Ransomware Analysis and Defense

WanaCry and the Win32 environment

Justin Jones

June 10, 2017

Contents

1 Introduction

A type of malware known as *ransomware* has recently become very prevalent in the cyber security world, taking over user systems and demanding compensation for the safe return of all functionality and data. While initially not incredibly sophisticated, this genre, if you will, has evolved from simple scripts that change file extensions and make empty threats to full-blown attacks affecting hundreds of thousands of systems worldwide that implement sophisticated NSA-developed exploits as their propogation vector. In this paper I will explore a specific piece of malware known as *WanaCry* that recently made headlines around the world, performing a full static and dynamic analysis. I will then endeavor to write a useable piece of software to detect, stop and remove this malware, and then expand it to encompass any general, definable piece of software.

2 Analysis

2.1 WanaCry/WCry

2.1.1 Background

WanaCry (referring to the general family consisting of all named variations of WannaCrypt, WCry, WanaCrypt, WanaCrypt0r, etc) came into prevalence during a massive attack starting on May 12, 2017. This software utilizes an exploit called EternalBlue[1], a known vulnerability in the Server Message Block (SMB) protocol used by Microsoft Windows which was previously patched in a critical update outlined in KB4013389[2]. As this vulnerability has been explored and detailed very thoroughly already, focus will be shifted to WanaCry's implementation and software aspects while avoiding the inner workings of the exploit.

The working sample of WanaCry has been obtained from theZoo[3], with SHA256 hash:

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

and is positively identified by VirusTotal as a member of the WanaCry family[4].

[1] http://bit.ly/2spdT15

[2] https://support.microsoft.com/en-us/help/4013389/title

[3] http://thezoo.morirt.com/

[4] http://bit.ly/2s93pCl

2.1.2 General File Data

Utilizing PEiD[5], it can be seen that the program was packed using Microsoft Visual Studio C++ 6.0 for Win32.

Utilizing Dependency Walker[6]

[5] https://www.aldeid.com/wiki/PEiD

[6] http://www.dependencywalker.com/