

Applying dynamic taint propagation in order to enforce domain driven security

Specification and Time Schedule

FREDRIK ADOLFSSON

Master in Computer Science

Date: February 2, 2018

Supervisor: Musard Balliu

Examiner: Mads Dam

Swedish title: Tillämpa dynamic taint propagation för att genomdriva domändriven säkerhet

School of Computer Science and Communication

Contents

1	Background	1
1.1	Objective	1
2	Research Question & Method	2
2.1	Problem Definition	2
2.2	Examination Method	2
2.3	Expected Scientific Results	3
3	Evaluation & News Value	4
3.1	Evaluation	4
3.2	Work's Innovation/News Value	4
4	Pre-study	5
5	Conditions & Schedule	6
5.1	Resources	6
5.2	Limitations	6
5.3	Company Supervisor	6
5.4	Time Plan	7
	Bibliography	8

Chapter 1

Background

Domain Driven Security (DDS) is a methodology that can be seen as 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 making sure that they are correctly modeled and implemented can we make sure that validation before propagation can be correctly executed. [2, 3, 8, 4]

A functionality that is tightly built around correctly validated objects, that some developers and even some programming languages even have incorporated, is taint checking. [6, 5, 1] This thesis will look at the possibility to apply the later to enforce the programming paradigm of DDS.

1.1 Objective

The concept of DDS have been born and is in development by consultants at Omegapoint. This means that everything that might validate, invalidate or evolve the mythology in any way is of interest for them. The topic for this thesis was born and discussed at Omegapoint's latest tech talk, OP Tech Talks [7]. Since Omegapoint regularly offers master thesis positions was this an 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, which can support the claims in the thesis.

Chapter 2

Research Question & Method

How can dynamic taint propagation enforce none Domain Driven security practitioners to follow the security paradigm of Domain Driven Security.

2.1 Problem Definition

The first challenge is to implement the dynamic taint propagation tool. The implementation also includes a evaluation about how detainting should be handled and implemented. The second 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 if it effect 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 different implementations of application, both with and without the methodology of DDD/DDS. The applications could be self written but it would also be interesting to validate, if possible, existing applications.

2.3 Expected Scientific Results

The relevance in the report lies in the hypothesis that Domain Driven Design can be used to develop 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 the security level of applications increase. 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.

Chapter 3

Evaluation & News Value

3.1 Evaluation

The value of the the implemented tool should be discussed and possible compared with similar tools. The research questions should as well have been answered and discussed in relation to the results. The evaluation will be based on variables such as the possible increased security gained trough 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.

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 trough dynamic taint checking and Domain Driven Design. However, practitioners of DDD/DDS might find it more interesting since it addresses the use of DDS.

Chapter 4

Pre-study

The literature study will focus on taint propagation, DDD/DDS and what they are all about. 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 Omega-point and are accessible for questions. Conduction interviews with the founders might be of interest.

Chapter 5

Conditions & Schedule

5.1 Resources

No special equipment is needed for the project. But to save some time will the development of the dynamic taint propagation tool continue on the work that Simon Tardell have started. But depending on the direction of the thesis might a, or more, already existing program implemented on the JVM be needed. They can be used to evaluate the dynamic taint propagation tool to see if we can find vulnerabilities in them.

5.2 Limitations

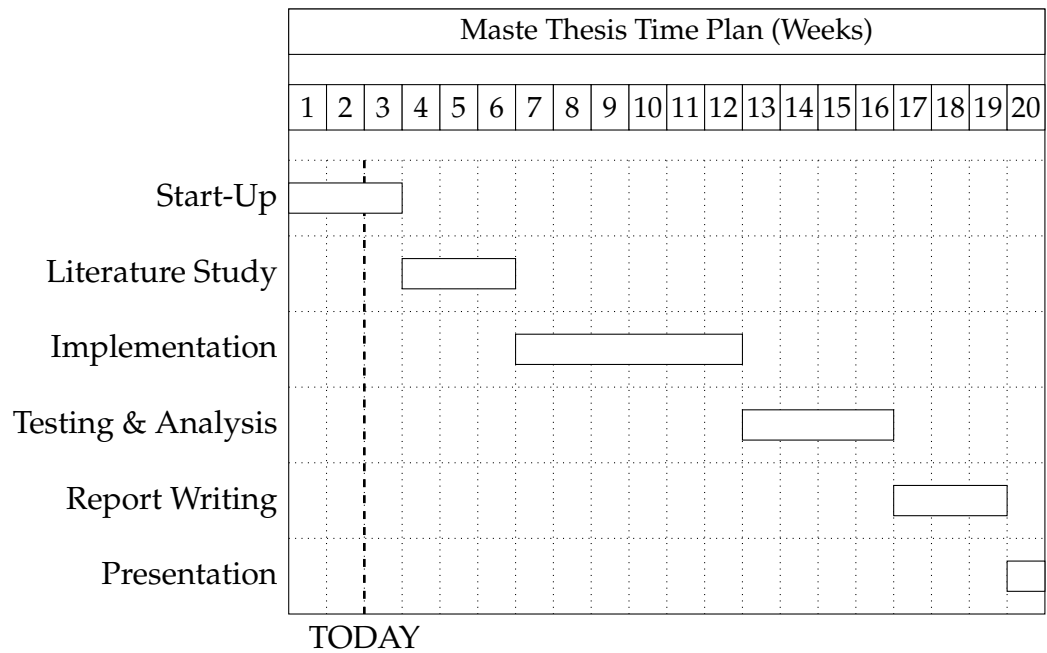
The dynamic taint propagation tool is to backup the report and dose not have to be a production ready tool. It should be a proof of concept.

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, through out 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] *perlsec - perldoc.perl.org*. URL: <http://perldoc.perl.org/perlsec.html> (visited on 01/25/2018).
- [7] 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).
- [8] Johan Wilander. *OWASP Sweden: Domändriven säkerhet / Domain-Driven Security*. 2009. URL: <http://owaspsweden.blogspot.se/2009/09/domanddriven-sakerhet-domain-driven.html> (visited on 01/25/2018).