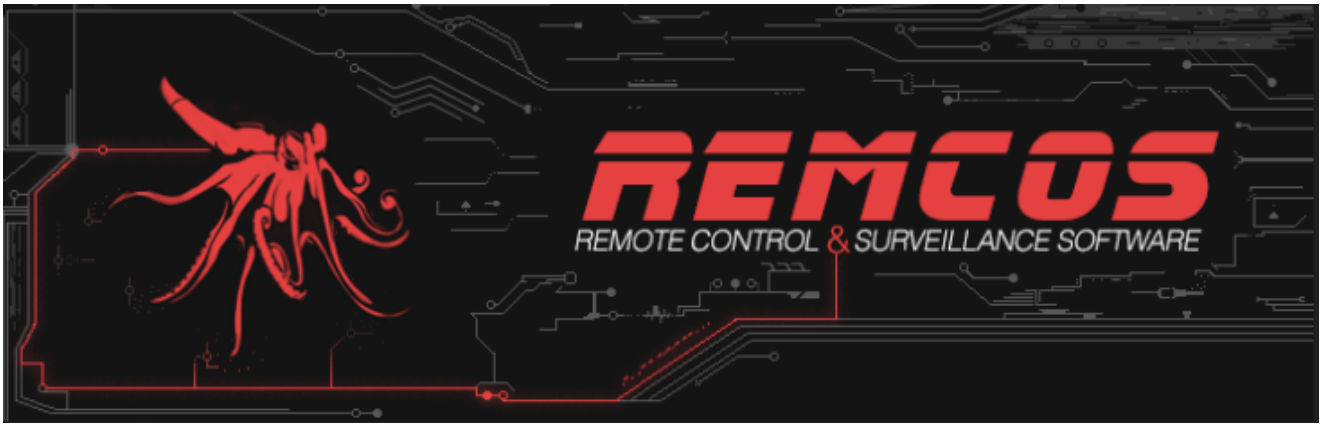
Malware Analysis — Remcos RAT

medium.com/@b.magnezi/malware-analysis-ramcos-rat-48fd986328f5

0xMrMagnezi February 20, 2024



0xMrMagnezi

--

Ramcos RAT is a sophisticated type of malware called a remote access trojan (RAT). It evades antivirus detection and gives cybercriminals remote access and control over infected systems. Typically , it's used for stealing information, installing more malware, or using the infected system in a botnet.

MalwareBazaar sample

Stage 1:

As usual I downloaded the file and extracted it using the password "infected".

Original CMD file

After extracting the file and examining its contents , I noticed it consisted of two large chunks of code , along with sets and loops. The code was lengthy and heavily obfuscated. To better understand its behavior , I ran it and monitored for any new processes launched by the original file.

PowerShell was being executed under the original file

Stage 2:

Obfuscated PowerShell

This PowerShell code was much easier to deobfuscate , it used simple replacements to create a list of words and then used those words in the hidden PowerShell code.

Deobfuscated PS — Highlighting the AES Decryption and Decompress

After renaming the variables and deobfuscation it was clear to me why the original file was impossible to understand and deobfuscate — **it was encrypted and compressed.** Using the above code I had all the things I needed to decrypt the original file ; using the AES key and IV.

CyberChef — Extracting EXE from the original file

At first, I converted from Base64 to Hex and then decrypted using AES with the Key and IV found in the previous part. Finally, I decompressed the output using Gunzip. Essentially, I followed the decoding steps as intended. I knew I was on the right path as soon as I saw the MZ Header, which is the file format of an EXE file. There were actually two chunks of code , indicating two files inside, as shown in the next picture.

Using DIE, I determined the language in which those files were written (.NET)

I chose to debug those files in DNSPY because they were written in .NET. Once I opened them, it was clear what the program was trying to do. The attacker hadn't obfuscated this final sample.
As shown in the next picture, I was able to see exactly the names of the functions and what they were supposed to do.

Disable Defender Function

Persistence Function

Finding The file that is being used as persistence

Important to note that this file that is being saved in the startup path is the original cmd file that was analyzed.
Moreover , while analyzing this sample , I monitored network connections and discovered additional IOCs.

IOCs:

- lods.cmd — 194118c43c65faad06bf5ff6cd9b52a2
- IxsqpAscrubb.exe — 3ca5a8e1e0217d89b4926ca68e5f41c8
- MAEmka.tmp(exe) — e60e82df05c02ec173655dd9c41dd829
- Domain — api[.]ipify[.]org
- Domain — ads[.]hostloads[.]xyz

In conclusion , the analysis of Ramcos RAT highlights the sophisticated techniques used by cybercriminals to evade detection and gain remote access to infected systems. The malware's multi-stage approach , from obfuscated CMD and PowerShell scripts to encrypted and compressed payloads , showcases the complexity of modern malware threats.