A Complete Dynamic Malware Analysis

Navroop Kaur
PA, Cstd
CDAC Mohali

Amit Kumar Bindal, PhD
Associate Professor
Department of Computer Engineering
M M University Mullana

ABSTRACT
Now a days thousands of malware samples are received by anti-malware companies on daily basis. And these large numbers are send for analysis by a number of automated analysis tools. These tool automatically execute a program in a controlled environment and generate a report describing the program's behaviour.

This research paper is a contribution towards the Dynamic Malware analysis. The aim is to provide the general malware features found in recent malware by performing dynamic malware analysis using cuckoo sandbox executed on Windows XP (SP3). This paper also discusses the detailed information about techniques & tools used in dynamic malware analysis.

Keywords
Malware, Sandbox, Malicious.

1. INTRODUCTION
Today main source for social interaction is the internet. As maximum crowd prefer to use the services that are offered on the Internet, people becoming more prone to cyber-crime and can be widely conned by miscreant with the help of Trojans, viruses, bots etc, which results in propagation of bulks of malwares. Analysing these malwares are a vast domain and can analyze malwares by various ways i.e. static analysis and dynamic analysis. Static analysis involve examine the code without executing it and dynamic analysis involve Analysing the behaviour of malware by executing it. But due to limitations of static analysis [14] it promoted researchers and students to focus on dynamic malware analysis.

Automated, dynamic malware analysis system involves execution of a binary in a safe environment, monitoring the program's execution and generating an analysis report summarizing the behaviour of the program. It is helpful for executing a packed binary after unpacking it (binaries who's code is not available for static analysis). Instead of benefits there is a danger of dynamic malware analysis, as it can damage the whole system or can block the complete network so there is need to handle it properly. Dynamic malware analysis involves various types of tools and techniques which highlights its strength.

Generally a virtual machine or sandbox is used for dynamic malware analysis. List of sandboxes available on internet may in the form of service, open source or commercial are as follows- (Anubis [15], Cuckoo Sandbox [11], Joe Sandbox Document Analyzer [13], Norman Sandbox [6], and Threat-Expert [7]). Based on a suspicious features, binary need to execute for shorter or longer time [2]. After dynamic analysis feature extraction is main concern if we want to identify unknown hashes in future and later plan them to cluster. So our motive is extracting the features lying in malwares. For this research problem the input is Honeynet data. And output is feature extraction (i.e. file system changes, registry changes, mutexes created, packers anti debugging techniques used by malwares) etc.

1.1 Malware
Virus, worms and Trojan belongs to class Malwares and malwares are malicious software, used to damage or gain access to a computer system without owner consent. Viruses, Worms, Trojan and Spyware are belongs to class malware. The main goal of gaining control is to disturb the normal operations of the target, steal secret information etc. The variant of a malware or malware exhibit similar malicious behaviour can be easily classify by the terms, such as 'virus', 'worm', or 'Trojans'.

Whereas a bot is a piece of malware that infects computer systems connected on internet and even allows that external entity called bot master, to remotely control this system. For more detail refer [1][16][17].

2. MALWARE ANALYSIS
The purpose of Malware analysis is to determine the motive and functionality of the malware samples such as a viruses, worms or Trojan horse. Traditionally, Malware analysis is a critical manual process that even consumes more time. Analysing the run time behaviour of a code is called dynamic analysis while static analysis refers to code analysis by reverse engineering and pcap dump analysis involves network traffic analysis.

The main focus of authors is dynamic malware analysis. It is to find the actual motive of malware by providing it real, virtual or sandbox environment to do its work.

2.1 Static Malware Analysis
Static Analysis [3] is analyzing software without executing it. Techniques of Static analysis can be apply on different parts of a program. On availability of source code, static analysis tools are useful in finding flaws in memory corruption.
Static analysis tools can also be used on the binary of a program. But one of the limitation of this process is that at the time of compilation of program's source code to a binary executable, it leads to loss of information (e.g, variables or size of data structures). Hence makes the analyses of the code difficult.

Limitations of static analysis:-
Due to the unavailability of source code of malware samples, it forces further techniques of static analysis to retrieve the information from the binary form of the malware. Many times the disassembly of such programs provides ambiguous results if the binary uses self-modifying code techniques (e.g., packer). [1]

2.2 Dynamic Malware Analysis
In Dynamic Malware Analysis, Malware is executed in a controlled environment and monitors its run time behaviour in order to analyze the malicious behaviour. In Dynamic Malware Analysis, malware if packed then it unpacks itself &

execute in a controlled environment, dynamic malware analysis evades the restrictions come with static analysis (i.e. issues related to unpacking and obfuscation).

The two basic ways for dynamic malware analysis. These are as follows:-

Comparative Approach: A malware sample is executed for particular time and changes made in the system are analyzed by comparing the two states of system. So this approach provides a comparison report which states behaviour of malware.

Run Time Behaviour Analysis: Here we use tools for monitoring the malicious activities made by malware during runtime.

Example of one parameter taken to observe Malware Behaviour:-

It involves File System changes, Registry changes and Network changes [12]. Very interesting parameter for analysing the malware is operating system services accessed by malwares i.e. which services of operating system are requested by the binary. All the activities that involves communication with the environment that can be working with file system in order to make network vulnerable and controlled by bots etc. all this needs to access appropriate operating system services.

Dangers of dynamic malware analysis

As it uses a simple approach, the purpose of an unknown program is to run it and see what happens but there are major problems comes with this approach. The program could run destructively and can damage all information on the machine [8]. This can easily lead to network traffic congestion or the program could send malware to other people in any form. All this would not make a good impression.

Rather than running malicious program in an environment where it can destroy the whole system, it's safer to run the program in a sandbox. Sandbox is stolen word from ballistics, where weapons are tested by shooting bullets into a box filled with sand, so that the bullets can do no harm. Similarly sandbox is a controlled environment for executing an unknown malware.

Sandboxes can be used in several ways. It can act as a sacrificial lamb: a real, but disposable machine with either limited network or with no network access at all. This is a realistic approach, but can be inconvenient if you want to make reproducible measurements.

Instead of providing the unknown program the complete sacrificial machine, you can use more subtle techniques. These helps to execute a program and passively monitoring it, just like hanging off wires that are under the control by an investigator.

3. TECHNIQUES USED IN DYNAMIC MALWARE ANALYSIS

3.1 Monitoring Function Call

Commonly known function call is a call that passes control to a subroutine, after execution it passes the execution control returns to the next instruction in main program. Here we monitor the whole process that which function is called by which program as it helps us analysing the behaviour of the program [5]. This is a hooking concept; i.e. intercepting function call is called hooking. The manipulation of analyzed program is such that in addition to the concern function, a so-called hook function is invoked. It is this hook function which is responsible for implementing the analysis functionality required by us, such as analyzing the input parameters or recording all its invocation to log file.

3.2 Analysis of Function Parameter

This is second very important technique used in Dynamic Malware Analysis in which function parameters are analyses dynamically which monitor what the actual values passed at the time of invoking the function.

The output of a system call “CreateFile” is used further as input to WriteFile call, such a sequence is very helpful. Function calls grouping into a logical sets provides detailed information about the program’s behaviour from a different angles.

3.3 Information Flow Tracking

The Main approach to monitoring function call during the execution of program, is the analysis on how the program actually working on data. And information flow tracking tracks the flow i.e within a program how data flow and being modified. These techniques are the core methodology used by dynamic analysis tools and working at operating systems various levels.

Dynamic Anlaysis Tools

Table -1 Tools used in Malware Analysis

No.	Process Explorer	Monitor Currently Running Process
1.	FileMon	Monitor File Operation
2.	Regmon	Monitor Operation on Registry
3.	Regshot	Takes Snapshot of the registry and associated file
4.	TCPVIEW	Displays all TCP & UDP open connections and the process that opened and using the port
5.	TDIMon	Network connectivity is logged, but packet contents are not logged
6.	Ethereal	Packet Sniffer, helps in viewing of contents/payload

Collection of these tools are used in malware analysis tools (can be called Sandbox) for providing consolidated report.

4. MALWARE ANALYSIS TOOLS

4.1 Anubis

Anubis executes the sample in an emulated Windows XP environment running as the guest in Qemu.

Input to this Sandbox: - Binary & URL.

Output from this sandbox:- modified Registry details, File system modification detail, Process modification detail, other activities.

Like the core-Anubis does for Windows PE executables, a latest Andrubis used to analyse Android apps by executing apps in a sandbox and provides a detailed report on their behaviour along with what files it access, network access, crypto operations and what information sent. As Anubis perform dynamic analysis and its brother Andrubis does static analysis, provides information on e.g. the app's activities, services.[3]

4.2 GFISandbox

GFI Sandbox is commercial Sandbox that is used by various Antivirus Companies for the purpose of Dynamic Malware Analysis. Dynamic analysis shows how the applications are executed, what file system changes are done, monitor the network traffic and the severity level of the threat all this procedure happens in a secure and controlled environment. This Sandbox helps researchers or students to analyze the behaviour of suspected viruses, worms, trojans and other malware by executing the program inside a controlled environment then recording all changes (at file system and registry level) such as any Windows API calls made and network traffic. Presently they are having 300 plus rules for malware analysis.

Input to this sandbox:- File and URL.

Output from this sandbox:- File system changes, Registry changes, Network changes, Memory changes.

4.3 Norman Sandbox Analyzer Pro

Norman SandBox is used for monitoring memory, disassembled code, virtual hard disk, registers and network activity and if required it can be manipulated in order to understand the complete behaviour of the suspicious code[4].

Many advanced debugging features are lying in this sandbox like the ability to take snapshots, search and dump memory contents, simulate execution in reverse, log and save network packets, and many others.

4.4 Joe Sandbox Document Analyzer

This Sandbox detects potential malicious documents which are in the following format:

Acrobat Reader 9.5.0

Office (Excel, Word, PowerPoint) 2003

Office (Word, Excel, PowerPoint) 2010 SP2

The behaviour of the Acrobat reader is analyzed with Joe Sandbox Desktop which provides static, dynamic and hybrid code analysis [13]. Any captured behaviour is rated and classified by an extensive and generic behaviour signature set.

4.5 Cuckoo Sandbox [11]

It is an open source dynamic malware analysis system. It involves execution of the Malware in an isolated Windows Operating System and provides the consolidated report

It can retrieve the following type of results:

Provides traces of win32 API calls.

Detail about files being created, deleted and downloaded by the malware during its execution.

Help to take the Memory dumps of malware processes.

It provides the network traffic trace in PCAP format.

Screenshot of all the processes going in Windows Operating System.

Input:- Binary and URL

Output:- File system changes, registry changes, mutexes created, memory dumps, PEID detail.

Now next section will be useful as providing brief about Malware indicative features extracted from Honeynet data by passing the Malware from Cuckoo Sandbox and Threat Analyzer. These feature can be further useful in clustering the malwares.

5. MALICIOUS BINARY FEATURES

As lots of work is going on in this dynamic malware analysis domain. Features from various levels like File system changes, registry changes are already being used for dynamic malware analysis. The detail mention below is addon to previous work which list out features found in recent malwares. We have also added one more level in existing research i.e mutexes created for which very less work has been done in past.

Packers:-Packers are the routines basically decompress the given program and jump to it once achieved. Recent malwares packs itself with following types of packer. And the most common one is UPX packing

Table-2 Malicious Binary packed with following packers. [9]

S. No.	Packers	Reference
1	ASProtectV2XDLLAlexeySolodovnikov	Prevalent Characteristics in Modern Malware, Blackhat USA 2014
2	ASPackv212AlexeySolodovnikov	-do-
3	PECompactv2xx	-do-
4	NETexecutableMicrosoft	-do-
5	UPXProtectorv10x2	-do-
6	Armadillov1xxv2xx	-do-
7	UPX20030XMarkusOberhumerLaszloMolnarJohnReiser	-do-
8	UPX290LZMAMarkusOberhumerLaszloMolnarJohnReiser	-do-
9	Armadillov171	-do-
10	UPXv20MarkusLaszloReiser	-do-
11	UPXV200V290MarkusOberhumerLaszloMolnarJohnReiser	-do-
12	Others	-do-

The figure 1 shows the screenshot of the report generated after executing binary in cuckoo sandbox and displays that binary is packed with Armadillo v1.71 packers. Which indicates that binary can be a Malware as it is packed with one of the packers which recent malwares usually used to pack themselves.

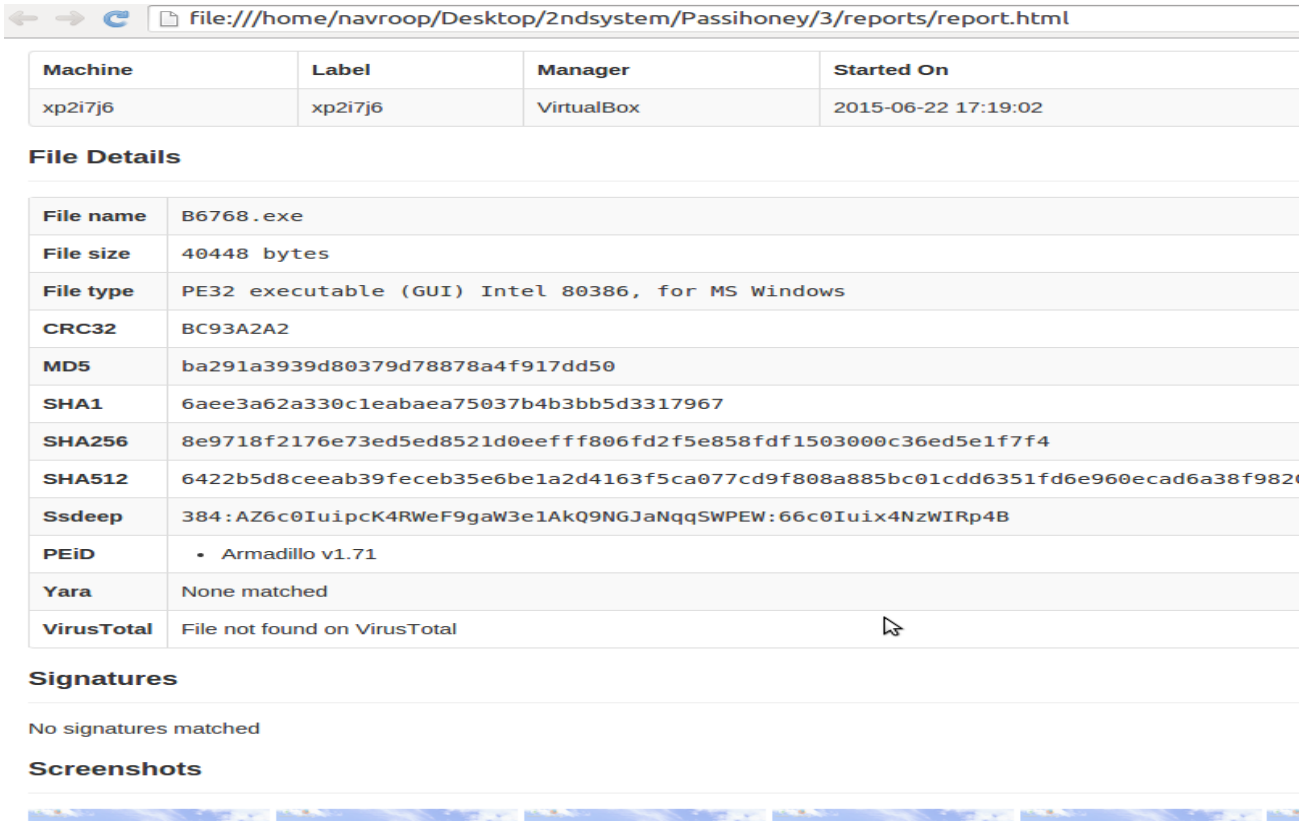

Figure 1 Report display that binary B6768.exe is packed with Armadillo v1.71pakcer.

6. MUTEXES

A mutex, also called a lock, is a program to control the simultaneous access of the system resources. They are used by malware creators to overcome the effect made by the different instances of the same malware on the system. When the trojan infects a system, then first of all try to obtain a handle to a "named" mutex, if the process fails, then the malware exits. One of the easiest way to check whether mutex is present is "CreateMutex Function". This function is used by malwares for checking if the system is infected so one approach to detect the presence of existence of malware is trying to obtain a handle to the created mutex. This is an add-on in analysing recent malwares comparatively easier way.

Table 3 : **Mutexes found in recent malwares**

No.	Mutexes
1	Shame Cache Mutex
2	Groove:PathMutex:Ze0sHV3d1w5GKIk3Fk0r3Vg4RV8=]
3	Local_MSFTHISTORY!
4	Local\WininetStartupMutex
5	Local\WininetProxyRegistryMutex
6	MSCTF.Shared.MUTEX.MM
7	Local\ZoneAttributeCacheCounterMutex(Even found in ZBot)
8	huaxiacmd.f3322.org
9	VogA.14

The below given figure is a screenshot of the report found after executing a binary sample in cuckoo sandbox. Which displays that binary is trying to create a

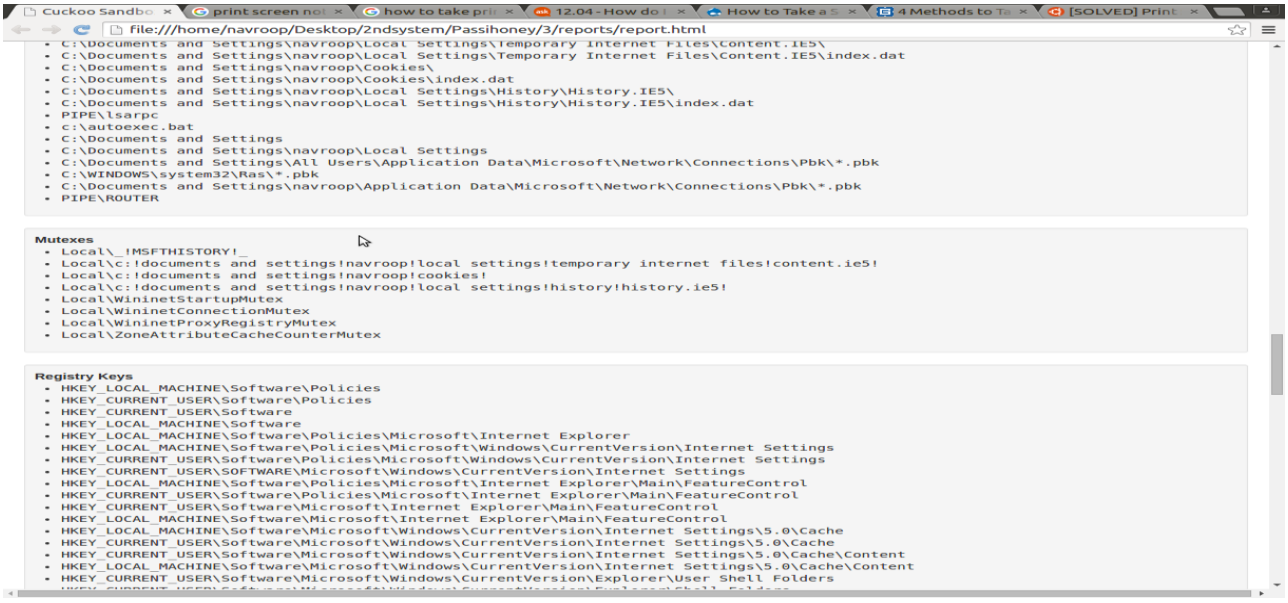

Figure 2 Screenshot of report display mutexes created by binary sample while executing in cuckoo sandbox.

mutex which is mainly created by malwares. This a second indicator that binary can be malware.

6.1 File System Changes

Explains the changes in file system i.e files open, write and deleted etc. The given below are the new features found in recent Malware. This is an addon to already existing malware features [12][10].

S. No.	File System Changes (Create , Write, Delete)
1	Create File:- FileName => C:\Documents and Settings\navroop\Cookies\index.dat
2	WWriitefile:- ahref="http://www.iziu.net/hjs/">HttpFileServerv2.3e293 \xe9\x9a\x8f\xe6\xb3\xa2\xe6\xb1\x89\xe5\ x8c\x96\ xe7\x99\x78</a “
3	CreatedProcess:-C:\WINDOWS\system32\Ras*.pbk
4	Create file:- FileName => c:\Win_lj.ini
5	CreateFile C:\WINDOWS\system32\msctfime.ime
6	CreateFile:-C:\WINDOWS\Registration\ R000000000007.clbFileName=> C:\DOCUME~1\navroop\LOCALS~1\Temp\B6775.exe

Figure 3 File system changes made by recent malwares

The below given figure is a screenshot of the report found after executing the given binary sample in cuckoo sandbox. This screenshot displays how a file system changes can be displayed in report and further used for manual analysis. This is a third indicator if this type of file system changes exhibit after executing binary sample in sandbox then binary can be malicious.

6.2 Registry Changes

Changes in the registry made by malwares in order to take the control on the system depending on the malware. This are the features found in recent malwares and add-on to already existing features [12].

S. No.	Registry Changes
1	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Vwxyab Defghijk Mno
2	SubKey => SYSTEM\CurrentControlSet\Services\Vwxyab Defghijk Mno
3	SubKey => SYSTEM\CurrentControlSet\Services\RemoteAccess\RouterManagers\Ip
4	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mnopqr Tuvwxyab Def
5	SubKey =>Software\Microsoft\windows\CurrentVersion\Internet Settings\Connections

Figure 4: Registry changes made by recent malwares The

Figure 5 given below shows that what registry changes is made binary sample after executing it in cuckoo sandbox. The binary is trying the access the complete control on operating system by making following changes.

Figure 5: Screenshot of the report showing Registry changes after executing given binary sample in cuckoo sandbox.

7. CONCLUSION

This research paper reveals the new suspicious features found in recent malwares and can be add on to earlier existing features, which can be quite more effective for analysing now a days unknown samples. This research paper also provide complete detail about dynamic malware analyses and motivates the researchers and students to move further in this domain

8. REFERENCES

[1] Manuel Egele, Theodoor Scholte, Engin Kirda, Christopher Kruegel, "A Survey on Automated Dynamic Malware Analysis Techniques and Tools", ACM Computing Surveys Journal, February 2012

[2] Ulrich Bayer, Engin Kirda, Christopher Kruegel, "Improving the Efficiency of Dynamic Malware Analysis" , 25th Symposium On Applied Computing (SAC), March 2010.

[3] Gadhiya, Kaushal Bhavsar "Techniques for Malware Analysis".

[4] http://www.insectraforensics.com/sandbox-analyzer-proDolly Uppal1, Vishakha Mehra2 and Vinod Verma3, "Basic survey on Malware Analysis, Tools and Techniques",International Journal on Computational Sciences & Applications (IJCSA), February 2014

[5] NormanSandbox.http://www.norman.com/microsites/nsic/ , 2009.

[6] ThreatExpert. http://www.threatexpert.com/ , 2009.

[7] MalwareAnalysisBasics,http://www.porcupine.org/forensics/forensic-discovery/chapter6.html

[8] Gabriel Negreira Barbosa,Rodrigo Rubira Branco, "Prevalent Characteristics in Modern Malware", Black Hat USA 2014

[9] Ulrich Bayer, Andreas Moser, Christopher Kruegel, and Engin Kirda, "Dynamic Analysis of Malicious Code" , Journal in Computer Virology, Springer Computer Science

[10] Cuckoo Sandbox, http://cuckoosandbox.org

[11] Ulrich Bayer, Imam Habibi, Davide Balzarotti ,Engin Kirda , and Christopher Kruegel "A View on Current Malware Behaviors".

[12] http://www.document-analyzer.net/

[13] Moser, A., Kruegel, C., and Kirda, E. 2007b, "Limits of static analysis for malware detection" in 23rd Annual Computer Security Applications Conference (ACSAC)

[14] Anubis. http://anubis.iseclab.org , 2009

[15] Ed Skoudis, "Malware: Fighting Malicious Code", dec 2003.

[16] C. Kruegel, W. Robertson and G. Vigna, " Detecting Kernel-Level Rootkits Through Binary Analysis" In Annual Computer Security Application Conference (ACSAC), 2004.