Applying Dynamic Taint Propagation in Order to Enforce Domain Driven Security

Specification and Time Schedule

FREDRIK ADOLFSSON - FREADO@KTH.SE

Master in Computer Science
Date: February 15, 2018
Supervisor: Musard Balliu
Examiner: Mads Dam
Swedish title: Tillämpa dynamisk taint propagation för att genomdriva domändriven säkerhet
School of Computer Science and Communication

Contents

Chapter 1

Background

Domain Driven Security (DDS) is a methodology that is an extension to Domain Driven Design (DDD). The core concept, in simplified form, is about focusing on the development of the core domain models. By being certain that they are correctly modeled and implemented can we make sure that validation before propagating data into vulnerable parts of the system, such as the database, can be correctly executed. [2, 3, 10, 4]

A functionality that is tightly built around correctly validated objects, that some developers and even some programming languages have incorporated, is taint checking. [7, 5, 1] A form of taint checking is Dynamic Taint Propagation (DTP). DTP works by marking input from the user as tainted through a taint variable attached to the input. This taint variable follows the user input throughout the system and propagates onto the other variables it comes in contact with. But the taint variable can be removed, this is done after the user input have been properly 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 sinks. [6, 9]

1.1 Goal

The goal of this thesis is to evaluate and benchmark how well an implementation of Dynamic Taint Propagation can help and ensure protection from injection attacks through enforcing the use of DDS. Test cases such as injection prevention rate, false positive rate, added time overhead will be conducted.

1.2 Objective

The concept of DDS have been born and is in development by Omegapoint consultants. This means that everything that might validate, invalidate, evolve or bring a further value to the mythology in any way is of interest to them.

The topic for this thesis was born and discussed at Omegapoint's latest tech talk, OP Tech Talks [8]. Since Omegapoint regularly offers master thesis positions was this a excellent topic to offer. Omegapoint would like to see, except for a thesis that is of KTH's expected standard, a prototype of a possible implementation of a Dynamic Taint Propagation tool, including well thought through detainting rules.

Chapter 2

Research Question & Method

How can Dynamic Taint Propagation help and enforce users to follow the security paradigm of Domain Driven Security.

2.1 Problem Definition

The first task will be the literature study where information about Domain Driven Security, Dynamic Taint Propagation, injection attacks and general information about application development need to be gathered and presented in the thesis. This is then followed by a discussion of how the detainting rules should be implemented to support DDS.

The following problem is to further develop the Dynamic Taint Propagation tool. This also includes the implementation of the detainting rules. Next challenge will be to evaluation how well the tool might help to enforce DDS. It might be possible that it is not worth using at all because of the possible overhead it might entail. Questions such as its effects of preventing security flaws, false positive and added time complexity should be answered.

2.2 Examination Method

A Dynamic Taint Propagation tool will be implemented and logic for detainting needs to be evaluated and developed. This tool will then be used to evaluate applications. Questions that will be evaluated during the benchmark for example; possible performance issues and possible taint leaks. One possible application to evaluate is one of Omegapoint's internal systems.

2.3 Expected Scientific Results

The relevance in the report lies in the hypothesis that Domain Driven Design can be used to develope complex and secure software. Which is currently discussed and developed by consultants at Omegapoint. Then by enforcing users to follow the security paradigm through usage of Dynamic Taint Propagation will lead to more security applications. 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

3.1 Evaluation

The value of the the implemented tool should be discussed and evaluated. The research questions should as well have been discussed in relation to the results. The evaluation will be based on variables such as the possible increased security gained through increased prevention of Injection Attacks and Cross-Site Scripting. But also if there is extra work and complexity added to the work of the developer or the applications runtime.

3.2 Work's Innovation/News Value

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 dynamic taint checking which enforces Domain Driven Design. However, practitioners of DDD/DDS might find it extra interesting since it addresses the use of DDS and gives a tool to help in the process of using DDS.

Chapter 4

Pre-study

The literature study will focus on Dynamic Taint Propagation, DDD/DDS and injection attacks. But information about application structure will also be needed. Research into JVM modifications must also be included since it is needed for the implementation of the Dynamic Taint Propagation tool. The information will be obtained by researching for relevant books, reports and other possible material. Two of the founders of the concept of Domain Driven Security 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 Dynamic Taint Propagation tool continue on the work that Simon Tardell have started. Applications to evaluate the implementation is also of need. Omegapoint have some internal systems which could be used.

5.2 Limitations

The Dynamic Taint Propagation tool is to backup the report and dose not have to be a production ready tool. It should be a proof of concept. The report will also only talk about Dynamic Taint Propagation and not static.

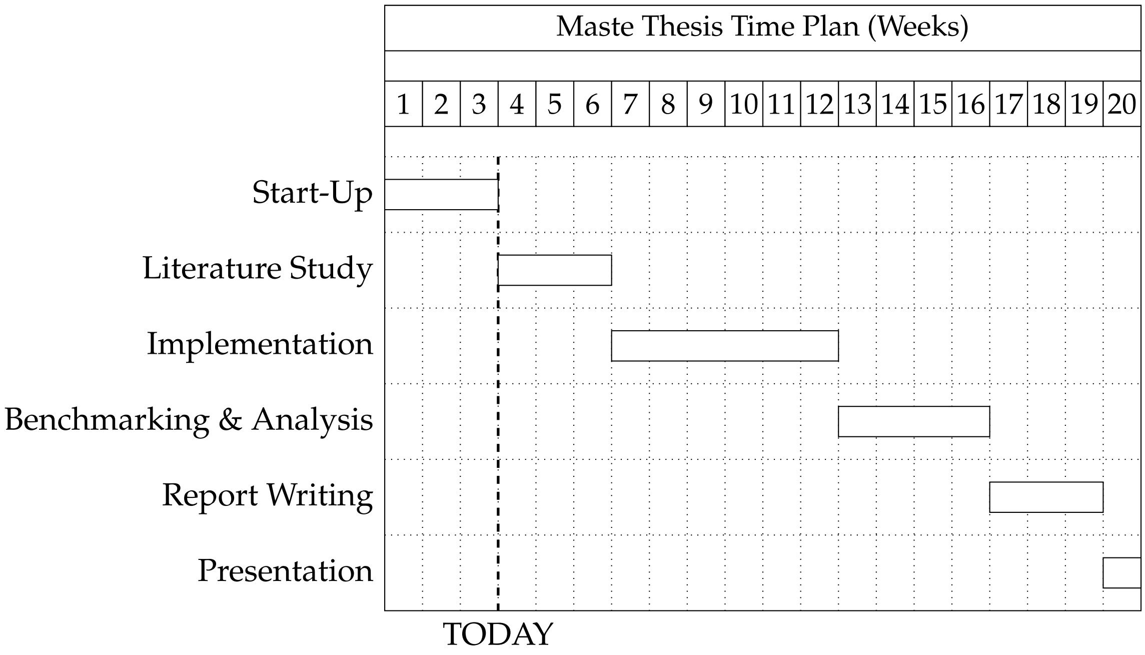
5.3 Company Supervisor

- **Simon Tardell:** Is my supervisor in the technical parts of the thesis. He's also constructed a first draft of the Dynamic Taint Propagation tool which I am free to use.

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

5.4 Time Plan

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



Bibliography

[1] James Clause, Wanchun Li, and Alessandro Orso. "Dytan: a generic dynamic taint analysis framework". In: *Proceedings of the 2007 international symposium on Software testing and analysis* (2007), pp. 196–206. DOI: 10.1145/1273463.1273490. URL: http://doi.acm.org/10.1145/1273463.1273490.

[2] Eric Evans. *Domain-driven design reference: definitions and patterns summaries*. Dog Ear Publishing, 2015.

[3] Eric Evans. *Domain-driven design : tackling complexity in the heart of software*. eng. Boston, Mass.: Addison-Wesley, 2004. ISBN: 0-321-12521-5.

[4] 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).

[5] *Locking Ruby in the Safe*. URL: http://ruby-doc.com/docs/ProgrammingRuby/html/taint.html (visited on 01/25/2018).

[6] 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.

[7] *perlsec - perldoc.perl.org*. URL: http://perldoc.perl.org/perlsec.html (visited on 01/25/2018).

[8] Simon Tardell. *Dynamic Taint Propagation | OP Tech Talks (Stockholm, Sweden) | Meetup*. URL: https://www.meetup.com/en-AU/op-tech-talk/events/245435404/ (visited on 01/30/2018).

[9] 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`.

[10] 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).