Revealing the Abyss Ransomware | Idan Malihi

Archived: 2026-04-05 17:11:14 UTC

During Abyss ransomware execution, a new log file may be created, and logs may be written to it.

The ransomware will destroy the contents of files and then change their extension to .XPbS1.

When the ransomware is executed, it attempts to infect any external drives present in the system and creates an autorun file.

The ransomware will create two files in the system — a JPG file and a TXT file.

The Abyss ransomware will attempt to perform lateral movement techniques within the local network by searching for SMB shares, external drives, and other accessible resources.

When the ransomware creates the ‘work.log’ file, it adds the file to an exclusion list.
The ransomware loads the effective address of the ‘work.log’ string into the rax register. Then, it moves the rax register content to the ‘cs:qword_140033CA0’ memory location, which indicates an exclusions list.

The ‘cs:qword_140033CA0’ content:

Additionally, the ransomware loads the effective address of the ‘.XPbS1’ string into the rax register. Then, it moves the rax register content to the ‘cs:qword_140033A60’ memory location, which indicates an exclusion list with files’ extensions.

The ‘cs:qword_140033A60’ content:

The ransomware uses the GetTickCount function to obtain the number of milliseconds since the system started. This technique is used to avoid detection in sandboxes or virtual machines. Additionally, the ransomware establishes a connection with the Windows Service Control Manager using the OpenSCManagerA function. It then goes through a list of service names, which are pointed to off_140009990, and tries to open each service using the OpenServiceA function.

The ransomware uses the CreateToolhelp32Snapshot function to create a snapshot of current processes running on the host system. It then compares the list of running processes with an executable names list, which is pointed to an item named ‘off_140009E90’, using the Process32FirstW and Process32NextW functions. If a match is found, the malware uses the OpenProcess function to open and handle the process and the TerminateProcess function to terminate the process.

The processes list (off_140009E90):

Then, it sets up a semaphore (hSemaphore) and a handle (hHandle) for thread synchronization. Before the synchronization, the ransomware uses the GetSystemInfo function to retrieve information about the system and

the number of processors. Based on the number of processors, the ransomware adjusts the number of threads (nCount) and initializes several handles for synchronization.

The subroutine ‘sub_14001A4F0’ uses semaphore thread synchronization and calls the CreateSemaphoreA function.

The ransomware creates multiple threads using the CreateThread API to execute the ‘sub_14001C870’ subroutine concurrently. These threads appear to be assigned tasks related to network shares and paths.

The ransomware performs network share searches in the ‘sub_14001CB80’ subroutine.

In the ‘sub_14001CB80’ subroutine code, the ransomware uses the NetShareEnum function to search through the network shares and disk devices present in the host. The ransomware disregards any hidden administrative shares (ADMIN$) and records the paths of the detected network shares in the ‘sub_14001A240’ subroutine.

The ransomware uses several functions to enumerate and determine the type of drives in the system.

In the ‘sub_14001A740’ subroutine, the ransomware iterates through the predefined drive letters and checks the drive type using the GetDriveTypeW function. It enumerates fixed and removable drives on the system. For each drive with drive type 1 (DRIVE_FIXED), it attempts to assign a corresponding drive letter to the volume using the SetVolumeMountPointW function.

Also, in the ‘sub_14001CD20’ subroutine, the ransomware constructs the drive path in the format \\?\X: where ‘X’ is the drive letter. It uses GetDriveTypeW to determine the type of the drive, whether it’s removable, fixed, or network.
If the drive type is 1 (root path), 2 (removable drive), or 3 (fixed drive), it logs information and processes the drive path further.

After the ransomware enumerates network shares and logical drives, it starts the encryption operation. The ransomware uses the FindFirstFileW and FindNextFileW functions to go through every file in every directory and sub-directory.

The CreateFileW function is used in a loop to create the WhatHappened.txt file in the file system and write the ransom note content using the WriteFile function.

The threat actors state that they can restore the files on the file system. They claim their motive is purely financial and open to negotiation.

The threat actors offer two options to the victim. The first option is to seek help from authorities, but the threat actors threaten to cause the company to face fines, legal actions, and reputational damage if they try to help with the decryption. The second option is to negotiate with the threat actors, pay the ransom, and receive the decryption. Importantly, the victim’s privacy will be maintained, and no one will know about the incident.

The attackers instruct the company to access a specific URL using the TOR browser to initiate negotiations.

The BMP content is the ransom note that the ransom spread in the file system earlier.

The ransomware opened the ‘HKEY_CURRENT_USER\Control Panel\Desktop’ registry path using the RegOpenKeyExW function. It edited the ‘WallpaperStyle’ and ‘TileWallpaper’ entries to 0 using the RegSetValueExW API.

As a result, the Desktop wallpaper is changed to the ransom note.

Yara Rule

Detection
