Compile-time techniques for detecting Javascript exploits

Noam Rinetzky[1], Shmulik Regev[2] and Shir Landau-Feibish[3]

[1]*maon@cs.tau.ac.il*
[2]*shmul@il.ibm.com*
[3]*lfshir@gmail.com*

1 General Information

1.1 Workshop Spirit

This workshop will get you familiar with Javascript based exploits and compile time techniques that can be efficient in their detection. Your goal would be the implementation of tools for automatically detecting malicious code in javascript.

Simplifying Assumptions The projects listed below provide a lot of freedom and you are *encouraged* to make intelligent choices of simplifying assumptions. Our only requirements are that your simplifying assumptions will be *justifiable*, *consistent*, and *documented*.
Example: It is ok to assume that the size of the JavaScript source code analyzed is below 100K. *Counter example:* You may not assume familiarity with the obfuscation toolkits.

Getting Started JavaScript files are the input to all the projects. All of our projects rely on existing frameworks for the fairly standard problem of parsing a program and transforming it to some intermediate representation that is designed for analysis and transformation. For example the [**?**] toolkit may be used. Therefore, the first step in your project should be to create a few example programs and make sure that you can run the basic framework on them to produce the intermediate representation.

It is important to get a feeling of how JavaScript based exploits are used. Good starting points would be [2] and [3]. Some examples of exploit bearing JavaScript files can be found here [4].

Project Submissions

Project Presentation When the project is completed, you will be required to present it in a short evaluation meeting. While you will be running the project for the demonstration, we should be able to run your project independently, so please provide the appropriate documentation for doing so as part of your submission.

1.2 Administration

The projects will be done in groups of 2-3 students. Compilation is a prerequisite, but this requirement can be wavered by special permission. The meetings will be held on Monday 4pm-6pm in Schreiber 8. We will have only a few meetings during the semester. The tentative schedule for the project is:

- 17/February/2014: Problem description.

- 24/February/2014: Projects description.
- 7/April/2014: Progress report (design).
- 9/June/2014: Progress report (initial implementation).
- 2/September/2014: project submission.
- Around 15/September/2014: Individual evaluation meetings with the different teams.

This tentative schedule is subject to changes. The exact date of the project presentation will be set after the project submission.

Requirements:

- You are required to provide 5 example programs that your analysis can handle.
- You are required to provide a script for running your analyzer from the command line.
- You are required to provide a **short** description of your implementation.

2 How to Submit

Projects should be submitted through github . Every submission should exist in its own repository which is properly *tagged* prior to submission. Note that untagged projects will not be accepted.

Directory structure:

- src - source files of your project
- doc - documentation. The directory should contain a README.txt file that explains how to run your project.
- bin - binary files of your project
- scripts - any scripts for running/testing your project
- examples - example inputs

3 Available Projects

The list of available projects will be listed soon. In addition, you are encouraged to suggest your own project.

References

[1] Esprima, ECMAScript parsing infrastructure for multipurpose analysis `http://esprima.org/`

[2] Heap Feng Shui in JavaScript, Sotirov A. BlackHat Europe 2007 `http://www.blackhat.com/presentations/bh-europe-07/Sotirov/Presentation/bh-eu-07-sotirov-apr19.pdf`

[3] Engineering Heap Overow Exploits with JavaScript Daniel M., Honoroff J. `https://www.usenix.org/legacy/event/woot08/tech/full_papers/daniel/daniel.pdf`

[4] Examples of malicious javascript `http://aw-snap.info/articles/js-examples.php`