



Deception Engineering: exploring the use of Windows Service Canaries against ransomware

 research.nccgroup.com/2021/03/04/deception-engineering-exploring-the-use-of-windows-service-canaries-against-ransomware/

March 4, 2021

Date: 2021 Mar 03 11:05:20.878872 (UTC) **IP:** 81.134.98.19
Channel: DNS

Geo Info	
Country	GB 
City	London
Region	England
Organisation	AS2856 British Telecommunications PLC

Tor	
Known Exit Node	False

Basic Info	
Memo	SWOLLENRIVER
Generic Data	DESKTOP-LBU71F2 hibernating

tl;dr

We prototyped a Windows Service Canary in order to target parts of the ransomware kill chain to minimize impact and overall success of operations. Multiple instances are installed masquerading as common Windows services that are targeted by threat actors prior to encryption. If multiple instances of these services are stopped then a Canary token is triggered and the host hibernated. In doing so we are able to alert, minimize the impact and give the best possible chance of recovery.

Background – Ryuk’s TTPs

Ryuk is a well known and active ransomware first discovered in 2018 and covered by CrowdStrike (January 2019), Carbon Black Threat Analysis Unit (February 2020) and Abdallah Elshinbary (May 2020). These write-ups all detail certain trade-craft used in Ryuk deployments around service and process termination prior to encryption beginning via *kill.bat*.

The list of Windows services and processes killed via *kill.bat* is extensive with Abdallah Elshinbary providing a comprehensive list of both.

Thesis

Given that a number of services are stopped prior to encryption there exists the opportunity not only for detection but also impact minimization.

The thesis is we can deploy a number of Canary Windows Services which keep track of how many are running. If these Windows services are stopped (via *net stop*, *sc* or similar) not during a host shutdown we are then able to respond automatically. This automated response involves firstly triggering a canary token and then hibernating the host.

By doing so we:


- Alert the defensive function with a high-signal alert.
- Minimize the impact via likelihood of successful encryption.
- Give the best chance of recovery.

Prototype

The implementation follows what we describe. It is installed in place of commonly targeted services and there is a shared counter for the number of running instances which gets decremented when stopped via *sc.exe* or *net.exe*. When the total number of running instances drops below two we then fire a DNS canary token and then hibernate the host.

Canary Token Use

We've used an extensibility feature of DNS canary tokens to encode arbitrary data effectively using them as Alerting as a Service. Specifically the host name which is hibernating is encoded into the DNS canary token. This means we know not only did it fire but also which host. This obviously becomes useful when running on a large number of hosts.

Date: 2021 Mar 03 11:05:20.878872 (UTC) IP: 81.134.98.19	
Channel: DNS	
Geo Info	
Country	GB 
City	London
Region	England
Organisation	AS2856 British Telecommunications PLC
Tor	
Known Exit Node	False
Basic Info	
Memo	SWOLLENRIVER
Generic Data	DESKTOP-LBU71F2 hibernating

Real World Efficacy

In short it is today unknown, but we hope others will take the concept, experiment and share real-world efficacy data back with the community. Some initial response has been a commoditized product which does this will likely not work for any significant period of time, which seems rational:

in a conversation today with The Record, [MalwareHunterTeam](#), a security researcher who has tracked and analyzed hundreds of ransomware strains across the years, said that ransomware gangs are also likely to deploy countermeasures for Killed Process Canary if the tool ever becomes a hindrance to their operations.

However the real answer is we don't know how effective it will or won't be today. What is clear however is there is opportunity to make the operating environment for threat actors significantly more hostile than it is today.

The Code

<https://github.com/nccgroup/KilledProcessCanary>

Thanks and Feedback etc.

Thanks to Harry for the discussions which led to the prototype.

Feedback welcome via pull requests, e-mail (ollie dot whitehouse at nccgroup dot com) or @ollieatnccgroup on Twitter.