Evolution of the Linux system identity and authentication stack

Dmitri Pal
Director, Software Engineering, Red Hat, Inc.
09-May-2019
Preamble
It is really a tale!
Once upon a time...

PAM + NSS
Over the time...

PAM + NSS
By RHEL 7
... instead of
The Story...
PAM & NSS
- Connects Linux system to **central identity stores** (IdM, AD, LDAP)
- All information is **cached locally** for offline use
- **Advanced integration** with IdM and AD, integration with Linux (SUDO, SELinux, 2FA)
Evolution of the modules

RHEL-7
- libnss_sss.so.2
- pam_sss.so
- pam_krb5.so
- pam_pkcs11.so
- libnss_ldap.so
- pam_ldap.so

RHEL-8.0
- libnss_sss.so.2
- pam_sss.so
- pam_krb5.so
- pam_pkcs11.so
- pam_ldap.so

Deprecated
pam_krb5

- SSSD has already a fully sufficient Kerberos integration
- Caveats
  - SSSD does not have **AFS integration** - can be resolved with [pam-afs-session](#)
- pam_krb5 ⇒ /dev/null
- **Migration**: see [Migrating from pam_krb5 to sssd](#)
pam_pkcs11

- pam_pkcs11 was abandoned in the upstream
- SSSD already had functional Smart Card integration in RHEL-7:
  - RHEL-7.2: initial Smart Card support in IdM and SSSD (cert in user identity)
  - RHEL-7.4: Smart Card mapping and matching rules (rules in IdM Server or locally)
    - In AD or local cases the full certificate had to be added as an attribute to the user entry
pam_pkcs11 (continued)

- To cover *pam_pkcs11-exclusive* features, RHEL-8.0 introduces:
  - Certificate mapping and matching rules for any SSSD domain (local and AD users in direct integration)
    - The full cert no longer needs to be stored in the user entry
  - Smart Card authentication for *local users* (no IdM or AD)
- `pam_pkcs11 => /dev/null`
- **Migration:** see [Migrating from pam_pkcs11.conf](#)
nss_ldap & pam_ldap

- Will be removed at next major release, bug fix only in RHEL-8
- SSSD already contains functionality for the major nss-pam-ldapd use cases
  - nss-pam-ldapd is only recommended for very specific use cases that SSSD does not cover
- **Customer Knowledge Base**
  - [What is the support status for nss-pam-ldapd and NIS packages in Red Hat Enterprise Linux?](#)
Kerberos Credential Manager

- SSSD now introduces **Kerberos Credential Manager** service
  - Parallel Kerberos `ccache` storage to File/Directory/Keyring ([ccache types](#))
  - Default in RHEL-8.0

- **Benefits**
  - Subject to UID namespaces, unlike Kernel keyring (useful in containers)
  - Stateful daemon, will enable us to perform advanced tasks like `ccache` renewals in one of following releases

- **Possible future ideas**
  - **RFE** To delete kerberos tickets once the user logs out
  - **RFE** Expand kerberos ticket renewal
authselect

- Brand new tool **replacing authconfig**
  - **Main motivation**: Administrator no longer builds a PAM stack by a tool (potentially ending with broken configuration), but rather **selects** a tested PAM profile
  - Other motivations: authconfig was a dated component (initiated back in 1999), with no Python 3 support and deprecated GUI (python 2.7 will EOL support in 2020)

- **Benefits**
  - **Properly tested profiles** - lower risk of lock out
  - **Clarity and quality** - profiles are easy to read, modify and test
  - **Custom profiles** - allows Administrator to create and ship own profiles
    - In `/etc/authselect/custom/`
  - **Smaller footprint** - written in C
authselect (continued)

- NIS  
  *deprecated*

- SSSD  
  *recommended (default)*

- Winbind

RHEL-8.0

- /etc/nsswitch.conf
- /etc/pam.d/fingerprint-auth
- /etc/pam.d/password-auth
- /etc/pam.d/postlogin
- /etc/pam.d/smartcard-auth
- /etc/pam.d/system-auth
...
authselect (continued)

- **Scope**: configures authentication and identity resources
  - Generates /etc/nsswitch.conf and PAM configuration from selected profile
  - Does NOT configure actual PAM modules
    - Done by ipa-client-install, realmd, Ansible
- **Compatibility**: For applications, scripts and kickstarts that were relaying on the authconfig there is now a wrapper around authselect
  - It is **translating** calls to authconfig into calls to authselect
  - Not all options are supported but the main ones are
custodia

- Part of the attempt to build secrets management API
  - Was intended to provide an API to serve applications and container use cases
  - Too little too late and was not a part of the Kubernetes ecosystem
- custodia \(\Rightarrow /dev/null\)
  - Note: there is still an internal version used for secrets sharing in IdM
sssd-secrets

- Local REST API interface to store secrets
  - Part of the attempt to build secrets management API
  - Was targeting developers
- sssd-secrets ⇒ /dev/null
hesiod future

- NSS module that looks passwd entries in a DNS server - TXT records
- Has been created in 80s
- Have not been used since LDAP emerged couple decades ago
- It is a part of the Red Hat support promise so we can’t remove it for 3 major versions
- hesiod is now deprecated and will be removed in RHEL 10
LDAP
openldap

- Openldap client libraries are a part of the platform
- Openldap server component has been deprecated since 7.4
- openldap-server ⇒ /dev/null
  - Motivation:
    - Better support guarantees
  - Options:
    - IdM - free with RHEL subscription
    - Red Hat Directory Server - a separate product
    - Partner offering
Apache modules
# Apache Modules

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Authentication</th>
<th>Access check</th>
<th>Extra user info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerberos</td>
<td><code>mod_auth_gssapi</code>&lt;br&gt;<code>(mod_auth_kerb)</code></td>
<td><code>mod_authnz_pam</code></td>
<td><code>mod_lookup_identity</code></td>
</tr>
<tr>
<td>Certificate</td>
<td><code>mod_ssl</code>&lt;br&gt;<code>(mod_nss, mod_revokator)</code></td>
<td><code>mod_authnz_pam</code></td>
<td><code>mod_lookup_identity</code></td>
</tr>
<tr>
<td>Formas based</td>
<td><code>mod_intercept_form_submit</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAML</td>
<td></td>
<td><code>mod_auth_mellon</code></td>
<td></td>
</tr>
<tr>
<td>OpenID Connect</td>
<td></td>
<td><code>mod_auth_openidc</code></td>
<td></td>
</tr>
</tbody>
</table>
mod_auth_kerb

- Not well maintained in community, spaghetti code
- mod_auth_gssapi have been available for several years in RHEL7 and completely replaces mod_auth_kerb
- mod_auth_kerb ⇒ /dev/null
mod_nss & mod_revocator

- Reliance on the NSS crypto library is reduced throughout the platform
- mod_ssl is much more popular and has an active community
- mod_ssl and mod_nss were continuously colliding in customer setups
- NSS tooling is cumbersome to use
- Result:
  - mod_nss ⇒ /dev/null
  - mod_revocator is ⇒ /dev/null too,
    - Module was checking CRL for revocation
    - There is no matching functionality in mod_ssl
Other notable changes
Other packages gone

- Old python libraries that allow Kerberos 5 authentication and security for Python applications:
  - python-kerberos $\Rightarrow$ python-gssapi
  - python-krbV $\Rightarrow$ /dev/null
  - python-requests-kerberos $\Rightarrow$ /dev/null
NIS

- NIS related components are deprecated since 7.6 but still there for interoperability purposes
  - ypserv, ypbind, portmap, yp-tools
- Moved to Application Stream for a more flexible support cycle
- A couple of changes:
  - Sun RPC and NIS interfaces removed from glibc → libtirpc
  - This also allows IPv6 support
- NIS client is there to stay to support huge HPC cluster cases
- NIS server is intended to be removed in RHEL 9
IdM and its client are a module

- RHEL-8.0 introduces a support for **Application Streams** - including **IdM Server** and **Client**
- New “**idm**” module **streams** and **profiles**:
  - **idm:DL1** (+ pki-core, pki-ds, 389-ds) - IdM Server (Domain Level 1)
  - idm:client - default stream
- **Benefits**
  - Clearly defined packaging and public interfaces
  - Safer upgrades - whole (tested) collection of packages, not piecemeal
- **Red Hat resources**
  - Documentation
    - 5.5. Packages required for an Identity Management server - list of modules
Moving forward...
Winbind & SSSD

- Winbind or SSSD - which to use?
  - SSSD for most cases
  - Winbind is still there - it is needed for Samba FS

- Challenges in 7.6 - Samba FS in IdM domain with SSSD integration
  - Winbind is needed but does not support IdM and trust integration
  - SSSD is needed for the IdM and trust integration
  - So far they can’t coexist - work is underway to make them complementary
What about containers?

- We have SSSD container in RHEL 7 - fully supported
- We have IdM container in RHEL 7 - tech preview
- We do not have either container in RHEL 8 yet
- Before we do containers in RHEL 8 we want to better understand:
  - Who is (will be) using those containers?
  - Why these containers are appealing?
  - What problem do they solve?
    - Feedback is needed
    - Please get in touch with us
Other improvements

- SSSD REST API for containers for cloud native use cases
- SSSD code cleanup in preparation for FIPS
- SSSD configurable prompt for two-factor auth with RADIUS
- Authconfig 1.1 to address community and user feedback
- mod_auth_openidc configuration via keycloak-httpd-client-install
Questions?
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat