volatility memory analysis concepts based on scenarios

Some abbreviations

file=collected memory file,dump,vmem file

1. to check the mem version/profile
use command :-

python vol.py -f mem imageinfo
Or
python vol.py -f mem kdbgscan

2.after getting profile work start

scenario one :-

If any malware has alert is there but don't know what is happening and how to find out .

there are many plugins in volatility which might be useful(pslist,pstree etc)

a. python vol.py -f file --profile="prfile" pslist

will gather all the processes are executed check for any malicious/unknown process over here . if found
something suspicious to recheck check on the pstree how it has came into your machine

b. python vol.py -f file --profile="prfile" pstree

will give us the tree structure of the process eg. if any malicious is downloaded from outlook it can be seen from here
like outlook ---> process

c. python vol.py -f file --profile="prfile" psscan --output=dot --output-file=infected.dot
will give us diagramatic graph for the process . To see the Dot file it can open in graphwiz,xdot

d. python vol.py -f file --profile="prfile" cmdline -p "PID"
Will give us the path that file is stored or how/cmdline command has executed that proccess.

e. python vol.py -f file --profile="prfile" procdump -p "PID" -D dump/
will download the executable and do any reversing if needed.

Scenario Two :-

if any CNC Communication alert has triggered from the machine then how to find out.

a. python vol.py -f file --profile="profile" netscan
will give us the confirmation about communication and will provide info what process is making CNC Communication

b. python vol.py -f file --profile="prfile" yarascan -Y "IPADDRESS"
if unable to get the process from there then use yara scan and get the process

YARASCAN WILL CHECK THE KEYWORD IN WHOLE MACHINE AND GIVE US CONSTRUCTIVE OUTPUT
AFTER GETTING THE PROCESS SAME PROCESS AND IN CASE OF MALWARE
BUT IF THAT COMMUNICATION IS DONE BY DLL THEN HOW TO WORK

c. python vol.py -f file --profile="prfile" dlllist -p "PID"
will list the dll which is doing cnc communication and will give us base address like (0x0x00000)

d. python vol.py -f file --profile="prfile" dlldump -p "PID" -b "BASEADDRESS" -D dump/
will download the file
e. python vol.py -f file --profile="prfile" pstree
will give us the tree structure of the process

Scenario 3

To find the malicious things from backends.

1. python vol.py -f file --profile="prfile" callbacks
will give you the malicious driver if there is any

2. python vol.py -f file --profile="prfile" modscan | grep -i "address"
will give the name of drivers

3. python vol.py -f file --profile="prfile" devicetree
will give the driver info

4. python vol.py -f file --profile="prfile" handles -t File | grep -i "filename"
will give us the address and work on it

5. python vol.py -f file --profile="prfile" cmdline -p "PID"

Will give us the path

Miscellaneous flags

1. check for any registry hives
 python vol.py -f file --profile="prfile" hivelist

2. Download registry
  python vol.py -f file --profile="prfile" dumpregistry -D dump/

3.to check anything in registry
  strings -f -a -el * | grep -1 "filecheck"

Mutex :- It is a process when any process get executed into registry at two location but only running one at a time .

to check Mutex

 python vol.py -f file --profile="prfile" handles -p "PID" -t mutant