

RAT Tool Disguised as Solution File (*.sln) Being Distributed on Github

: 9/1/2022



The ASEC analysis team has recently discovered the distribution of a RAT Tool disguised as a solution file (*.sln) on GitHub. As shown in Figure 1, the malware distributor is sharing a source code on GitHub titled “Jpg Png Exploit Downloader Fud Cryter Malware Builder Cve 2022”. The file composition looks normal, but the solution file (*.sln) is actually a RAT tool. It is through methods like this that the malware distributor lures users to run the RAT tool by disguising it as a solution file (*.sln). Generally, programmers who receive the code that includes the solution file run the file in order to open the project. Users should take caution against social engineering techniques that take advantage of such a thought process.

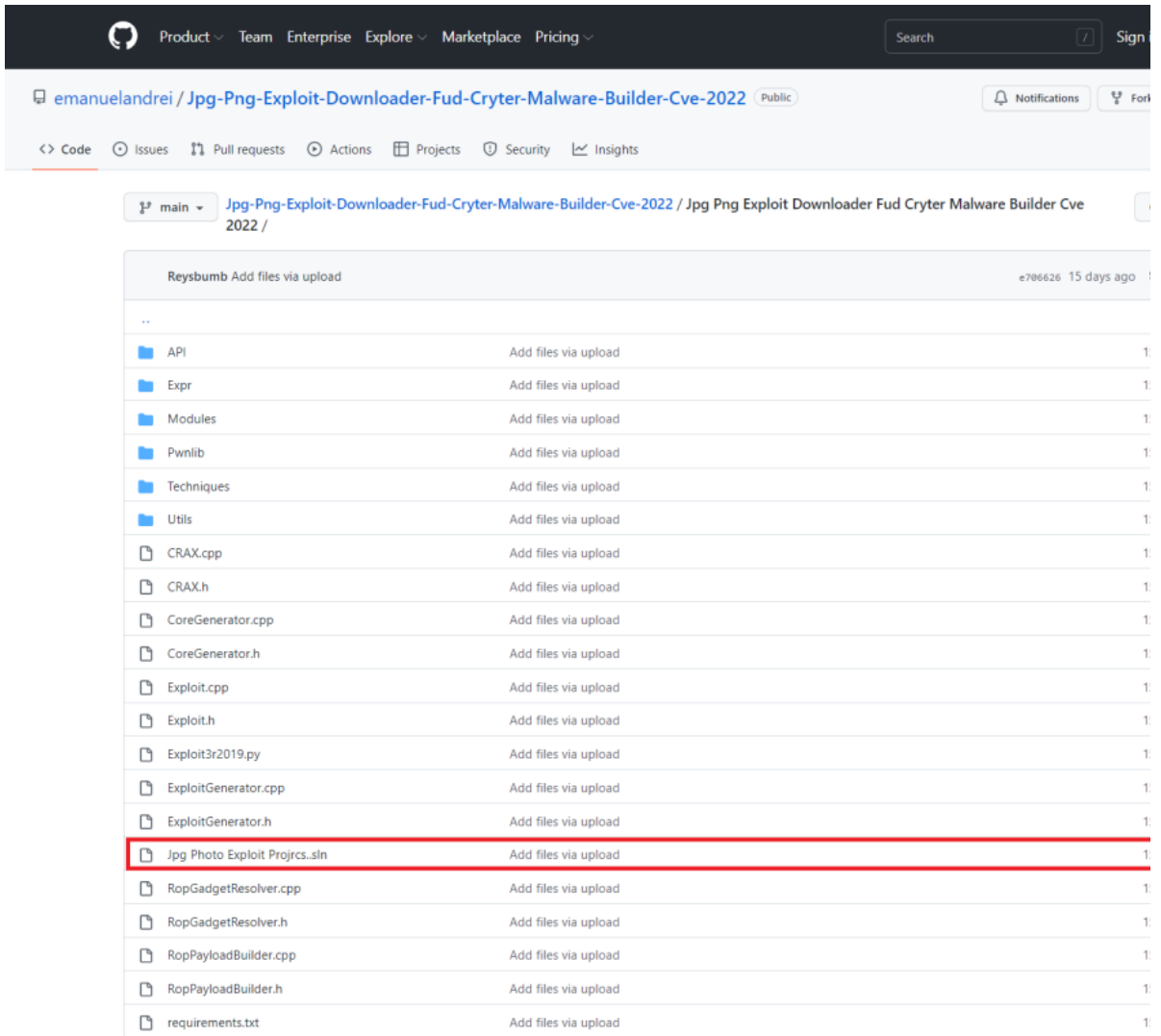


Figure 1. Disguised file shared on GitHub

If you download the files shown above, you will receive the following files as shown in Figure 2. In Figure 2, the feature 'Hide extensions for known file types' was disabled. Users should be cautious of a file with a solution file (*.sln) icon as it also has a name similar to the solution file. This malware was created to prompt users to run it, but you can tell that it is actually a screen saver if you look at the malware type. In a Windows environment, .scr is an extension that can be run. Therefore, running the file will infect your system with malware.

이름	수정된 날짜	유형	크기
API	2022-08-03 오전 4:51	파일 폴더	
Expr	2022-08-03 오전 4:51	파일 폴더	
Modules	2022-08-03 오전 4:51	파일 폴더	
Pwnlib	2022-08-03 오전 4:51	파일 폴더	
Techniques	2022-08-03 오전 4:51	파일 폴더	
Utils	2022-08-03 오전 4:51	파일 폴더	
CoreGenerator.cpp	2022-08-03 오전 4:51	C++ Source	3KB
CoreGenerator.h	2022-08-03 오전 4:51	C/C++ Header	2KB
CRAX.cpp	2022-08-03 오전 4:51	C++ Source	15KB
CRAX.h	2022-08-03 오전 4:51	C/C++ Header	13KB
Exploit.cpp	2022-08-03 오전 4:51	C++ Source	5KB
Exploit.h	2022-08-03 오전 4:51	C/C++ Header	6KB
Exploit3r2019.py	2022-08-03 오전 4:51	Python File	14KB
ExploitGenerator.cpp	2022-08-03 오전 4:51	C++ Source	7KB
ExploitGenerator.h	2022-08-03 오전 4:51	C/C++ Header	3KB
Jpg Photo Exploit Projcs..sln	2022-08-03 오전 4:51	화면 보호기	682KB
requirements.txt	2022-08-03 오전 4:51	텍스트 문서	1KB
RopGadgetResolver.cpp	2022-08-03 오전 4:51	C++ Source	4KB
RopGadgetResolver.h	2022-08-03 오전 4:51	C/C++ Header	4KB
RopPayloadBuilder.cpp	2022-08-03 오전 4:51	C++ Source	10KB
RopPayloadBuilder.h	2022-08-03 오전 4:51	C/C++ Header	5KB

Figure 2. List of downloaded files

```

19         Settings.Key = Encoding.UTF8.GetString(Convert.FromBase64String(Settings.Key));
20         Settings.aes256 = new Aes256(Settings.Key);
21         Settings.Por_ts = Settings.aes256.Decrypt(Settings.Por_ts);
22         Settings.Hos_ts = Settings.aes256.Decrypt(Settings.Hos_ts);
23         Settings.Ver_sion = Settings.aes256.Decrypt(Settings.Ver_sion);
24         Settings.In_stall = Settings.aes256.Decrypt(Settings.In_stall);
25         Settings.MTX = Settings.aes256.Decrypt(Settings.MTX);
26         Settings.Paste_bin = Settings.aes256.Decrypt(Settings.Paste_bin);
27         Settings.An_tti = Settings.aes256.Decrypt(Settings.An_tti);
28         Settings.Anti_Process = Settings.aes256.Decrypt(Settings.Anti_Process);
29         Settings.BS_OD = Settings.aes256.Decrypt(Settings.BS_OD);
30         Settings.Group = Settings.aes256.Decrypt(Settings.Group);
31         Settings.Hw_Id = HwidGen.HWID();
32         Settings.Server_signature = Settings.aes256.Decrypt(Settings.Server_signature);
33         Settings.Server_certificate = new X509Certificate2(Convert.FromBase64String(Settings.aes256.Decrypt(Settings.Server_certificate)));
34         result = Settings.VerifyHash();
35
36     catch
37     {
38         result = false;
39     }

```

Name	Value	Type
Client.Algorithm.Aes256.Decrypt returned	"217.64.31.3"	string
result	false	bool

Figure 3. AsyncRAT C2 Decryption

The malware disguised as a solution file used a cryptor to change its appearance and avoid detection. Once executed, it is injected into a normal Windows program such as AppLaunch.exe, RegAsm.exe, and InstallUtil.exe, ultimately running a RAT tool.

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 50 4B 03 04 14 00 00 08 08 00 B3 1E D1 54 AE 87 PK.....'.Ñt@+
00000010 C5 66 B4 DF 06 00 00 7E 0C 00 1F 00 00 00 56 65 Åf'8...~.....Ve
00000020 6E 6F 6D 20 43 6F 6E 74 72 6F 6C 20 43 6C 69 65 nom Control Clie
00000030 6E 74 E2 80 AE 6E 6C 73 2E 2E 73 63 72 EC 5C 79 nt@e@nls..scri\y
00000040 78 54 D5 15 7F B3 64 32 09 13 66 80 04 02 24 18 xT0...'d2..fe...$.
00000050 35 5A 6C D0 C6 0E 54 C2 10 1C 24 93 04 25 30 21 5ZlDE.TA..$.!0!
00000060 30 21 02 01 2B D0 69 8A 35 86 37 40 2B 6B 5F 42 0!...+DiS5t7@+k_B

```

Figure 4. Data of malware compressed as a ZIP file

As for how the extension appears as a solution file (*.sln) in GitHub and Windows Explorer, compressing the file gives us the answer: the file uses the 'RIGHT-TO-LEFT OVERRIDE' unicode string (see Figure 4).

Similar cases have been occurring with more frequency on GitHub, which has recently been getting a lot of traffic. Malicious malware distributors are disguising their malware as solution files (*.sln) and making them seem like source codes. Users should therefore be cautious when viewing files from unreliable sources. Also, they must keep their anti-malware software updated to the latest version.

AhnLab V3 detects and blocks the malware strains using the aliases below.

[File Detection]

- Trojan/Win.Leonem.C5218555 (2022.08.04.00)
- Trojan/Win.Agent.C4526491 (2021.06.30.03)
- HackTool/Win32.Vbinder.R12127 (2015.02.14.01)
- Trojan/Win.SmokeLoader.R510280 (2022.08.12.04)
- Trojan/Win.MSILZilla.C5129545 (2022.05.15.02)
- Trojan/Win.Generic.C5198415 (2022.07.08.03)

[Behavior Detection]

- Malware/MDP.Inject.M3037
- Execution/MDP.Powershell.M3991
- Malware/MDP.AutoRun.M1037
- Execution/MDP.SystemManipulation.M1788
- Malware/MDP.Inject.M1252

[IOC Info]

- <https://github.com/emanuelandrei/Jpg-Png-Exploit-Downloader-Fud-Cryter-Malware-Builder-Cve-2022-0cfa5f7c008e3dc2df275a99aef9cbbb> // Jpg Photo Exploit Projnls..scr
- [b1f02c7efc154019e9f1974939e204b9](https://github.com/VortexRadiation/VenomControl-Rat-Crack-Source)
- <https://github.com/VortexRadiation/VenomControl-Rat-Crack-Source>
- [98d7999986d63fbd914bddc3d7b7ecf9](https://github.com/VortexRadiation/VenomControl-Rat-Crack-Source) // Venom Control Clientnls.
- [8b662719e44ab11419fe3e1d7e96cc03](https://github.com/VortexRadiation/VenomControl-Rat-Crack-Source)
- [https://github.com/VortexRadiation/Jpg-Png-Exploit-Downloader-Fud-Cryter-Malware-Builder-Cve-2022-0cfa5f7c008e3dc2df275a99aef9cbbb](https://github.com/emanuelandrei/Jpg-Png-Exploit-Downloader-Fud-Cryter-Malware-Builder-Cve-2022-0cfa5f7c008e3dc2df275a99aef9cbbb) // Jpg Photo Exploit Projnls.
- <https://github.com/Lessermask/Discord-Image-Token-Password-Grabber-Exploit-Cve-2022-9a01d2f0aad78bcc4a4ca07552154ee1>
- <https://github.com/Lessermask/Discord-Image-Token-Password-Grabber-Exploit-Cve-2022-9a01d2f0aad78bcc4a4ca07552154ee1>

- 9fd996ce42d667ba01c902124bf95f6d // Discord Image Token Grabber.

Subscribe to AhnLab's next-generation threat intelligence platform 'AhnLab TIP' to check related IOC and detailed analysis information.