

## New RegretLocker ransomware targets Windows virtual machines

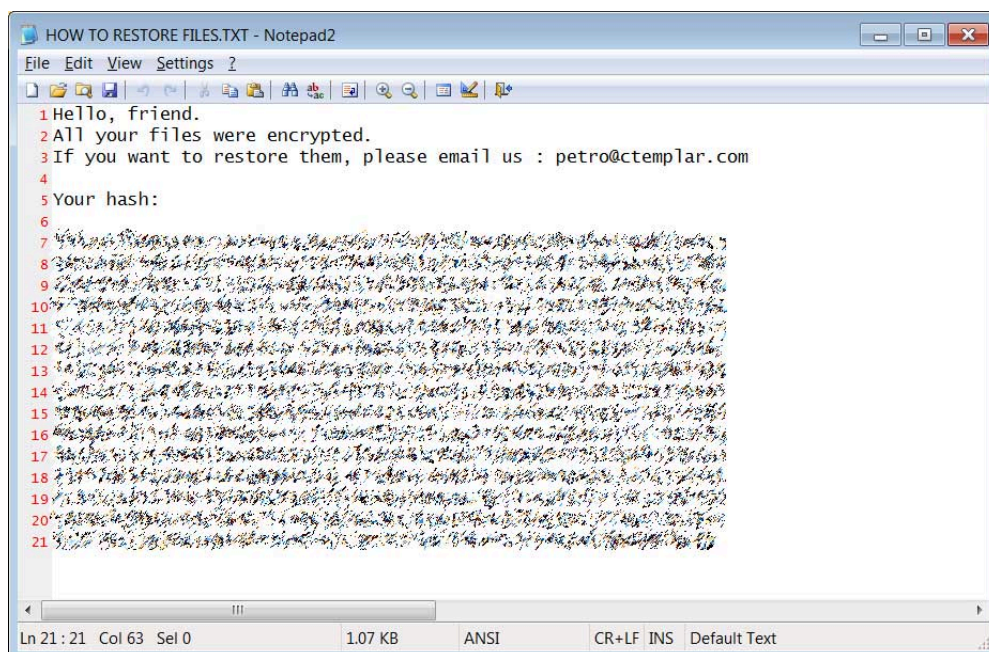
By Lawrence Abrams

Published: 2020-11-03 · Archived: 2026-04-05 20:06:14 UTC



A new ransomware called RegretLocker uses a variety of advanced features that allows it to encrypt virtual hard drives and close open files for encryption.

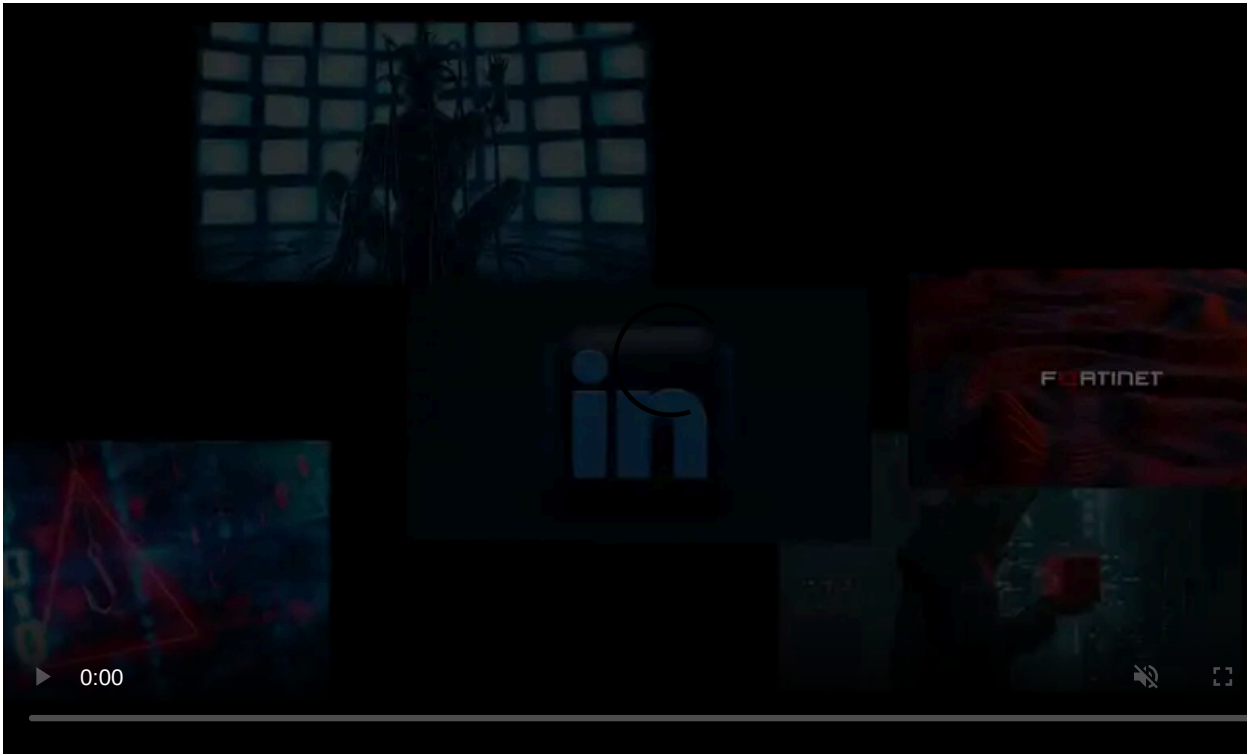
RegretLocker was discovered in October and is a simple ransomware in terms of appearance as it does not contain a long-winded ransom note and uses email for communication rather than a Tor payment site.



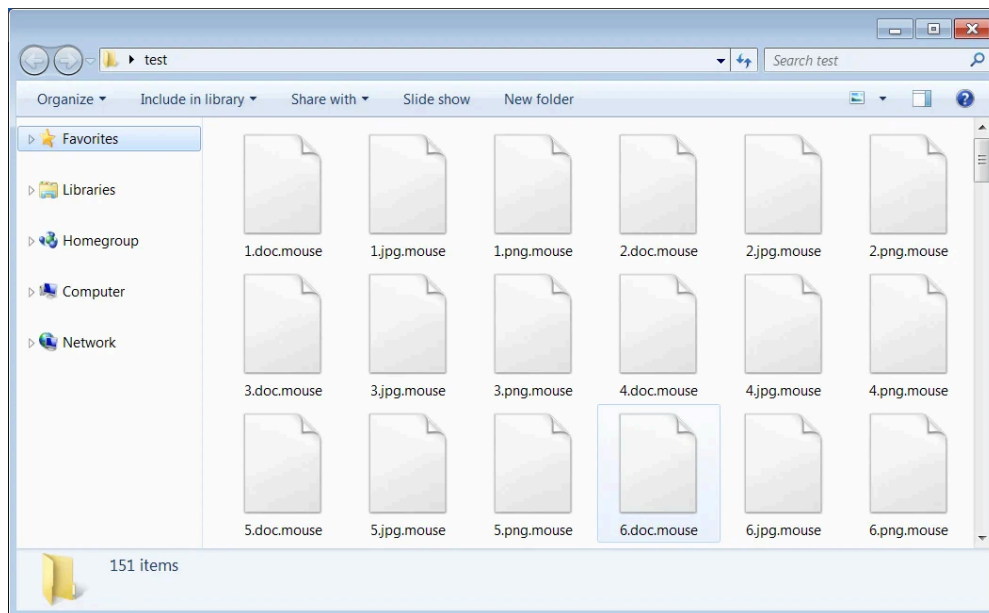
### RegretLocker ransom note

Source: BleepingComputer

When encrypting files, it will append the innocuous-sounding **.mouse** extension to encrypted file names.



Visit Advertiser website [GO TO PAGE](#)



### **RegretLocker encrypted files**

Source: BleepingComputer

What it lacks in appearance, though, it makes up for in advanced features that we do not usually see in ransomware infections, as described below.

### **RegretLocker mounts virtual hard disks**

When creating a Windows Hyper-V virtual machine, a virtual hard disk is created and stored in a VHD or VHDX file.

These virtual hard disk files contain a raw disk image, including a drive's partition table and partitions, and like regular disk drives, can range in size from a few gigabytes to terabytes.

When a ransomware encrypts files on a computer, it is not efficient to encrypt a large file as it slows down the entire encryption process's speed.

In a sample of the ransomware discovered by [MalwareHunterTeam](#) and analyzed by Advanced Intel's [Vitali Kremez](#), RegretLocker uses an interesting technique of mounting a virtual disk file so each of its files can be encrypted individually.

To do this, RegretLocker uses the Windows Virtual Storage API [OpenVirtualDisk](#), [AttachVirtualDisk](#), and [GetVirtualDiskPhysicalPath](#) functions to mount virtual disks.

```
.text:00410600_05 noud
.text:00410601 EB 76 50 00 00 call OpenVirtualDisk
.text:00410606 85 C0 test eax, eax
.text:00410608 74 32 jz short loc_41060C
.text:00410609 FF 15 74 00 45 00 call ds:GetLastError
.text:0041060D 50 push eax
.text:00410611 68 88 10 45 00 push offset a0pen_virtual_d ; "open_virtual_drive() | OpenVirtualDisk"...
.text:00410616 loc_410606: call log ; CODE XREF: sub_410637+001j
.text:0041061C EB 0B DE FF FF call log
.text:00410620 8B 75 08 mov esi, [ebp+8]
.text:00410622 8D 45 08 lea eax, [ebp+8h]
.text:00410624 59 pop ecx
.text:00410626 59 pop ecx
.text:00410628 88 50 F0 mov [ebp-10h], bl
.text:0041062A 8B CE mov ecx, esi
.text:0041062C FF 75 F8 push dword ptr [ebp-10h]
.text:0041062E 89 5E 10 mov [esi+10h], ebx
.text:00410630 50 push eax
.text:00410632 89 5E 14 mov [esi+14h], ebx
.text:00410634 EB 00 50 FF FF call sub_405702
.text:00410636 E9 CB 01 00 00 jmp loc_410687
.text:0041063C ;
.text:0041063E loc_41060C: push ebx ; CODE XREF: sub_410637+811j
.text:00410640 53 lea eax, [ebp-70h]
.text:00410642 50 push eax
.text:00410644 53 push ebx
.text:00410646 04 push ebx
.text:00410648 53 push ebx
.text:0041064A FF 75 00 push dword ptr [ebp-30h]
.text:0041064C EB 35 50 00 00 call AttachVirtualDisk
.text:00410650 85 C0 test eax, eax
.text:00410652 74 08 jz short loc_410709
.text:00410654 50 push eax
.text:00410656 EB DC 1B 45 00 push offset a0pen_virtual_0 ; "open_virtual_drive() | AttachVirtualDis"...
.text:00410658 EB 80 jmp short loc_410606
.text:0041065E ;
.text:00410660 loc_41060E: ; CODE XREF: sub_410637+811j
.text:00410662 53 push ebx
.text:00410664 04 lea eax, [ebp-70h]
.text:00410666 50 push eax
.text:00410668 53 push ebx
.text:0041066A 04 push ebx
.text:0041066C 53 push ebx
.text:0041066E FF 75 00 push dword ptr [ebp-30h]
.text:00410670 EB 35 50 00 00 call AttachVirtualDisk
.text:00410672 85 C0 test eax, eax
.text:00410674 74 08 jz short loc_410709
.text:00410676 50 push eax
.text:00410678 EB DC 1B 45 00 push offset a0pen_virtual_0 ; "open_virtual_drive() | AttachVirtualDis"...
.text:0041067A EB 80 jmp short loc_410606
.text:00410680 ;
.text:00410682 loc_410709: ; CODE XREF: sub_410637+C81j
.text:00410684 33 C0 xor eax, eax
.text:00410686 8D 80 4C FB FF FF lea edi, [ebp-40Ah]
.text:00410688 B9 82 00 00 00 mov ecx, 82h
.text:0041068A BE 04 01 00 00 mov esi, 100h
.text:0041068C F3 8B rep stosd
.text:0041068E 8D 85 4C FB FF FF lea eax, [ebp-40Ah]
.text:00410690 89 75 D4 mov [ebp-2Ch], esi
.text:00410692 50 push eax
.text:00410694 80 45 D4 lea eax, [ebp-2Ch]
.text:00410696 83 CB FF or ebx, 0FFFFFFFh
.text:00410698 50 push eax
.text:0041069A FE 75 00 push dword ptr [ebp-30h]
.text:0041069C EB 02 50 00 00 call GetVirtualDiskPhysicalPath
```

2020-11-04: RegretLocker Ransomware | open\_virtual\_drive()

### Mounting a VHD file

As shown by a debug message in the ransomware, it is specifically searching for VHD and mounting them when detected.

```
parse_files() | Found virtual drive: %ws in path: %s
```

Once the virtual drive is mounted as a physical disk in Windows, the ransomware can encrypt each one individually, which increases the speed of encryption.

The code used by RegretLocker to mount a VHD is believed to have been taken from a [recently published research](#) by security researcher [smelly\\_vx](#).

In addition to using the Virtual Storage API, RegretLocker also utilizes the [Windows Restart Manager API](#) to terminate processes or Windows services that keep a file open during encryption.

When using this API, Kremez told BleepingComputer if the name of a process contains 'vnc', 'ssh', 'mstsc', 'System', or 'svchost.exe', the ransomware will not terminate it. This exception list is likely used to prevent the termination of critical programs or those used by the threat actor to access the compromised system.

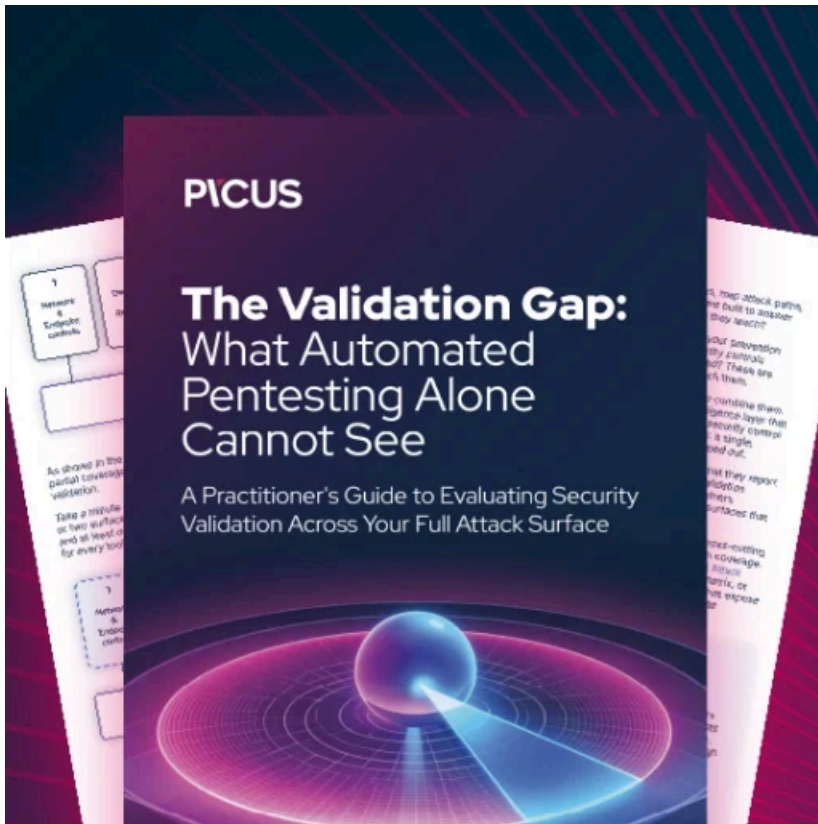
```
84 v10 = GetProcessId(0);
85 Log("net process opened file() | RmGetList Error: 0x%X", u13);
86 RmEndSession(*(DWORD *)(a1 - 20));
87 sub_418341(u12);
88 LABEL_48:
89 v23 = *(DWORD *)(a1 + 8);
90 sub_400877(a1 - 32);
91 sub_4070FC(a1 - 32);
92 sub_407182(a1 + 12);
93 goto LABEL_49;
94 }
95 v14 = *(DWORD *)(a1 - 16) == 0;
96 *(DWORD *)(a1 - 64) = 0;
97 if ( !v14 )
98 {
99 do
100 {
101 v15 = u12->Process.dwProcessId;
102 if ( v15 != GetCurrentProcessId() )
103 {
104 v16 = u12->ApplicationType;
105 if ( v16 != 4 && v16 != 1000 && v16 != 3 && v15 != -1 )
106 {
107 sub_410020(a1 - 60, v15);
108 v14 = *(DWORD *)(a1 - 44) == 0;
109 *(BYTE *)(a1 - 4) = 2;
110 if ( v14 || *(DWORD *)(a1 - 44) < 3u )
111 goto LABEL_52;
112 v17 = a1 - 60;
113 if ( *(DWORD *)(a1 - 40) >= 8u )
114 v17 = *(DWORD *)(a1 - 60);
115 if ( sub_431591(v17, L"unc") )
116 goto LABEL_52;
117 v18 = a1 - 60;
118 if ( *(DWORD *)(a1 - 40) >= 8u )
119 v18 = *(DWORD *)(a1 - 60);
120 if ( sub_431591(v18, L"ssh") )
121 goto LABEL_52;
122 v19 = a1 - 60;
123 if ( *(DWORD *)(a1 - 40) >= 8u )
124 v19 = *(DWORD *)(a1 - 60);
125 if ( sub_431591(v19, L"mstsc") )
126 goto LABEL_52;
127 v20 = a1 - 60;
128 if ( *(DWORD *)(a1 - 40) >= 8u )
129 v20 = *(DWORD *)(a1 - 60);
130 if ( sub_431591(v20, L"System") )
131 goto LABEL_52;
132 v21 = a1 - 60;
133 if ( *(DWORD *)(a1 - 40) >= 8u )
134 v21 = *(DWORD *)(a1 - 60);
135 if ( sub_431591(v21, L"svchost.exe") )
136 {
137 LABEL_52:
138 0000F271 : 84
```

2020-11-04: RegretLocker  
Ransomware |  
get\_process\_opened\_file()  
-> RMGetList | Exception

#### Windows Restart Manager exception list

The Windows Restart Manager feature is only used by a few ransomware such as [REvil \(Sodinokibi\)](#), Ryuk, [Conti](#), [ThunderX/Ako](#), [Medusa Locker](#), [SamSam](#), and [LockerGoga](#).

RegretLocker is not very active at this point, but it is a new family that we need to keep an eye on.



### [Automated Pentesting Covers Only 1 of 6 Surfaces.](#)

Automated pentesting proves the path exists. BAS proves whether your controls stop it. Most teams run one without the other.

This whitepaper maps six validation surfaces, shows where coverage ends, and provides practitioners with three diagnostic questions for any tool evaluation.

---

Source: <https://www.bleepingcomputer.com/news/security/new-regretlocker-ransomware-targets-windows-virtual-machines/>