Department of Mathematics and Information Technology: Cloud Platform comparison for malware development

Kamil Janowski

Master’s Thesis in Computer Science

Autumn Term 2018

Department of Mathematics and Information Technology

University of Jyväskylä

ABSTRACT

Janowski, Kamil. 2018. Master’s thesis in computer science. University of Jyväskylä. Department of Mathematics and Information Technology (MIT).

The cloud platforms such as AWS, Google Cloud or Azure are designed to cover most popular cases in terms of web development. They provide services that make it easy to create a new user based on his email address, provide tools for inter-service communication, tools to manage the access rights of different users. Malware development however is more of a corner case, where the client application running on the victim’s machine does not have an email address or a google account to authenticate itself and it does not run directly in the cloud, what can make it more difficult to manage the appropriate access rights. Also, the potential attacker may not want to write his own self-contained service, since, especially when managing a large number of clients, it might be much cheaper to run the backend serverlessly.

//TODO: describe research methods (exploratory?)

The following paper explores possible malware backend architectures for different cloud platforms, aiming to optimise the performance, minimize the development time while keeping the code easy to maintain and to minimize the execution cost.

//TODO: write the final conclusion

Keywords: malware, development, cloud, CnC, backend, serverless

CONTENTS

[1 INTRODUCTION 5](#_Toc517629186)

[1.1 Introduction 5](#_Toc517629187)

[1.2 Research Problem 6](#_Toc517629188)

[1.3 Research Objective 6](#_Toc517629189)

[1.4 Research Question 7](#_Toc517629190)

[1.5 Key Definition 7](#_Toc517629191)

[1.5.1 Hacker 7](#_Toc517629192)

[1.5.2 Botnet 7](#_Toc517629193)

[1.5.3 Serverless computing 7](#_Toc517629194)

[1.5.4 Cloud Computing 8](#_Toc517629195)

[1.5.5 Malware 8](#_Toc517629196)

[1.5.6 CnC server 8](#_Toc517629197)

[1.6 Structure of the thesis 8](#_Toc517629198)

[2 Theoretical Background 8](#_Toc517629199)

[2.1 Common botnet architectures 9](#_Toc517629200)

[2.2 Command delivery methods 9](#_Toc517629201)

[2.3 File Upload 9](#_Toc517629202)

[2.4 Video streaming 9](#_Toc517629203)

[3 Methodology 9](#_Toc517629204)

[4 Findings – case study on 3 platforms 9](#_Toc517629205)

[5 Conclusion 9](#_Toc517629206)

# INTRODUCTION

The popularity of computing clouds have increased drastically during the recent years. It is perfectly understandable, taken into account that renting the infrastructure from a cloud provider tends to be significantly cheaper than maintaining it inside the company. Things like the rental of the server room, electricity consumed by the servers, cooling of the server room and salaries of people responsible for the maintenance of the servers generate unnecessary overhead in terms of costs of maintenance, which can be drastically reduced when switching to the cloud, while in the same time providing higher availability and better monitoring of the hosted services. Furthermore the cloud providers constantly introduce new solutions allowing to reduce the maintenance costs even further. As we can read in “Serverless Computing: Economic and Architectural Impact” by Gajko Adzic and Robert Chatley (2017, p. 884):

Amazon Web Services unveiled their ‘Lambda’ platform in late 2014. Since then, each of the major cloud computing infrastructure providers has released services supporting a similar style of deployment and operation, where rather than deploying and running monolithic services, or dedicated virtual machines, users are able to deploy individual functions, and pay only for the time that their code is actually executing. These technologies are gathered together under the marketing term ‘serverless’ and the providers suggest that they have the potential to significantly change how client/server applications are designed, developed and operated.

It is important to note however that those technologies are not only available to big corporations trying to lower their cost of server maintenance, but also to hobby software developers and hackers.

A successful attacker may have thousands of devices under his control. In order to control such a large number of devices remotely a highly scalable Command-and-Control (CnC) server is required. Scaling up the virtual machines (VM) however can be costly, while having only a small number of administrators leads to a situation where most of the resources assigned to those VMs are seriously underutilized. While all the remote malware subscribes to the push notification service, it mostly just waits for a command to be generated by an administrator. Effectively, while our CnC server has to be scalable in order to maintain the connection to numerous clients, it requires fairly low computing power until an administrator decides to generate certain load. This suggests that the serverless approach could be applied in this case, what could potentially not only save the attacker a lot of money, but also make such a large scale attack possible in the first place.

## Research Problem

There are many various cloud providers out there. While they all provide services allowing to easily and quickly build secure web applications, the problem of building a CnC server is more of a corner case, that is not necessarily properly addressed by certain clouds. This might yield it impossible to implement such an application in a serverless manner at all, or require to make some compromises and implement workarounds for services that work in a different manner than desired.

## Research Objective

The main objective of the research is to find a way to use the cloud as a CnC server without implementing any application that requires a constantly running server in a Virtual Machine, as those are the main cost generators of the web applications. For this reason we are going to investigate the serverless solutions provided by various cloud providers as well as other services that come with specific clouds that could potentially allow us to set up the communication between the backend and the client application, enable the file transfer, make it easy to manage the access rights of different clients as well as enable the client management in as a whole. We are also going to take a closer look at how the continuous deployment can be solved in various cloud systems.

Each of the approaches will be backed up by a small Prove of Concept (POC) if possible at all. In order to optimise the development time and ensure multi-platform and multi-cloud support of at least parts of our code, all solutions will be implemented with Node.js.

## Research Question

When focusing on various cloud platforms, such as Amazon Web Services (AWS), Google Cloud Platform (GCP) and Azure the approach to the problem might be completely different and the cost of execution may different significantly as well. The question in this case is, which one of the platforms is the best suited and the cheapest to run our CnC server.

## Key Definition

### Hacker

Hacker is a malicious attacker attempting to access resources of a remote machine. In this thesis the term “hacker” will be used to describe the administrator of the CnC server and in the same time the administrator of the botnet.

### Botnet

A botnet is a network of private computers infected with malicious software and controlled as a group without the owners' knowledge, e.g. to send spam.

### Serverless computing

The “serverless” computing is a marketing term that relates to developing single functions, rather than a large monolithic application and then being charged only for the actual execution time of the function, rather than for the constantly running server that technically is still there, but is hidden from the service user. The concept was originally introduced by Amazon in their AWS cloud in 2014 under the name of Lambda. Since then all major cloud providers introduced various equivalents in their solutions. As many instances of lambda can be triggered in parallel, this solutions is not only cheaper to execute, but also potentially infinitely scalable. This is why it’s commonly used for a wide range of applications, starting with REST API call processing and ending with Big Data event handling.

### Cloud Computing

As Amazon defines it on <https://aws.amazon.com/what-is-cloud-computing> (24-06-2018):

Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing.

### Malware

Malware, or malicious software, is any program or file that is harmful to a computer user. Malware includes computer viruses, worms, Trojan horses and spyware. These malicious programs can perform a variety of functions, including stealing, encrypting or deleting sensitive data, altering or hijacking core computing functions and monitoring users' computer activity without their permission.

### CnC server

In “Survey on botnet: its architecture, detection, prevention and migration” by Ihsan Ullah et al. (2013) the CnC servers are defined as centralised servers allowing the malicious attacker to remotely control a number of clients applications that connect to it.

## Structure of the thesis

//TODO: finish

# Theoretical Background

## Common botnet architectures

## Command delivery methods

How do you do it in P2P and how in CnC. Explain why CnC approach is faster and more convenient.

## File Upload

## Video streaming

Is it even possible when you go p2p?

# Methodology

What you’re going to use to research your paper.

Describe that you’re going to use the qualitative research method and shit.

Describe the study framework. (check the link to that research from Nguyen, 7.2.3 – Study Framework)

# Findings – case study on 3 platforms

Market overview of cloud computing / 3 platforms. All the info about all 3 platforms. (in separate sub-points)

Comparison of the platforms.

# Conclusion