Limitations of Dynamic Malware Analysis

Lisa Fiander | 4th Year BSc Ethical Hacking Student | University of Abertay, Dundee

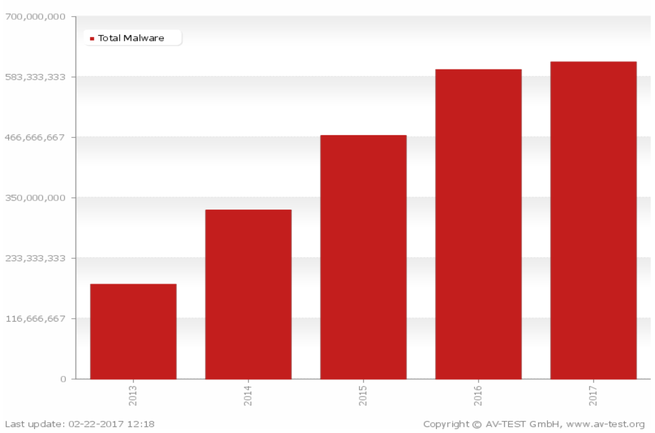
Abstract

With new strains of malware created on a daily basis, techniques used by Anti-Virus software can not be relied on to detect them. Malware analysis is a technique that is used in order to try to identify the new characteristics of the malware samples. The aim of this project is to analyse the limitations of dynamic malware analysis on Windows malware. This was done by setting up three virtual machines to analyse the malware dynamically to examine what effects it had at run time. As well as statically, to analyse it in a stable state and reverse engineering it to find any additional information that may be lost during the previous techniques, such as function calls enabling the malware to connect out to a Command and Control (C2) Server.

Introduction

In this modern day world, the rapid advances in technology have resulted in a large surge of the use of PC's, laptops and smart phones, to mention a few. However, this has also been met by a steep rise is the number of malware outbreaks surrounding these devices. 4000 ransomware attacks have been occurring each day since 2016.

Malware comes in many different forms and complexity. A handful of malware families include Viruses, Trojans, Worms, Adware, Botnets and Ransomware. With the newer strains being more stealthy and complicated. The diagram below shows the total amount of new malware samples, which have been rising since 2013.



With the increase in new malware samples, it is the job of malware analysts to identify these new strains as Anti Virus software cannot be relied on 100%.

Methodology

In order to analyse the limitations of dynamic malware analysis, the following practical steps were completed:

- Create a testing environment of three virtual machines for testing
- Identify suitable malware for testing, in this case the malware used was an IRC Bot as this was my first time reversing malware and newer types can prove a lot more difficult to examine

1.Within the dynamic analysis virtual machine, execute the malware using the tools Regshot, Process Monitor, AutoRun and Wireshark

2. Within the static analysis virtual machine, examine the malware using the tools File Alyzer, CFF Explorer, Strings, PiED, Malware Bytes and StudPE without executing the malware

3. Within the reverse engineering virtual machine, upload the malware sample to IDA and reverse the malware using debugging tools

4. Collect results and compare to find the differences in techniques

Results

Dynamic Malware Analysis	
Registry Key Added	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
Path Added	C:\Windows\rundll23.exe
C2 Server IP	130.211.59.20
IRC Commands sent in packets	PrivMsg, Join, Nick, User
Static Malware Analysis	
MD5	98F9FF31B6A93D8D8DF6CD9A5B06D355
Creation Date	Wednesday, December 19, 2012 4:32:14 PM
Entry Point	00003E63, .text
Language	Microsoft Visual C++ 6.0
AV Identified as	Trojan.zbot
Reverse Engineering	
Function calls relating to C2 Servers	Connect & Send
Execution	Malware would only execute if conditions were met

Conclusion

The aim of this research project was to analyse the limitations of dynamic malware analysis methods. Based on previous research, the expected results were that static malware analysis and reverse engineering would perform better than dynamic malware analysis. Dynamic malware analysis failed to pick up through two techniques that the malware was trying to add values to Windows Registry however, another dynamic malware analysis tool did detect this along with paths that were being used by the malware to hide, but it required a lot of digging around to find that this was happening. The technique was also able to detect that outside connections were being made along with the use of IRC commands. Time stamps were recovered during the process but these were two years behind the actual creation date.

Static malware analysis and reverse engineering were able to detect other interesting information. Including MD5 hashes, correct timestamps, the language the malware was written in, characteristics determined in a stable state, entry points, strings, section alignments, further IRC commands, interesting function calls and again, proof that the malware is attempting to connect out to an external source.

Therefore the conclusion that can be drawn from this project is that although all three techniques returned different but useful information, not one can be deemed as the most effective. The main limitations within dynamic malware analysis during this project is that it returned incorrect timestamps, failed to detect that values and paths were being added to registry in certain tools that were used and was unable to give a full picture of the malwares functionality.

Future Work

Future work surrounding this project will be to continue on with the practical carried out within this project, but with the difference of an unknown malware sample. The sample used within this project was created back in 2012 and therefore has known signatures, however using the same techniques on an unknown sample could prove more difficult but interesting for future developments in this field.

Dynamic malware analysis as a whole is ok as a technique at identifying the characteristics of current malware, however it does have many flaws. Malware samples should be analysed with a mixture of the three techniques discussed.

References

- AVTest, 2017. AVTest. [Online] Available at: https://www.av-test.org/en/statistics/malware/ [Accessed Tuesday 28th March 2017]
- Brianna Gammons. 2017 [Online]. *6 Must-Know Cybersecurity Statistics for 2017 .* Available: https://blog.barkly.com/cyber-security-statistics-2017. [Accessed Monday 1st May 2017]