Methods used in Malware Detection

Christos-Nikolaos Mitsopoulos
December 9, 2018

Abstract

Malware, malicious software that aims to infect and harm a vulnerable computer system without the user's consent, has been on the rise with more sophisticated malicious software being produced on the daily basis. To cope with this, cybersecurity companies and researchers constantly develop better and faster malware detection methods. In this paper, I will analyze the current methods used in malware detection, such as signature-based and behavior-based methods, as well as the future trends expected in malware detection.

Contents

1.Introduction

Malware refers to malicious programs that aim to harm vulnerable systems without the knowledge and consent of the user. Such programs have been in circulation since the dawn of computers and present a significant cybersecurity problem. This is why computer users rely on malware detection engines, such as Antivirus programs to reduce their risk of exposure. These engines are responsible for detecting malicious software present in files (usually as executables, because malware is by definition dynamic, that is, it alters the system adversely). Usually, malware detection engines use a signature-based, or a behavior based approach to detecting malware. None of these approaches are perfect though, as they have several important shortcomings that need to be addressed for faster and more effective malware detection to be possible. In this paper, I will provide an analysis of the current methods used by malware detection engines, and present ways for them to be improved.

But first, what does a malware-detection engine have to achieve? This engine must be able to monitor the system configuration and keep track of any malicious changes, such as persistent calls to access system processes made by a third-party service. It must be able to scan each file before execution in order to classify it as malicious or benign and must also be able to remove the file if found to be malicious. In this paper, I will only focus on the scanning functi of the anti-malware engine.

1.1 To the Community

New malware or variations of existing malicious software are being developed at an alarming rate. For example, this year alone Kaspersky Labs, a cybersecurity company based in Moscow, published a report stating that its web-antivirus detected over 21 million unique malware objects[1] . More and more malware files circulate the internet and their sophistication does not seem to decrease. Therefore, this is a critical time for the cybersecurity world as malware detection engines have to cope with the rapid creation of malware with faster classification systems, in order to avoid system intrusions.

2 Signature-Based Malware Detection

A signature-based approach determines if a file is malicious or not by trying to match bytecode patterns in the file to a database of malicious code signatures[2] (call it a blacklist). This is a form of static code pattern matching analysis that is used by most antivirus software. However, as a single approach, it is insufficient due to its drawbacks:

Malware authors can use several obfuscation techniques to mask well-known signature byte patterns in order to evade detection. By adding garbage commands and unnecessary code jumps[3], the code signature generated differs and can evade detection. For example, a malware author can rearrange detectable malware by a series of unconditional jumps (ie go-to's in assembly), to

[1] "Kaspersky Security Bulletin 2018. Statistics." *Securelist - Kaspersky Lab's Cyberthreat Research and Reports*, 4 Dec. 2018, securelist.com/kaspersky-security-bulletin-2018-statistics/89145/.

[2] Mujumdar, Ashwini, et al. *Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches*. June 2013, ijarcet.org/wp-content/uploads/VOLUME-2-ISSUE-6-2037-2039.pdf.

[3] Bazrafshan, Zahra & Hashemi, Hashem & Hazrati Fard, Seyed Mehdi & Hamzeh, Ali. (2013). A survey on heuristic malware detection techniques. IKT 2013 - 2013 5th Conference on Information and Knowledge Technology. 113-120. 10.1109/IKT.2013.6620049.

connect consecutive codeblocks, thus altering the signature of the malware. Another method is code encryption, the technique of encrypting the files in which malware is present to avoid detection. Moreover, signature-based approaches rely on the fact that the malware present in the file under inspection has been detected by the engine previously, and the code signature has been manually stored in the blacklist by an analyst, so if the malware is new the approach is ineffective so zero-day attacks will evade detection.

Signature-based approaches are ineffective against malware that employs metamorphic strategies [4]. Malware that is able to mutate itself in order to mask previous similarities will not lead to the same pattern signature and thus will evade detection. As seen above, due to the wide range of obfuscation problems and inefficiencies in the signature-based approaches, more complex detection approaches are usually employed.

3 Behavior-Based Malware Detection

In a behavior based approach, the engine tries to understand the functionality and intended actions of the software under inspection. It is a more dynamic approach than signature-based detection, as it does not rely on a database of already detected malware, so it is able to detect new malware. By monitoring the software's use of system resources, API calls and general behavior, a behavior-based detection engine can flag any suspicious activity in the software and classify it as malware. Examples of suspicious activity in malware may be any attempt to

[4] Bazrafshan, Zahra & Hashemi, Hashem & Hazrati Fard, Seyed Mehdi & Hamzeh, Ali. (2013). A survey on heuristic malware detection techniques. IKT 2013 - 2013 5th Conference on Information and Knowledge Technology. 113-120. 10.1109/IKT.2013.6620049.

discover a sandbox environment (virtual simulations used to detect malware), installing rootkits, disabling security controls etc[5]. Behavior-based approaches can also detect malware mutations, as the fundamental system processes are similar[6] even though their code signature will differ. This is a very big advantage over static signature-based approaches. However, behavior-based methods usually suffer from higher scanning times which present a scalability issue. As behavior-based engines try to interpret what the software is trying to do, rather than check its "fingerprint" against known files, it has a higher probability to return a false positive result. The false positive rate for malware detection has to be extremely low, due to the delicate nature of the topic. If malware goes undetected, the user is at risk of data theft, extortion, loss of property.

In order to extract the relevant features of a file in a behavior-based approach, software operations can be classified using the Behavior Operation Set[7] into:

1. File Actions: Read, Write, Delete etc
2. Process Actions
3. Network Actions: Listen TCP/UDP etc
4. Registry actions

---

[5] Cloonan, John. "Advanced Malware Detection - Signatures vs. Behavior Analysis." *Infosecurity Magazine*, 11 Apr. 2017, www.infosecurity-magazine.com/opinions/malware-detection-signatures/.

[6] Bazrafshan, Zahra & Hashemi, Hashem & Hazrati Fard, Seyed Mehdi & Hamzeh, Ali. (2013). A survey on heuristic malware detection techniques. IKT 2013 - 2013 5th Conference on Information and Knowledge Technology. 113-120. 10.1109/IKT.2013.6620049.

[7] Liu, W., Ren, P.: 'Behavior-based malware analysis and detection'. First Int. Workshop on Complexity and Data Mining (IWCDM), 2011

If the engine, detects a series of these actions that can have malicious consequences, it can then further analyze the file to check if it is malicious or not. The engine uses heuristics to try and match series of these operations to those of known malware families. This goes to show that this method draws from relevant information in the signature-based approaches.

3.1 Anomaly detection

An example of a behavior-based approach is an anomaly detection engine. Engines like this, have already performed feature extraction on a normal/benign file, in order to have a profile of a normal file[8]. They use this file profile as their reference file. Then, all files are compared against the reference profile. Files that deviate from the normal profile are extracted for further inspection. Note that this approach is similar to the approach banks use for credit card fraud prevention. The problems with anomaly detection engines are the following:

1. Obfuscation: If the malware author is able to adjust the malware in order to conform to the normal profile, anomaly will not be detected.
2. High FPR: It is very hard to define a "one-size fits all" file profile that is considered normal. It is also very hard to define what is normal for a benign file. Therefore, as benign files may not conform to the normal profile set, anomaly detection suffers from a high false positive rate.

[8] Mujumdar, Ashwini, et al. *Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches*. June 2013, ijarcet.org/wp-content/uploads/VOLUME-2-ISSUE-6-2037-2039.pdf.

3.2 Virtualization

In this computationally heavy malware detection approach, the execution of a file is observed in an emulation machine, where binary code is translated to run in this emulation and API calls are replaced to modify virtual resources[9]. The file under inspection is virtually isolated in order to be closely examined and studied without risks of infection. This method basically works like a sandbox for executing unknown software and determining whether it is malicious. The problem with this approach is scalability. It is not a feasible approach for scanning large amount of files, as it takes time to set up the virtualization process and emulate the software for every executable. Other detection techniques have to be used as a first step before deciding whether virtualization techniques should be employed.

4. Action Items

The field of malware detection and analysis sees innovation on a daily basis, with a number of new approaches that employ machine learning methods and techniques for faster and more efficient detection. The development of such methods for malware detection, seems like the most promising course of action. For example, DeepInstinct[10] a cybersecurity company, claims to have created a real-time malware-detection agent that is OS and file type agnostic, by using Deep Learning to create a lightweight trained model that does not use feature extraction and can identify malware in any type of file. Having such a memory-light agent with fast file processing,

---

[9] Aubrey-Jones, Tristan. *Behaviour Based Malware Detection* . 2007, pdfs.semanticscholar.org/08ec/24106e9218c3a65bc3e16dd88dea2693e933.pdf.

[10] "Real-Time Threat Prevention Powered by Deep Learning." *Deep Instinct*, www.deepinstinct.com/.

it can be placed in system endpoints and examine executables before letting them run on the system. Many researchers and companies are working on state-of-the-art malware detection using such methods which is why lately, most malware detection methods lean towards Machine Learning based approaches. The ideal malware detection engine is one that can detect mutations in malware, new malware with very low computational overhead and extremely low false positive rate. If this can be achieved, then these engines can be installed in system endpoints with very little overhead.

5. Conclusion

Malware is here and has always been here to stay. There will be no decline in malware creations and infections unless better scalable methods for malware detection are created. Through this brief introduction in the methods used for pre-execution malware detection engines, I hope to have shed some light into methods that need improvement. As technology as a whole progresses, and computational power increases malware authors will work harder and better to create harmful malware that renders undetected. The time is now for malware engines to harness these advancements in technology to create a new era of detection mechanisms.

6. References

1. Mujumdar, Ashwini, et al. *Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches*. June 2013, ijarcet.org/wp-content/uploads/VOLUME-2-ISSUE-6-2037-2039.pdf.

2. Bazrafshan, Zahra & Hashemi, Hashem & Hazrati Fard, Seyed Mehdi & Hamzeh, Ali. (2013). A survey on heuristic malware detection techniques. IKT 2013 - 2013 5th Conference on Information and Knowledge Technology. 113-120. 10.1109/IKT.2013.6620049.

3. Aubrey-Jones, Tristan. *Behaviour Based Malware Detection* . 2007, pdfs.semanticscholar.org/08ec/24106e9218c3a65bc3e16dd88dea2693e933.pdf.

4. Liu, W., Ren, P.: 'Behavior-based malware analysis and detection'. First Int. Workshop on Complexity and Data Mining (IWCDM), 2011

5. Cloonan, John. "Advanced Malware Detection - Signatures vs. Behavior Analysis." *Infosecurity Magazine*, 11 Apr. 2017, www.infosecurity-magazine.com/opinions/malware-detection-signatures/.

6. "Kaspersky Security Bulletin 2018. Statistics." *Securelist - Kaspersky Lab's Cyberthreat Research and Reports*, 4 Dec. 2018, securelist.com/kaspersky-security-bulletin-2018-statistics/89145/.

7. "Real-Time Threat Prevention Powered by Deep Learning." *Deep Instinct*, www.deepinstinct.com/.