

Forum:[Hack Forums](#)

Thread: [Complex Multi-Layered Encryption Challenge: Recovering Data from Old Drives](#)

MitVap - September 9, 2024, 7:42 am

I've got a complex encryption setup that I'm trying to crack my 2 drives from 14 years ago have been saved by my parents, my mtgox has 2200 btc that they wont give me and curious what is on these drives. In particular, my electrum cold storage wallet.

Here's what I'm dealing with:

Physical Drives:

1. Disk 0 (931.51 GB total): (Fake OS)
 - D: 100 MB NTFS
 - G: 931.41 GB NTFS
 - H: 1 MB RAW (Photorec pulled 2 dll's (1 dll after decompiling mentions bitlocker) and 2 reg files.(1 reg file is 68.74 mb containing information about the Ghost OS)
2. Disk 1 (149.05 GB): (Ghost OS)
 - Shows as "Unallocated" and "Not Initialized"
 - Fully encrypted thrice it seems
 - Contains a VM that shows as 4.5 GB but is compressed from 41 GB

Virtual Machine (pulled by photorec from disk 1):

- VMDK file (119185848.vmdk) - fragmented into 5 x 1GB sections

Encryption Layers (at least 4):

1. Custom obfuscation layer
2. TrueCrypt
3. BitLocker
4. PGP

What I've tried:

- TestDisk for partition recovery
- PhotoRec for file carving (found 4 files from H: drive, 55000 on the Disk 1 (unallocated space & uninitialized disk))
- Binwalk, xxd, and ent for analysis
- Cryptsetup for LUKS detection (negative)
- Autopsy for forensic analysis - Has resulted in 0 findings
- Passware

3. File System Anomalies:

- NTFS signature detected at the 2MB mark of Disk 1, despite being "unallocated".
- Presence of INDX and PE header signatures at various offsets.

Additional Key Information:

I've got a bunch of PGP Volume Master Keys extracted from an E01 image created by FTK Imager. These keys were pulled by Elcomsoft after a scan, but interestingly, Elcomsoft couldn't detect the

encryption to apply the keys. Here's a breakdown:

Total Keys: 38

AES-128 Keys: 23

AES-256 Keys: 15 (including 2 marked as 'PGP WDE FlpDisk1')

Key examples (first 4 chars shown):

1. 842d... (AES-128)
2. a565... (AES-256)
3. bafc... (AES-256, PGP WDE FlpDisk1)
4. 0705... (AES-256)

The full keys are 32 characters (128-bit) or 64 characters (256-bit) long.

What's puzzling is that despite having these keys, I can't seem to find where or how to apply them. The encryption layers are not being detected in a way that allows straightforward key application.

Any ideas on how to use these keys effectively, given the multi-layered encryption setup I described earlier? Could these keys be for a specific layer, or might they be part of a larger encryption scheme?

What I've found:

- BitLocker references in a decompiled DLL
- Multi-language strings in DLL (English, German, Norwegian)
- High entropy in later portions of the drive

Specific Technical Challenges:

1. Decrypting the custom outer layer without corrupting inner layers. (I made backups for obvious reasons)
2. Identifying the correct sequence of decryption keys for each layer.
3. Reconstructing the fragmented and sparse VMDK file.
4. Dealing with potentially nested encrypted volumes within the VM.
5. Handling the mix of full-disk, file-based, and possibly sector-based encryption methods.

I'm particularly interested in techniques for:

- Analyzing the transition points between encryption layers.
- Methods to identify and separate the custom obfuscation layer from standard encryption.
- Tools or scripts for reassembling and mounting sparse, encrypted VMDK files.
- Approaches to apply multiple decryption methods sequentially without data corruption.

Any insights on tackling these specific technical challenges would be immensely helpful. I wish I never gave up the security life for straight dev work, clearly my much younger self was brighter at this than I am now.

Overall, is this a feasible crack given the circumstances?

DramaKing - September 9, 2024, 1:28 pm

TL;DR. Give me some time to work through this.

MitVap - September 10, 2024, 9:04 am

(Sep 9, 2024 01:28 PM)DramaKing Wrote: [\[CITE\]](#) TL;DR. Give me some time to work through this.

Update:

From the "1 MB" logical drive unallocated space

Files analyzed:

- 2 DLL files (36 KB and 79 KB)
- 2 registry files (192 KB and 68,740 KB)

Key findings:

1. Custom obfuscation layer references in the 68MB registry file, with offsets 0x20000 and 0x30000 mentioned.
2. One DLL references a 0x40000 offset, but not getting much else from this DLL.
3. Registry values in Korean and Mandarin Chinese for registered devices, with a translation to English as "Stupid" surrounded by ASCII characters.
4. Evidence of sophisticated system manipulation:
 - Disabling BitDefender (repeatedly killed my scans) and Sysmon logging
 - LSASS credential dumping
 - File association modifications (.evtx, plist/d)
 - Modification of explorer.exe default behaviors
 - Poison Ivy (PIVY) RAT installation utilizing WinRing0.sys
 - BitDefender couldn't remove the items it did detect threats which were both Application.Agent.JDL (Aided Frame, Aided Direction) which is from the time these reg files were created.
 - Attempted camera access via a dead Zoom URL join

MitVap - September 12, 2024, 10:03 am

(Sep 10, 2024 09:04 AM)MitVap Wrote: [\[CITE\]](#) Update:

From the "1 MB" logical drive unallocated space

Files analyzed:

- 2 DLL files (36 KB and 79 KB)
- 2 registry files (192 KB and 68,740 KB)

Key findings:

1. Custom obfuscation layer references in the 68MB registry file, with offsets 0x20000 and 0x30000 mentioned.
2. One DLL references a 0x40000 offset, but not getting much else from this DLL.
3. Registry values in Korean and Mandarin Chinese for registered devices, with a translation to English as "Stupid" surrounded by ASCII characters.
4. Evidence of sophisticated system manipulation:
 - Disabling BitDefender (repeatedly killed my scans) and Sysmon logging
 - LSASS credential dumping
 - File association modifications (.evtx, plist/d)
 - Modification of explorer.exe default behaviors
 - Poison Ivy (PIVY) RAT installation utilizing WinRing0.sys
 - BitDefender couldn't remove the items it did detect threats which were both Application.Agent.JDL (Aided Frame, Aided Direction) which is from the time these reg files were created.
 - Attempted camera access via a dead Zoom URL join

I was able to find the other fragments of the vmrk.

I used Autopsy, it got stuck in an infinite loop finding f00000000.vmdk. Outside of the loop that produced no new results, I was able to pull some data from the much smaller vmdks.

I will report back what I find if there is anything else in here.