IJCES International Journal of Computer Engineering Science , Volume1 Issue 3, December 2011
ISSN : 2250:3439
https://sites.google.com/site/ijcesjournal
http://www.ijces.com/

Computer Viruses in UNIX Environment: Case Study

Asmaa Shaker Ashoor
Computer Science Department
Pune University-pune
asmaa_zaid218@yahoo.com

Prof. Sharad Gore
Statistic Department
Pune University-pune
sdgore@stats.unipune.ernet.in

Prof. Vilas Kharat
Computer Science Department
Pune University-pune
hod_cs@cs.unipune.ac.in

Abstract. All of people who don't know how to use a computer have heard a bout viruses through programs such as hackers and some means like that. There is no doubt that our culture is fascinated by the potential danger of these viruses. Computer virus have become threat to computer users and almost every field in the advance technology industrial nowadays. Know about virus is very necessary for anti-virus researchers as well as operating systems makers. With the development of the open source systems today, computer viruses on these systems should be considered strictly. The goal of this paper is to present my concept of classification virus computer in UNIX environment. This paper provides some subjective comments on some of the most widely known environment and some methods available to protect UNIX today. propose some viruses that can work on this environment and suggest some methods to prevent as well as restrain damages of these viruses.

Keywords: Unix, Viruses, Computer security, Technologies

1 Introduction

The term computer virus as a program that can infect other programs by modifying them to include a possibly evolved copy of itself. With the infection property, a virus can spread throughout a computer system or network using the authorizations of every user using it to infect their programs. Every program that gets infected may also act as a virus and thus the infection grows [1].

1.1 Parts Of Computer Viruses

A computer virus consists of three parts[2]

- The infection mechanism
- The trigger
- The payload

As mentioned above, a computer virus must at least have the infection mechanism part.

1.1.1 The Infection Mechanism

Searches for one or more suitable victims and checks to avoid multiple infections if the host is already infected or (not every virus does this[3]; some viruses infect a host multiple times due to bugs). After that, the virus body is copied into the victim. The easiest method to do so is (by) over writing the code of the victim. Other methods are putting the code in front of or at the end of a file.

1.1.2 The Trigger

A trigger is used for starting the possible payload[3], i.e. on a particular event, the payload is executed. Such an event could a special day or when the infection counter has reached a pre-defined value.

1.1.3The Payload

A possible payload causes transient or permanent damage e.g. displaying an animation on the screen or formatting the hard disk drive or manipulation of data[3]. Damage may even happen unintentionally, e.g. due to a programming error or if an old DOS virus causes trouble within the windows environment. Damage may be caused by over-reaction the user, too[4].

1.2 Classification Of Computer Viruses

The classification of computer viruses can be done via several ways[2]:

- Type of host victim
- Type of infection technique
- Special virus features

1.2.1 Type Of Host Victim

We can distinguish between:

- Boot (DBR) sector and master boot record (MBR) virus
- File virus
- Companion virus
- Multipartite virus

- **A boot virus**: infects the boot sector of a floppy disc and / or master boot record or boot sector of a hard disc. Such a virus can infect the computer system, when the computer is booted from an infected floppy disc. As the code in the MBR/DBR is started by the BIOS after it does the POST(Power On Self Test) the virus gets activated even before the operation system has

been started and most likely" hooks" some particular, interrupts for performing its tasks. Most boot sector viruses are memory-resident, so they can easily infected every non-write protected floppy when it is accessed. Most viruses of this type save a copy of the original boot sector/master boot record in an unused sector of the disc. A boot virus may be "placed" into the computer system by a so-called "dropper".

- **A file virus**: infects(executable)files, either by overwriting the file (overwriting virus) or by appending the virus code at the beginning or end of the file(appending virus). An overwriting virus destroys the original file upon infection. Most appending viruses put their virus code at the end of the file and put a jump to the virus code at the beginning of the file, so than the virus code is started first upon execution.
- **A companion virus**: if a program with the extension .BAT or EXE and then creates a .COM file with the same name (i.e. TETRIS.COM, if a program TETRIS.EXE exists). If only the program name is entered(here: TETRIS), DOS per default looks up first for a matching.COM.EXE and then .BAT file. So , TETRIS.COM will be started (instead of TETRIS.EXE, which was originally the intention of the user). Therefore the companion virus is started first and can then start TETRIS.EXE[5].
- **A multipartite or hybrid virus**: uses more than one infection technique, e.g. a combination of a boot sector and file virus and therefore infects DBR/MBR and files. Or viruses which infect office files via Visual Basic for Applications (VBA) and Visual Basic Script(VBS) files; or viruses which infect Win32 files and office files. The basic infection technique of file viruses for Windows systems are somewhat similar to DOS viruses, but more complicated as the file format is more complex, too[6]. This applies basically to Linux viruses[7]. Even viruses for both platforms are possible.

1.2.2 Type Of Infection Technique

The technique can be distinguished between[2]:

- **A direct action virus**: does not stay memory so it is only active when an infected program has been started and only by this event it can replicate. A direct action virus is not very complex and can therefore be very small. In most cases a direct action virus does not spread as fast as a memory resident virus[2].
- **A memory resident virus:** installs itself into RAM and may be active as long as the computer is running. This can be achieved via several ways, depending on the operation system: DOS provides a mechanism called "terminate-and-stay-resident" (TSR), for windows as a "virtual device driver" (VxD),for windows NT as an NT-service, for Linux as a loadable kernel module. Only a memory resident virus may use some "modern" virus techniques like stealth capabilities. For the memory resident virus, one can

differentiate between a **fast infector** [5] and **slow infector** [2][3]. Both got their name due to the speed they spread. The first one infects every program which is being accessed(read/write) or even all files being executed).

1.2.3 Special virus features

The following special virus features will be explained briefly[2]:

- Stealth technique
- Retro capabilities
- Polymorphism

Some special virus features can only be used by memory-resident viruses.

- **A stealth virus:** tries to hide itself by hooking several interrupts like BIOS Int 13h or DOS Int 21h[2][3][5]. Assumed, an anti-virus program reads the MBR via BIOS Int 13h to scan for viruses, the virus can intercept this and "redirect" the read call to the saved copy of the original, uninfected MBR. Therefore, the anti-virus program will not find any virus. Or, if a virus scanner scan a file, this file must be opened first. The open call, "redefined" by the virus, will first remove the virus from the file and then call original open call. After the scanning of the file has been finished, the file will be closed by the virus scanner. And the modified close call will infect the file again.
- **A retro virus;** avoids to infect particular file names, like "scan.exe" or "f-prot.exe" as most anti-virus software checks their integrity upon start[14]. This mechanism can be used by non-memory resident viruses, too. A resident virus may even intercept the execution of "scan.exe" and display a "faked" error message like "not enough memory".

A polymorphic virus: is being "encrypted" and changes infection its shape and structure of the de/encryption routine by each infection but the basic functionality is always the same[2][3][5][9]. Here is vary easy example to simply to get the basic idea how it works: a CPU has a set of registers e.g. the accumulator register AX. This can be done by setting the register to zero. Or subtracting the current value of the AX register with itself. In short, the effect is just the same, but each operation will result in a different opcode. This technique is also known as "mutation".

2 Anti-virus approaches

The ideal solution to the threat of viruses is prevention[10]: do not allow a virus to get in to the system in the first place. This goal is, in general, impossible to achieve, although prevention can reduce the number of successful viral attacks. The next best approach is to be able to do the following:

- Detection: once the infection has occurred, determine that it has occurred and locate the virus.
- Identification: once detection has been achieved. identify the specific virus that has infected a program.
- Remove the virus from all infected systems, so that the disease cannot spread further.
- Removal: once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state.

2.1 Anti-virus technologies

2.1.1 Scanner

virus scanner are the far most used method detect(and clean) a virus[11]. It may either work on-demand (i.e. the user has to start the virus scanner) or on-access, which means the program runs in the background and scans a file while it is being accessed. Several virus detection methods are possible, and are used may depend of the type of the virus and /or file type:

- **Pattern matching**: for each known virus , a particular sequence of code is "extracted", mostly called pattern, signature or search string, and stored in a virus-definition file (some kind of database). Therefore, the code as scanner " is looking for an exact match which will identify the code as a virus. To detect, variants or minor modifications of a virus, a search string may contain wildcards. Not only the search string is stored, but also information which file types can by infected by this particular virus and at which byte position/offset the search string may occur. This is used to speed up the virus scan process and to avoid false positive. moreover , the virus definition file or some pseudo-code for performing various scanning tasks[11]. For identifying a virus, more than one signature could be used; once again, to reduce the likeliness of a false positive[12]. Inexact identification is also important for cleaning a virus, otherwise it may happen that the cleaning process removes not only the virus parts. The basic advantage of pattern matching is that the virus can be named, whereas heuristics may only report " file looks suspicious" back to the user. The basic disadvantage obviously is that only known viruses can be detected i.e. the signature has been added to the virus definition file[12].
- **Heuristics:** heuristics is used to "detect" new viruses. The heuristics approach a program is "analysed" for instructions (or set of instructions) which are known as typical for viruses. Each of such suspicious instruction is given a special weight, which is summed up. If the sum exceeds a particular threshold, the file is regarded as suspect of infection. Another approach is a rule-based system, which "simply compares" found functionality with a set

of rules. If a predefined rule is found within the code, the rule-based system returns with a positive result [13].

There are two different ways of applying heuristic rules**:** static and dynamic. The static method checks the presence of suspicious code fragments. The dynamic method emulates the program and checks which actions are really performed(that is simulation of a virus execution in a virtual environment, frequently called a sandbox or an emulator buffer). Those methods are sometimes also referred as "passive" and "active" approach. Both can be combined, too[12].

- **Code emulation:** code emulation was originally developed to detect polymorphic viruses[11][12]. So, if a program is being scanned by the anti-virus program, this program is being executed in a virtual environment. Therefore, "when a scanner loads a file infected by a polymorphic virus into this virtual computer, the virus decryption routine executes and decrypts the encrypted virus body". The exposes the virus body to the scanner, which can then search for signatures in the virus body that precisely identify the virus strain[9]. As mentioned above, code emulation is also used together with heuristics. Code emulation is slow so code emulation should only be used when really needed[12].

2.1.2 Integrity Checker

An integrity checker basically generates a checksum for files, sectors(i.e. boot sector) and the macros, stored in e.g. an office document. The checksums and being stored in a kind of database and later being compared. If a checksum does not match, a file has been modified (which could be caused due to a virus infection). Obviously, when generating the checksum, it must be assured the file is clean[12][14].

2.1.3 Behaviour Blocker

A behaviour blocker runs in the background and monitors the execution of the currently running programs on the computer[12]. If a program tries to do a suspicious action(e.g. open a file and appending code or formatting hard disc), this will be intercepted. The behaviour blocker may then terminated this program or ask the user which action should be taken(e.g. allow, do not allow move program into quarantine). But for most users this decision is a "tough choice" and behaviour blocking may generate a high level of false positives, although some techniques are possible to reduce the likelihood of false positives. Although some techniques are possible to reduce the likehood of false positives.

2.2 Anti-Virus Strategy

Nowadays, a basic anti-virus strategy is a 3-tier approach[16]:

1. The desktop

2. File& print, email or web servers
3. The internet gateway, like mail gateways or web proxy servers.

A virus should be stopped as possible, before it can enter the network(tier3). According to since 2000 more than 80% of the virus incidents have been caused by infected email attachments whereas diskettes as source of infection are next to nothing. Encrypted emails/ attachments can not be checked at this level. Virus scanning require lots of resources, so this task should probably be off-loaded onto another machine. Anti-virus software on the gateway must take precautions to not suffer from a denial-of-service attack by special crafted mails and/ or mail attachment. As files, documents are shared via file servers, those are a vector for distributing infected documents (some viruses /worms use network shares to propagate itself). Therefore, on-access scanning of file servers is the next line of defense(tier2). The last resort is the desktop, (e.g. for scanning an encrypted file when it is being decrypted.

3 UNIX Operating System

The UNIX system was originally developed by expert programmers for their own use[10][15]. Speed and accuracy are not normally the favored needs of beginners and as a result there was a general opinion that UNIX operating system was more programmer friendly. This was compounded by the earlier lack of proper documentation, the smooth/concise syntax and the complexity of administering the system[10]. Unix operating system is a powerful and complex one, it has become more regular, controllable and user friendly. It is a tribute to UNIX that so many people have found it easily adaptable to their needs. Its user interfaces, though imperfect, can be replaced, there is every reason to believe that UNIX will continue to be used far more widely.

3.1 UNIX Features

Unix is a comprehensive operating system with a number of features and capabilities. Its major features are[10][15]:

- Multi-user, Time-sharing OS
- Multi-tasting OS
- Portability
- Modularity
- System security
- File structure and security
- I/O independence
- I/O redirection and piping
- Communication

3.2 Unix vulnerabilities

There are 10 UNIX vulnerabilities[17]:

1. Remote Procedure
2. Apache Web Server
3. Secure Shell (SSH)
4. Simple Network Management

Protocol (SNMP)

5. File Transfer Protocol (FTP)
6. R-Services-----trust relationships
7. Line Printer Daemon (LPD)
8. Send mail
9. BIND/DNS
10. General UNIX authentication-----Accounts with No Passwords or Weak Passwords

3.3 Viruses On UNIX Operating System

virus works by replicating inside programs. Each infected program then viruses can be used to spread an attack throughout a system or network. A spreads the virus further. The UNIX protection mechanisms are inadequate for virus defense. Unix has the reputation of being " not so buggy", and of being a good maintainer of system sanctity via good protection mechanisms. A few years ago tom duff created a very persistent UNIX virus. the virus lived in the slack space at the end of the executable, and changed the entry point to itself. When the program was executed, it searched the current directory, subdirectories , /bin/usr/bin for writable, uninfected files and then infected them if there was enough space. A channel(or a mechanism) used by virus to spread is called a vector. There is no dearth of potential vectors on UNIX(for example, buffer overflow vulnerabilities).

3.4 How To Hide Viruses On UNIX?

There are several candidates on UNIX for being a virus runtime environment. Similarly, there are several places for a virus to hide on UNIX:

1. **The UNIX shells**

Shell scripts are a powerful way to program. Unix shells are ubiquitous, accessible, and provide homogeneity across otherwise heterogeneous systems(for example, with differing application binary interfaces). Shell scripts are simple text files, and lend themselves easily to be modified.

2. **Binary executables**

A virus writer may want his virus to hide in a binary executable, for obvious reasons(such files provide more obscure hiding places, and are often more" active"). However, given the diverse nature of different UNIX platforms(including different executable formats), modifying an executable might be rather painful to implement.

For example, the feasibility and difficulty of injecting a stream of instructions into an executable to modify program execution would depend on the file format. The executable and linking format(ELF) is meant to provide developers with a set of binary interface definitions that extend across multiple platforms. ELF is indeed used on several platforms, and is flexible enough to be manipulated creatively. A virus could attach viral code to an ELF file, and re-route control-flow so as to include the viral code during execution.

3. Jingle bell: a simple virus in C

Jingle bell is an extremely simple minded virus written in c that attaches itself to an executable by appending the latter to itself and recording the offset. This process repeats itself. The virus infects the first executable found, if any, on its command line. Other infection policies could be programmed too. The virus would some how need to be introduced in the system, through a downloaded binary, for example.

3.5 Detection & Prevention Virus On UNIX

- **Detection:** Viruses can reliably be detected by using an integrity shell instead of the normal UNIX shell. Integrity shells for UNIX have been in use for several years, and work transparently to the normal user.
- **Prevention:** Viruses can not be completely prevented under UNIX or any other modern operating system except by eliminating sharing, or eliminating programming. This is almost never feasible in a modern UNIX system.
- **Cure:** Viruses are best cured with on-line backups which automate the restoration of corrupted information under an integrity sell. Off-line backups are also effective in many cases, as long as good detection is in place. Without good detection, backups are ineffective against viruses.

3.6 Viruses Available On UNIX Environment

1- Viruses are spread in tftp to obtain password files if possible use tfbootd in place of tftp.

2- Programs such as telnet, su and login are being replaced by viruses programs.

3- Viruses have been leaving files and directories with both usual and unusual names such as "mail.", ".. " these files may be found in the home directories of compromised accounts or in /tmp or /usr/tmp.

4- Viruses may be introduced through the introduction of scripts that set the user id to root . than use the "find" command to verify that all such scripts are authorized.

5-The viruses may attempt to leave an additional account on the system to be used at a later time. Therefore, check password file to assure that all accounts are authorized and properly passworded.

5- The viruses may be used terminal on the network to access other hosts on the network.

6-The send mail function has several problems which viruses can exploit.

7- There is also a well-known problem with finger in less recent versions of UNIX. A virus continue to exploit this vulnerability.

4 Conclusion& suggestion

- **Conclusion**

1- A virus attacks specific file types.

2- A virus manipulates a program to execute tasks unintentionally.

3- An infected program produces more viruses.

4- An infected program may run without error for a long time.

5- Viruses can modify themselves and may possibly escape detection this way.

- **Suggestion**

1- The increasing reliance by business on use of data processing systems and the increasing use of networks and communications facilities to build distributed systems have resulted in a strong requirement for computer and network security. computer security relates to mechanisms inside and related to a single computer system. The principal object is to protect the data resources of that system. Network security deals with the protection of data and messages that are communicated. Another important to prevention the virus is access control. The purpose access control is to ensure that only authorized users have access to particular system and its individual resources and that access modification of particular portion of data are limited to authorized individuals and programs. These viruses exploit vulnerabilities in system software either to gain unauthorized access to information or to degrade system service.

References

1. Fred Cohen, Computer Viruses-Theory and Experiments, chapter2-A computerViruses,1984. http://www.all.net/books/virus.
2. Martin Roesler, FAQder Virus.Ger.Version2.3,1995. http://www.vhm.haitac.de/faq.
3. CHIP special, Computer-Virenkl"95 Vogel Verlag,1995.
4. BSI, Informationen zu Computer-Viren, Schriftenreihe zur IT-Sicherheit, Band2, 1994.
5. Rune Skardhamar, Virus detections and elimination, Academic press,Inc,1996.
6. Peter Szor, Attacks on Win32-partII, proceeding of virus Bulletin conference,2000.
7. Marius Van Ores, Linux viruses-ELF file format, proceeding of virus Bulletin conference,2000.
8. Carey Nackenberg, computer parasitology, proceeding of virus Bulletin conference,1999.
9. Carey Nachenberg, understanding and Managing Polymorphic Viruses, The Symantec Enterprise paper, Symantec, 1996.
10. William Stallings, operating system(internals & design principles),third edition, prentices-hall international.

11. Katherine carry, sophos Anti-virus detection a technical over view, sophos, Oxford,UK,2002. http://www.sophos.com/sophos/docs/eng/papers
12. Francisco Fernandez, Heuristic engines, proceeding of virus Bulletin conference, 2001.
13. Markus Schmall, Heuristic Techniques AV solutions: An overview,2002. http://www.securityfocus.com/infocus.
14. Andy Nikishin, Advantages and Dis advantages of modern integrity checkers, proceeding of virus Bulletin conference,1999.
15. A government of India enterprise UNIX O.S. , second Indian edition-1999, published by: et&t corporation, new Delhi.
16. ICSA Labs, Larry Birdwell, ICSA.. Labs.8th Annual Computer Virus Prevalence Survey, ICSA Labs, a division of Trusecure Corp.,2002. 17. John McCormick, Top Linux/UNIX security threats, 2002, http://www.Zdnet.com.au/.