogus data packets. For the rest of the month, infected computers will be scanning for uninfected machines. It also briefly replaces web pages with the text "Hacked by Chinese". The Code Red Worm degrades performance of Microsoft Index Server 2.0 and Windows 2000 Indexing service on computers running Microsoft Windows NT 4.0 and Windows 2000, which run IIS 4.0 and 5.0 web-servers. [4] The worm uses a known *buffer overflow vulnerability* contained in the Idq.dll file. On July 19, 2001, in just 9 hours the virus infected around 250,000 computers in North America and Europe [5].

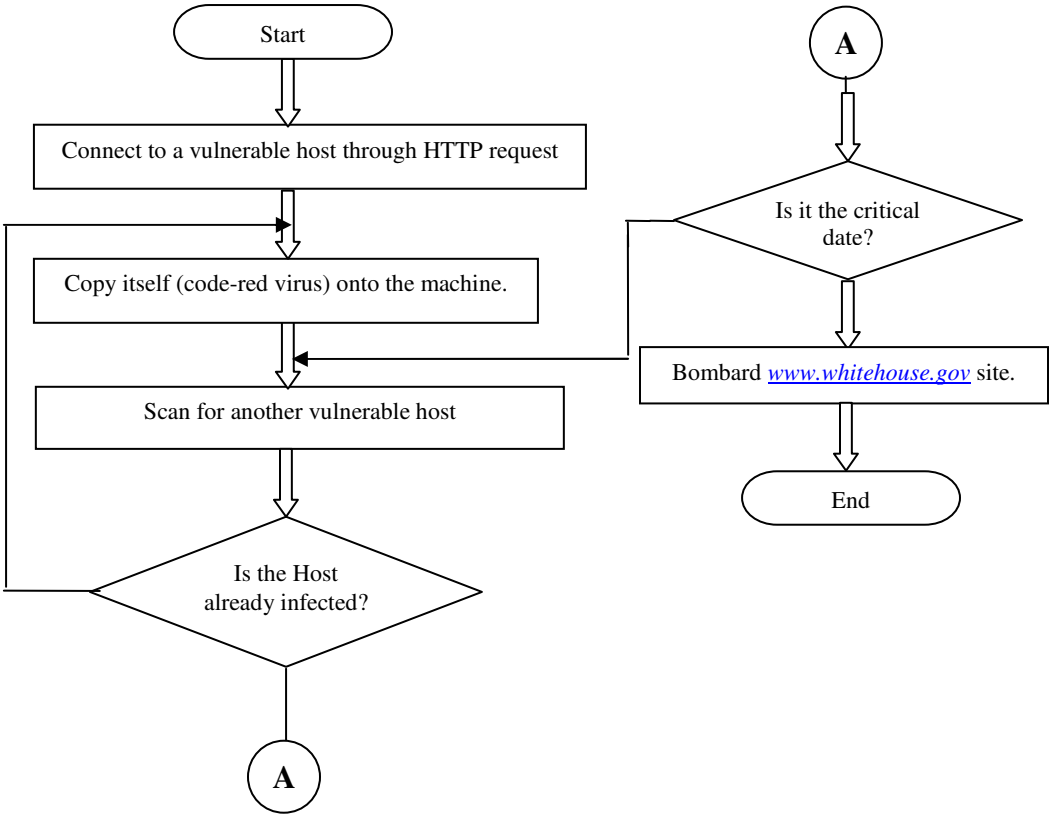

Figure 1 Code-Red Pattern

Code Red Worm is the first such instance, which does not require the presence of any file altogether. It neither exists in a file format in the infected computer, nor manifests itself in any file form when being distributed. The virus accommodates itself solely in the memory of the infected computer and exists as a running process. This makes it difficult to detect and remove.

2.2 Blaster

Figure 2 depicts the behavior of the Blaster worm. The worm generates a random IP address to start with and scans for a machine that has TCP port 135 open, the most frequently used port for RPC's. Once it gets hold of a vulnerable machine, it will send a variation of the dcom.c *exploit*, which makes use of a security loophole in RPC protocol. Then the exploit will open a shell either on TCP port 4444 or UDP port 69. The target (*exploited machine*) now sends ftp *get* to the source (*exploiting machine*), downloads the worm binary *msblast.exe* and installs a registry key in the run area of registry so that the code is executed automatically, at every system restart. The worm spreads this way and all such exploited machines launch a distributed denial of service (DDoS) attack on port 80 of www.windowsupdate.com website, on a specific date i.e. August 16, 2003[6].

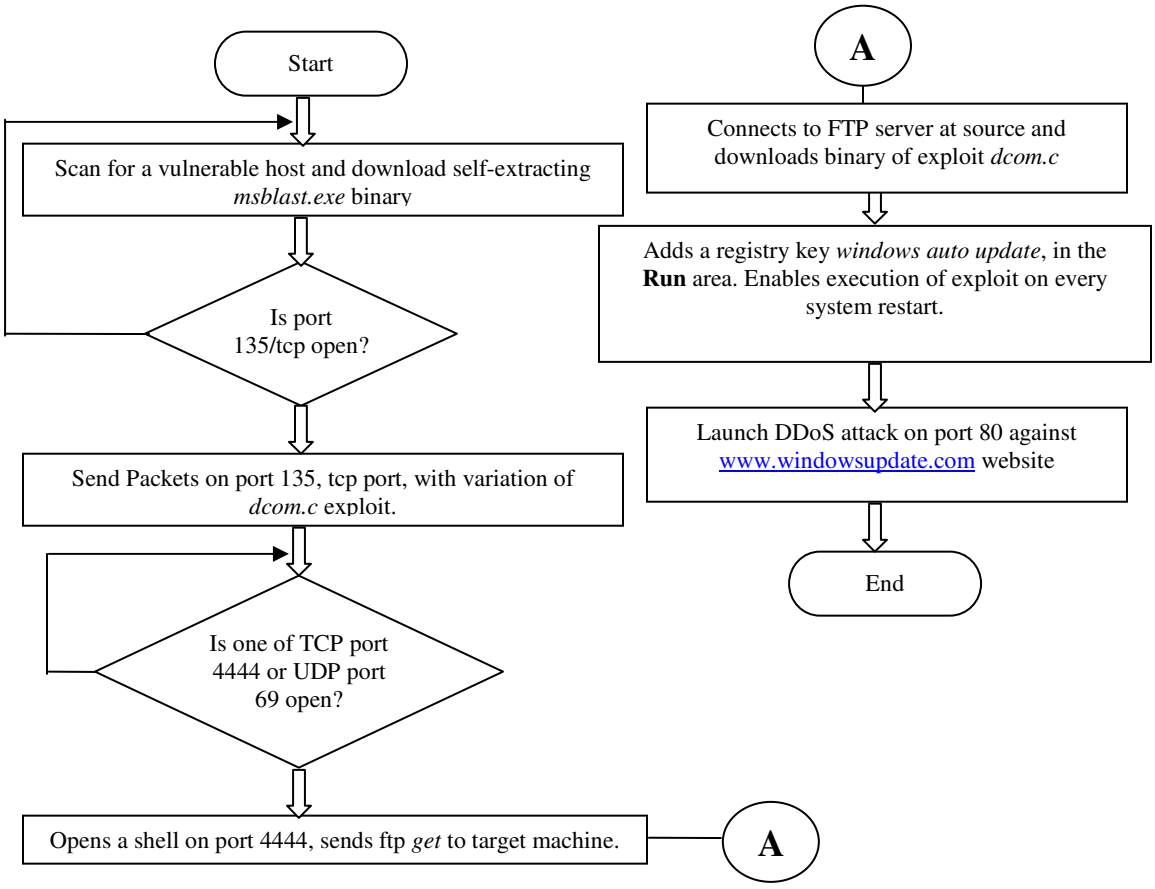

Figure 2 Blaster Pattern

This worm exploits vulnerability in the Windows operating system for systems that do not have the Microsoft RPC security patch applied. It generates enough traffic to shut down an internet host. The Blaster worm hit hundreds of thousands of computers, exploiting the DCOM RPC vulnerability using TCP port 135 in Windows 2000 and XP operating systems. [7]

2.3 Love-Bug

Figure 3, shows the behavior of the Love-Bug worm. It will first create a Windows Shell object and a File system object. It then resets the registry Timeout value to 0 and writes 3 files to the Run area of the registry. Setting the Timeout to 0 enables the virus to work at its own pace. It then generates a random number between 1 and 4 and takes the user to one of the 4 pre-programmed websites and sets the homepage in IE to one of them. Then, it creates an HTML page with a VB script embedded in it which creates a backdoor and gets ready to transfer all the confidential information, like passwords, accounts etc., to the author of the virus. It also creates an Outlook object, scans the address book and sends itself to all the entries, damaging their systems as well. It also destroys all files with particular extensions.

The “ILOVEYOU” worm struck hundreds of thousands of computers in Asia, Europe and the United States as workers clicked on an e-mail attachment called LOVE LETTER FOR YOU.TXT.vbs. In less than six hours, Love Bug spread worldwide. Some users received another version that substituted the "I love you" wording with "very funny joke." CERT reported 270,000 computers to be affected. The source code shows how simple it was to write these worms. Richard M. Smith, the programmer who helped arrest the author of the 1999 Melissa virus was surprised at the speed with which the ILOVEYOU e-mail worm spread. [8]

As soon as the attachment was opened, the malicious code accessed the Outlook address book and sent a copy of itself to every entry. Generating an extreme number of messages took almost no time. Next, the worm would delete the contents of Images (jpg and jpeg), Visual Basic scripts (vbs and vbe) and Java (je and jse), music files (mp3 and mp2) and a .vbs file would replace it. The worm also infected files on networked and mapped drives as well. Finally, the virus attempted to contact one of the four web sites in the Philippines that had a file called WIN-BUGSFIX.exe and download it. [8]

The key to ILOVEYOU is the Visual Basic Script macro language for Windows. For computers that have the scripting language turned on -- the default Microsoft setting for Windows 98 -- VBS can allow access to almost any system function: Copying, deleting and changing files are all possible. Some believe, Microsoft should have taken the ability to run such scripts out of Outlook a long time ago. Users, who want to take matters into their own hands and disable the scripting host, can do so by going to Control Panel > Add/Remove Programs > Windows Settings > Accessories and de-select the scripting host option.[9]

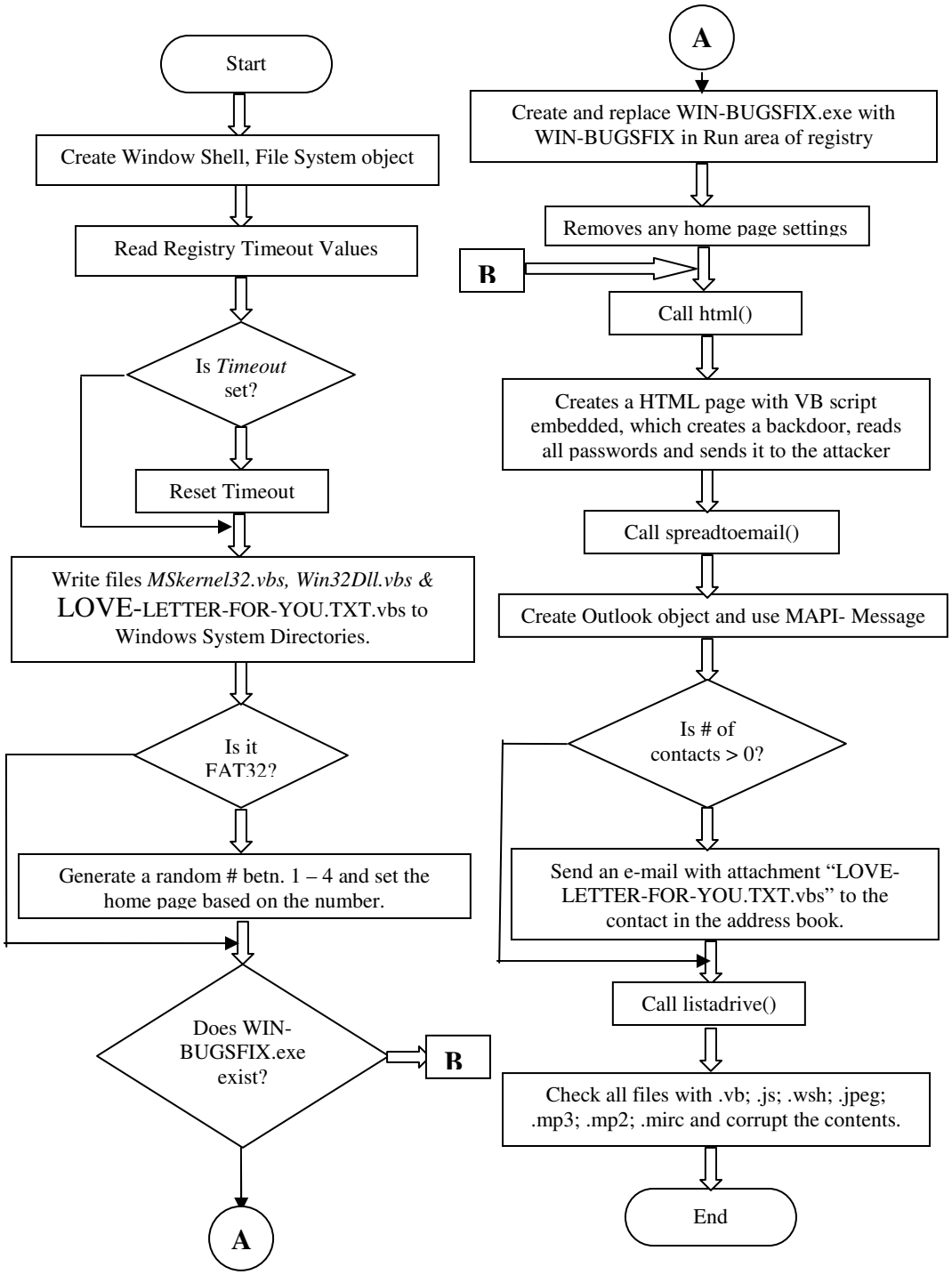

Figure 3 Love-Bug Pattern

2.4 Homepage

Figure 4, presents the behavior of the Homepage virus which was an e-mail virus. It is one of those typical viruses which, once it gains access to a system, will read the contacts in Outlook address book and e-mail them an attachment. Clicking on it would take you to one of adult sites, but it doesn't damage any files.

The behavior is somewhat similar to that of Melissa worm, released in 1999, Lovebug in 2000 and other such e-mail worms. About 23 hours were lost in curtailing its destruction spree. E-mailed messages had the subject "homepage" and the body contained the message, "Hi! You have got to see this page! It's really cool", with an attachment named "Homepage.HTML.vbs" [10]. "vbswg", a simple virus-writing tool[11], available on the internet was used, indicating the author might not have been a very skilled programmer. Bottom line, be careful while clicking on attachments, especially, if the extension of the file is vbs.

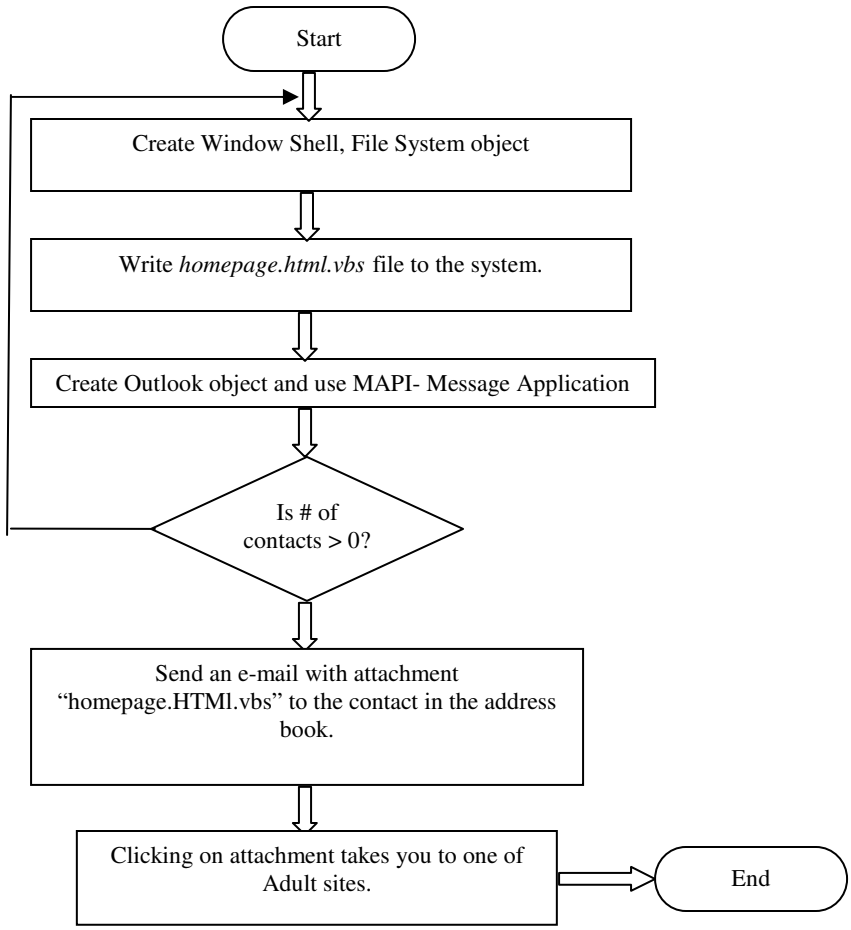

Figure 4 Homepage Pattern

As you can see, the behavior of Love-Bug and Homepage, both e-mail worms, exhibit similarity in attack patterns, but the objectives are clearly different. While the

former damaged the files and generated potentially huge losses, the latter only caused embarrassment. Also, the terms Virus, Worm and Trojan horse are all overlapping, if the behavior is compared to that in the literature definition.

2.5 Melissa

Melissa worm was released in 1999. It can be classified as a macro-virus, which primarily affected Word-97/word-2000 documents [21].

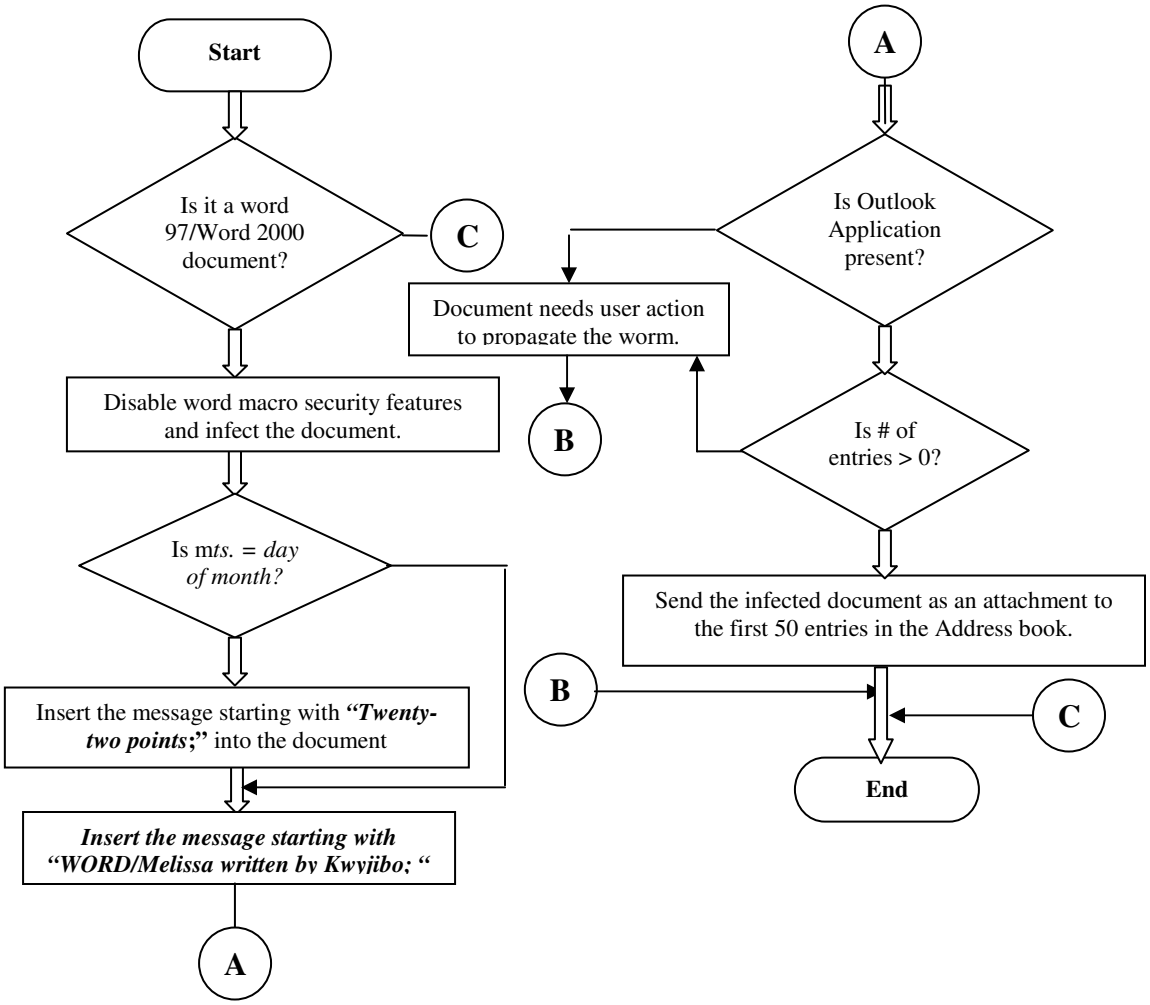

Figure 5 Melissa Pattern

It would start by looking for documents in the above mentioned versions, disable the documents macro security working It would then look for Outlook application in the system and if found, mail the infected document to the first 50 entries in the address book. If there was no outlook, it would infect the existing documents and would spread when someone transmits them, either through e-mail or a floppy disk It would also insert

the messages "*'WORD/Melissa written by Kwyjibo; 'Works in both Word 2000 and Word 97; 'Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!'Word -> Email | Word 97 <--> Word 2000 ... it's a new age!*" into the document and the message "*Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game's over. I'm outta here.*", if the minutes and day of the month, matched. David Smith is the author of this worm. The damages from this worm and its variants were estimated to be around $1.1 billion dollars [22].

3. Network Performance Modeling

This section briefly describes Network performance modeling and its importance. The purpose of network performance modeling is mainly to analyze network traffic patterns. This will help us identify any deviations from normal behavior. This is extremely important as this enhances the reaction time of administrators and helps improve the process of securing the network.

For example, if we have a network that can process packets at the rate of 1Gb/Sec, if a worm carries a payload of 4kb, it will need to generate at least 250,000 packets per second, to saturate the network. This simple calculation helped the defense in a court case. In general, this kind of information can prove crucial, in determining a worm's effectiveness.

A computer worm that randomly scans new hosts to infect can be expected to follow the simple epidemic model known from *biological epidemiology* [12]. This model assumes that a population of constant N hosts is all vulnerable but uninfected except for a small number that are infected and contagious. Initially, the infection spreads at a faster rate as more systems are vulnerable but not yet infected. As time progresses, the *infection* growth rate slows down.. The machines are scanned for vulnerabilities, in a random order throughout the network, increasing the volume of traffic generated; ultimately leading to a denial-of-service (DoS) attack.

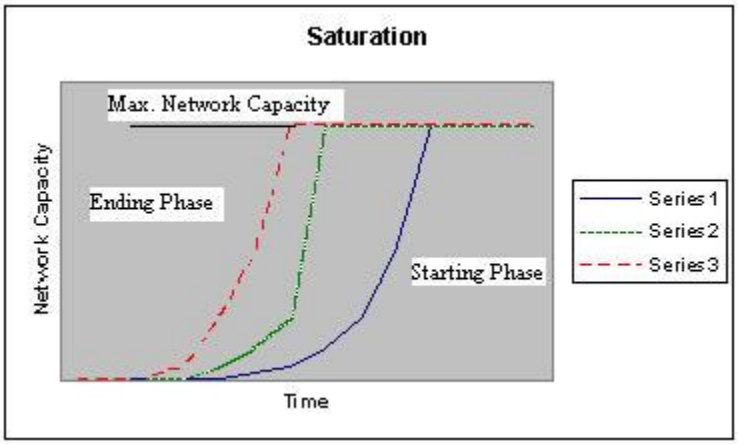

Figure 5 Malware Behavior

Figure 5, shows 3 virus transmissions, (selected randomly just to give an idea of the behavior), in which, series-3 saturates the network, the fastest and series 1, the slowest.

Kephart and White presented the *Epidemiological model*, on the lines of the biological epidemic model, to understand the behavior of viruses. This model fails in predicting the behavior of viruses in the current scenarios. So, Zesheng Chen, Lixin Gao and Kevin Kwiat came up with the Analytical Active Worm Propagation (AAWP) model, which characterizes the propagation of worms that employ random scanning. One of the ways proposed to detect the spread, is to look at the unused IP addresses to determine the scans they received. AAWP model helps estimate the number of these addresses needed to come up with an effective scan result. To do this, AAWP assumes that machine cannot infect others, until it is completely infected. It takes into account factors like worm's infection rate, systems are being patched at a reasonable rate and that a worm can scan the same machine twice or more. Another interesting result of this study was that worms that use localized scanning, like Code Red II and Nimda worms, spread at a slower rate, when compared with a worm that employs random scanning. But the ones employing localized scanning have the capacity to penetrate firewalls. [13][14] More and more research is going in this area and people are coming up with more models and simulators to better understand the behavior of these worms.

4. Hardening the System

This section lists the ways to harden the system, on a corporate level and personal level. As network security becomes increasingly critical to securing business transactions and computer resources, organizations must integrate security into their network design and infrastructure. Security policy enforcement is most effective when it is a built-in component of the network.

Cisco Systems protect organizations by providing low cost, easy-to-use and easy-to-integrate security solutions for networks. This Intrusion Detection System (IDS) is a security-specific option for Cisco IOS Software, integrating robust firewall functions and intrusion detection for every network segment. IDS' provide a higher level of protection as compared to the firewalls by protecting the network from internal and external threats and attacks. This IDS identifies more than one hundred of the most common security attacks using "signatures" to detect patterns of misuse in network traffic, including information and attack signatures. The signatures are chosen from a cross section of intrusion-detection signatures, represent severe breaches of security and the most common network attacks and information - gathering scans. This IDS is a very good choice for integrating multi-protocol routing with security policy enforcement. It allows customers to choose a router platform based on bandwidth, LAN or WAN density, and multi-service requirements simultaneously. [15]

To harden a stand-alone system, one of the first things to look at, is the system requirements [16]. We have to take care that we don't overdo the process i.e. we should disable services that might be vulnerable for exploitation but at the same time enable

those that are necessary for the normal functioning of the system. Basic guidelines while hardening could be built around the following points : [17]

- Is the system in a network or stand-alone?
- What are the protocols that you want to allow?
- What are the services you are expecting from the system?
- How many Operating Systems do you plan to support?
- Always go for the most basic install.
- Then start making additions to the system configurations. This option allows us maximum control over the features we are installing and helps close loop-holes.
- Create user accounts and delegate responsibilities clearly.
- Use the **Administrator** account only when absolutely necessary.
- Use encryption for storing files locally and use strong passwords. Put in place a policy for passwords i.e. how long a password should be, how many days it will be active and so on.
- Put in place a strong *system policy.*
- Maintain regular system logs.
- Put in place security options for services i.e. whether a service is disabled or not or how it can be started and by whom.
- Disable *Microsoft's File and Print services.* This prevents free access to our system's resources from the Internet.
- Decide what ports are to be open and make sure the remaining are closed. This will reduce chances of a backdoor entry.
- Update your system regularly with patches. Sometimes, this may also be a potential source for a problem.
- Use tools like netstat to test your system security settings.

5. Conclusion

In this paper we have analyzed various types of viruses and worms. We also took a look at the various network performance modeling techniques, which, will help design better anti-virus and IDS systems and broaden our defense systems. Also listed are some of the hardening techniques that a person can use on his PC, which might help us thwart threats, to some extent at least.

It is quite clear that the so called worms being released now, are in-fact, exhibiting the behavior of a virus, worm and Trojan horse combined. Also they are growing in sophistication and are becoming increasingly powerful and the scope is no longer restricted to wired devices but also the wireless devices [19]. Most of the viruses and worms follow the above patterns, though their objectives might be different. They will enter into a device, install themselves, damage files, try and propagate themselves, spread the damage and ultimately bring down the network.

With the internet becoming ubiquitous, there has to be an integrated effort, on part of the ISP's, Anti-Virus organizations, OS developers and the regular users to tackle the menace.

Acknowledgments

We sincerely thank Dr. Don Hamnes and Dr. Herath for their time, effort and feedback , in the writing of this paper.

References

1. David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, Nicholas Weaver, "*Inside the Slammer Worm*", IEEE Security and Privacy, 2003.

2. Andrew. S. Tanenbaum, *Modern Operating Systems II ed*., 2001, Chapter 9, pp. 606-628.

3. Matt Piercy, ITWales.com, "*Virus Attack on Mobile Phones*", http://www.itwales.com/998594.htm, November 15, 2004.

4. Eric Chien, Symantec Security Response- "*Code Red Worm*", "http://securityresponse.symantec.com/avcenter/venc/data/codered.worm.html"

5. Imran Saleh, "*A tutorial on what the Code Red Worm Virus is*". http://www.isaleh.com/codered/

6. SANS Institute - Computer Security Education and Information Security Training. http://isc.sans.org/diary.php?date=2003-08-11

7. John Leyden, *The Register*, "http://www.theregister.co.uk/2003/08/14/blaster_rewrites_windows_worm_rules/", Aug 14, 2003.

8. Robert Lemos, Tech News on ZDNET, "*Inside the I LOVE YOU worm*", http://news.zdnet.com/2100-9595_22-520463.html?legacy=zdnn, May 4, 2000.

9. CERT/CC, "*Frequently Asked Questions about malicious web-scripts redirected by web-sites*", "http://www.cert.org/tech_tips/malicious_code_FAQ.html#steps", 02/02/2000-12/07/2004."

10. Bill Goodwin, "*Homepage e-mail virus attack could have been prevented*", http://www.computerweekly.com/Article101990.htm, May 17th, 2001.

11. Sophos Corporation, "Virus Information, VBS/VBSWG-X", http://www.sophos.com/virusinfo/analyses/vbsvbswgx.html

12. Thomas M.Chen, Jean-MarcRobert, "*Worm Epidemics in High Speed Networks"*, IEEE Computer Society, http://engr.smu.edu/~tchen/papers/Computer_Jun2004.pdf, Jun 2004, pp 48-53, Vol. No. 37, No.6

13. Zesheng Chen, Lixin Gao, Kevin Kwiat, "*Modeling the Spread of Active Worms*", IEEE Infocom, http://www.ieee-infocom.org/2003/papers/46_03.PDF, 2003, issue no. 0-7803-7753-2/03.

14. J. O. Kephart, S. R. White, "*Directed-graph Epidemiological Models of Computer Viruses*," Proc. of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy, May 1991, pp. 343–359.

15. Cisco IOS Security Configuration Guide, Release 12.2, "*Configuring Cisco IOS Firewall IDS*", http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7c6.html

16. *Windows 2000 Security Handbook*, University of Waterloo, Canada. Information Systems and Technology, Jan. 2003, Chapter 21, copyright Osborne/McGraw-Hill). http://winxp.uwaterloo.ca/Documentation/Hardening_WXP.asp

17. Philip Cox, *Hardening Windows 2000*, "http://www.systemexperts.com/tutors/HardenW2K101.pdf", Version 1.0, March 30, 2001.

18. 62NDS Solutions Ltd, "*Virus Codes*". http://www.62nds.com

19. AME Info Business News. "*Handheld devices under virus attack*", http://www.ameinfo.com/news/Detailed/53789.html, 2005.

20. Nicholas Weaver, Security Focus Home, "*A brief history of the worm*", http://www.securityfocus.com/infocus/1515, November 26, 2001.

21. Sophos Corporation, "Virus Information, w97/Melissa", http://www.sophos.com/virusinfo/analyses/wm97melissa.htm

22. Paul A. Henry, CyberGuard Corporation, "*A brief look at the evolution of killer worms*", http://www.csoonline.com/whitepapers/050504_cyberguard/EvolutionoftheKillerWorms.pdf