Garmin Attack Incentives:

<https://techbeacon.com/security/8-lessons-garmin-ransomware-attack>

* Money , large corp, why garmin

<https://us.norton.com/internetsecurity-malware-ransomware-5-dos-and-donts.html#:~:text=The%20idea%20behind%20ransomware%2C%20a,a%20ransom%20to%20restore%20access.&text=And%20since%20malware%20attacks%20are,ensure%20access%20will%20be%20restored>.

* What is ransomware

<https://techhq.com/2020/11/forget-osi-layers-its-the-customer-experience-or-nothing-in-todays-enterprise-networks/>

* Specific targets

<https://www.theguardian.com/technology/2020/jul/27/ransomware-attack-on-garmin-thought-to-be-the-work-of-evil-corp>

* J

3287

Rozzy113

({0: 0, 1: 1.0, 2: 2.0, 3: 2.0, 4: 3.0, 5: 3.0}, {0: '-', 1: 0, 3: 1, 2: 0, 4: 3, 5: 3})

({0: 0, 1: 1.0, 2: 2.0, 3: 2.0, 4: 3.0, 5: 3.0}, {0: 0, 1: 0, 3: 1, 2: 0, 4: 3, 5: 3})

Cobalt Strike is a tool that white-hat hackers can use to conduct security testing. The software comes with Cobalt Strike Beacon tool which can run shell commands, increases privileges, inject other payloads, and even loads itself into the memory of a process. These features are used in the WastedLocker attack on Garmin. During the attack, the previously downloaded Powershell script contains the Cobalt strike payload (which is the Beacon) and first decodes the base64 payload two times. Then the payload is decrypted using an AES Block cipher (Advanced Standard Encryption). This is done by using a key that is created from the first 16 bytes of the payload to create the salt, and the string ‘saN9s9pNlD5nJ2EyEd4rPym68griTOMT’ in a sha256 hash. This payload is then converted from a string into bytes, which is ultimately injected into memory. The payload is injected into memory by a loader and .NET injector that is downloaded from the ‘Donut’ Project. Although most .NET injectors do not allow .NET assemblies (Common Intermediate Language representations of code) to be injected into running processes, the injector and loader from ‘Donut’ Project generate shellcode payloads from the .NET assemblies that can be injected into remote running processes and it allows the user to decide how the payload is injected.

The interesting feature of the injector and loader is that they detect whether the Crowdstrike software exists on the victim computer in an effort to be undetected by Crowdstrike. If Crowdstrike is installed, the loader will load the Beacon and then call the command FreeConsole(). If CrowdStrike is not installed, the FreeConsole() command will be called first and then the loader will load the beacon into memory. The FreeConsole() command is used to close a console after the last process terminates.

Finally, after the Cobalt Strike Beacon is loaded into memory, it is then used to download the code for WastedLocker.

<https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm> AES alg

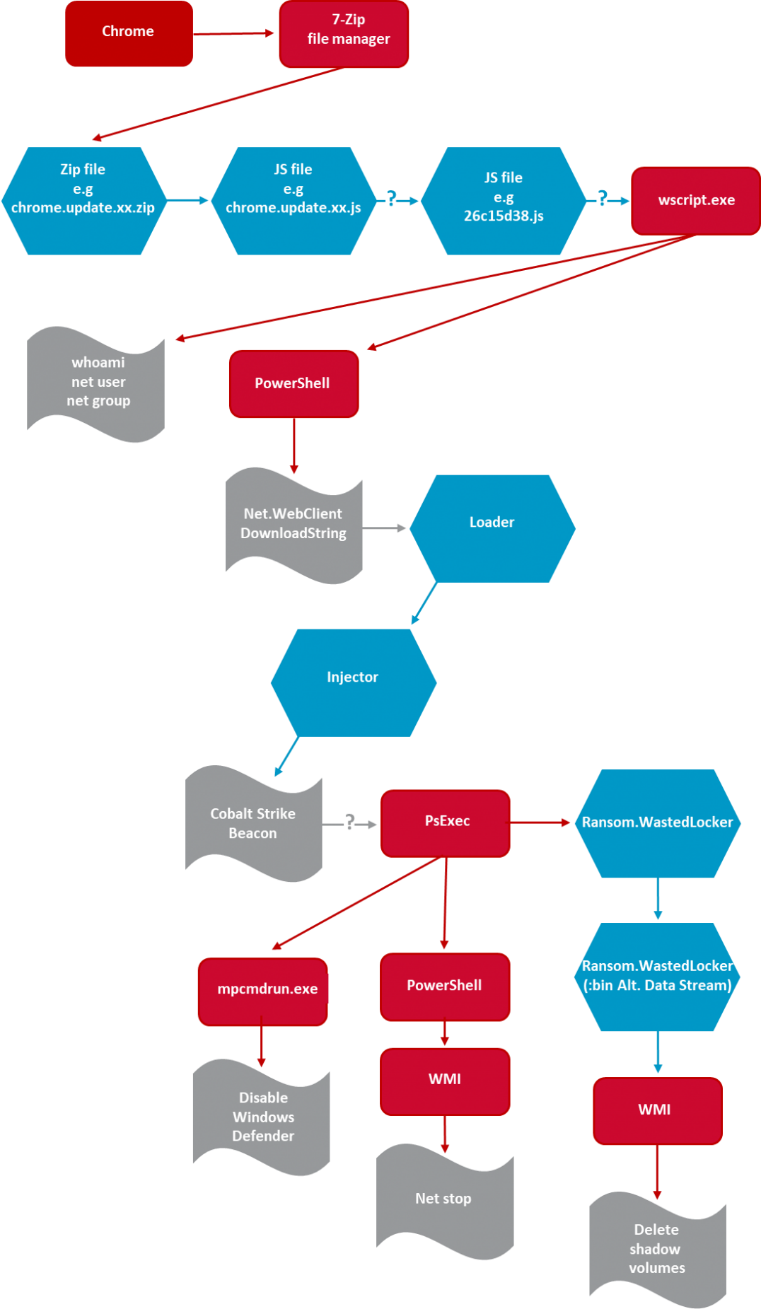
<https://research.nccgroup.com/2020/06/23/wastedlocker-a-new-ransomware-variant-developed-by-the-evil-corp-group/> almost everything

<https://www.cyberswachhtakendra.gov.in/alerts/WastedLockerRansomware.html> good picture

<https://www.crowdstrike.com/blog/getting-the-bacon-from-cobalt-strike-beacon/> beacon powers

<https://thewover.github.io/Introducing-Donut/> donut injector and loader

<https://docs.microsoft.com/en-us/windows/console/freeconsole#:~:text=A%20process%20can%20use%20the,it%20terminates%20or%20calls%20FreeConsole>. Free console command





On July 23, 2020, Garmin’s systems were taken down by a ransomware attack. Garmin specializes in products that utilize GPS, most notably their watches which are used by many people for fitness purposes. Besides health and fitness, Garmin also provides products for pilots, specifically flyGarmin. flyGarmin is used by pilots to download aviation databases, which pilots must do before they can fly. The ransomware rendered Garmin’s products useless. WastedLocker was the ransomware used for this attack, and it is believed that the Russian Hacker group, Evil Corp, was behind this attack. Evil Corp is a notorious hacker group, having performed at least 31 ransomware attacks on various organizations using WastedLocker. In fact, anyone who provides information leading to the arrest of the alleged leaders of Evil Corp will receive a five million dollar reward. This is the largest reward ever offered for cyber criminals. The WastedLocker ransomware encrypts all of the files on the victim’s system, and then the victim must pay a ransom to receive the decryption key. The goal of the attack is not to steal information from the victim. The attackers are not trying to gain access to the people’s payment information, or any other information that they can sell. The goal of WastedLocker is to render the victim’s systems useless, so they have to pay the ransom to get their system back online. It is believed that Evil Corp demanded a ten million dollar ransom from Garmin. It is also believed that Garmin did pay this ransom.

**Technical Analysis:**

Start of the Attack:

The attack begins when the victim visits a website that has been compromised with SocGholish. SocGholish is a Remote Access Trojan, which means that it is malware that gives the attacker remote access to the infected machine, and also administrative privileges. SocGholish gets the victim to download the malware by pretending to be a fake browser update. First, the victim must visit a website that contains this malware. Then, when on the site, a pop-up window will appear. This pop-up window looks like a browser update. This is how the victim is tricked into downloading the malware onto their machine. The user believes they are being prompted with a legitimate browser update, but when they hit update, it will download and run the SocGholish zip file, which contains malicious JavaScript. The file that came from SocGholish will evaluate the infected computer, and see if it is part of a larger network. This is because a computer that is not connected to anything else is useless to the attacker. Following this, SocGholish will distribute the Cobalt Strike payload.

Cobalt Strike:

Cobalt Strike is a tool that white-hat hackers can use to conduct security testing. The software comes with Cobalt Strike Beacon which can run shell commands, increases privileges, inject other payloads, and even loads itself into the memory of a process. These features are used in the WastedLocker attack on Garmin. During the attack, the previously downloaded Powershell script contains the Cobalt strike payload (which is the Beacon) and first decodes the base64 payload two times. Then the payload is decrypted using an AES Block cipher (Advanced Standard Encryption).

This is done by using a key that is created from the first 16 bytes of the payload to create the salt, and the string ‘saN9s9pNlD5nJ2EyEd4rPym68griTOMT’in a sha256 hash. This payload is then converted from a string into bytes, which is ultimately injected into memory. The payload is injected into memory by a loader and .NET injector that is downloaded from the ‘Donut’ Project. Although most .NET injectors do not allow .NET assemblies (Common Intermediate Language representations of code) to be injected into running processes, the injector and loader from ‘Donut’ Project generate shellcode payloads from the .NET assemblies that can be injected into remote running processes and it allows the user to decide how the payload is injected.

The interesting feature of the injector and loader is that they detect whether the Crowdstrike software exists on the victim computer in an effort to be undetected by Crowdstrike. If  Crowdstrike is installed, the loader will load the Beacon and then call the command FreeConsole(). If CrowdStrike is not installed, the FreeConsole() command will be called first and then the loader will load the beacon into memory. The FreeConsole() command is used to close a console after the last process terminates.

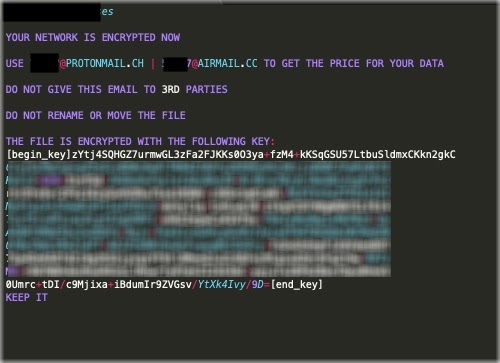
Finally, after the Cobalt Strike Beacon is loaded into memory, it is then used to download the code for WastedLocker.

Administrative Access:

Normally, Windows’ User Account Control (UAC) will warn about or block the use of suspicious software. WastedLocker bypasses this feature entirely. It will first create a directory in the computer’s %appdata%, copy an arbitrary file from the computer to this directory, injecting itself into the data stream of this copy sequence. From here, WastedLocker creates a deceptive “C:\Windows \system32 directory”. There is a deliberate space after “Windows”. It then copies the legitimate winsat.exe and winmm.dl to this directory. The starting code of winmm.dl is appended, so that it redirects to the malware, which is located in %appdata%. With the previous steps, launching winsat.exe will load the tampered winmm.dl, and will run WastedLocker at elevated privileges, therefore preventing any UAC messages.

Encryption:

For the encryption, WastedLocker targets any file larger than 10 bytes by breaking them into blocks of 64 megabytes. The first step is the encrypting each block using a 256 bit AES key created from the hash of a hardcoded string and the Installation Vector (IV), created from the first 16 bytes of the malware code itself. Then the key and IV are encrypted with a public RSA key. The resulting encrypted key and IV are displayed in the ransom note. When the ransom is paid the victim will receive the necessary secret keys and be able to decrypt their files.



**Prevention:**

There are preventative measures that can be taken to avoid a WastedLocker attack. First, it is important that employees are trained to recognize the potential signs of a ransomware attack. The first step of the WastedLocker attack involves using the SocGholish framework to infect the user’s system with malware. SocGholish utilizes a fake browser update to spread the malware. Therefore, a user has to accept this fake browser update for the malware to infect the system. A preventative measure would be to make employees aware of this attack and to recognize the fact that it is possible for a fake browser update to occur, and to not install or download anything onto the computer unless they are one hundred percent sure that it is safe. This is a very simple, yet extremely effective measure that can be taken to prevent an attack like this. If the employees are trained to recognize when something is not right with their machine, then this attack could be prevented.

Another preventative measure is to better secure the system in general, and invest more resources into cybersecurity. One way to do this is to make use of certain products that can detect that the system is infected with WastedLocker so the attack can be stopped before it is too late. Cisco offers a variety of products that could be used to prevent WastedLocker. Cisco Umbrella keeps track of compromised sites, and blocks users from accessing any of these sites. It will also show the user an alert of why the domain they attempted to access was blocked. This would help prevent WastedLocker because if the compromised site was in Cisco Umbrella’s records, then the user would be blocked from visiting the site, and the ransomware would have no way to get on their machine. If the victim never visits the compromised site, then there is no way for this attack to begin because there is no way for the ransomware to get on the victim’s machine. However, say Cisco Umbrella had not yet discovered this compromised site, and the user ended up visiting the site and they downloaded the malicious zip file thinking it was a browser update. AMP for Endpoints can prevent malicious files from being downloaded. AMP for Endpoints analyzes the file and blocks it from being downloaded if it is deemed malicious. AMP for Endpoints checks if this file is similar to any other malicious files that have been downloaded before. It is able to detect unique malware through loose fingerprinting, which checks for similarities between the current file, and various types of malware to see if there is any similarity between the two.

WastedLocker can be prevented by training employees on safe internet practices, and it can also be prevented by strengthening the security of a system by employing products that can detect and prevent this attack from happening. Garmin could have prevented this attack by happening. If more emphasis was put on employee training then the ransomware would not have been able to affect the system. If they also put more resources into better securing their system, then they would have been able to detect and stop the ransomware.

**Incentives:**

**SLIDE 1:** The main motive for performing a ransomware attack is to obtain a large sum of money from the victim in exchange for the victim’s data.

But why Evil Corp specifically choose Garmin.

* One reason is because Garmin is a large corporation, so they will have the money to pay the hackers.
* Another reason is that Garmin’s data is mission-critical since they have an aviation sector that uses their GPS services. This puts more pressure on Garmin to pay up because the information that is being held hostage is so important.

**SLIDE 2:** Additionally, Garmin’s customers overall were greatly affected by the attack, so Garmin was not able to spend a long time trying to figure out how to decrypt the data without letting customers know what was happening.

**SLIDE 3:** The timing of this attack also added pressure to Garmin to pay up quickly. The attack came on July 23 2020 🡪 6 days before Garmin’s quarterly earnings were due to be reported.

this is not great timing:

* the first is that Garmin obviously does not want their revenue stats to be affected in general
* is that Garmin does not want investors to see a decline in revenue and pull out their funds.

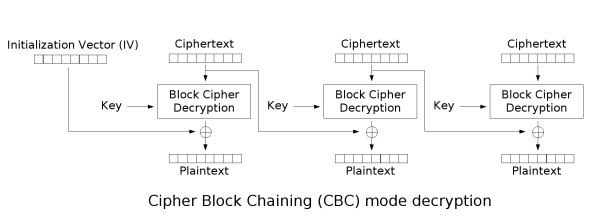
GARmin’s finances as they saw a 9% overall decline in revenue for quarter 2

* stated to be mostly due to coronavirus and not the attack.

Cobalt Strike is a tool that white-hat hackers can use to conduct security testing. The software comes with Cobalt Strike Beacon which can run shell commands, increases privileges, inject other payloads, and even loads itself into the memory of a process.

During the attack, the previously downloaded Powershell script contains the Cobalt strike payload (which is the Beacon) and first decodes the base64 payload two times.

* Then the payload is decrypted using AES (Advanced Standard Encryption) in CBC mode.



* This is done by using a key that is created from the first 16 bytes of the payload to create the salt, and a hard coded stringin a sha256 hash.
* This payload is then converted from a string into bytes, which is ultimately injected into memory. The payload is injected into memory by a loader and .NET injector that is downloaded from the ‘Donut’ Project.
* injector and loader generate shellcode payloads from the .NET assemblies that can be injected into remote running processes and it allows the user to decide how the payload is injected.
* the injector and loader detect whether the Crowdstrike software exists on the victim computer in an effort to be undetected by Crowdstrike.

Finally, after the Cobalt Strike Beacon is loaded into memory, it is then used to download the code for WastedLocker.