Portable Dynamic Malware Analysis with an Improved Scalability and Automatisation

Abdurrahman Pektaş and Tankut Acarman(✉)

Computer Engineering Department, Galatasaray University,
Ortaköy, 34349 İstanbul, Turkey
apektas@yandex.com, tacarman@gsu.edu.tr
http://gsu.edu.tr

Abstract. A malware is deployed ubiquitously to steal safety or liability-critical information and damage the compromised systems. In this paper, we present a portable, scalable and transparent system for dynamic analysis of malware targeting Windows OS. The portability feature is enabled by introducing a driver capable of collecting the behavioural activities of analysed samples in low kernel level and detection of a new malware in the latest version of Windows OS is guaranteed without waiting for its signature update. A large volume and variety of malicious behaviour is monitored and analysed by the presented virtual, scalable and automated system deployment. End-to-end design is presented and functional tests of portability feature are conducted by compiling the developed kernel driver component in the analysis machine. Evaluation is performed by using recently captured malware samples that are automatically analysed and detected on a Windows 8 Ultimate 64-bit and Windows 10 OS.

Keywords: Malware · Dynamic analysis · Portability · Detection

1 Introduction

A malicious software, or malware for short, is a software used or created by an attacker to perform his bad intentions on a computer system without authorisation and knowledge of its user. Stealing safety or liability-critical information and damaging the compromised system is mainly targeted. The recent developments in the field of computation systems and the proliferation of systems such as smart phones, tablet, Internet of Things (IoT), cloud computing have led to an increased interest in malware development. According to [1], more than 430 million new malware samples were released in 2015 with an increase of 36% from the previous year, and a new zero-day vulnerability was discovered at each week on average with an doubled release frequency in comparison with the previous year. Security tools compare suspicious files with their malware definition database,

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_22

which are constructed based on known security issues by analysts. Based on definition database, these tools check whether a given file is malware or not. For example, anti-viruses running on end-user computers are mainly signature-based solutions. From an abstract point of view, they read the suspicious file in a binary format and look for a match with their signature database. According to [2], anti-virus solutions are fast and effective for known malware but their accuracy of analysis and adequacy are easily degraded by slightly changed malware. Although signature-based solutions are subject to delay on identifying new and obfuscated version of malware and updating their databases, they are convenient over previously registered attacks. Subject to an increasing number of sophisticated, advanced, and targeted attacks (a.k.a., advanced persistent threat, APT), security researchers and practitioners have explored more robust and timely solutions to new and unknown threats. Generally, the security community prefers sandboxing where malware activities are monitored during their execution in an isolated and controlled environment. These systems track and inform about file, registry, network, and process activities. To successfully detect malware and take appropriate counter measures, dynamic analysis can be considered as an integrated scenario and solution of an environment provided to malware for being deployed and performing its tasks. CWSandbox leverages API hooking technique in user mode to track malware's activities [3]. Once the sample is loaded into memory, API hooking is performed by in-line code overwriting. The sample is executed in a suspend mode and then all loaded DLL's API functions are overwritten. Hence, CWSandbox collects all called functions and their related parameters. Then, it generates a high-level report about activities and malware analyst can quickly follow them. Since it collects data in user mode, low level operations and undocumented function calls can not be captured. Cuckoo is an open source analysis system and relies on virtualization technology to run a given file [4]. It can analyze both executable and non-executable files. These activities including pre-defined Win API functions and their parameters are monitored and captured by its user-space API hooking technique. Owing to the fact that it runs at user level, malware can easily notice presence of the analysis attempt causing to change its behaviour. Capture-BAT is another dynamic analysis tool developed by New Zealand chapter of honeynet.org [5]. Capture-BAT monitors process, registry, and file activities at kernel level, and it captures network traffic using winpcap library. Furthermore, it offers selection of events through its filtering interface that can be used by the analyst to prevent noisy events to be captured. Since Capture-BAT is not an automated malware analysis system, serious concerns exist on whether it can efficiently handle the high penetration of new and existing malware. To the best of our knowledge, dynamic analysis systems still use old versions of Windows OS, for instance Cuckoo merely employs Windows XP and 7, as an underlying analysis environment. However, computer owners generally prefer to upgrade their OS to the newest version. Therefore, existing systems may fail at analysing malware targeting new versions of Windows OS. Reliability and availability of malware analysis scheme may be a major concern subject to the release of new Windows OS version. Consequently, the

next generation of dynamic malware analysis solutions should be adaptable to the future versions of OS. This chapter is an extended version of [6]. In this chapter, kernel callback mechanism is integrated which allows detailed view of run-time events based on defined conditions on a system basis and new malware samples with their targeted features are elaborated and evaluation study is enriched. This chapter is organized as follows: The implementation details of VirMon components and their functionalities are elaborated in Sect. 2. In Sect. 3, two real-world malware samples captured in the wild are used to illustrate the effectiveness and analysis results about monitoring the malware activities. Some conclusions are given in Sect. 4.

2 Design of VirMon: System Components and Functionalities

Analysis machine components include mini-filter driver and driver manager. They are responsible for reporting host-based process, registry, and file system activities performed by the analysed file, see Fig. 1. Activities of the process (or processes) initiated by that executable and (if any) child process (i.e., additional

Fig. 1. Overview of the analysis machine components (e.g., filter drivers)

process created by some of these processes) are monitored by means of the kernel callback functions being embedded in kernel driver. API hooking is one of the preferred methods for dynamic malware analysis. We use kernel callback mechanism, which provides detailed view of run-time events based on defined conditions on a system basis [7]. To be able to use callbacks, a kernel driver needs to be built. Basically, this driver, also known as mini-filter driver, intercepts all IRP requests made by an application and decides whether allowing or refusing these operations according to the given rule set. The portability is followed by the claim of Microsoft, which states that kernel-callback mechanism is reliable and compatible with all versions of Windows including their 64 bit versions. Malware creates new process or changes an existing one in order to run its actions without being detected. To obtain the information about run-time events provided by the callback mechanism, related functions need to be called with their relevant parameters. For process monitoring, the "PsSetCreateProcessNotifyRoutine" function allows the mini-filter driver to monitor changes applied to the running processes. Many malware uses Windows registry to gain persistent access to the system. Attackers can use these registry keys to gain authority over the OS and persistent access to it. When an event is occurred on the registry, "CmRegisterCallback" function can provide related information to the mini-filter driver. To track registry events, one needs to identify which actions to be monitored in the driver via some self-explanatory constant values (e.g., *RegNtPostCreateKey*, *RegNtPreDeleteKey*, *RegNtEnumerateKey*). In VirMon, registry operations about *OpenKey*, *CreateKey*, *DeleteKey*, *SetValueKey*, *DeleteValueKey*, *QueryValueKey* and *EnumerateKey* are monitored. Malware copies itself or its variants to various locations in the file system and then adds a registry key to start automatically while booting. "FltRegisterFilter" function along with its callback actions can be used to monitor file system activities on the system. Like registry monitoring, the actions to be tracked have to be addressed in the driver accompanied by some constant values (e.g., *IRP_MJ_WRITE*, *IRP_MJ_READ*, *IRP_MJ_QUERY_INFORMATION*). In VirMon, to avoid redundant and distracting file operations, we consider only read, write, and delete events performed by the tracked processes. Network components are responsible for reporting network activities of the analysed file. Malware needs to connect very often to the C&C servers to send confidential information collected from compromised machines or to receive C&C servers' commands. This bi-directional communication makes analysis of the network activities of malware as an inevitable requirement to be fulfilled by malware researchers. In VirMon, we use different network solutions, such as VLAN, VPN, IPDS, and firewall to monitor network activities of suspicious files. Intrusion Prevention and Detection System (IPDS) is a network security solution monitoring network traffic and system activities [8]. In VirMon, an IPDS is introduced to prevent possible networks attacks caused by suspicious files in the system. This secure scheme fulfils the analysis requirement of the malware analyzer by giving an opportunity to acquire all network events including the requested web pages, downloaded files by malware. Suricata, an open source IPDS solution, can prevent malicious attacks such as distributed denial of service (DDoS), port scanning and shell codes [9]. It can

also extract files and HTTP requests from live network traffic. In order to circumvent network attacks caused by malware sample under dynamic analysis, we use Suricata as one of the IPDS component. Bro [10] is different from the typical IPDS since it can not block the attacks and does not rely on network signatures but it enables monitoring all network traffic. It supports well-known network protocols, extracts related information from network packets and exposes network activities at high-level [11]. In the hierarchy of VirMon, Bro runs on IPDS server and reads network interface. To extract files from live network traffic, a custom script compatible with HTTP, FTP, IRC, SMTP protocol is created. The developed script logs hostname, URL, filename, file type, and transport layer information (e.g., IPs and ports) to a file which is parsed periodically for storing these information in database. Meanwhile, if the files extracted by Bro Engine have not already been dissected beforehand, they are queued into application server's priority queue. The open source Oracle Virtual-Box [12] is chosen as the virtualization infrastructure to host and to deploy malware analysis machines. Virtual-Box supports both 32 and 64 bit CPU. Virtual-Box also provides accessibility features such as remote machine management, display of multiple remote machines via web interface, and offers command-line interface (VBoxManage) for automated tasks. The application server is responsible for the management of the malware analysis processes. It assigns an analysis machine to the submitted file. It collects the file activities from analysis system components. Then, it formats the collected data and stores them in a database. After the analysis operations are completed, the application server commands the analysis machines to be restored to a clean state.

3 Evaluation

The procedure followed by VirMon to analyze a submitted file, and the interaction between the application server and the analysis machine is plotted in Fig. 2:

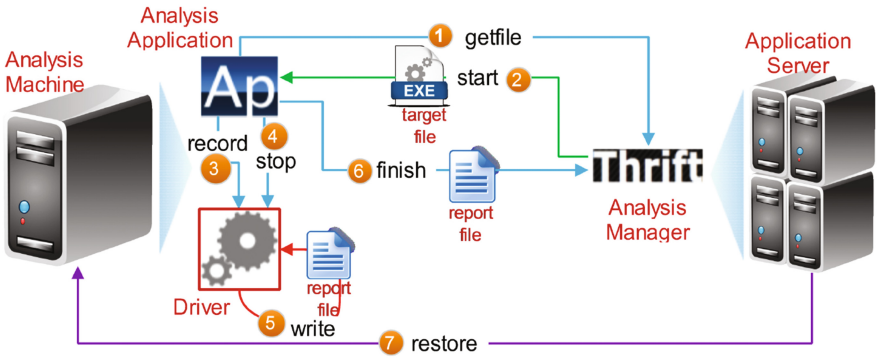

Fig. 2. Information flow and interactions between application server and analysis machine

- Analysis application sends a request to the application server to start a new analysis process.
- The application server chooses a file in its priority queue and sends it to a analysis machine.
- The analysis application injects codes into explorer.exe process. This process executes submitted file in the suspended mode and then explorer.exe writes PID of this recently created process to a shared memory. Subsequently, the analysis application reads PID info from the shared memory, sends a message involving this information to the driver and a request about recording the events initiated by this process.
- The analysis application waits until the analysed process exits or a timeout of 3 min occurs. Then, it sends a message to the driver to stop recording events.
- The driver stops recording and writes collected events to a log file.
- The analysis application sends the log file to the application server for parsing.
- The application server parses the log file and stores it in the database. Finally, it reverts the analysis machine to the clean state.

We analyze a trojan, named as Hesperbot [13] detected on August, 2014. This trojan is focused on stealing banking account information to be used towards unauthorized money transfers. The attackers social engineer victims to execute attached files by sending e-mail which looks like it is originated from one of the service providers in Turkey. The analysis of hesperbot is executed automatically on a Windows 8 Ultimate 64-bit OS. Since the number of events including system dll file and registry accesses gathered from VirMon for this malware is too high (10000+), only important events occurred on the system are displayed in Table 1. The intention of the malware sample can be easily derived from this table. When the sample is executed, the process created by explorer.exe creates a new process entitled "fatura_874217.exe" having the same name in its directory. In Turkish, "*fatura*" means "*the bill*". This technique, named as process hollowing, [14,15], has been recently used by malware to hide itself. Then, the process created by explorer.exe terminates itself. The fatura_874217.exe process created by hollowing technique, creates %APPDATA%\Sun and %APPDATA%\yseszpkf directories and drop randomly named binary files under them. To be hidden, the fatura_874217.exe process creates new explorer.exe, which in turn is used to carry out remaining activities, download configuration files from C&C server, drop new executable and writes it to the auto-start line in the registry, respectively. This analysis shows that the VirMon dynamic malware analysis system successfully collects the run-time behaviours of the file sample.

3.1 VirMon Compatibility on Windows 10

In order to show that VirMon (e.g., its mini-filter driver) is adaptable to the newest version of Windows OS, we conducted functional testing of developed mini-filter driver on Windows 10. This test is based on the analysis results of the recent malware sample, known as cyrptolocker, a variant of ransomware, that encrypts sensitive documents on the infected machine and forces to pay a ransom

Table 1. Important run-time activities of a trojan

Time	Event	Process	Detail
26/08/2014 2:48:44.423	Create Process	C:\Windows\explorer.exe	fatura_874217.exe (analysed sample) MD5:186C097B9D85B3501EFCC4D8D374AFE1
26/08/2014 2:48:55.790	Create Process	%Desktop%\fatura_874217.exe (analyzed sample)	fatura_874217.exe (analyzed sample)
26/08/2014 2:48:55.920	Terminate Process	fatura_874217.exe (analyzed sample)	-
26/08/2014 2:48:56.264	Create Folder	fatura_874217.exe	%APPDATA%\yseszpkf
26/08/2014 2:48:56.269	Create Folder	fatura_874217.exe	%APPDATA%\Sun
26/08/2014 2:48:56.468	Create File	fatura_874217.exe	%APPDATA%\yseszpkf\yqoletyz.dat
26/08/2014 2:48:56.687	Create File	fatura_874217.exe	%APPDATA%\Sun\yqoletyz.bkp
26/08/2014 2:48:57.299	Create Process	%Desktop%\fatura_874217.exe (analyzed sample)	C:\WINDOWS\system32\attrib.exe
26/08/2014 2:48:57.301	Terminate Process	%Desktop%\fatura_874217.exe (analyzed sample)	-
26/08/2014 2:48:57.542	Create Process	C:\WINDOWS\system32\attrib.exe	C:\Windows\explorer.exe
26/08/2014 2:48:57.882	Terminate Process	C:\WINDOWS\system32\attrib.exe	-
26/08/2014 2:48:58.063	DNS Query	C:\Windows\explorer.exe	followtweetertag.com
26/08/2014 2:48:59.272	Send Data	C:\Windows\explorer.exe	https://followtweetertag.com (possibly download config files)
26/08/2014 2:49:01.120	Create File	C:\Windows\explorer.exe	"yqomswoc.bkp, ajukiveq.bkp, yqoletyz.bkp under %APPDATA%\Sun "
26/08/2014 2:49:01.715	Create File	C:\Windows\explorer.exe	"ajukiveq.dat, cfovpdiq.dat, oquthmjk.dat, yqoletyz.dat, yqomswoc.dat under %APPDATA%\yseszpkf"
26/08/2014 2:49:02.012	Create File	C:\Windows\explorer.exe	C:\windows\esem\ohotuzuf.exe (MD5: D082B6AD2F24040E6D651D271823D51C)
26/08/2014 2:49:02.114	Create Reg Key	C:\Windows\explorer.exe	HKLM\SOFTWARE\Microsoft\Windows\ CurrentVersion\Run\qzofpbuk=C:\windows\ esem\ohotuzuf.exe
26/08/2014 2:49:02.345	Send Data	C:\Windows\explorer.exe	https://webislemx.com (for further commands)

Table 2. Run-time activities of the cyrptolocker on Windows 10

Time	Event	Process	Detail
29/11/2014 16:15:53.478	Create Process	C:\Windows\explorer.exe	%DESKTOP%\fatura_892738105.exe (MD5: 76387075C90533AAD14E82A5D94E8486)
29/11/2014 16:15:53.678	Create Process	%DESKTOP%\fatura_892738105.exe	%DESKTOP%\fatura_892738105.exe
29/11/2014 16:15:53.755	Terminate Process	%DESKTOP%\fatura_892738105.exe	-
29/11/2014 16:15:53.964	Create Folder	%DESKTOP%\fatura_892738105.exe	%APPDATA%\ytivyteqyfypequs
29/11/2014 16:15:53.973	Create File	%DESKTOP%\fatura_892738105.exe	%APPDATA%\ytivyteqyfypequs\01000000
29/11/2014 16:15:54.055	Create Process	%DESKTOP%\fatura_892738105.exe	%WINDOWS%\explorer.exe
29/11/2014 16:15:54.888	Terminate Process	%DESKTOP%\fatura_892738105.exe	-
29/11/2014 16:15:54.906	Create File	%WINDOWS%\explorer.exe	%WINDOWS%\whdhufel.exe (MD5: 76387075C90533AAD14E82A5D94E8486)
29/11/2014 16:15:54.925	Create Reg Key	%WINDOWS%\explorer.exe	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ ebotigob=%WINDOWS%\whdhufel.exe
29/11/2014 16:15:55.162	Create File	%WINDOWS%\explorer.exe	%APPDATA%\Microsoft\Address Book
29/11/2014 16:15:55.162	Create Process	%WINDOWS%\explorer.exe	%SYSTEM%\vssadmin.exe
29/11/2014 16:15:55.198	Create File	%WINDOWS%\explorer.exe	%APPDATA%\Microsoft\Address Book\user.wab
29/11/2014 16:15:55.508	Terminate Process	%SYSTEM%\vssadmin.exe	-
29/11/2014 16:15:59.066	Create File	%WINDOWS%\explorer.exe	“02000000, 03000000, 04000000, 05000000 under %APPDATA%\ytivyteqyfypequs\”
29/11/2014 16:15:59.466	DNS Query	%WINDOWS%\explorer.exe	IT-NEWSBLOG.RU
29/11/2014 4:15:59.786	Query Directory	%WINDOWS%\explorer.exe	C:*
29/11/2014 4:15:59.900	Create File	%WINDOWS%\explorer.exe	Start to encrypt all files located under C:*
29/11/2014 16:16:00.068	Send Data	%WINDOWS%\explorer.exe	https://IT-NEWSBLOG.RU
29/11/2014 16:16:01.951	Send Data	%WINDOWS%\explorer.exe	https://IT-NEWSBLOG.RU
29/11/2014 16:16:02.859	Send Data	%WINDOWS%\explorer.exe	https://IT-NEWSBLOG.RU
29/11/2014 16:17:23.564	Create Process	%WINDOWS%\explorer.exe	C:\Program Files\Internet Explorer\iexplore.exe
29/11/2014 16:17:24.863	Create Process	C:\Program Files\Internet Explorer\iexplore.exe	C:\Program Files\Internet Explorer\iexplore.exe
29/11/2014 16:17:26.933	Send Data	%WINDOWS%\explorer.exe	https://IT-NEWSBLOG.RU
29/11/2014 16:17:27.186	Send Data	%WINDOWS%\explorer.exe	https://IT-NEWSBLOG.RU
29/11/2014 16:17:27.186	Send Data	%WINDOWS%\explorer.exe	https://IT-NEWSBLOG.RU

to make them usable again. Since other system components of VirMon work independently, it has been sufficient to install the developed mini-filter driver into the analysis machine to integrate new OS. Accordingly, we successfully installed VirMon's mini-filter driver on Windows 10 without any need to modify or build driver's code. Table 2 shows the summarised run-time activities of the cyrptolocker still observed by VirMon during its analysis. Cyrptolocker malware sample (see [16]) was active in the wild and a signature has not yet been created at the time of writing this paper. This malware's activities show that it uses the process hollowing technique as in the previously analyzed sample (i.e., hesperbot sample). Then, it probably searches specific file types under C:\drive to encrypt and makes them unusable unless one does not have the description key. It sends some information to its C&C server and asks ransom from its victim user.

4 Conclusions

In this paper, the virtualization-based dynamic malware analysis system and its components is presented. A mini-filter driver is built to monitor run-time activities of the file to be analyzed. Since kernel-callback scheme is reliable and compatible with all versions of Windows OS, analysis machine can support all Windows OS versions. In addition to its portability feature, the design supports virtualization and scalability, for instance the average rate of analysis can be adjusted by the virtualization approach. Two recent malware samples captured in the wild are analyzed to illustrate the portability feature and analysis success of the presented dynamic analysis system. The activities of the analyzed samples are extracted accurately and details of each activity are given with the timestamp, event and process description, which enhances readability of the analysis.

Acknowledgements. The authors gratefully acknowledge the support of Galatasaray University, scientific research support program under grant #16.401.004.

References

1. Internet Security Threat Report (2016). Symantec: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf. Accessed 15 June 2016
2. Sukwong, O., Kim, H., Hoe, J.: Commercial antivirus software effectiveness: an empirical study. Computer **44**, 63–70 (2011)
3. Willems, C., Holz, T., Freiling, F.: Toward automated dynamic malware analysis using CWSandbox. IEEE Secur. Priv. **5**, 32–39 (2007)
4. Cuckoo Foundation, Cuckoo Sandbox. http://www.cuckoosandbox.org/. Accessed 1 June 2016
5. Seiferta, C., Steensona, R., Welcha, I., Komisarczuka, P., Endicott-Popovskyb, B.: A behavioral analysis tool for applications and documents. Digit. Invest. Int. J. Digit. Forensics Incident Response **4**, 23–30 (2007)

6. Tirli, H., Pektaş, A., Falcone, Y., Erdogan, N.: Virmon: a virtualization-based automated dynamic malware analysis system. In: The Proceedings of the 6th International Information Security & Cryptology Conference, Istanbul, Turkey, pp. 1–6 (2013)
7. Microsoft Corporation, Writing Preoperation and Postoperation Callback Routines. https://msdn.microsoft.com/windows/hardware/drivers/ifs/writing-preoperation-and-postoperation-callback-routines. Accessed 1 Mar 2013
8. Lazarevic, A., Kumar, V., Srivastava, J.: Intrusion detection: a survey. Massive Comput. **5**, 19–78 (2005)
9. Open Information Security Foundation, Suricata IDS. http://suricata-ids.org/. Accessed 1 Jan 2012
10. Bro Project, The Bro Network Security Monitor. http://www.bro.org/. Accessed 15 Feb 2014
11. Chen, B., Lee, J., Wu, A.S.: Active event correlation in Bro IDS to detect multi-stage attacks. In: The Fourth IEEE International Workshop on Information Assurance (2006)
12. Oracle, Oracle VM Virtual Box. https://www.virtualbox.org
13. Hesperbot malware sample: Google Corp., Antivirus scan results for 186c097b 9d85b3501efcc4d8d374afe1. https://www.virustotal.com/en/file/a34f954ffb49f5c0b8f42376e062971284c9bec864e1d90a7e8d2910ae7c2077/analysis/
14. White, A.: Identifying the unknown in user space memory. Institute for Future Environments Science and Engineering, Faculty Queensland University of Technology, pp. 138–140 (2013)
15. Ligh, M.H., Adair, S., Hartstein, B., Richard, M.: Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code. Wiley Publishing Inc, Indianapolis (2011)
16. Cyrptolocker malware sample: Google Corp., Antivirus scan results for 76387075c 90533aad14e82a5d94e8486. https://www.virustotal.com/en/file/09fe21dd9561603217cc8b419f01c7996b1440aa3e64967f136e38e7f306d625/analysis/