CERIAS

The Center for Education and Research in Information Assurance and Security

Fileless Malware: The Phantom Menace

Brady Morgan
Department of Computer and Information Technology
Purdue University

Project Overview

A growing problem in cyber security is the inevitable advancement of malware. Improvements in cyber security and forensics techniques have served as the catalyst for change in the way malware is written and delivered. Specifically, as it has become easier to detect and prevent conventional malware attacks, new methods and types of attacks have taken their place. One prominent example is fileless malware. Fileless malware is malicious code that once executed resides solely in random-access memory making it difficult to detect. It is differentiated from memory only malware by its distinct ability to persist through reboots and be detected with disk forensics. This study sought to better understand how fileless malware spreads, functions, and persists. In the pursuit of a full understanding, the study used both static and dynamic analysis techniques on a variety of fileless malware samples.

Objectives

It was the intent of this research to investigate and analyze multiple samples of fileless malware in order to draw conclusions concerning the delivery, function, and persistence of fileless malware in general. In addition, this research sought to locate indicators of compromise as well as other artifacts left by the obtained samples.

Methodology

Samples. Three samples of fileless malware were obtained from Payload Security's online repository. These were chosen to insure the investigation and analysis was performed without bias or prior knowledge of a sample's functionality.

Environment. The analysis was conducted on two virtual machines. The victim machine was running Windows 7 x64. The analysis machine was running Windows XP x32.

Static. Static analysis of the obtained samples was conducted using the programs IDA Pro, OllyDbg, and Strings. This allowed for the individual samples to be analyzed before they were ran in an effort to be better prepared for their unknown effects.

Dynamic. Dynamic analysis of the obtained samples was conducted multiple times for each sample. The programs FakeNet, Wireshark, Process Monitor, Process Explorer, RegShot, DumpIt, and Magnet Ram Capture were used to monitor the samples as they executed as well as collect images before and after the sample's execution.

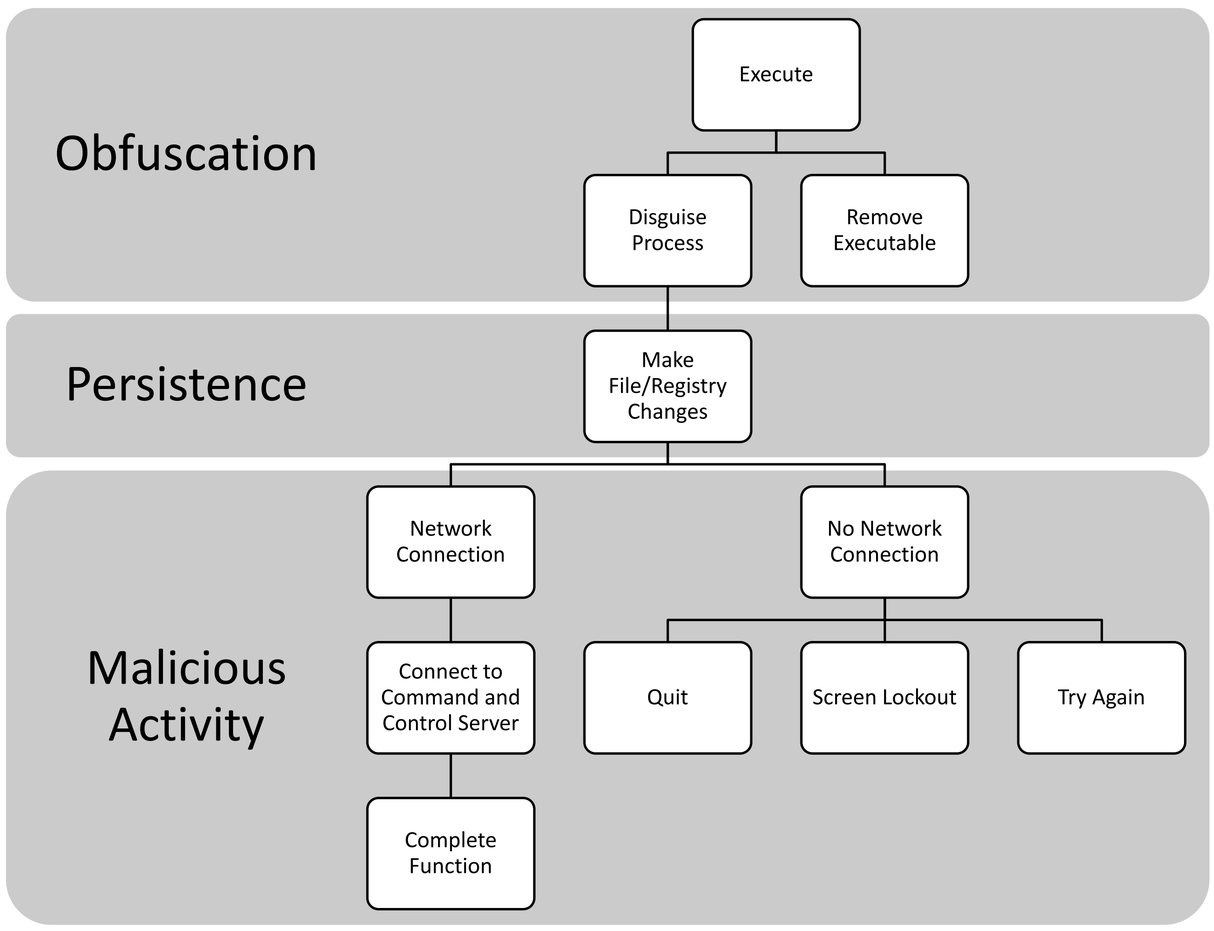
Comparisons. Comparisons of the data were focused on the initial file type, process trees, registry changes, run time, and observed effects. Individual indicators of compromise and artifacts were also recorded and compared across the samples.

Analysis

The samples obtained were similar in delivery, function, and persistence.

Static. Static analysis showed that all three samples held functions concerning file creation and manipulation. Two of the samples utilized regsvr32, Microsoft's Register Server. Two of the samples scanned for analysis tools.

Dynamic. Dynamic analysis showed that all of the samples immediately removed their executable from the machine. They then attempted to disguise their process as another legitimate process such as svchost.exe. They all made several registry changes and attempted to connect to the network. When no connection was present, one sample did nothing, one sample disabled task manager before locking the screen and requesting network access, and the last sample simply continued to try to connect to the network on a loop.



PURDUE UNIVERSITY