Domain Trusts

The “Trusts you might have missed”

Domain Trusts

- Trusts allow domains to form inter-connected relationships
  - All a trust does is **link up the authentication systems** of two domains and allows authentication traffic to flow between them
  - This is done by each domain negotiating an “inter-realm trust key” that can relay Kerberos referrals
- Communications in the trust work via a system of referrals:
  - If the SPN being requested resides outside of the primary domain, the DC issues a referral to the forest KDC (or trusted domain KDC)
  - Access is passed around w/ inter-realm TGTs signed by the inter-realm key (not the krbtgt account!)
- Tons more information:
  - http://www.harmj0y.net/blog/redteaming/a-guide-to-attacking-domain-trusts/

Trust Types

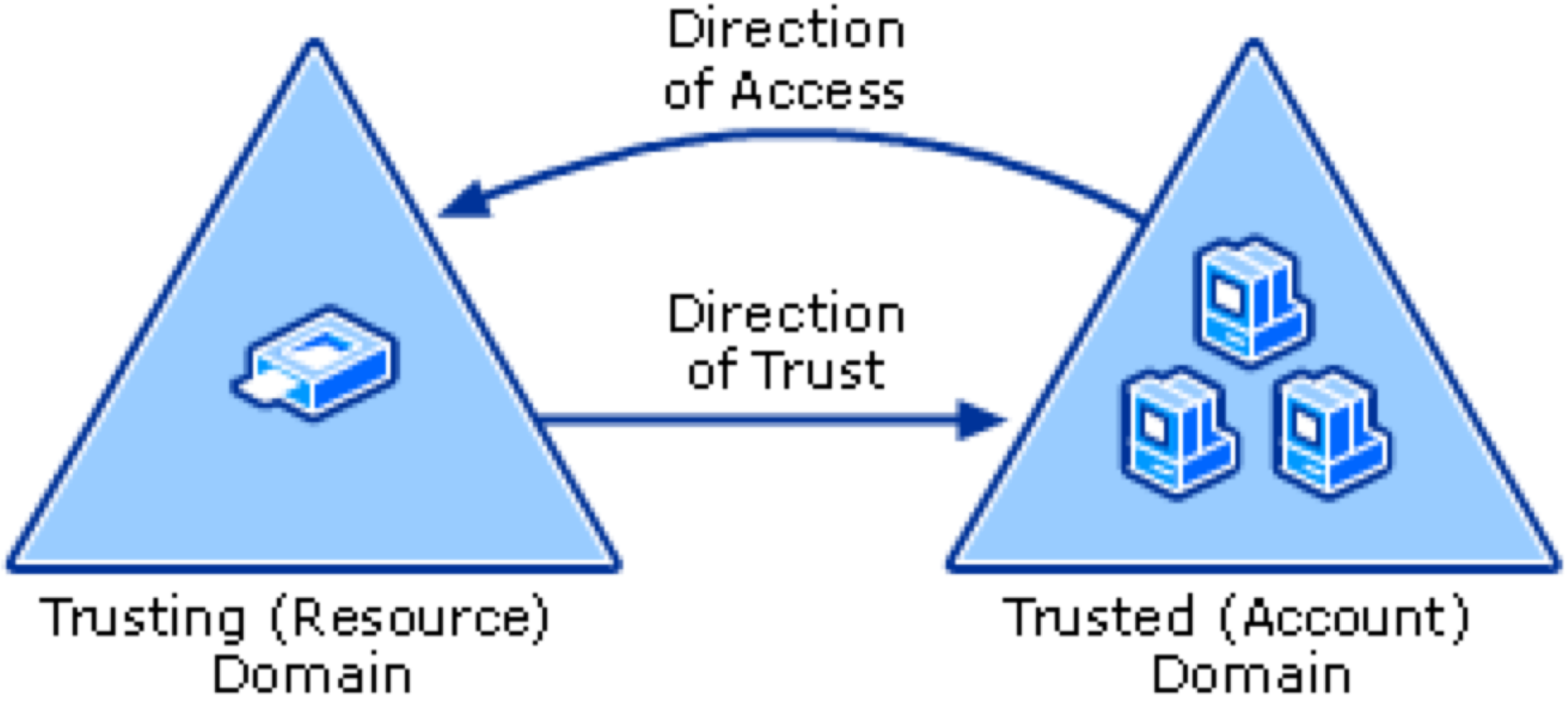
- General types:
  - **Parent/Child** - part of the same forest- a child domain retains an implicit two-way transitive trust with its parent, “intra-forest”
  - **Cross-link -** “shortcut” between child domains to improve logon times
  - **External -** non-transitive, created between disparate domains
  - **Tree-root -** implicit two-way transitive trust between the forest root domain and the new tree root you’re adding, “intra-forest”
  - **Forest -** transitive, established between two forests
- Directions/transitivity:
  - **One-way** - one domain trusts the other
  - **Two-way** - both domains trust each other (2x one-way trusts)
  - **Transitive**- domain A trusts Domain B and Domain B trusts Domain C, so Domain A trusts Domain C

Trust Types; redux

- From a security perspective, all we really care about is whether a domain trust exists *within* a forest or is *external* to a forest

- **The forest is the trust boundary, not the domain!**
  - *Intra*-forest trusts (**parent/child**, **tree-root**, **cross-link**) have an attack that allows for the abuse of sidHistory to elevate from any child domain in a forest the forest root domain
  - *Inter*-forest trusts (**external**, **forest**) have a security protection called “SID Filtering” that prevents this particular type of abuse

Trust Direction



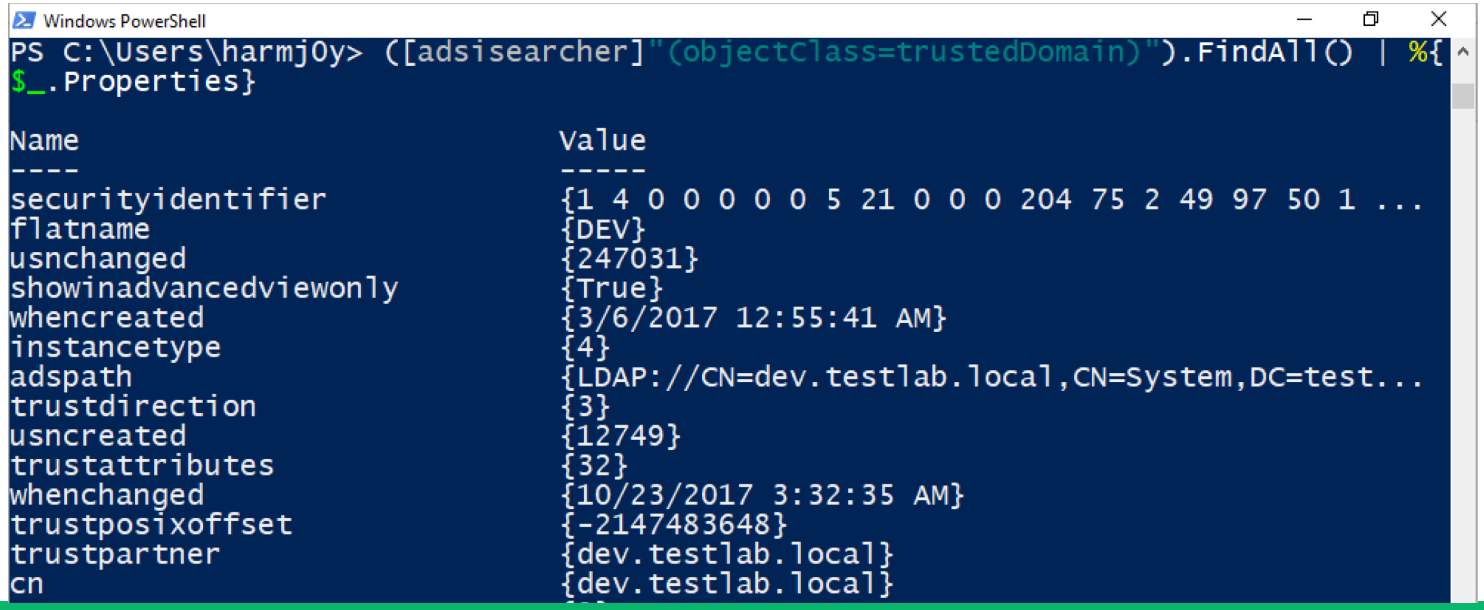
Manual Trust Enumeration

- Using **[System.DirectoryServices.ActiveDirectory]**:
  - **[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().GetAllTrustRelationships()**
  - **[System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().GetAllTrustRelationships()**
  - PowerView: **Get-DomainTrust -NET** / **Get-ForestTrust**

- Using Win32 API calls:
  - **DsEnumerateDomainTrusts() / DsGetForestTrustInformationW()**
  - **nltest /domain_trusts [/server:secondary.dev.testlab.local]**
  - PowerView: **Get-DomainTrust -API**

Trusted Domain Objects

- When a domain establishes a trust with another domain, the foreign domain is stored as a “trusted domain object” in AD
  - LDAP filter: **(objectClass=trustedDomain)**



LDAP trustedDomain - **TrustType**

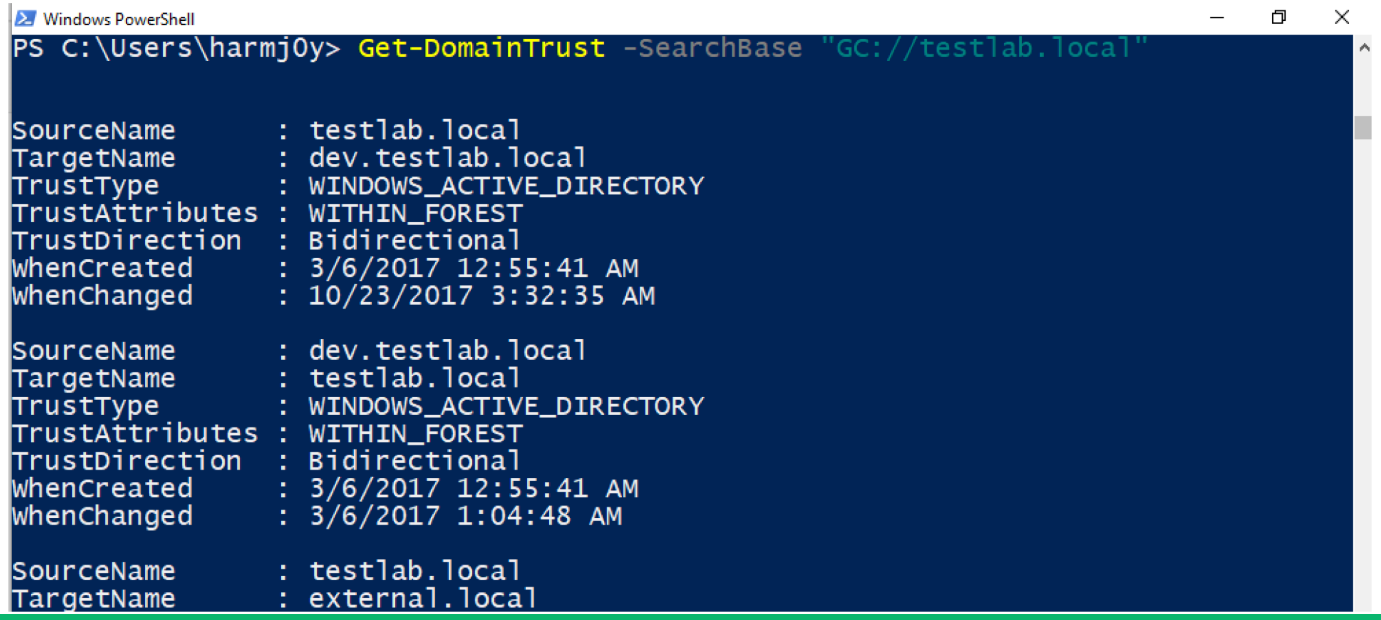
- **DOWNLEVEL** (0x00000001) - a trusted Windows domain that IS NOT running Active Directory
  - Output as **WINDOWS_NON_ACTIVE_DIRECTORY** in PowerView
- **UPLEVEL** (0x00000002) - a trusted Windows domain that IS running Active Directory
  - Output as **WINDOWS_ACTIVE_DIRECTORY** in PowerView
- **MIT** (0x00000003) - a trusted domain that is running a non-Windows (*nix), RFC4120-compliant Kerberos distribution

LDAP trustedDomain **-TrustAttributes**

- **NON_TRANSITIVE** (0x00000001) - trust cannot be used transitively
- **QUARANTINED_DOMAIN / FILTER_SIDS** (0x00000004) - the SID filtering protection is enabled for the trust
- **FOREST_TRANSITIVE** (0x00000008) - trust between two forests
- **WITHIN_FOREST** (0x00000020) - the trusted domain is within the same forest (parent/child, cross-link, tree-root)
- **TREAT_AS_EXTERNAL** (0x00000040) - external trust

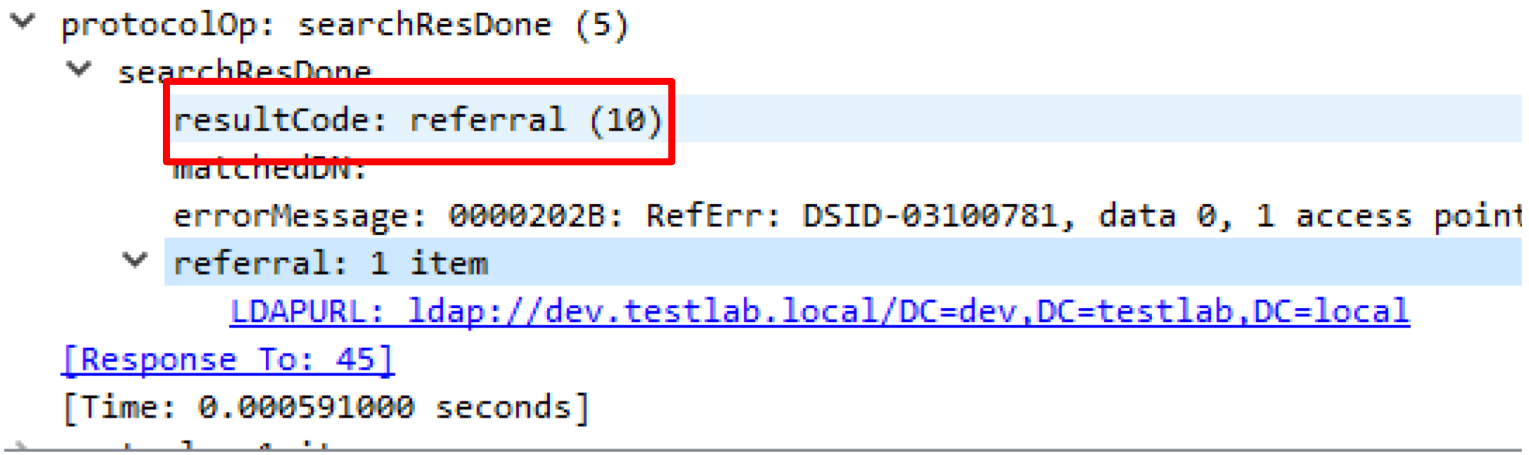
The Global Catalog and Trusts

- **trustedDomain** objects are replicated in the global catalog!
  - This means that we can enumerate ***all*** trusts (including external ones) for every domain in the entire forest, just by querying our local GC!



PowerView and Trusts

- If a trust exists, most functions in PowerView can accept a **-Domain <name>** flag to operate across a trust:
  - **Get-DomainComputer**, **Get-DomainComputer**, etc.
- If a trust exists, a referral is returned by your PDC, and the searcher binds to the remote DC using a referral ticket



Trust Attack Strategy

1. First map all trusts (forest and domain) that you can reach from your current domain context

1. Enumerate any users or groups in one domain that either:
   a. Have access to resources (including ACEs) in another domain
   b. Are in groups, or (if a group) have users from another domain
   c. **General idea:** find the hidden 'trust mesh' of relationships that administrators have set up (likely incorrectly ;)

1. Compromise specific target accounts in the domain you control in order to hop across the trust boundary to the target
   a. Caveat: if crossing an intra-forest trust, sidHistory-hopping is an option

Get-DomainForeignUser

- To enumerate *users* who are in *groups* outside of the user's primary domain
  - This is a domain's "outgoing" access
  - Only works for *intra*-forest trusts

```
PS C:\Users\harmj0y\Desktop> Get-DomainForeignUser -Domain dev.testlab.local


UserDomain              : dev.testlab.local
UserName                : jason.a
UserDistinguishedName   : CN=jason.a,CN=Users,DC=dev,DC=testlab,DC=local
GroupDomain             : testlab.local
GroupName               : ServerAdmins
GroupDistinguishedName  : CN=ServerAdmins,CN=Users,DC=testlab,DC=local
```

Get-DomainForeignGroupMember

- To enumerate *groups* with *users* who are outside of the group's primary domain
  - This is a domain's "incoming" access
  - Works for any trust type

```
PS C:\Users\harmj0y\Desktop> Get-DomainForeignGroupMember


GroupDomain             : TESTLAB.LOCAL
GroupName               : ServerAdmins
GroupDistinguishedName  : CN=ServerAdmins,CN=Users,DC=testlab,DC=local
MemberDomain            : dev.testlab.local
MemberName              : jason.a
MemberDistinguishedName : CN=jason.a,CN=Users,DC=dev,DC=testlab,DC=local
```

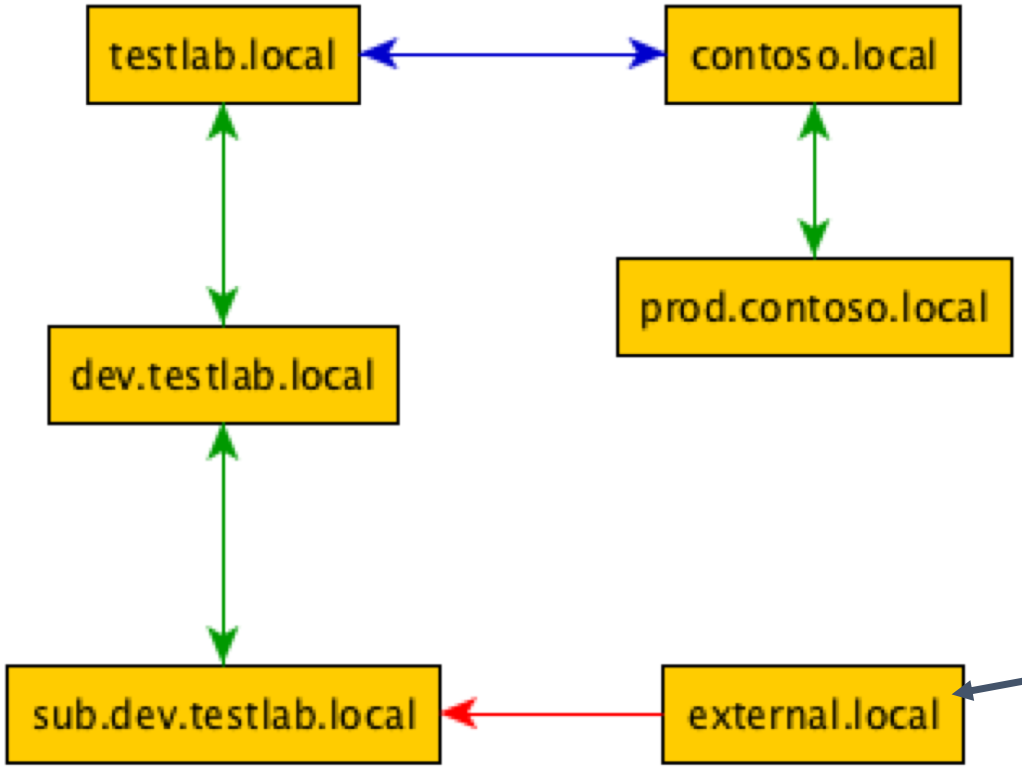
CN=ForeignSecurityPrincipals

- When a user from an external domain/forest are added to a group in a domain, an object of type **foreignSecurityPrincipal** is created at **CN=<SID>,CN=ForeignSecurityPrincipals,DC=domain,DC=com**
- You can quickly enumerate all incoming foreign trust members from the global catalog with:
  - **Get-DomainObject -Properties objectsid,distinguishedname -SearchBase "GC://testlab.local" -LDAPFilter '(objectclass=foreignSecurityPrincipal)' | ? {$_.objectsid -match '^S-1-5-.*-[1-9]\d{2,}$'} | Select-Object -ExpandProperty distinguishedname**

Why the ***Forest*** is the "trust boundary"

- A user's privilege access certificate (PAC, part of the TGT) contains:
  - Their security identifier (SID)
  - The SIDs of any security groups they're a part of
  - Anything set in **sidHistory** (ExtraSids in the PAC)
- When a user's TGT is presented to a *trusting* domain, specific SIDS are filtered out/ignored depending on settings
  - Sensitive SIDs like "S-1-5-21-<Domain>-519" are always filtered for external/forest trusts, but NOT intra-forest trusts!
  - This is why we can "hop up" a trust with sidHistory
- One exception- a forest-internal trust can be "Quarantined"
  - All sensitive sids are filtered EXCEPT S-1-5-9 ;)

Example trust “mesh”



green = within forest

red = external external

blue = inter-forest

say you land here

what do you do?

what are the implications?

This would be a good time to attempt Lab: Domain Trusts