Sentinel One | Blog

Breaking TA505's Crypter with an SMT Solver

By Jim Walsh | Jan 27, 2021 | 10 min read

Today's malicious crypter blamers, SMT/SAT solvers, to break the obfuscation and decrypt the crypter doing unique work.

Executive Summary

- TA505 has been leveraging the Hancitor loader and the crypter used to hide it since at least September 2019.
- Hancitor is a loader that has been used by many threat actors to deliver other malware.
- Crypter is the name we have given to a custom crypter that TA505 has been using to obfuscate, protect, and deliver Hancitor.
- This blog contains a brief review of the TA505 packing and crypt routines and a PANDAMAN.

Background

TA505 has been leveraging their crypter for many years, but they are using this crypter to deliver many different malware families, not just Hancitor, and the functionality has changed. The crypter is actually used to orchestrate the execution of Hancitor, and the functionality of the crypter goes through the process of setting up a new execution environment.

Research Insight

The crypter is the first thing that Hancitor binaries hit after they are loaded, so it will control the binary data, so the crypter is the binary data that is of interest. We are interested in the crypter because it is a custom packer, and we want to understand how it works, so we can write a unpacker for it.

The crypter is a PE file, and it has a single .text section, and it has a single entry point. The crypter is a 32-bit binary, and it is compiled using Microsoft Visual Studio, so it is a Windows binary, and it is not a .NET binary.

```
Machine:  014C (i386)
Characteristics:  0102
  Relocations stripped:  Yes
  Executable:  Yes
  Line numbers stripped:  Yes
  32bit word machine:  Yes
Time date stamp:  5F9A0A0A (Thu Oct 29 17:17:14 2020)
Image base:  00400000
Entry point:  00001000
  Section(s) in file:  1
  Section(s) in memory:  1
  Size of image:  00001000
  Size of headers:  00000400
  Checksum:  00000000
  Subsystem:  0002 (Windows GUI)
  Stack reserve:  00100000
  Stack commit:  00001000
  Heap reserve:  00100000
  Heap commit:  00001000
  Loader flags:  00000000
  Number of rvas and sizes:  00000010
Section(s) in file:
  .text  00000200-00000BFF  00000800  00001000  00000000  00000000  60000020
```

The crypter has a single section, and it has a single entry point, so it will control the binary data, so the crypter is the binary data that is of interest.

```
Section(s) in file:
  .text  00000200-00000BFF  00000800  00001000  00000000  00000000  60000020
Section(s) in memory:
  .text  00001000-00001FFF  00000FFF  00000000
Entry point:  00001000
```