Implementing Dynamic Taint Propagation to Enforce Domain Driven Security

Specification and Time Schedule

FREDRIK ADOLFSSON - FREADO@KTH.SE

Master in Computer Science
Date: February 23, 2018
Supervisor: Musard Balliu
Examiner: Mads Dam
Principal: Jonatan Landsberg & Simon Tardell
School of Computer Science and Communication

Contents

1 Background **1**
1.1 Goal & Objective 1

2 Research Question & Method **3**

3 Evaluation & News Value **5**

4 Pre-study **6**

5 Conditions & Schedule **7**
5.1 Resources 7
5.2 Limitations 7
5.3 Company Supervisor 7
5.4 Time Plan 8

Bibliography **9**

Chapter 1

Background

One of the greatest strengths with deploying applications on the World Wide Web (web) is that they are accessible from everywhere where there exists a internet access. This is sadly one of its greatest weaknesses as well. The applications are easily accessible for people who whishes to abuse or cause them harm. Among the number of security risks that a web application is vulnerable to is two of the more common Injection Attack and Cross-Site Scripting. [4, 1]

To minimize the risk of accidentally introducing security flaws in to the application have a variety of tools and methodologies been created. One of these is Dynamic Taint Propagation (DTP) which marks input from the user as tainted through a taint variable attached to the input. This taint variable follows the input throughout the system and propagates onto the other variables it comes in contact with. It is possible to detaint the input but this is only done after the input have been validated. The taint value is later checked in sensitive areas through something called sinks. Execution is halted if a tainted variable is detected trying to enter the sensitive area through the sink. [5, 6] One of the methodologies that have been coined is the programming paradigm Domain Driven Security (DDS). DDS aim to secure applications by focusing on the core domain models and making certain that validation of the value object is correct. [7, 3]

1.1 Goal & Objective

The goal of this thesis is to implement and benchmark a DTP tool. The benchmark will check the values; injection prevention rate, false pos-

itive rate and added time overhead. A discussion whether this tool also helps to enforce the programing paradigm DDS will also be conducted.

The principal, Omegapoint, is interested in everything that might validate, invalidate, evolve or bring a further value to the programming paradigm DDS. The reason for this is because the concept of DDS was born and is in development by Omegapoint consultants.

Chapter 2

Research Question & Method

How can an implementation of a Dynamic Taint Propagation tool enforce the security gains of Domain Driven Security.

The assignment would be to evaluate the implementation of a DTP tool and discuss if it helps to enforce the security gains of DDS. The process of this thesis would be to conduct, in order:

Literature Study The literature study is where information relevant to the thesis need to be gathered and presented.

Tainting & Detainting This step is the part where tainting and detainting rules are decided. These need to be decided since the next step is the implementation of the DTP tool.

Implementation The implementation step is where the DTP tool is implemented. Omegapoint have developed a proof of concept product which I will continue my work upon. This tool is developed in and for Java with help of the Javassist [2] which makes the manipulation of bytecode easier. The proof of concept is developed to check taint on HTTP query strings trough a Spring server.

Benchmarking This step is where the DTP tool will be benchmarked. The DTP tool should be tested on a larger set of applications to make the result significant. The values that is in focus during the benchmark is the values in the table below.

- Injection Prevention Rate

- False Positive Rate
- Added Time Complexity

Analysis The analysis step is where the benchmarking results is reflected upon and written into the report.

Report Writing & Presentation The last steps is to finalize the report and present the thesis.

The relevance in the thesis lies in the problem with software security. Since we are going towards an age where digitalization only grows larger is the question about how we can secure our software extremely relevant. The hypothesis is that we can help in the process of enforcing more secure software. But the question is with how much and if there are negative side effects such as to much overhead to the runtime.

Chapter 3

Evaluation & News Value

There should be a discussion and evaluation of the implemented DTP tool. This evaluation should contain well thought comments and observations about the benchmarking result. A comparison/analysis of the possibility for the DTP tool to enforce the security gain of DDS shall also be conducted.

The work should be of interest for anyone wanting to see a gain in security. The core idea is to enforce more secure software through DTP. However, since the relation between DTP and DDS will be discussed will the practitioners of DDS find it extra interesting.

Chapter 4

Pre-study

The literature study will focus on gathering the relevant information needed for the report. These areas are listed in the table below:

- Web Applications
- Dynamic Taint Propagation
- Domain Driven Security
- Injection Attacks
- XSS
- Javassist

Research into JVM modifications must also be included since it is needed for the implementation of the DTP tool. The information will be obtained by researching for relevant books, reports and other possible material. Two of the founders of the concept of DDS work at Omegapoint and are accessible for questions. Conduction interviews with the founders might be of interest.

Chapter 5

Conditions & Schedule

5.1 Resources

To save some time will the development of the DTP tool continue on the work that Simon Tardell have started. Which is a tool developed in and for Java with help of the Java library Javassist [2]. Applications to evaluate the implementation is also of need. The thesis is at the moment aimed towards web applications which means that a number, 10 should be sufficient, of web applications need to be gathered. Omegapoint have some internal systems which could be used. Other usable web applications can be found on open source platforms.

5.2 Limitations

- The DTP tool dose not have to be a production ready. The goal is to develop a prototype.
- Web Applications is the targeted applications.
- The scope of the thesis will not contain Static Taint Propagation.
- The tool is developed in Java with Javassist.

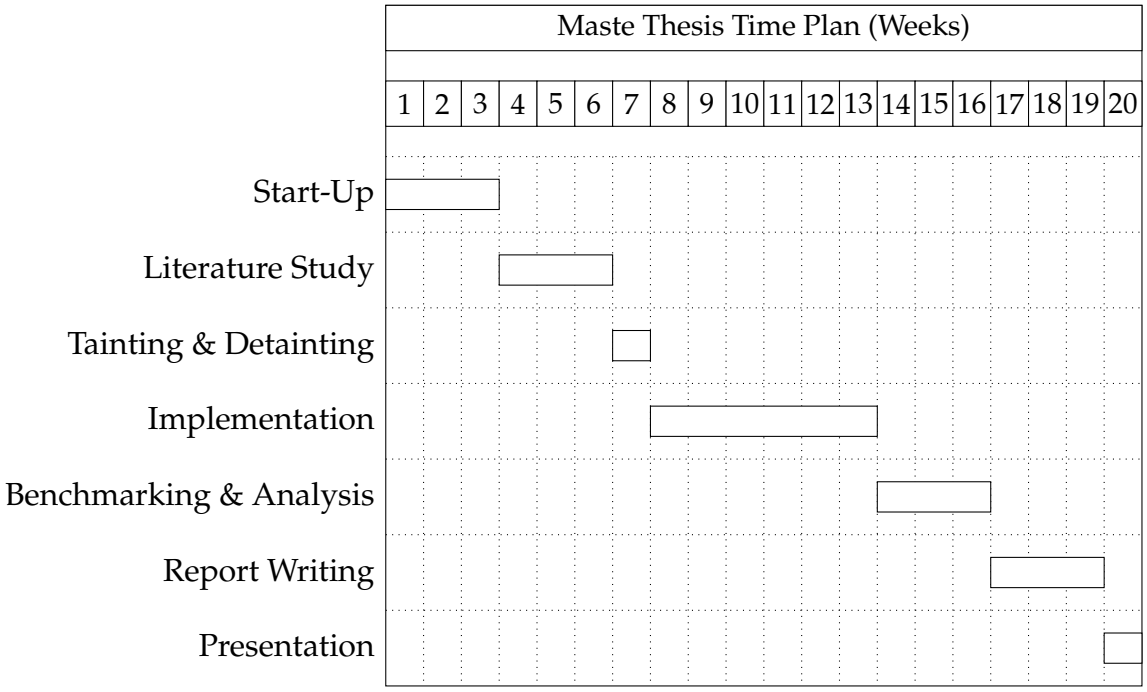
5.3 Company Supervisor

- **Jonatan Landsberg:** Will assist with supervision on the academic part if the thesis.

- **Simon Tardell:** Supervisor in the technical parts of the thesis. He is also the author of the first draft of the DTP tool which this thesis will continue its work upon.

5.4 Time Plan

Below is my time plan for the Masters Thesis. The goal is to continuously, throughout all phases, add to the report. But I have also reserved a couple of weeks in the end for writing the report. I believe that this time can be used to add to or rewrite sections if needed.



Bibliography

[1] Michael Cross. *Developer's guide to web application security*. eng. Rockland, MA: Syngress Publishing, 2007. ISBN: 1-281-06021-6.

[2] *Javassist by jboss-javassist*. URL: http://jboss-javassist.github.io/javassist/ (visited on 02/22/2018).

[3] Dan Bergh Johnsson. *Dear Junior - Letters to a Junior Programmer: Introducing Domain Driven Security*. 2009. URL: http://dearjunior.blogspot.se/2009/09/introducing-domain-driven-security.html (visited on 01/25/2018).

[4] Open Web Application Security Project. *OWASP*. URL: https://www.owasp.org/index.php/Main%7B%5C_%7DPage (visited on 02/01/2018).

[5] Jinkun Pan, Xiaoguang Mao, and Weishi Li. "Analyst-oriented taint analysis by taint path slicing and aggregation". In: *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS* 2015-November (2015), pp. 145–148. ISSN: 23270594. DOI: 10.1109/ICSESS.2015.7339024.

[6] Guru Venkataramani et al. "FlexiTaint: A programmable accelerator for dynamic taint propagation". In: *Proceedings - International Symposium on High-Performance Computer Architecture* (2008), pp. 173–184. ISSN: 15300897. DOI: 10.1109/HPCA.2008.4658637.

[7] Johan Wilander. *OWASP Sweden: Domändriven säkerhet / Domain-Driven Security*. 2009. URL: http://owaspsweden.blogspot.se/2009/09/domandriven-sakerhet-domain-driven.html (visited on 01/25/2018).