American Journal of Networks and Communications
2015; 5(1-1): 20-22
Published online August 6, 2015 (http://www.sciencepublishinggroup.com/j/ajnc)
doi: 10.11648/j.sjbm.s.2016050101.15
ISSN: 2326-893X (Print); ISSN: 2326-8964 (Online)

A Malware Analysis Using Static and Dynamic Techniques

Bymbadorj Dondogmegd[1, 2, *], **Usukhbayr B.**[2], **Nyamjav J.**[2]

[1]Department of Electrical Engineering, Ulaanbaatar State University, Ulaanbaatar, Mongolia
[2]Department of Electronic and Communication Engineering, National University of Mongolia, Ulaanbaatar, Mongolia

Email address:
pheelectro@gmail.com (B. Dondogmegd), usukhbayar@num.edu.mn (Usukhbayr B.), nyamjav@num.edu.mn (Nyamjav J.)

To cite this article:
Bymbadorj Dondogmegd, Usukhbayr B., Nyamjav J.. A Malware Analysis Using Static and Dynamic Techniques. *American Journal of Networks and Communications*. Special Issue: Traffic and Performance Engineering for Networks. Vol. 5, No. 1-1, 2015, pp. 20-22.
doi: 10.11648/j.sjbm.s.2016050101.15

Abstract: In this survey work we analyze “win 32 malware gen” it’s genre, procedure, harm using static and dynamic techniques. Static and dynamic methods were used to analyze a software program for any threats to the system. Static analysis involves testing its own source code and analyzing the threat itself, dynamic analysis involves specific secure, keeping threats within a system and analyzing the working progress of threats within the system.

Keywords: Malware, Threat - Data Fail Safe

1. Introduction

Malware spreads over LAN, online networks intended to steal information or spy on computer users for an extended period without their knowledge. Malware, short for malicious software, is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. 'Malware' is an umbrella term used to refer to a variety of forms of hostile or intrusive software, including computer viruses, worms, trojan horses, ransomware, spyware, adware, scareware and other malicious programs. Malware could be arranged into 14 main types [2]. It can take the form of executable code, scripts, active content, and other software. To test whether the software may contain potential threats we used static source code and dynamic code analysis [2, 3]. Win32.Malware-Gen refers to a range of malware applications that infect computers running a 32-bit version of the Windows operating system. Depending on which version of Win32, Malware-Gen of your computer is infected with, the virus may download and install other viruses, monitor computer activities, log keystrokes or corrupt your system registry and files. Programs that contain malicious code use this technique for analyzing the static and dynamic core of your computer.

Dynamic analysis involves the testing and evaluation of a program by executing data analysis in real-time. The objective is to find errors in a program while it is running, rather than by repeatedly examining the program’s code offline. Snapshots of the uninfected original source code are used to compare and analyze the infected system. Dynamic analysis compares the system’s processes, system registry, and downloaded networks [4-6]. The testing and analysis process cannot be completed by only doing one attempt, successful analysis often requires multiple runs.

Static code analysis records the HTML, GUI, Scripts, passwords, control string, and other commands [4]. By accumulating and analyzing these codes, static analysis creates certain “signatures” (algorithms, codes), describes which files are malware related, and identifies which commands the malware is using.

2. Related Works

The power of such test programs to isolate and destroy harmful malware is needed to effect control of the spread of viruses and create a transparent protective environment. Create a first state to collect information using the following program, it is controlled following the operating system, isolated environment can be carried out by a malicious program.

- PEview program’s viewing of structure and content of 32 bit. Portable Executable.(PE) file. Show date and time stamp of the malware complied and created.
- PEiD program’s malware authors often pack the malware
- Detect packers used (if any)
- Detect the language being used to write the malware
- DependsWalker program’s scan any 32 bit or 64 bit windows module

- Build a hierarchical tree diagram of all dependent modules.
- Used to identify any suspicious API(s) and DLL(s) imported
- Process Explorer program's The unique capabilities of *Process Explorer* make it useful for tracking down DLL-version problems or handle leaks, and provide insight into the way Windows and applications work [11].
- Process Monitor is an advanced monitoring tool for Windows that shows real-time file system, registry and process/thread activity [9].
- Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues. [10].
- RegShot program's windows registry comparison tool that allows you to take and compare two registry snapshots.[12]

3. Experimental Result

In this survey work we tested static code analyses and dynamic one too. Rig used for test hardware has 2nd Generation i7 CPU, 4 GB ram and Windows 7, Virtual XP are for software. In Virtual XP we used Reshot software for collecting registry data. Then we used Review, Pied, and Depends Walker software's for comprehensive study. That malware is complete date: 2013.11.04.

Has an ability to work on windows XP and NT software. Written by C++. DLL and functions for threat are described above in shows Figure 1.

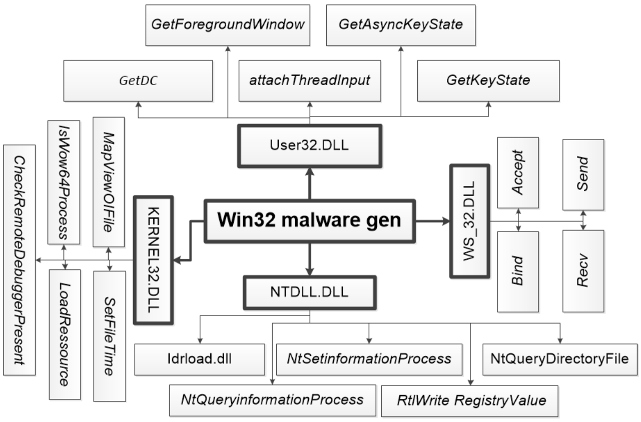

Figure 1. *Destructive load function of DLL files and system software.*

Figure 1 Win32 malware gen program uses for those DLL files:

Kernel32.dll is most often used file that contains:

- IsWoW64Process: Used by a 32-bit process to determine if it is running on a 64-bit operating system [12].
- MapViewOfFile: Maps a file into memory and makes the contents of the file accessible via memory addresses. Launchers, loaders, and injectors use this function to read and modify PE files. By using MapViewOfFile, the malware can avoid using Write File to modify the contents of a file [12].
- Load Resource: Loads a resource from a PE file into memory. Malware sometimes uses resources to store strings, configuration information, or other malicious files [12].
- SetFileTime: Modifies the creation, access, or last modified time of a file. Malware often uses this function to conceal malicious activity [12].

User32.dll contains user interface codes.

- AttachThreadInput: Attaches the input processing for one thread to another so that the second thread receives input events such as keyboard and mouse events. Key loggers and other spyware use this function [12].
- GetAsyncKeyState: Used to determine whether a particular key is being pressed. Malware sometimes uses this function to implement a key logger [12].
- GetDC: Returns a handle to a device context for a window or the whole screen. Spyware that takes screen captures often uses this function [12].
- GetForegroundWindow:Returns a handle to the window currently in the foreground of the desktop. Key loggers commonly use this function to determine in which window the user is entering his keystrokes [12].
- GetKeyState: Used by key loggers to obtain the status of a particular key on the keyboard [12].

Ntdll.dll The DLL ntdll.dll is primarily concerned with system tasks. It includes a number of kernel-mode functions which implements much of the functionality of the Windows Application Programming Interface (API). As usual Ntdll.dll doesn't work alone, works through kernel.dll. If it is used via any software alone that software aims hiding malicious acts. Hiding progress, controlling process etc.

- RtlWriteRegistryValue: Used to write a value to the registry from kernel-mode code [12].
- NtQueryDirectoryFile: Returns information about files in a directory. Rootkits commonly hook this function in order to hide files [12].

Ws2_32.dll File that contains the Windows Sockets API used by most Internet and network applications to handle network connections. This is a module that contains many different internet functions, like all DLL's, many of them are used to share functions for various applications. Such as FTP, HTTP, NTP

- Connect: Used to connect to a remote socket. Malware often uses low-level functionality to connect to a command-and-control server [12].
- Recv: Receives data from a remote machine. Malware often uses this function to receive data from a remote command-and-control server [12].
- Send: Sends data to a remote machine. Malware often uses this function to send data to a remote command-and-control server [12].
- Bind: Used to associate a local address to a socket in order to listen for incoming connections [12].

We analyzed Win32 Malware gen has functions to damage

OS, hiding itself, using network ports. As for dynamic analysis we used Process monitor, Process explorer and We Shack on virtual OS. We found which area Win32 Malware Gen affects. Image 2 displays areas affected in Table 1.

Table 1. *Changes to the system contains the parts.*

File system	Malware gen
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Comon-Controls_6595b64144ccf1df_6.0.2600.6028_x-ww_61e65202	X
C:\Documents and Settings\Administrator\Local Settings\Temp\	X
C:\Documents and Settings\Administrator\Application Data	X
C:\Windows\Prefetch\	X
Windows registry	X

As shown is image 2 Win32 Malware Gen copies ScreenSaver.scr into C:\Documents and Settings\Administrator\ApplicationData. OS registry affected by Win32 Malware gen displayed above. Windows Registry is a hierarchical database that stores configuration settings and options on Microsoft Windows operating systems Windows operating system has six type of registry special Om their own [7]. Table 2 shown, Win32 malware gen related registries are shown above.

Table 2. *Registry changed section.*

File system	Malware gen
HKLM\Software\Microsoft	X
HKLM\System\ControlSet001\Control\	X
HKLM\Hardware\	
HKLM\System\CurrentControlset\Services\	X
HKLM\Software\Microsoft\Cryptography\	X
HKLM\Software\Microsoft\Windows NT\CurrentVersion\	X
HKU\S-1-5-21-602162358-492894223-299502267-500\Software\Microsoft\Windows\CurrentVersion\Run	X
HKLM\SOFTWARE\Microsoft\DirectDraw\MostRecentApplication	X
HKLM\SOFTWARE\Microsoft\Windows\Current Version\Extensions\	X
HKLM\SYSTEM\ControlSet001\Enum\Root\	X

The Windows Registry contains a root key titled HKEY_LOCAL_MACHINE, or HKLM. The HKLM root key contains settings that relate to the local computer software is and drivers [8]. HKEY_USERS contains user-specific configuration information for all currently active users on the computer [8].

By editing HKLM and HKU malware called screensaver.scr and changed its code. Network activities collected by Wireshark software are shown in Table 3.

Table 3. *Host a weak port is used to traffic.*

Port	Protocol	Process
1140	TCP	Windows\syswow64\vmnate.exe
1140	TCP	Windows\syswow64\vmnate.exe

Win32 malware gen infects active networks, multiplies itself, and calls certain websites to exacerbate OS.

4. Conclusion

Analyzed Win32 Malware gen uses static and dynamic methods. Based on static analyze, we knew win32 malware gen was a threat. By the dynamic method, we studied how malware affecting OS. As a result of these methods, the main result proves malware copies screensaverpro.scr to system, changes HKLM. HKEY_USERS registry hides itself, works through network.

References

[1] www.symantec.com/connect/articles/malware-analysis-administrators.

[2] Chris Gates, “Hacker Defender Rootkit for the Masses”, 2007.

[3] “Malware challenge” , jerome.segura@gmail.com.

[4] Dean De Beer, “Malware Analysis Challenge III”, 2007.

[5] Bill Arnold, David Chess, John Moral, “An Environment for Controlled Worm Replication and Analysis”, 2008.

[6] Alla Segal, “Reverse-Engineering Malware”, 2006.

[7] http://en.wikipedia.org/wiki/Windows_Registry

[8] http://kb.chemtable.com/ru/windows-registry-main-keys.htm#hkcu

[9] https://technet.microsoft.com/en-us/library/bb896645.aspx

[10] “Wireshark FAQ”, Retrieved 31 December 2011.

[11] https://technet.microsoft.com/en-us/sysinternals/bb896653.aspx

[12] “Practical Malware Analysis” The Hands-On Guide to Dissecting Malicious Software.