Faculty of Mathematics and Computer Science

Heidelberg University

Master thesis

in Computer Science

submitted by

Stefan Machmeier

born in Heidelberg

1996

Honeypot Implementation

in a

Cloud Environment

This Master thesis has been carried out by Stefan Machmeier

at the

Engineering Mathematics and Computing Lab

under the supervision of

Herrn Prof. Dr. Vincent Heuveline

(Titel der Masterarbeit - deutsch):

(Title of Master thesis - english):

Contents

Acronyms III

List of Figures IV

List of Tables V

1 Introduction 1
- 1.1 Problem description . . . 1
- 1.2 Justification, motivation and benefits . . . 1
- 1.3 Research questions . . . 1
- 1.4 Limitations . . . 1

2 Background 2
- 2.1 Cloud Computing . . . 2
  - 2.1.1 Definition of Cloud Computing . . . 2
  - 2.1.2 Service models . . . 3
  - 2.1.3 Deployment models . . . 4
  - 2.1.4 Cloud Security . . . 4
  - 2.1.5 HeiCloud . . . 4
- 2.2 Honeypots . . . 4
  - 2.2.1 Definition of a Honeypot . . . 4
  - 2.2.2 Honeynets . . . 4
  - 2.2.3 Legal Issues . . . 4
- 2.3 Intrusion Detection System . . . 5

3 Previous Work 6
- 3.1 The Bait'n'Switch Honeypot . . . 6
- 3.2 Intrusion Trap System . . . 6
- 3.3 Honeycomb . . . 6
- 3.4 Honeypots in a cloud environment . . . 6

4 Practical Work 7
- 4.1 Attack vectors . . . 7
  - 4.1.1 Primer . . . 7
- 4.2 Proposed Honeypots . . . 7
  - 4.2.1 Cowire . . . 7
  - 4.2.2 Dionaea . . . 7
  - 4.2.3 Honeyd . . . 7

4.3 Concept 7
4.3.1 HoneyTrap 7
4.4 Implementation 7

5 Experimental Work **8**
5.1 SNORT 8

6 Evaluation **9**
6.1 T-Pot 9
6.2 Analyzation 9

7 Conclusion **10**
7.1 Future work 10

Bibliography **I**

Acronyms

DaaS Data-as-a-Service

HTTP Hypertext Transfer Protocol

IaaS Infrastructure-as-a-Service

NIST National Institute of Standards and Technology

PaaS Platform-as-a-Service

SaaS Software-as-a-Service

List of Figures

2.1 Cloud functionalities (derived from [WvLY+10]) 3

List of Tables

1 Introduction

1.1 Problem description

1.2 Justification, motivation and benefits

1.3 Research questions

1.4 Limitations

2 Background

This chapter concludes the fundamental knowledge that is needed to comprehend the upcoming practical work. Firstly, an introduction to cloud computing will be held. Next, a throrough understanding of honeypots is given. Lastly, we introduce some concepts of intrusion detection systens.

2.1 Cloud Computing

Nowadays it is one of the well-known keywords and has been used by vary large companies such as Google, or Amazon, however, the term "cloud computing" dates back to the late 1996, when a small group of technology executives of Compaq Computer framed new business ideas around the Internet.[Reg20] In this section, we want to give basic unterstandings of cloud computing, and give a short introduction to HeiCloud.

2.1.1 Definition of Cloud Computing

Considering the definition of Brian Hayes, cloud computing is "a shift in the geography of computation" [Hay08]. Thus, computational workload is moved away from local instances towards services and datacenters that provide the need of users [AFG+10].

Considering the definition of the National Institute of Standards and Technology (NIST), cloud computing "is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [MG11]. NIST not only reflects the geographical shift of resources such as datacenters, but also mentions on-demand usage that contributes to a flexible resource management. Morever, NIST composes the term in five essential characteristics, three service models (see 2.1.2), and four deployment models (see 2.1.3):

On-demand-self-service refers to the unilaterally provision computing capabilities. Consumers can acquire server time and network storage on demand without a human interaction.

Broad network access characterizes the access of capabilities of the network through standard protocols such as Hypertext Transfer Protocol (HTTP). Heterogeneous

thin and thick client platforms should be supported.

Resource pooling allows the provider's computing resources to be pooled across several consumers. A multi-tanent model with different physical and virtual resources are assigned on demand. Other aspects such as location are independent and cannot be controlled on a low-level by consumers. Moreover, high-level access to specify continent, state, or datacenter can be available.

Rapid elasticity offers consumers to extend and release capabilities easily. Further automization to quickly increase resources when demand skyrockets significantly can be supported regardless limit and quantity at any time.

Measured service handles resources in an automated and optimized manner. It uses additional metering capabilities to trace storage, processing, bandwith, and active user accounts. This helps to monitor, and control resource usage. Thus, contributing to transparency between provider and consumer.

2.1.2 Service models

Software-as-a-Service (SaaS)

Platform-as-a-Service (PaaS)

Infrastructure-as-a-Service (IaaS)

Data-as-a-Service (DaaS)

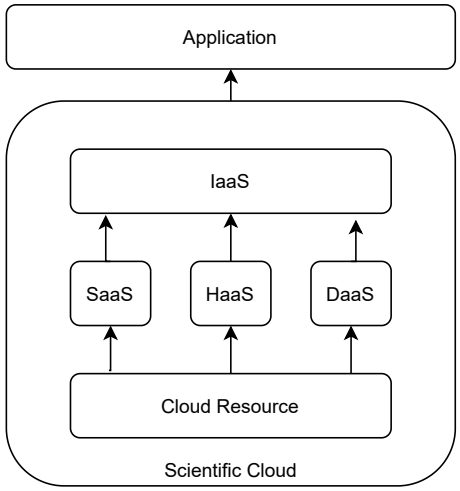

Figure 2.1: Cloud functionalities (derived from [WvLY+10])

Examples for such service models are:

2.1.3 Deployment models

Private Cloud

Community Cloud

Public Cloud

Hybrid Cloud

Examples for such deployment models are:

2.1.4 Cloud Security

[NCM12]

2.1.5 HeiCloud

2.2 Honeypots

The first public honeypot [Spi03]

2.2.1 Definition of a Honeypot

On the Internet there are a dozen of defintions for honeypots. Thus, to cope with all the subtle differences, we want to take a closer look at some of the definitions and narrow down our own one.

Spitzner defines honeypots as a "security resource whose value lies in being probed, attacked, or compromised."[Spi03]

High-interaction honeypots

Low-interaction honeypots

Pure honeypots

2.2.2 Honeynets

[Spi03]

2.2.3 Legal Issues

[Spi03]

2.3 Intrusion Detection System

3 Previous Work

3.1 The Bait'n'Switch Honeypot

[PD05]

3.2 Intrusion Trap System

[PD05]

3.3 Honeycomb

[PD05]

3.4 Honeypots in a cloud environment

[KPM+21]

4 Practical Work

4.1 Attack vectors

4.1.1 Primer

4.2 Proposed Honeypots

4.2.1 Cowire

4.2.2 Dionaea

4.2.3 Honeyd

4.3 Concept

4.3.1 HoneyTrap

4.4 Implementation

5 Experimental Work

Connect results of Honeypots with NIDS/IDS to update rules.

5.1 SNORT

6 Evaluation

6.1 T-Pot

6.2 Analyzation

7 Conclusion

7.1 Future work

Bibliography

[AFG+10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, April 2010.

[Hay08] Brian Hayes. Cloud computing, 2008.

[KPM+21] Christopher Kelly, Nikolaos Pitropakis, Alexios Mylonas, Sean McKeown, and William J. Buchanan. A comparative analysis of honeypots on different cloud platforms. *Sensors*, 21(7):2433, April 2021.

[MG11] P M Mell and T Grance. The NIST definition of cloud computing. Technical report, 2011.

[NCM12] SR Nithin Chandra and TM Madhuri. Cloud security using honeypot systems. *International Journal of Scientific & Engineering Research*, 3(3):1, 2012.

[PD05] G. Schaefer P. Diebold, A. Hess. A honeypot architecture for detecting and analyzing unknown network attacks, Feburary 2005.

[Reg20] Antonio Regalado. Who coined 'cloud computing'?, Feb 2020.

[Spi03] Lance Spitzner. *Honeypots - Tracking Hackers*. Addison-Wesley, Amsterdam, 2003.

[WvLY+10] Lizhe Wang, Gregor von Laszewski, Andrew Younge, Xi He, Marcel Kunze, Jie Tao, and Cheng Fu. Cloud computing: a perspective study. *New Generation Computing*, 28(2):137–146, April 2010.