Virus: A Menace for Information Security

Arun Bakshi
Assistant Professor(Sr) (IT),
Gitarattan International Business School, Rohini, Delhi,
lakshayabakshi@gmail.com

Vikas Dixit
Head Online Division Educosoft International India Pvt. Ltd.
urvikas@gmail.com

Kaushal Mehta
Assistant Professor, Bhai Parmanand Institute of Business Studies, Delhi,
kpu_713@yahoo.com

ABSTRACT

Computer virus is a program that copy itself to harm the computer without the knowledge of user. A virus can spread from one computer to another through some executable code. The user can sent it over a network or the Internet, or carried it on a removable medium such as a floppy disk, CD, DVD, or USB drive. Its chances of spreading from one computer to other increases by infecting files on a network file system or a file system that is accessed by another computer.

KEYWORD

Virus	Security
Information System	Computer
Worm	Linux
Unix	Compact Disc

Preface

The term "virus" is also commonly but erroneously used to refer to other types of malware, adware, and spyware programs. The correct term that should be used is “Malware”. Malware includes computer viruses, worms, Trojan horses, most rootkits, spyware, dishonest adware, crime ware, and other malicious and unwanted software), including true viruses. . A worm can exploit security vulnerabilities to spread itself to other computers without needing to be transferred as part of a host, and a Trojan horse is a program that appears harmless but has a hidden agenda.

Now-a-days, almost all computers are connected to the Internet which increases the chance of spreading malicious code. Viruses may also take advantage of network services such as the World Wide Web, e-mail, Instant Messaging, and file sharing systems to spread.

HISTORY

Creeper was the first virus detected on ARPANET in early 1970s. It was a self-replicating progam written by Bob Thomas at BBN in 1971. It copied itself to the remote system and displays a message, "I'm the creeper, catch me if you can!” It used ARPANET to infect DEC PDP-10 computers running the TENEX operating system. The Reaper program was created to delete Creeper.

“Rother J” was the first computer virus that appears “in the wild” means can spread outside the computer or lab where it was written. It was created by Richard Skrenta in 1981 as a practical joke when he was in high school. This program attached itself to the Apple DOS 3.3 operating system and spread via floppy disk. On its 50th use the Elk Cloner virus would be activated, infecting the computer and displaying a short poem beginning "Elk Cloner: The program with a personality."

A boot sector virus named “Brain” created by Farooq Alvi Brothers in 1986. It was operated out of Lahore, Pakistan, reportedly to detect piracy of the software they had written. A variant of Brain named “Ashar” has predated Brain on the basis of code within the virus.

In early days, users use floppy disks to exchange informat5ion and programs. PCs of the era would attempt to boot first from a floppy. Therefore, most viruses spread using floppy disks and other removable media. Some viruses spread by infecting programs stored on these disks, while others installed themselves into the disk boot sector, ensuring that they would be run when the user booted the computer from the disk, usually inadvertently. Until floppy disks fell out of use, this was the most successful infection strategy and boot sector viruses were the most common in the wild for many years.

Traditional computer viruses emerged in the 1980s, driven by the spread of personal computers and the resultant increase in BBS, modem use, and software sharing. Bulletin board-driven software sharing contributed directly to the spread of Trojan horse programs, and viruses were written to infect popularly traded software. Shareware and bootleg software were equally common vectors for viruses on BBS's. Within the "pirate scene" of hobbyists trading

illicit copies of retail software, traders in a hurry to obtain the latest applications were easy targets for viruses.

Macro viruses were introduced in the mid-1990s. Most of these viruses are written in scripting languages to infect Microsoft programs such as Word and Excel. Since Word and Excel were also available for Mac OS, most could also spread to Macintosh computers. Some old versions of Microsoft Word allow macros to replicate themselves with additional blank lines. If two macro viruses simultaneously infect a document, the combination of the two, if also self-replicating, can appear as a "mating" of the two and would likely be detected as a virus unique from the "parents."

Virus can also be spread though instant message by sending a web address link. If the recipient think that it is from trusted source, he/she will follow the link. The virus hosted at the link can able to infect the computer and continue propagating.

Cross-site scripting viruses emerged recently, and were academically demonstrated in 2005. Since 2005 there have been multiple instances of the cross-site scripting viruses in the wild, exploiting websites such as MySpace and Yahoo.

INFECTION STRATEGIES

Virus must have permission for execution of code and written to memory to replicate itself. Therefore, viruses attached with executable files and if the user executes the infected file, the virus code will execute simultaneously.
Viruses can be divided into two types based on their behavior when they are executed.

NONRESIDENT VIRUSES

These viruses search for other hosts or applications to spread infection, infect those target hosts and then transfers control to the application they had infected. It can be seen as the combination of the finder module and the replication module. Finder module finds the target hosts which further calls the replication module to infect that file. For each new executable file, finder module is encountered.

RESIDENT VIRUSES

Rather than searching for new hosts immediately, resident virus loads itself into memory on execution and transfers control to the host program. The virus stays active in the background and infects new hosts as they are accessed.

Resident viruses has replication module similar to the one used by non-resident viruses but does not contain the finder module. As the virus has been executed, it loads the replication module into the memory and called each time a new operation is executed by the operating system.

Resident viruses can be divided into two categories fast infectors and slow infectors.

Fast infectors can infect as many files as possible. For instance, a fast infector can infect every potential host file that is accessed. It might create a problem while using anti-virus software, since the virus scanner will scan all the potential host file while performing system-scan and if the scanner fails to find such virus, the virus can "piggy-back" on the scanner and can infect all the files that are scanned. Infect5ing too much files becomes t5he disadvantage of fast infectors as such infections can detect more easily because of slow performance of computer or any other suspicious action detect by the anti-virus software.

On the other hand, slow infectors infect the hosts infrequently. For instance, slow infectors infect files only when they are copied. Slow infectors are designed to avoid detection by limiting their actions and cannot be easily triggered by the anti-virus software that detects suspicious behavior of the programs. However, this approach does not seem vey successful.

CROSS-PLATFORM VIRUSES

With the popularity of cross-platform applications, cross-platform viruses are identified in 2007. This was brought to the forefront of malware awareness by the distribution of an Openoffice.org virus called *Bad Bunny*.

As per the statement of Stuart Smith of Symantec, "What makes this virus worth mentioning is that it illustrates how easily scripting platforms, extensibility, plug-ins, ActiveX, etc, can be abused. All too often, this is forgotten in the pursuit to match features with another vendor. The ability for malware to survive in a cross-platform, cross-application environment has particular relevance as more and more malware is pushed out via Web sites. How long until someone uses something like this to drop a JavaScript infector on a Web server, regardless of platform?"

ABOUT VECTORS AND HOSTS

Viruses have targeted various types of transmission media or hosts. This list is not exhaustive:

- Binary executable files (such as COM files and EXE files in MS-DOS, Portable Executable files in Microsoft Windows, and ELF files in Linux)
- Volume Boot Records of floppy disks and hard disk partitions
- The master boot record (MBR) of a hard disk
- General-purpose script files (such as batch files in MS-DOS and Microsoft Windows, VBScript files, and shell script files on Unix-like platforms).
- Application-specific script files (such as Telix-scripts)
- System specific autorun script files (such as Autorun.inf file needed to Windows to automatically run software stored on USB Memory Storage Devices).
- Documents that can contain macros (such as Microsoft Word documents, Microsoft Excel spreadsheets, AmiPro documents, and Microsoft Access database files)
- Cross-site scripting vulnerabilities in web applications

- Arbitrary computer files. An exploitable buffer overflow, format string, race condition or other exploitable bug in a program which reads the file could be used to trigger the execution of code hidden within it. Most bugs of this type can be made more difficult to exploit in computer architectures with protection features such as an execute disable bit and/or address space layout randomization.

Malicious code can be embedded in the PDFs or in HTML code. Operating systems use file extensions to determine program association. These extensions may be hidden from the user by default. For example, an executable may be created named "picture.png.exe", in which the user sees only "picture.png" and therefore assumes that this file is an image and most likely is safe.

An additional method is to generate the virus code from parts of existing operating system files by using the CRC16/CRC32 data. The initial code can be quite small (tens of bytes) and unpack a fairly large virus. This is analogous to a biological "prion" in the way it works but is vulnerable to signature based detection.

TRICKS OF VIRUS TO AVOID ITS DETECTION

Many approaches are used to avoid detection of virus by users. One oldest approach is, if the file is infected, “last-modified” date of the host file remains same. This approach is especially used in MS-DOS platform. However, this approach does not fool anti-virus software, especially those which maintains date and Cyclic redundancy checks on file changes.

Another approach to avoid detection is: viruses can infect files without increasing their sizes or damaging the files. This can be done by overwriting unused areas of executable files. These are called cavity viruses. For example the CIH virus, or Chernobyl Virus, infects Portable Executable files. As these files have many empty gaps, the virus, which was 1 KB in length, did not add to the size of the file. Some viruses try to avoid detection by killing the tasks associated with antivirus software before it can detect them.

Old hiding techniques need to be replaced or updated as computers and operating systems are growing and becoming complex. File systems may need detailed and explicit permissions of every kind of file access to prevent the computer against viruses.

AVOIDING BAIT FILES AND OTHER UNDESIRABLE HOSTS

Virus needs to infect the host files to spread further. However, infecting the host files may lead to the detection of virus more easily as much anti-virus software performs an integrity check for their own code. For this reason, some viruses are programmed not to infect programs that are known to be part of anti-virus software.

Another host files that virus needs to avoid are Bait files (or goat files). Bait files are designed by the anti-virus

professionals to be infected by virus which helps to detect the virus.

As Bait files are designed to infect themselves by the virus, these files can be used by the anti-virus professionals to find different samples of virus Professionals use these samples to study the behavior of the virus and evaluate detection methods for them. It is more practical to store and exchange a small, infected bait file, than to exchange a large application program that has been infected by the virus. Bait files are especially useful when the virus is polymorphic. In this case, the virus can be made to infect a large number of bait files. The infected files can be used to test whether a virus scanner detects all versions of the virus.

Some Bait files accessed regularly. If any modification finds in these files, the anti-virus software warns the user that virus may be active on the system. Hence, virus needs to avoid such files. This can be done by avoiding the small program files or programs that contain certain patterns of garbage instructions.

Another strategy to avoid Bait files is sparse infection. Sometimes, sparse infectors do not infect a host file that would be a suitable candidate for infection in other circumstances. For instance, virus may decide whether to infect the file or not, or it may infect the host files on a particular day of week.

STEALTH

STEALTH is a technique used by virus to befool anti-virus software by intercepting the request to the operating system. As anti-virus software requests to read a file, virus intercepts it and receives the request. Thus, the request is passed to the virus rather than to the operating system. The virus will then return the uninfected version of file which seems clean to the anti-virus software. Many techniques are used to avoid stealth but the most reliable technique is to boot from medium which is known to be clean.

SELF-MODIFICATION

Virus can be easily find using virus signatures while scanning programs through anti-virus software. A signature is a characteristic byte-pattern that is part of a certain virus or family of viruses. If the scanner finds such pattern, it notifies the user that the file is infected. Then it is up to the user whether to delete, clean or heal the file. Some virus makes the detection difficult using signatures as they modify their code at each infection. However, the detection of virus through signatures is not the impossible task.

ENCRYPTION WITH A VARIABLE KEY

Virus can also be spread using encryption. For this, virus needs decrypted module and the encrypted module. As the virus is encrypted using different keys for each new file which makes the detection of virus difficult. However, the decryption module remains same through which the indirect detection of virus could be possible. Since these would be symmetric keys, stored on the infected host, it is

in fact entirely possible to decrypt the final virus, but this is probably not required, since self-modifying code is such a rarity that it may be reason for virus scanners to at least flag the file as suspicious.

An old method used for encryption is XORing each byte of the virus program with a constant and the same XOR operation will repeated for decryption. It is suspicious code that modifies itself, so the code to do the encryption/decryption may be part of the signature in many virus definitions.

POLYMORPHIC CODE

Polymorphic code uses the concept of encryption to infect files. However, in encryption, decrypted module remains same whereas in polymorphic code, decryption module is also modified on every new infection. It is a serious threat to virus scanner as there is no identical part between infections which makes the detection of virus too difficult.

Anti-virus software can detect such viruses using an emulator or by statistical pattern analysis of the encrypted virus body. To generate polymorphic code, virus needs to have a polymorphic engine (also called mutating engine or mutation engine) somewhere in its encrypted body

Such slow polymorphic code makes it more difficult for anti-virus professionals to obtain the samples of virus. Polymorphic code makes the detection by virus scanner unreliable and also helps to avoid detection even through Bait files which infect themselves in only one run and contains similar or identical samples of virus.

METAMORPHIC CODE

Polymorphic code can be detected using emulation. To avoid this detection, metamorphic code is used. Using this technique, virus rewrites themselves completely on the infection of any new executable file. To enable metamorphism, a metamorphic engine is needed. A metamorphic virus is usually very large and complex. For example, W32/Simile consisted of over 14000 lines of Assembly language code, 90% of which is part of the metamorphic engine.

LINUX VULNERABILITY

Linux supports multi-user environment where users require privileges to access which is implemented using some access control technique. To cause any serious consequence over Linux, malware needs to have the root access to the system.

Shane Coursen, a senior technical consultant with Kaspersky Lab noted, “The growth in Linux malware is simply due to its increasing popularity, particularly as a desktop operating system. The use of an operating system is directly correlated to the interest by the malware writers to develop malware for that OS.”

SecurityFocus's Scott Granneman stated, some Linux machines definitely need anti-virus software. For instance,

Samba or NFS servers, may store documents in undocumented, vulnerable Microsoft formats, such as Word and Excel which may propagate viruses.

Linux mail servers send mails to other computers which are using different operating systems. Therefore, Linux operating system also needs to run AV software to detect viruses before they show up in the mailboxes of Outlook and Outlook Express users. For example the open source ClamAV "Detects viruses, worms and trojans, including Microsoft Office macro viruses, mobile malware, and other threats." Hence, Linux virus scanners search for all known viruses for all computer platforms.

VULNERABILITY AND COUNTERMEASURES

THE VULNERABILITY OF OPERATING SYSTEMS TO VIRUSES

Just as genetic diversity in a population decreases the chance of a single disease wiping out a population, the diversity of software systems on a network similarly limits the destructive potential of viruses.

This became a particular concern in the 1990s, when Microsoft gained market dominance in desktop operating systems and office suites. The users of Microsoft software (especially networking software such as Microsoft Outlook and Internet Explorer) are especially vulnerable to the spread of viruses.

Microsoft gained market dominance because of its desktop operating system and office suites in 1990s. Hence, the Windows become the most popular OS for virus writers and are often criticized for including many errors and holes for virus writers to exploit. Integrated and non-integrated Microsoft applications (such as Microsoft Office) and applications with scripting languages with access to the file system (for example Visual Basic Script (VBS), and applications with networking features) are also particularly vulnerable.

Windows is the most popular OS among virus writers; however, some viruses also exist for other operating systems. Operating system that allows third-party programs to run over it can affect from virus. Unix-based Operating systems are more secure as they provide the facility to run executable code into its own protected memory space.

Mac OS X (with a Unix-based file system and kernel) is considered better OS than MS-Windows as MAC OS X has relatively few security exploits. One older version of Apple OS named “Mac OS Classic” states that there are only 4 known viruses and independent sources states that t5here are as many as 63 viruses. Virus vulnerability between Macs and Windows is a chief selling point, one that Apple uses in their Get Mac advertising.

As the first virus for Linux named “Bliss” has been released, anti-virus vendors issued a warning that Unix-like systems could fall prey to viruses just like Windows. Bliss needs to run it explicitly and can harm only the files

which the users have access permission to modify. Unlike Windows OS, Linux and UNIX blocks normal users access to make changes to the environment and users do not usually log in as an administrator which can save the OS to get infected.

THE ROLE OF SOFTWARE DEVELOPMENT

Because software is often designed with security features to prevent unauthorized use of system resources, many viruses must exploit software bugs in a system or application to spread. Software development strategies that produce large numbers of bugs will generally also produce potential exploits.

ANTI-VIRUS SOFTWARE AND OTHER PREVENTIVE MEASURES

Anti-virus software's are used to detect and eliminate the known viruses after the computer downloads or runs the executable.

Anti-virus software application uses two common methods to detect viruses. The first common method of virus detection is using a list of virus signature definitions. This can be done by examining the content of the computer's memory (its RAM, and boot sectors) and the files stored on fixed or removable drives (hard drives, floppy drives), and comparing those files against a database of known virus "signatures".

The disadvantage of this detection method is that users are only protected from viruses that pre-date their last virus definition update. The second method is to use a heuristic algorithm to find viruses based on common behaviors. This method has the ability to detect viruses that anti-virus security firms have yet to create a signature for.

Some anti-virus software's uses "on-access scanning" means scanning is performed as and when the file is opened and even while sending and receiving e-mails. Anti-virus software does not change the underlying capability of host software to transmit viruses. Users must update their software regularly to patch security holes.

Anti-virus software also needs to be regularly updated in order to prevent the latest threats.

Damages caused by viruses could be minimized by taking the regular back-ups of data either on devices which kept unconnected to the system (most of the time), read-only or not accessible for other reasons, such as using different file systems. This way, if data is lost through a virus, one can start again using the backup (which should preferably be recent).

Optical media such as CD/DVD stores data in read-only format. Therefore, the data cannot be affected by virus on such devices. Hence, if the OS becomes unusable, an OS on a bootable CD can be used to start the system.

Backups on removable media must be carefully inspected before restoration. The Gammima virus, for example, propagates via removable flash drives.

RECOVERY METHODS

Once the computer gets infected by virus, it is unsafe to use the infected system without reinstalling the operating system. However, there are number of recovery options available while the actions depend upon the type of virus.

VIRUS REMOVAL

There is a tool available on Windows Me, Windows XP, and Windows Vista named "System Restore" which restores the registry and critical system files to a previous checkpoint. A virus may hang the system and a subsequent hard reboot corrupt the system restores point on the same day. Restore point from previous days works only if the virus is not designed to corrupt restore files.

Some viruses such as CiaDoor disable system restore and other tools such as Task Manager and Command Prompt. Administrators can disable such tools to access it by other users. However, a virus can block all users to access these tools by modifying the registry. When an infected tool activates it gives message "Task Manager has been disabled by your administrator.", even if the user trying to open the program is the administrator.

OPERATING SYSTEM REINSTALLATION

Another approach for virus removal is reinstallation of operating system. This is done by formatting the OS partition and install OS using its original media. This approach is faster than using antivirus software and scans the system multiple times. However, it includes the overhead of reinstallation of all other software and drivers

VIRUSES, WORMS AND OPERATING SYSTEMS

VIRUSES, TROJAN HORSES AND LINUX SYSTEMS

Some viruses may threat to Linux systems. Execution of infected binary may infect the system. However, the infection level depends upon the privileges of user which executes the infected binary. Binary file run under root account may infect the entire system. Privilege escalation vulnerabilities may permit malware running under a limited account to infect the entire system.

Virus generators do not require any special malware writing skills. They can simply add a code snippet to any program and as the user downloads that program, it will download through the modified login server. This additional code run anytime, the user logs in. however, special skill may be needed for tricking the user to run the program in the first place.

Threat of installation of malware can be reduced using software repositories. Software repositories are checked by maintainers to ensure that the software is malware-free. For this purpose, md5 checksums are used. Through this, modified versions are identified that may be introduced by different malware attacks. It limits the scope of attacks by only including the original authors, package and release maintainers and possibly others with suitable administrative access, depending on how the keys and checksums are handled.

If the user executes the code which is not from trusted user, vulnerability of Trojan horses and viruses may cause. It is also the fault of distributers which do not provide the default checking for authenticity of software downloaded.

WORMS AND UNIX-LIKE SYSTEMS

UNIX systems have vulnerability in network daemons such as and WWW servers can be used or attacks. Server takes immediate action against vulnerabilities. There is no guarantee on the installation if attack is on targets which are not publicly known. Servers having weak passwords can also be attacked.

WWW SCRIPTS AND LINUX SERVERS

Rather than attacking the system, Linux servers can also be used by malwares. E.g. WWW content and scripts are restricted as it may be used by malware to attack visitors.

POTENTIAL THREATS

New malwares are introduced and increasing day by day to cause threat to the system Some of them are given as:

TROJANS

Kaiten	**Linux.Backdoor.Kaiten trojan horse**
Rexob	Linux.Backdoor.Rexob trojan

VIRUSES

- ✓ **Alaeda - Virus.Linux.Alaeda**
- ✓ Bad Bunny - Perl.Badbunny
- ✓ Binom - Linux/Binom
- ✓ Bliss
- ✓ Brundle
- ✓ Bukowski
- ✓ Diesel - Virus.Linux.Diesel.962
- ✓ Kagob a - Virus.Linux.Kagob.a
- ✓ Kagob b - Virus.Linux.Kagob.b
- ✓ MetaPHOR (also known as Simile)
- ✓ Nuxbee - Virus.Linux.Nuxbee.1403
- ✓ OSF.8759
- ✓ Podloso - Linux.Podloso (The iPod virus)
- ✓ Rike - Virus.Linux.Rike.1627
- ✓ RST - Virus.Linux.RST.a
- ✓ Satyr - Virus.Linux.Satyr.a
- ✓ Staog
- ✓ Vit - Virus.Linux.Vit.4096
- ✓ Winter - Virus.Linux.Winter.341
- ✓ Winux (also known as Lindose and PEElf)
- ✓ Wit virus
- ✓ ZipWorm - Virus.Linux.ZipWorm

WORMS

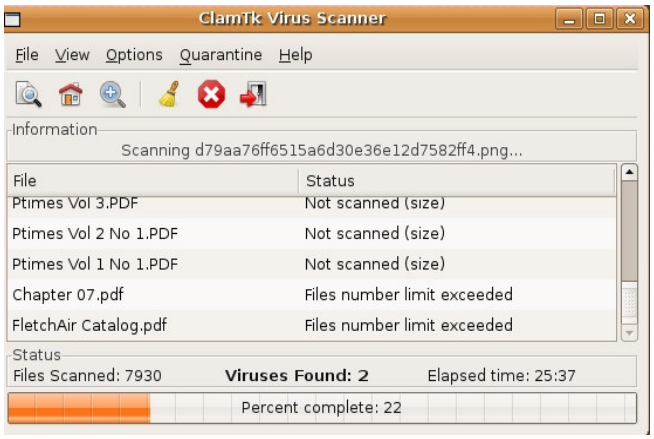
- ✓ **Adm - Net-Worm.Linux.Adm**
- ✓ Adore
- ✓ Cheese - Net-Worm.Linux.Cheese
- ✓ Devnull
- ✓ Kork
- ✓ Linux/Lion
- ✓ Mighty - Net-Worm.Linux.Mighty
- ✓ Millen - Linux.Millen.Worm
- ✓ Ramen worm

- ✓ Slapper
- ✓ SSH Bruteforce

SOME ANTI-VIRUS APPLICATIONS

There is a number of anti-virus applications available are including:

- ✓ **Avast! (freeware and commercial versions)**
- ✓ AVG (freeware and commercial versions)
- ✓ Avira (freeware and commercial)
- ✓ Bitdefender (freeware and commercial versions)
- ✓ ClamAV (free open source software)
- ✓ Eset (commercial versions)
- ✓ F-Secure Linux (commercial)
- ✓ Kaspersky Linux Security (commercial)
- ✓ McAfee VirusScan Enterprise for Linux (commercial)
- ✓ Panda Security for Linux (commercial version)
- ✓ Sophos (commercial)
- ✓ Symantec AntiVirus for Linux (commercial)
- ✓ Trend Micro ServerProtect for Linux (commercial)



CONCLUSION

As prevention is better than cure, one should take all the preventive measures to safeguard computer against virus threats. Though viruses are dangerous, but there is no need to panic. The name virus itself seems like they can destroy your computer any moment. But it is not the only truth. One can take security measures to protect computer against malicious code using update antivirus and by knowing about the extent of damage and recovery procedures against viruses. The final word of wisdom will be to avoid access of any untrustworthy sources of data whether CD, Pen Drive or online data, and keep your virus scanner updated always.

REFERENCES

i. http://www.bartleby.com/61/97/C0539700.html 5-Nov-2009
ii. http://www.actlab.utexas.edu/~aviva/compsec/virus/whatis.html 7-Nov-2009
iii. "Virus list". http://www.viruslist.com/en/viruses/encyclopedia?chapter=153310937. Retrieved on 2008-02-07. 11-Nov-2009
iv. Thomas Chen, Jean-Marc Robert (2004). "The Evolution of Viruses and Worms". http://vx.netlux.org/lib/atc01.html. Retrieved on 2009-02-16. 1-Nov-2009
v. See page 86 of *Computer Security Basics* by Deborah Russell and G. T. Gangemi. O'Reilly, 1991. ISBN 0937175714. 4-Nov-2009
vi. Anick Jesdanun. "Prank starts 25 years of security woes". http://news.yahoo.com/s/ap/20070831/ap_on_hi_te/computer_virus_anniversary;_ylt=A9G_R3Ga1NhGH0QBIwZk24cA."The anniversary of a nuisance". http://www.cnn.com/2007/TECH/09/03/computer.virus.ap/. 8-Nov-2009
vii. Boot sector virus repair
viii. Dr. Solomon's Virus Encyclopedia, 1995, ISBN 1897661002, Abstract at http://vx.netlux.org/lib/aas10.html 9-Nov-2009
ix. Vesselin Bontchev. "Macro Virus Identification Problems". *FRISK Software International*. http://www.people.frisk-software.com/~bontchev/papers/macidpro.html.
x. Wade Alcorn. "The Cross-site Scripting Virus". http://www.bindshell.net/papers/xssv/.
xi. http://www.pcsecurityalert.com/pcsecurityalert-articles/what-is-a-computer-virus.htm
xii. http://www.virusbtn.com/resources/glossary/polymorphic_virus.xml 5-Nov-2009
xiii. Perriot, Fredrick; Peter Ferrie and Peter Szor (May 2002). "Striking Similarities" (PDF). http://securityresponse.symantec.com/avcenter/reference/simile.pdf. Retrieved on September 9, 2007. 6-Nov-2009
xiv. http://www.virusbtn.com/resources/glossary/metamorphic_virus.xml. 3-Nov-2009
xv. Need a computer virus?- download now. 2-Nov-2009
xvi. http://blog.didierstevens.com/2007/05/07/is-your-pc-virus-free-get-it-infected-here/
xvii. "Malware Evolution: Mac OS X Vulnerabilities 2005-2006". Kaspersky Lab. 2006-07-24. http://www.viruslist.com/en/analysis?pubid=191968025. Retrieved on August 19, 2006. 4-Nov-2009
xviii. Apple - Get a Mac. 7-Nov-2009

xix. Sutter, John D. (22 April 2009). "Experts: Malicious program targets Macs". *CNN.com*. http://www.cnn.com/2009/TECH/04/22/first.mac.botnet/index.html. Retrieved on 24 April 2009. 9-Nov-2009
xx. McAfee. "McAfee discovers first Linux virus". *news article*. http://math-www.uni-paderborn.de/~axel/bliss/mcafee_press.html. 3-Nov-2009
xxi. Axel Boldt. "Bliss, a Linux "virus"". *news article*. http://math-www.uni-paderborn.de/~axel/bliss/. 2-Nov-2009
xxii. "Symantec Security Summary — W32.Gammima.AG." http://www.symantec.com/security_response/writeup.jsp?docid=2007-082706-1742-99
xxiii. "Yahoo Tech: Viruses! In! Space!" http://tech.yahoo.com/blogs/null/103826
xxiv. "Symantec Security Summary — W32.Gammima.AG and removal details." http://www.symantec.com/security_response/writeup.jsp?docid=2007-082706-1742-99&tabid=3. 2-Nov-2009