WastedLocker Ransomware: Abusing ADS and NTFS File Attributes

labs.sentinelone.com/wastedlocker-ransomware-abusing-ads-and-ntfs-file-attributes/

Jim Walter

Background

WastedLocker is a relatively new ransomware family which has been tracked in the wild since April/May 2020. The name comes from the 'wasted' string which is appended to encrypted files upon infection. Similar to families like Maze and NetWalker, WastedLocker has been attacking high-value targets across numerous industries. Their campaigns have targeted several United States-based Fortune 500 companies as well.

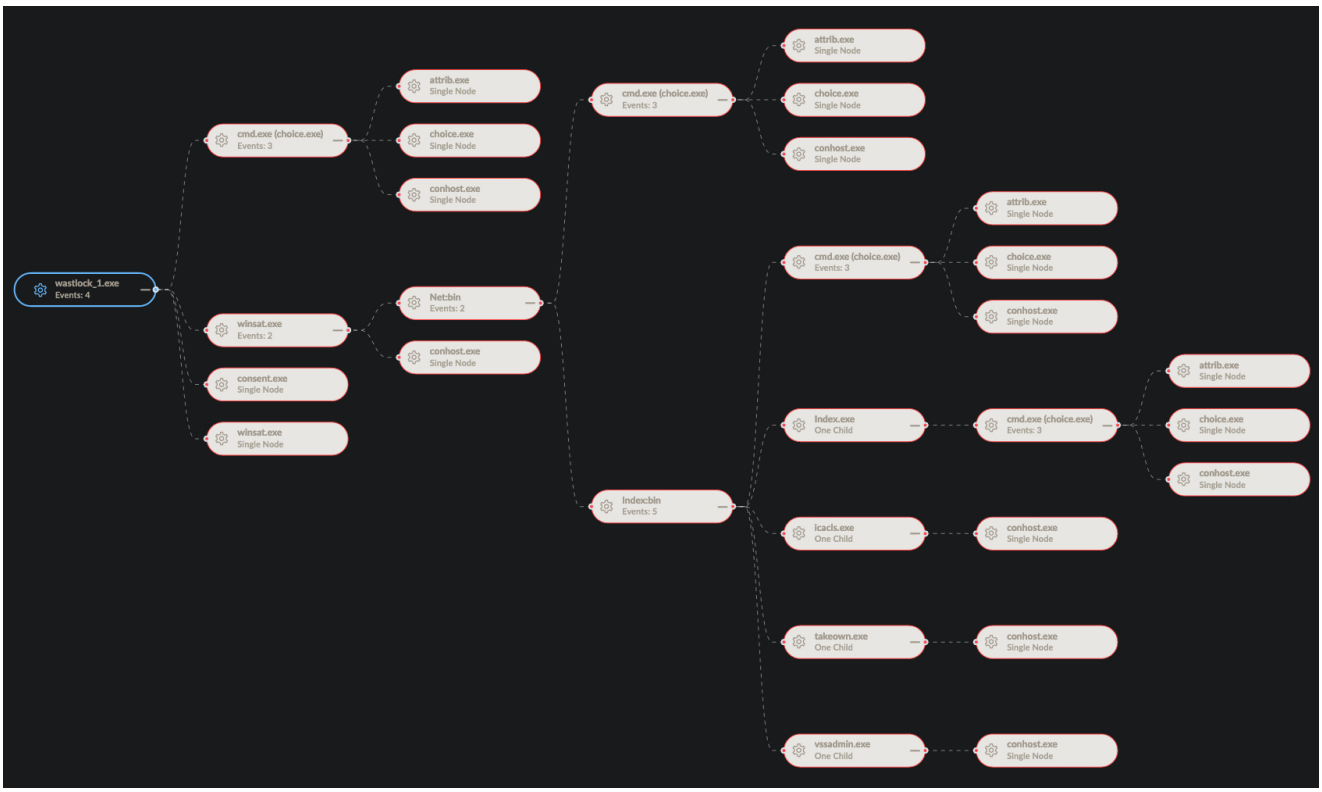
Leveraging SocGholish & Cobalt Strike

Payload delivery is achieved through multiple methodologies. Once attackers have gained a foothold in the targeted environment, Cobalt Strike is often used to directly deliver the ransomware payloads. We have also seen the actors behind WastedLocker leverage the SocGholish framework, which is a JavaScript-based framework that allows attackers to spread malware payloads masquerading as system or software updates.

The SocGholish toolset has been observed in use with a plethora of malware campaigns since 2018. That is to say, it is not exclusive to WastedLocker. In the past, SocGholish has been used with NetSupport RAT, Lokibot, and other commodity malware types and families.

Websites containing the malicious JavaScript code can then be used to deliver the malware once users are enticed into visiting the site(s).

Once victims have been compromised via SocGholish, Cobalt Strike is used to laterally move as well as gain additional profile data on the targeted hosts or environment. Prior to delivering the WastedLocker payload, attackers typically disable core Windows Defender features, as well as deleting Volume Shadow Copies. Additional LOTL-style tools are also often observed in the campaigns. For example, in some cases PsExec will be used to initiate the launch of the WastedLocker ransomware. PowerShell and WMIC are also sometimes utilized in profiling and tuning the environment.



Hiding via NTFS' Alternate Data Stream

WastedLocker has an affinity for running with administrative privileges. If the payload is executed with non-administrative permissions, it will attempt to elevate privileges via UAC bypas (Mocking Trusted Directories).

Once elevated, the ransomware will write a copy of a random file from System32 to the `%APPDATA%` directory. The newly copied file will have a random and hidden filename. This process allows for the ransomware to copy itself into the file by way of an alternate data stream (ADS).

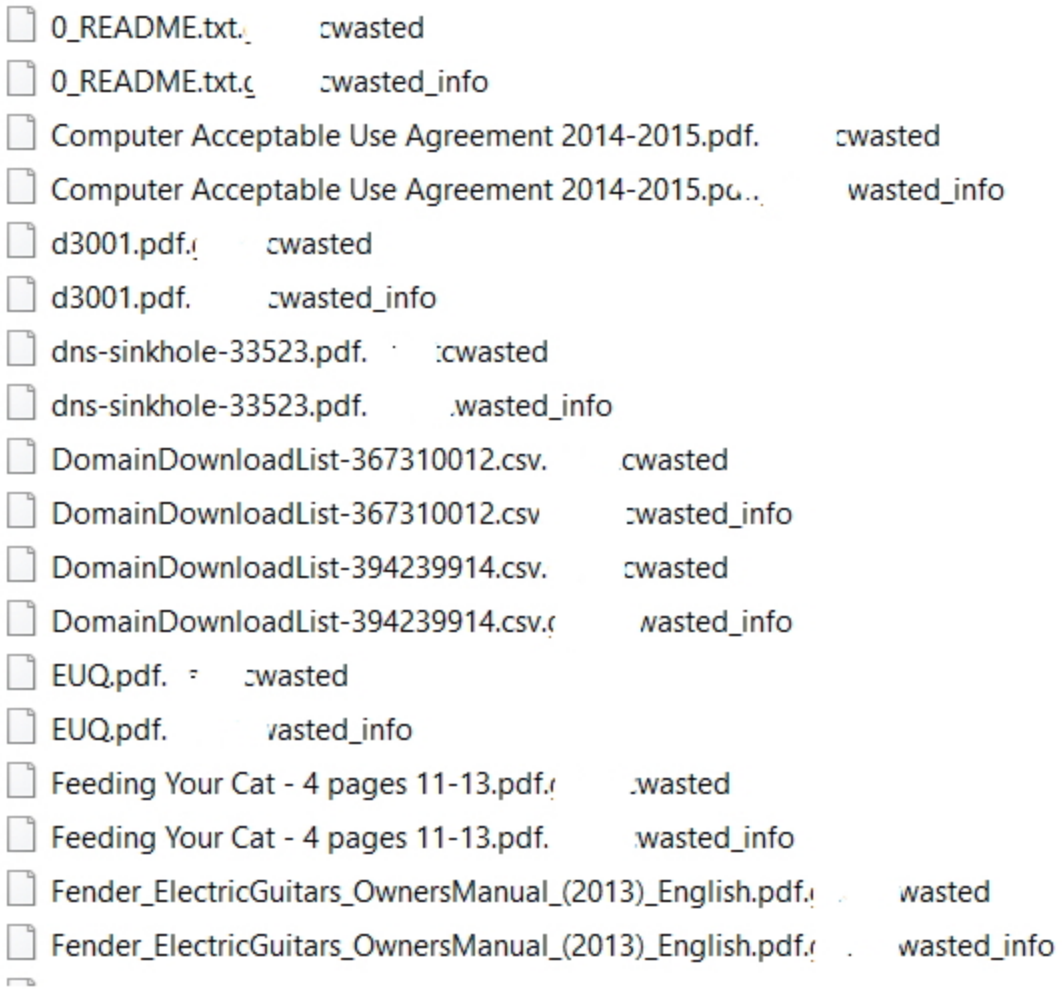
This is followed by the creation of a new folder in `%TEMP%` which contains copies of WINMM.DLL and WINSAT.EXE. The `%TEMP%` copy of WINMM.DLL is then leveraged to execute the ransomware from the previously generated alternate data stream.

WastedLocker Encryption Routine

The encryption style does not differ significantly from other prominent ransomware families. WastedLocker will attempt to encrypt files on local as well as remote (network adjacent and accessible) and removable drives. Once the eligible drives are located, the ransomware will begin the encryption process.

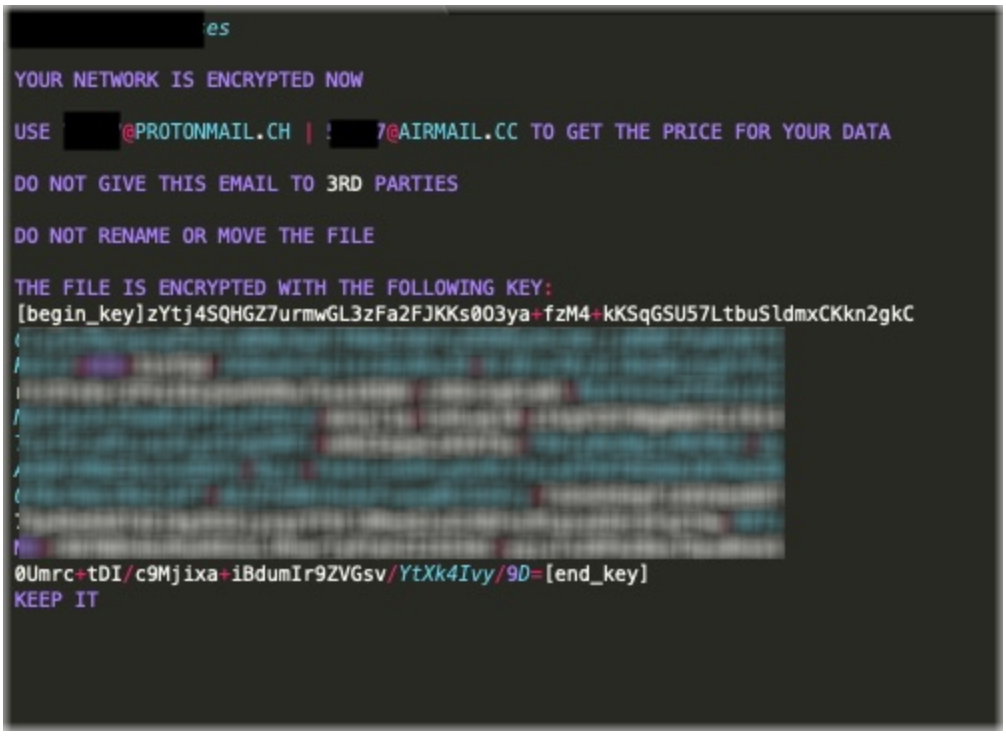
All file types are potential candidates for encryption; however, the ransomware does contain a 'whitelist' of sorts, with directories and extensions to exclude from encryption. This functionality can vary across campaigns. Files are encrypted via AES (Cipher Block Chaining mode + IV / Initialization Vector) with keys generated for each encrypted file. The AES keys (+IV) are then encrypted using a RSA-4096 public key.

The ransom notes contain a base64 representation of the RSA public key. Encrypted files will be renamed with a combination of the targeted companies name along with the string "wasted". For example, if the non-existent company "Turbo Chicken Audio" were infected, the files would look something like "file.pdf.turbochickenaudiowasted' (from file.pdf). The example below shows a set of encrypted files post-infection *(partially redacted for privacy)*.



For each encrypted file, an additional file will be created with "_info" appended to the end of the file extension. These individual files are the ransom notes. Each ransom note also contains the company/target name and an encoded copy of the public key specific to the host. This is in addition to very limited instructions on how to engage the attackers and potentially "get the price for" the encrypted data. Victims are instructed to email the attackers for further instructions.

The email addresses provided are associated with public, secure, email providers (ex: ProtonMail, Eclipso, Tutanota, and Airmail). An example ransom note is below *(partially redacted for privacy)*.



Additional Details

It is also important to note that some analyzed samples support specific command-line parameters. The following are examples of supported parameters:

`-p` (path) Encrypt specified folder/directory before continuing to the rest of the drive/device
`-r` Multi-Purpose: Delete VSS, create a copy of the payload in SYSTEM32, create the ransomware's service entry and execute
`-f` (path) Only encrypt file in the specified directory/folder

Most samples analyzed execute with the `-r` parameter by default, such as:

```
C:UsersadminxAppDataRoamingNetwork:bin -r
```

Persistence is achieved via system service. However, the service is removed once the encryption process has completed. Additional tools are used to manipulate the file system and suppress any requests for user input and/or confirmation. For example, `choice.exe` is leveraged to set file attributes as well as delete files (the service executable) when needed.

Example:

```
cmd.exe (choice.exe) /c choice /t 10 /d y & attrib -h
"C:UsersxxxxxxAppDataRoamingIndex" & del "C:UsersxxxxxxAppDataRoamingIndex"
```

Example:

```
cmd.exe (choice.exe)" & del "C:UsersxxxxxMusicwastlock_5.exe"
```

Upon launching, and as part of the `-r` parameter, the ransomware process has to take ownership of the copy of the payload dropped into SYSTEM32. This is achieved via commands similar to the following:

```
takeown.exe /F C:Windowssystem32Setup2.exe
```

Basic VSSADMIN commands are used for deletion of Volume Shadow copies; for example:

```
vssadmin.exe Delete Shadows /All /Quiet
```

Conclusion

WastedLocker is just one more example of the highly-aggressive ransomware families following in the footsteps of REvil, NetWalker, and others. Prevention, in these attacks, is absolutely critical. Stopping the attackers before they gain any traction is the most effective way to protect you and your sensitive data. This will especially be true should the actors behind WastedLocker decide to leak the data of their victims. SentinelOne's Endpoint Protection and Singularity platform are the most robust and powerful tools at the disposal of today's defenders.

Indicators & IOCs

MITRE ATT&CK
Hide Artifacts: Hidden Files and Directories T1564
Hide Artifacts: NTFS File Attributes T1564
System Services: Service Execution T1569
Abuse Elevation Control Mechanism: Bypass User Access Control T1548
Native API T1106
Command and Scripting Interpreter T1059
File Permissions Modification T1222
Command-Line Interface T1059
Data Encrypted for Impact T1486
Inhibit System Recovery T1490

Hashes SHA256
ed0632acb266a4ec3f51dd803c8025bccd654e53c64eb613e203c590897079b3
e3bf41de3a7edf556d43b6196652aa036e48a602bb3f7c98af9dae992222a8eb
bcdac1a2b67e2b47f8129814dca3bcf7d55404757eb09f1c3103f57da3153ec8
aa05e7a187ddec2e11fc1c9eafe61408d085b0ab6cd12caeaf531c9dca129772
9056ec1ee8d1b0124110e9798700e473fb7c31bc0656d9fc83ed0ac241746064
8897db876553f942b2eb4005f8475a232bafb82a50ca7761a621842e894a3d80
887aac61771af200f7e58bf0d02cb96d9befa11deda4e448f0a700ccb186ce9d
97a1e14988672f7381d54e70785994ed45c2efe3da37e07be251a627f25078a7

85f391ecd480711401f6da2f371156f995dd5cff7580f37791e79e62b91fd9eb
7a45a4ae68992e5be784b4a6da7acd98dc28281fe238f22c1f7c1d85a90d144a
5cd04805f9753ca08b82e88c27bf5426d1d356bb26b28188557305104891136 7

Hashes SHA1

9292fa66c917bfa47e8012d302a69bec48e9b98c
be59c867da75e2a66b8c2519e950254f817cd4ad
70c0d6b0a8485df01ed893a7919009f099591083
4fed7eae00bfa21938e49f33b7c6794fd7d0750c
763d356d30e81d1cd15f6bc6a31f96181edb0b8f
e13f75f25f5830008a4830a75c8ccacb22cebe7b
b99090009cf758fa7551b197990494768cd58687
809fbd450e1a484a5af4ec05c345b2a7072723e7
e62d3a4fe0da1b1b8e9bcff3148becd6d02bcb07
91b2bf44b1f9282c09f07f16631deaa3ad9d956d
f25f0b369a355f30f5e11ac11a7f644bcfefd963