Research Article

Literati Journals
Empowering Researchers

Detection of Malwares, APTs and Their Propagation in an Enterprise Network Using the Combination of Scoring Model and Process-Network-File Approach

Srikanth P Vasist[1]*, Sailaja Thota[1]

Abstract: With the advances in storage and internet technologies more and more data is moving into digital format which imposes a risk of information theft by hacking using some malicious software and advanced persistent threats(APT). In case of large enterprises, this malicious software are made to propagate within network to obtain as much data as possible and also to reach to the section of most valuable information like patents, finances etc. This paper covers the work done on this area and also proposes a novel approach based on the combination of module scoring and process-network-file relationship to detect such propagations.

Asian Journal of Engineering and Technology Innovation
Volume 4, Issue 7
Published on: 7/05/2016

Cite this article as: Srikanth P Vasist, Detection of Malwares, APTs and Their Propagation in an Enterprise Network Using the Combination of Scoring Model and Process-Network-File Approach. Asian Journal of Engineering and Technology Innovation, Vol 4(7): 80-83, 2016.

INTRODUCTION

With the growing size and importance of digital information, it is becoming increasingly difficult to secure them which in turn becoming more vulnerable to digital hackers. This fact shoots up the necessity of proper information security system in place.

Every organization, may it be small startups or large enterprise, deals most of its information in digital format by storing it in physical disks, private cloud hosted within the organization or on public cloud. But, most of the critical information will reside within the enterprise network because of the obvious security reasons.

The nature of information can be source code, financial data, results, future roadmaps, designs, patent documents etc. Looking at the nature of data it can be safely assumed that a copy of most of the data mentioned above will also reside on the computers of the employees.

Hackers take the advantage of this fact and introduce malicious code into the enterprise network which will eventually try to spread itself with the enterprise by acquiring the credentials from user inputs and gathering IP addresses of other computers and storage.

The malwares while spreading possesses certain behavior like pinging random IPs continuously, logging keystrokes, mounting shares and so on. With all these means it will replicate itself on other computers or storages. These malwares along with the replication will try to steal information from each of the computers and try to send it the hacker.

This paper proposes a novel method to determine the spreading of such malwares by scoring the executables based on the unusual behavior and by analyzing the process, file and network activity of the high scored executables.

LITERATURE SURVEY

The detection of malware propagation gained more importance currently because of the amount of breaches that are being reported. The authors [1] has taken a route of mathematical model by analyzing the amount number of entries at given point of time. They have found out that the number obeys particular mathematical distributions such at specific point of time. Even though the pattern of the numbers obey the models, the time required to detect the malware will take ample amount of time which is not much desirable as it would have done the necessary damage by then. Also, this model would require good amount of real data to give precise results.

The authors of [2], [3] mainly discuss on the nature of the propagation and their analysis by deploying in distributed network nodes. They also explain how the environmental factor and the implementation of the malware hide itself from the detection mechanisms. The data that they obtain indicate only the propagation of the malware only if the malware is previously known. Otherwise, there can be many applications and processes which would want to propagate legitimate software such as

[1]Reva Institute of Technology and Management, Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Near Border Security Bustop, Bengaluru, Karnataka-560064, India.
E-mail: ashwin@revainstitution.org
*Corresponding author

updates and the scalability test data etc. So this is more of a forensic option and would not provide us any detection capability.

There are discussions [3]–[6] on several methods to detect and analyze certain category of malwares like metamorphic malwares by creating a framework based on the behavior. As it is pertaining to special cases of malware behavior it cannot serve the purpose of detecting any generic new malware. Similarly, the static binary analysis of [5] also doesn't provide concrete picture as it can miss many of the behaviors.

SCORING SCHEME

Every file is treated as a module. Every module will have a score associated with it. This score will be used to judge the mischief the module would have done. For e.g. an unsigned executable dropping another executable on to the disk is a suspicious activity. This activity will be associated with a risk value. It is possible that a module is involved in more than one suspicious activity.

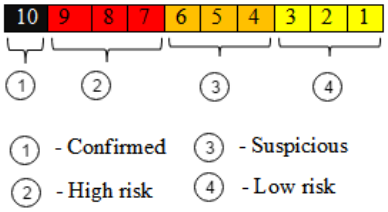

Figure 1: Bit representation of the score.

So, the score is a function of risk values associated with each of the suspicious activity. The score is calculated by setting the appropriate bit fields based on the criticality of the behavior. Fig. 1 shows the logic of bit setting. There are three bits assigned to low(shown in yellow), medium(shown in orange), high(shown in red) and one bit for critical(shown in black). The three bits can hold the maximum number of 7. Any number of event greater than 7 is ignored.

The three bits of all the severity category will be filled as per the number of events of that category occurred. The critical section is an exception with only one bit. This bit, when set,signifies the module is known bad. The engine would have got the information from external sources or by administrator setting. When the critical bit is set, all other bits are ignored and thus the maximum value any module can obtain is limited to 512.

Sample Score Calculation

For e.g. Let us assume there are binaries with yellow=2, orange=2, red=3 and black=0. The bit pattern looks like 0-111-110-110. The value of this bit representation is 502.

PROCESS-NETWORK-FILE (PNF) RELATIONSHIP

The process-network-file (PNF) is a relationship formed based on the access of the resources and network by a process. This information can be critical in case of the detection of the spreading of malware.

For instance, a key logger malware may have two different modules one for logging the keys and the other to try hacking with the obtained password. In this case, the key logger will write the password on to a file or registry or any shared location. The location is termed as file in the name. The two modules, the shared location or object, and the network IP or domain, if it tries to send password out, is treated as a single entity for the purpose of tracking.

BEHAVIORS

The below are the examples of the behaviors exhibited by the modules. The behaviors are categorized by the severity of the action and impact on the system.

1. **Critical**
   - Blacklisted by administrator.
   - Blacklisted entry in hosts file.
2. **High**
   - Suspicious SVCHOST running.
   - IE enhanced security disabled.
   - CMD.EXE copy files to network path.
   - Autorun unsigned uncommon registry startup method.
   - Unsigned EXE dropping another executable on to the disk.
   - An unsigned process creating remote thread to another process.
3. **Medium**
   - Autorun unsigned Active executable.
   - Unsigned EXE hidden.
   - Invalid signature of EXE.
   - Invalid PE compile date.
4. **Low**
   - Writable code section.
   - Beaconing an outside network IP Address.
   - The PE executable is packed.

DETECTION USING SCORING AND PNF

Once the module is scored and the PNF graph is obtained, it is straight forward to do the detection. In the example of key logger, only the key logger will get the high score but the rest of the modules which are associated with key logger might not be scored high because they just read from a file which is not a suspicious operation. With this approach all the modules in the PNF graph of the high scored module will be assigned with the same high score and will be immediately tracked it any of them are present on any machine in the network. In advanced persistent threat, not all necessary modules will arrive at the machine at once. They will be downloaded from the command and control center of the hacker over time. With this approach these kind of attacks can be detected well before the actual attack takes place.

RESULTS

The Trojan "njRAT" is run on a test computer with this detection technique running. Below is the result which shows the behaviors on which the Trojan is detected.

Process Artifacts

The following processes were started when the "Njrat.exe" malware was executed:

- C:\Windows\System32\netsh.exe [high]
- %APPDATA%\msnco.exe [high]

File System Artifacts

The following files were created when the "njrat.exe" malware was executed:

- %APPDATA%\msnco.exe [high]
- C:\WINDOWS\Prefetch\NJRAT.EXE-0AD199D6.pf [low]
- C:\Documents\%USERNAME%\StartMenu\Programs\Start up\28132e8017cb6e82db2c.exe [high]
- C:\WINDOWS\Prefetch\NETSH.EXE-085CFFDE.pf [low]
- [CMD] \.tmp (or when created by the original dropper: "C:\Extracted\.tmp") [low]

Registry Artifacts

The following registry values were set by the "Njrat.exe" malware when it was executed:

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\b6554e5bcfef391ff7a7ffda58092e10 [Value: "[%APPDATA%]\msnco.exe" ..] [high]
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\b6554e5bcfef391ff7a7ffda58092e10 [Value: "[%APPDATA%]\msnco.exe" ..] [high]
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplications\List\[%APPDATA%]\msnco.exe[Value: [%APPDATA%]\msnco.exe:*:Enabled:msnco.exe] [high]

In total there are 7 high, 0 medium and 3 low incident.

As per the scoring system the score of this module is 0 111 001 011 = 459 which is a very high score.

CONCLUSION

A novel approach to detect the malware by assigning scores to the behavior is proposed. A novel concept of establishing the relationship between process, file and network is explained. Using these two techniques a new way of detection of malware propagation in an enterprise network is proposed. This model is self-adaptable to give better and promising results with the increase in the behaviors detected.

Acknowledgment

I express my sincere gratitude to my Principal and Head of the Department, Computer Science and Engineering Dr. Sunilkumar S Manvi, Reva Institute of Technology and Management for providing facilities, encouragement and guidance. I would like to thank Asst. Prof. SailajaThota, Department of Computer Science on her periodic inspection and guidance. I wish to thank Laxmi B Rananavare, Associate Professor and PG Coordinator, Department of Computer Science for providing encouragement.

REFERENCES AND NOTES

1. B. V. A. . G. G. . A. B. . S. G. S. Yu ; Sch. of Inf. Technol., Deaki Univ.,"Malware propagation inlarge-scale networks," in IEEE Transactions onKnowledge and Data Engineering, 2014.
2. A. A. M. . Z. M. M. . F. J. E. Cooke ; Dept. of Electr. Eng. & Comput. Sci.,Michigan Univ., "Hotspots: Theroot causes of non-uniformity in self propagating malware, dependable systems and networks," in DSN 2011. International conference, 2011.
3. S. S. Hansen, T. M. T. Larsen, M. Stevanovic, and J. M. Pedersen, "Anapproach for detection and family classification of malware based onbehavioral analysis," in 2016 International Conference on Computing, Networking and Communications (ICNC), Feb 2016, pp. 1–5.
4. S. Alam, R. N. Horspool, and I. Traore, "Mard: A framework formetamorphic malware analysis and real-time detection," in 2014 IEEE28th International Conference on Advanced Information Networking andApplications, May 2014, pp. 480–489.
5. L. Chen, T. Li, M. Abdulhayoglu, and Y. Ye, "Intelligent malwaredetection based on file relation graphs," in Semantic Computing (ICSC),2015 IEEE International Conference on, Feb 2015, pp. 85–92.
6. S. Das, Y. Liu, W. Zhang, and M. Chandramohan, "Semantics-basedonline malware detection: Towards efficient real-time protection againstmalware," IEEE Transactions on Information Forensics and Security,vol. 11, no. 2, pp. 289-302, Feb 2016.
7. F. C. C. Osorio, H. Qiu, and A. Arrott, "Segmented sandboxing - a novelapproach to malware polymorphism detection," in 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), Oct 2015,pp. 59–68.
8. S. Naval, V. Laxmi, M. Rajarajan, M. S. Gaur, and M. Conti, "Employingprogram semantics for malware detection," IEEE Transactions on InformationForensics and Security, vol. 10, no. 12, pp. 2591–2604, Dec 2015.
9. X. Han, J. Sun, W. Qu, and X. Yao, "Distributed malware detection basedon binary file features in cloud computing environment," in The 26thChinese Control and Decision Conference (2014 CCDC), May 2014, pp.4083–4088.