Enterprise Content Management

Richard Robinson

Senior Software Engineer

October 20, 2014

Product Implementation Training (PIT)

IBM FileNet Content Platform Engine 5.2.1

Multi-Domain Authentication

Multi-Domain Security Realms (Part 1)

Realm definition : *“A collection of users and groups”*

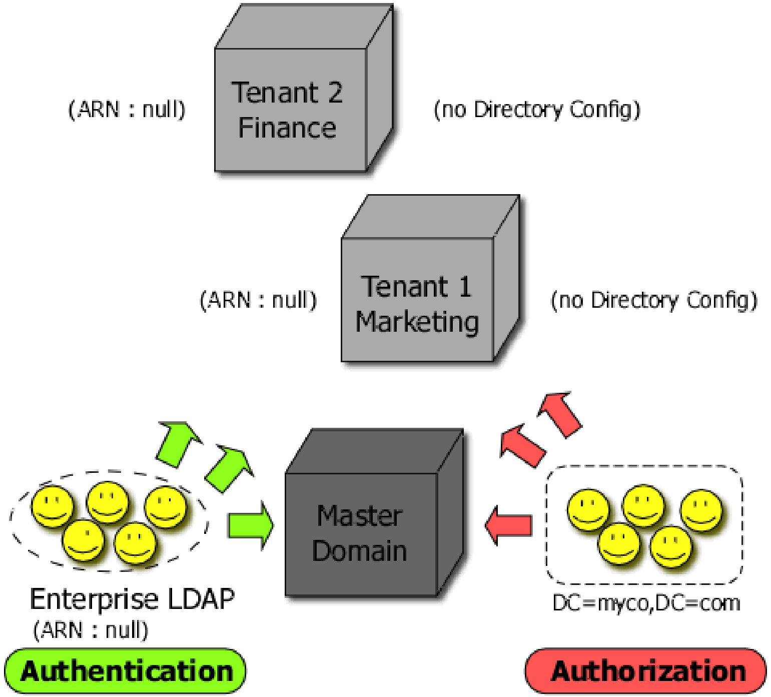
- Two types: Authentication and Authorization realms
- Authentication Realms (*who you are*)
  - Configured on application server (usually via CMUI)
  - On WebSphere can be a Federated Repository, Standalone LDAP or external SAML Identity Provider (IdP)
- Authorization Realms (*what you can do*)
  - Defined by CPE Directory Configurations via ACCE or FEM
- Authentication and Authorization realms can be different sources
  - But users' shortnames should be same if sources different
- All above applies currently to regular domains but with multi-domain...
  - Realms are optional for tenant domains but required for master
  - Tenants always inherit the master domain's realms
  - Master and tenant domains have Authentication Realm Names (ARNs)

Multi-Domain Security Realms (Part 2)

A domain's Authentication Realm Name (ARN) property controls entry into that master/tenant domain

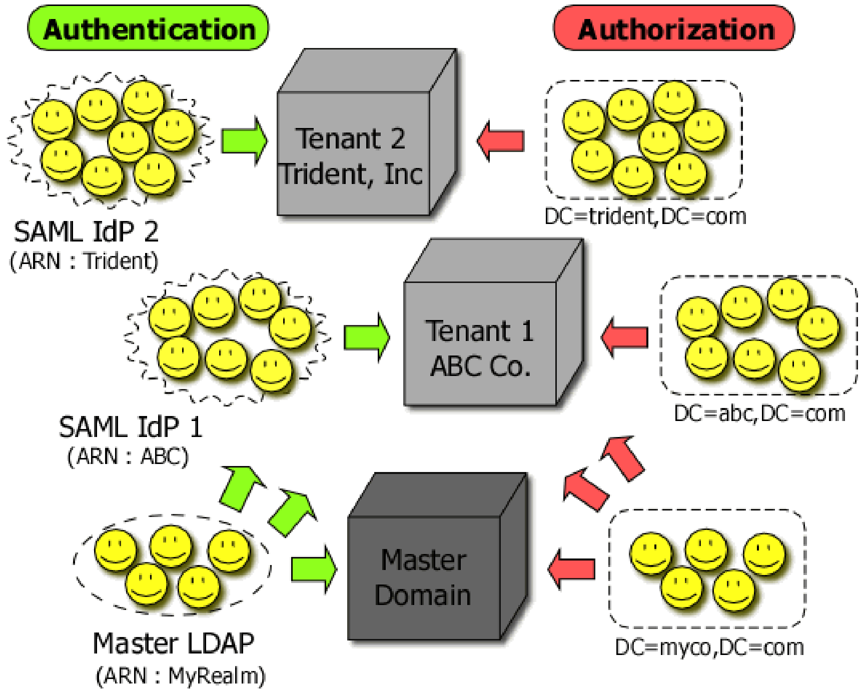
- ARN's value can be the allowed application server realms
  - On WebSphere, subject has name of realm where authenticated
  - Use WebSphere admin console to find or set this Realm name
  - This type of ARN not supported on WebLogic or JBoss
  - Value starts with "/" e.g., "/myldap:389", "/defaultWIMFileBasedRealm"
- Or the ARN's value can be the allowed email domains
  - Parsed from user's shortname (should be email or UPN)
  - Value starts with "@" e.g., "@us.ibm.com, @uk.ibm.com"
- Or the ARN can be null – meaning all are allowed into this domain
  - Low security, e.g., one tenant's users can enter another tenant
- Master and tenant ARNs must all be null or all be non-null
  - Can go from null to non-null but must set master's ARN first

Use Case 1 – On-Premises Tenants



On-Prem is easy to set up since only the master's authentication and authorization realms are used and are shared by all domains.
However, because all users are in the same realm, this setup has no realm isolation.
For example, users of Marketing can also access Finance and vice versa.

Use Case 2 – Tenants in the Cloud



Here, realms are isolated from each other so, e.g., Trident users cannot access ABC's tenant domain. Cloud tenants can make use of external SAML IdPs that are used to authenticate users but this setup also needs a “backing” LDAP. Note : master users can also enter tenants if ACLs of tenant domain/object stores allow.

Multi-domain Security Caveats

- Restrictions on CPE Directory Configuration (authorization) User Base DNs in master and any of its tenants...
  - On eDirectory and OID, user base DNs cannot end in same suffix, e.g., dc=myco,dc=com and dc=trident,dc=com bad as both end in dc=com
  - On other LDAP types, user base DNs must be in different namingContexts
  - If this restriction not followed, cannot access master realm users
  - Tenant vs. other tenant's base DNs do not have this restriction
- Any tenant will always “inherit” the master's authentication realm(s) so do not add master realms to tenant ARNs
- Duplicate shortnames between the master and a tenant are an error
  - Recommendation is to use another shortname attribute for master
- Null ARNs are handy but insecure since cases are possible where tenant 2's “Joe” is able to enter tenant 1 and be accepted as tenant 1's “Joe”
- A tenant need not have a CPE Directory Configuration but, if it doesn't, its ARN is ignored and only master realm users may enter that tenant

High-level Overview of Multi-domain Security Logic

1) If user is anonymous, throw **SECURITY_ANONYMOUS_DISALLOWED**
2) If current domain is a tenant, check user's realm against tenant's ARN
   - If ok (or ARN is null) and if tenant has a Directory Configuration (DC), attempt to look up the user using tenant's DC
3) Check user's realm against the master's ARN
   - If ok (or ARN is null), attempt to look up the user using the master's DC
4) If no lookup succeeded, throw **E_NOT_AUTHENTICATED**
5) If two lookups succeeded, throw **SECURITY_TOO_MANY_MATCHES**
   - *!!! Don't let the master and a tenant have duplicate shortnames !!!*
6) Use the one successful lookup and its DC to get the User Access Token
7) Check current domain's ACL to make sure user has access
8) If applicable, check object store's ACL to make sure user has access

Questions?