ISSN: 2278 - 1323
International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)
Volume 2, Issue 6, June 2013

Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches

Ashwini Mujumdar, Gayatri Masiwal, Dr. B. B. Meshram

Abstract— **Malware is one of the major security threats in computer and network environment. However, Signature-based approach that commonly used does not provide enough opportunity to learn and understand malware threats that can be used in implementing security prevention mechanisms. In order to learn and understand the malwares, behavior-based technique that applied dynamic approach is the possible solution for identification, classification and clustering the malwares.[1] However, it is still unpopular because of its rigid and restrictive nature. In this paper, we study and analyze both approaches and try to determine the best and optimal anti-malware approach.**

Index Terms—**Anomaly, Behavior-based, Signature-based, Specification-based.**

I. Introduction

Malware (Malicious Software) is software that is designed to deliberately infiltrate or damage a computer system without the owner's knowledge. It can appear in the form of code, scripts, active content and other software.[2] Numerous attacks made by malware pose a major security threat to all computer users. Hence, malware detection is one of the computer security topics that are of great interest.

The amount of malware threats on the Internet has increased significantly over the past few years. Hence the traditional methods of malware detection do not suffice. Newer techniques and mechanisms have to be explored.

In this paper we will study and analyze various techniques which use either Signature-based or Behavior-based Malware detection approach.

II. An Overview of Anti-Malware Technologies

Malware and other potentially harmful software have a great impact on user's security, reliability and privacy. Hackers are becoming increasingly motivated by financial gain to steal confidential or personal information rather than just vandalizing the client machine. Moreover, users can experience serious performance and stability problems with their computer, due to presence of spyware and other malware.

An anti-malware engine is responsible for detection and removal of malware as it attempts to infect a computer. This engine performs three main tasks:

A. Scanning

The engine must examine and monitor various locations of the computer such as the hard disk, registry and main memory. If a change to a critical component is detected, it could be a sign of infection.

B. Detection

Once the engine has detected an item that requires further examination, called candidate, by detecting a change or by explicit request by the user, it must identify the presence of malware, if any. The engine refers to a frequently updated list of known malware, called the Blacklist, which contains "signatures" or identifiable patterns of known malware. Using this list, the engine can determine whether any file matches any of the known malware. If a match is found, the file is classified according to the signature as worm, virus, Trojan etc.

C. Removal

The final step for this engine is to take appropriate actions on files that are identified as malware. In most circumstances, the engine removes the program or file completely and restores the computer to its ore-infection state. Otherwise, a file can be disabled or quarantined, so that the user could enable it later.

III. Signature-Based Anti-Malware Approach

Signature-based detection is an anti-malware approach that identifies the presence of a malware infection or instance by matching at least one byte code pattern of the software in question with the database of signatures of known malicious programs, also known as blacklists. This detection scheme is based on the assumption that malware can be described through patterns (also called signatures).[3] Signature-based detection is the most commonly used technique for anti-malware systems.

However, this technique has certain disadvantages:

A. Susceptible to evasion

Since the signature byte patterns are derived from known malware, these byte patterns are also commonly known. Hence they can be easily evaded by hackers using simple obfuscation techniques such as inserting no-ops and code re-ordering. Thus malware code can be altered and

***Ashwini Mujumdar**, Department of Computer Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, India.*

***Gayatri Masiwal,** Department of Computer Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, India*

***Dr. B. B. Meshram**, Department of Computer Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, India*

signature-based detection can be evaded.

B. Zero-day attacks

Since the signature-based anti-malware systems are constructed on the basis of known malware, they are unable to detect unknown malware, or even variants of known malware. Thus, without accurate signatures, they cannot effectively detect polymorphic [4] malware. Therefore, signature-based detection does not provide zero-day protection. Moreover, since a signature-based detector uses a separate signature for each malware variant, the database of signatures grows at an exponential rate.

IV. Whitelisting: Another Malware Detection Technique for Signature-based Approach

Signature-based Blacklisting of malware is no longer enough for protection. Whitelisting is an alternative to this.

Whitelisting is a popular technique among computer users to actively manage the software that is being installed on their computer. Whitelisting involves permitting only approved software to install and run. Software products that are not explicitly on the control list lock down the computer. Whitelisting is a very promising way to protect computers, but it also creates a very rigid environment where rules about what software can be downloaded and installed are strict.

But whitelisting detection has three drawbacks.

- Firstly, it can create an annoying computer experience. Users are subjected to pop-up warnings constantly.
- Secondly, whitelisting limits users' ability to easily download and use new software.
- Thirdly, whitelisted applications can be vulnerable. For example, if you whitelist a browser, then any malware that operates inside the browser will not be detected. In fact, a lot of malware inject themselves into the browser.

V. Behavior-based Anti-Malware Approach

Behavior-based approaches of malware detection monitor behaviors of a program to determine whether it is malicious or not. Behavior based method observes behaviors of a program from outside by actually executing it, and if the program performs the pre-defined malicious behaviors, it can be identified as malware.[5] The behavior of a program that is typically monitored is the stream of system calls that the program issues to the operating system. Since behavior-based techniques monitor what a program does, they are not susceptible to the shortcomings of signature-based detection discussed earlier. Simply put, a behavior-based detector determines whether a program is malicious by inspecting what it does rather than what it says. Several types of behavior-based detections exist.

VI. Anomaly detection: A behavior-based malware detection technique

One major approach of behavior-based detection is anomaly detection. In this approach of malware detection, a profile of normal program behavior is constructed. Any deviations from that profile are flagged as anomalous and thus suspicious.

Anomaly detection is analogous to credit card fraud detection. Credit card companies maintain "spending profiles" for their customers. Any significant deviation from these profiles is flagged as suspicious.

For example, if a credit card company notices a large expense in a shop in Europe, and the customer has not shopped in Europe in the last few years, they will flag that transaction as anomalous. Similarly, let's say a program, during its normal execution, never writes to a certain sensitive directory. If the monitoring system notices writes to that sensitive directory from the program, the detection system will flag that behavior as anomalous.

Anomaly detection has the following two shortcomings:

A. It is susceptible to false positives

Normal behavior for complex programs is very complicated. For example, the set of behaviors of Mozilla Firefox are very complex. Therefore, it is very hard to construct a model of normal behavior of a complex program. An inadequate model of normal behavior can lead to false positives.

B. It is susceptible to mimicry attacks

It has been demonstrated that anomaly detection-based techniques are susceptible to mimicry attacks. In a mimicry attack, an attacker transforms his attack into another equally-malicious attack, but the transformed attack is allowed by the model of normal execution of the program. For this, the attacker has to be familiar with the normal execution model of the program.

VII. Specification-based Monitoring

Specification-based monitoring is a type of behavior-based detection technique, which also makes use of signature-based detection to some extent. In the specification-based approach of malware detection, all events from the program to the operating system are mediated by a specification or policy. The policy specifies what action should be taken for a sequence of events. Typically, the actions are allow, deny or log.

For example, we might have a policy for a browser which states that "any files downloaded from a Web site (not on a whitelist) cannot be automatically executed." This policy will not allow a user to download files from a Web site which are not on a whitelist and execute them. These kinds of specification policies can be very effective in addressing important infection vectors such as drive-by-downloads.

Specification-based monitoring has the following two advantages over anomaly detection:

A. It has flexibility

Specification-based monitoring decouples policy construction from enforcement. For example, one can imagine having a policy in a specification-based monitoring

system that is derived using anomaly detection. Therefore, in an abstract sense, specification-based monitoring is more general than anomaly detection.

B. It has lower false-positives

Since policies in a well-engineered, specification-based monitoring system can be easily tuned, it can result in very low false positives.

VIII. CONCLUSION

Both, Signature-based and Behavior-based detection approaches have their pros and cons. The signature-based systems work well against the technique of attaching a worm to normal traffic, but they are weak against polymorphism. On the other hand, behavior-based systems are able to handle polymorphism only when the worm is largely separated from the background and does not carry too much garbage.[6]

Behavior-based is a more recent and more intelligent approach to malware detection as compared to signature-based. However, it is more rigid and can lead to many false-positives. Right now, specification-based monitoring is the most useful anti-malware technique. It strikes a balance between both, signature-based detection (by using a whitelist) and behavior-based detection (to take action regarding non-whitelisted programs) to give the users optimal protection.

REFERENCES

[1] Mohamad Fadli Zolkipli and Aman Jantan, "Malware Behavior Analysis: Learning and Understanding Current Malware Threats", 2010 Second International Conference on Network Applications, Protocols and Services.
[2] http://en.wikipedia.org/wiki/Malware
[3] Mila Dalla Preda, Mihai Christodorescu, Somesh Jha and Soumya Debray, "A Semantics-Based Approach to Malware Detection", ACM Transactions on Programming Languages and Systems, Vol. 30, No. 5, Article 25, Pub. Date: August 2008.
[4] Yong Tang, Bin Xiao and Xicheng Lu, "Signature Tree Generation for Polymorphic Worms", IEEE TRANSACTIONS ON COMPUTERS, VOL. 60, NO. 4, APRIL 2011.
[5] Yoshiro Fukushima, Akihiro Sakai, Yoshiaki Hori and Kouichi Sakurai, " A Behavior-Based Malware Detection Scheme for Avoiding False Positives", 978-1-4244-8915-2/10/$26.00 ©2010 IEEE
[6] Yong Tang and Shigang Chen, "An Automated Signature-Based Approach against Polymorphic Internet Worms", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 7, JULY 2007.
[7] Wei Yu, Nan Zhang, Xinwen Fu and Wei Zhao, "Self-Disciplinary Worms and Countermeasures: Modeling and Analysis", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 10, OCTOBER 2010.
[8] Asaf Shabtai, Eitan Menahem and Yuval Elovici, " F-Sign: Automatic, Function-Based Signature Generation for Malware", IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS – PART C: APPLICATIONS AND REVIEWS, VOL. 41, NO. 4, JULY 2011.

Ashwini Mujumdar has completed B.E. in Computer Engineering from Pune University and is now pursuing M.Tech. in Computer Engineering from Veermata Jijabai Technological Institute (VJTI).

Gayatri Masiwal has completed B.E. in Computer Engineering from Mumbai University and is now pursuing M.Tech. in Compuetr Engineering from Veermata Jijabai Technological Institute (VJTI).

Dr. B. B. Meshram is a Ph.D. holder who is the Professor and HoD of the Computer Department in Veermata Jijabai Technological Institute (VJTI).