Artifacts of WIN_JELLY Malware using GPU Forensics

Yazeed Albabtain
Department of Computer and Information Technology
Purdue University

Project Overview

Graphics Processing Unit (GPU) is an essential part in every computer. The purpose of the GPU is to reduce the load on the Central Processing Unit (CPU) by creating graphics, colors, and textures if needed. The security of the GPU has been breached by a malware called Jellyfish. The malware was discovered in May 2015. The malware targets the GPU to avoid detection and monitor the user activity. The first version that has been created of Jellyfish rootkit targets Linux operating system, specifically AMD and NVIDIA cards (Maheux, 2014). Afterward, anonymous developers developed a new malware called WIN_JELLY which targets Windows operating system. A forensics tool will be developed in this research using OpenGL as an attempt to detect Jellyfish malware. The study will develop a detailed analysis of Jellyfish malware using malware forensics tools, such as MEMORYZE, PEview, Strings, Process Monitor, and Process Explorer. A variety of network analysis tools will be used in this study to determine the malware behavior, such as Wireshark, ApateDNS, and iNETsim. After having a clear background about the behavior of the Jellyfish malware, a forensic framework will be developed for graphics processing unit as an attempt to detect such a malware. C/C++, CUDA, OpenGL, OpenCL and few other programming languages, technologies, platforms, framework and libraries will be used to detect any evidence for WIN_JELLY malware.

Research Methodology

Stage 1:

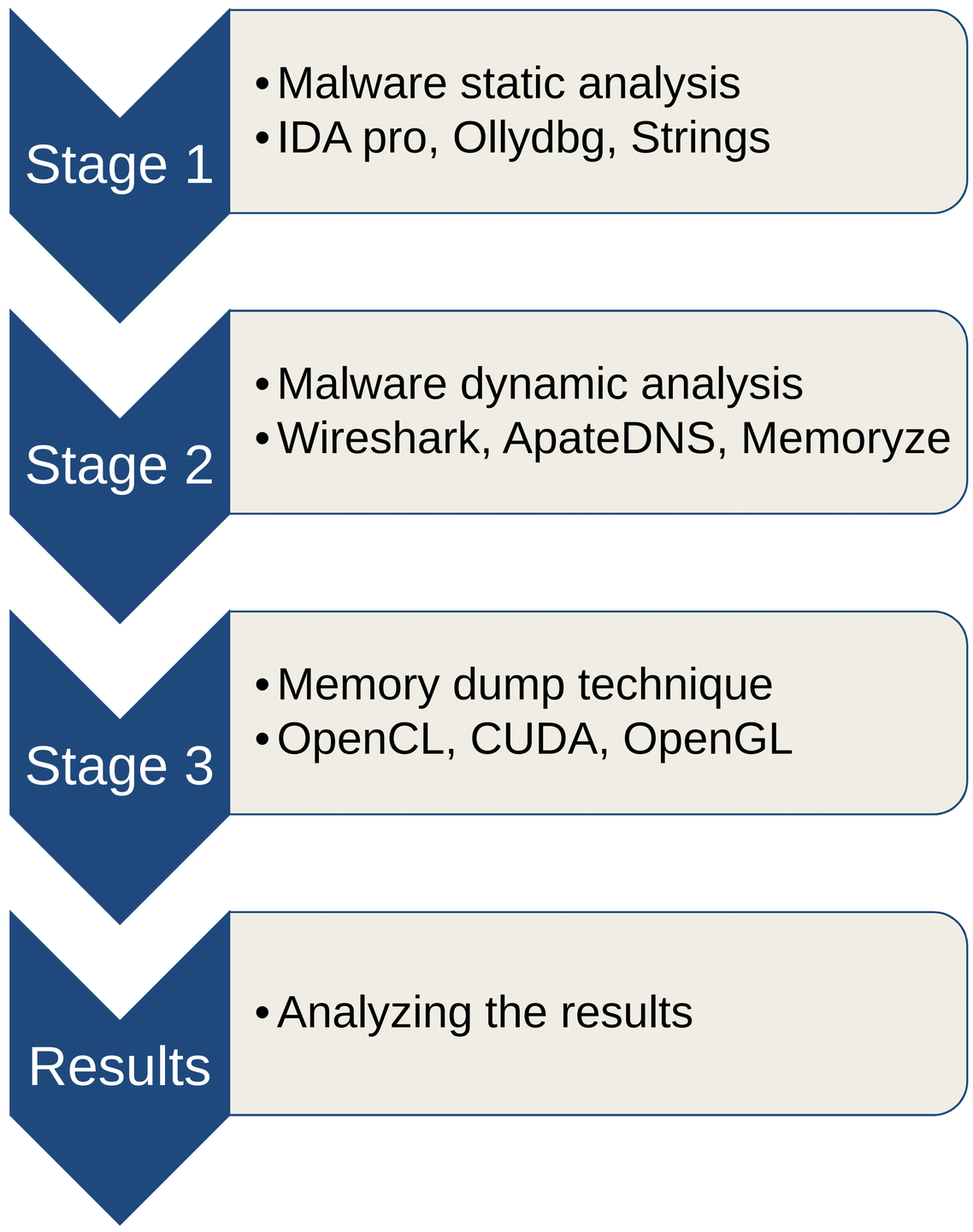
Malware static analysis will be held at this stage. A variety of static analysis tools will be used in this stage such as strings, IDA pro, and Ollydbg. Stage one will help in understanding the malware behavior. IDA Pro typically is the disassembler. Since it is a disassembler, it investigates binary programs, in support of code source that is often available to generate. The real interest for this disassembler is the fact that it indicates the instructions that are essential. OllyDbg is a debugger which highlights the analysis of binary code. The binary code for OllyDbg is helpful whenever the source code is never accessible (Park and Ruighaver, 2008). OllyDbg tool maps out registers, distinguishes procedures, API calls, tablets, constants, switches and strings. Thus, a thorough report of WIN_JELLY malware will be obtained in stage one.

Stage 2:

In this stage, a malware dynamic analysis will be performed. Wireshark application and ApateDNS tool will be used to detect any network activity performed by the malware. Memoryze application will be used to discover any malicious code within the memory of the computer system. Memoryze may obtain or rather analyze images within the memory together with those on live systems might incorporate the paging files within its analysis.

Stage 3:

It is noted that WIN_JELLY malware resides in the GPU to avoid detection. Thus, a memory dump technique will be used in this stage to detect if any artifacts remain in the GPU memory. The tool will be developed using C++ programming language with a variety of frameworks and libraries.



Conclusion

Dealing with volatile memory is often considered a challenge due to its nature of handling data. Therefore, not many forensics tools have been developed in this field. The goal of this research is to discover if WIN_JELLY malware can be detected using a GPU memory dump technique and to propose a new detection approach for malware that targets the graphics processing unit.

References

Maheux, B. (2014). Assessing the Intentions and Timing of Malware. *Technology Innovation Management Review*, *4*(11).

Park, S., & Ruighaver, T. (2008). Strategic approach to information security in organizations. In *Information Science and Security, 2008. ICISS. International Conference on* (pp. 26-31). IEEE.