FreeIPA
Open Source Identity Management Solution

Global Catalog project progress

Alexander Bokovoy
Sr. Principal Software Engineer

Florence Blanc-Renaud
Sr. Software Engineer

Red Hat

Overview

- Introduction to FreeIPA
- Differences between FreeIPA and Active Directory implementations
- User login scenarios
- Managing user permissions in Active Directory
- Implementation discoveries
- Demos
  - Permissions of FreeIPA users in Active Directory
  - Login to Windows as a FreeIPA user
  - Access file services as a FreeIPA user
- What works and what doesn’t
- Testing Global Catalog on Fedora 32

Consolidation approach

FreeIPA is a consolidation project, bringing together dozens of open source projects centered around identity and authentication. Fitting everything together means working directly with projects' upstreams and operating system distribution downstreams.

- Authentication: MIT Kerberos, 389-ds LDAP directory server, and Samba
- Identity: 389-ds LDAP directory server, Samba, and SSSD
- Certificate management: Dogtag PKI and Java ecosystem
- Cryptography: OpenSSL, NSS (Mozilla), and GnuTLS
- FreeIPA management: Python ecosystem, Ansible ecosystem, Apache
- Resource management: systemd, SELinux, SUDO
- Distributions: Fedora, Red Hat Enterprise Linux, Debian, Ubuntu, BaseALT (ALT Linux)

FreeIPA
Open Source Identity Management Solution

Identity management for the enterprise grid

- Since 2012, FreeIPA supports a forest trust to Active Directory:
  FreeIPA deployment is seen by Active Directory deployment as “another Active Directory forest”
  - Linux systems enrolled to FreeIPA
  - Windows systems enrolled to Active Directory
  - Users from Active Directory can access resources on Linux systems

Red Hat

Different implementation approach

LDAP differences

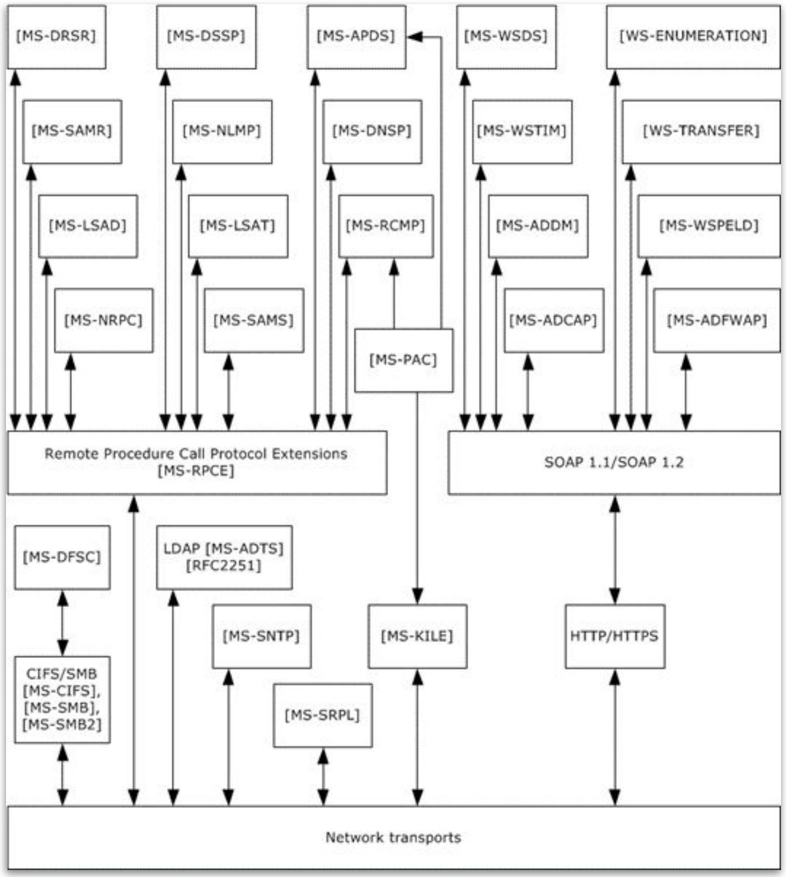
FreeIPA uses LDAP to store its information about users, groups, machines, and other objects

- LDAP directory information tree (DIT) is simplified
  - No organizational units, flat structure for users and groups
  - Placement of objects in DIT is different from Active Directory
- LDAP schema is different from Active Directory
  - Active Directory LDAP schema is conflicting with traditional UNIX environment LDAP schema in both attributes and objects identifiers and semantics of certain LDAP rules
  - FreeIPA objects are specific to Linux (POSIX-like) environments

Minimal implementation approach

FreeIPA does not implement full Active Directory compatibility

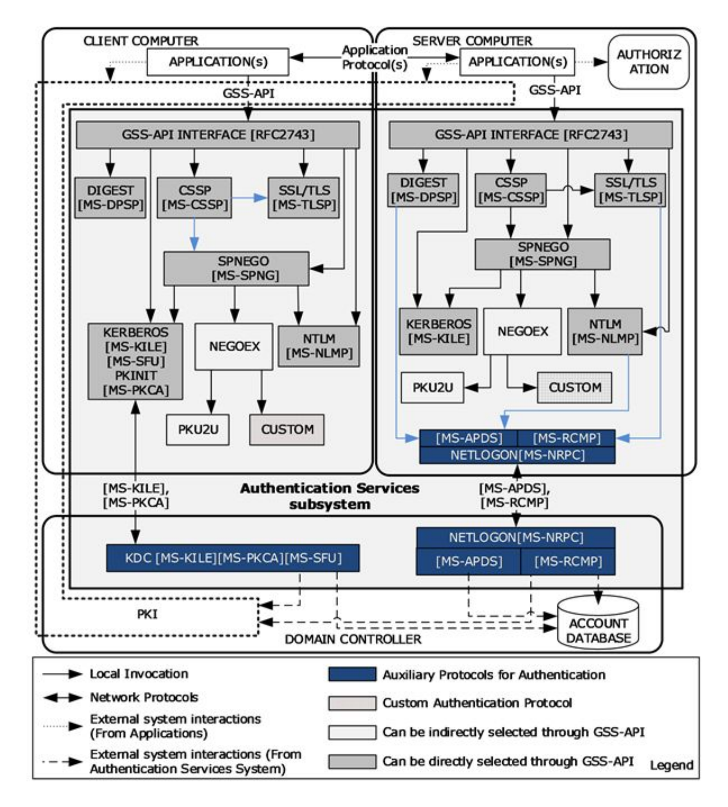
- Use of forest trust avoids a need to implement Active Directory-specific replication protocol, [MS-DRSR], and no need to have Active Directory LDAP schema in use for Linux systems
- Limited contact points allow to implement less protocol specifications
  - Authentication: Kerberos ([MS-KILE], [MS-PAC], [MS-SFU], [MS-SPNG]) and DCE RPC ([MS-RPCE], [MS-NRPC], ..)
  - Identity: DCE RPC ([MS-LSAD], [MS-LSAT], ..)
- Even with minimal approach, there is a lot to work on
  - Protocol relationship on the side is from [MS-ADOD] overview document



Bordering on a trust edge

Authentication in Active Directory environment is a complex mix of multiple protocols and protocol extensions.

- Example: network logon protocol interactions from [MS-AUTHSOD]
- Most of complexity is amplified by having multiple domains in the forest
- On top of that, forest trust adds own flavor
- Summary of login use cases: [MS-AUTHSOD] 2.5



Active Directory user login to Linux systems

“Easy part”, when single sign-on is available

Login to Linux systems as Active Directory user with Kerberos ticket is a solved problem

- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a service ticket to Linux host service
- Windows client presents the service ticket to Linux host service
- Linux host service authenticates the Windows client
- Linux host service authorizes the access
    - Involves resolving Active Directory user’s identity by talking back to AD DCs
        - Not directly (through a FreeIPA DC) but this is a detail

Active Directory user login to Linux systems

“Easy part”, when single sign-on is available

Login to Linux systems as Active Directory user with Kerberos ticket is a solved problem

Kerberos protocol

- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a service ticket to Linux host service
- Windows client presents the service ticket to Linux host service
- Linux host service authenticates the Windows client
- Linux host service authorizes the access
    - Involves resolving Active Directory user’s identity by talking back to AD DCs
        - Not directly (through a FreeIPA DC) but this is a detail

Active Directory user login to Linux systems

“Easy part”, when single sign-on is available

Login to Linux systems as Active Directory user with Kerberos ticket is a solved problem

Kerberos protocol

- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a service ticket to Linux host service
- Windows client presents the service ticket to Linux host service
- Linux host service authenticates the Windows client
- Linux host service authorizes the access
  - Involves resolving Active Directory user’s identity by talking back to AD DCs
    - Not directly (through a FreeIPA DC) but this is a detail

One-way trust between FreeIPA and Active Directory is enough:

FreeIPA has to trust Active Directory but Active Directory does not need to trust FreeIPA

FreeIPA user login to Windows systems

“Medium complexity”, when single sign-on is available

Login to Windows systems as FreeIPA user interactively

- FreeIPA user credentials entered to Windows login dialog
- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a ticket granting ticket for IPA user
- Windows client asks AD DC to issue service ticket to Windows client machine service
- Windows client authenticates FreeIPA user
- Windows client authorizes the access
    - Involves resolving FreeIPA user’s identity by talking back to AD DCs
        - And AD DCs talking to FreeIPA DC to resolve IPA user identity

FreeIPA user login to Windows systems

“Medium complexity”, when single sign-on is available

Login to Windows systems as FreeIPA user interactively

Kerberos protocol

- FreeIPA user credentials entered to Windows login dialog
- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a ticket granting ticket for IPA user
- Windows client asks AD DC to issue service ticket to Windows client machine service
- Windows client authenticates FreeIPA user
- Windows client authorizes the access
  - Involves resolving FreeIPA user’s identity by talking back to AD DCs
    - And AD DCs talking to FreeIPA DC to resolve IPA user identity

FreeIPA user login to Windows systems

“Medium complexity”, when single sign-on is available

Login to Windows systems as FreeIPA user interactively

Kerberos protocol

- FreeIPA user credentials entered to Windows login dialog
- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a ticket granting ticket for IPA user
- Windows client asks AD DC to issue service ticket to Windows client machine service
- Windows client authenticates FreeIPA user

LSA protocol

- Windows client authorizes the access
  - Involves resolving FreeIPA user’s identity by talking back to AD DCs
    - And AD DCs talking to FreeIPA DC to resolve IPA user identity

FreeIPA user login to Windows systems

“Medium complexity”, when single sign-on is available

Login to Windows systems as FreeIPA user interactively

Kerberos protocol

- FreeIPA user credentials entered to Windows login dialog
- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA domain controller
- Windows client asks FreeIPA DC to issue a ticket granting ticket for IPA user
- Windows client asks AD DC to issue service ticket to Windows client machine service
- Windows client authenticates FreeIPA user

LSA protocol

- Windows client authorizes the access
  - Involves resolving FreeIPA user’s identity by talking back to AD DCs
    - And AD DCs talking to FreeIPA DC to resolve IPA user identity

Two-way trust between FreeIPA and Active Directory is required:

FreeIPA has to trust Active Directory and Active Directory has to trust FreeIPA

FreeIPA user login to Windows systems

“Medium complexity”, when single sign-on is available

Login to Windows systems as FreeIPA user interactively

Kerberos protocol

- FreeIPA user credentials ente
- Windows client talks to own d
- AD DC gives a referral to Free
- Windows client asks FreeIPA D
- Windows client asks AD DC to issue service ticket to Windows client machine service
- Windows client authenticates FreeIPA user

LSA protocol

- Windows client authorizes the access
  - Involves resolving FreeIPA user’s identity by talking back to AD DCs
    - And AD DCs talking to FreeIPA DC to resolve IPA user identity

Permissions on Active Directory side need to be granted prior to the authorization step to succeed

Two-way trust between FreeIPA and Active Directory is required:

FreeIPA has to trust Active Directory and Active Directory has to trust FreeIPA

Granting permissions on the Active Directory side

What can go wrong?

As part of Open Protocol Specifications, Microsoft only documents server side protocols. Windows client behavior is mostly undocumented.

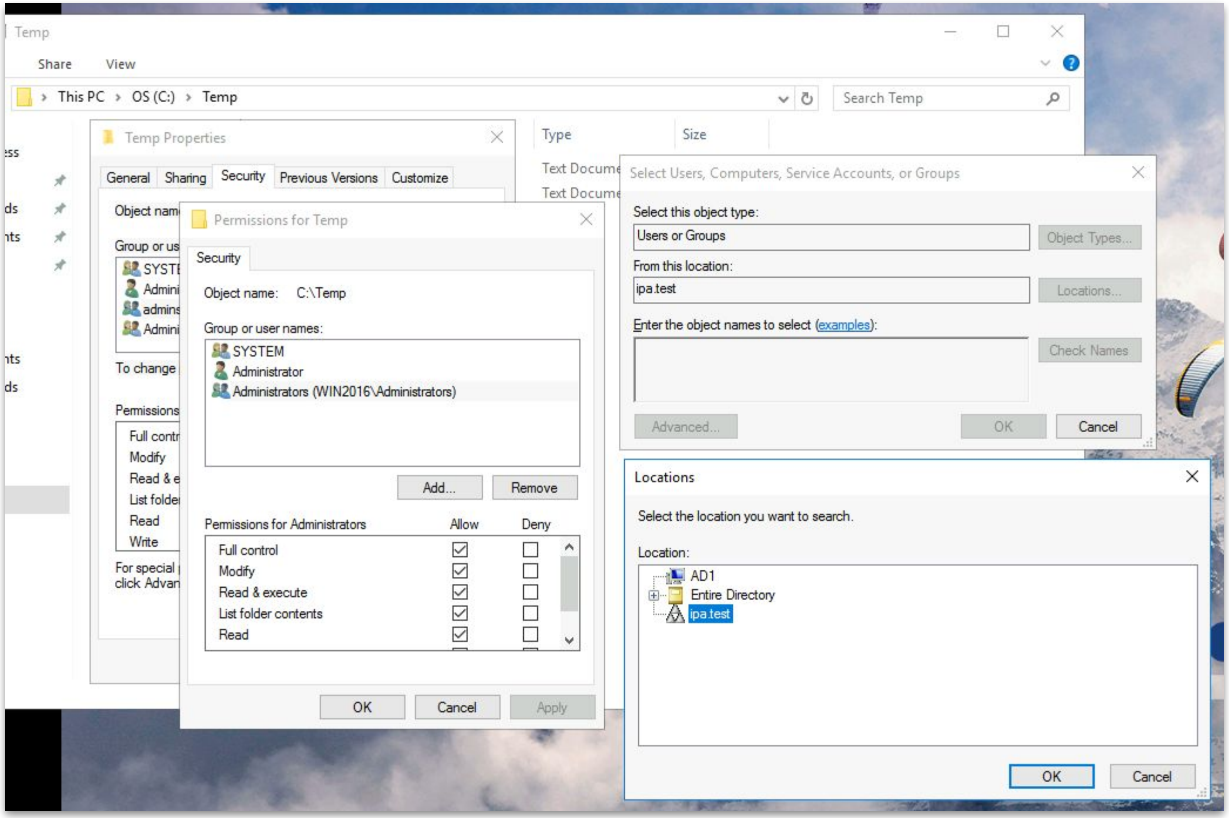
- Windows UI behavior changes between Windows versions
- Windows UI uses Windows system API calls in various ways not always documented in Windows API documentation
- Windows UI combines multiple APIs, including other, non system, API calls
- Windows UI talks directly to Active Directory LDAP service

Adding permissions in Windows UI

“Easy”

Most known way of adding permissions is through the “Security tab” UI

- “Security tab” allows to search users or groups through known “locations”
- Locations include local computer, own AD forest, and all trusted forests
- Search can do a partial name match
  - Windows system API has no way to perform partial name searches
  - Window UI does Active Directory LDAP searches instead
    - In fact, not just LDAP but Global Catalog search



Temp

This PC › OS (C:) › Temp

Temp Properties

Permissions for Temp

Object name: C:\Temp

Group or user names:

SYSTEM

Administrator

Administrators (WIN2016\Administrators)

Permissions for Administrators	Allow	Deny
Full control	☑	□
Modify	☑	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□

Select Users, Computers, Service Accounts, or Groups

Select this object type: Users or Groups

From this location: ipa.test

Enter the object names to select (examples):

Locations

Select the location you want to search.

Location:

AD1

Entire Directory

ipa.test

Global Catalog service

Global Catalog (GC) service contains a unified view onto all LDAP directories (“naming contexts”) of the Active Directory forest domains.

- It is a subset of all objects and their attributes from all forest domains
- Each Active Directory forest has at least one GC server
- Global Catalog allows to expose those attributes of users/groups that otherwise impossible to query through LSA protocols
  - LSA protocols used widely for security and identity resolution but have no ways to extend what could be retrieved and how

Global Catalog service

Windows clients require use of Global Catalog to be able to add users or groups to permissions in the Windows UI. There is no way around it if UI should be used. They also assume Active Directory LDAP schema and directory information tree (DIT) design is in place

- FreeIPA LDAP DIT design is incompatible with AD LDAP design
- FreeIPA LDAP server also have limitations on how it can present the same information through different and incompatible views
- Result: we have to implement a separate AD-compatible Global Catalog service

Global Catalog service in FreeIPA

Global Catalog service in FreeIPA is a set of services

- Separate LDAP instance with Active Directory schema and directory information tree
- A dedicated service to synchronize content from FreeIPA LDAP instance to GC service
- An installation tooling that maintains Global Catalog LDAP instance

Global Catalog service in FreeIPA

Global Catalog service is read-only

- Windows clients never write to Global Catalog, only search it
- Greatly simplifies object lifecycle management
- Allows to define object transformation logic more precisely
- GC LDAP instance can be dropped and re-populated from scratch from the primary FreeIPA LDAP tree

FreeIPA user login to Windows systems

“Medium complexity”, when single sign-on is available

Login to Windows systems as FreeIPA user interactively

Kerberos protocol

- FreeIPA user credentials entered to Windows login dialog
- Windows client talks to own domain controller (AD DC)
- AD DC gives a referral to FreeIPA do
- Windows client asks FreeIPA DC to is
- Windows client asks AD DC to issue s ce
- Windows client authenticates FreelP

LSA protocol

- Windows client authorizes the access
  - Involves resolving FreeIPA user’s identity b alking back to AD DCs
  - And AD DCs talking to FreeIPA DC to resolve IPA user identity

FreeIPA DC needs to understand format of the names in the requests from Active Directory DCs

Two-way trust between FreeIPA and Active Directory is required:

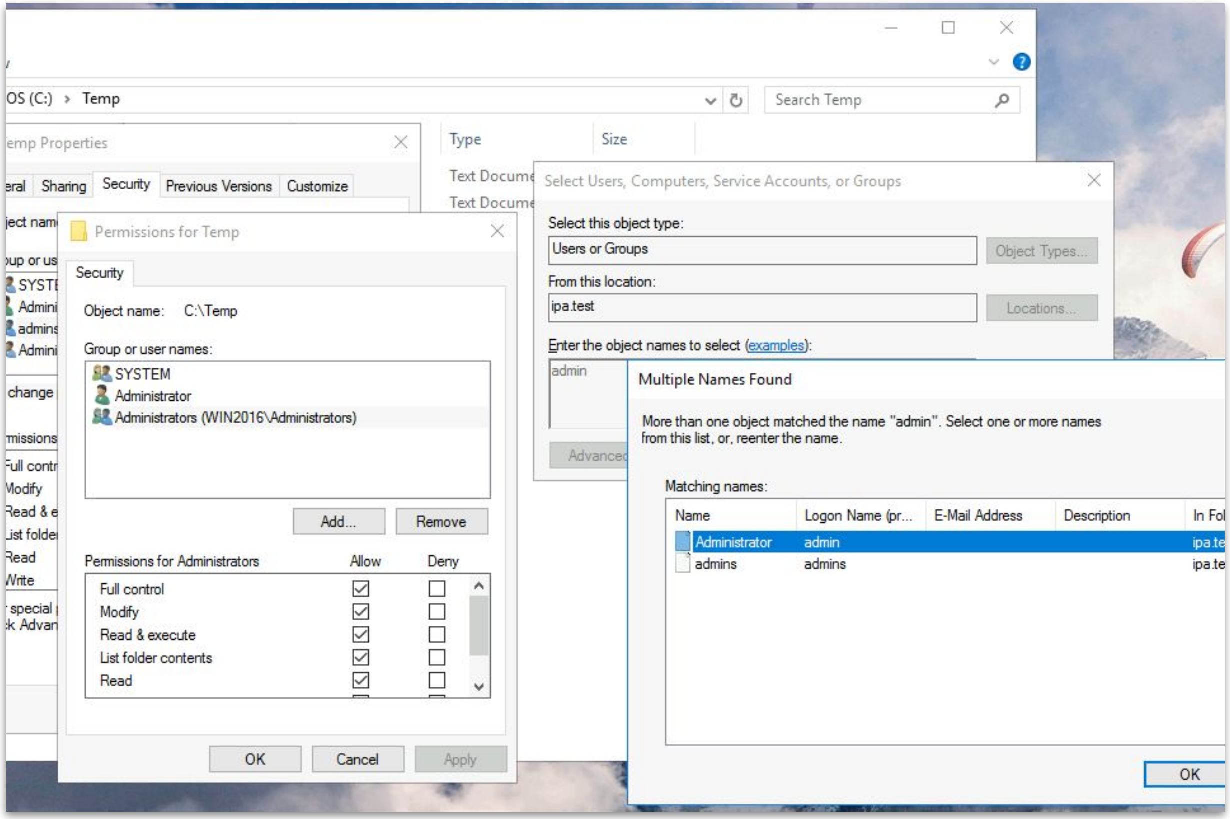
FreeIPA has to trust Active Directory and Active Directory has to trust FreeIPA

Adding permissions in Windows UI (part II)

“Easy”

Searching for users or groups in Global Catalog may return multiple results

- “Security tab” allows to select the right object
- Choosing the object will cause Windows UI to ask Windows system API about Security Identifier (SID) of the object
- Translation user or group name to SID is done through the LSA protocol
    - Even though Windows already has SID information from Global Catalog, it does search it again via LSA because it is required by other system APIs to set permissions (ACLs)



Object name: C:\Temp

Group or user names:

- SYSTEM
- Administrator
- Administrators (WIN2016\Administrators)

Permissions for Administrators	Allow	Deny
Full control	☑	□
Modify	☑	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□

Select this object type: Users or Groups

From this location: ipa.test

Enter the object names to select (examples): admin

More than one object matched the name "admin". Select one or more names from this list, or, reenter the name.

Matching names:

Name	Logon Name (pr...	E-Mail Address	Description	In Fol
Administrator	admin			ipa.te
admins	admins			ipa.te

LSA

Welcome to Samba

Local Security Authority (LSA) is a set of network calls widely used by Windows systems to talk about identities of objects. Most important family of calls in the context of FreeIPA trust to Active Directory is lsa_LookupNames:

- Allows to translate names to security identifiers (SIDs)
- Allows to look up names with various contexts (builtin, local computer only, domain only, trusted domains, etc.)
- On Linux it is implemented as a part of Samba suite

LSA

Also found in: Medical, Encyclopedia, Wikipedia.

Category filter: Show All (125)

Acronym	Definition
LSA	Light Sport Aircraft
LSA	Link State Advertisements
LSA	Literature, Science, and the Arts *(university)*
LSA	Latent Semantic Analysis
LSA	Landes Sachsen Anhalt *(German state)*
LSA	Linguistic Society of America
LSA	Local Security Authority *(Windows NT)*

Identity resolution in Samba

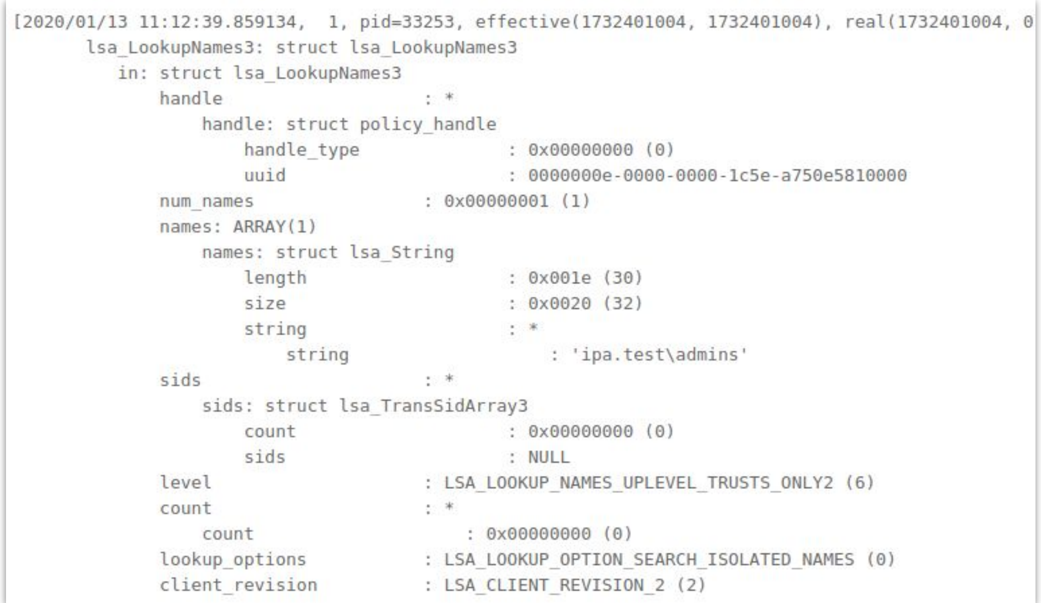
Samba is an application with a large scope. It can be configured to perform multiple different tasks. FreeIPA already uses Samba to provide a minimal “AD domain controller” for forest trust to Active Directory.

- Samba is configured as “classic domain controller” (NT4-like)
- Samba looks into FreeIPA LDAP database for user and group details
- Internally, Samba uses different lookup methods and different logic depending on how and what clients ask it about information.

Lookup Names

This is how a lookup from Windows UI looks like

- A client asks to look up ‘ipa.test\admins’ name
- A client needs to know if it exists in trusted domains of this server (normal request for forest trust)
- NT4-style domain controller is unprepared to receive reference to its primary domain as its Kerberos realm
  - It is NT4-style, not Active Directory, after all
- Fixing this is complicated for multiple reasons as it breaks other Samba use cases
  - We had a prototype fix, merged it upstream, only to revert it back
  - New version was merged upstream as MR#1658, will be out in Samba 4.14



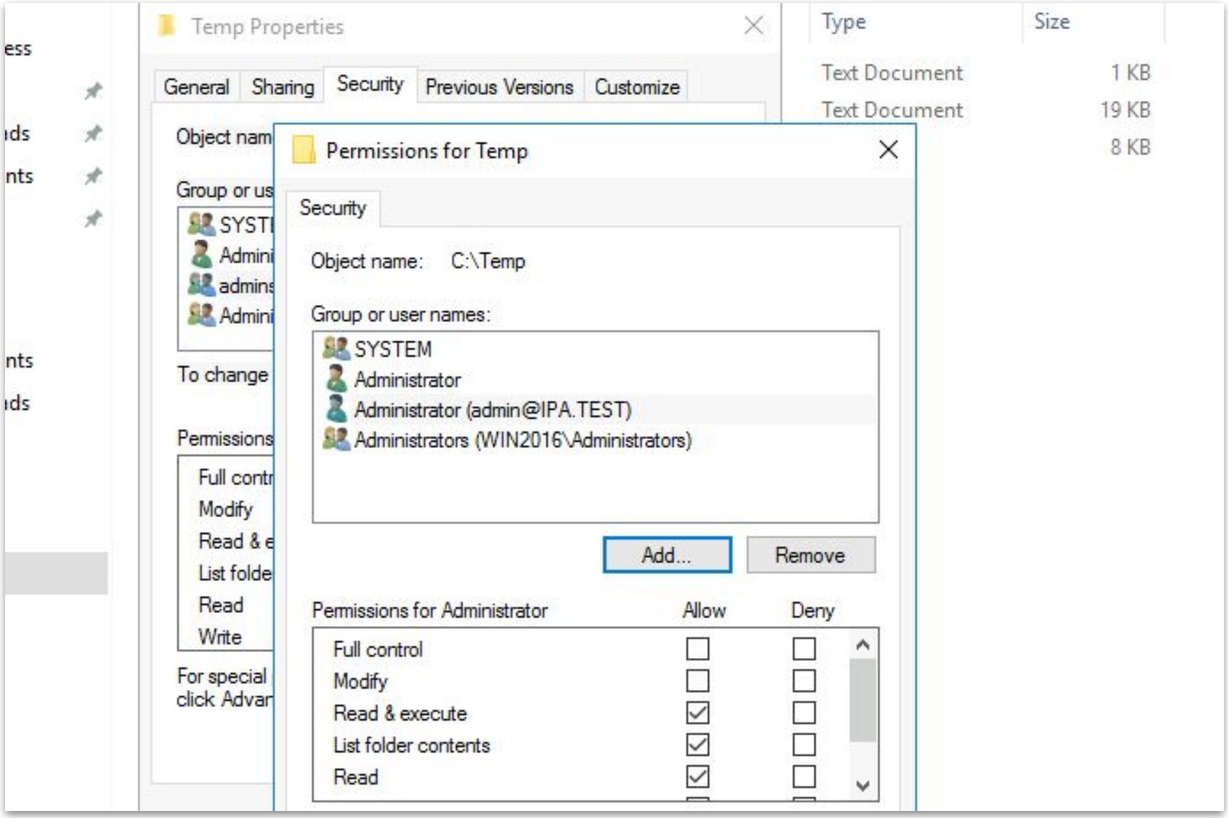
```
[2020/01/13 11:12:39.859134,  1, pid=33253, effective(1732401004, 1732401004), real(1732401004, 0
       lsa_LookupNames3: struct lsa_LookupNames3
          in: struct lsa_LookupNames3
              handle                    : *
                  handle: struct policy_handle
                      handle_type               : 0x00000000 (0)
                      uuid                      : 0000000e-0000-0000-1c5e-a750e5810000
              num_names                 : 0x00000001 (1)
              names: ARRAY(1)
                  names: struct lsa_String
                      length                    : 0x001e (30)
                      size                      : 0x0020 (32)
                      string                    : *
                          string                    : 'ipa.test\admins'
              sids                      : *
                  sids: struct lsa_TransSidArray3
                      count                     : 0x00000000 (0)
                      sids                      : NULL
              level                     : LSA_LOOKUP_NAMES_UPLEVEL_TRUSTS_ONLY2 (6)
              count                     : *
                  count                     : 0x00000000 (0)
              lookup_options            : LSA_LOOKUP_OPTION_SEARCH_ISOLATED_NAMES (0)
              client_revision           : LSA_CLIENT_REVISION_2 (2)
```

When name is resolved

“There is a catch”

When name is resolved both in Global Catalog and through LSA service, Windows client can add a permission

- If resolution works through GC and LSA, almost everything will work in order to assign permissions
- One can add FreeIPA users and groups to permissions or groups in Active Directory
- There is a catch: group membership is limited by the Active Directory design
  - Only domain local groups can contain users from other forests



Temp Properties

General | Sharing | Security | Previous Versions | Customize

Permissions for Temp

Security

Object name: C:\Temp

Group or user names:

- SYSTEM
- Administrator
- Administrator (admin@IPA.TEST)
- Administrators (WIN2016\Administrators)

Add... | Remove

Permissions for Administrator	Allow	Deny
Full control	□	□
Modify	□	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□

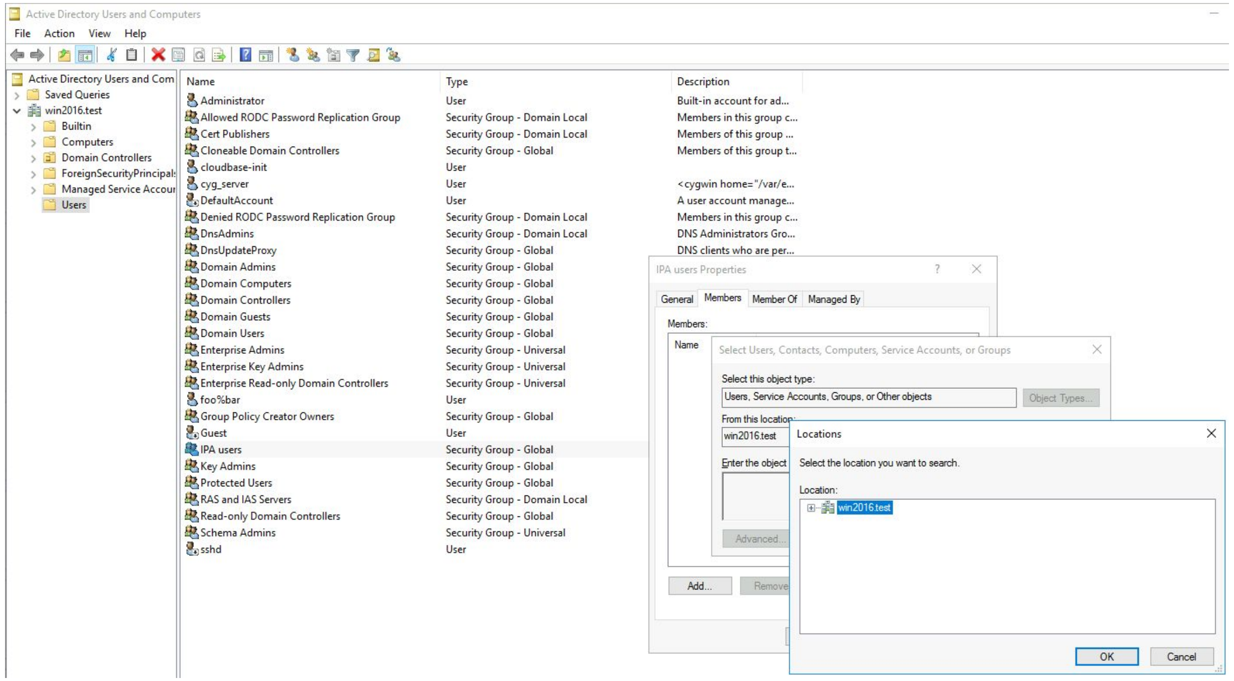
Type	Size
Text Document	1 KB
Text Document	19 KB
	8 KB

Group types and their scope

"There is a catch"

Windows has several types of groups that can be used to assign permissions and validate permissions

- Domain Local groups can include any user or group from any trusted domain, including any trusted forest
  - Only give access to resources where the group itself is located
- Domain Global groups only can include users and groups from the same domain where the group itself is located
  - Gives access to resources in any in-forest domain
- Universal groups can contain members from the domains of the same forest



Name	Type	Description
Administrator	User	Built-in account for ad...
Allowed RODC Password Replication Group	Security Group - Domain Local	Members in this group c...
Cert Publishers	Security Group - Domain Local	Members of this group ...
Cloneable Domain Controllers	Security Group - Global	Members of this group t...
cloudbase-init	User	
cyg_server	User	<cygwin home="/var/e...
DefaultAccount	User	A user account manage...
Denied RODC Password Replication Group	Security Group - Domain Local	Members in this group c...
DnsAdmins	Security Group - Domain Local	DNS Administrators Gro...
DnsUpdateProxy	Security Group - Global	DNS clients who are per...
Domain Admins	Security Group - Global	
Domain Computers	Security Group - Global	
Domain Controllers	Security Group - Global	
Domain Guests	Security Group - Global	
Domain Users	Security Group - Global	
Enterprise Admins	Security Group - Universal	
Enterprise Key Admins	Security Group - Universal	
Enterprise Read-only Domain Controllers	Security Group - Universal	
foo%bar	User	
Group Policy Creator Owners	Security Group - Global	
Guest	User	
IPA users	Security Group - Global	
Key Admins	Security Group - Global	
Protected Users	Security Group - Global	
RAS and IAS Servers	Security Group - Domain Local	
Read-only Domain Controllers	Security Group - Global	
Schema Admins	Security Group - Universal	
sshd	User	

IPA users Properties

Select Users, Contacts, Computers, Service Accounts, or Groups

Select this object type:
Users, Service Accounts, Groups, or Other objects

From this location:
win2016.test

Enter the object

Locations

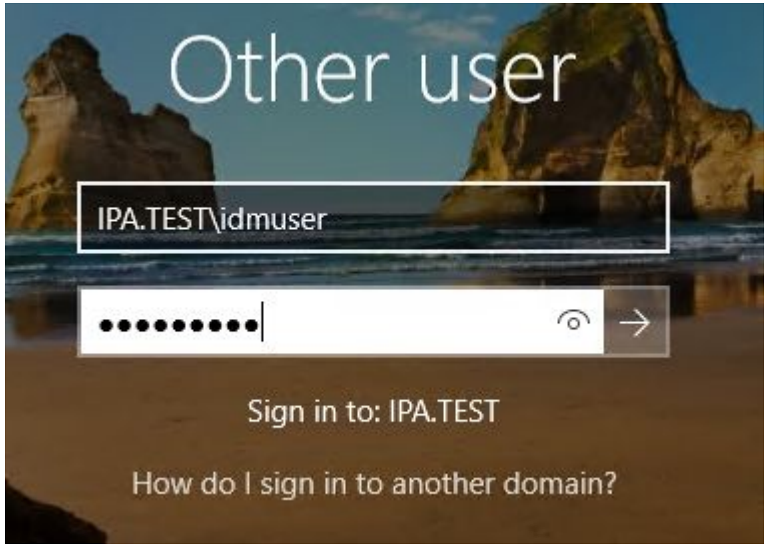
Select the location you want to search.

Location:
win2016.test

Life after permissions

When permissions are granted, one can login to Windows clients or use SMB services provided by Windows machines

- Interactively:
  - by using either DOMAIN\username, REALM\username, or username@realm
    - username@domain with NetBIOS domain name is not possible at the moment
- Remotely with RDP:
  - With XFreeRDP built with GSSAPI support (broken and disabled in Fedora)
  - Password-based login over RDP requires additional fix in Samba
    - Mergedas MR#1677, will be part of Samba 4.14, available in Fedora 33+ and RHEL 8.4
- To login to Active Directory domain controller, one has to be a member of a local administrators group on AD DC



Other user

IPA.TEST\idmuser

••••••••••

Sign in to: IPA.TEST

How do I sign in to another domain?

Progress so far

FreeIPA Global Catalog project is a work in progress. So far, we were able to do the following:

- Create Global Catalog service on FreeIPA domain controller
- Synchronize content of GC with primary FreeIPA LDAP store
- Establish two-way trust to Active Directory forest
- Resolve FreeIPA users and groups through Global Catalog service and LSA calls
- Add FreeIPA users and groups to groups with domain local scope on Active Directory side
- Add FreeIPA users and groups to access controls of Active Directory resources through Windows UI's "Security Tab" and Windows command line tools
- Login to Windows interactively with FreeIPA user credentials
- Login to Windows remotely with FreeIPA user with the help of XFreeRDP, with Kerberos or password
- Access file resources on Windows file servers through SMB protocol as FreeIPA user

Still to do

FreeIPA Global Catalog project is a work in progress. Active Directory protocols and implementation details on both Windows server and client side are still not fully covered. We also identified a number of improvements in FreeIPA stack.

- LDAP synchronization protocol (RFC 4533) implementation in 389-ds has a number of issues we are still trying to fix
- Security descriptors of objects in the Global Catalog LDAP instance in FreeIPA are not fully compatible with Active Directory expectations.
- LDAP extended operations and controls provided in Active Directory are mostly not implemented yet. We need to see applications that require these controls.
- Forest trust validation from Active Directory does not always succeed due to internal Samba logic for searching domain controllers specified by NetBIOS names
- Due to stricter crypto requirements in RHEL 8 and Fedora, use of NTLM authentication might become impossible in FIPS mode
- More testing is needed
- Documentation is still mostly missing

How to test and help

FreeIPA Global Catalog project is being developed on Github:
https://github.com/abbra/freeipa/tree/gc-wip

We rebase regularly to FreeIPA upstream git master and provide automatic builds for Fedora in COPR repository

- Global Catalog: https://copr.fedorainfracloud.org/coprs/abbra/gc-wip
- XFreeRDP built with GSSAPI support: https://copr.fedorainfracloud.org/coprs/abbra/gssapi-test
- We are available for discussion on FreeNode #freeipa and #ipaintegration IRC channels

Deploying FreeIPA Global Catalog

1. Enable COPR repository abbra/gc-wip:

   ```
   dnf  copr -y enable abbra/gc-wip
   dnf install freeipa-server{,-dns,-trust-ad}
   ```

2. Install FreeIPA from the COPR:

   ```
   ipa-server-install --setup-dns ....
   ipa-adtrust-install ...
   ```

3. Establish bi-directional trust to Active Directory:

   ```
   kinit admin
   ipa trust-add <ad.domain> --two-way=true --password
   ```

4. Add ID override for AD administrator that will be used to set up IPA users’ access on AD side:

   ```
   ipa idoverrideuser-add 'Default Trust View' <user>@<ad.domain>
   ```

Using FreeIPA Global Catalog

See demos for practical use cases

a. Login to Windows client using <user>@<IPA.domain> format:

b. Use Security tab in File Explorer folder properties to add IPA users/groups to access files

c. In order to login to Active Directory domain controller, IPA users must be part of a local group that allows such access

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat