Technical Analysis of Emerging, Sophisticated Pandora Ransomware Group

cloudsek.com/technical-analysis-of-emerging-sophisticated-pandora-ransomware-group/

Anandeshwar Unnikrishnan May 12, 2022

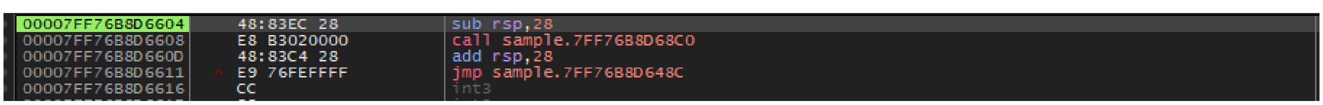
2021 saw an outbreak of ransomware groups and attacks that affected every major industry across the globe. This trend is expected to continue and even surpass the previous year's numbers by a significant margin in 2022.

In March 2022, researchers detected a new ransomware strain known as Pandora which leverages double extortion tactics to exfiltrate and encrypt large quantities of personal data. The operators offer the decryption key once the victim pays the ransom demanded. Pandora ransomware is a relatively new operation and hence its infection techniques are unknown.

However, after infiltrating the target system, the ransomware appends the ".pandora" file extension to the encrypted files and leaves a ransom note "Restore_My_Files.txt" with instructions on how to recover the data. Researchers believe that the Pandora ransomware is a rebranded version of Rook ransomware, which in turn is a spawn of the leaked Babuk code. This article explores the technical analysis of the Pandora ransomware, its evasion tactics, the process of encryption, and more in detail.

Technical Analysis of Pandora

The analysis of Pandora's binary file sample, `5b56c5d86347e164c6e571c86dbf5b1535eae6b979fede6ed66b01e79ea33b7b`, indicates that it is a UPX (Ultimate Packer for eXecutables) packed binary file. UPX is an executable file compressor used by threat actors to add a layer of obfuscation (creation of code that is difficult for humans to understand) to their malware. The ransomware code runs from the original entry point after getting unpacked in the memory.



Ransomware code running from the entry point

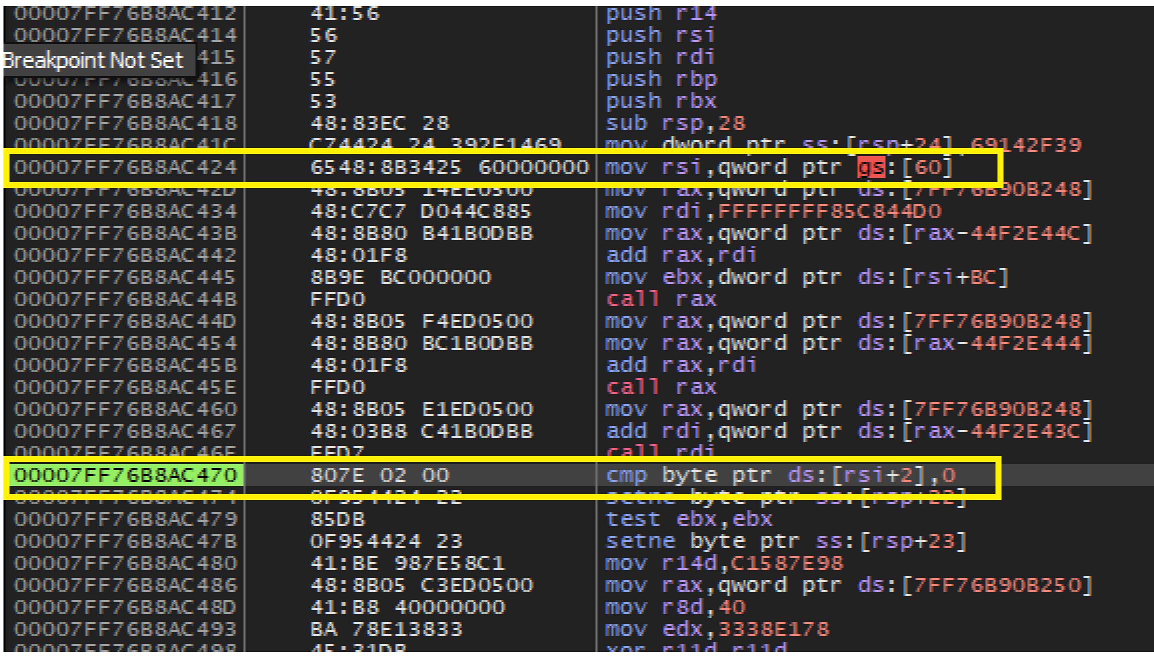
The ransomware uses obfuscated strings and deobfuscates library names and internal functions at runtime. The library modules used by Pandora are dynamically loaded on a per-use basis via the following APIs:

- **LoadlibraryA**
- **GetProcAddress**
- **GetModuleHandleA**

Initially, the ransomware creates a mutex (mutual exclusion object, which enables multiple program threads to take turns sharing the same resource) to make sure only one instance of the malware is running on the system. The mutex string, “ThisIsMutexa”, gets deobfuscated in the memory. It checks for any existing mutex on the system via **OpenMutexA**, if not present the malware creates a new one with the value “ThisIsMutexa” via **CreateMutexA.**

Anti-debug Mechanism

The malware implements anti-debug checks to hinder analysis.



Anti Debug Check

- The code highlighted in the image above reads data at the offset 0x60 from segment register **GS**. Windows stores the **Thread Information Block (TIB)** in **FS** [x86] and **GS** [x64] segment registers.
- The TIB holds the **Process Environment Block (PEB)** at the offset 0x60. The malware accesses PEB of the process via the GS register.
- Later the malware reads the data at the offset 0x2 in PEB (ds:[rsi+2]), which is the **BeingDebugged** member in the PEB structure, and then compares the obtained value with 0. If the process is being debugged then BeingDebugged will have a non zero value. If the test fails, the malware goes into an infinite loop and does not proceed further.

Evasion Techniques

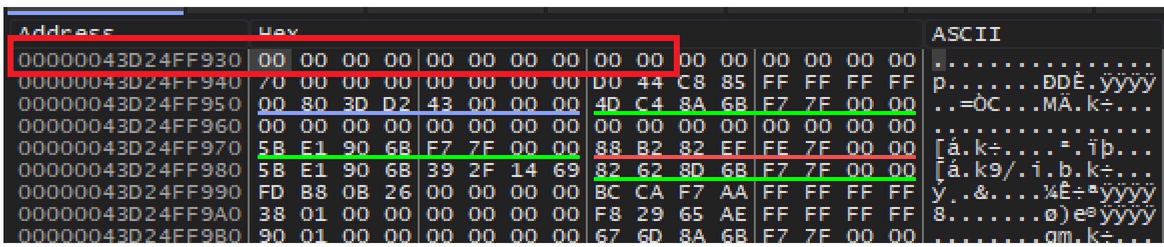
Instrumentation Callback Bypass

The security endpoints (especially ETWTi) of a device use the instrumentation callback process to check for behavioral anomalies and detect novel malware on the system. Pandora ransomware bypasses such a callback mechanism via `ntsetinformationprocess`, which changes the process information.

- ntsetinformationprocess is invoked with `ProcessInstrumentationCallback` as a part of **ProcessInformationClass**.

ntsetinfromationprocess being invoked

- The third argument in the above image is a 10-byte long structure associated with the provided ProcessInstrumentationCallback information class.



The third argument (10-byte long structure)

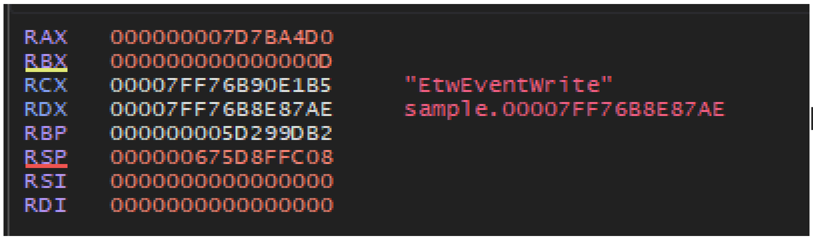
- The members and associated values in the structure are as follows:
  - Version=0 (0 for x64, 1 for x86)
  - Reserved=0
  - Callback=0

If the process created for the malware is hooked by security services via callback member, invoking the ntsetinformationprocess in a way mentioned above with callback set to 0, it helps the malware bypass such hooks.

Event Tracing Bypass

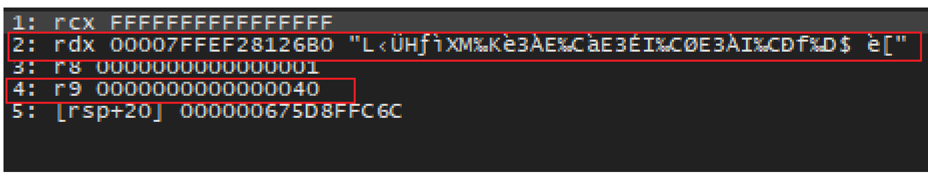
Event Tracing for Windows (ETW) is a powerful tracing facility built into the operating system, to monitor various activities of both userland and kernel land applications running on the system. This feature has become a vital instrument to endpoint security solutions to detect anomalous behavior in running programs. As a result, malware developers have started integrating functionalities in their malware to neutralize the tracing capability. One such vector is patching ETW related functions defined in ntdll.dll in the memory.

- The ransomware dynamically loads **ntdll.dll** into the memory and deobfuscates the string "`EtwEventWrite`".



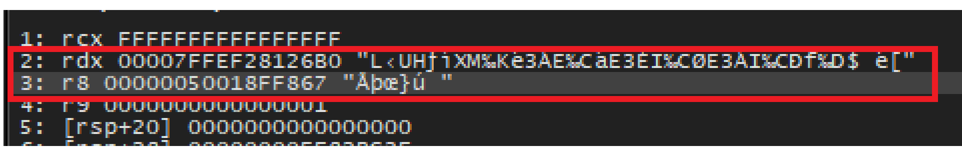
Deobfuscation of “EtwEventWrite”

- The address of the EtwEventWrite function is obtained using **GetProcAddress** API. Getting the function address is a very important step in patching, to bypass the ETW feature.
- Before the malware commences patching, the memory protections on the region of committed pages, where EtwEventWrite resides in virtual address space, need to be changed, which is done via **VirtualProtectEx** API.
- The memory region of pages where the first instruction of EtwEventWrite resides is changed to **PAGE_EXECUTE_READWRITE** to be patched.



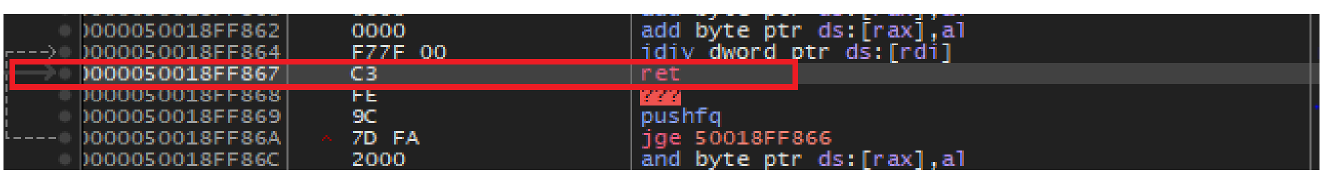
Arguments passed to VirtualProtectEx

The **WriteProcessMemory** API is used to write one byte at the beginning of the EtwEventWrite function. The second argument points to the beginning of EtwEventWrite, and the third argument is the one byte long payload that gets written at the address of EtwEventWrite.



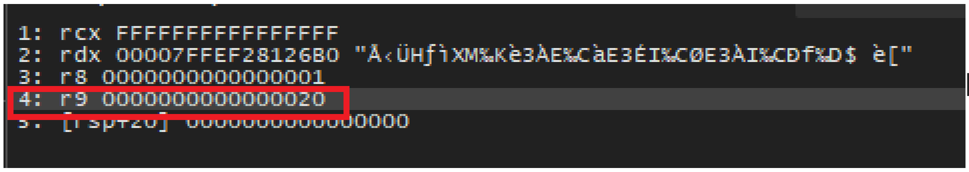
The data passed to WriteProcessMemory

The one byte payload is **0xC3**, which is the opcode for the instruction “**ret**”. This makes EtwEventWrite to simply return back to the caller function, without executing its logic to log an event when EtwEventWrite is invoked by other applications.



One byte payload – 0xC3

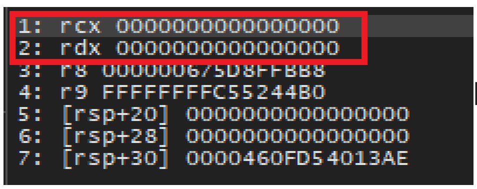
After patching, the memory protection of EtwEventWrite is reverted back to the initial permission of **PAGE_EXECUTE_READ** via VirtualProtectEx.



Memory protection of EtwEventwrite

Pre-encryption Phase

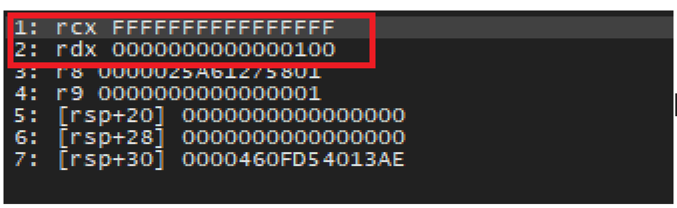
Before the encryption begins, the malicious software changes the shutdown parameters for the system via **SetProcessShutdownParameters** API. This function sets a shutdown order for the calling process relative to the other processes in the system. Here, the malware invokes the API with zero value so that the ransomware program is the last to shut down by the Operating System.



Data passed to SetProcessShutdownParameters

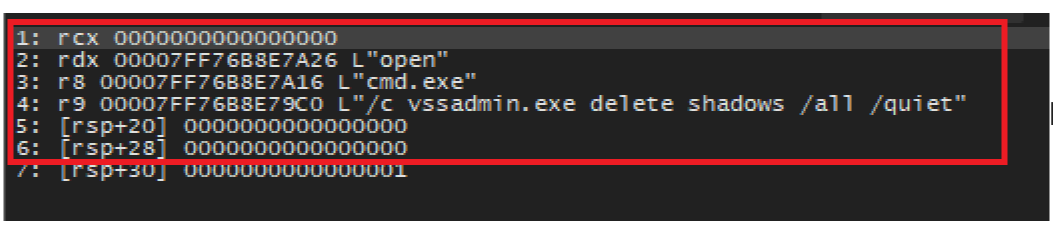
After setting these shutdown parameters, the malware empties the recycle bin via **SHEmptyRecyclebinA** API.

The ransomware raises the priority of the running process to the highest possible priority which is **REALTIME_PRIORITY_CLASS** via **SetPriorityClass** API. The second argument is the "dwPriorityClass" parameter which has a value of 0x100.



Data passed to SetPriorityClass

Finally, the volume shadow copies are deleted by executing a string of commands via **ShellExecuteA**. It uses vssadmin to perform the task of deleting the shadow files.



Deleting shadow files using vssadmin

Encryption Phase: Threading Model

The main thread of malware creates two new threads that are responsible for the encryption of user data.

Number	ID	Entry	TEB	RIP	Suspend Count
3	10892	00007FFEF27F3D60	000000675D717000	00007FFEF285FA64	1
Main	11004	00007FF76B91BC40	000000675D711000	00007FFEF285D204	1
1	7308	00007FFEF27F3D60	000000675D713000	00007FFEF285FA64	1
2	4436	00007FFEF27F3D60	000000675D715000	00007FFEF285FA64	1
6	860	00007FFEF27F3D60	000000675D71D000	00007FFEF285FA64	1
4	12200	00007FFEF12D7870	000000675D719000	00007FFEF285CC24	1
5	8072	00007FFEF27F3D60	000000675D71B000	00007FFEF285FA64	1
7	9504	00007FFEF27F3D60	000000675D71F000	00007FFEF285FA64	1
8	5052	00007FFEF1A1ACA0	000000675D721000	00007FFEEFB39A84	1
9	3160	00007FF76B8A4D60	000000675D725000	00007FFEF0DE2170	1
10	1456	00007FF76B8A4D60	000000675D723000	00007FFEF285C154	1

Creation of two new threads

The following APIs are used to create the threads:

- **CreateThread**
- **SetThreadAffinityMask**
- **ResumeThread**

The threads are created with dwCreationFlags set to **CREATE_SUSPENDED**, later the execution of threads is resumed via **ResumeThread**.

The main thread starts to enumerate the drives present on the system via the following APIs:

- **GetDriveTypeW**
- **FindFirstVolumeW**
- **GetVolumePathNamesForVolumeNameW**
- **SetVolumeMountPointW**
- **FindNextVolumeW**
- **GetLogicalDrives**

Pandora utilizes Windows I/O Completion Ports to efficiently speed up the encryption process. Following APIs are used to orchestrate the search and locking of the user data:

- **CreateIoCompletionPort**
- **PostQueuedCompletionStatus**
- **GetQueuedCompletionPort**

Initially, the main thread of the malware creates an input/ output (I/O) completion port via CreateIoCompletionPort API.

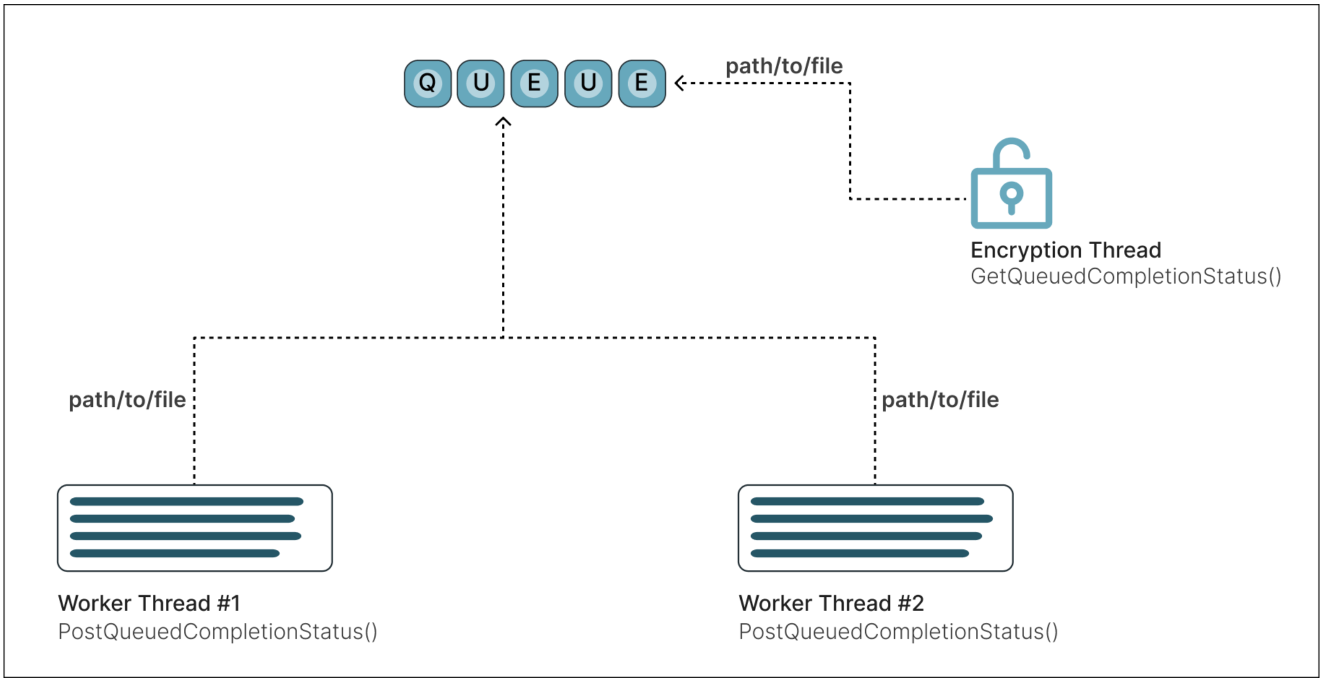
```
1: rcx FFFFFFFFFFFFFFFF
2: rdx 0000000000000000
3: r8 0000000000000000
4: r9 0000000000000002
5: [rsp+28] 0000000000000004
6: [rsp+30] 000000675D8FFD80 "è+"
7: [rsp+38] 0000460FD54013AE
```

Data passed to CreateIoCompletionPort

- The fourth argument is "NumberOfConcurrentThreads". In our case, two threads are allowed to concurrently process I/O completion packets for the I/O completion port.

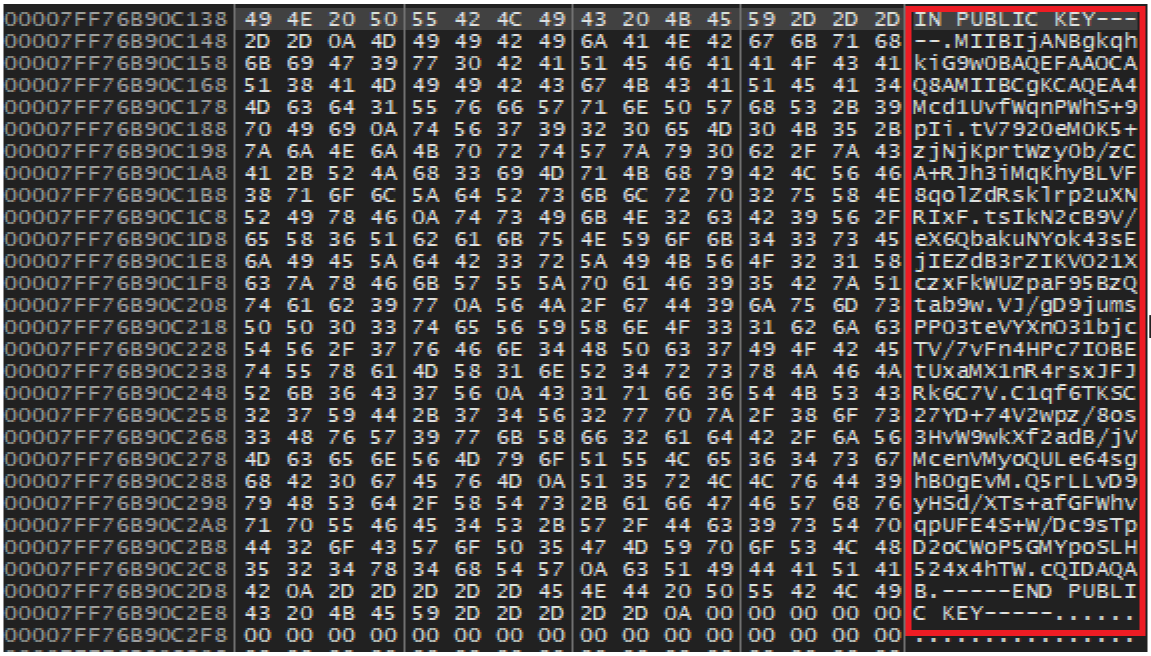
- After the creation of the I/O port, a queue is created internally, to which threads can push the completion status.
- The two threads created previously will be accessing I/O ports to perform file enumeration and encryption on the infected system.

In general, ransomware in the wild has adopted a model to optimize the encryption process. The goal here is to efficiently utilize the power of multicore processors to concurrently perform file enumeration and encryption. A group of worker threads would fetch the file paths and post them in the queue via **PostQueuedCompletionStatus**, and another thread can retrieve the posted files (paths) for encryption via **GetQueuedCompletionStatus**.



Optimization of the encryption process

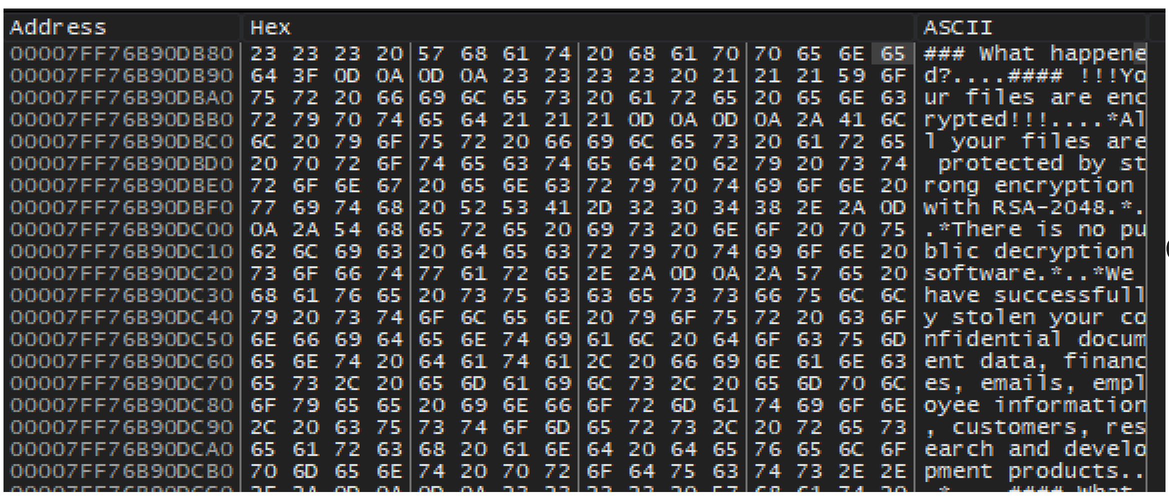
Pandora uses the RSA 4096 algorithm for encryption, the public key is embedded within the malware.



Public key embedded in the malware

As a prior step to the encryption process, the malware accesses directories in the network drives and dumps the ransom note (**Restore_My_Files.txt**). The ransom note is created using the following three APIs:

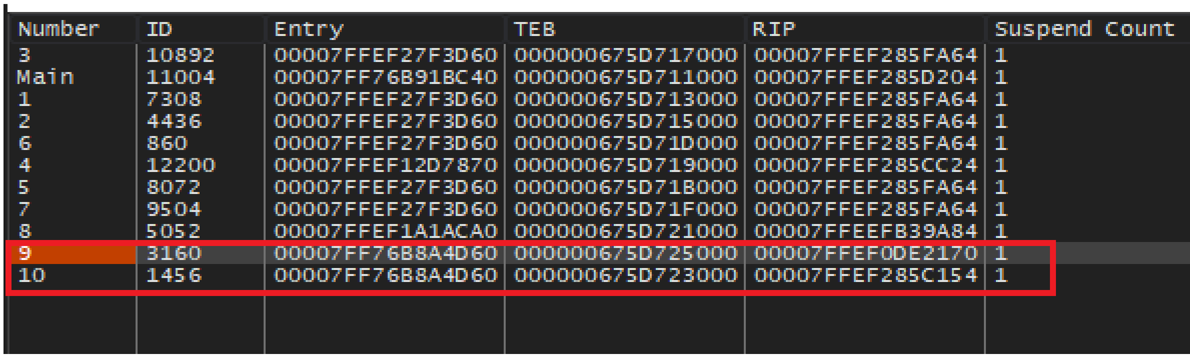
- **CreateFileW**
- **WriteFileW**
- **CloseHandle**



Contents of the ransom note

Encryption Process

The process explained in this section is executed by worker threads highlighted in the image below. These threads can concurrently enumerate and encrypt data via the Windows I/O completion port.



Number	ID	Entry	TEB	RIP	Suspend Count
3	10892	00007FFEF27F3D60	000000675D717000	00007FFEF285FA64	1
Main	11004	00007FF76B91BC40	000000675D711000	00007FFEF285D204	1
1	7308	00007FFEF27F3D60	000000675D713000	00007FFEF285FA64	1
2	4436	00007FFEF27F3D60	000000675D715000	00007FFEF285FA64	1
6	860	00007FFEF27F3D60	000000675D71D000	00007FFEF285FA64	1
4	12200	00007FFEF12D7870	000000675D719000	00007FFEF285CC24	1
5	8072	00007FFEF27F3D60	000000675D71B000	00007FFEF285FA64	1
7	9504	00007FFEF27F3D60	000000675D71F000	00007FFEF285FA64	1
8	5052	00007FFEF1A1ACA0	000000675D721000	00007FFEEFB39A84	1
9	3160	00007FF76B8A4D60	000000675D725000	00007FFEF0DE2170	1
10	1456	00007FF76B8A4D60	000000675D723000	00007FFEF285C154	1

Worker Threads

- After dumping the ransom note, the malware uses `FindFirstFileW` to open a handle to the files on the disk.
- The retrieved handle is checked against a set of directory names and file extensions.
- The following directories are excluded from getting locked:

AppData	**Opera Software**
Boot	**Mozilla**
Windows.old	**Mozilla Firefox**
Tor Browser	**ProgramData**
Internet Explorer	**Program Files**
Google	**Program Files (x86)**
Opera	**#recycle**

The following files are excluded from getting encrypted:

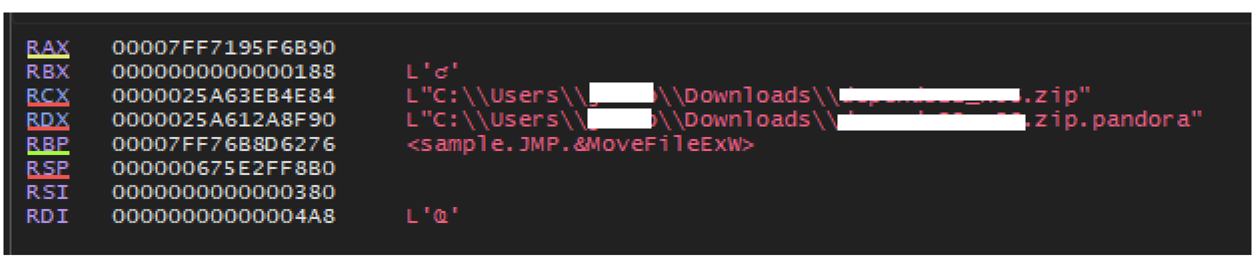
Autorun.inf	**bootmgfw.efi**
boot.ini	**desktop.ini**
bootfont.bin	**iconcache.db**
bootsect.bak	**ntldr**
bootmgr	**Ntuser.dat**
bootmgr.efi	**Restore_My_Files.txt**

And the following extensions are excluded from getting locked:

.hta	**.cur**
.exe	**.drv**

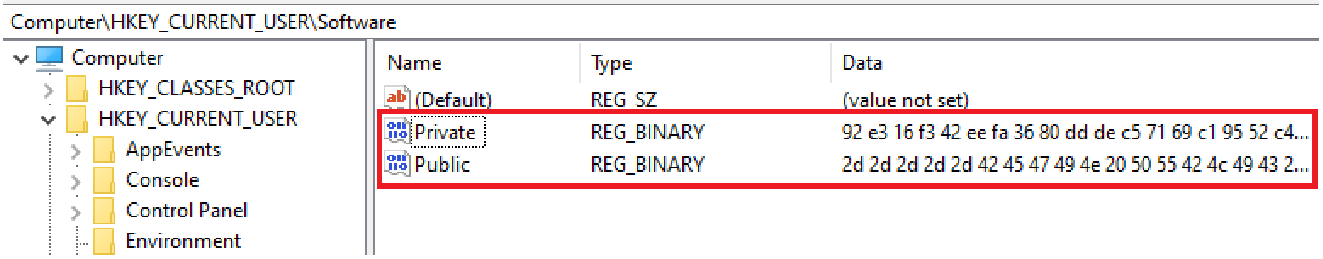
.dll	.hlp
.cpl	.icl
.ini	.icns
.cab	.ico
.idx	.sys
.spl	.ocx
.pandora	

- After performing exclusion checks, the absolute path of the file that passed the check is computed and then the thread calls for **PostQueuedCompletionStatus** to submit the path to the I/O queue previously created via **CreateIoCompletionPort**.
- Right after the PostQueuedCompletionStatus call, the same worker thread can resume fetching the absolute path of the next file via FindNextFileW API.
- Another worker thread can now call **GetQueuedCompletionStatus** to retrieve the absolute path of the target file to start encrypting the files.
- Next, the file attribute is changed via SetFileAttributesW API to **FILE_ATTRIBUTE_NORMAL** and then the file is fetched for encryption via the following APIs:
  - **CreateFileW**
  - **GetFileSizeEx**
  - **ReadFile**
  - **SetFilePointerEx**
- After setting up the file pointer to the target data, the encryption begins by loading the public key in the memory, and the encrypted data is written to the file via **WriteFile** API. Later the file is renamed via **MoveFileExW** API to add "**.pandora**" extension to the encrypted file.



Renamed file with the ".pandora" extension

Registry Keys



HKCU registry key

Pandora ransomware writes two values, **Private** and **Public**, under the **HKCU/ Software** registry key. The public value has the public key used by the ransomware to encrypt the user files, while the private value has the protected private key stored for decryption. The decryptor tool that the victim receives after paying the ransom uses this information stored in the registry to decrypt the locked files.

Indicators of Compromise

Binary
5b56c5d86347e164c6e571c86dbf5b1535eae6b979fede6ed66b01e79ea33b7b
Registry
HKCU\Software\Private
HKCU\Software\Public
Dropped Files
Restore_My_Files.txt

Author Details

Anandeshwar Unnikrishnan
Threat Intelligence Researcher , CloudSEK

Anandeshwar is a Threat Intelligence Researcher at CloudSEK. He is a strong advocate of offensive cybersecurity. He is fuelled by his passion for cyber threats in a global context. He dedicates much of his time on Try Hack Me/ Hack The Box/ Offensive Security Playground. He believes that “a strong mind starts with a strong body.” When he is not gymming, he finds time to nurture his passion for teaching. He also likes to travel and experience new cultures.

-
-

Aastha Mittal

Total Posts: 0

Sorry! The Author has not filled his profile.

×

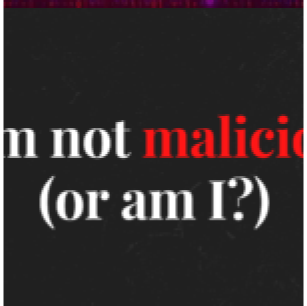
Anandeshwar Unnikrishnan

Threat Intelligence Researcher , CloudSEK

Anandeshwar is a Threat Intelligence Researcher at CloudSEK. He is a strong advocate of offensive cybersecurity. He is fuelled by his passion for cyber threats in a global context. He dedicates much of his time on Try Hack Me/ Hack The Box/ Offensive Security Playground. He believes that “a strong mind starts with a strong body.” When he is not gymming, he finds time to nurture his passion for teaching. He also likes to travel and experience new cultures.

-
-

Latest Posts

- 
pandora
- 
not
(or am I?)
- 
wiper
- 
malware