Closing the KDC Blindspot

Enforcing strong authentication across the trust boundary between FreeIPA and Active Directory

Alexander Bokovoy · Red Hat / Samba Team

SambaXP 2026

The Problem

Active Directory domain controllers cannot tell *how* an AD user authenticates

- Was it done with a smartcard or a password?
- MS-PAC structure may contain *some* information to infer but no details

For other methods no information how user authenticated can be derived at all

- Entra ID issues Kerberos tickets that have 'pre-authentication: false' flag even if users were authenticated with MFA

Authentication Mechanism Assertion

- Obscure Active Directory feature since 2008
- Requires ADFS to setup and use

SID derivation from the certificate extensions:

- if a smartcard certificate contains certain OID extension
  - a group membership is recorded in the LOGON_INFO buffer in PAC
  - membership is dynamic: the group is empty in AD

Sample configuration: U.S. General Services Administration's ficam-scripts-public Github repository

The Problem

Protocol transition (S4U2Self): trust into service is limited

- An web server authenticates a user with an x.509 certificate, then calls S4U2Self extension on their behalf.
- The KDC receives `PA-FOR-USER` — a plaintext username or `PA-FOR-X509-USER` with x.509 certificate.

No method signal, no proof the user authenticated at all. AMA feature cannot be used here.

Applications cannot enforce authentication strength for either case

- "This resource is accessible only for users who authenticated *this* way"

Kerberos Authentication Indicators (RFC 8129)

MIT Kerberos implements 'authentication indicators': strings in `AD-AUTHENTICATION-INDICATOR` authorization data in the ticket.

Enforcement:

- **KDC**: service ticket issuance
- **Applications**: `mod_auth_gssapi` (Apache) / `pam_sss_gss` (PAM)

FreeIPA

FreeIPA supports a number of indicators:

Indicator	Meaning
`pkinit`	Certificate / smartcard
`otp`	Two-factor (OTP)
`radius`	RADIUS-based
`passkey`	FIDO2 passkey-based
`idp`	OAuth2 device authorization grant flow-based
(custom)	Site-defined

Active Directory trust to FreeIPA

- Windows Server-issued Kerberos tickets
  - Contain MS-PAC structures with group membership information
  - Have no authentication indicators support (Windows Server has no support for RFC 8129)
- FreeIPA-issued Kerberos tickets
  - Contain MS-PAC structures
  - Have authentication indicators support

Can we convert PAC marks to authentication indicators on the trust boundary?

History hindsight

MIT Kerberos PR#965

- "Allow to modify authentication indicators as part of sign_authdata process"
- August 2019, from Greg Hudson:

> Alexander said he would attempt an experiment using FreeIPA and Samba, so I will likely wait for that before merging.

- Available in MIT Kerberos 1.18, Greg merged it before I completed experiments six years later

S4U2Self Protocol Variants

Variant	Padata type	KDC sees	Indicators
Plain S4U2Self	`PA-FOR-USER`	username string	not possible
Attested S4U2Self	**`PA-FOR-X509-USER`**	**X.509 cert + auth context**	**yes**
S4U2Proxy	constrained delegation TGS-REQ	ticket chain	inherited

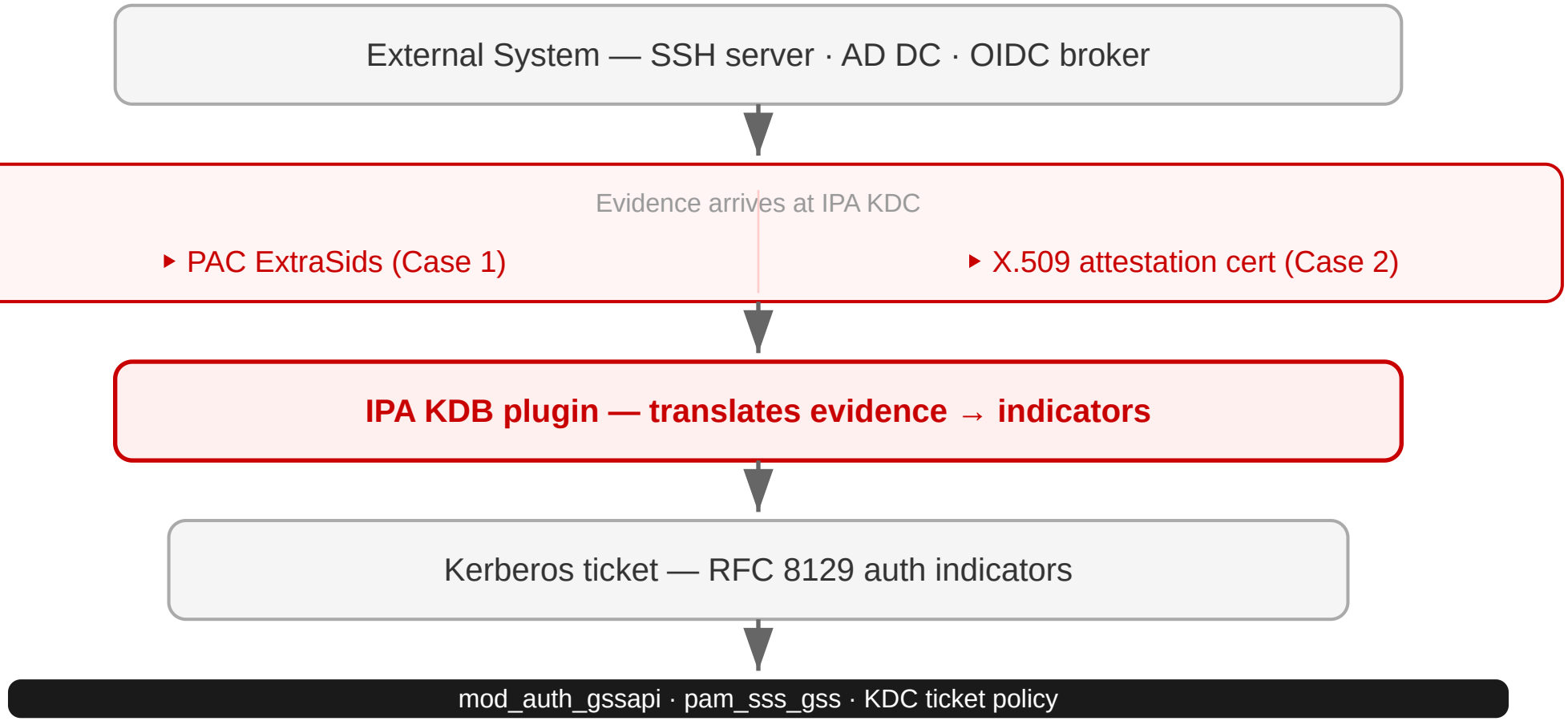
Plain S4U2Self — service asserts a username; KDC cannot verify the claim, cannot see the auth method, and cannot add indicators.

Attested S4U2Self — service presents a short-lived certificate encoding the auth event; KDC verifies and injects indicators.

S4U2Proxy — service uses the S4U2Self ticket to request tickets for further services (constrained delegation); inherits any indicators from the S4U2Self step.

The Shared Pattern

Both features implement the same architecture in the IPA KDB plugin:



IPA does **not** re-verify the original authentication — it verifies the *evidence* and maps it to indicators.

Case 1

AD Trust: SID → Authentication Indicator Mapping

Active Directory Authentication Mechanism Assurance

When a user authenticates with a **smartcard**, the AD DC:

1. Validates the certificate chain against trusted CAs
2. Extracts Issuance Policy OIDs from the certificate
3. Looks up configured OID → security group mappings
4. Injects matched group SIDs into the PAC (`ExtraSids`)

These memberships are **ephemeral** — the same user with a password gets a different PAC.

FreeIPA's KDC already processes the PAC for SID filtering.

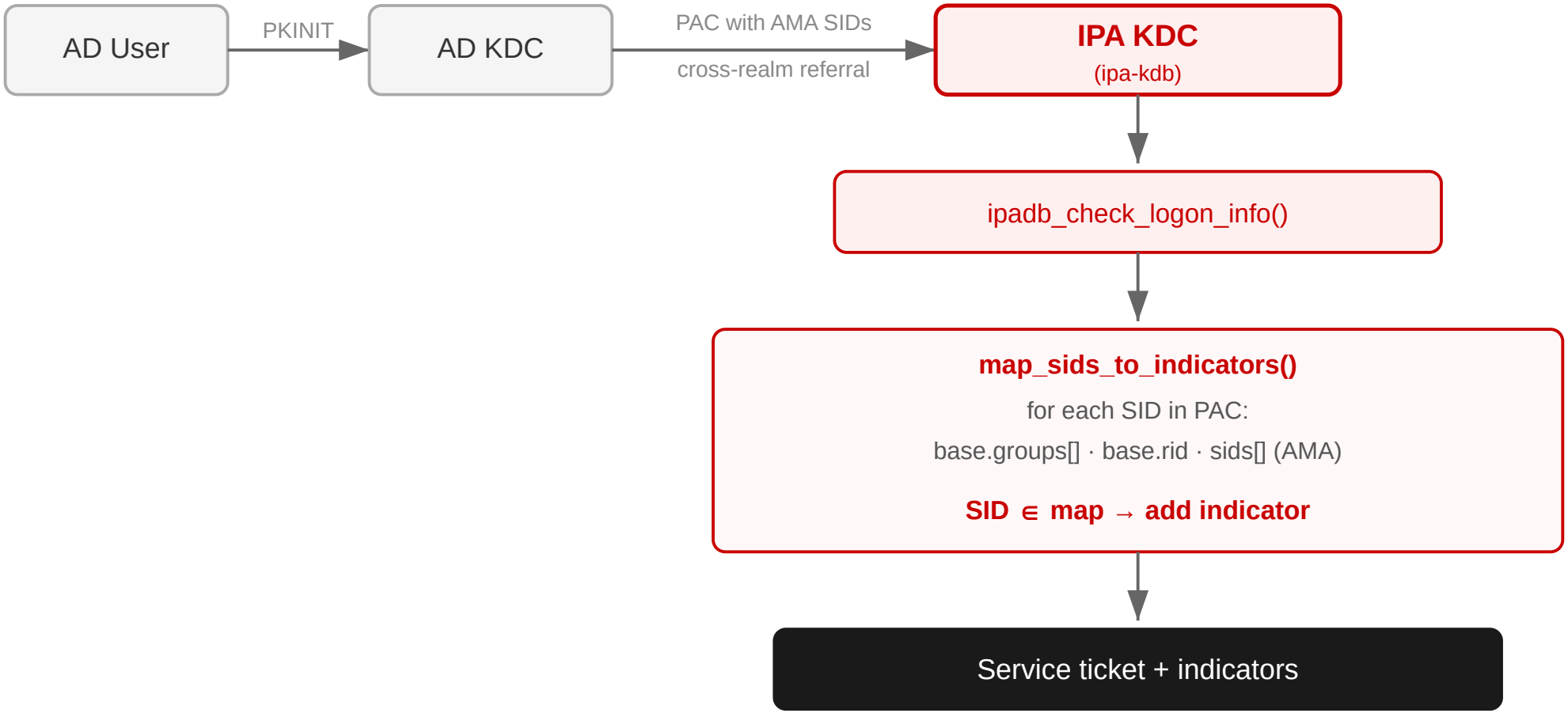
New: scan SIDs → add Kerberos indicators on a configured match.

AMA: What Lands in the PAC

```
PAC LOGON_INFO:
  base.groups[]   ← permanent AD group memberships (RIDs)
  base.rid        ← user's primary group RID
  sids[]          ← extra SIDs — AMA groups appear here
    S-1-5-21-...-512   Domain Admins  (permanent)
    S-1-5-21-...-1234  "fpki-common-High"  (AMA, only w/ smartcard)
```

If a user's PAC contains SID X → add Kerberos indicator Y

Case 1: Architecture



Case 1: LDAP Schema and CLI

New schema (`60basev5.ldif`):

```
attributeTypes: ( 2.16.840.1.113730.3.8.27.2
  NAME 'ipaSIDIndicatorMap'
  SYNTAX DirectoryString )   ← multi-valued: SID:indicator:bool
```

CLI (`ipaserver/plugins/trust.py`):

```
# Map by group name (resolved to SID automatically)
ipa trust-mod ad.example.com \
  --addattr=ipasidindicatormap="Domain Admins:pkinit:true"

# Map by explicit SID (required for AMA groups)
ipa trust-mod ad.example.com \
  --addattr=ipasidindicatormap=="S-1-5-21-...-1234:fpki-high:true"
```

Format: `<group-name-or-SID> : <indicator> [: <smartcard-flag>]`, `smartcard=true` forced additional indicator `pkinit` (MS-PKCA §3.1.5.2)

Example: Federal PKI / DoD CAC

AD AMA maps Federal PKI certificate OIDs to temporary group SIDs:

OID suffix	Policy name	Indicator
`.13`	id-fpki-common-authentication	`fpki-auth`
`.16`	id-fpki-common-High	`fpki-high`
`.7`	id-fpki-common-hardware	`fpki-hardware`
`.41`	id-fpki-common-pivAuth-derived-hardware	`fpki-piv-hw`

```
ipa trust-mod dod.mil \
  --addattr=ipasidindicatormap=="id-fpki-common-High:fpki-high:true"
```

AD validates the certificate: IPA maps the AMA SID to the indicator.

With password auth: PAC never contains the AMA SID → no indicator → denied.

Case 2

S4U2Self: X.509 Attestation Certificates

Case 2: The Attestation Certificate

```
X.509 v3 (valid ≤ 300 s) {
  Issuer:  CN = host/hostname@REALM
  Subject: CN = <username>
  SPKI:    user's auth key (publickey) or ephemeral key (password)

  Extensions:
    subjectAltName (critical): PKINIT SAN — KRB5PrincipalName

    id-ce-kerberosServiceIssuerBinding  (2.16.840.1.113730.3.8.15.3.1)
      { serviceType, principal, enctype, kvno,
        sigAlg, serviceKey SPKI, binding-signature }

    id-ce-sshAuthnContext  (2.16.840.1.113730.3.8.15.3.2)  [SSH]
      { authMethod, sessionId, keyFingerprint, clientAddress }

    id-ce-oidcAuthnContext  (2.16.840.1.113730.3.8.15.3.3)  [OIDC]
      { issuer, accessTokenHash, amrValues, clientAddress }
}
```

Case 2: Key Derivation

Both client and KDC independently compute the same signing key — no key distribution needed

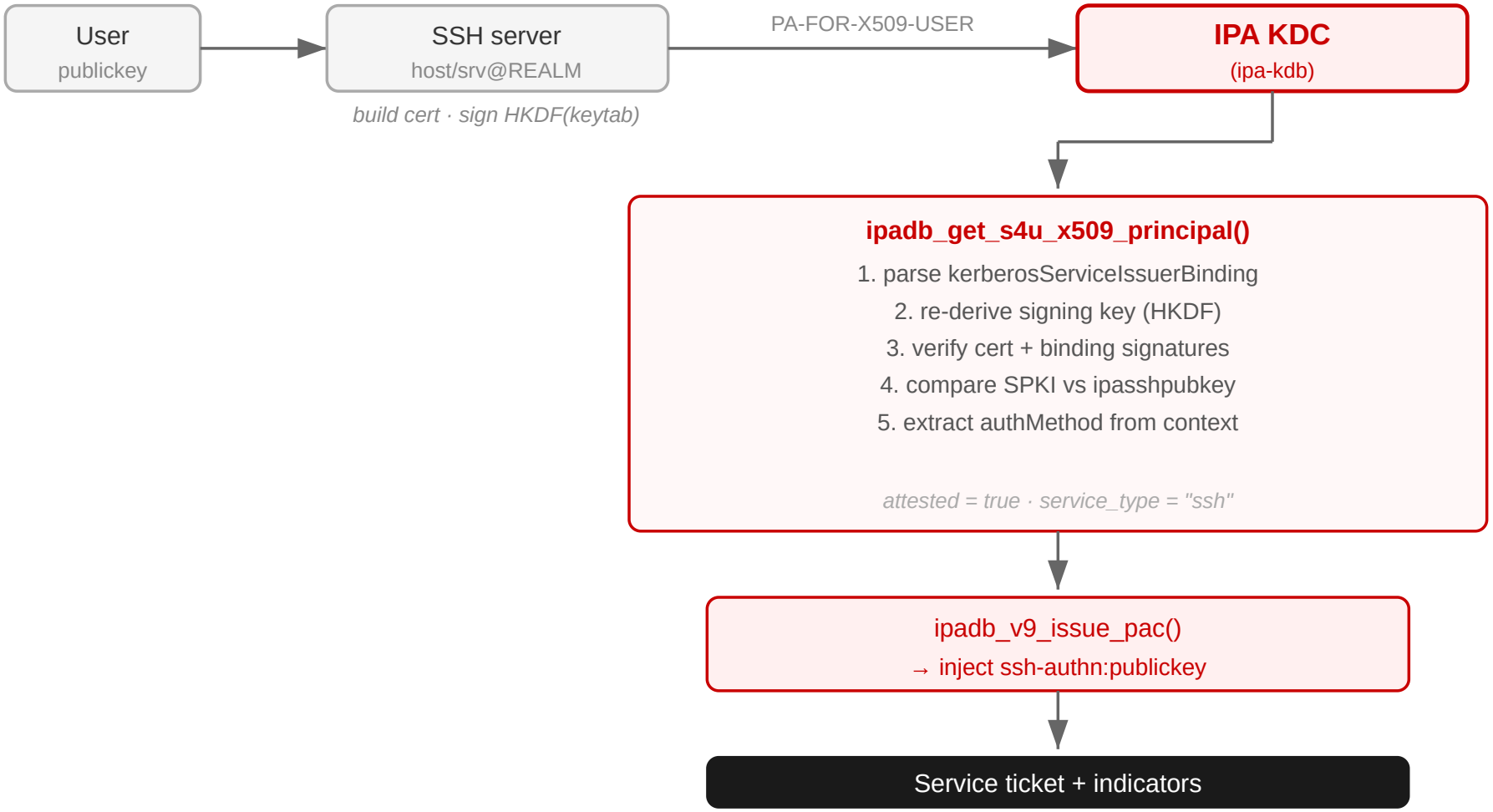
```
IKM   = host keytab key (raw bytes for enctype+kvno)
salt  = "ssh-attestation-v1"   (domain-separates per service type)
info  = hostname || realm || kvno (big-endian)

→ 32 bytes → Ed25519 seed          (non-FIPS)
→ 48 bytes → EC P-256 scalar       (FIPS mode)
```

Binding signature proves the keytab holder endorses the specific registered service key (SSH host key, TLS key, attestation key):

```
digest = SHA256(serviceKey_DER || "ssh-attestation-binding-v1"
                || principal || kvno)
sig    = Sign(derived_key, digest)
```

Re-keying (KVNO increment) rotates signing keys automatically.

Case 2: Architecture



OpenSSH: S4U2Self/Proxy Support

Not in upstream OpenSSH — downstream patch maintained by Red Hat.

Available in: **Fedora 45** · **RHEL 10.3** · **RHEL 9.9**

After a user authenticates via any method, `sshd-session` calls `ssh_gssapi_s4u2self()`:

1. Requests `PA-FOR-USER` to the IPA KDC → S4U2Self ticket without indicators
2. Optionally performs S4U2Proxy — constrained delegation to configured services
3. Places resulting credentials in the user's Kerberos ccache for the session
4. Shell (and applications) can use Kerberos tickets

```
# sshd_config
GSSAPIAllowS4U2Self       8h
GSSAPIProxyS4U2Services   cifs/stor.example.com@REALM
```

A privilege-separated `sshd-gssapi-helper` handles keytab access and S4U2Self/Proxy on behalf of `sshd-session`, then drops to the user's UID.

OpenSSH: S4U2Self x.509 attestation certificate

Not in upstream OpenSSH — downstream patch maintained by Red Hat.

Currently under development

After a user authenticates via any method, `sshd-session` calls `ssh_gssapi_s4u2self()`:

1. Builds X.509 attestation cert encoding the SSH auth event
2. Presents `PA-FOR-X509-USER` to the IPA KDC → ticket with indicators
3. Optionally performs S4U2Proxy — constrained delegation to configured services
4. Places resulting credentials in the user's Kerberos ccache for the session

```
# sshd_config
GSSAPIAllowS4U2Self       8h
GSSAPIProxyS4U2Services   cifs/stor.example.com@REALM HTTP/critical.example.com@REALM
```

S4U2Self ticket will have a per-service configured indicator

Case 2: Authentication Indicators

Naming scheme: `<serviceType>-authn:<detail>`

Service	Auth method	Indicator
SSH	public-key	`ssh-authn:publickey`
SSH	password	`ssh-authn:password`
SSH	keyboard-interactive	`ssh-authn:keyboard-interactive`
OIDC	(per RFC 8176 `amr`)	`oidc-authn:pwd`, `oidc-authn:otp`, `oidc-authn:mfa`
OIDC	no `amrValues`	`oidc-authn:sso`
generic	no context	`<service>-authn:unknown`

Multiple OIDC AMR values → one indicator each.

Existing PKINIT indicators are preserved and combined.

Case 2: Schema and CLI

New schema (non-SSH services; SSH uses existing `ipasshpubkey`):

```
attributeTypes: ( 2.16.840.1.113730.3.8.15.2.4
  NAME 'ipaKrbServiceAttestationKey'
  SYNTAX OctetString )   ← DER SubjectPublicKeyInfo, multi-valued

objectClasses: ( 2.16.840.1.113730.3.8.24.11
  NAME 'ipaKrbServiceAttestation' SUP top AUXILIARY
  MAY ( ipaKrbServiceAttestationKey $ ipaKrbServiceAttestationType ) )
```

```
# Register attestation key (OIDC broker; SSH is automatic)
ipa service-add-attestation-key "oidc/broker.example.com@REALM" \
    --type=oidc --pub-key-file=broker-attest-pub.pem

# Require SSH public-key indicator for a backend Web service
ipa service-mod HTTP/api.example.com@REALM --auth-ind ssh-authn:publickey

# Set per-indicator ticket lifetime
ipa krbtpolicy-mod --ssh-authn-maxlife=28800
```

Shared Enforcement Layer

Both features produce RFC 8129 indicators enforced identically:

At KDC (service ticket issuance):

```
krbAuthIndMaxTicketLife;ssh-authn--publickey    28800
krbAuthIndMaxTicketLife;oidc-authn--mfa         43200
krbAuthIndMaxTicketLife;fpki-high               86400
```

At application (mod_auth_gssapi / Apache):

```
AuthGSSAPIIndicatorsRequired ssh-authn:publickey
```

At login (pam_sss_gss / SSH):

```
pam_gssapi_indicators_map sudo:ssh-authn:publickey
```

One enforcement mechanism. Two sources of indicators.

Security Model: Case 1 — SID → Indicator

IPA trusts:

- AD's certificate chain validation and AMA group assignment

IPA independently verifies:

- PAC signature before any mapping runs
- SID must match an explicitly configured entry — no wildcards
- `smartcard=true` forces `indicator=pkinit` at config time
- Only `trust admins` members may configure mappings (LDAP ACI)

Security Model: Case 2 — S4U X.509 Attestation

IPA trusts:

- The service's authentication of the user

IPA independently verifies:

- Re-derives signing key and verifies cert outer signature
- Verifies binding signature — prevents key-substitution attacks
- Compares service SPKI against LDAP-registered keys
- For `publickey` auth: compares cert SPKI against user's `ipasshpubkey`
- Certificate validity ≤ 300 seconds — limits replay window

Security Model: Shared Limitations

- Indicators are lost on password-based ticket renewal
- Certificate revocation not re-checked after ticket issuance
- Only `trust admins` / service owners may configure mappings (LDAP ACIs)

Status: Case 1 — SID → Indicator

Done:

- LDAP schema · ACIs · IPA framework · KDB backend

Status: Case 2 — S4U X.509 Attestation

Done:

- IPA KDB plugin · Python attestation library
- LDAP schema · service plugin · ticket policy integration
- OpenSSH integration · out-of-process `sshd-gssapi-helper`

Questions?

Alexander Bokovoy · `abokovoy@redhat.com` / `ab@samba.org`