Semantic Detection of Drive-by Attacks in a Browser

Marie-Therese Walter  
s9mewalt@stud.uni-saarland.de

Alessandra Gorla  
gorla@cs.uni-saarland.de

Christian Hammer  
hammer@cs.uni-saarland.de

Saarland University, Saarbrücken, Germany

1 Introduction

With the internet being ubiquitous and pervading almost every aspect of life, it is imperative to protect the integrity of our devices when browsing the World Wide Web. A major obstacle in this context is the rise of so-called drive-by attacks where an attacker either creates a malicious website or infects a benign one by injecting a malicious JavaScript program. Whenever someone visits the infected site, a script is downloaded into the victim’s browser and executed there. Upon execution, this program detects weaknesses in the browser, e.g. a plugin with a known vulnerability that has not been updated, and exploits these weaknesses to take control over the browser or even the user’s whole system.

2 Exposition

Current defense techniques against this type of attack are mostly based on static signatures, i.e. the specific code structures of a program [3]. Due to the highly dynamic nature of JavaScript, however, it is easy to create different versions of the same malicious program with different signatures. This means that they require a manual analysis not only for the basic attack itself but for each new version of this attack.

The amount of effort is significantly reduced when using semantic-based approaches, which focus on the behavior of a program instead of its structure. Since the key behavior does not change much between different versions of the same program, these approaches only need to analyze one version of an attack in order to be able to detect all of them.

However, even with semantic-based approaches there is still the problem that a new attack has to be analyzed manually to understand its characteristic behavior before it can be detected. This gives the malicious program a window of opportunity where it can freely infect defenseless victims without being detected. Our working assumption is that the behavior of benign programs differs from those that try to load malware. This means that instead of analyzing each malicious script separately to determine its characteristic behavior, we monitor the behavior of scripts executing in our browser and stop them as soon as their behavior deviates from that of benign scripts. In particular, we want to detect malicious code before it gets a chance to load a potentially unknown exploit. To do this, we train a classifier, which can then be used to detect anomalous script behavior at runtime.

3 Evaluation

To evaluate our assumption, we instrument WebKit [2], a framework that is used to build popular browsers such as Apple’s Safari or other open-source browsers, to extract information about the execution of JavaScript directly from its interpreter. We use the instrumented WebKit in conjunction with phantomjs [1], a headless browser designed for testing web pages, and Crawljax [4], a web crawler that emulates a user browsing the web, to automatically gather data from several hundred web sites. The gathered data is then evaluated using machine learning techniques to create a model for the behavior of benign scripts. We evaluate the reliability of the classifier that implements this model by testing it against several malicious (and benign) scripts which are not part of the originally gathered data; if our assumption holds, the classifier will be able correctly identify malicious samples.

4 Conclusion

We present a new approach for automatically detecting drive-by attacks in a browser. In contrast to existing approaches our approach does not require manual analysis of an attack in order to be able to detect it.

References