Back to Static Analysis for Kernel-Level Rootkit Detection

Jeff Malavasi
Dept. of Computing Security
Rochester Institute of Technology
Email: jm3378@rit.edu

I. Summary

A. Problem

Rootkits are designed to be extremely stealthy and their main goal is to subvert the operating system in order to prevent exposure. Detecting rootkit malware has been incredibly difficult. Some work has been proposed to classify these binaries via static analysis, but with little success compared to models based on dynamic feature sets.

B. Solution

The authors propose a novel rootkit detection system using static analysis based on two features [1]. First, they observe that legitimate kernel code does not often use obfuscation techniques as well as monitoring drivers.

C. Methodology

In order to determine the presence of a rootkit, a detection process is called each time a driver is installed. Each novel driver is dissembled and then classified as either benign or malicious. The classifier proposed was based on two distinct feature sets. First, it would look for the presence of obfuscation, which is a strong indicator of malware. This is because drivers require a high amount of maintenance and debugging, which encourages developers to both document and write code in a standardized manner. Secondly, 50 behavioral features were proposed based on historical trends of rootkits. These features were broken down into five categories: general behavior, communications rootkit like functionalities, overall static features, and suspicious behaviors.

D. Results

The researchers tested their proposed design against 4400 benign and malicious samples equally distributed. Samples were clustered based on similarity and 1459 distinct clusters were ultimately selected. After training the classifier using tree classification, the authors achieved a 98% accuracy rate, 0.6% false negative rate and 3% false positive rate.

II. Evaluation

A. Strengths

Although the proposed solution is unable to detect every rootkit, it has many strengths. Firstly, it does not rely on dynamic analysis which can be extremely resource intensive and may not assess all potential code paths. Additionally, the sample does not need to be executed in order to be classified, which reduces the overall risk of the analysis.

B. Weaknesses

While the authors were able to design an extremely accurate rootkit classifier, there were also some limitations. For example, upon analysis of the false negative samples, many of them implemented process-hider drivers that were not properly detected by the behavioral analysis. Additionally, rootkit authors could design malware in a way that does not incorporate drivers, which would not be detected by the proposed system. Finally, they did not implement any kernel integrity verification methods prior to analysis, which could be used to decrease the false negative rate and improve accuracy.

C. Significance

This problem addressed in this paper is significant for a few reasons. Firstly,detecting rootkits via static analysis is extremely advantageous compared to dynamic analysis because it requires less resources, analyzes all potential code paths and does not require the binary to be executed. Secondly, while generalized classifiers are important, creating classifiers based on specific malware families has been shown to be much more effective [2].

III. Proposal

The authors could improve their overall design in a few ways. Firstly, the authors are unable to detect rootkits that are not implemented through drivers. One way they could potentially account for these types of rootkits is through the use of hybrid analysis. While the use of static analysis has many advantages, coupling it with dynamic analysis which has shown success historical could account for these edge cases. Additionally, the authors did not utilize any kernel verification techniques which could help to reduce evasion.

References

[1] S. A. Musavi and M. Kharrazi, "Back to static analysis for kernel-level rootkit detection," vol. 9, no. 9, 2014, pp. 1465–1476.

[2] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "UNVEIL: A Large-Scale, automated approach to detecting ransomware," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 757–772. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kharaz