C3IT-2012

Predictability in viral computing

Somsuhabhra Guptaa, Sanjib Dasb, Prasenjit Kanti Palc

aDepartment of Information Technology, JIS College of Engineering, Kalyani-741235, West Bengal, INDIA.
Email: gsomsuhabhra@gmail.com.
bWIPRO Bangalore, INDIA. Email: sanjib.dasos@gmail.com
cDepartment of Electronics and Communication Engineering, JIS College of Engineering, Kalyani, INDIA.
Email: prasenjit.jisce@gmail.com

Abstract

Computers are top-rated essential commodities to the people across the world. The main concern related to it is its vulnerability and information security in the era of internet. Cyber intrusion, infiltration and invasion are a regular phenomenon. There is existing Anti-virus technology to deal with it. It has been observed that most of the Anti-virus is excessively dependent on knowledge base. This means the knowledge of virus signature is an essential need so as to ensure its optimum performance. In the proposed work, effort has been made to study various aspects of the Anti-viral mechanisms. The investigation includes survey of structure and behavior of the Anti-virus. At end a case study is provided to deal with common virus and to get read of them manually without the aid of the Anti-virus.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of C3IT
Open access under CC BY-NC-ND license.
Keywords: Malware; Rootkit; Trojan Horse; Virus; Worm;

1. Introduction

It’s a rare phenomenon to the wide range of computer users ranging from beginner to the level of advanced user who never encountered or affected by Virus problem. Consumer protection in the form of anti-virus software is available in both the form of general types as well as trivial ones like anti-malware software from attack of virus. The intrusion to users space i.e. the methods of attacks appears in various forms viz. in to a system like remote monitoring by another system, corrupting essential system file and user file, data transferring without knowing, enhancing system usage by not accessing any needed process, occupying the system primary as well as secondary storages. Moreover, the main threats include theft of national electronics data and financial data. To get rid of these the anti-virus software and non-traditional anti-malware program are available in the market. The objective of the present work is to investigate their behaviors and provide some different remedial measure.

The threat perception mainly transmitted via Internet [1, 2] which revolutionized the works by bringing in ease and convenience, but at the same time invited a plethora of Malware the lethality of
which is an attack every fraction of minutes according to recent study [3]. The home computers are the most attack prone. Even if most of the home users are acquainted to use of anti-virus, a well designed virus can evade detection. Security software is excessively dependent on system calls in order to detect malicious code. If the system calls are manipulated to illusion anti-virus, the detection of virus becomes more difficult.

A virus infects computers with multi method criteria. Methodologies used for attacking the system vary from exploiting vulnerabilities of the operating system/software, buffer overflow, spamming, code manipulation and more. The hackers successfully exploit these vulnerabilities in the system. The Anti-virus program though can be fixed by updates for the system but is dependent on Internet connectivity. Moreover, the automatic updates may be disabled by the infection.

The most vulnerable and easy going accessibility remains with the Consumer computers. A deterministic algorithm that is fully dependent on virus definitions and behavioral patterns may not perfectly detect the presence of malicious code [4, 5, 6]. The creation of virus with generic signature patterns was investigated in [7].

Malware are the malicious software that infects a computer without the knowledge of the user. Malware can be classified into Viruses, Spy ware, Root kits, Trojan horse, etc [8]. A malware may contain a virus, a root kit and a password logger. The malware codes are written through software tools that are available on the internet. The code produced is very complex, made more difficult by the fact that most viruses generated nowadays are polymorphic [9] in nature.

Computer viruses are programs that are deliberately designed to interfere with computer operation, record, corrupt, or delete data, or spread themselves to other computers and throughout the Internet [10]. Internet worm was one of the first well known malware released [11]. An attacked computer started getting infected multiple times; the consequence was that the machines became very slow to the point that they could not be used. Personal Computer Viruses first started appearing in the 1980's. In the next section effort has been made to categorize the virus.

The first few viruses were simple machine language programs [13]. Today the number of known computer virus is estimated to be around over hundred thousand [3, 12]. The viruses had the ability to infect other executable code. Viruses have grown up lethal capabilities over the years. The following section describes the types of viruses and the methodology employed by virus writers.

2. Types of Virus

Virus can be broadly categorized into following:

Trojan Horse: This is a program that enters a machine disguised or embedded inside legitimate software. The Trojan looks harmless or something interesting to a user, but is actually very harmful when executed. Each Trojan has its own characteristic that is dependent on what the designer intended it to do.

Worms: A worm is a self replicating program. Unlike a virus, it does not attach itself to any existing program. It uses the network resources to infect other machines in the network. Worms always harm the network whereas viruses always infect or corrupt files on a targeted computer.

File Infectors: This type of virus embeds itself onto files or executables without changing its content. Majority of known viruses belong to this category [14].

Boot Virus: Boot sector viruses although minority in population [14] possesses more striking capabilities. These types of viruses operate by infecting the Master Boot Record (MBR) of a PC. It runs every time a machine is boot and is responsible for loading the rest of the Operating System. 'Crazy Boot', 'Brain', 'Anti.EXE' are few illustrations.

Date Viruses: These are mainly parasite to some event such as a particular date or a day of the week to get activated [15]. This type of virus or worm to gain momentum and can create havoc. Examples of such viruses are: 'Michelangelo', 'Sunday', and 'Century'.
Encrypted Virus: This is a type of virus whose body is encrypted. The encryption key varies from infection to infection causing the encrypted body to appear differently in every instance [16]. This methodology was used by virus writers to hide the virus from signature scanning techniques. ‘Cascade’ was one of the first viruses that used this methodology [17].

Generic Virus: This contains a encrypted body and a decryption engine like the encrypted virus. In addition to this, it also has a mutation engine that generates new encryption schemes for every infection. If a user runs a program that contains the virus, the decryption engine first executes and places the virus body in memory. The virus then starts the mutation engine which generates an encryption decryption routine for the next infection [18].

Rootkit: Rootkit is term derived from the UNIX term root. It was designed to give administrator privileges to the attacker. A well written rootkit can hook on the Operating System's Page fault handler and Virtual Memory controller to conceal its presence, and that of its files [19].

In addition to above categories, virus can be classified based on attacks rather than identities viz. social engineering [20, 21, 22] and vulnerability exploiters.

3. Anti-virus

The emergence of Anti-virus was the necessity for security since the appearance of computer virus in the early eighties. Anti-virus is a set of programmes that scans the system either continuously, or at specified times for the presence of malicious entities. The methodologies used by anti-virus software are as follows:

Signature Detection or Pattern Matching: Signature detection involves the anti-virus application scanning the computer for files that contain a code that it recognizes as the virus. The initial anti-virus programs would scan the entire executable file to find the presence of the virus code. Later the programmers found out that most of the virus infections involve virus placing its code on the entry point of the program. Hence the scanners started checking the entry point of the program; this methodology became obsolete when encrypted and polymorphic viruses emerged.

Image sensing: As encrypted viruses arrived, anti-virus software started using brute force decryption of the code. This was possible since they had to do known plaintext attack on the encrypted code; the plaintext was the known virus definitions. The virus writers used random encryption algorithms; hence the protocols were easy to crack [23].

Heuristics: Virus writers slowly started using techniques such as entry point obfuscations, by which they would not keep the virus code at the entry point of the executable. The emergence of viruses that could change their properties required a change in virus detection technology.

DNA scan: This method focus on what it is rather than what it does. The idea behind this is to identify viral code and then counter attack the code by neutralizing the software medium [29].

Frequency Analysis: This is involved in scanning a code to check presence of particular opcodes. Ordinarily they use DOS interrupts. This interrupt was hardly visible in malicious codes. Ordinarily frequency of opcodes are examined by them, and programs having them would not be scanned.

Since creation of virus program don’t follow any defined mechanism. Though these are rule based, it may not be possible to design the remedial measure in a purely procedural way. Rather mechanism designed out of tracking and responsive behavior may be taken into consideration as temporary relief amidst a bit random or trivial in nature. How the remedial measure can be taken out of exhaustive domain specific study of the most commonly known threats is presented In the following section.

4. The Case Study and remedial illustrations

On the way of identifying viral behaviors in a rule based and knowledge based way, some
investigation have been made with most commonly known virus. The goal was to observe their behavior, identify their working patterns and mine some properties so as to quarantine them or erase them permanently. Though the entire approach is based on generate and test mechanisms, during the course of mining the properties, some interesting observations have come out. Brief History of *autorun* with working principles and pattern of manipulation has been provided.

**Start Up folder for Windows 7:**

```
%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
%ProgramData%\Microsoft\Windows\Start Menu\Programs\Startup
%userprofile%\start menu\programs\Startup
%alluserprofile%\start menu\programs\Startup
```

**Common StartUp folder:**

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
```

**Uncommon Registry StartUp:**

```
Before affecting:
HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon
  UIHost=logonui.exe
  Shell=explorer.exe
  Userinit=%systemroot%\system32\userinit.exe
  HKEY_USERS\S-1-5-21-796845957-1303643608-68200330-1004\Software\Microsoft\Windows NT\CurrentVersion\Windows
  load="virus path"

After affecting:
HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon
  UIHost=worm path
  Shell=explorer.exe, worm path
  Userinit=%systemroot%\system32\userinit.exe, worm path
  Taskman= Value entered by any worm *
```

There is an effective difference between other startup value of registry and this “Taskman” value. Other startup value starts when system starts. If explorer.exe terminated and starts again these values are not affected any more. But Taskman value in registry at Particular address (mentioned) starts any exe or executable file it locate whenever explorer.exe starts run after any termination.

**4.1. Type 1 (temp.exe)**

The main .EXE file coated itself in Recycle Bin, My Computer, and Control Panel etc like special folder type.

**Working Mechanisms:** This type of worms is self executed with the help of Autorun.inf file. It replicates one computer to another computer with help of external portable storage device (e.g. pen drive). This type is not part of any continuous process. Computer infected by these worms, runs an exe file only when a new device is inserted. If the device contain some worm specifically autorun.inf file then that .exe stays inactive.

**Path:** System folder coated recycler but the .exe coated in recycle bin folder and it runs when the portable storage device inserted is not same at all. Their name and icon are different.

**Caution:** Replication of one device to another computer and so on.

**Removal Suggestion:**

- Show all system hidden files
- Connect infected portable storage
- Disable explorer.exe and turn it on again
Delete autorun.inf and recycle bin type folded worm.
Now insert the portable storage device and check which process starts immediately Task manger may help but it is really required to be done quickly and the .exe will be noticed
Create a batch file –
```bash
@echo off
wmic process list full > anyname.txt
```
It will show the process list in details as a text file anyname.txt

4.2. Type 2 (Foldername.exe)

It is very interesting type of worm. It replicates as folder name in USB and use Folder icon of Windows XP SP2.
- **Path:** %SystemDrive%
- **Name:** These worms named itself to a critical system process name (eg: Winlogon.exe) which is not terminated from task manager. If someone creates an exe and names it say winlogon.exe. Then the exe is run. But the process can’t be terminated from the task manager. This trick is used by these worms.
- **Behavior:** It sends a copy of itself to system folder and named itself to a system exe which is a critical process for the system. It sends its information to the registry for startup.

**Removal Way:**
- Check task manager to locate the process (you found a system process twice in the process list)
- Check for startup object using ‘MSCONFIG’ (Start → RUN → MSCONFIG → Startup)
- Delete/Rename the exe file of that worm in system folder.
- Delete registry entry for startup
  ```
  HKLM\Software\Microsoft\Windows\CurrentVersion\Run Name= “Worm Path”
  ```

**Caution:**
1) Huge Replication
2) Space eater
3) Memory Eater (for its folder icon user may start several processes)

4.3. Type 3 (Anti.VBS)

It is not harmful at all. Even it causes a recovery from any worm’s damage in registry. Main VBS/VBE files placed at system32
- **Path:** %systemfolder%
- **Behavior:** These worms use WSCRIPT.exe as a continuous process for its replication.
- **Caution:** Make internet explorer Home page particular (eg: www.yahoo.com.com , about: blank)

**Removal Suggestion:**
- Stop Wscript.exe from task manager
- Edit registry value as following
  ```
  HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon
  Userinit=C:\Windows\system32\userinit.exe
  ```

**NOTE:** It is very vital key for the system. If wrong information or path is entered then the system may get damaged

4.4. Type 4 (regsvr.exe)

Main exe placed itself in a system folder.
- **Behavior:** As other worm work it change particular value(s) of registry to startup itself. But this is a total new startup location. Regsvr.exe behaves like type 3. So we can named it type 3.6
Removal Suggestion
Edit registry value as following
HKLM\Software\Microsoft\Windows NT\Current Version\winlogon
Shell= explorer.exe

Similarly other common infections are investigated and the remedial measures are suggested by the proposed work.

5. Conclusions

The proposed investigation suggests that any threat can be dealt with if enough information can be accumulated about it. The series of instruction followed by an Anti-virus may be manually replicated as well in order to get rid of the Virus.

The reliability of the anti-virus process may be improved by hiding its presence from other processes on the machine. The inherent reason is that a malware may infect any process on the system, in such a situation no component of a consumer computer can be trusted. This is achieved step-wise by changing the names of the files and changing the registry entries by installing the process under a different name followed by the migration of the process continuously to different address spaces to avoid detection by any malware. This employable approach may reduce the Anti-virus dependency.

References

1. Salus. PH. Casting the Net: From ARPANET to Internet and Beyond. Addison- Wesley; 1995.
2. Abbate, J. Inventing the Internet. Cambridge, MA: The MIT; 1999.
3. http://www.cknow.com/vtutor/NumberofViruses.html.
4. Cohen, FB. Operating system protection through program evolution. Comput. Secur. 1993; 12(6):565-584.
5. Chess, DM, White SR. An undetectable computer virus. Proc. of the 2000 Virus Bulletin Conference (VB2000); 2000.
6. Schneier B, Attack trends: 2004 and 2005. Queue, 2005; 3(5): 52-53.
7. Wang Y. Using Mobile Agent Results to Create Hard-to-detect Computer Viruses. Fac. Math., University of Waterloo; 2000.
8. Chen TM. Trends in Viruses and Worms. The Internet Protocol Journal, 2003; 6(3): 23-33.
9. http://www.webopedia.com/TERM/P/polymorphic_virus.html.
10. http://www.microsoft.com/athome/security/viruses/intro_viruses what.mspx.
11. Spaford EH. The internet worm program: an analysis. ACM SIGCOMM Comp Comm. Review, 1989; 19(1):17-57.
12. Munro J. Anti virus research and detection techniques. 2002, http://www.extremetech.com/article2/0 %2C1558 %2C325439 %2C00.asp.
13. Nachenberg C. Computer virus-antivirus coevolution. Communications of the ACM, 1997; 40(1):46-51.
14. Keppart JO, Sorkin JB, Chess DM, White SR. Fighting computer Viruses. Scientific American, 1997; 277(5):56-61.
15. http://courses.cs.vt.edu/professionalism/Viruses/viruses.html.
16. Nachenberg C. Understanding and managing polymorphic viruses. The Symantec Enterprise Papers, 30, 1996.
17. Sanok DJ, Jr. An analysis of how antivirus methodologies are utilized in protecting computers from malicious code. Proc. of the 2nd annual conference on Information security curriculum development, pp. 142-144, 2005.
18. Virus list. http://www.viruslist.com/en/viruses/encyclopedia?chapter=153311150.
19. Szor P. The Art of Computer Virus Research and Defense. Addison Wesley for Symantec Pres, 2005.
20. Social engineering fundamentals. http://www.securityfocus.com/infocus/1527.
21. Cert incident note in-99-02. http://www.cert.org/incident_notes/IN-99-02.html.
22. Chen TM. Trends in Viruses and Worms. The Internet Protocol Journal, 2003; 6(3):23-33.
23. Malware report: The impact of malicious code file attacks. http://www.computersecurity.com/article.cfm?id=1090.
24. http://www.eweeek.com/article2/0,1895,2034680,00.asp.
25. Beast trojan. http://www.answers.com/topic/beast-trojan
26. The beast. http://lists.virus.org/dshied-0310/msg00337.html.
27. http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=42627.
28. Witter F. A.I. Handling, and H. Exploits. Responding to downloader-w.
29. Chen TM. Trends in Viruses and Worms. The Internet Protocol Journal, 6(3):23-33, 2003.