Red Hat
Enterprise Linux

Red Hat Authentication with AD

Marc Skinner
Principal Solutions Architect

Red Hat

What is SSSD?

System Security Service Daemon

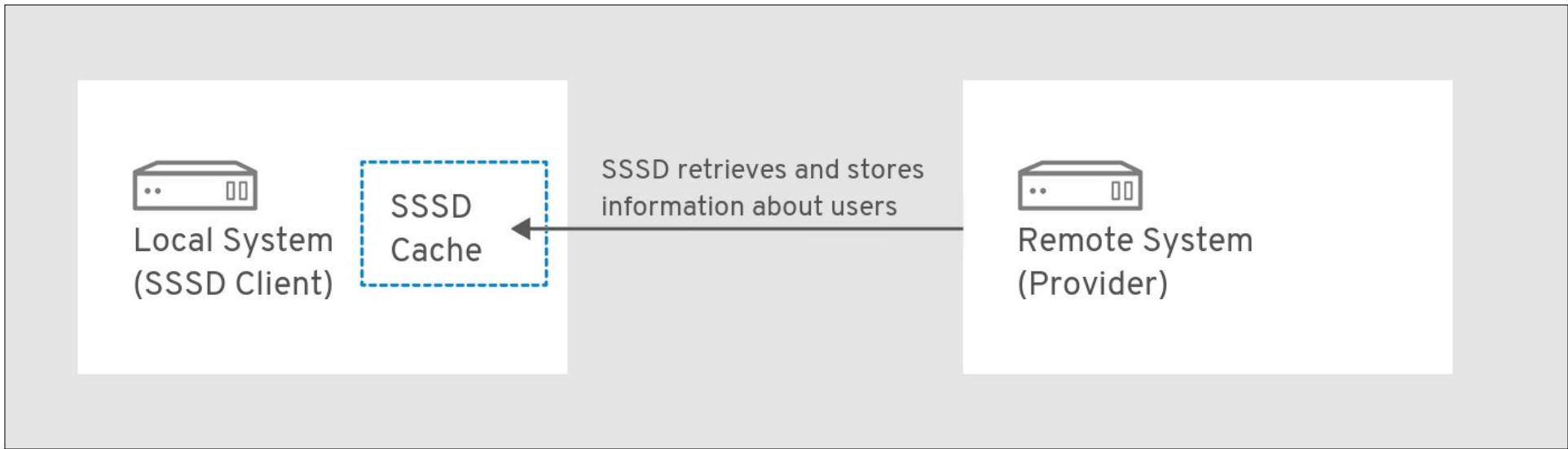
- What is SSSD?
  - **S**ystem **S**ecurity **S**ervice **D**aemon

- Client → Provider
  - LDAP
  - IDM
  - **AD**
  - Kerberos Realm

System Security Service Daemon

- SSSD works in two stages:

- Connects the client to a remote provider to retrieve identity and authentication information
- Uses the obtained authentication information to create a local cache of users and credentials on the client



AD is not AAD

- Windows Active Directory is not Azure Active Directory
- RHEL SSSD can only directly connect to AD
- AAD can connect to Windows Active Directory with Azure AD Connect
- RHEL SSSD can connect to AAD via RHEL IDM via OATH2 integration

Remember when …

Not so long ago

- Manually configure Samba
- Manually configure Winbind
- Manually configure Kerberose

- Wish upon a star ...

Easy as one command

Really, just one command

#realm join ...

Reality

Let’s dive into the process

- Supported RHEL Versions
  - 7, 8, 9
- Supported Windows Server Versions
  - 2022, 2019, 2016, 2012R2
- Required packages

  # dnf install samba-common-tools realmd oddjob oddjob-mkhomedir sssd adcli krb5-workstation samba cifs-utils

Let’s dive into the process

- Ensure your firewall is configured to allow connectivity to all the AD resources

- **DNS**: 53/udp and 53/tcp
- **LDAP**: 389/udp and 389/tcp
- **LDAPS**: 636/tcp
- **SAMBA**: 445/udp and 445/tcp
- **KERBEROS**: 88/udp and 88/tcp, 464/udp and 464/tcp
- **LDAP GLOBAL CATALOG**: 3268/tcp
- **LDAP GLOBAL CATALOG SSL**: 3269/tcp
- **NTP** (Optional): 123/udp

Let’s dive into the process

- DNS is critical – find the AD Server

```
# dig +short SRV _ldap._tcp.ad.skinnerlabs.com
    0 0 389 win2k22.
# dig +short SRV _kerberos._udp.ad.skinnerlabs.com
    0 0 88 win2k22.
# dig +short SRV _kerberos._tcp.ad.skinnerlabs.com
    0 0 88 win2k22.
```

- Test AD Server connectivity with ping?

```
# ping win2k22.ad.skinnerlabs.com
```

Let’s dive into the process

- Test AD Server connectivity with netcat

  # nc -zv win2k22.ad.skinnerlabs.com 53

  Ncat: Version 7.92 (https://nmap.org/ncat)

  Ncat: Connected to 192.168.40.239:53.

  Ncat: 0 bytes sent, 0 bytes received in 0.03 seconds.

  # nc -zv win2k22.ad.skinnerlabs.com 54

  Ncat: Version 7.92 (https://nmap.org/ncat)

  Ncat: TIMEOUT.

- Netcat each port to validate communication:

  "z = report connection status only"

  defaults to TCP, use "u" option for UDP

Let’s dive into the process

- Time is just as important

```
# chronyc sources
# systemctl stop chronyd
# chronyd -q "server win2k22.ad.skinnerlabs.com iburst"
# systemctl start chronyd
# systemctl status chronyd
# chronyc tracking
```

Let’s dive into the process

- What can we discover?

```
# realm discover ad.skinnerlabs.com
ad.skinnerlabs.com
        type: kerberos
        realm-name: AD.SKINNERLABS.COM
        domain-name: ad.skinnerlabs.com
        configured: no
        server-software: active-directory
        client-software: sssd
        required-package: oddjob
        required-package: oddjob-mkhomedir
        required-package: sssd
        required-package: adcli
        required-package: samba-common-tools
```

Let’s dive into the process

- Simple join

  # realm join ad.skinnerlabs.com

  Password for Administrator:

  See: journalctl REALMD_OPERATION=r525.29447

  realm: Couldn't join realm: Failed to join the domain

  Please check

  https://red.ht/support_rhel_ad

  to get help for common issues.

- Comment out “includedir /etc/krb5.conf.d/ from /etc/krb5.conf

- Retry join – and it will work!

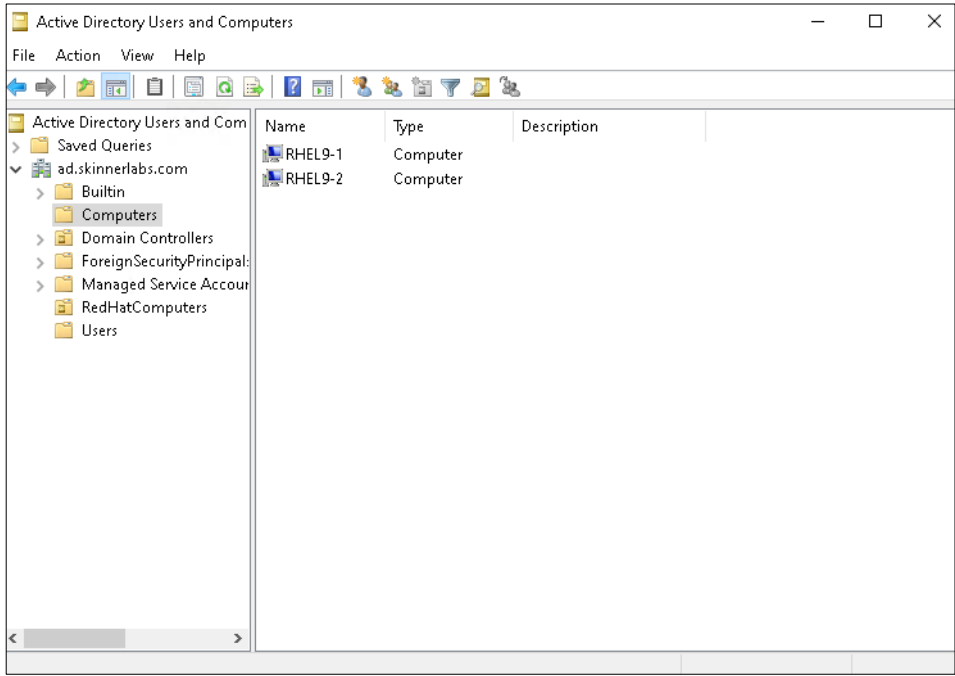
- Join with another user?

  # realm join -U DomainAdmin ad.skinnerlabs.com

Let's dive into the process

- Confirm the join was successful

```
# realm list
    ad.skinnerlabs.com
        type: kerberos
        realm-name: AD.SKINNERLABS.COM
        domain-name: ad.skinnerlabs.com
        configured: kerberos-member
        server-software: active-directory
        client-software: sssd
        required-package: oddjob
        required-package: oddjob-mkhomedir
        required-package: sssd
        required-package: adcli
        required-package: samba-common-tools
        login-formats: %U@ad.skinnerlabs.com
        login-policy: allow-realm-logins
```



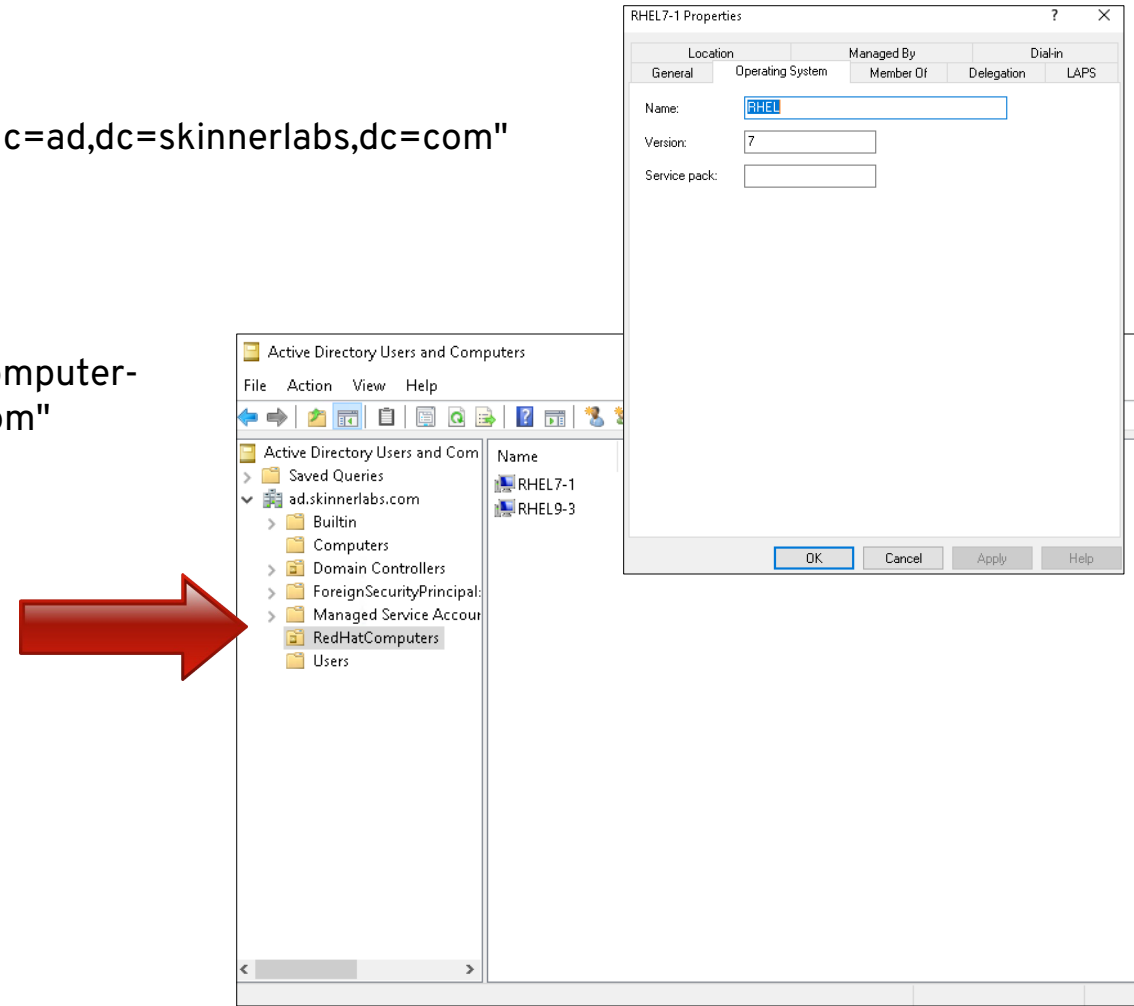
Let’s dive into the process

- Join to a specific AD OU: “RedHatComputers”

  # realm join -v --computer-ou="ou=RedHatComputers,dc=ad,dc=skinnerlabs,dc=com" ad.skinnerlabs.com

- Pass in AD attributes for OS Name and OS Version

  # realm join -v --os-name="RHEL" --os-version="7" --computer-ou="ou=RedHatComputers,dc=ad,dc=skinnerlabs,dc=com" ad.skinnerlabs.com



Let’s dive into the process

- Request AD user information

  # getent passwd dvader@ad.skinnerlabs.com

  # getent passwd dvader@ad

  dvader@ad.skinnerlabs.com:*:985401104:985400513:Darth Vader:/home/dvader@ad.skinnerlabs.com:/bin/bash

- Request AD group information

  # getent group "domain users@ad"

  domain users@ad.skinnerlabs.com:*:985400513:dvader@ad.skinnerlabs.com

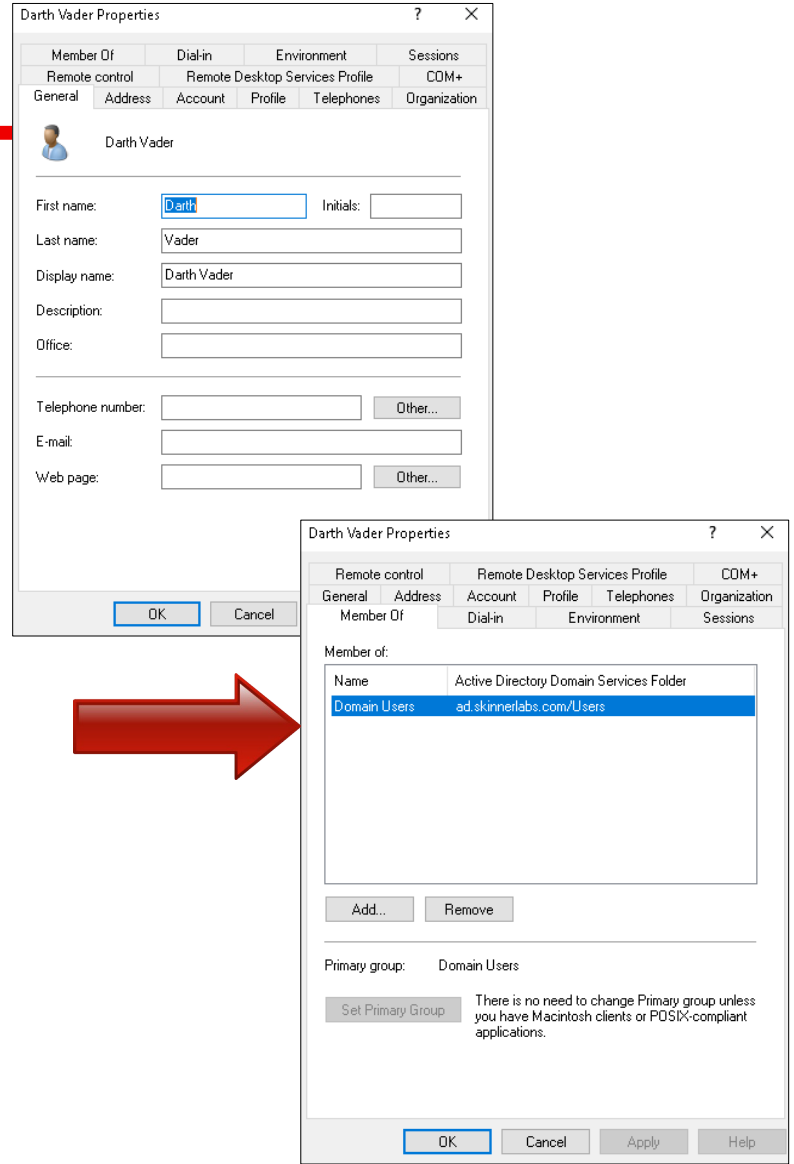
- Confirm information is the same locally

  # id dvader@ad.skinnerlabs.com

  # uid=985401104(dvader@ad.skinnerlabs.com)

  gid=985400513(domain users@ad.skinnerlabs.com)

  groups=985400513(domain users@ad.skinnerlabs.com)



Let’s dive into the process

- Add user to LinuxAdmin group for SUDO management

  # getent group "LinuxAdmin@ad"

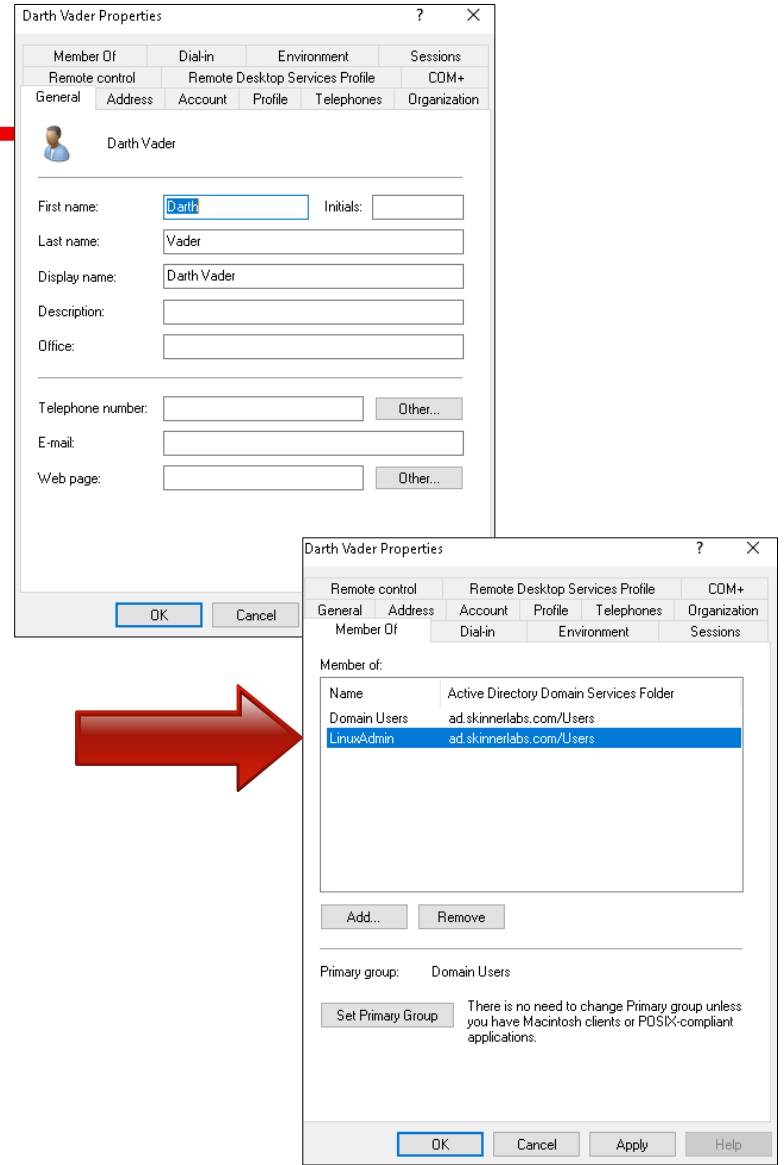
  linuxadmin@ad.skinnerlabs.com:*:985401119:dvader@ad.skinnerlabs.com

- Update SUDO file

  # visudo

  INSERT LINE:

  %LinuxAdmin@ad.skinnerlabs.com ALL=(ALL) ALL



Let’s dive into the process

- Can we change our password?

  # passwd

  Changing password for user dvader@ad.skinnerlabs.com.

  Current Password:

  New password:

  Retype new password:

  passwd: all authentication tokens updated successfully.

- Can we change it to a simple password?

  # passwd

  Changing password for user dvader@ad.skinnerlabs.com.

  Current Password:

  Password change failed. Server message: Old password not accepted.

  passwd: Authentication token manipulation error

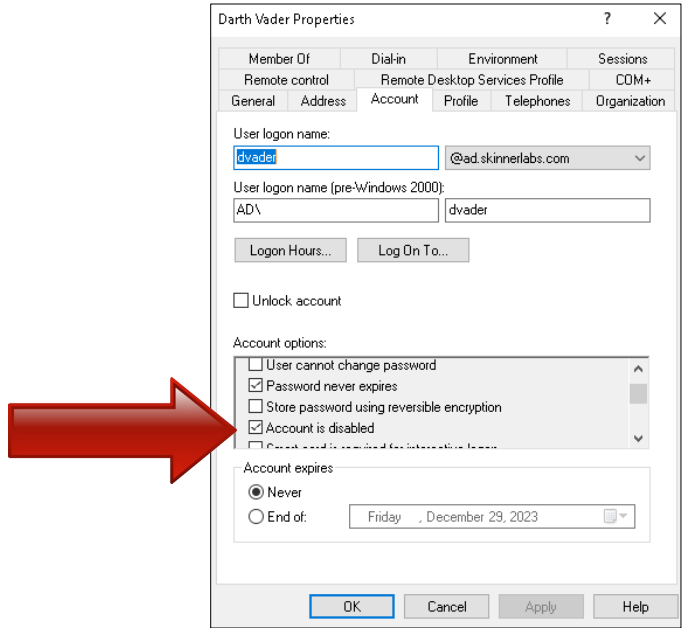
Let’s dive into the process

- Can we login is AD user is disabled?

  # ssh -l dvader@ad.skinnerlabs.com rhel9-1.rhlab.skinnerlabs.com

  dvader@ad.skinnerlabs.com@rhel9-1.rhlab.skinnerlabs.com's password:

  Permission denied, please try again.



Automating Join

AD Integration System Role

- What are system roles?
  - Red Hat provided Ansible playbooks for 28 RHEL common configurations

- Install RHEL System Roles

  # dnf -y install rhel-system-roles

- View all System Roles

  # cd /usr/share/doc/rhel-system-roles/

  # ls

ad_integration collection ha_cluster kernel_settings metrics network postgresql ssh systemd vpn certificate crypto_policies journald keylime_server nbde_client podman rhc sshd timesync cockpit firewall kdump logging nbde_server postfix selinux storage tlog

AD Integration System Role

- Documentation
  /usr/share/doc/rhel-system-roles/ad_integration/README.md

```
- hosts: all
  vars:
    ad_integration_realm: "domain.example.com"
    ad_integration_password: !vault | ...vault encrypted password...
    ad_integration_manage_crypto_policies: true
    ad_integration_allow_rc4_crypto: true
  roles:
    - rhel-system-roles.ad_integration
```

AD Integration System Role

- Encrypting AD Password for Ansible Playbook
- Create encrypted variable:

```
# ansible-vault encrypt_string --ask-vault-pass 'RedHat2023' --name 'ad_password'
New Vault password:
Confirm New Vault password:
Encryption successful
ad_password: !vault |
      $ANSIBLE_VAULT;1.1;AES256
      30343762623933306430393136323362666139653335343937353434633732353262323434383
      8316665393830396535353132386138613236646232336330613 90a6532623436666366666636
      3653866353538626230343937653362613330323465663063373761666463383930626532613
      7323964326439323335653837616631 0a366463356664316333326666653036333239336437313
      33613230663139396637
```


Red Hat

AD Integration System Role

- Example Ansible Playbook using Ansible Vault

```
# cat ad_integration.yml
---
- hosts: localhost
  vars:
   ad_password: !vault |
      $ANSIBLE_VAULT;1.1;AES256
      30343762623933306430393136323362666139653335343937353434633732353262323434383
      8316665393830396535353132386138613236646232336330613 90a6532623436666366666633
      6538663535386262303439376533626133303234656630633737616664633839306265326137
      3239643264393233356538376166310a366463356664316333326666653036333239336437313
      3361323066313939663 7
```


Red Hat

AD Integration System Role

- Playbook continued …

```
 ad_integration_realm: "ad.skinnerlabs.com"
ad_integration_password: "{{ ad_password }}"
ad_integration_manage_crypto_policies: false
ad_integration_allow_rc4_crypto: false
ad_integration_timesync_source: "win2k22.ad.skinnerlabs.com"
ad_integration_computer_ou: "ou=RedHatComputers,dc=ad,dc=skinnerlabs,dc=com"
```

```
roles:
 - linux-system-roles.ad_integration
```

- Run AD Integration RHEL System Role Playbook

  # ansible-playbook ad_integration.yml --ask-vault-pass

CIFS Mount with AD auth

Windows, please share with me

- Windows shares are mounted with the full permission (0755) in Linux. To change the default permission use the dir_mode and file_mode options to set directory and file permission.
- Domain Users GID = 985400513

- Mounting a CIFS share
  - /etc/fstab
    - //win2k22.ad.skinnerlabs.com/share /mnt/share cifs vers=3.0,credentials=/root/share.creds,_netdev,**dir_mode=0770**, **file_mode=0660**,gid=985400513 0 0
  - /root/share.creds

    username=administrator

    password=RedHat2023

    domain=AD

UID / GID Mappings

Map this way!

- Option 1:

  When SSSD detects a new AD domain, it assigns a range of available IDs to the new domain
  It uses a algorithmic hash to create a repeatable POSIX ID mapping to Windows SID.

  When all clients are SSSD this works, if not use Option 2

Map this way!

- Option 2

  Use POSIX attributes defined in AD

  AD can create and store POSIX attributes:

  uidNumber, gidNumber, unixHomeDirectory, or loginShell

  These must be configured for the users in AD

  # realm join **--automatic-id-mapping=no** ad.skinnerlabs.com

Map this way!

- If already joined to the DOMAIN,

  Edit the /etc/ssd/ssd.conf and disable the mapping:
  ldap_id_mapping = false

  Remove the SSSD cache files (the cache stores the algorithmic hash POSIX ID mappings)

  # rm -f /var/lib/sss/db/*

  # systemctl restart sssd

  AD administrator is responsible for setting uidNumber, gidNumber, unixHomeDirectory, or loginShell values in the AD User Attribute Editor

Crypto Policies

DEFAULT / AD-SUPPORT-LEGACY / FUTURE?

- SSSD supports RC4, AES-128 and AES-256 for Kerberos encryption types
- RC4 is deprecated and disabled by default

  # update-crypto-policies --set DEFAULT

- Are you rocking an older AD which requires RC4?

  # update-crypto-policies --set DEFAULT:**AD-SUPPORT-LEGACY**

- Want to tighten up security more?

  # update-crypto-policies --set FUTURE

DEFAULT / AD-SUPPORT-LEGACY / FUTURE?

- Viewing the details of your current crypto policy
  # update-crypto-policies --show
  DEFAULT

- Dump the details of the current policy
  # cat /etc/crypto-policies/state/CURRENT.pol

```
# Policy DEFAULT dump
#
# Do not parse the contents of this file with automated tools,
# it is provided for review convenience only.
#
# Baseline values for all scopes:
cipher = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CTR AES-128-CBC
group = X25519 SECP256R1 X448 SECP521R1 SECP384R1 FFDHE-2048 FFDHE-3072 FFDHE-4096 FFDHE-6144 FFDHE-8192
hash = SHA2-256 SHA2-384 SHA2-512 SHA3-256 SHA3-384 SHA3-512 SHA2-224 SHA3-224 SHAKE-256
key_exchange = ECDHE RSA DHE DHE-RSA PSK DHE-PSK ECDHE-PSK RSA-PSK ECDHE-GSS DHE-GSS
mac = AEAD HMAC-SHA2-256 HMAC-SHA1 UMAC-128 HMAC-SHA2-384 HMAC-SHA2-512
protocol =
sign = ECDSA-SHA3-256 ECDSA-SHA2-256 ECDSA-SHA2-256-FIDO ECDSA-SHA3-384 ECDSA-SHA2-384 ECDSA-SHA3-512 ECDSA-SHA2-512 EDDSA-
ED25519 EDDSA-ED25519-FIDO EDDSA-ED448 RSA-PSS-SHA3-256 RSA-PSS-SHA2-256 RSA-PSS-SHA3-384 RSA-PSS-SHA2-384 RSA-PSS-SHA3-512
 RSA-PSS-SHA2-512 RSA-PSS-RSAE-SHA3-256 RSA-PSS-RSAE-SHA2-256 RSA-PSS-RSAE-SHA3-384 RSA-PSS-RSAE-SHA2-384 RSA-PSS-RSAE-SHA3
-512 RSA-PSS-RSAE-SHA2-512 RSA-SHA3-256 RSA-SHA2-256 RSA-SHA3-384 RSA-SHA2-384 RSA-SHA3-512 RSA-SHA2-512 ECDSA-SHA2-224 RSA
-PSS-SHA2-224 RSA-SHA2-224 ECDSA-SHA3-224 RSA-PSS-SHA3-224 RSA-SHA3-224
arbitrary_dh_groups = 1
min_dh_size = 2048
min_dsa_size = 2048
min_rsa_size = 2048
sha1_in_certs = 0
ssh_certs = 1
ssh_etm = 1
__ems = DEFAULT
# Scope-specific properties derived for select backends:
cipher@gnutls = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@gnutls = TLS1.3 TLS1.2 DTLS1.2
cipher@java-tls = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@java-tls = TLS1.3 TLS1.2 DTLS1.2
mac@krb5 = HMAC-SHA2-384 HMAC-SHA2-256 AEAD UMAC-128 HMAC-SHA2-512 HMAC-SHA1
protocol@libreswan = IKEv2
cipher@libssh = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-128-GCM AES-128-CCM AES-128-CTR
cipher@nss = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@nss = TLS1.3 TLS1.2 DTLS1.2
cipher@openssh-client = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-128-GCM AES-128-CCM AES-128-CTR
cipher@openssh-server = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-128-GCM AES-128-CCM AES-128-CTR
cipher@openssl = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@openssl = TLS1.3 TLS1.2 DTLS1.2
```

DEFAULT / AD-SUPPORT-LEGACY / FUTURE?

- Customize Crypto Policy
- Examples: /usr/share/crypto-policies/policies

  # vi /etc/crypto-policies/policies/modules/**SKINNERLABS**.pmod

  min_rsa_size = 4096

- Activate new Crypto Policy

  # update-crypto-policies --set DEFAULT:**SKINNERLABS**

  # update-crypto-policies --show

  DEFAULT:**SKINNERLABS**

```
arbitrary_dh_groups = 1
min_dh_size = 2048
min_dsa_size = 2048
min_rsa_size = 4096
sha1_in_certs = 0
ssh_certs = 1
ssh_etm = 1
__ems = DEFAULT
# Scope-specific properties derived for select backends:
cipher@gnutls = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@gnutls = TLS1.3 TLS1.2 DTLS1.2
cipher@java-tls = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@java-tls = TLS1.3 TLS1.2 DTLS1.2
mac@krb5 = HMAC-SHA2-384 HMAC-SHA2-256 AEAD UMAC-128 HMAC-SHA2-512 HMAC-SHA1
protocol@libreswan = IKEv2
cipher@libssh = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-128-GCM AES-128-CCM AES-128-CTR
cipher@nss = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@nss = TLS1.3 TLS1.2 DTLS1.2
cipher@openssh-client = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-128-GCM AES-128-CCM AES-128-CTR
cipher@openssh-server = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CTR AES-128-GCM AES-128-CCM AES-128-CTR
cipher@openssl = AES-256-GCM AES-256-CCM CHACHA20-POLY1305 AES-256-CBC AES-128-GCM AES-128-CCM AES-128-CBC
protocol@openssl = TLS1.3 TLS1.2 DTLS1.2
```

- Dump the details of the current policy

  # cat /etc/crypto-policies/state/CURRENT.pol

Dynamic DNS Updates

Tuning SSSD Dynamic DNS

- If your AD DOMAIN accepts dynamic DNS updates
- SSSD will attempt to update the DNS record:
  - Every time SSSD comes online
  - Using the **dyndns_refresh_interval** option in the /etc/sssd/sssd.conf file
    - (default is 86400 seconds = 24 hours)
- Two timing config options to use:

  dyndns_refresh_interval = 43200

  dyndns_ttl = 3600
- Disable Dynamic DNS on client side:

  dyndns_update_ptr = false

Leave the Domain

Disconnecting from AD

- How to leave AD

  # realm leave

  # id dvader

  id: dvader: no such user

- Still need to manually delete the computer resource on the AD side

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.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat