ACIDBOX Clustering

epicturla.com/blog/acidbox-clustering

June 26, 2020

Jun 26
Written By J A G-S

Update 06.29.2020: As in all things, the wider wisdom of the threat intel community provides greater context. The code overlaps I was pointing to while scarcer were not in fact unique. As @TheEnergyStory points out, it's Mbed TLS put through Visual Studio. Relevant tweets pictured here:

R136a1 @TheEnergyStory · 8m

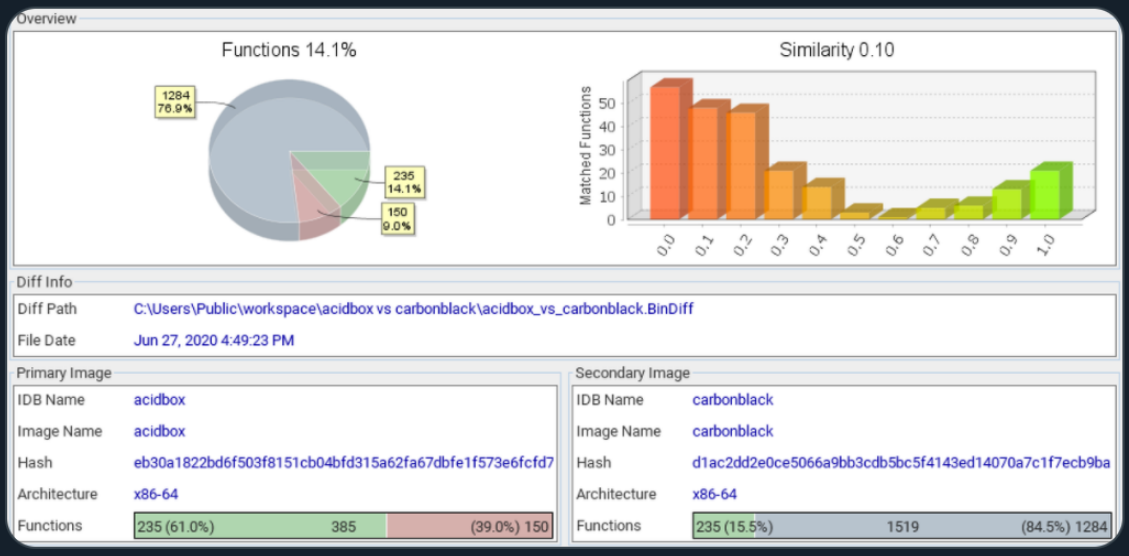
Replying to @TheEnergyStory

However, I don't think there is a connection between Turla Nautilus and AcidBox. The code parts that match are almost entirely from the RSA implementation beside some API functions.

1

R136a1 @TheEnergyStory · 8m

The RSA code is taken from Mbed TLS and can be also found in other files. For example, if you compare AcidBox to a legit Carbon Black driver that uses the same RSA code version from Mbed TLS compiled with MS Visual Studio 2008 SP1, you get a similar high code overlap:



1

R136a1 @TheEnergyStory · 9m

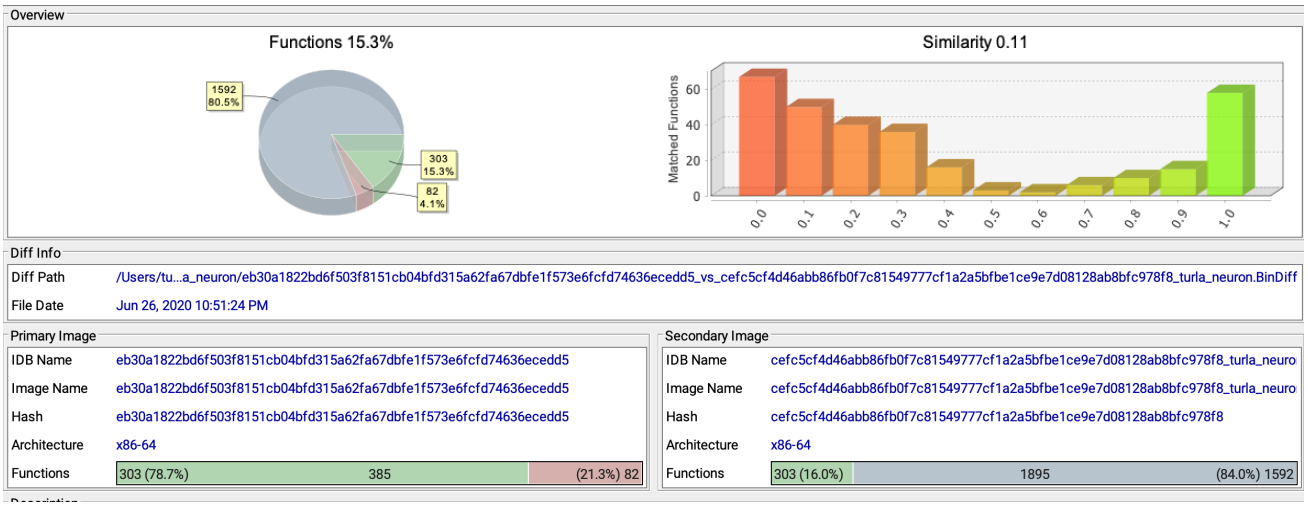
The only difference between the Turla Nautilus RSA code and that of AcidBox is within the former the default strings weren't removed. Also, the coding style of the Turla Nautilus sample is very straight forward and clear while the one of AcidBox is the exact opposite.

With that nailing the coffin on this false start, I hope folks will have better luck figuring out what to do with this interesting new cluster :)

As I was looking over this blog, I realized I'd inadvertently settled into a once a month post and hadn't been planning to post anything for June. However, doing some routine code similarity work, I stumbled upon something I felt like sharing for others to enjoy over the weekend.

Last week, Palo Alto Unit42 researchers Dominik Reichel and Esmid Idrizovic revealed their discovery of ACIDBOX (a.k.a. KL: MagicScroll), a new and fascinating activity set reminiscent of Remsec (a.k.a., KL_Project Sauron or SYMC_Strider) and using an exploit technique of our dear Turla (namesake of this blog). Researchers were cautious about affirmatively clustering ACIDBOX to either in lieu of stronger evidence.

With the prospect of possibly finding a new Remsec or Turla campaign or more of an unknown threat actor, I was eager to work on rules for this set in the hopes of unearthing more context. A few samples have trickled onto VirusTotal over the past week and based on these I was able to construct a well tested code similarity rule. While I haven't had a chance to dig deeper into the findings, I wanted to present a curious early finding others may find interesting:



ACIDBOX (left) and Turla Nautilus Payload (right)

This ACIDBOX DLL (sha256: eb30a1822bd6f503f8151cb04bfd315a62fa67dbfe1f573e6fcfd74636ecedd5) shows a good portion of code overlap with a peculiar Turla Nautilus sample reported by NCSC in November 2017. The next stage payload, 'oxygen.dll', is built with the same compiler (Microsoft Visual C/C++ 2013) and linker (Microsoft Linker 12.0), and meant to be decrypted in order to inject into a target process via reflective loading.

While a portion of that similarity may be accounted for by common statically linked code, a shared crypto implementation merits more attention.

> eb30a1822bd6f503f8151cb04bfd315a62fa67dbfe1f573e6fcfd74636ecedd5: sub_18001A524
>
> > cefc5cf4d46abb86fb0f7c81549777cf1a2a5bfbe1ce9e7d08128ab8bfc978f8: sub_18002E694

- eb30a1822bd6f503f8151cb04bfd315a62fa67dbfe1f573e6fcfd74636ecedd5: sub_1800179AC

    cefc5cf4d46abb86fb0f7c81549777cf1a2a5bfbe1ce9e7d08128ab8bfc978f8: sub_1800275CC

- eb30a1822bd6f503f8151cb04bfd315a62fa67dbfe1f573e6fcfd74636ecedd5: sub_180017664

    cefc5cf4d46abb86fb0f7c81549777cf1a2a5bfbe1ce9e7d08128ab8bfc978f8: sub_180024D70

Enjoy your weekend and *Happy Hunting.*

J A G-S