

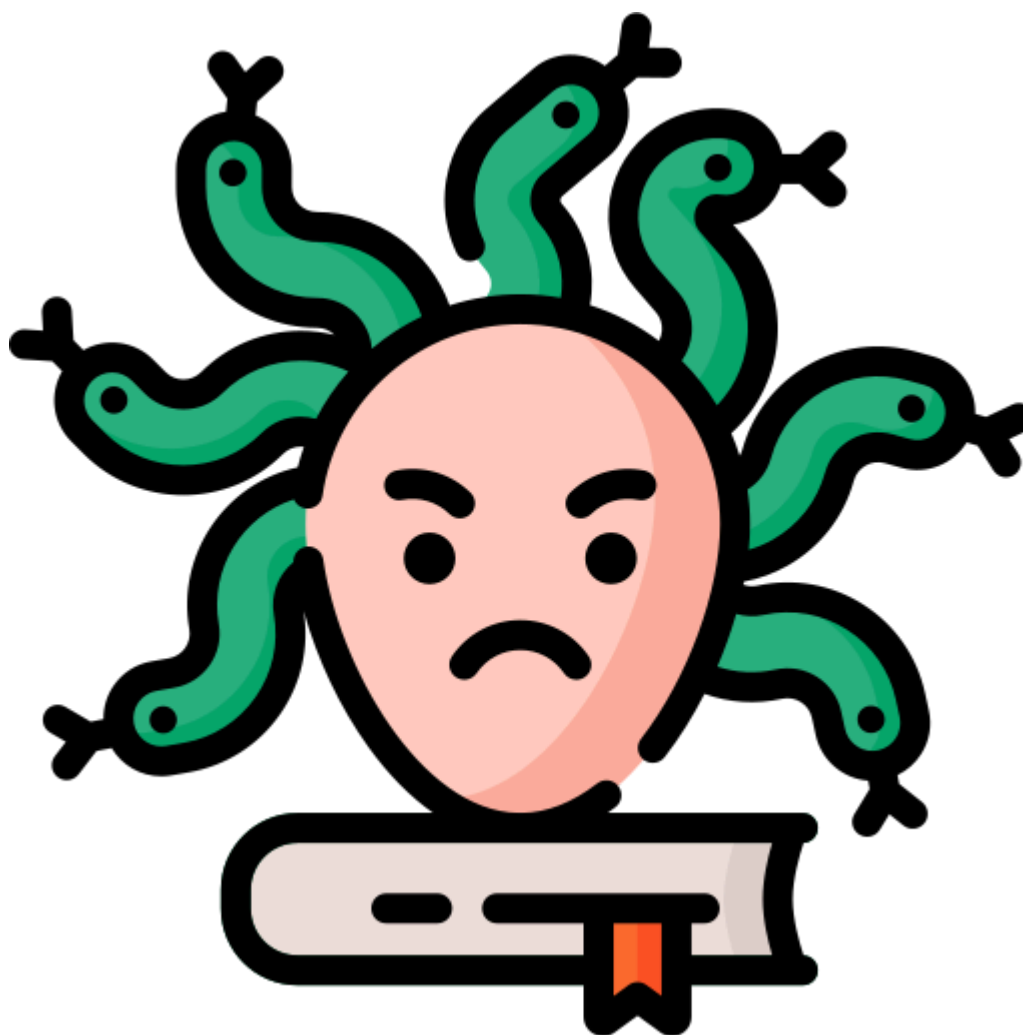
Try not to stare - medusalocker at a glance

By f0wL

Published: 2019-11-05 · Archived: 2026-04-06 01:04:00 UTC

Tue 05 November 2019 in [Ransomware](#)

Mystic but also a new(-ish) threat: Medusa ransomware. Let's take a quick peek, but don't look too close or you may need to fetch backups soon.



A general disclaimer as always: downloading and running the samples linked below will lead to the encryption of your personal data, so be f\$cking careful. Also check with your local laws as owning malware binaries/sources might be illegal depending on where you live.

medusa.exe @ [AnyRun](#) --> sha256 3a5b015655f3aad4b4fd647aa34fda4ce784d75a20d12a73f8dc0e0d866e7e01

dix_16.exe @ [HybridAnalysis](#) --> sha256
49da42d00cc3ad6379ead2e07fd5f09bd358b144a6e78aad4bb1a8298e2bb568

Taking a look at the stringdump that [stringsifter](#) produced one of the first things that stood out was this base64 encoded image:

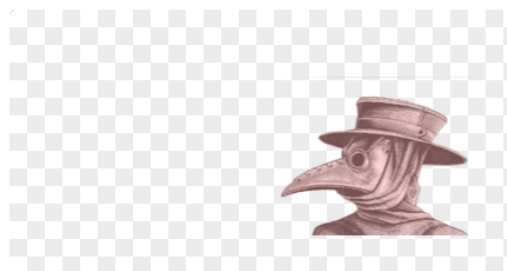
```
background: url("data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAL4AAADACAMAAACNgikeAAAC/VMVUEUAAA5AgK+hoY3AADBiyFjIy7g40nb2/IkJCgaGircnK1fHyenq4gICaYWGtdXWwd3ekbGyTW1vDioqXXL7GjY21fn7Mk5PPL5c9BQVHDg7Smpo6AgJ5QUF1PDx/R0eLU1NbIiKES0udZWV8RERoMdB0FhbaoaFz0jrwnp5YHx+JUFBLEXONVVVhKSleJSXgp6eBSkpsMzNKEhJBCAiHTk5UHB2KUVGkbw1wNzdSHBxRG RndpKQ6AwSeZWVBCguhampkLXL6QWFiPWFHEDAx2Pz+GTU2QWVnMLZXmr69fJyhHDxHa2tggZ2fjq6uDSkpiKiVbjo5tNjZWHx9EDQ9ECwty0TLli79xcXVpaXjrKxqMzNvNjbXoKDxULjnsbHco6PutbXEjY08Bc3P0u7vCiorrtLT6w8NLFZbIiW1eXn/yMj2vr7Fj4+4g4PpsrK8iIiamqPVLaqc30rdnadZWXDi4vNk50faGicZGR9RESfZ2e2fn65goKXX1+jamqbZGSMVFSjbGyaY2Sxe3uyenu0fHyZYGCVNFS3fX2RWFilbm7Kk50GUIL7ztdxvLpBdLc9ETE5TW1t009vYV163200U1750UEv0TiACE5a3fLcV20cLbU1fk11TFic0h3L4gT8E9IETvDMEGw9hTAd7cDwUkHhcmhVChy0TmLhAFiTi3f3LUECDU
```

After decoding it we get an image of a medieval pest doctor. Fun fact: They wore these masks because they thought it would protect them from the black death. One day someone will probably start selling these for endpoint protection.

base64

```
iVBORw0KGgoAAAANSUHEUgAAAL4AAADACAMAAACNgikeAAAC/VMVUEUAAA5AgK+hoY3AADBiyFjIy7g40nb2/IkJCgaGircnK1fHyenq4gICaYWGtdXWwd3ekbGyTW1vDioqXXL7GjY21fn7Mk5PPL5c9BQVHDg7Smpo6AgJ5QUF1PDx/R0eLU1NbIiKES0udZWV8RERoMdB0FhbaoaFz0jrwnp5YHx+JUFBLEXONVVVhKSleJSXgp6eBSkpsMzNKEhJBCAiHTk5UHB2KUVGkbw1wNzdSHBxRG RndpKQ6AwSeZWVBCguhampkLXL6QWFiPWFHEDAx2Pz+GTU2QWVnMLZXmr69fJyhHDxHa2tggZ2fjq6uDSkpiKiVbjo5tNjZWHx9EDQ9ECwty0TLli79xcXVpaXjrKxqMzNvNjbXoKDxULjnsbHco6PutbXEjY08Bc3P0u7vCiorrtLT6w8NLFZbIiW1eXn/yMj2vr7Fj4+4g4PpsrK8iIiamqPVLaqc30rdnadZWXDi4vNk50faGicZGR9RESfZ2e2fn65goKXX1+jamqbZGSMVFSjbGyaY2Sxe3uyenu0fHyZY
```

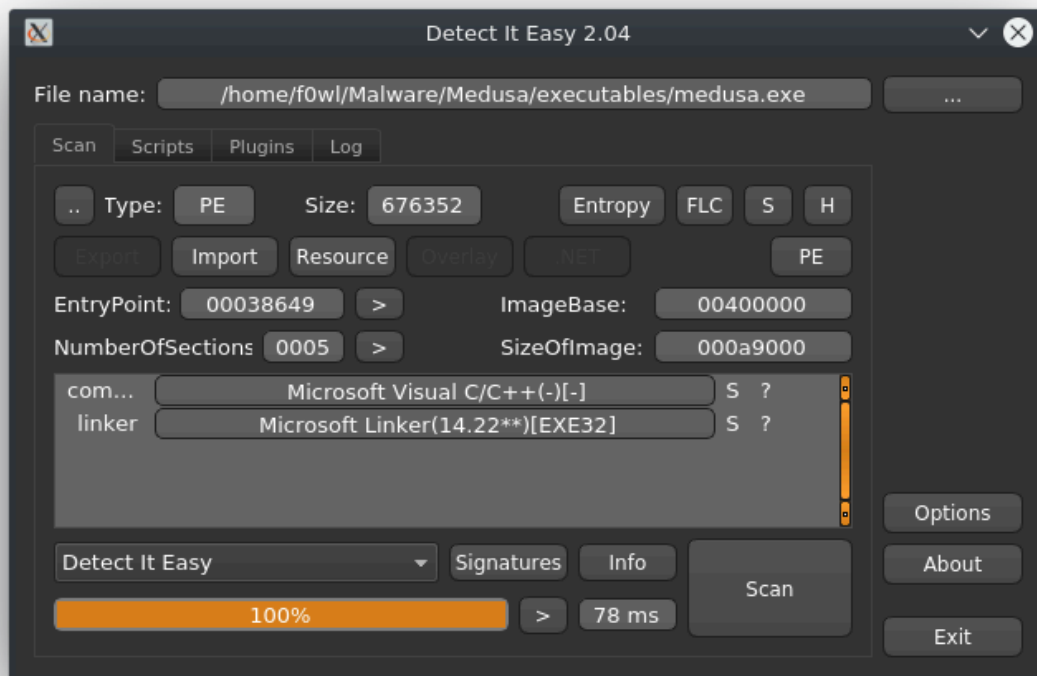
png



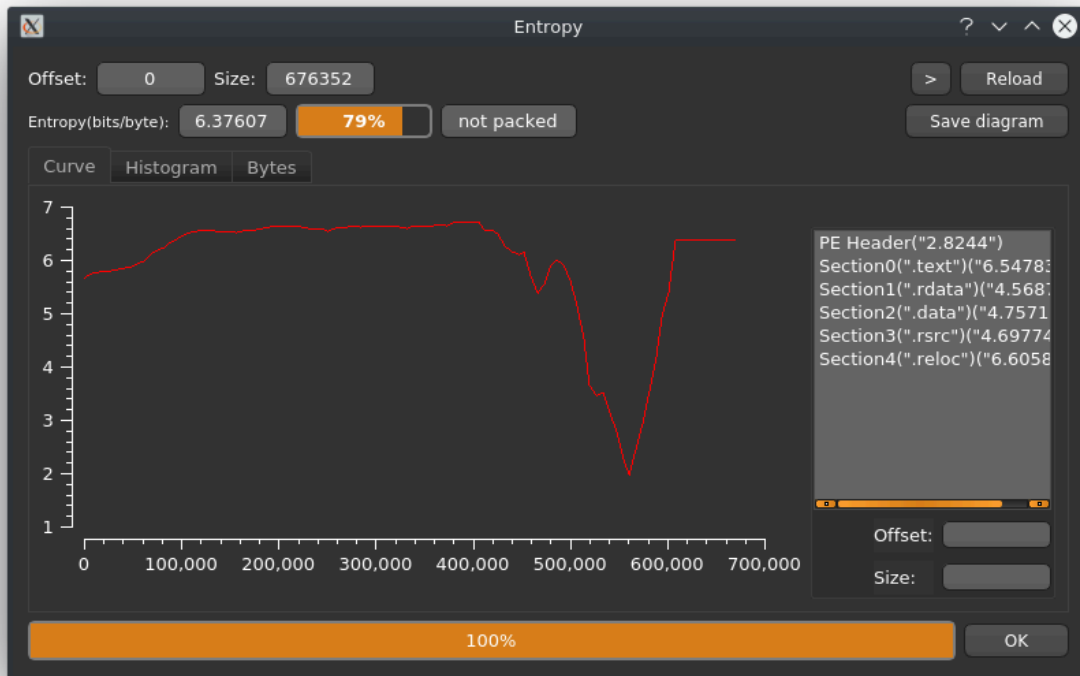
Another interesting extracted string is this PDB-Path:

C:\Users\Gh0St\Desktop\MedusaLockerInfo\MedusaLockerProject\MedusaLocker\Release\MedusaLocker.pdb

Running it through Detect it easy returns that MedusaLocker was built with Visual C++ and a (in malware-terms) relatively new Linker Version.



Entropy-wise it doesn't look like this sample is packed and the sections found don't look out of the ordinary either.



After digging around in Ghidra for a bit I found **FUN_00405bc0** which seems to be the main program routine of MedusaLocker. The strings shown here match the output in the debug console present in the second sample discussed below.

```
Decompile: FUN_00405bc0 - (medusa.exe)
58  function-void __cdecl(void) { local_90 [24],
59  undefined local_78 [44];
60  undefined local_4c [24];
61  error_condition local_34 [12];
62  error_condition local_28 [12];
63  undefined local_1c [15];
64  _Mutex_base local_d;
65  _Mutex_base local_c;
66  _Mutex_base local_b;
67  _Mutex_base local_a;
68  _Mutex_base local_9;
69  uint local_8;
70
71  local_8 = DAT_0049d074 ^ (uint)&stack0xffffffffc;
72  FUN_004528d8(0,&DAT_004804da);
73  uVar1 = FUN_00405410(&DAT_004804db,0x3f);
74  FUN_00424a8f(local_13c,uVar1);
75  FUN_004054a0();
76  FUN_004054a0();
77  pwVar4 = L"[LOCKER] Is running\n";
```

Yet another mysterious CLSID that I can't make sense of at the moment: {8761ABBD-7F85-42EE-B272-A76179687C63}. Search results referencing it are around since October 21st and might make tracking Medusa a bit easier.

```

pwVar4 = L"[LOCKER] Is running\n";
_Mymtx(&local_ed);
FUN_00401720(pwVar4);
FUN_00407ae0(L"{8761ABBD-7F85-42EE-B272-A76179687C63}");
local_d9 = FUN_00405580(local_234);
~function<void__cdecl(void)>(local_234);
if (local_d9 == '\0') {
    _Mymtx(&local_a);
    FUN_0041f570();
    uVar2 = FUN_0041f4d0();
    if ((uVar2 & 0xff) == 0) {
        userLevel = L"[LOCKER] Priv: USER\n";
    }
    else {
        userLevel = L"[LOCKER] Priv: ADMIN\n";
    }
    local_134[0] = userLevel;
    ppwVar5 = local_134;
}

```

Next up the Locker will "initialize the crypto module" which uses [CryptGenKey](#) provided by WinCrypt to derive a keypair. I'll have a closer look at the encryption routine later.

```

pwVar4 = L"[LOCKER] Init cryptor\n";
_Mymtx(&local_f0);
FUN_00401720(pwVar4);
p_Var3 = _Mymtx((_Mutex_base *)&DAT_004a1ac0);
uVar1 = move<>(p_Var3);
uVar2 = FUN_004150f0(uVar1);
if ((uVar2 & 0xff) == 0) {
    pwVar4 = L"[LOCKER] Init cryptor is failed\n";
    _Mymtx(&local_f1);
    FUN_00401720(pwVar4);
    FUN_00405ba0();
    local_118 = 0;
    FUN_004150c0();
    FUN_004016f0();
}
else {
    pwVar4 = L"[LOCKER] Put ID to HTML-code\n";
    _Mymtx(&local_f2);
    FUN_00401720(pwVar4);
    FUN_00407e10("{IDENTIFIER}");
    uVar1 = FUN_00415230(local_264);
    uVar1 = move<>(uVar1);
    uVar2 = FUN_00411a80(local_1d4, uVar1);
    local_da = (uVar2 & 0xff) == 0;
    local_104 = (uint)local_da;
    FUN_00407d60();
    FUN_00407d60();
}
}

```

It will skip files with the following suffixes:

```
exe, dll, sys, ini, lnk, rdp, encrypted
```

As it is very popular with Ransomware to disable the Automatic Startup Repair and delete System Restore Points plus shadow copies Medusa will do so as well. After that it will also relaunch **LanmanWorkstation** to ensure that mapped network drives are available.

```

pwVar4 = L"[LOCKER] Remove backups\n";
_Mymtx(&local_f4);
printConsole(pwVar4);
FUN_0041d910();
FUN_00407ae0(L"vssadmin.exe Delete Shadows /All /Quiet");
FUN_0041d860(local_174);
~function<void__cdecl(void)>(local_174);
FUN_00407ae0(L"bcdedit.exe /set {default} recoveryenabled No");
FUN_0041d860(local_18c);
~function<void__cdecl(void)>(local_18c);
FUN_00407ae0(L"bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures");
FUN_0041d860(local_1a4);
~function<void__cdecl(void)>(local_1a4);
FUN_00407ae0(L"wbadmin DELETE SYSTEMSTATEBACKUP");
FUN_0041d860(local_1bc);
~function<void__cdecl(void)>(local_1bc);
FUN_00407ae0(L"wbadmin DELETE SYSTEMSTATEBACKUP -deleteoldest");
FUN_0041d860(local_24c);
~function<void__cdecl(void)>(local_24c);
FUN_00407ae0(L"wmic.exe SHADOWCOPY /nointeractive");
FUN_0041d860(local_1ec);
~function<void__cdecl(void)>(local_1ec);
FUN_00405770(1);
FUN_00407ae0(L"LanmanWorkstation");
FUN_0041dab0(local_204);
~function<void__cdecl(void)>(local_204);
FUN_00407ae0(L"LanmanWorkstation");
FUN_0041db70(local_21c);
~function<void__cdecl(void)>(local_21c);
_Mymtx(&local_c);
if (((uVar1 & 0xff) == 0) &&
    (local_24c = (HANDLE)CreateToolhelp32Snapshot(2,0), local_24c != (HANDLE)0xffffffff)) {
    FUN_0044f430(local_244,0,0x22c);
    local_244[0] = 0x22c;
    iVar2 = Process32FirstW(local_24c,local_244);
    while (iVar2 != 0) {
        FUN_00407ae0(local_220);
        local_245 = FUN_0041d630(param_1,local_270);
        ~function<void__cdecl(void)>(local_270);
        if ((local_245 != '\0') && (local_254 = OpenProcess(1,0,local_23c), local_254 != (HANDLE)0x0))
        {
            TerminateProcess(local_254,0);
            CloseHandle(local_254);
            goto LAB_0041da91;
        }
        iVar2 = Process32NextW(local_24c,local_244);
    }
    CloseHandle(local_24c);
}

```

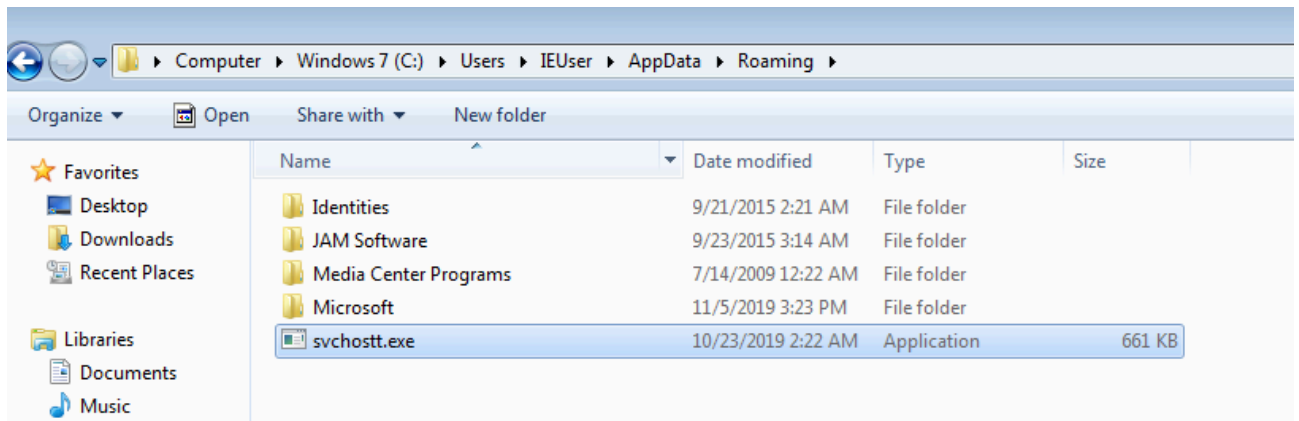
After the "Adding to Autoload" debug message it will rename itself to svchost.exe and add it's Registry Key to the System startup.

```
if (local_da == false) {
    _Mymtx(&local_d);
    pwVar4 = L"[LOCKER] Add to autoload\n";
    _Mymtx(&local_f7);
    printConsole(pwVar4);
    FUN_00407ae0(L"svchostt");
    uVar1 = FUN_00411d50();
    FUN_0041e120(local_15c,uVar1);
    ~function<void__cdecl(void)>(local_15c);
    pwVar4 = L"[LOCKER] Stop and delete services\n";
    _Mymtx(&local_f6);
    printConsole(pwVar4);
    _Mymtx(&local_9);
    local_108 = (error_condition *)FUN_00411d10();
    local_e8 = (error_category *)value(local_108);
}
```

MedusaLocker will try to terminate the following processes by their name. The List contains Security Software as well as Services commonly used in productive environments such as SQL or Webservers.

wrapper, DefWatch, ccEvtMgr, ccSetMgr, SavRoam, sqlservr, sqlagent, sqladhlp, Culserver, RTVscan, sq
QBIDPService, Intuit.QuickBooks.FCS, QBCFMonitorService, sqlwriter, msmdsrv, tomcat6, zhudongfangyu,
vmware-usbarbitator64, vmware-converter, dbsrv12, dbeng8wxServer.exe, wxServerView, sqlservr.exe, sq
RAgui.exe, supervise.exe, Culture.exe, RTVscan.exe, Defwatch.exe, sqlbrowser.exe, winword.exe, QBW32
qbupdate.exe, QBCFMonitorService.exe, axlbridge.exe, QBIDPService.exe, httpd.exe, fdlauncher.exe, Msl
tomcat6.exe, java.exe, 360se.exe, 360doctor.exe, wdsfwfsafe.exe, fdlauncher.exe, fdhost.exe, GDscan.e

It also copies itself to %APPDATA% after renaming to executable to "svchostt.exe".



To check if an instance of MedusaLocker previously ran on the system it will create a Registry Key at HKEY_CURRENT_USER\Software\Medusa

```

if (RegCreateKeyW(HKEY_CURRENT_USER, L"SOFTWARE\\Medusa", (int32_t **)&phkResult) == 0) {
    // 0x40561d
    g8 = v4;
    function_407830();
    g8 = v4;
    int32_t lpData = function_407850(); // 0x405632
    g8 = phkResult;
    RegSetValueExW((int32_t *)phkResult, L"Name", 0, 1, (char *)lpData, (int32_t)&g1130);
    g10 = phkResult;
    RegCloseKey((int32_t *)phkResult);
}

```

Furthermore it tries to read the State of EnableLinkedConnections via **RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" ...** and enables the key if necessary since Medusa tries to encrypt Shared Network Drives and removable Media as well.

```

if ((a1 & 255) == 0) {
    // 0x4057d8
    char * phkResult; // bp-20
    g10 = (int32_t)&phkResult;
    int32_t v3 = RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System", 0, 0xf003f, (int32_t **)&phkResult); // 0x4057ed
    g5 = v3;
    if (v3 == 0) {
        // 0x4057fc
        RegDeleteValueW((int32_t *)phkResult, L"EnableLinkedConnections");
        g8 = (int32_t)phkResult;
        g5 = RegCloseKey((int32_t *)phkResult);
    }
} else {
    // 0x405788
    int32_t phkResult2; // bp-16
    g8 = &phkResult2;
    int32_t v4 = RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System", 0, 0xf003f, (int32_t **)&phkResult2); // 0x40579d
    g5 = v4;
    if (v4 == 0) {
        int32_t lpData = 1; // bp-12
        g10 = &lpData;
        RegSetValueExW((int32_t *)phkResult2, L"EnableLinkedConnections", 0, 4, (char *)&lpData, 4);
        g8 = phkResult2;
        g5 = RegCloseKey((int32_t *)phkResult2);
    }
}

```

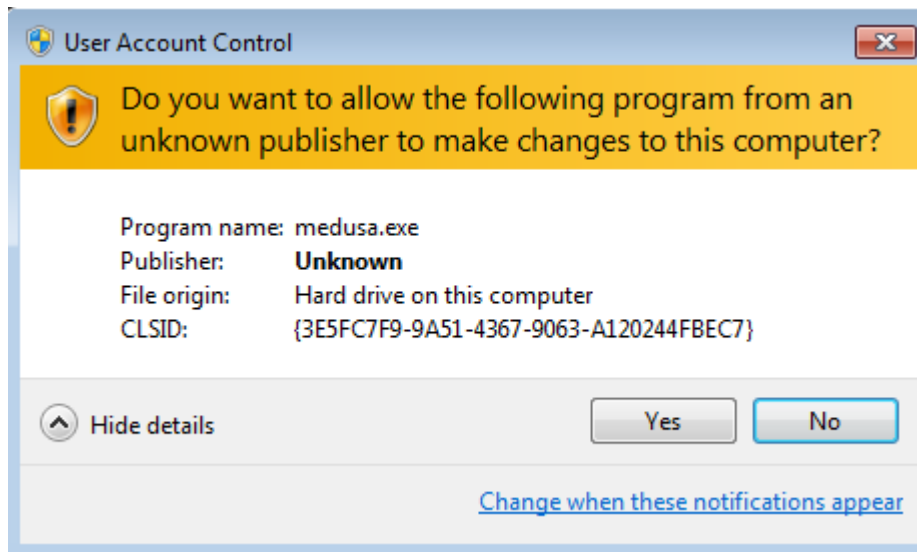
After terminating the encryption loop the Ransomware will wait for 60 seconds and start a new scan to check for new unencrypted files.

```

pwVar4 = L"[LOCKER] Sleep at 60 seconds...\n\n";
_Mymtx(&local_f9);
printConsole(pwVar4);
Sleep(60000);
FUN_004076c0();
FUN_004076c0();
} while( true );

```

Running MedusaLocker in a VM yields us this UAC Prompt with a mysterious CLSID ({3E5FC7F9-9A51-4367-9063-A120244FBEC7}). A quick google search brings us to Wikileaks Page for the [CIA Vault7 leaks](#) and the ID seems to be corresponding to *cmstplua.dll*. Turns out this is an UAC bypass known and implemented since August 2017 (mentioned [here](#)).



The Ransomnote (which is dropped in every directory that contains files to encrypt) is delivered as a HTML file. In this early sample they seem to have messed up their text alignment. This was fixed in a later version (see below) and will make it easier to identify new samples as they may appear.

A screenshot of a ransom note displayed in a browser. The text is as follows:

All your data are encrypted!

What happened?
Your files are encrypted, and currently unavailable.
You can check it: all files on you computer has new expansion.
By the way, everything is possible to recover (restore), but you need to buy a unique decryptor.
Otherwise, you never cant return your data.

For purchasing a decryptor contact us by email:
Folieloi@protonmail.com
If you will get no answer within 24 hours contact us by our alternate emails:
Ctorsenoria@tutanota.com

What guarantees?
Its just a business. If we do not do our work and liabilities - nobody will not cooperate with us.
To verify the possibility of the recovery of your files we can decrypted 1 file for free.
Attach 1 file to the letter (no more than 10Mb). Indicate your personal ID on the letter:
D6CA4F73CFDE831A49B26C300CFE4AF4792190E4C5D358305CAD014FA7B7C16FEE632957282D83506E41FCD5D2594877BDFDE38AA17BCB837841A90F3DFD68C1
EF91E9A90AB18358DC593CF01AF7FA0522FBCC96C3F011A42E0320F5468E47289720595245C45CED7142D674EFBC4B3082E66A94934942A9CB375DB9924B
5D6B999E2ACD1BD28BCECFED416B2B6378FC0A9A0B85F046B184A341711918F6D0A4702527D11EBD59D77A8812E3080E89EE783137E05CD3A70310C1571C
9B8EAE4DDEF0C396046319A2700D56A987A3A22F554B14AD8D0B78AF7B98E0DA6443953FD82A865B8DEF9A7D2C71DEA4AC17A03853630A63BBE5B63D1A11
8464A04A116D2F9BA6D80A4D6A6C02EE859E24E3F164553B43F3BA03F32374C2698E27B2E01840A6C65B3107B0A9922FE41DF148F32293CA88F6A5AC6270
2C047CB99EB44235D3939E8609D8EEE05E7ADCAD50614A0D0787892E8D106CA7EE57CC85F3D24577E7EE4642AD177953DCA6644C3C241748EF55B103B20F
8E97213402CED50E95059851DBF343749883A3233374E1350A041E43F48F9444A7A9E44D1A636293FC2CD7500619BA22A403950681FD3F29D1B86C1928BD
79C9B4641280A15E1139EFB5B04EC56FAD626A1C188D32833FCFFCCF08F4057B8974643273956709A5AF72FA9A716694414B78C9C3644A6C25A34544BD2B
65527E2F8EA80BF770C6CCF197A

Attention!

- Attempts of change files by yourself will result in a loose of data.
- Our e-mail can be blocked over time. Write now, loss of contact with us will result in a loose of data.
- Use any third party software for restoring your data or antivirus solutions will result in a loose of data.
- Decryptors of other users are unique and will not fit your files and use of those will result in a loose of data.
- If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we have the private key.

A small illustration of a man's head with a very long, curved beak, resembling a stork or a similar bird. He is wearing a hat.

Looking at the section list compared to the



This sample seems to have an enabled debug console which allows us to trace the steps of the infection.

```
C:\Users\IEUser\Desktop\dix_16.exe
[LOCKER] Is running
[LOCKER] Priv: ADMIN
[LOCKER] Init cryptor
[LOCKER] Put ID to HTML-code
[LOCKER] Add to autorun
[LOCKER] Scan hidden devices
[LOCKER] Stop and delete services
[LOCKER] Kill processes
[LOCKER] Remove backups
[LOCKER] Run scanning...

[LOCKER] Scan C:\
[LOCKER] Scan D:\
[LOCKER] Drive: D:\, total size <gb>: 0, total time <min>: 1.66667e-05
[LOCKER] Scan E:\
[LOCKER] Drive: E:\, total size <gb>: 0, total time <min>: 3.33333e-05
```

Below you can see the new ransomnote. The Protonmail E-Mail address was exchanged for a cock.li one and the Victim ID blob was fitted to the textbox.

All your data are encrypted!

What happened?

Your files are encrypted, and currently unavailable.
You can check it: all files on you computer has new expansion.
By the way, everything is possible to recover (restore), but you need to buy a unique decryptor.
Otherwise, you never cant return your data.

For purchasing a decryptor contact us by email:

mrromber@cock.li

If you will get no answer within 24 hours contact us by our alternate emails:

mrromber@tutanota.com

What guarantees?

Its just a business. If we do not do our work and liabilities - nobody will not cooperate with us.

To verify the possibility of the recovery of your files we can decrypted 1 file for free.

Attach 1 file to the letter (no more than 10Mb). Indicate your personal ID on the letter:

098BE04ED57A226C0BEB69B1A7D46561A934447F3695F52DD30AE20EDE75C7398A558FB4B9A8208D62EF673AA73F3527D36726D269DC2CB3AF4B05109E8832ED
DE8690A12D94BC20FD527E2319EBEDEC70F675AF1754691D2476F270DC27E68A8C1AC5080C2ACD1445574C155A8EBD74E521D95C6B26FA3641C49EED36
2A68031052025D900164B7003A6C1DA85DE5C15F5AAF1A411EE0105333D8F074D2C58A29C4023E41DE8A16C3746C9CB12D8AFB02EB6ED92841E89D7A75E4
8288906058538229F8997B11A69EA3A402F7EC10D63FA3F3BC385E4A536FF233E4C79CD9FE1575281C00581382D6666F183615FB88800F9F9D3E96B80EE0
7B8C4A47983E62D93337AA4E786A166386270A5FB91E56AF6DE5357428D7F5C008CEBFE75F0C25178C8DBA413FAFE511B4578C3D3DFA987A951ADDA71ADB
689B4D3C38C5B36C722741EB0EA7209062568774F1D1B978887F93902310FB2DA696271F8A374FBC725E109104274C0DEBAE3431D50F249D2DB4CA1FE33
4615F48C3D585214768BB4CE6E7370082B3860859A74FA73898228F09D0493D8DD98A9A125AA6D22F96C5489C2809648832F906868E602ACDE1F58B38006
778047582E417FAC0CC7E9937884FAA8267848B14A74488AAC51C0BE9C518AED2269FF9924561A4489CECA0864E2E16C73201D7A688F19EC0AD0EEA12C2
F7F8397B1BEFF2E3728FA2F3533E

Attention!

- Attempts of change files by yourself will result in a loose of data.
- Our e-mail can be blocked over time. Write now, loss of contact with us will result in a loose of data.
- Use any third party software for restoring your data or antivirus solutions will result in a loose of data.
- Decryptors of other users are unique and will not fit your files and use of those will result in a loose of data.
- If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we have the private key.



BleepingComputer Forum User trifonov who was hit by the ransomware as well found suspicious files on his Desktop after the Infection took place. Fortunately for us Medusa skipped the executables.



trifonov - 1 week ago

Hello, I got infected with this ransomware, but the person who infected the system forgot this tools on my desktop and most of them are on affected from the ransomware.

Here is a link to the ransomware the person used on my PC and what ever else...

<https://drive.google.com/open?id=1P6-ZyzyhDjYaETHrC9vefryJCp1aCpSLW>

I put a password on the zip file - pass is 123123

And here is a link with 2 files infected and the same clean:

https://drive.google.com/open?id=1r2opIaU9dQW_7CC2MUrhca4j04gmbC7S

same password on the zip file: 123123

Hope anyone to find a way to decrypt the files.

This would be a huge discovery infection vector-wise as this looks like the attacker gained access to the machine via RDP. (Yet another proof [if we would need any] that RDP exposed to the internet isn't a good idea)

```
4096 Oct 29 01:16 .
4096 Nov 6 23:46 ..
8830152 Aug 5 2018 'Advanced Port Scanner 2.4.2750.exe'
8728 Oct 29 01:16 b.bat.decryptme
2685952 Oct 21 14:39 dix_16.exe
2616832 Oct 21 14:40 dix_16_xp.exe
98816 Aug 10 09:00 d_upd1008.exe
27825 Oct 29 01:16 HOW_TO_OPEN_FILES.html
4096 Oct 29 01:16 kamikadze
115200 Nov 1 2017 NetworkShare.exe
128000 Dec 11 2018 NetworkShare_pre2.exe
374944 Jun 28 2016 PsExec64.exe
339096 Jun 28 2016 PsExec.exe
```

Looks like the attacker left a few files related to Mimikatz as well...

```
4096 Oct 29 01:16 .
4096 Oct 29 01:16 ..
814232 Aug 14 01:32 32.exe
1013912 Aug 14 01:32 64.exe
16920 Oct 29 01:16 64_log.txt.decryptme
536 Oct 29 01:16 86_log.txt.decryptme
8728 Oct 29 01:16 dump.bat.decryptme
27825 Oct 29 01:16 HOW_TO_OPEN_FILES.html
29416 Jan 22 2013 'mimidrv (2).sys'
36584 Jan 22 2013 mimidrv.sys
41624 Aug 14 01:32 'mimilib (2).dll'
46744 Aug 14 01:32 mimilib.dll
```

As I mentioned earlier the keypair is generated via CryptGenKey. I'm still trying to map out all the actions on the key material.

```

*****
*                               FUNCTION                               *
*****
uint __thiscall generateKey(void * this, HCRYPTKEY * hKey)
uint EAX:4 <RETURN>
void * ECX:4 (auto) this
HCRYPTKEY * Stack[0x4]:4 hKey XREF[1]: 00415c99(R)
undefined1 Stack[-0x5]:1 local_5 XREF[3]: 00415cb5(W),
00415cbb(W),
00415cbf(R)
undefined4 Stack[-0xc]:4 local_c XREF[2]: 00415c96(W),
00415ca4(R)
XREF[1]: FUN_00415aa0:00415ad5(c)

generateKey
00415c90 55 PUSH EBP
00415c91 8b ec MOV EBP,ESP
00415c93 83 ec 08 SUB ESP,0x8
00415c96 89 4d f8 MOV dword ptr [EBP + local_c],this
00415c99 8b 45 08 MOV EAX,dword ptr [EBP + hKey]
00415c9c 50 PUSH EAX
00415c9d 6a 01 PUSH 0x1
00415c9f 68 10 66 PUSH 0x6610
00000000
00415ca4 8b 4d f8 MOV this,dword ptr [EBP + local_c]
00415ca7 8b 51 08 MOV EDX,dword ptr [this + 0x8]
00415caa 52 PUSH EDX
00415cab ff 15 18 CALL dword ptr [->ADVAPI32.DLL::CryptGenKey]
00000000
00415cb1 85 c0 TEST EAX,EAX
00415cb3 74 06 JZ LAB_00415cbb
00415cb5 c6 45 ff 01 MOV byte ptr [EBP + local_5],0x1
00415cb9 eb 04 JMP LAB_00415cbf

```

The encryption itself is done via the [CryptEncrypt](#) function. It seems to use AES for the files and then encrypts the key with a RSA-2048 public key that is stored via a keyblob in the executable.

Input	start: 368 end: 368 length: 0	length: 368 lines: 1	+ [] [] [] []
<pre> BgIAAAckAABSU0EXAAgAAEAQAQ5or+RgfErIaLFVzjyVkmKuKntqEltkzU8ysp9i8hx1MnB0DJk2Q4D040ZE3rFmviVcd78XvB2+cxUcyvmykj04L13om7 bwlCzY06/AbtJjn6CmBQBwmxY1FZTIcUPxdefJGYQCzYdHKGqAk8LRu1N1gmU++hwKZmbk3FCcEKIgzDZ+4MqixEtHx51Edq604hRm1yBjwTj79LE /tlEzbh80BntCQV5BAfSkVYcd1BbSb/XjuttIM7RmoerHdZAxiojgRhdLY62zzPDVwb6S0PeCutd0mmBpe10blntg7c9Trf911 /dNKCfprYOPX4Rr73hmEnJ3EwgD5hxQP0TcRiWq </pre>			
Output	start: 276 end: 276 length: 0	time: 4ms length: 276 lines: 2	[] [] [] [] []
<pre>RSA1.....yCz..ñ+!@EW8ðVC ,@i`Im.5<ÉÉ}.Èq0ÉÁ82dÜ..ó...zA.ø.qpú^ðvùìTs+æEHiã%wcn0ÁP³`î¿.»I.~...Älr0VS!A.Åx.\$f..6..jª.O.Fim0 .ùèV)...qBpB..6~àÈcÀKGÇ.Dv@.â.f× cÁ8ú0±?¶Q3n...{BA^A.ú\$U..ô.0oðã°0b3`fjêÇu.1..âF.Kc.ªîð0Á%.Ð±.Rxt.`iz).. {`îIS.ÿux+M('é...kîxf.rw...æ.P?DÚF%ª </pre>			

The image displays a debugger window titled '00415da0 - moreCryptEncryptStuff'. The main window shows assembly code with a stack frame table at the top. The stack frame table lists parameters: HCRYPTKEY (hKey), byte (final), undefined4 (param_3), DWORD (blkSize), DWORD (dwBufLen), and three undefined4 locals (local_10, local_11, local_18). The assembly code includes instructions like PUSH, MOV, SUB, XOR, LEA, CALL, and JZ. A yellow circle highlights the code from 00415da0 to 00415df0. A control flow graph below shows three branches: a red arrow from 00415df0 to 00415df2, a green arrow from 00415df0 to 00415df8, and a blue arrow from 00415df2 to 00415df8. The 00415df2 window shows a JMP instruction to 00415dfc. The 00415df8 window shows a MOV instruction. The 00415dfc window shows the end of the function with a RET instruction.

```
uint __stdcall moreCryptEncryptStuff (HCRYPTKEY hKey, byt...
uint          EAX:4          <-RETURN->
HCRYPTKEY      Stack[0x4]:4   hKey
byte          Stack[0x8]:1   final
undefined4    Stack[0xc]:4   param_3
DWORD         Stack[0x10]:4   blkSize
DWORD         Stack[0x14]:4   dwBufLen
undefined4    Stack[-0x10]:4 local_10
undefined1    Stack[-0x11]:1 local_11
undefined4    Stack[-0x18]:4 local_18

moreCryptEncryptSt ...
...5da0 PUSH  EBP
...5da1 MOV  EBP,ESP
...5da3 PUSH -0x1
...5da5 PUSH LAB_0046e5f0
...5daa MOV  EAX,FS:[0x0]
...5db0 PUSH EAX
...5db1 SUB  ESP,0x8
...5db4 MOV  EAX,[DAT_0049d074 ]
...5db9 XOR  EAX,EBP
...5dbb PUSH EAX
...5dbc LEA EAX=>local_10,[EBP + -0xc]
...5dbf MOV  FS:[0x0],EAX
...5dc5 MOV  dword ptr [EBP + local_18]...
...5dc8 MOV  EAX,dword ptr [EBP + dwBuf...
...5dcb PUSH EAX
...5dcc LEA ECX=>blkSize,[EBP + 0x14]
...5dcf PUSH ECX
...5dd0 PUSH 0x0
...5dd2 MOV  ECX,dword ptr [EBP + param...
...5dd5 CALL FUN_00412ec0
...5dda PUSH EAX
...5ddb PUSH 0x0
...5ddd MOVZX EDX,byte ptr [EBP + final]
...5de1 PUSH EDX
...5de2 PUSH 0x0
...5de4 MOV  EAX,dword ptr [EBP + hKey]
...5de7 PUSH EAX
...5de8 CALL dword ptr [->ADVAPI32.DLL: ...
...5dee TEST  EAX,EAX
...5df0 JZ   LAB_00415df8

00415df2
...5df2 MOV  byte ptr [EBP + local_11],...
...5df6 JMP  LAB_00415dfc

00415df8 - LAB_00415df8
LAB_00415df8
...5df8 MOV  byte ptr [EBP + local_11],...

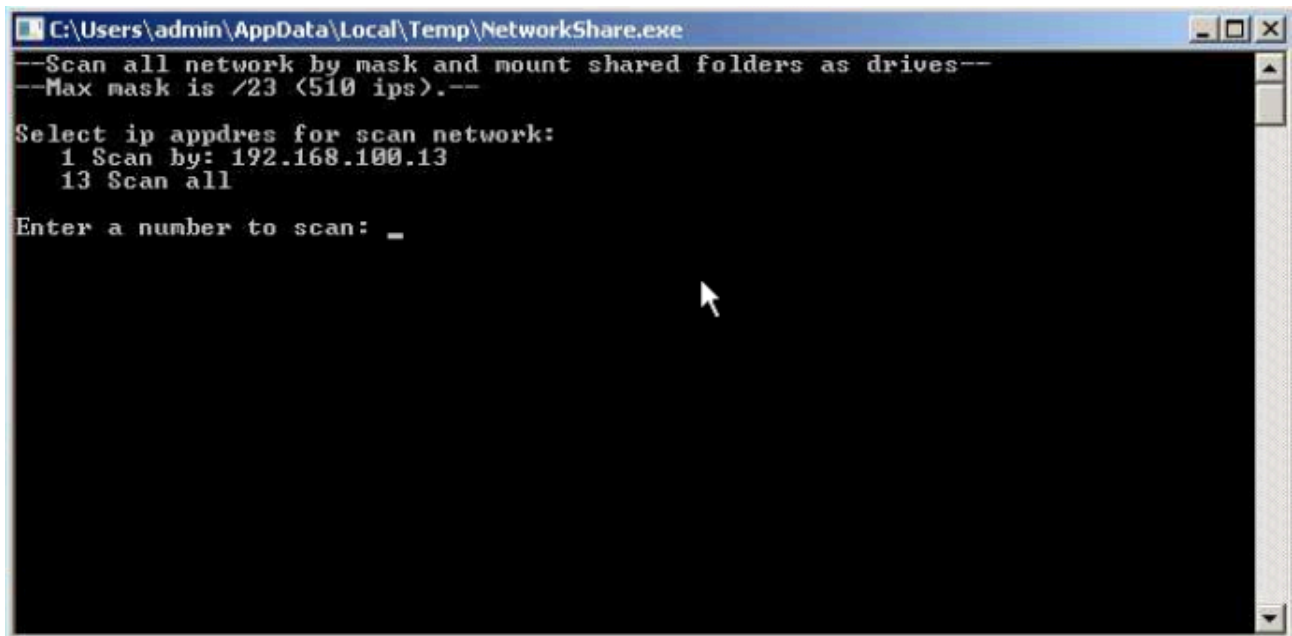
If / Else
00415dfc - LAB_00415dfc
LAB_00415dfc
...5dfc MOV  AL,byte ptr [EBP + local_1...
...5dff MOV  ECX,dword ptr [EBP + local...
...5e02 MOV  dword ptr FS:[0x0],ECX
...5e09 POP  ECX
...5e0a MOV  ESP,EBP
...5e0c POP  EBP
...5e0d RET  0x14
```

After the encryption routine is done the generated hKey is deleted via [CryptDestroyKey](#).

```
*****  
*                               FUNCTION                               *  
*****  
undefined __stdcall destroyKey(HCRYPTKEY hKey)  
undefined HCRYPTKEY AL:1 <RETURN>  
Stack[0x4]:4 hKey XREF[2]: 00415cd7(R),  
00415cdd(R)  
undefined4 Stack[-0x8]:4 local_8 XREF[1]: 00415cd4(w)  
destroyKey XREF[1]: FUN_00415190:004151aa(c)  
00415cd0 55 PUSH EBP  
00415cd1 8b ec MOV EBP,ESP  
00415cd3 51 PUSH ECX  
00415cd4 89 4d fc MOV dword ptr [EBP + local_8],ECX  
00415cd7 83 7d 08 00 CMP dword ptr [EBP + hKey],0x0  
00415cdb 74 0a JZ LAB_00415ce7  
00415cdd 8b 45 08 MOV EAX,dword ptr [EBP + hKey]  
00415ce0 50 PUSH EAX  
00415ce1 ff 15 40 CALL dword ptr [->ADVAPI32.DLL::CryptDestroyKey]  
20 47 00
```

Update 23.11.2019:

Now I want to take a closer look at the files left by the attacker on the Victim's Desktop as it was reported multiple times on the BleepingComputer Forum. Besides the Mimikatz files in the kamikadze directory there is a semi-legit tool called "Advanced Port Scanner" ([AnyRun](#), which is basically just a garbage Zenmap alternative for Windows people) and another one called "NetworkShare.exe" ([AnyRun](#), seems to scan for reachable network shares and tries to mount them).



It also looks like there's a dedicated version of MedusaLocker for Windows XP called *dix_16_xp.exe*. As you can see below the Debug Messages start with **[LockerXP]** instead of **[Locker]**.

```
C:\Users\admin\AppData\Local\Temp\dix_16_xp.exe
[LOCKER XP] Is running
[LOCKER] Init cryptor
[LOCKER XP] Put ID to HTML-code
[LOCKER XP] Add to autorun
[LOCKER] Scan hidden devices
[LOCKER] Assign device \\?\Volume{e1a82db3-a9f0-11e7-b142-806e6f6e6963}\ letter
Z:\

[LOCKER XP] Stop and delete services
[LOCKER XP] Kill processes
[LOCKER XP] Remove backups
[LOCKER XP] Run scanning...

[LOCKER] Scan C:\
-
```

The Decryptor 🤖

The Decryptor is delivered per Machine with a 4 letter filename indicating to which victim ID it belongs.

```
12288:Tz0sQ2igqLuKbeht10RbxekoPfaTVNyXx2GAD0TFo2PTUvmPKV8nKnXzi:IgigKdbehpL0NxeKo6TVN8AD0po0r6va,"/home/f0wL/Malware/Medusa/decryptors/4BC.exe"
12288:Tz0sQ2igqLuKbeht10RbxekoPfaTVNyXx2GAD0TFo2PTUvmPKV8nKnXzi:IgigKdbehpL0NxeKo6TVN8AD0po0r6va,"/home/f0wL/Malware/Medusa/decryptors/8479.exe"
12288:Tz0sQ2igqLuKbeht10RbxekoPfaTVNyXx2GAD0TFo2PTUvmPK18nKnXzi:IgigKdbehpL0NxeKo6TVN8AD0po0r+vva,"/home/f0wL/Malware/Medusa/decryptors/d21.exe"
12288:Tz0sQ2igqLuKbeht10RbxekoPfaTVNyXx2GAD0TFo2PTUvmPKc8nKnXzi:IgigKdbehpL0NxeKo6TVN8AD0po0rLva,"/home/f0wL/Malware/Medusa/decryptors/D70.exe"
12288:Tz0sQ2igqLuKbeht10RbxekoPfaTVNyXx2GAD0TFo2PTUvmPK3S8nKnXzi:IgigKdbehpL0NxeKo6TVN8AD0po0rqSS,"/home/f0wL/Malware/Medusa/decryptors/F059.exe"
```

```
C:\Users\admin\AppData\Local\Temp\D70.exe
[UNLOCKER] Is running
[UNLOCKER] Priv: ADMIN
[UNLOCKER] Kill Locker
[UNLOCKER] Delete task: svhost
[UNLOCKER] Locker name not found
[UNLOCKER] Run scanning...

[UNLOCKER] Scan C:\
-
```

Offset	Name	Func. Count	Bound?	OriginalFirstTh	TimeDateStar	Forwarder	NameRVA
8A6D4	KERNEL32.dll	130	FALSE	8BFF0	0	0	8C3B8
8A6E8	ADVAPI32.dll	15	FALSE	8BF9C	0	0	8C4DC
8A6FC	SHELL32.dll	1	FALSE	8C224	0	0	8C4FC
8A730	ole32.dll	3	FALSE	8C234	0	0	8C53E
8A724	OLEAUT32.dll	4	FALSE	8C210	0	0	8C548
8A738	MPR.dll	1	FALSE	8C1FC	0	0	8C56C
8A74C	NETAPI32.dll	2	FALSE	8C204	0	0	8C598
8A760	IPHLAPI.DLL	4	FALSE	8BFDC	0	0	8C5EC
8A774	WS2_32.dll	1	FALSE	8C22C	0	0	8C5FA

Call via	Name	Ordinal	Original Thunl	Thunk	Forwarder	Hint
74018	CryptDestroyKey	-	8C4CA	8C4CA	-	C8
7401C	CryptAcquireContextW	-	8C482	8C482	-	C2
74020	CryptEncrypt	-	8C4A2	8C4A2	-	CB
74024	CryptDuplicateKey	-	8C48E	8C48E	-	CA
74028	CryptDecrypt	-	8C47E	8C47E	-	C5
7402C	CryptImportKey	-	8C46C	8C46C	-	DB
74030	CryptReleaseContext	-	8C456	8C456	-	DC
74034	RegDeleteKeyW	-	8C438	8C438	-	26F
74038	GetTokenInformation	-	8C3C6	8C3C6	-	170

IOCs

Medusa (SHA256)

```

medusa.exe --> SHA256: 3a5b015655f3aad4b4fd647aa34fda4ce784d75a20d12a73f8dc0e0d866e7e01
                SSDEEP: 12288:f+IZ+bobAyYFJPrsU4VwryxjpBx8aji0hA8tsV1YRbRb7:2++EMyYFJPoUecOh8aWdD1UB7

dix_16.exe --> SHA256: 49da42d0cc3ad6379ead2e07fd5f09bd358b144a6e78aad4bb1a8298e2bb568
                SSDEEP: 24576:nJC1YA0p0eRaNaQgxPubcoiukAby3LV1jqjx9/WBRQ/8PxS//LTQKJfF27:nw10fMGxRoiul

dix_16_xp.exe --> SHA256: 6c7eda3f5e9bbc685b0eefde2a51f0ccb06ad33805e617876a5124410cac9945
                SSDEEP: 24576:Sx7USQ2bEdBF4XUCAdbpH7KYlvnIVGDDUWuXr00VY/QjFdIkyoRn:MISXu5C47KMIaDW
    
```

E-Mail Addresses

```

Ctorsenoria@tutanota[.]com
Folieloi@protonmail[.]com
mrromber@cock[.]li
mrromber@tutanota[.]com
sambolero@tutanoa[.]com
rightcheck@cock[.]li
fartcool@protonmail[.]ch
bestcool@keemail[.]me
tanoss@protonmail[.]com
sypress@protonmail[.]com
    
```

Associated Files

```
svchostt.exe
HOW_TO_OPEN_FILES.html
Advanced Port Scanner 2.4.2750.exe
d_upd1008.exe
NetworkShare_pre2.exe
PsExec64.exe (legitimate)
PsExec.exe (legitimate)
b.bat
NetworkShare.exe
kamikadze/32.exe
kamikadze/64.exe
kamikadze/64_log.txt
kamikadze/dump.bat
kamikadze/mimidrv (2).sys
kamikadze/mimilib (2).dll
kamikadze/86_log.txt
kamikadze/mimidrv.sys
kamikadze/mimilib.dll
```

Registry Keys

```
HKCU\SOFTWARE\Medusa
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System --> EnableLinkedConnections = 1
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System --> ConsentPromptBehavior = 1
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System --> EnableLUA = 1
```

Ransomnote

All your data are encrypted!
What happened?
Your files are encrypted, and currently unavailable.
You can check it: all files on you computer has new expansion.
By the way, everything is possible to recover (restore), but you need to buy a unique decryptor.
Otherwise, you never cant return your data.

For purchasing a decryptor contact us by email:

mrromber@cock.li

If you will get no answer within 24 hours contact us by our alternate emails:

mrromber@tutanota.com

What guarantees?

Its just a business. If we do not do our work and liabilities - nobody will not cooperate with us.

To verify the possibility of the recovery of your files we can decrypted 1 file for free.

Attach 1 file to the letter (no more than 10Mb). Indicate your personal ID on the letter:

54E87CD3C1529DD06EB22FF80C49B5374ABB8E5B30D06E13BBE2E81411234A20DF1ADA53FDA68BD6294C96DAC3049B4BDC50

```
FE764BF468AF1A029B41162759D6164EB0652E95D3FAE3939773B505073E6090079C9C9243EE8B96AEB41A43B787B47DD01D  
425E042C6CBDE89BB5F2E7F9CC6601BD9430E87B42A56BEEFF207F20F9E4E5E48FA3274AE0DE8D65EEC0F2BA2CC4AECB22A9I  
6FD2B21FF152A6A11BD86D063A965C1571078A439C97D52215738104F7B6EF7415CC4A2C03260BCB9A84E71E088326874774.  
39CFF3002697B8AD04E01A6B6DC0A460F4273778429962A7AECEEE3BA16A577A6B1D6B67A7FAEFA5C9CB8BBCEFC3FF6B04I  
BE5D37B69B42BBEE2EA0D00C7439858D2D9BD4A57B47F3E05EBF913F5FAB195AF0575DD345E84347A82010CDC4C0507C9868  
C61ED4091E4155585A687EAB73CBEA8ADA7B93B5EB67877CDD0E35C9116B8DCADD2038C4EEAC42302F3B787E54F8AD24012E.  
A89B3C32252BD438399FAE630A1E099E9D130E7EA7E042841B468FF00FCF86B9C07C054827EE76956211CE70FEB686EC1997  
34C96D1D35DD713CA33774C4D5D0
```

Attention!

- Attempts of change files by yourself will result in a loose of data.
- Our e-mail can be blocked over time. Write now, loss of contact with us will result in a loose of c
- Use any third party software for restoring your data or antivirus solutions will result in a loose
- Decryptors of other users are unique and will not fit your files and use of those will result in a
- If you will not cooperate with our service - for us, its does not matter. But you will lose your t

Medusa Icon made by Freepik from www.flaticon.com

Source: <https://dissectingmalwa.re/try-not-to-stare-medusalocker-at-a-glance.html>