Date: March 13, 2023
Reference: C-X-1542

Iranian Government-Sponsored Threat Actor MuddyWater Conducts Cyber Attack Against Israel

Executive Summary

MuddyWater is an Iranian government-sponsored threat actor, part of Iran's Ministry of Intelligence and Security (MOIS). Since Q4 of 2022 the group has been active against Israeli organizations.
The malicious activities were detected as a result of in-depth knowledge of the threat actor TTPs and infrastructure. The threat actor targeted various sectors including finance, academia and government.
On February 2022, the threat actor conducted a ransomware attack against an organization in the academic sector. Indications show that ransom was not the main purpose for this attack, but disseminating disinformation with Anti-Israeli content.
The group is exploiting the Log4j vulnerability, and deploys RAT tools, including lately the SyncroRAT tool.

Background

1. MuddyWater is also known as Static Kitten, Mercury, Temp.Zagros, Seedworm and Earth Vetala, and has been active since 2017.
2. The threat actor conducts cyber espionage operations using social engineering in its phishing campaigns, exploiting 1-day vulnerabilities and using malicious tools such as PowerShower, PowerStallion and MuddyWater proxy.
3. Over the last 2 years, the threat actor has been connected to several attacks, against government organizations and other sectors.

Cyber Attack Analysis

1. On February 2023, for the first time, the threat actor combined a destructive operation with an influence campaign against an Israeli target.
2. A few days before the attack the threat actor launched a Telegram channel under the name "Darkbit", in order to publish leaked data from that attack.

3. MuddyWater chose to conceal its identity behind the name "Darkbit" in order to avoid being identified as a state-sponsored threat actor. This concealment involved several aspects:
    a. Publishing Anti-Israeli narrative, including criticism on the Israeli government regarding the Israeli-Palestinian conflict. This type of publications are usually expressed by hacktivists, rather than government-sponsored groups.
    b. The ransomware file extension was "darkbit".
    c. Ransom demand was significantly higher than the average ransomware attack in Israel, demanding 80 Bitcoin (more than $1.6 Million).
4. The threat actor conducted the attack using 2 methods, CNA and CNE:
    a. CNE – the attackers published they stole 4TB of data from the victim organization.
    b. CNA – the attackers encrypted servers and end points in order to disrupt business-critical systems. After the encryption phase, a text file named "RECOVERY_DARKBIT" was left as a message to the attacked organization.

Analysis of malware used by the threat actor

1. First variant, used to encrypt Windows systems
    1. Filename is 8thcurse.exe, size 5.1MB, compiled by the open source tool MinGW. The malware is written in the Go Programming language.
    2. The malware is 64-bit version, and was compiled at 12/2/2023 00:10:5.
    3. A copy was uploaded to virustotal.com at 13/2/2023. Identified by 50 AV engines as malicious (as of 9/3/23).
    4. MD5: 9880fae6551d1e9ee921f39751a6f3c0
    5. SHA1: 30466ccd4ec7bcafb370510855da2cd631f74b7a
    6. SHA256:9107be160f7b639d68fe3670de58ed254d81de6aec9a41ad58d91aa814a247ff
    7. The malware uses AES symmetrical encryption to encrypt files located on the filesystem. The key used for the encryption is kept in a file encrypted by Asymmetrical RSA encryption. Decryption is only possible using the private key associated with the public key used for encryption.
    8. Files that have been encrypted, get their extension changed to darkbit.

9. The string DARKBIT_ENCRYPTED_FILES is added to the file content, to indicate to the malware that the file was encrypted and prevent a second encryption.
10. Information about the encryption process (key and IV used for the AES encryption), is encrypted using an RSA public key. The corresponding private key is kept by the attacker and can be used to decrypt the file.
11. By default, the malware waits 10 seconds before starting to encrypt files. At that point, the malware is stoppable.
12. The malware has several running options, including:
    - -domain string : domain
    - -force : force blacklisted computers
    - -list string: list
    - -nomutex : force not checking mutex
    - -noransom : spread with no encryption
    - -passwordf string: password
    - -path string : path
    - -t int : thread (default -1)
    - -username string: username
13. By default, the malware defines a mutex (Global\\dbdbdbdb) upon starting, to prevent multiple executions.
14. The malware supports using multiple threads for parallel execution.
15. The malware uses built-in commands to prevent restoring files after the encryption phase. It erases Shadow Copies using the built-in command vssadmin, running to following command:
    - vssadmin.exe delete shadow /all /Quiet
16. Indications show that the malware was specifically built to attack the Technion:
    - The ransom note was addressed to the Technion.
    - The malware included a hard-coded list of servers on the Technion network.
17. Based on the above list, it is probable that the attacker mapped the Technion's network prior to the attack.
18. The malware contains a list of file extensions that will not be encrypted, to prevent operating system failure. These files are:
    - msilog,log,ldf,lock,theme,msi,sys,wpx,cpl,adv,msc,scr,key,ico,dll, hta,deskthemepack,nomedia,msu,rtp,msp,idx,ani,386,diagcfg,bi

n,mod,ics,com,hlp,spl,nls,cab,diagpkg,icl,ocx,rom,prf,themepack, msstyles,icns,mpa,drv,cur,diagcab,exe,cmd,shs,Darkbit

19. The malware uses configuration definitions for the encryption process. In this version, large files are not encrypted as a whole, but broken down to multiple parts that are encrypted separately.
20. Extension that is labeled "1" in the configuration, is not encrypted.
21. The malware code can be modified modularly and simply, to fit the attacker's purpose.

2. Second variant, used to attack ESXi servers
    1. MD5:ad2c3054f9de589030269d12a9cbbeeb
    2. SHA1:9ed8db1620dac9efc78ebb4d209d3281c50e24da
    3. SHA256:0bb1d29ede51d86373e31485d0e24701558e50856722357372518edfb98265a1
    4. This malware is a dedicated payload of ransomware used by the threat actor DarkBit.
    5. It is designed to work against unix/unix-like systems (such as Linux) and the Binary is ELF 64-bit.
    6. It appears to target VMware ESXi servers.
    7. File Size: 1.52MB (1595968 bytes)
    8. Date of change: Friday, 30 November 1970. The date was manipulated by the malware authors to hide the actual date is was developed.
    9. The malware contains in the code the ransom note, as can be seen in the attached figure:

10. The malware is written in C/C++, compiled by GCC (4.4.7 20120313).

11. It uses the same method to encrypt the files as the Windows version. Files are encrypted using AES symmetrical key, and the key used is encrypted using RSA public key.
12. Usually, running the malware requires passing command line arguments, in the form:
    - <Payload Path> DarkBit /vmfs time excludeVm
13. The time parameter is used to define sleep time, before the malware starts.
14. The ExcludeVm is used to exclude Virtual Machines from getting encrypted.
15. The malware can be run without supplying all the arguments.
16. The code contains a hard-coded list of file extensions to encrypt, including:
    - Vmdk
    - Vswp
    - Vmsd- (Vmware snapshot metadata)
    - Vmsn - (VMware virtual machine snapshot)
17. The malware uses the ESXCLI to map the virtual machines on the host (VM LIST).
18. After getting the list, the malware runs a command designed to shutdown all VMs that were shown in the list:
    - Esxcli –formatter=csv –format-param=fields=="WorldID,DisplayName" vm process list | sed -n '1!p' | grep -vi",/vmfs," | cut-d ',' -f1| awk '{system("esxcli vm process kill -t=force -w="$1)}'
19. The threat actor uses other tools, such as:
    - CLI tool used for OS automation:

      tacoscript.exe,
      f33c16530844f1c4bfe9eb2878379d135dfdf56f59a2e2b2362f069e6583c3a6
    - A remote management tool using SSH and other protocols:

      rport.exe
      b9cf785b81778e2b805752c7b839737416e3af54f64f1e40e008142e382df0c4