International Journal of Computer Networks and Communications Security

VOL.1, NO.7, DECEMBER 2013, 324–328
Available online at: www.ijcncs.org
ISSN 2308-9830

# Manual Malware Analysis Using Static Method

**NORKHUSHAINI AWANG[1], ARIFIN SALLEH[2] and MOHAMAD YUSOF DARUS[3]**

[1, 2, 3] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Malaysia.

*E-mail: [1]shaini@tmsk.uitm.edu.my, [2]arifin@ump.edu.my, [3]yusof@tmsk.uitm.edu.my*

**ABSTRACT**

Today malware threats represent the greatest challenge to information security. Combat between malware writer and malware researcher never end. Malware writers use a variety of avoidance techniques such as Code Obfuscation, Packing, Anti-Debugging and Anti-Virtualisation Technologies to foil researcher's analysis. On behalf of researchers they try to find out many techniques to defend Information Technology (IT) services from access or stolen by unauthorized parties. Most of the researches perform malware analysis in Virtualisation Technology in the isolation environment because of security issues. This research focuses on analysis malware using static method in operating system environment. Thus, we focus on malware analysis that uses Anti-Virtualisation avoidance technique. Although our platform environment exposed to the threat by malware sample, we protect this environment by using Toolwiz TimeFreeze and window backup image to protect and secure our environment. This research proved that our environment capable to do malware analysis and compare our environment with the virtual machine environment to prove that our analysis more accurate.

**Keywords:** *Malware, Security, Threats, Static Analysis, Dynamic Analysis, Operating System.*

## 1 INTRODUCTION

Now day malware threats were assessed by IT security organizations has been growing more than ten thousand every day. Symantec Internet Security Threat Report (2011) reveals that the total number unique variants of malware in the world in 2011 around 403 million compared to 286 million variants in 2010. By using many avoidance techniques such as self-defending code, packing, anti-debugging and anti-Virtualization techniques has a leading a problems on computer network especially cause of bottlenecks in the network and increased threat of criminal for corporate and individual data. The most challenging for antivirus organization and researcher is about the threat that occurs in computer applications because of the unknown vulnerability or known as a zero-day attack. This attack will take advantage of an application that has issue of security vulnerability. Thus, this research endeavours to discover the best solution by conducting malware analysis. The malware analysis can conduct in many environments or platform such as using the virtual machine environment such as Virtual PC, VMware and QEMU. The online analysis tools like Sandbox or use traditional way with using real machine environment in a secure environment. Choose the right malware analysis environment is very import to make sure the result from analysis can get the accuracy of information about the malware threat.

## 2 RELATED WORKS

Basically there are two techniques that used to conduct malware analysis, Static or code analysis and dynamic or behavior analysis. But Zeltser [1] divided to three techniques such as above and another one is about memory analysis. This analysis will extract artifacts that related to malware program by examining memory. The malware such as Rootkits trying to hide it during malware analysis can identify by using this analysis. Other advantages of this analysis, it can save time and get results immediately when studying the sample of malware in dynamic or static analysis.

Malware analysis environments are most important part of malware analysis to get the accurate result of analysis especially to get the correction about the malware behavioral information. But most

of researchers today preferred to use Virtualization software such as virtual machine to conduct malware analysis [2]. In order to minimize the potential damage of the analysis environment Virtualization technology is very useful for creating control environment [3]. Thus this environment can reduce the costing compared to using real machine.

Malware writers use a variety of avoidance techniques to foil researchers during the analysis the malware especially when doing reverse engineering analysis and forensic analysis. Basically, there are four common avoidances or anti-analysis techniques such as Anti-Virtualization, Packing, Code Obfuscation and Anti-Debugging Techniques [3, 4, 5, 6].

In year 2011, in research [7] try to prove that top 10 malware list given by Microsoft that this malware not aware about the virtual environment during malware analysis. The reason because of the most of current infrastructure now a day like server moving towards Virtualization today and not only use by researchers tools. But in 2012, Black Hat USA [5], published the first result of a security research project calls Dissect || PE that different point of view about the technique of malware that bypass Virtualization technology.

## 3 TESTING AND ANALYSIS

In this research, we focused on analysis malware using static and in operating system environment. Figure 1 shows the process during conduct malware analysis in the operating system environment for static analysis. In the first stage, we must provide a secure environment with open the Toolwiz TimeFreeze tools. Then, run or open IDA Pro to analysis malware code. The python script is use to find out the instruction that the malware bypass Virtualization technology. Lastly, we record the analysis finding and stop Time Freeze and reboot the PC.

Table 1 shows the malware sample base on categories of malware. In this research, 20 samples of malware from different categories are chosen and analyzed in IDA Pro tools. The purpose of this analysis is to identify which malware may bypass or refuse to run in a virtual technology environment like VMware.


*Fig. 1. The Process Implementing Setup OS Environment for Static Analysis.*

*Table 1: Malware Sample*

| Categories of Malware | File Name | MD5 |
|---|---|---|
| Virus | Virus.Win32.Virut.av | 618f9fa051f431d73ed78102625c9feb |
| | Virus.Win32.Texel.A | 2bfd26a3b18f12fc503d2e1bde263d6c |
| | Win32.Hawey.A | 26dfda6b503351ba682700665e221fa1 |
| | W32.Virut-3 | 863f15b7b6c67cbce3d27cd73b2ac86c |
| | Virus.Win32.Enerlam.c | 720937f20ce0b2de51e208cc261c41f3 |
| Trojans | Trojan.Dropper.Agent!IK | ebc33f1bd945a281ab8d23848d679b39 |
| | W32Autorun.KA | d1d01439bf404998853790193cc0c79c |
| | Win32Malware-gen | 429bc88c3386819a9a70e461e76dd59c |
| | Trojan.Genome.njip | 3F4F40D772887C3D1FE499CACA06FECB |
| | W32Trojan2.JRCA | 6BF1B791A874DBD39040077555CB0D88 |
| Worms | W32Conficker!Generic | 0921282d4ed6008aa7c04e268d8367ae |
| | W32.Virut.Gen.D-175 | 7832d34b64973bee2c805721efb4ad3b |
| | W32Conficker | 9812D086BB0373209C8E09FA5A7F7B70 |
| | Worm_win32_autorun_pga | 4a4c8f3a3e8c2c019f54a39664f4eab9 |
| | Worm.Blaster.A | fc14eaf932b76c51ebf490105ba843eb |
| Bots | Backdoor_32_sdbot_fmf | 602435b0571b0da5006a9313f2ae72d3 |
| | Gbot.2764 | 884575b63995839bcaf791fa8f5aad7a |
| | PHPPbot.A | b0c4d5b9d5c3a4391ae4ebc08ce1277b |
| | Sniper Bots Makerv2.exe | 18b38a7d995ba09c53e02c354ef4527e |
| | W32MalwareF.BOTS | 63a9d088985f82890f8d219f9f920a01 |
| | W32Zbot.AMG | c3e375deabb6ca2c95a6e0f2a3ed5677 |

Figure 2 shows about the process of analysis results in order to generate the output along with the conclusion that came out from all the findings. The result will be compared by generating the report in table and graph forms.

Table 2 shows the summary of the static analysis result from the type of malware which are Virus and Trojan and Table 3 for Worms and Bots. By using IDA Pro and python script, it is found that three of five malware samples have contained potential anti-VME instructions for the Virus. Instead, none of the instruction was found from Trojan. The sample type of Virus such as Virus.Win32.Virut.av, Win32.Hawey.A and W32.Virut.3 found the anti-VME instruction in memory instruction artifact (sldt) and VME Communicational Channel by using port "IN". Research in [5], notice that 99.45 % of the VME detective is coming from the "IN" port

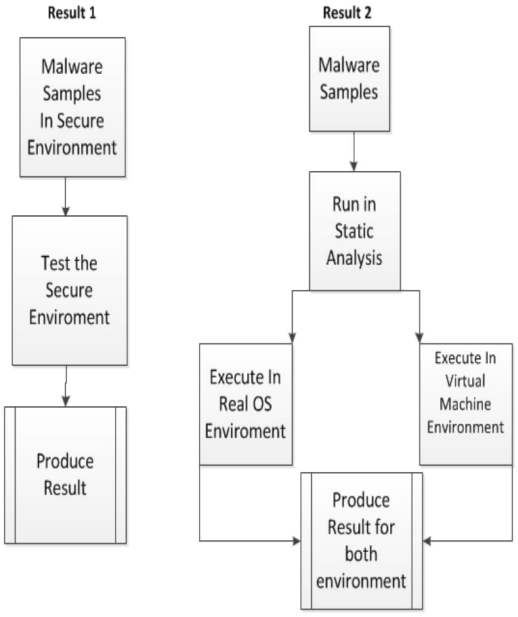


*Fig. 2. The Process of Analysis Results*

*Table 2: The Static Analysis Result (Virus and Trojan)*

| Types of Malware | File Name | Results |
|---|---|---|
| Virus | Virus.Win32.Virut.av | One Potential Anti-VME instruction "Sldt" found at 0042303B location. |
| | | May file was packed or modified due to make it more difficult to analyze. |
| | | Take time to load. |
| | Virus.Win32.Texel.A | No Instructions of Anti-VME found |
| | | Take time to load. |
| | Win32.Hawey.A | One Potential Anti-VME instruction "Sldt" found 00404467 locations |
| | W32.Virut.3 | Five Potential Anti-VM instructions; 'Sldt" instruction found at 00403288 locations, "In" instruction found at 0040338C, 004033CB, 004033D2 and 00403407 |
| | Virus.Win32.Ener | No Instructions of Anti-VM found |
| Trojan | Trojan-Dropper.Agent!IK | No Instructions of Anti-VM found |
| | W32Autorun.KA | No Instructions of Anti-VM found |
| | Win32Malware-gen | No Instructions of Anti-VM found |
| | Trojan.Genome.njip | No Instructions of Anti-VM found |
| | W32Trojan2.JRCA | No Instructions of Anti-VM found |

*Table 3: The Static Analysis Result (Worms and Bots)*

| Types of Malware | File Name | Results |
|---|---|---|
| Worms | W32Conficker !Generic | Two Potential Anti-VM instructions; "In" instruction found at 100073F5 and 10007CD1. |
| | | May file was packed or modified due to make it more difficult to analyze. |
| | Worm.Virut.Gen.D-175 | Four Potential Anti-VM instructions; "In" instruction found at 0044E0AB, 0044E0B0, 0044E101 and 0044E12B |
| | | May file was packed or modified due to make it more difficult to analyze. |
| | W32Conficker | One Potential Anti-VM instructions; "Sldt" instruction found at 0026766B |
| | Worm_win32_autorun_pga | No Instruction of Anti-VM found |
| | | May file was packed or modified due to make it more difficult to analyze |
| | Worm.Blaster.A | One Potential Anti-VM instructions; "Sidt" instruction found at 004060A5. |
| | | May file was packed or modified due to make it more difficult to analyze. |
| Bots | Backdoor_32_sdbot_fmf | No Instruction of Anti-VM found |
| | Gbot.2764 | No Instruction of Anti-VM found |
| | PHPPbot.A | No Instruction of Anti-VM found |
| | Sniper Bots Makerv2.exe | No Instruction of Anti-VM found |
| | W32MalwareF.BOTS | No Instruction of Anti-VM found |

The static analysis is figuring out in the graph as shown in Figure 5. The result shows Virus and Worms malware types contain the potential malware instruction. From 20 samples of malware, it is found that 35% of malware are having potential anti-VME instruction. 20% comes from Worms, 15% from Virus and none from Trojans and Bots. This result is expected because of the Trojans and Bots are performing as a network malware while the Virus and Worms are the computer malware which are targeted on the computer system.

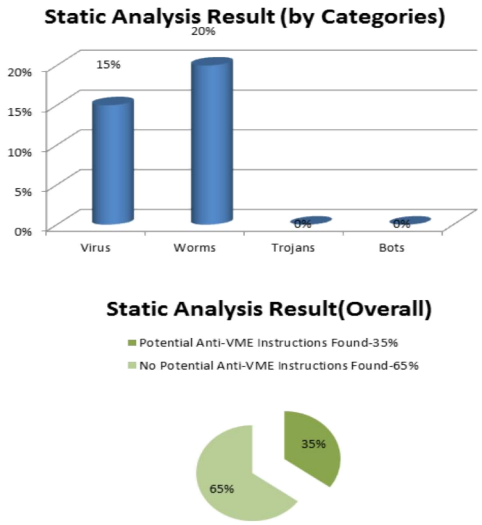


*Fig. 2. The Static Malware Analysis*

## 4 CONCLUSION

This research is to identify which malware may bypass or refuse to run in a virtual technology environment like VMware. Commonly, malware writers include the list of instructions such as memory instruction artifact (sldt) to employ anti-VME technique. From extensive testing, the result shows Virus and Worms malware types contain the potential malware instruction. In future, we plan to using dynamic or behavior malware analysis and automated analysis that involves more samples malware used simultaneously and give more consistent and accurate result.

## 5 REFERENCES

[1] L. Zeltser. (2010,Oct 9), "Phases-Malware-Analysis-Behavioral-Code-Memory-Forensics" Available:http://computer-forensics.sans.org/blog/2010/10/11/3-phases-malware-analysis-behavioral-code-memory-forensics.

[2] L. Sun et al., "An automatic Anti-VMware Technique Application for Multi-stage Packed Malware", 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE), Fairfax, Oct 7-8, 2008, pp. 17-23.

[3] M. Fadli and A. Jantan, "Secure Environment Platform for Host-based Dynamic Analysis using DeepFreeze", International Conference on Computer Application and Education Technology (CCAET2011), Beijing, China, December 3-4, pp. 164-167.

[4] W. Gharibi and A. Mirza, "Software Vulnerabilities, Banking Threats, Botnets and Malware Self-Protection Technologies", IJCSI International Journal of Computer Science Issues, Vol.8.Issue 1, January 2011.

[5] R. Rubigo et al. (2010 July 30), "Scientific but Not Academical Overview of Malware Anti-Debugging, Anti-Disassembly and Anti-VM Technologies", [online]. Available: Http:// http://research.dissect.pe/docs/blackhat2012-paper.pdf

[6] M. Michael and A. Honig, " Practical Malware Analysis ",William Pollock,38 Ringold Street, San Francisco, 2012.

[7] A.Mushtaq (2011,Jan,27)," The Dead giveways of VM-aware Malware" [online]. Available: http://blog.fireeye.com/research/2011/01/the-dead-giveaways-of-vm-aware-malware.html