Banking malware, when to extract configuration data from system memory

Jeroen Kraan
Hogeschool van Amsterdam
Email: Jeroen.Kraan@hva.nl

Ricardo van Zutphen
Hogeschool van Amsterdam
Email: Ricardo.van.Zutphen@hva.nl

I. Introduction

Dynamic malware analysis is an important tool to recognize and help understand new threats in the form of malware. Dynamic malware analysis is studying the behaviour of malware while it is being executed on a controlled host computer. This can be in a virtualized environment or on a physical machine. The purpose of dynamic analysis is to collect behavioural data like arguments used in system API calls, the contents of the system memory, and network communications. Network communication is an important way to recognize who the malware is communicating with. The who is called a command and control server (C2). C2s are part of what is called a malware configuration. This can contain C2s, version numbers, and any other information used by a specific type of malware.

One way to collect parts of the configuration data is to look at the network communication of malware. Another type is trying to retrieve the configurations or part of it from process memory dumps.

Automated Dynamic malware analysis systems like Cuckoo Sandbox try to automate this process by making memory dumps at a point during the execution of the malware. These memory dumps can then be analysed by various tools. The question is: at what point in execution does the system memory contain the malware configuration?

Cuckoo Sandbox is planning to develop a new module to increase the chance of extracting a configuration from the memory. To support this development, this research will be focused on finding the most likely time slot and possibly the required events during execution in which the malware configuration will be located in the system memory.
Our thesis: Banking malware does not load the malware configuration data into memory directly after the execution of the binary, will help determine when not to create process memory dumps during the malware execution.

During this research, we will focus on banking malware. By banking malware, we mean malware that has a focus on stealing financial information, stealing login credentials, keystroke logging, and form manipulation. Examples of names of banking malware are: Dridex [1], Zeus [2], and Vawtrak [3]. The reason that we choose banking malware is that the binaries for this malware are likely to contain configuration data like C2s, because the malware will need to upload the collected data.

This paper will be organized as follows: section 2 will contain a statement of the problem. Section 3 will contain an overview of the collected dataset used for measurement. Section 4 explains how to recognise the in-memory malware configuration for the collected dataset. Section 5 will contain the result of the measurement. Section 6 will contain the conclusion.

II. Problem statement

Malware usually communicates with command and control (C2) servers. We call the IP addresses, ports, URLs, other information used to communicate, and other information used by the malware: configuration data. This data can be embedded in the malware binary, where it is usually encrypted or obfuscated in some way. If malware wants to use the configuration data while being executed on a system, it will need to decrypt the information and load it into the system memory [4].

Dynamic malware analysis can try to take advantage of this. A process memory dump can be made, so that the configuration information can be retrieved from the dump. The problem is that the exact moment in time where the information resides in the memory is not always known [4]. This research will focus on what the most likely moment during execution of banking malware is at which the malware configuration data resides in the system memory.

A. Research question and goal

The thesis this research will measure is Banking malware does not load the malware configuration data into memory directly after the execution of the binary. By directly, we mean within 10 seconds after the execution.

Using the results of this measurement, we want to answer the question: What is the most likely moment during execution of banking malware at which the system memory contains

the malware configuration data?

The goal is to find a moment during execution in which the configuration data is most likely located in the system memory. This information will be used to develop a new analysis module for the dynamic malware analysis system Cuckoo Sandbox to automatically try to extract malware configurations from memory dumps. The development of this module is not included in this research.

B. Measurement

To measure our thesis we use the ratio of time in seconds until a malware configuration is loaded into the system memory.

C. Methods

For this research we will use multiple malware samples of two or three banking malware families.

To analyze these samples, we will use a Cuckoo Sandbox instance configured with virtual machines using Windows 7 64bit as the operating system. We will use Cuckoos process memory dump functionality to create multiple memory dumps during the time of execution of the malware samples.

Using Yara, a tool using signatures to recognize data structures in binary data, we will determine if a malware configuration is present in the memory dump or not [5].

The data collected by measuring the presence of the malware configuration in the memory dumps will be used to show the moment where is most likely that the configuration resides in the system memory.

III. Research design

We will first start with collection malware samples of two to three banking malware families. A researcher and developer of Cuckoo Sandbox will help us to collect samples from sources like VirusTotal.

We will set up a Cuckoo Sandbox instance. For each malware family we will now create Yara signatures to recognize the in-memory malware configuration.

After creating the signatures, we will use Cuckoo Sandbox to analyze the complete malware dataset and make memory dumps at different moments in each analysis. Using the created Yara signatures, we will measure the presence of the configuration data. The results of this measurement will be used to prove or disprove our thesis and to answer the research question.

References

[1] D. O'Brien, Dridex: Tidal waves of spam pushing dangerous financial Trojan, Symantec, 2016, pp. 21-24

[2] J. Wyke, What is Zeus, Sophos, 2011, pp. 10-13.

[3] J. Koustek, Analysis of Banking Trojan Vawtrak, AVG Technologies, Virus Lab, 2015, pp. 10-11.

[4] J. Wyke, Breaking the bank(er): Automated configuration data extraction for banking malware, Sophos, 2015, pp. 5,9.

[5] C. Roberston, "Indicators of compromise in memory forensics," SANS, February 2013.