University of Kent

Corpus

CO880 - Project and Dissertation

Yann Ferrere
ID 15906875
yf57@kent.ac.uk

SUMMARY

Table of contents

1 Introduction

1.1 Context

With the development of the Internet, the number of computer servers has significantly increased. Indeed, Internet is based on interconnection between all devices such as computers and smartphone but also servers. The aim of those servers is to store data such as video, image, music, databases, but it also provides interaction between software executed on them and remotes users. These interactions are called client/server communication. Servers can, for instance, execute a software such as a web server, which consists of making accessible web pages stored on it to remote users. Evidently, plenty of other servers usages exists but all of them widely implies to store data, run software and communicate information on the network.

Meanwhile, the usage of devices connected to internet has become extremely widespread. Some basic tasks of daily living such as consulting its bank accounts, communicate with each other or read the news, can be performed anywhere by using an internet connected device such as a smartphone. All of these services can be provided by a private company, but also by a government or an individual. In order to function efficiently, these services are developed by using a computer language and are hosted on a server to store users data and to be accessible remotely.

However, all of these communication and stored data can be more or fewer sensitives. Indeed, communication between two people, bank account information, password, and other credentials data have a value that is interesting to steal or alter by an ill-intentioned person.

1.2 Goals of the project

The risk of a server attack is now becoming ubiquitous. Based on this known risk I decided to analyze what are the steps of an attack on a server. Moreover, servers work as common computers and need to have an operating system to operate properly. This dissertation will focus on Linux servers distributions. Indeed, the majority of servers are Linux based and can run, for instance, known OS such as Ubuntu or Debian.

Consequently, this project consists of putting in place a Linux server accessible remotely by its unique public IP address and then to install a Honeypot on it. This type of tool is a software that emulates a fake Linux environment in order to analyze attackers behaviour. Moreover, a honeypot makes a server accessible remotely through a fake SSH connection with really common identifiers (username/password) in order to be bruteforced easily. Hence, this tool provides a way to allow malicious connection on its server while restricting the access to the machine.

During 6 months, our honeypot server received around 72500 login attempts with 62 percent of successful connection. Those attacks were from 2550 distinct IP addresses and from a large number of different countries all around the world. Following these attacks multiple data such as command lines executed by attackers, scripts and malware downloaded have allowed me to perform a more detailed analysis of attackers behaviours and their motivations.

1.3 Contribution

The honeypot technic to collect attacks on a Linux server is not new. Several papers have already put in place this configuration in order to analyze attackers behaviours. However, there are plenty of ways to implement a honeypot. Some papers have created their own network architectures with multiple machines, while others have used existing solutions such as Kippo. In this dissertation, the Honeypot is cowrie, a fork of the kippo project which is not longer maintained. Consequently, this paper shows in a real case scenario that this honeypot can be an appropriate choice in a project that needs to collect data from Linux server attacks.

It is also important to note that attackers technics to compromise Linux server are constantly evolving. Indeed, security researchers and developers are continuing to look at what are the newest vulnerabilities and try to patch them (CVE). Then, attackers find new attacks scenario in order to bypass new security detection and protection tools and so on. However, security researchers have to put sensors in order to collect data and analyze security issues. Consequently, all along this project research, a databases containing all attacks information has been created in order to update and improve knowledge of current Linux server attacks.

Furthermore, because of a large number of ELF malicious collected binaries I performed a static analysis of them by analyzing strings. In fact, when malware are not packed or stripped multiple hard coded strings can be extracted from them. During the analysis of these strings, 10 differents types were observed such as IP addresses, URLs or readable english messages. In order to automate their extraction, I developed a python script that automatically extract strings from a given malware, extract, sort and store them into a database. The aim of this could be to generate a large dataset from strings contained in malware to be applied through a data mining algorithm in order to perform a pre-analysis of a malware.

1.4 Structure of the disseration

In order to understand the functioning of Honeypots and their objectives, a short review of the existing literature about similar project already performed is exposed first. Then, the methodology of my research will be explained.

The technical implementation and configuration of the server, the honeypot and tools used to store and read collected data will be examined in details. In the same way, the development and usage of the malware strings analyzer will be shown.

A concluding discussion will summarize this analysis and results will be described in the same section. Finally, in a future research part will provide a way to a possible future development and reflexion.