

# WastedLocker: technical analysis

By Fedor Sinitsyn

Published: 2020-07-31 · Archived: 2026-04-05 13:35:24 UTC

The use of crypto-ransomware in targeted attacks has become an ordinary occurrence lately: new incidents are being reported every month, sometimes even more often.

On July 23, Garmin, a major manufacturer of navigation equipment and smart devices, including smart watches and bracelets, experienced a massive service outage. As confirmed by an [official statement](#) later, the cause of the downtime was a cybersecurity incident involving data encryption. The situation was so dire that at the time of writing of this post (7/29) the operation of the affected online services had not been fully restored.

According to currently available information, the attack saw the threat actors use a targeted build of the trojan WastedLocker. An increase in the activity of this malware was noticed in the first half of this year.

We have performed technical analysis of a WastedLocker sample.

## Command line arguments

It is worth noting that WastedLocker has a command line interface that allows it to process several arguments that control the way it operates.

***-p <directory-path>***

Priority processing: the trojan will encrypt the specified directory first, and then add it to an internal exclusion list (to avoid processing it twice) and encrypt all the remaining directories on available drives.

***-f <directory-path>***

Encrypt only the specified directory.

***-u username:password \\hostname***

Encrypt files on the specified network resource using the provided credentials for authentication.

***-r***

Launch the sequence of actions:

1. Delete ;
2. Copy to %WINDIR%\system32\<rand>.exe using a random substring from the list of subkeys of the registry key SYSTEM\CurrentControlSet\Control\;
3. Create a service with a name chosen similarly to the method described above. If a service with this name already exists, append the prefix “Ms” (e.g. if the service “Power” already exists, the malware will create a

new one with the name “MsPower”). The command line for the new service will be set to “%WINDIR%\system32\<rand>.exe -s”;

4. 4 Start this service and wait until it finishes working;
5. 5 Delete the service.

-s:







Start the created service. It will lead to the encryption of any files the malware can find.

## UAC bypass

Another interesting feature of WastedLocker is the chosen method of UAC bypass. When the trojan starts, it will check the integrity level it was run on. If this level is not high enough, the malware will try to silently elevate its privileges using a known bypass technique.

1. 1 Create a new directory in %appdata%; the directory name is chosen at random from the substrings found in the list of subkeys of the registry key SYSTEM\CurrentControlSet\Control\;
2. 2 Copy a random EXE or DLL file from the system directory to this new directory;
3. 3 Write the trojan’s own body into the alternate NTFS stream “:bin” of this system file;
4. 4 Create a new temporary directory and set its mount point to “C:\Windows ” (with a trailing whitespace) using the API function NtFsControlFile with the flag IO\_REPARSE\_TAG\_MOUNT\_POINT;
5. 5 Create a new subdirectory named “system32” inside the temporary directory. As a result of the previous step, this new subdirectory can be equally successfully addressed as “%temp%\<directory\_name>\system32” or “C:\Windows \system32” (note the whitespace);
6. 6 Copy the legitimate winsat.exe and winmm.dll into this subdirectory;
7. 7 Patch winmm.dll: replace the entry point code with a short fragment of malicious code whose only purpose is to launch the content of the alternate NTFS stream created on step 2;
8. 8 Launch winsat.exe, which will trigger the loading of the patched winmm.dll as a result of DLL hijacking.

The above sequence of actions results in WastedLocker being relaunched from the alternate NTFS stream with elevated administrative privileges without displaying the UAC prompt.

1768	 CreateFile	C:\Windows	SUCCESS	Desired Access: Generic Write, Disposition: OpenIf, Options: Dir
1768	 FileSystemControl	C:\Windows	SUCCESS	Control: FSCTL_SET_REPARSE_POINT
1768	 CreateFile	C:\Windows	REPARSE	Desired Access: Read Data/List Directory, Synchronize, Disposit
1768	 Process Create	C:\Windows\system32\winsat.exe	SUCCESS	PID: 4344, Command line: "C:\Windows\system32\winsat.exe"
1768	 FileSystemControl	C:\Windows	SUCCESS	Control: FSCTL_DELETE_REPARSE_POINT
1768	 CloseFile	C:\Windows	SUCCESS	

### *Procmon log fragment during the launch of WastedLocker*

## Cryptographic scheme

To encrypt victims’ files, the developers of the trojan employed a combination of the AES and RSA algorithms that has already become a ‘classic’ among different crypto-ransomware families.

The search mask to choose which files will be encrypted, as well as the list of the ignored paths are set in the configuration of the malware.

```

00002C80: 0078 0078 000A 0002-C411 770C 002A 028A-0282 7631 1701 002A-005C 004E 0054 004C xx00??*????*NNTL
00002CA0: 0044 0052 007C 002A-005C 0042 004F 004F-0054 004D 0047 0052-007C 002A 005C 0047 DR|*\BOOTMGR|*\NG
00002CC0: 0052 004C 0044 0052-007C 002A 002E 0033-0038 0036 007C 002A-002E 0070 0073 0031 RLDR|*.386|*.ps1
00002CE0: 007C 002A 002E 006D-0073 0075 007C 002A-002E 0061 006E 0069-007C 002A 002E 0077 l*.msui*.ani*.w
00002D00: 0070 0078 007C 002A-002E 0068 006C 0070-007C 002A 002E 006F-0063 0078 007C 002A px|*.hip|.ocx|.
00002D20: 002E 0063 006F 006D-007C 002A 002E 0063-0070 006C 007C 002A-002E 0061 0064 0076 .com|.cpl|.adv
00002D40: 007C 002A 002E 0063-006D 0064 007C 002A-002E 006C 006E 006B-007C 002A 002E 0064 l*.cmd|.lnk|.d
00002D60: 0072 0076 007C 002A-002E 0073 0079 0073-007C 002A 002E 0069-0063 006C 007C 002A rvl*.sys|.icli*
00002D80: 002E 006E 0065 0073-007C 002A 002E 0069-0061 0062 007C 002A-002E 0062 0061 0074 nls|.cab|.bat
00002DA0: 007C 002A 002E 0074-0068 0065 006D 0065-007C 002A 002E 0062-0069 006E 007C 002A l*.themel*.bin|
00002DC0: 002E 006B 0065 0079-0073 002A 002E 0074-0068 0065 006D 0065-0070 0061 0063 006B *.key|.themepack
00002DE0: 007C 002A 002E 006D 0069 007C 002A 002E 0064-0078 0069 0063 006E-0079 007C 002A 002E l*.msil|.icns|.
00002E00: 0069 0063 0073 007C-007A 002E 0069 0064-0078 007C 002A 002E 0068 0074 0061 007C ics|.idx|.hta
00002E20: 002A 002E 0063-0072 007C 002A 002E 006D 0073 0073 0074-0079 006C 0065 0073 l*.scri|.msstyles
00002E40: 007C 002A 002E 0064-0069 0061 0067 0063-0066 0067 007C 002A-002E 0064 0069 0061 *.diagcfg|.dia
00002E60: 0067 0063 0061 0062-007C 002A 002E 006E 006F 006D 0065 0064-0069 0061 007C 002A gcabi*.nomedia|
00002E80: 002E 006D 0073 0063-007C 002A 002E 0063-0075 0072 0070 002A-002E 006D 006F 0064 mscl*.cur|.mod
00002EA0: 007C 002A 002E 0073-0068 0073 007C 002A-002E 0072 0074 0070-007C 002A 002E 0072 i*.shs|.rt|.r
00002EC0: 006F 006D 007C 0063-007C 002A 002E 0070-007C 002A 002E 0069 0069 007C 002A om|.msp|.ini|
00002EE0: 002E 0064 0061 0074-007C 002A 002E 0073-0064 0069 007C 002A-002E 0077 0069 006D dat|.sdi|.wim
00002F00: 007C 002A 002E 0064-006C 006C 007C 002A 002E 0065 0078 0065-0156 3DAE 37D6 i*.dll|.exe|R0??
00002F20: 0062 0069 0065 007C-0042 006F 006F 0074-007C 0062 006F 006F-0074 007C 0064 0065 bin|Boot|boot|de
00002F40: 0076 007C 0065 0074-0063 007C 006C 0069-0062 007C 0069 006E 0069 0074 0064 0072 v|etc|lib|nitr
00002F60: 007C 0073 0062 0069-006E 007C 0073 007C 0076 006D-006C 0069 006E 0075 l|bin|sys|yml|pu
00002F80: 007A 007C 0072 0065-006E 007C 0076 0061-0072 007C 005C 0042-006F 006F 0074 007C z|run|var|Boot|
00002FA0: 0053 0079 0073 0074-0065 006D 0020 0056-006F 006C 0075 006D-0065 0020 0049 006E System Volume In
00002FC0: 0066 006F 0072 006D-0061 0074 0069 006F-006E 007C 0024 0052-0045 0043 0059 0043 formation|$_REC|C
00002FE0: 004C 0045 002E 0042-0049 004E 007C 0057-0065 0062 0043 0061-0063 0068 0065 007C LE|.BIN|WebCache|
00003000: 0043 0061 0063 0068-0065 0073 007C 0057-0069 006E 0064 006F 0077 0073 0041 0070 Caches|WindowsAp
00003020: 0070 0073 007C 0041-0070 0070 0044 0061-0074 0061 007C 0050-0072 006F 0067 0072 ps|AppData|Prog
00003040: 0061 006D 0044 0061-0074 0061 007C 005C-0055 0073 0065 0072-0073 005C 0041 006C amData|Users|A|
00003060: 006C 0020 0055 0073-0065 0072 0073 00BE-00B6 CDCD 6185 0025-0050 0072 006F 0067 l Users|??|Prog
00003080: 0072 0061 006D 0044-0061 0074 0061 007C 0025-007C 0025 0077 0069-006E 0064 0069 0072 ramData%|windir
000030A0: 0025 007C 0025 0074-0065 006D 0070 0025-007C 0025 0041 0070-0070 0044 0061 0074 %|temp%|AppDat
000030C0: 0061 0025 007C 0043-003A 005C 0052 0065-0063 006F 0076 0065-0072 0079 007C 0043 a%|C:\Recovery|C
000030E0: 003A 005C 0050 0072-006F 0067 0072 0061-006D 0020 0046 0069-006C 0065 0073 007C :|Program Files|
00003100: 0043 003A 005C 0050-0072 006F 0067 0072 0061 006D 0020 0046 0069-006C 0065 0073 C:\Program Files
00003120: 0020 0028 0078 0038-0036 0029 0024 001C-C88B E804 0025 0074-0065 006D 0070 0025 (x86)|$.??|temp%

```

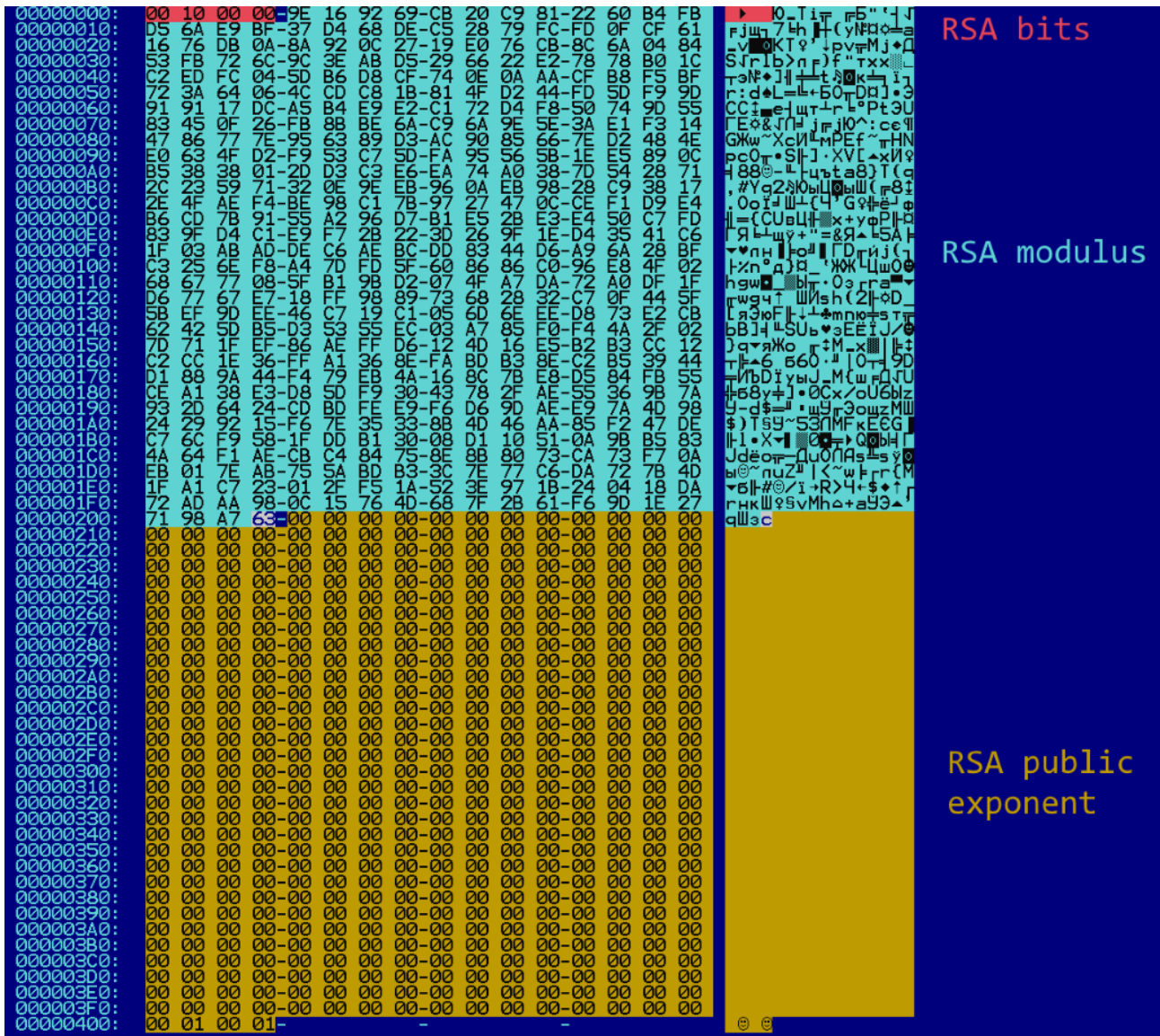
**Part of the trojan config showing the ignored path substrings**

For each processed file, WastedLocker generates a unique 256 bit key and a 128 bit IV which will be used to encrypt the file content using the AES-256 algorithm in CBC mode. The implementation of the file operations is worthy of note, as it employs file mapping for data access. It must have been an attempt by the criminals to maximize the trojan’s performance and/or avoid detection by security solutions. Each encrypted file will get a new additional extension: “.garminwasted”.

The trojan also implements a way of integrity control as part of its file encryption routine. The malware calculates an MD5 hash of the original content of each processed file, and this hash may be utilized during decryption to ensure the correctness of the procedure.

WastedLocker uses a publicly available reference implementation of an RSA algorithm named “rsaref”.

The AES key, IV and the MD5 hash of the original content, as well as some auxiliary information, are encrypted with a public RSA key embedded in the trojan’s body. The sample under consideration contains a 4096 bit public RSA key.

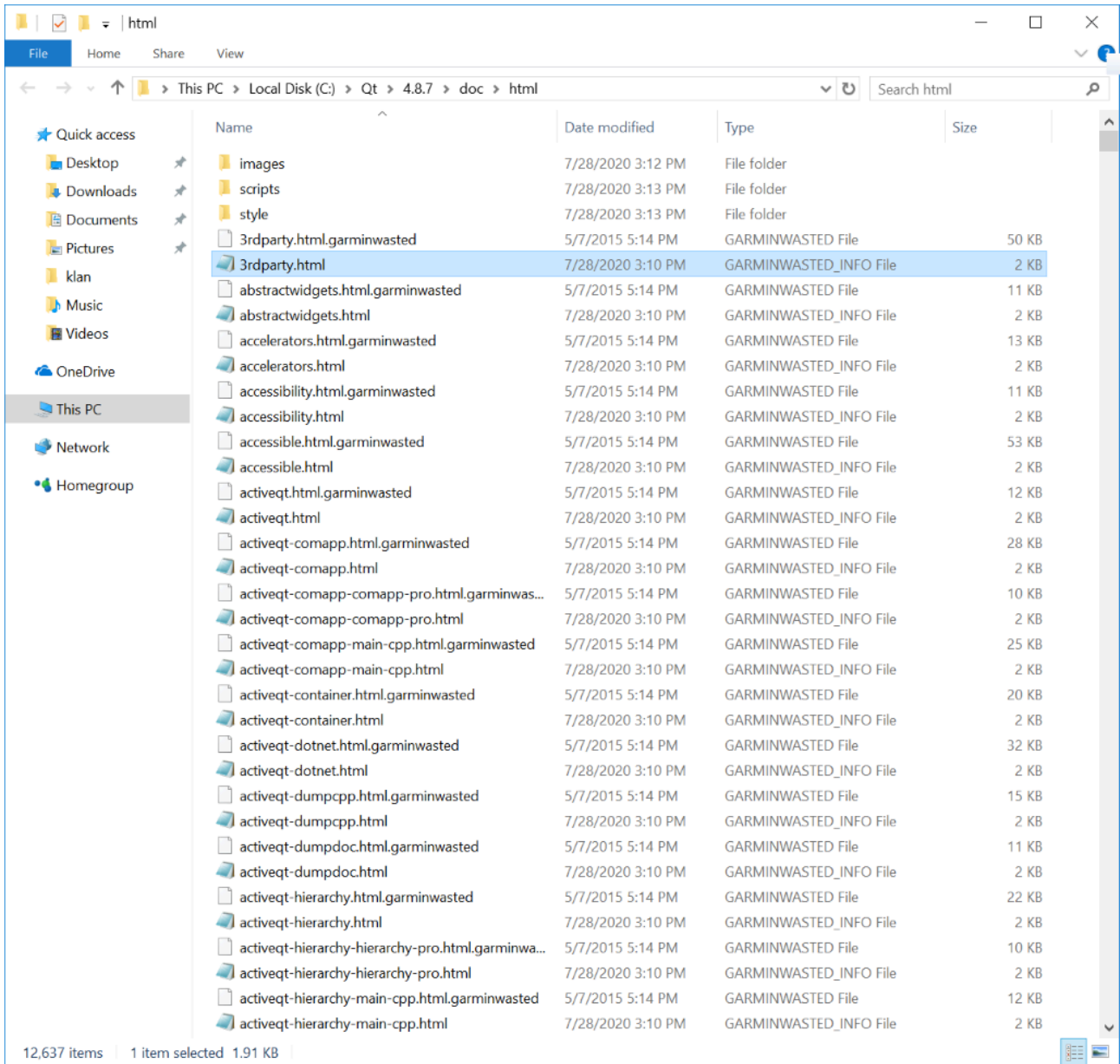


### The public RSA key format used by WastedLocker

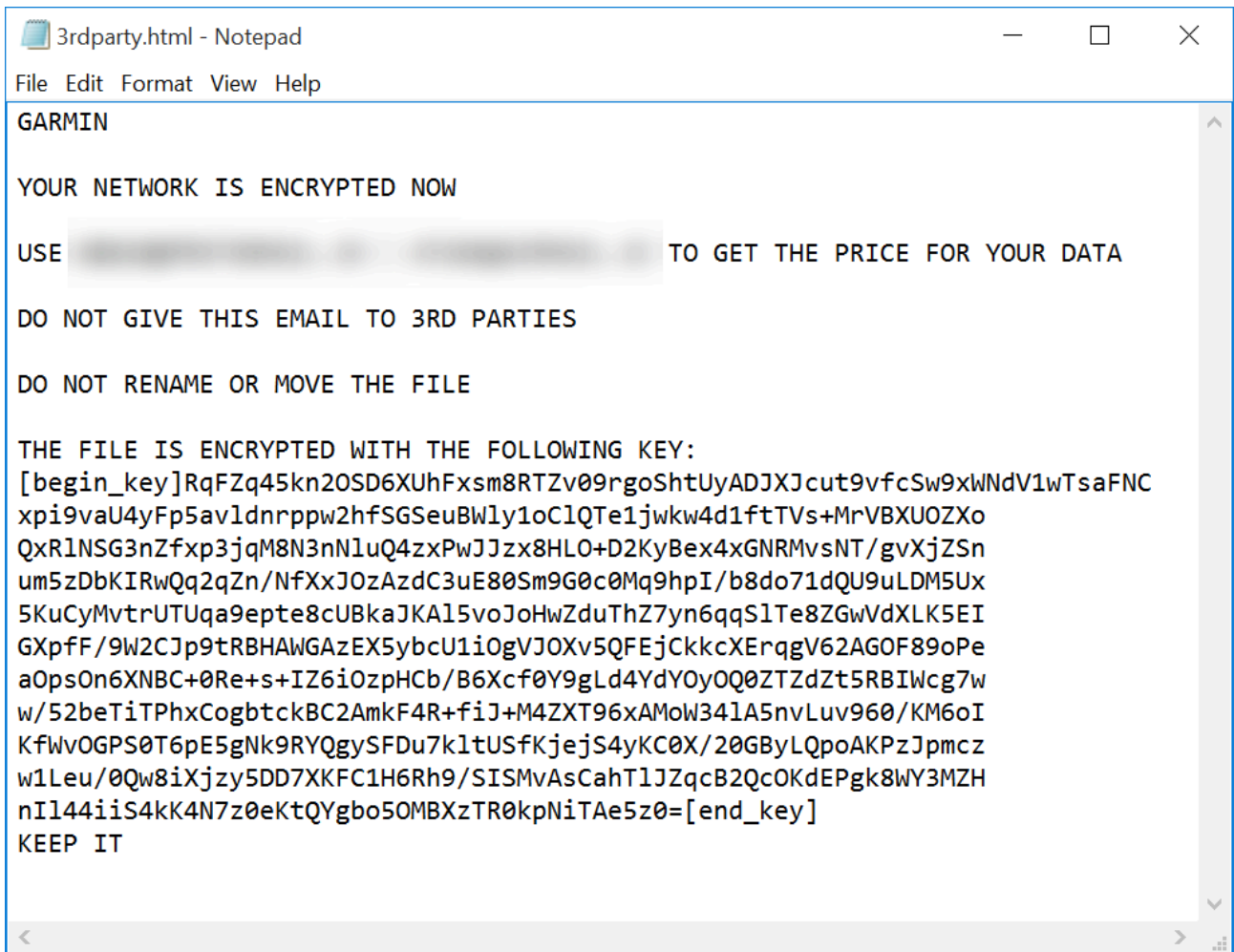
It should be noted that this kind of cryptographic scheme, using one public RSA key for all victims of a given malware sample, could be considered a weakness if WastedLocker were to be mass-distributed. In this case a decryptor from one victim would have to contain the only private RSA key that would allow all the victims to decrypt their files.

However, as we can see, WastedLocker is used in attacks targeted at a specific organization which makes this decryption approach worthless in real-world scenarios.

The result of RSA encryption is Base64 encoded and saved in a new file with the extension **.garminwasted\_info**, and what is notable, a new info file is created for each of the victim's encrypted files. This is a rare approach that was previously used by the BitPaymer and Doppelpaymer trojans.



*An example list of encrypted files from our test machine*



*Ransom note left by the trojan*

## Recommendations

This WastedLocker sample we analyzed is targeted and crafted specifically to be used in this particular attack. It uses a “classic” AES+RSA cryptographic scheme which is strong and properly implemented, and therefore the files encrypted by this sample cannot be decrypted without the threat actors’ private RSA key.

The Garmin incident is the next in a series of targeted attacks on large organizations involving crypto-ransomware. Unfortunately, there is no reason to believe that this trend will decline in the near future.

That is why it is crucial to follow a number of recommendations that may help prevent this type of attacks:

1. 1 Use up-to-date OS and application versions;
2. 2 Refrain from opening RDP access on the Internet unless necessary. Preferably, use VPN to secure remote access;
3. 3 Use modern endpoint security solutions, such as [Kaspersky Endpoint Security for Business](#), that support behavior detection, automatic file rollback and a number of other technologies to protect from ransomware.
4. 4 Improve user education in the field of cybersecurity. [Kaspersky Security Awareness](#) offers computer-based training products that combine expertise in cybersecurity with best-practice educational techniques and technologies.

5. 5 Use a reliable data backup scheme.

Kaspersky products protect from this threat, detecting it as Trojan-Ransom.Win32.Wasted.d and PDM:Trojan.Win32.Generic. The relevant behavioral detection logic was added in 2017.

**IoC**

[2cc4534b0dd0e1c8d5b89644274a10c1](https://securelist.com/wastedlocker-technical-analysis/97944/2cc4534b0dd0e1c8d5b89644274a10c1)

---

Source: <https://securelist.com/wastedlocker-technical-analysis/97944/>