SunCrypt adopts attacking techniques from NetWalker and Maze ransomware

Archived: 2026-04-05 18:17:36 UTC

Summary

- Discovered in October 2019
- Claims to be a member of the Maze ransomware cartel and uses some of the Maze techniques
- Similar to Netwalker, SunCrypt starts with an obfuscated PowerShell loader
- Uses the custom FNV hash function to hide strings in the code and configuration. The original values of the parameters can only be brute-forced.
- Uses ChaCha20 as a cryptographic algorithm to encrypt the user's files.

This ransomware family was first spotted in October 2019, but it was not very active at that time. The group behind it was independent in the beginning, but they recently joined the so-called Maze cartel – combining forces to rob individuals and companies around the world. This cartel included Maze and LockBit when it first started, but later welcomed Ragnar Locker and now SunCrypt.

PowerShell Loader

Suncrypt starts by launching an obfuscated PowerShell script containing about 130,000 rows. It uses the ‘Add-Type’ cmdlet to define a Microsoft .NET Core class to use VirtualAlloc() and EnumDesktopsW() functions.

SunCrypt analysis 01

The PowerShell code is heavily obfuscated with junk if-else statements, functions, and variable assignments to complicate the code analysis.

SunCrypt analysis 03

SunCrypt analysis 02

After removing the junk code, the script has been shrunk to a few lines. The deobfuscated code is as follows:

SunCrypt analysis 04

The SunCrypt starts the script with the execution policy (‘-ep’) set to ‘bypass’ mode — nothing is blocked and there are no warnings or prompts.

Then the loader calls an EnumDesktopsW() function to execute the ransomware payload by exploiting the mechanism of callbacks.

EnumDesktopsW() takes three parameters:

- *hwinsta - A handle to the window station whose desktops are to be enumerated, which is equal to zero, which means to use the current desktop.*
- *lpEnumFunc - A pointer to an application-defined EnumDesktopProc callback function.*
- *lParam - An application-defined value to be passed to the callback function.*

The EnumDesktops function repeatedly invokes the lpEnumFunc callback function until the last desktop is enumerated or the callback function returns FALSE.

In other words, the EnumDesktopsW() function calls the shellcode by the address provided in *lpEnumFunc* parameter. This shellcode then loads the SunCrypt PE file (SHA256: E3DEA10844AEBC7D60AE330F2730B7ED9D18B5EEC02EF9FD4A394660E82E2219), the address to which is sent via *lParam* parameter.

SunCrypt analysis 05

Operation modes

The ransomware checks for the command line arguments that are used to enable specific operation modes. There are five modes, which are almost the same as Maze ransomware uses:

- -log — enables console output to log ransomware`s activity
- -noshares — disables encryption for network shares
- -path — indicates a directory which will be encrypted
- -noreport — does not transfer user data to the server
- -nomutex — enables running multiple copies of the ransomware.

To hide the strings in the code, SunCrypt uses the custom FNV hash function. For example, the following hashes are used for the mentioned above command-line arguments:

- '-log' is '1000C912h'
- '-noshares' is ;'3A1B96F1h'
- '-path' is '0B71AC6EDh'
- '-noreport' is 'D6A47853h'
- '-nomutex' is '1ACB53D2h'

FNV hash function

FNV (Fowler–Noll–Vo) is a non-cryptographic hash function created by Glenn Fowler, Landon Curt Noll, and Kiem-Phong Vo. It is pretty simple and has two primary operations: XOR and multiplication. The pseudocode is as follows:

algorithm fnv-1 is

hash := FNV_offset_basis do

for each byte_of_data to be hashed

```
        hash := hash × FNV_prime
        hash := hash XOR byte_of_data
    return hash
```

Going back to SunCrypt, the realization of the custom FNV algorithm is close to FNV32a, but it proceeds the strings in the reverse order, starting with '0x00' byte. As a result, there are no online databases possible to decrypt it.

SunCrypt analysis 06

Encryption

Static analysis of the ransomware is sufficiently complicated and does not give much information, as the strings and API functions are encrypted and decrypted during the runtime. The algorithms used for strings encryption are not complicated, but they vary for each string.

SunCrypt analysis 07

Before we go to file encryption, it is necessary to highlight some of the ransomware's tricks.

The first one is specifying a hashed list of blacklist extensions. After taking a file extension, the hashing function is applied and then the hash of the extension is compared with an array of the hashed extensions. If it is matched, the file starts to be encrypted. If not, the current value in the array is being compared with the string "13:19:00" to check the end of the array.

It impossible to see the actual extensions during the ransomware analysis but they can be brute-forced by calculating the hashes for all possible extensions and comparing with the hashes in the list.

Some of the file extensions that SunCrypt looks for to encrypt files are as follows:

.x3f .wpd .srw .srf .sr2 .sdf .rwl .raf .qbx .qbw .qba .py .pdd .p7c .p7b .oth .orf .odm .nrw .mef .kdc .kdbx .jpe .ibank .erf .eml .dxg .dng .der .crw .bpw .bay .backup .accdt .3fr .stm .pdb .log

.dat .bin .xlk .qbb .ptx .pfx .pef .odc .nsf .indd .fdb .dcr .cr2 .cdf .bkp .act .xlam .xla .wps .rw2 .r3d .ps .eps .dot .cdr .arw .ai .idx .html .dxf .dwg .csv .css .config .cfg .cer .aspx .accdb

.7zip .xls .rtf .prf .ppt .doc .dbx .m4a .m3u .wma .flv .mp4 .mov .avi .wmv .wav .mp3 .vdi .vmdk .vmx .wallet .upk .sav .re4 .ltx .litemod .lbf .iwi .forge .das .d3dbsp .bsa .bik .asset .apk .gpg .aes .tar.bz2

.bak .tar .tgz .7z .rar .zip .svg .png .gif .raw .jpeg .jpg .psd .bat .sh .java .rb .asp .cs .pl .js .cpp .php .ldf .mdf .odb .dbf .db .mdb .sql .asmx .config .3ds .3fr .7z .accdb .accdt .act

.ai .arw .asp .aspx .avi .backup .bak .max .mdb .mdf .mef .mov .mp3 .mp4 .nrw .nsf .txt .wallet .wav .wb2 .wmv .wpd .wps .x3f .xla .xlam .xlk .xlm .xls .xlsb .xlsm .xlsx .xlt .xltm .xltx .xlw .xml .zip

.sr2 .srf .srw .svg .sxc .tib .py .qba .qbb .qbw .qbx .r3d .raf .rar .raw .rtf .rw2 .rwl .sdf .sldm .sldx .sql .pdd .pdf .pef .pem .pfx .php .php5 .phtml .pl .png .pot .potm .potx .ppam .pps .ppsm

.ppsx .ppt .pptm .pptx .prf .ps .psd .pst .ptx .odb .odc .odm .odp .ods .odt .orf .oth .p12 .p7b .p7c .pdb .h .htm .html .ibank .idx .indd .java .jpe .jpeg .jpg .jsp .kdbx .kdc .key .doc .docm .docx

.dot .dotm .dotx .dwg .dxf .dxg .eml .eps .erf .fdb .flv .cpp .cr2 .crt .crw .cs .csv .bay .bik .bkf .bkp .bpw .c .cdf .cdr .cer .der .dng .db .dcr .dbf .dbx .sln .mdb

SunCrypt analysis 09

SunCrypt analysis 08

The encryption scheme has three layers that use file, session, and master keys.

1. Files encryption with ChaCha symmetric crypto algorithm
2. File keys and IV are encrypted with the Session RSA public key and added to the encrypted files.
3. Encryption of the Session RSA private key is done with the Master RSA public key. The encrypted Session RSA private key is then stored in the ransom note.

SunCrypt generates 32 random bytes using *SystemFunction036()* which stands for *RtlGenRandom()* from advapi32.dll, which is used for generating a random extension for a file and for creating the ChaCha20 file key.

SunCrypt analysis 10

If the file size is less than 512 bytes, it is skipped during the encryption process.

SunCrypt analysis 11

The file encryption code:

SunCrypt analysis 12

C&C Server

The user's data is sent to one of the following IP addresses that belong to the Maze ransomware cartel's subnet (91.218.114.0), located in Moscow, Russia:

- 91.218.114.30
- 91.218.114.31

Checking the IP information of the first one, which is actively used now, it belongs to Russia and was created on June 25, 2019.

SunCrypt analysis 13

Decryption service

The ransom note is titled ‘YOUR_FILES_ARE_ENCRYPTED.HTML’ and is available in five languages: English, German, French, Spanish, and Japanese. It contains a secret user message which identifies an infected system, the link to the website with the published companies’ data in the Tor network (nbzzb6sa6xuura2z.onion), and the chat with the SunCrypt operators (http://ebwexiymbsib4rmw.onion/chat.html?6a1dcf2506-24d447c336-052b4f4dd4-a66e12e526-918eb97c22-28ca2d80dd-df62bf2289-ba05daa71c).

SunCrypt analysis 14

The tab ‘News’ is created to publicly publish the private data of companies infected by SunCrypt that refused to pay a ransom for a decryptor.

SunCrypt analysis 15

The public chat connects with SunCrypt operators who work from 10 AM to 6 PM CST. They even provided a discount during the COVID pandemic.

SunCrypt analysis 16

SunCrypt analysis 17

Going through the chat, there are few disclosed BTC wallets where the operator is asked to transfer the money. At the time of writing, none of these wallets had any transactions received:

- 1HAJPBNuqJYX6WNnaZr44UHetA1boUvGG5
- 1AyUcys5UehMz6wFAG7ZpQ2xUsu6DGW9Ta
- 1CDrZGKAUkZGgiEtBn8hPZ1G7Dox9xubpg

Detection by Acronis

Acronis Cyber Protection Solutions – such as Acronis Cyber Protect or Acronis True Image 2021 – that have our integrated advanced antimalware and Acronis Active Protection technologies enabled successfully block SunCrypt and restore the encrypted files.

Acronis antimalware stops SunCrypt Ransomware 1

Acronis antimalware stops SunCrypt Ransomware 2

IoCs

- YOUR_FILES_ARE_ENCRYPTED.HTML
- ebwexiymbsib4rmw.onion nbzzb6sa6xuura2z.onion
- Mutex = {72 72 74 75 28 74 28 25 25 25 28 20 21 24 70 23 21 25 23 24 73 25 74 72 20 77 23 77 22 28 21 20 25 20 27 28 26 23 26 23 20 74 26 26 20 28 75 27 72 29 26 29 74 23 20 21 21 75 23 28 23 75 27 70 20 27 75 24 20 20 28 74 77 72 74 77 21 25 24 23 11 00 00 00 45 BB 00 00 79 65 65 61 2B 3E 3E 28 20 3F 23 20 29 3F 20 20 25 3F 22 20 2A 79 65 65 61 2B 3E 3E 28 20 3F 23 20 29 3F 20 20 25 3F 22}
- http://91.218.114.31
- http://91.218.114.30

- DLL payload SHA256: E3DEA10844AEBC7D60AE330F2730B7ED9D18B5EEC02EF9FD4A394660E82E2219
- Powerhsel loader MD5: d87fcd8d2bf450b0056a151e9a116f72 SHA1: 48cb6bdbe092e5a90c778114b2dda43ce3221c9f SHA256: 3090BFF3D16B0B150444C3BFB196229BA0AB0B6B826FA306803DE0192BEDDB80
