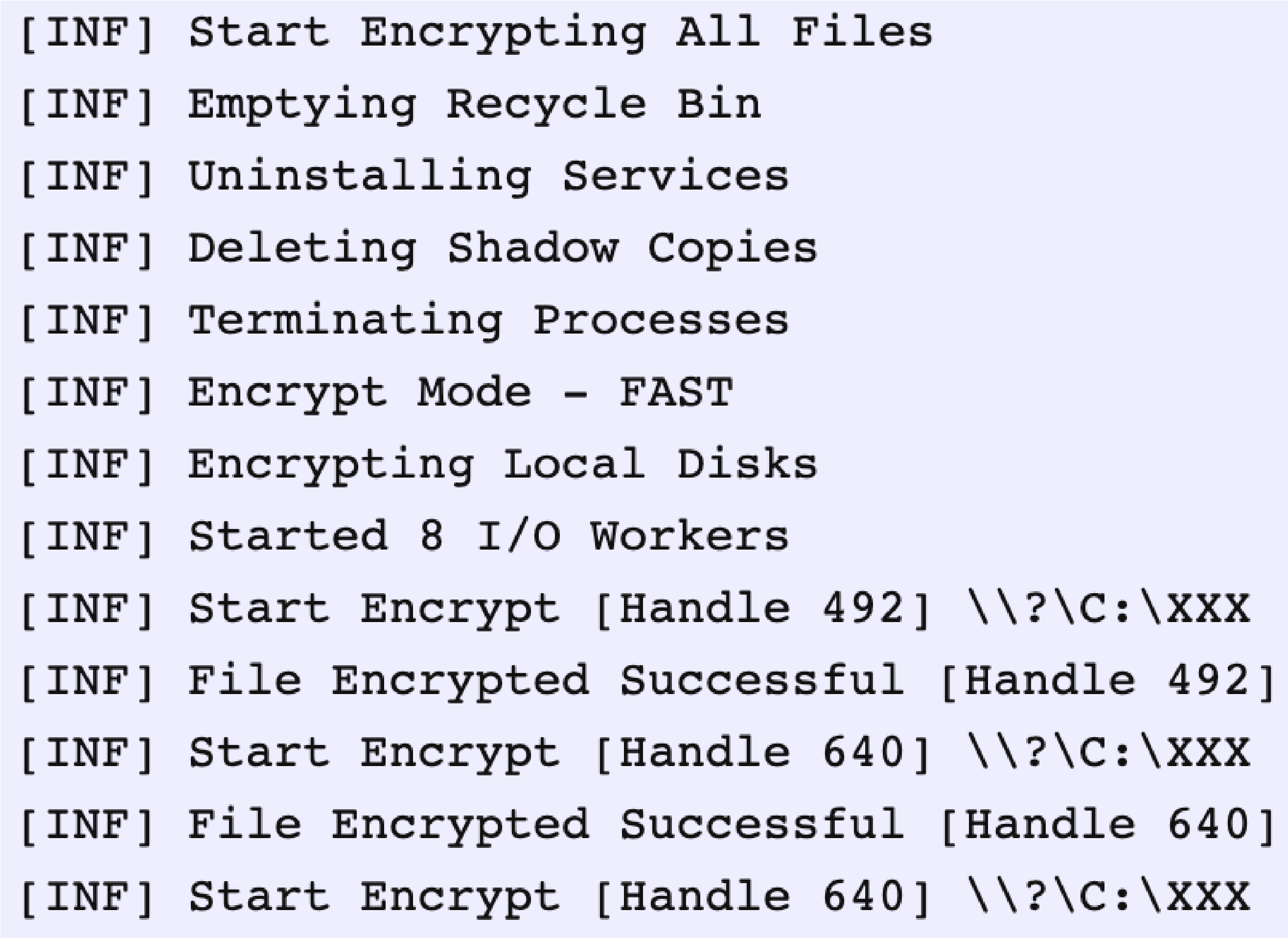
DarkSide ransomware analysis

zawadidone.nl/darkside-ransomware-analysis/

October 5, 2020



```
[INF] Start Encrypting All Files
[INF] Emptying Recycle Bin
[INF] Uninstalling Services
[INF] Deleting Shadow Copies
[INF] Terminating Processes
[INF] Encrypt Mode - FAST
[INF] Encrypting Local Disks
[INF] Started 8 I/O Workers
[INF] Start Encrypt [Handle 492] \\?\C:\XXX
[INF] File Encrypted Successful [Handle 492]
[INF] Start Encrypt [Handle 640] \\?\C:\XXX
[INF] File Encrypted Successful [Handle 640]
[INF] Start Encrypt [Handle 640] \\?\C:\XXX
```

October 5, 2020 2 minute read

This blog post will try to explain how the ransomware called DarkSide works. Based on my research, this ransomware uses Salsa20 encryption to encrypt files and RSA encryption to encrypt the key used by Salsa20. A new key is created per file based on random bytes.

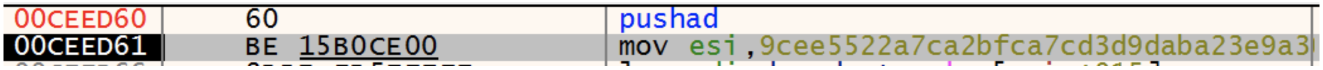
> A new ransomware operation named DarkSide began attacking organizations earlier this month with customized attacks that have already earned them million-dollar payouts.
>
> Starting around August 10th, 2020, the new ransomware operation began performing targeted attacks against numerous companies.
>
> In a “press release” issued by the threat actors, they claim to be former affiliates who had made millions of dollars working with other ransomware operations. https://www.bleepingcomputer.com/news/security/darkside-new-targeted-ransomware-demands-million-dollar-ransoms/amp/

Unpacking

The executable is compressed with UPX

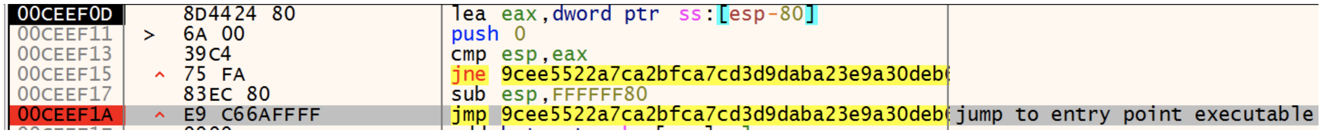
```
file 9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205f56c12045839075f7627297
[...]: PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
```

After the first instruction `pushad` I put a breakpoint on the `ESP` register and continue.

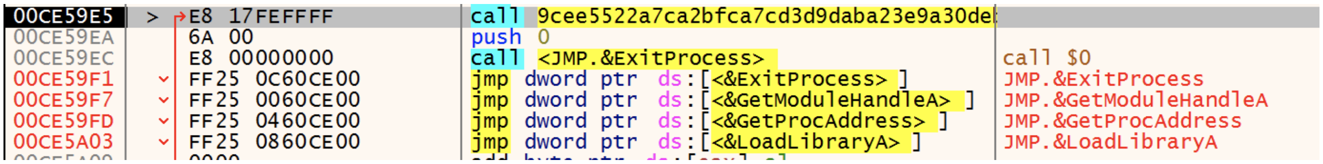


```
00CEED60      60              pushad
00CEED61      BE 15B0CE00     mov esi,9cee5522a7ca2bfca7cd3d9daba23e9a3
```

The execution breaks on the instruction `lea eax, dword ptr ss:[esp80]`. After the loop is executed it jumps to the entry point of the packed executable.



```
00CEEF0D      8D4424 80     lea eax,dword ptr ss:[esp-80]
00CEEF11  >   6A 00         push 0
00CEEF13      39C4          cmp esp,eax
00CEEF15  ^   75 FA         jne 9cee5522a7ca2bfca7cd3d9daba23e9a30deb
00CEEF17      83EC 80       sub esp,FFFFFF80
00CEEF1A  ^   E9 C66AFFFF   jmp 9cee5522a7ca2bfca7cd3d9daba23e9a30deb   jump to entry point executable
```

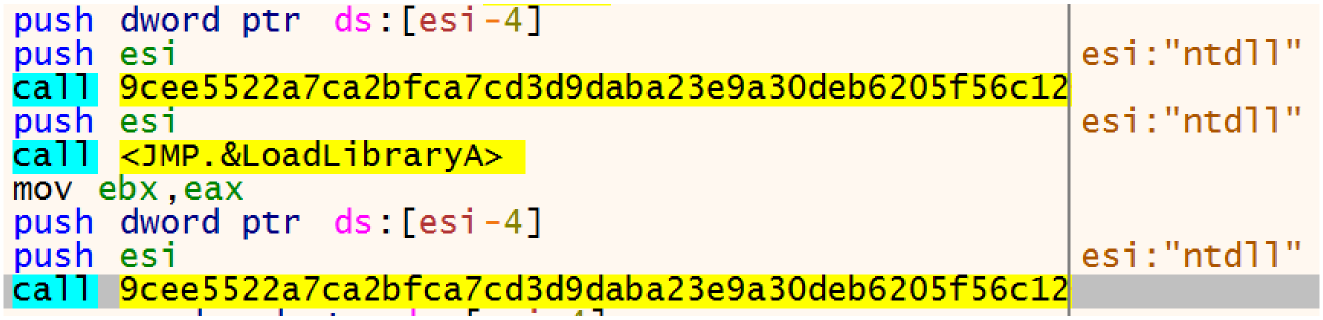


```
00CE59E5  >   E8 17FEFFFF     call 9cee5522a7ca2bfca7cd3d9daba23e9a30de
00CE59EA      6A 00           push 0
00CE59EC      E8 00000000     call <JMP.&ExitProcess>                     call $0
00CE59F1  v   FF25 0C60CE00   jmp dword ptr ds:[<&ExitProcess> ]          JMP.&ExitProcess
00CE59F7  v   FF25 0060CE00   jmp dword ptr ds:[<&GetModuleHandleA> ]     JMP.&GetModuleHandleA
00CE59FD  v   FF25 0460CE00   jmp dword ptr ds:[<&GetProcAddress> ]       JMP.&GetProcAddress
00CE5A03  v   FF25 0860CE00   jmp dword ptr ds:[<&LoadLibraryA> ]         JMP.&LoadLibraryA
```

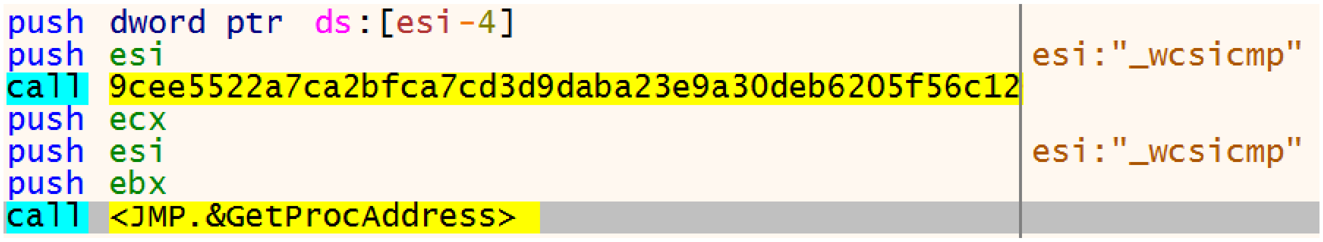
Once the executable is unpacked, we can analyze the ransomware

Anti-analysis

To make static analysis harder the ransomware resolves DLL’s and API calls dynamically using `LoadLibrary`, `GetProcAddress` and 2 custom functions shown below. In this screenshot, the address of `_wcsicmp` is resolved in memory.



```
push dword ptr  ds:[esi-4]
push esi                                                  esi:"ntdll"
call 9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205f56c12
push esi                                                  esi:"ntdll"
call <JMP.&LoadLibraryA>
mov  ebx,eax
push dword ptr  ds:[esi-4]
push esi                                                  esi:"ntdll"
call 9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205f56c12
```



```
push dword ptr  ds:[esi-4]
push esi                                                  esi:"_wcsicmp"
call 9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205f56c12
push ecx
push esi                                                  esi:"_wcsicmp"
push ebx
call <JMP.&GetProcAddress>
```

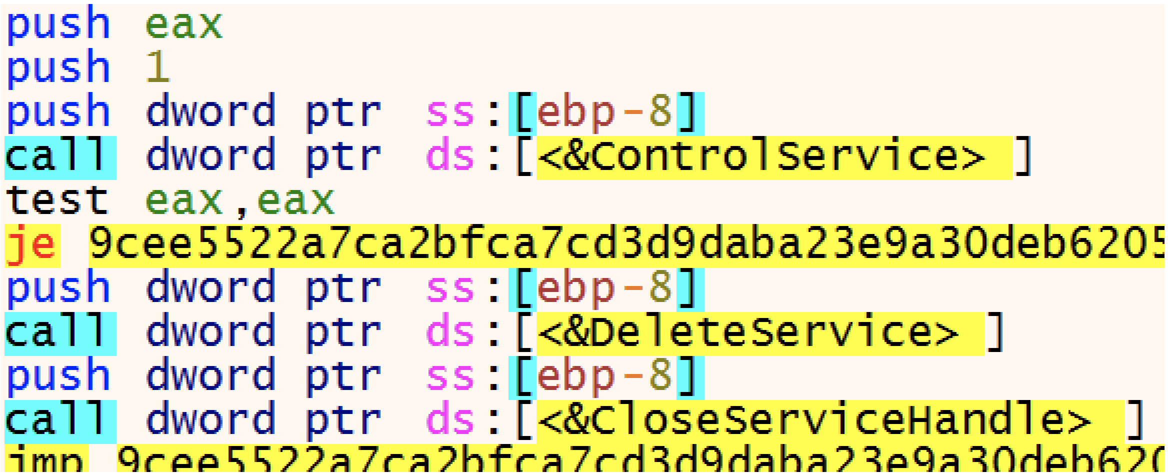
Preparation

The mutex `Global\\3e93e49583d6401ba148cd68d1f84af7` is created to make sure only one copy of the ransomware is running, otherwise the ransomware exits. This is done based on the name of the executable. Then `SetThreadExecutionState` is called to force the system to be in the working state by resetting the system idle timer.

Services

To make sure certain services are not running the following services are stopped using `ControlService - SERVICE_CONTROL_STOP` and `DeleteService`. Deleting a service is not useful if an organization pays the ransom and wants to go back into production quickly. As a system administrator, I wouldn't be happy about this.

- vss
- sql
- svc$
- memtas
- mepocs
- sophos
- veeam
- backup



```
push eax
push 1
push dword ptr  ss:[ebp-8]
call dword ptr  ds:[<&ControlService> ]
test eax,eax
je 9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205
push dword ptr  ss:[ebp-8]
call dword ptr  ds:[<&DeleteService> ]
push dword ptr  ss:[ebp-8]
call dword ptr  ds:[<&CloseServiceHandle> ]
jmp 9cee5522a7ca2bfca7cd3d9daba23e9a30deb620
```

Shadow Copies

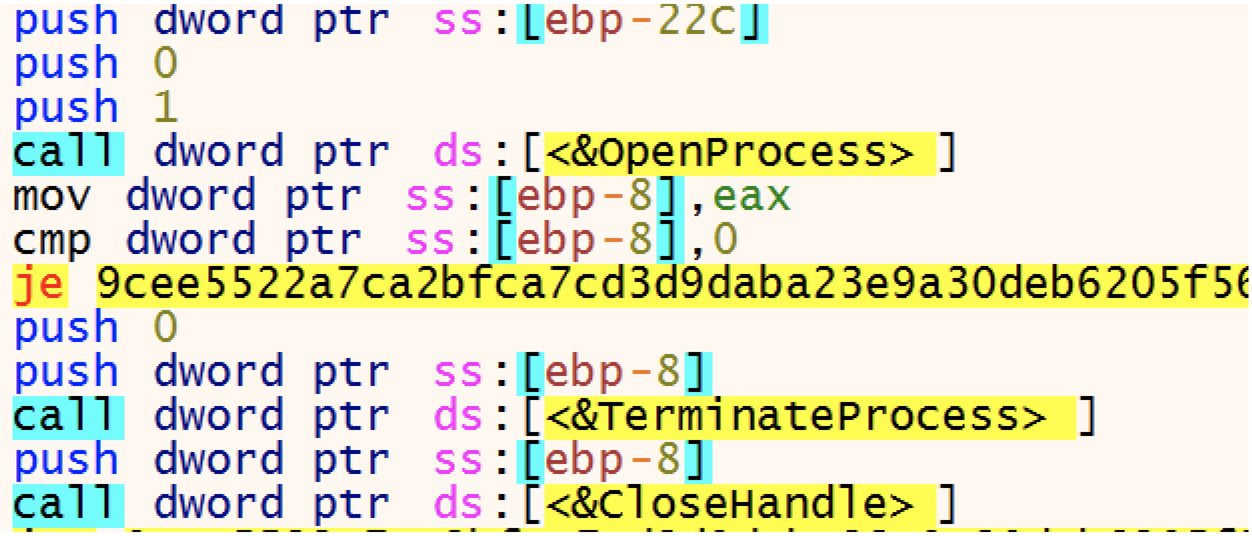
Using `CreateProcessW` the following Powershell script is executed which deletes Shadow Volume Copies.

```
powershell -ep bypass -c \"(0..61)|%{$s+=[char][byte]
('0x'+'4765742D576D694F626A6563742057696E33325F536861646F77636F7079207C20466F72456163
 $s\"
```

> When deobfuscated, we can see that this PowerShell command is used to delete Shadow Volume Copies on the machine before encrypting it.
>
> ```
> Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}
> ```
>
> https://www.bleepingcomputer.com/news/security/darkside-new-targeted-ransomware-demands-million-dollar-ransoms/amp/

Processes

To make sure certain processes are not running a list of processes are terminated (https://pastebin.com/WWSQxhcq.



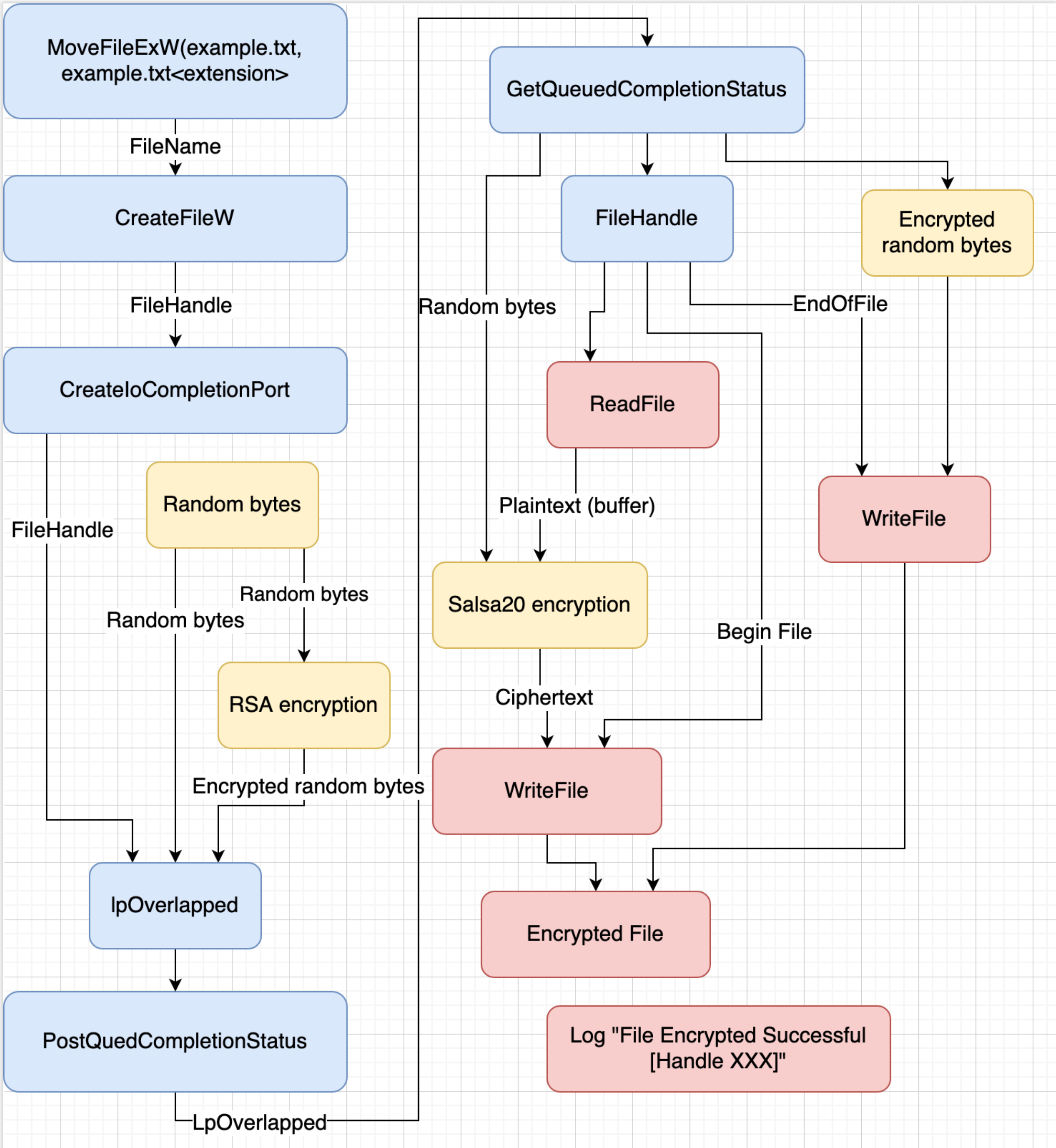
```
push dword ptr  ss:[ebp-22C]
push 0
push 1
call dword ptr  ds:[<&OpenProcess> ]
mov dword ptr  ss:[ebp-8],eax
cmp dword ptr  ss:[ebp-8],0
je 9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205f5
push 0
push dword ptr  ss:[ebp-8]
call dword ptr  ds:[<&TerminateProcess> ]
push dword ptr  ss:[ebp-8]
call dword ptr  ds:[<&CloseHandle> ]
```

Encryption

The encryption routine skips a few files, file extensions and directories (https://pastebin.com/WWSQxhcq).

Encryption flowchart

The encryption routine of the ransomware is shown below.



Debugging mode

I don't know why but it seems the authors have forgotten to disable debugging functionality in their code or maybe they are using this to verify that the files are encrypted. (XXX = file name). This file was in the same directory as the executable.

```
[INF] Start Encrypting All Files
[INF] Emptying Recycle Bin
[INF] Uninstalling Services
[INF] Deleting Shadow Copies
[INF] Terminating Processes
[INF] Encrypt Mode - FAST
[INF] Encrypting Local Disks
[INF] Started 8 I/O Workers
[INF] Start Encrypt [Handle 492] \\?\C:\XXX
[INF] File Encrypted Successful [Handle 492]
[INF] Start Encrypt [Handle 640] \\?\C:\XXX
[INF] File Encrypted Successful [Handle 640]
[INF] Start Encrypt [Handle 640] \\?\C:\XXX
[...]
```

IOC

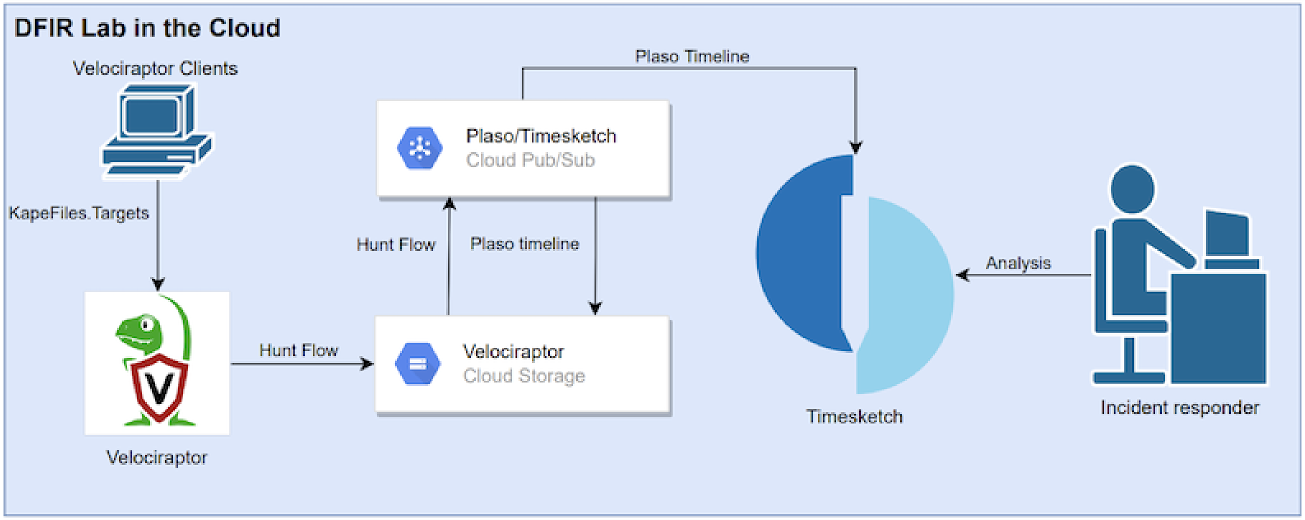
SHA256 - `9cee5522a7ca2bfca7cd3d9daba23e9a30deb6205f56c12045839075f7627297`

References

https://www.bleepingcomputer.com/news/security/darkside-new-targeted-ransomware-demands-million-dollar-ransoms/amp/

https://tria.ge/200828-r31s5nvvm2/behavioral1

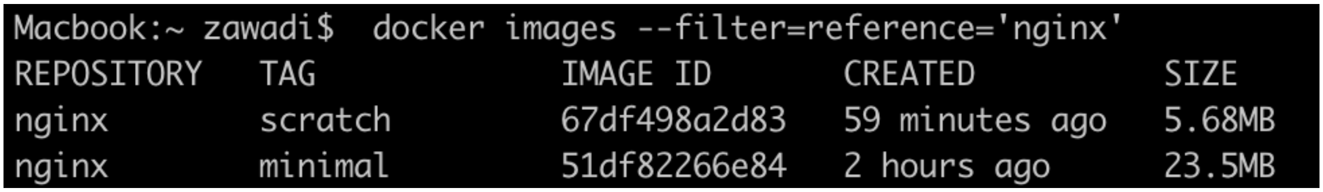
You May Also Enjoy



Automating DFIR using Cloud services

March 28, 2022 7 minute read

TL;DR: The DFIR lab can automate the processing of Plaso timelines to Timesketch using Velociraptor and Google Cloud services

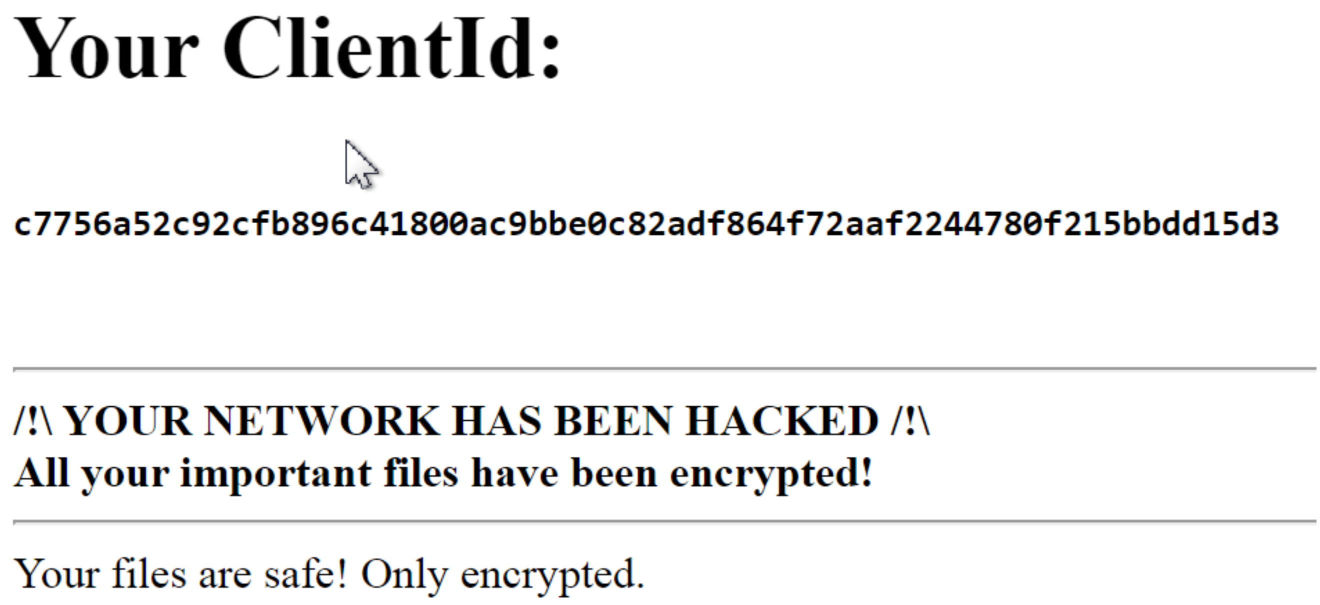


```
Macbook:~ zawadi$  docker images --filter=reference='nginx'
REPOSITORY    TAG            IMAGE ID       CREATED           SIZE
nginx         scratch        67df498a2d83   59 minutes ago    5.68MB
nginx         minimal        51df82266e84   2 hours ago       23.5MB
```

Using Docker images from scratch

January 12, 2022 5 minute read

The last blog post about Docker was about using non-root Docker containers and why this is safer. This time I want to go a step further and explain what I th...



Your ClientId:

c7756a52c92cfb896c41800ac9bbe0c82adf864f72aaf2244780f215bbdd15d3

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

Your files are safe! Only encrypted.

Mount Locker ransomware analysis

November 26, 2020 8 minute read

This blog post will explain how the ransomware called Mount Locker works. For encryption, Mount Locker uses Chacha20 to encrypt files and RSA-2048 to encrypt...

Preparation Tips for CKAD

November 12, 2020 3 minute read

5 days ago, I took the CKAD exam and passed. So I decided to write this short blog post about my experience and to share some tips that helped me pass it. Ev...