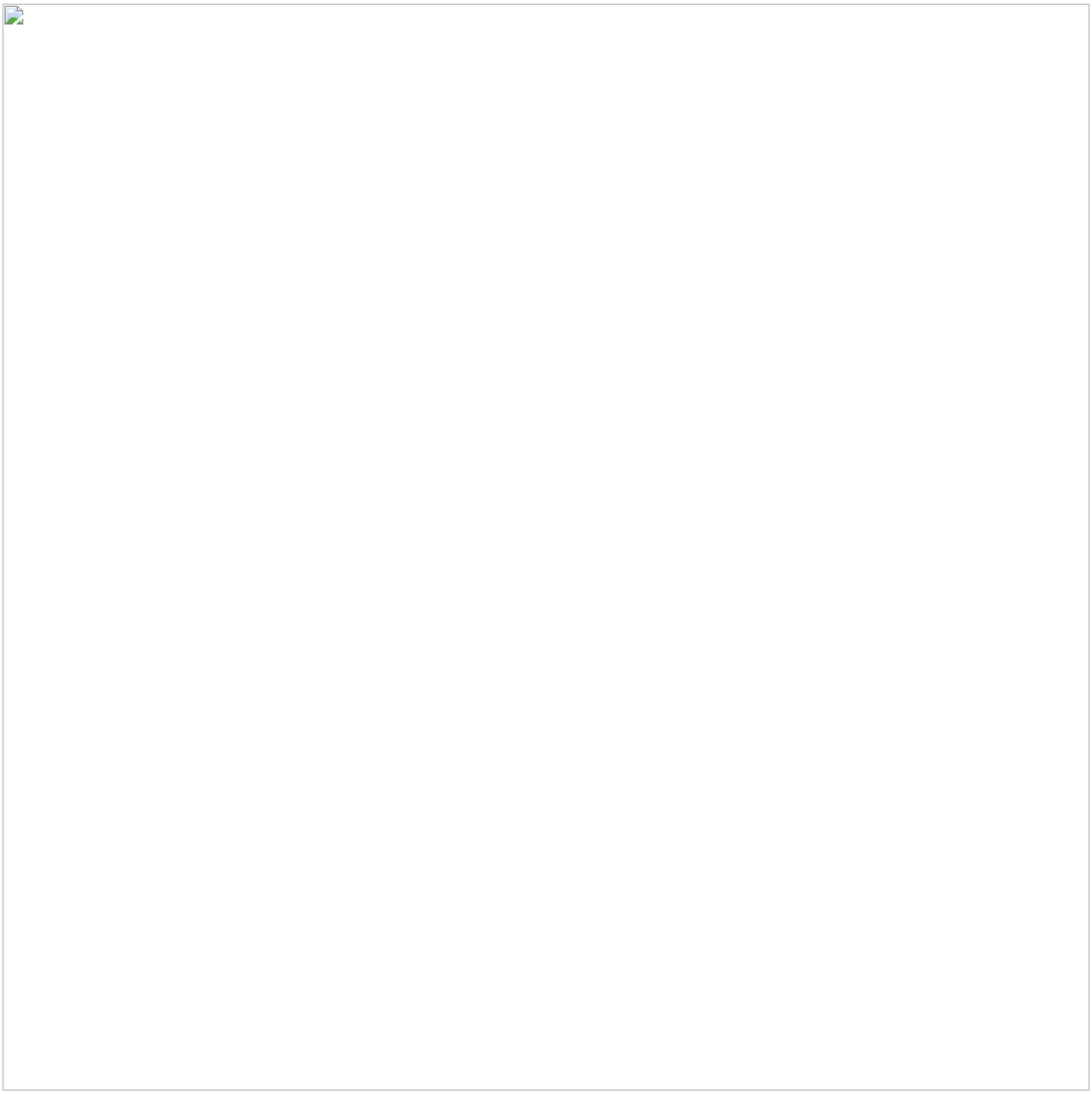
Kerberos Delegation: A Reference Overview

csandker.io/2020/02/15/KerberosDelegationAReferenceOverview.html

February 15, 2020



15 Feb 2020 (Last Updated: 20 Aug 2021)

> This post is based on Kerberos Delegation: A Wrap Up make sure to read this first for an introduction to the used terms.

Delegation allows a **server** application to **impersonate a client** when the server connects to other network resources.
In other words: Delegation specifies the client’s action to authorize a server in order to allow this server to impersonate itself (the client).

There are 3 Types of Kerberos Delegation:

- Unconstrained
- Constrained
- Resource Based Constrained

QUICK REFERENCE

TRUSTED_FOR_DELEGATION
All users that have the UserAccountControl attribute 'TRUSTED_FOR_DELEGATION' (524288) set are allowed for unconstrained delegation. Meaning: If you connect to a service run by such a user your TGT will be passed to this service.
→ This is is **Unconstrained Delegation**

Find users with 'TRUSTED_FOR_DELEGATION' with:

```
([adsisearcher]'(userAccountControl:1.2.840.113556.1.4.803:=524288)').FindAll()
```

ms-DS-Allowed-To-Delegate-To
All users that have a value specified (specific service on specific host) within the 'ms-DS-Allowed-To-Delegate-To' account attribute are allowed to delegate to the specified service on the specified host (on behalf of a user that connected to a service run by this user). Users with this attribute can request a service ticket to the specific service using S4U2Proxy.
→ This is is **Constrained Delegation**

Find users with the 'ms-DS-Allowed-To-Delegate-To' object attribute:

```
([adsisearcher]"(msds-allowedtodelegateto=*)").FindAll() | %{$_.Properties['msds-allowedtodelegateto']}
```

TRUSTED_TO_AUTH_FOR_DELEGATION
All users that have the UserAccountControl attribute 'TRUSTED_TO_AUTH_FOR_DELEGATION' (16777216) set are allowed to request a service ticket to itself for any arbitrary user (using S4U2Self).
These users should have a value specified within the 'ms-DS-Allowed-To-Delegate-To' account attribute as well. Combining both means: These users are allowed to delegate any arbitrary user to the specified service on the specified host.
→ This is is **Constrained Delegation with allowed Protocol Tranisition** (S4U2Self)

Find users with the 'TRUSTED_TO_AUTH_FOR_DELEGATION' UserAccountControl attribute:

```
([adsisearcher]'(userAccountControl:1.2.840.113556.1.4.803:=16777216)').FindAll()
```

msDS-AllowedToActOnBehalfOfOtherIdentity
All users that have a value specified (specific user Identity) within their 'msDS-AllowedToActOnBehalfOfOtherIdentity' account attribute allow the specified identity to delegate to services run by them.
→ This is is **Resource Based Constrained Delegation**

Find users with the 'msds-AllowedToActOnBehalfOfOtherIdentity' attribute set:

```
([adsisearcher]"(msds-AllowedToActOnBehalfOfOtherIdentity=*)").FindAll()
```

ATTACK REFERENCE

Attacker compromised UserAccount with attribute **TRUSTED_FOR_DELEGATION** set:
→ You will receive a TGT from every user connecting to one of your services.

For simplicity assuming the compromised user account is a Computer Account.
Use @tifkin_'s SpoolSample.exe (also known for being the exploit for "The PrinterBug") to force the DomainController Computer account to connect to you, steal it's TGT and DCSync your way to DomainAdmin.

Attacker compromised UserAccount with attribute **ms-DS-Allowed-To-Delegate-To** set:
→ You will be able to get a service ticket to the service specified in the attribute for any user connecting to you.

For simplicity assuming compromised the user account is a Computer Account
Use @tifkin_'s SpoolSample.exe (also known for being the exploit for "The PrinterBug") to force the DomainController Computer account to connect to you, get a service ticket for the DomainController Computer account to the specified service (e.g. for the 'cifs' service) using **S4U2Proxy**.
→ You gained access to the specified service (e.g. CIFS on the DC)

Attacker compromised UserAccount with attribute **TRUSTED_TO_AUTH_FOR_DELEGATION** set:
→ You will be able to get a service ticket to yourself for any user

For simplicity assuming compromised the user account is a Computer Account.
Get a service ticket for the DomainController Computer account to your services (e.g. 'cfis') by using **S4U2Self**
→ You already got access to your own services (as you own them), therefore this in itself does not get you anything

If the compromised user account also has the ms-DS-Allowed-To-Delegate-To set (which is likely), you can gain access to the service specified in the ms-DS-Allowed-To-Delegate-To attribute (e.g. 'cifs' on some host) by using the previously obtained Domain Controller accounts service ticket with **S4U2Proxy**.
→ You gained access to the specified service (e.g. 'cifs' on some host) without any user connecting to you (aka. no SpoolSample.exe)

Attacker compromised UserAccount with attribute **msDS-AllowedToActOnBehalfOfOtherIdentity** set:
→ You can allow any user to get a service ticket to any of your services

For simplicity assuming compromised the user account is a Computer Account.
If you compromised the entire user account (e.g. got the user's password) of the account this is barely of any use (as you already have access to the machine).
But if you have gained "only" write access to that user account or this attribute, you can gain access to the account by specifying your own user account. This allows your user account to obtain a service ticket to the account by using **S4U2Self** and **S4U2Proxy**

Note: In this case your user account (which you allow to delegate from) does not need to have the TRUSTED_TO_AUTH_FOR_DELEGATION UserAccountControl flag. In Resource Based Constrained Delegation the service that you created with S4U2Self, which you then used in S4U2Proxy does not need to have the FORWARDABLE flag (in contrast to classic Constrained Delegation).

Other Posts