2010 International Conference on Intelligent Network and Computing (ICINC 2010)

Penetrating the Virus Monitoring and Analysis System Using Delayed Trigger Technique

Fauzi Adi Rafrastara
Faculty of Information and Communication Technology
University of Technical Malaysia Melaka
Melaka, Malaysia
e-mail: fauzi_adi@yahoo.co.id

Mohd. Faizal Abdollah
Faculty of Information and Communication Technology
University of Technical Malaysia Melaka
Melaka, Malaysia
e-mail: faizalabdollah@utem.edu.my

***Abstract*— Virus Monitoring and Analysis System (VMAS) is generally used for monitoring and capturing virus behavior, and it can produce a report analysis which can be used by expert user to learn virus activity. There are several tools which have this capability, such as: Joebox, ThreatExpert, CWSandbox, and Sysinternals. Turns out, these tools are not fully perfect in analyzing the virus behavior. Therefore in this paper, we propose a technique to defeat such tools, by exploiting the limitation of VMAS in term of time monitoring, by introducing a new virus exploiting technique called Delayed Trigger Technique (DTT).**

Keywords- Virus Analysis; Behavior Monitor

I. INTRODUCTION

The existence of computer virus is undebatable making worry for computer users in all over the world. Indeed, there is an antivirus software that can be utilized to detect, prevent, and kill the virus from outside. But this is only for common users that do not care about behavior of virus in details. For particular users, especially those who interested to learn and analyse the virus behavior, they should use Virus Monitoring and Analysis System (VMAS), in which it can capture all virus activity accurately and generate the analysis report [1][2][3]. Further, by reading this report, expert users will be able to eliminate the virus from the PC and recover the Operating System [3]. There are several tools which capable to perform this things, e.g. Joebox [4], ThreatExpert [5], CWSandbox [2], and Sysinternals [6]. Although they can capture the behavior and produce the report, but they still have a limitation especially in term of time monitoring, that is a limited time frame which used to monitor and analyse as many suspicious binaries as possible [1].

This research focuses on VMAS which uses behavior or dynamic analysis, rather than static analysis. Moreover, this research pays more attention on the monitoring time which owned by VMAS to monitor and capture virus activity. This is because it could be a weakness of such system to be defeated. In this paper, we perform an experiment that will show how DTT capable to trick VMAS.

Section II of this paper describes the background of this research. Section III explains what Delayed Trigger Technique is, and an overview about this technique. In section IV, we perform the testing phase and followed by the result analysis in section V. The conclusion and future works will be discussed in section VI.

II. BACKGROUND

VMAS is a tool which concerning on virus behavior analysis and it generally can produce the report that showing the analysis result comes from data captured [1]. Actually there are two common methods which used in VMAS to perform analysis process, namely static and dynamic analysis.

Static analysis is performed by decoding the virus file, and analysing the code one by one, without actually executing the file [7]. Static analysis is less popular compared to dynamic analysis, because malware including virus is usually already well-protected to avoid static analysis, so it is hard to be disassembled [1][8]. This statement is also supported by [7], in which on their experiment, they found that approximately 90% of virus binary code cannot be fully disassembled by state of art disassembler.

On the other side, dynamic analysis is an opposite technique of static analysis. This technique will not touch the code at all, but it executes the malicious file instead, and observes behavior of the virus directly [9]. There are several general steps during conducting monitoring and analysis for virus behavior [1][2][3][10][11]. The first step to do is by putting the virus into a controllable environment, and tests it directly. The virus will be monitored for a certain time along. During this time, all activity of the viruses will be captured until the monitoring process is stopped. All information retrieved will be put into a single report, and this report is what needed by user to analyse virus behavior.

The problem here is, by using this dynamic analysis, there is a limited time in monitoring phase [1]. This restriction actually is intended in order that the monitoring process will not take a long time [1]. Turns out, this time limitation led to a new hole that vulnerable to be attacked. Further, virus maker can utilize this hole to trick and penetrate the system.

There are several researchers that discussed about VMAS as well as the comparison among VMASs [1][2][3][10][11]. Their research is actually to improve the performance of VMAS by comparing each other, but they never focus on penetrating VMAS to know how secure it is. In this paper, we attempt to find the hole of VMAS by attacking it directly. The next section will discuss about the technique which used in our experiment.

978-1-4244-8270-2/10/$26.00©2010 IEEE

III. DELAYED TRIGGER TECHNIQUE

Delayed trigger actually is not a new term in computer security world. This technique originally was used by worms to do "logical bomb" aggression into computer system, by replicating themselves to get a maximal dissemination [12]. The author of [13] also stated that delayed trigger is widely used by internet worms to lead a Trojan horse condition, in which they will attach themselves to the benign software and start to attack in the certain condition. Even though it was already used by worms, but actually they do not tend to use this technique to trick the VMAS.

In this paper, we focus to utilize delayed trigger technique to penetrate the security part of VMAS which uses dynamic analysis method. Dynamic analysis method is generally used by VMAS to analyze the virus by putting it into the controllable environment [2][3][11] (i.e. sandbox or virtual machine). Inside the sandbox, the virus will be executed and system will capture all activity of viruses by trapping all system function which called by virus during monitoring time [1][2][3][11]. There are two approaches which used by VMAS to limit the monitoring time [1]. First, it will always monitor the virus activity until the process end, including the process done by its children. Second, if the process is too long and reach the timeout limit, the system will stop it immediately. As mentioned by authors of [1], the tolerance time which given to VMAS during monitoring session is maximal four minutes long. Unfortunately, besides it can give the definite time for VMAS and avoid the very long time scanning just to scan one virus file, this restriction could become the great weakness of VMAS as well.

These two approaches can be tricked by virus which has Delayed Trigger Technique (DTT). It just pretends to act like benign file, and does not do some suspicious things for a certain time, with aim to trick the VMAS until it is truly free from VMAS monitoring. It could stay in idle condition or do some normal activities without doing malicious or suspicious one, i.e. touching the system files or duplicating itself, along five minutes or more. During that time, VMAS will only detect this file as a benign file as it does not do anything suspicious. When the virus is tested in the real environment, it still acts like normal file. But after certain minutes, the logical bomb will be detonated immediately, and unfortunately the virus is already free from VMAS monitoring.

Based on the experiment which performed by [1], they conclude that virus will attack the target as soon as possible, and it means that the virus attack will be performed soon after they reach the target. It should be done immediately to avoid user doing some prevention things, i.e. turning off the computer and installing antivirus, which can protect the computer before virus starts to attack. It looks like opposite of delayed trigger technique, in which virus no need to attack the target in a hurry. As mentioned before, it could be the problem for virus itself if not immediately attacks, because it can be detected first by antivirus or user, or losing the chance to paralyze the computer target.

This problem actually can be overcome by completing one important task, which is by modifying startup configuration either through registry, autoexec.bat, win.ini, or startup folder, so that it can be run every time computer starts up. Such modification is not categorized as a suspicious activity except if immediately followed by accessing system files [14]. But it is remains a virus, that always have the bad intention. And it will start to attack the target, after several minutes as determined by virus writer.

Fig. 1 shows the overview of DTT. This is started from the virus with DTT that executed. In the first minute, this virus tries to add itself to the startup list. As stated in previous paragraph, this activity aims to execute the virus file every time when computer rebooted. It also serves to anticipate if user suddenly turning off the computer soon after the virus is executed, but not yet started to attack. Next, in the second minute, the virus performs in idle mode. It will do nothing to give the delayed trigger before virus start to attack. This condition is also done in the third and fourth minute. So, during the idle mode, VMAS will not detect any suspicious things. Idle mode is also used to finish the timeout limit which owned by VMAS. By performing idle mode until timeout limit, VMAS will report only one activity which done by virus during monitoring time, which is adding itself to startup list. So, after timeout limit is reached, in the fifth minute or more, the virus can start to perform malicious activities without successfully captured by VMAS.

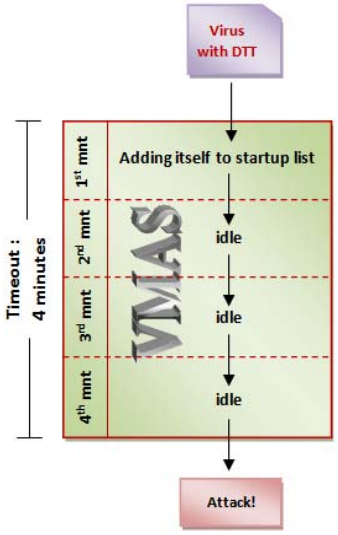

Figure 1. An overview of Delayed Trigger Technique (DTT)

After we created the virus with DTT included, furthermore we perform testing process and it will be discussed on the next section.

IV. TESTING

For this experiment, two viruses were created. These viruses have same attack technique actually, but different in term of availability of DTT inside them. In order to get the clear results, the developed virus here only focuses on attacks against windows registry. This is because registry is one of the popular characteristics of a computer virus, and that's why it always be monitored by VMAS or even antivirus software [1][2][3][10][15]. The first virus here tries

to set six key values as depicted in Fig. 2. In this virus, there is no DTT included. Each process will be performed soon after the previous process is executed. After execute the sixth process (Reg F), the process will be stopped automatically.

The second virus also will infect six registry keys like the first virus does, but it has DTT included with five minutes timeout limit. Five minutes is taken because it is fulfil the requirement of DTT in which the timeout limit should be more than four minutes. In this virus, DTT is located in between Reg A and Reg B as depicted in Fig. 3. The process will be started by executing Reg A, but not directly execute Reg B after that, since there is a DTT here. The idle process will run for four minutes. Next, Reg B, Reg C, Reg D, Reg E, and Reg F will be executed soon after the timeout limit of DTT is finish. Reg A should be there since it can be a trigger to keep the virus run in startup mode and locating itself into sleep or idle mode during timeout limit as a delayed trigger. But after timeout limit and after free from VMAS monitoring, however all parts of viruses will be executed and start to attack the computer.

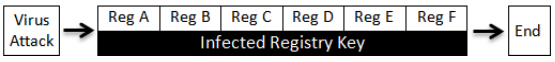

Figure 2. The flow of the first virus attack (without DTT)

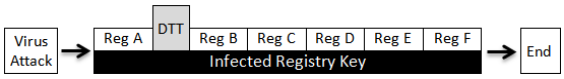

Figure 3. The flow of the second virus attack with DTT

After both viruses created, the next step is to test them by putting them into VMAS. There are several tools which have this capability, such as Joebox [4], ThreatExpert [5], CWSandbox [2], and Sysinternals [6]. Here, the viruses will be tested inside the Joebox and ThreatExpert, and waiting for the analysis report from them. The next section will discuss about the result derived from this experiment.

V. RESULT

By conducting the experiment as discussed in the previous section, it will come up with the report which able to prove that DTT technique turns out can be used to trick the VMAS.

In the snippet report from joebox that shown in Fig. 4, it shows that the first virus performing six changes on registry keys. This data point out that all activity of this first virus can be recorded successfully by Joebox. Meanwhile, on the second report as depicted in Fig. 5, there is only one activity captured that can be reported by Joebox. When we tested the virus on ThreatExpert, it reported the same result like joebox report. If we perform comparison between the first report and the second one, we find the indication that DTT can work well to trick VMAS. So by this comparison, we can conclude that the second virus which uses DTT successfully tricks the joebox as a VMAS. Therefore, the limitation of VMAS especially in term of time monitoring is vurnerable to be penetrated by using DTT. Moreover, such incomplete information that recorded inside the report, can mislead user that try to analyze the virus based on the report produced.

Key value deleted

+ Key value set

Reputation	Key Path	Name	Type	Data	Completion	Count
1	HKEY_USERS\S-1-5-21-220523388-1935655697-1343024091-1003\Software\Microsoft\Windows\CurrentVersion\Run	Trigger	String	C:\virus.exe	success or wait	1
1	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	Shell	String	Explorer.exe, C:\virus.exe	success or wait	1
1	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	Userinit	String	C:\Windows\System32\Userinit32.exe, C:\virus.exe	success or wait	1
1	HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SafeBoot	AlternateShell	String	C:\virus.exe	success or wait	1
1	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Trigger	String	C:\virus.exe	success or wait	1
1	HKEY_USERS\S-1-5-21-220523388-1935655697-1343024091-1003\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	DisableRegistryTools	Dword	1	success or wait	1

Key value replaced with new

Figure 4. Behavior analysis report of virus without DTT

Key value deleted

+ Key value set

Reputation	Key Path	Name	Type	Data	Completion	Count
1	HKEY_USERS\S-1-5-21-220523388-1935655697-1343024091-1003\Software\Microsoft\Windows\CurrentVersion\Run	Trigger	String	C:\virusDTT.exe	success or wait	1

Key value replaced with new

Figure 5. Behavior analysis report of virus with DTT

VI. CONCLUSION AND FUTURE WORK

There are several tools that able to be used to monitor and analyze the virus behavior. It is important to get the understanding for activity of certain virus by test it into these tools. Even though these tools are capable to analyze and produce the report in details, they are not really perfect since still can be tricked by virus which uses DTT. This is because such tools have time restriction for monitoring process. By using DTT, report produced will not show the complete information of virus behavior. Furthermore, such incomplete information can mislead the users who want to analyze virus activity from the report.

Besides it can be a good way to limit the process so that does not wasting much time, it can be a weak point as well, as it can be utilized by virus to enter the system and performing delayed trigger technique while waiting for the free condition from monitoring process done by VMAS. In the future, we plan to use this technique to implement the fast attack detection technique in network intrusion detection system.

REFERENCES

[1] U. Bayer, E. Kirda and C. Kruegel, "Improving the Efficiency of Dynamic Malware Analysis," Proc. SAC'10, 2010, pp. 1871-1878, doi:10.1145/1774088.1774484.

[2] W. Carsten, H. Torsten, and F. Felix, "Toward Automatic Dynamic Malware Analysis Using CWSandbox," IEEE Security & Privacy, vol. 5, March-April 2007, pp. 32-39. doi:10.1109/MSP.2007.45

[3] Z. FuYong, Q. Deyu and H. JingLin, "MBMAS: A System for Malware Behavior Monitor and Analysis," CNMT'09, Jan 2009, pp. 1-4, doi:10.1109/CNMT.2009.5374613.

[4] (2009) The Joebox website. [Online]. Available: http://www.joebox.org/

[5] (2009) The ThreatExpert website. [Online]. Available: http://www.threatexpert.com/

[6] (2010) Windows Sysinternals. [Online]. Available:. http://technet.microsoft.com/en-us/sysinternals/default.aspx

[7] J. Dai, R. Guha and J. Lee, "Efficient Virus Detection Using Dynamic Instruction Sequences," Journal Of Computers, vol. 4 (5), May 2009, pp. 405-414, doi:10.4304/jcp.4.5.405-414.

[8] C. Kruegel, W. Robertson, F. Valeur and G. Vigna. "Static Disassembly of Obfuscated Binaries," Proc. of the 13th conference on USENIX Security Symposium, 2004, vol. 2004, p. 18.

[9] E. Al Daoud, I. H. Jebril, and B. Zaqaibeh, "Computer Virus Strategies and Detection Methods," Int. J. Open Problems Compt. Math., vol. 1 (2), Sept. 2008, pp. 29-36.

[10] C. Seifert, R. Steenson, I. Welch, P. Komisarczuk and B. Endicott-Popovsky, "Capture - A behavioral analysis tool for applications and documents," Digital Investigation, vol. 4 (1), June 2007, pp. 23-30, doi:10.1016/j.diin.2007.06.003.

[11] U. Bayer, "TTAnalyze: A Tool for Analyzing Malware," Master's Thesis, Vienna University of Technology, Vienna, Austria, Des. 2005.

[12] C. Blaess. (2002) Viruses: a concern for all of us [Online]. Available http://linuxfocus.org/English/Archives/lf-2002_09-0255.pdf.

[13] J. Nazario, Defense and Detection Strategies Against Internet Worms, Norwood, USA: Artech House, Inc., 2004.

[14] C. Nachenberg. (2002) Behavior Blocking: The Next Step in Anti-Virus Protection [Online] http://www.bandwidthco.com/sf_whitepapers/malware/Behavior%20Blocking%20-%20The%20Next %20Step%20in%20Anti-Virus%20Protection.pdf

[15] J. Dai, R. Guha and J. Lee, "Dynamic Instruction Sequences Monitor for Virus Detection," Proc. CSIIRW'08, vol. 288 (18), May. 2008, doi:10.1145/1413140.1413161.