Active Directory

Principles

SPN

- Service principal name
- Unique identifier used in windows environments to link a specific network service to the Active Directory account running that service

Reddit src

An SPN is how your computer identifies a service on a network. That service could a be network protocol like HTTP or SMB. SPN and SPN binding are the same thing. Or rather, the binding is the literal registration of the SPN to the service account. See below.

When you open a file share to \\othermachine\share Windows is asking to get a ticket to the SPN 'cifs/othermachine'. The SMB stack knows enough to say it's service is 'cifs' and the machine is 'othermachine', so the SPN should be 'cifs/othermachine'. If you open a web browser it'll do the same thing and ask for an SPN of 'http/othermachine'.

Now the client stack (say SMB) has asked the Windows security system for a ticket to that SPN. The security system asks the Kerberos stack, and the Kerberos stack fires a request off to the Domain Controller. The Domain Controller looks up the service account in AD by the requested SPN and returns a ticket to the client encrypted to the service account password.

The client receives the encrypted ticket, fires it off to the service on the other machine, and the other machine decrypts the ticket because it has it's own password. It's that simple.

So the SPN identifies the service so AD can know what service account it needs to find and by extension which password it should encrypt the ticket to. If the SPN isn't found then the DC returns an error, and if the SPN is registered on a service account different than the service account running the service then the decryption will fail because the passwords don't match.

Windows doesn't really care about SPNs on the receiving side. Imagine you're the server. You receive an encrypted ticket. If you can decrypt it then that means it was issued by someone that knows your secret. If they know the secret then they can just fake any SPN, so there's no real need to verify the SPN. Therefore Windows really only cares if the passwords match.

That's why SPNs need to be unique per service account. You can have as many SPNs as you want associated to a single service account, but you can't have one SPN associated to more than one account. When a service is running as the local system or network service they are operating as the machine account. So SMB for instance is registered on the computer object in AD as cifs/computername.

The mapping of Windows Services to SPNs is a bit lopsided. There's only a handful of SPNs on any given computer object, but that's okay because there's a special SPN called host/computername that

is a catch-all. There's a mapping of 50 or so service types mapped to host so when you ask for say fax/computername, AD will treat that as a search 'find me fax/computername OR find me host/computername'. There's an official list on docs somewhere (can't find it), but here's a mapping I created from that official list. All of this lopsidedness works because the local services are running as local system or network service, and therefore use the computer account.

See here for a lot more specifics about the protocol bits: Kerberos Explained in a Little Too Much Detail (syfuhs.net)

Now all of this is well and good, but there's a very specific case where this fails: when the SPN can't be found. AD returns an error to the client so the client needs to decide what to do with it. Usually it falls back to NTLM. This is why NTLM doesn't provide server authentication. That is, because AD cannot guarantee the thing you're connecting to matches any service it knows about. That is also why things often “just work” right up until you need to enable delegation, which only works with Kerberos.