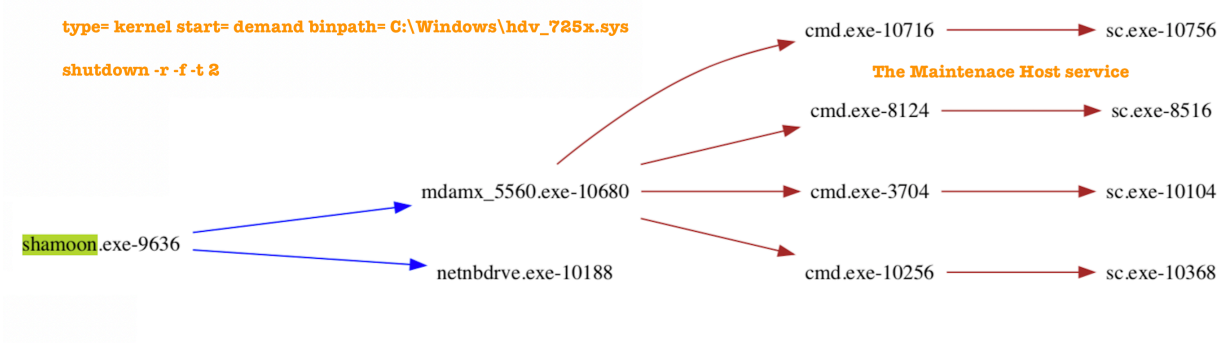
Shamoon

UDURRANI

I am not getting into all the details this time. I just want to cover the flow and IOC's of this payload. If you want to look at the previous shamoon report, please google Robert Falcone's report.

Here is the flow, I named the payload as shamoon.exe



As you can see above, it drops 2 payloads. The names are given randomly to these payloads. Names are within the payload (encrypted). Here is the list of all names

mmotsml
ql40xx23
netavpntt
setupapiev23
ks25
mdmtexasr
prntsa02
mdmnttr
mdmnovb
MSDTCB
netpgm1
prN2rC00b
mdmcxavu
mdmbr43
wOaky032
mdm2mcom
hdBaudio
dc21G4vm

winntusb3
mdm2bt2mdm
wvm2bus2video
faxcnnumber
mdmmhzell
mdmgl908
netnbdrve
prnod802
netrndiscnt
netrtl42l
mdmadccnt
prnca00
bth2bht_ibv32
cxfalcon_ibL32
mdmsupr30
digitalmediadevicectl
mdmetech2dmv
netb57vxx
winwsdprint
prnkwy005
composite005
mdmar1_ibv32
prnle444
kscaptur_ibv32
mdmzyxlga
usbvideob
input_ibv48
prnok002_ibv
averfx2swtvZ
wpdmtp_ibv32
mdmti_ibv32
printupg_ibv32
wiabr788
_wialx002
__wiaca00a
tsprint_ibv
acpipmi2z
prnlx00ctl
prngt6_4
arcx6u0
_tdibth
prncaz90x
mdmgcs_8
mdmusrk1g5
netbxndxlg2
prnsv0_56
af0038bdax
averfix2h826d_noaverir
megasasop
hidirkbdmvs2
vsmxraid
mdamx_5560
wiacnt7001

Stage 1 payload has the kill-time set to **20171272351 i.e. 2017-12-7 23:51.** This will make it execute on pretty much any machine.

File **netnbdrve.exe** is used for C2 channel, while **mdamx_5560** is used to drop the driver, wipe and shut down the machine. Payload, used for C2 channel has all the code to communicate to a C2 but its not really active. It does the following:

- Opens an handle to C:\Windows\inf**averbh_noav.pnf** via CreateFileW(). If the return value is **INVALID_HANDLE_VALUE**, it sleeps for another 5000 milliSeconds. If it returns a valid handle, it will use

```
BOOL ReadFile(
  HANDLE       hFile,
  LPVOID       lpBuffer,
  DWORD        nNumberOfBytesToRead,
  LPDWORD      lpNumberOfBytesRead,
  LPOVERLAPPED lpOverlapped
);
```

So e.g. if the handle returned from **CreateFileW**() is **0x00000154**, that will be provided to **ReadFile**(). **Hanlde** is a kernel object. Its just a pointer in OS memory space that you are not allowed to use directly. File **averbh_noav.pnf** keeps the wiping status. This loop continues every 5 seconds i.e. 5000 milliSeconds. In this particular payload this file is useless.

File **mdamx_5560** is used to conduct the wiper activity. Here are the commands.

```
mdamx_5560.exe 1
C:\Windows\system32\cmd.exe /c sc stop hdv_725x 2>&1 >nul
sc  stop hdv_725x
C:\Windows\system32\cmd.exe /c sc delete hdv_725x 2>&1 >nul
sc  delete hdv_725x
C:\Windows\system32\cmd.exe /c sc create hdv_725x type= kernel start= demand binpath=
C:\Windows\hdv_725x.sys 2>&1 >nul
C:\Windows\system32\netnbdrve.exe 1
sc  create hdv_725x type= kernel start= demand binpath= C:\Windows\hdv_725x.sys
C:\Windows\system32\cmd.exe /c sc start hdv_725x 2>&1 >nul
sc  start hdv_725x
```

And eventually:

```
C:\Windows\system32\cmd.exe /c shutdown -r -f -t 2
```

There are 2 drivers dropped:

```
MaintenaceSrv
hdv_725x
```

```
SERVICE_NAME: MaintenaceSrv
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                             (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

```
SERVICE_NAME: hdv_725x
        TYPE               : 1  KERNEL_DRIVER
```

The .sys file is called **hdv_725x.sys**. Payload **mdamx_5560** will first delete this sys file CreateProcessA ("C:\Windows\system32\cmd.exe", "C:\Windows\system32\cmd.exe /c sc delete hdv_725x 2>&1 >nul", NULL, NULL, TRUE, 0, NULL, NULL, …)

And then create it

sc create hdv_725x type= kernel start= demand binpath= C:\Windows\hdv_725x.sys

File hdv_725x.sys is the actual driver used, here is the actual hash

92fff1d754faab445e90651dfb0ded4d

Its using version **2.1.27.106**, name **elrawdsk.sys.**

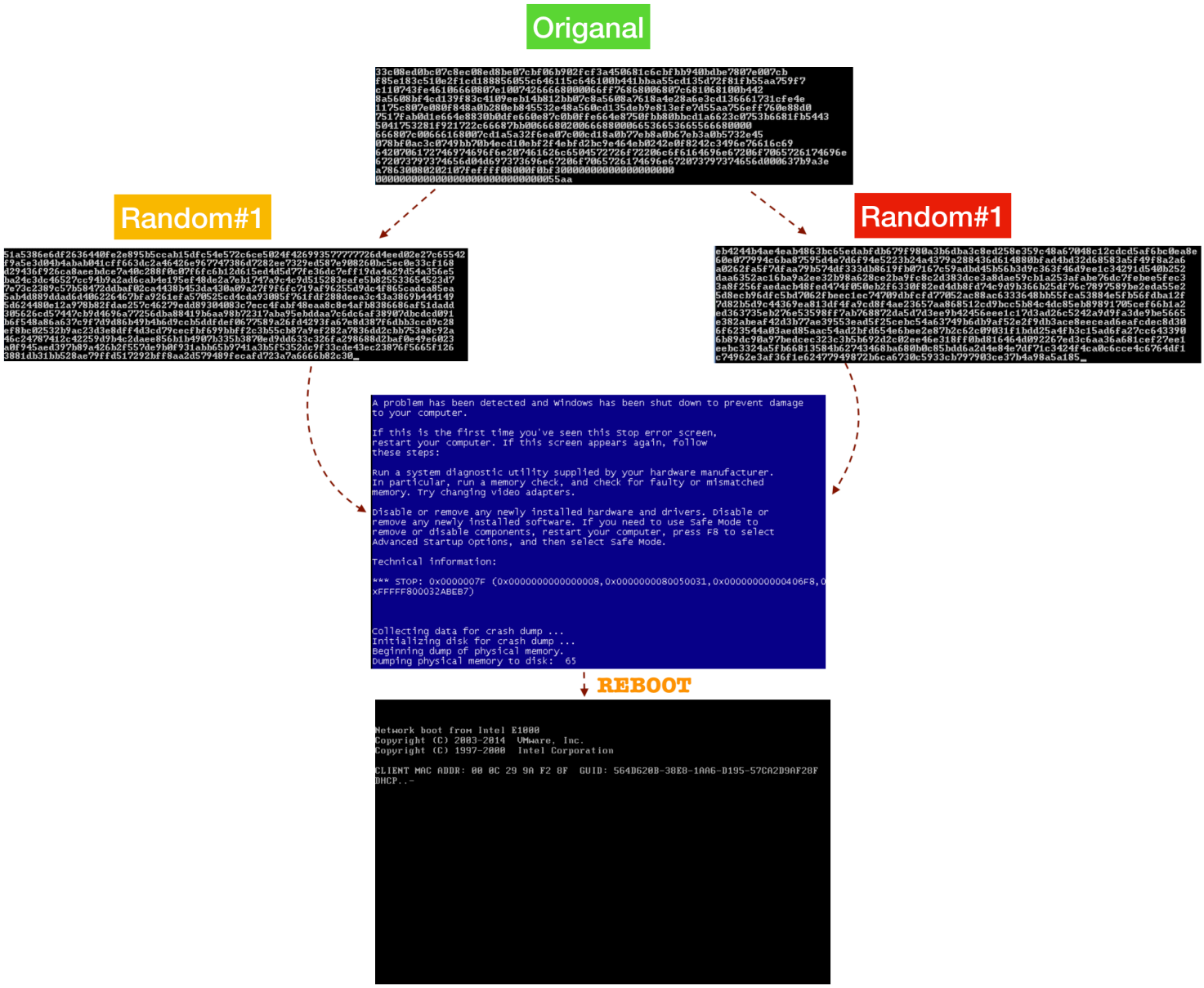
This driver Allows write access to files and raw disk sectors for user mode applications
Installation of the driver requires ***admin privileges***. Remember, this is a signed sys file.

```
++++++++++++++++++++ X ++++++++++++++++++++
[0032FAF8]->    (null)
[0032FB0C]->    GlobalSign ObjectSign CA
[0032FB10]->    EldoS Corporation
[0032FAFC]->    (null)
[0032FB00]->    (null)
[00FE2180]->    01 00 00 00 00 01 26 1d ec 28 f7
```

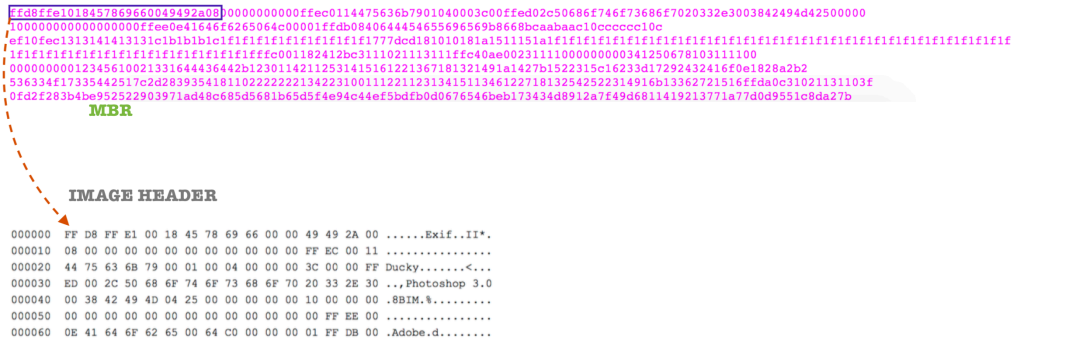
Once the driver is installed, the driver can provide the ability to write and read sector by sector, access to locked files and access to rawDisk. First stage uses *NtCreateFile* to drop the sysFile.

NtCreateFile (PHANDLE, FILE_READ_ATTRIBUTES | GENERIC_WRITE | SYNCHRONIZE, ObjectAttributes, IoStatusBlock, NULL, FILE_ATTRIBUTE_NORMAL, FILE_SHARE_READ, FILE_OVERWRITE_IF, FILE_NON_DIRECTORY_FILE | FILE_SYNCHRONOUS_IO_NONALERT, NULL, 0)

Let's look at the MBR overwrite in the following picture. The payload overwrites the MBR with random data. All the files are encrypted as well. Eldos driver makes sure that the files with already open handles get infected as well. This will cause a blue screen followed by No OS found screen. There is no message or image embedded in there. Its all about destruction.



In the past, e.g. shamoon 2, MBR was overwritten with an image. Here is how the MBR looked like after the overwrite. In the following picture, MBR is shown in color and the header information is shown in gray.



Encryption:

Yes, its encrypting the MBR and files with a key called **key8854321.pub**. Its a public key. Here is the hash

41f8cd9ac3fb6b1771177e5770537518

Conclusion:

Payload has kill-time set to 2017, this means it will run on any machine as the kill-time is already passed. Think of kill-time as a logic-bomb that will detonate when time >= (time provide by the attacker). Previous shamoon was designed for a specific victim. This means it wont propagate on any other network, since the credentials are hardcoded within the payload. On the other hand, this payload can work independently but it does need another stage that will carry it to other machines i.e. propagation / lateral movement.

IOC's

b41f586fc9c95c66f0967f1592641a85
887c614608e7cd9a691858caf468c28f
41f8cd9ac3fb6b1771177e5770537518
92fff1d754faab445e90651dfb0ded4d
de07c4ac94a50663851e5dabe6e50d1f
a4a9100413ebba59cab785d506448093

You can get the driver names from above.
For info on previous shamoon go to

http://udurrani.com/0fff/s00.pdf
http://udurrani.com/0fff/h8.html
https://www.youtube.com/watch?v=Bg6ia1FWrqw&t=2s
https://www.youtube.com/watch?v=UEOghELxfmo