Enterprise desktop at home with FreeIPA and GNOME

Alexander Bokovoy (abokovoy@redhat.com)

Enterprise?

* almost

local office network is not managed by a company's IT department

* almost

company services' hosting is cloudy

there is no one cloud to rule them all

* almost

I have FEW identities:

- A corporate identity for services sign-on

I want them to be usable at the same time

* almost

I have FEW identities:

- A corporate identity for services sign-on
- Home-bound identity to access local resources

I want them to be usable at the same time

* almost

I have FEW identities:

- A corporate identity for services sign-on
- Home-bound identity to access local resources
- Cloud-based (social networking) identities

I want them to be usable at the same time

* almost

I have FEW identities:

- A corporate identity for services sign-on
- Home-bound identity to access local resources
- Cloud-based (social networking) identities
- Free Software hats to wear

I want them to be usable at the same time

* almost

I have FEW identities:

- A corporate identity for services sign-on
- Home-bound identity to access local resources
- Cloud-based (social networking) identities
- Free Software hats to wear
- Certificates and smart cards to present myself legally

I want them to be usable at the same time

* almost

I have FEW identities:

- A corporate identity for services sign-on
- Home-bound identity to access local resources
- Cloud-based (social networking) identities
- Free Software hats to wear
- Certificates and smart cards to present myself legally
- Private data to protect and share

I want them to be usable at the same time

I work on FreeIPA, `https://www.freeipa.org`

Management of identities and policies:

- stored centrally
- applied locally

And it is available in:

- Fedora
- Red Hat Enterprise Linux / CentOS
- GNU/Linux Debian and Ubuntu
- `https://account.gnome.org/` runs FreeIPA since october 2014

How *enterprisey* are we?

Let's score by a password

Let's score by a password

A typical workflow for every laptop reboot

1. Sign into a local system account (enter a password)

Let's score by a password

A typical workflow for every laptop reboot

1. Sign into a local system account (enter a password)
2. Jump onto virtual private network (enter a password or more)

Let's score by a password

A typical workflow for every laptop reboot

1. Sign into a local system account (enter a password)
2. Jump onto virtual private network (enter a password or more)
3. Obtain initial Kerberos credentials (enter a password)

Let's score by a password

A typical workflow for every laptop reboot

1. Sign into a local system account (enter a password)
2. Jump onto virtual private network (enter a password or more)
3. Obtain initial Kerberos credentials (enter a password)
4. Use corporate applications (enter a password?)

Can we do better than this?

how far are we from

- Sign into a corporate environment
- Use corporate applications

?

Let's try to login!

Demo of interactive logon

What was that?

- The system is configured to be a client for FreeIPA

What was that?

- The system is configured to be a client for FreeIPA
- SSSD handles login and Kerberos keys

What was that?

- The system is configured to be a client for FreeIPA
- SSSD handles login and Kerberos keys
- Login to the system is verified over public network using a proxy for Kerberos protocol

What was that?

- The system is configured to be a client for FreeIPA
- SSSD handles login and Kerberos keys
- Login to the system is verified over public network using a proxy for Kerberos protocol
- Established VPN connection based on Kerberos ticket

What was that?

- The system is configured to be a client for FreeIPA
- SSSD handles login and Kerberos keys
- Login to the system is verified over public network using a proxy for Kerberos protocol
- Established VPN connection based on Kerberos ticket
- **Credentials were entered only once**

Kerberos proxy

Available on the client side with Microsoft Active Directory and MIT Kerberos 1.13

- protocol is called MS-KKDCP
- transparent for Kerberos library users

Kerberos proxy is implemented by FreeIPA 4.2, OpenConnect Server 7.05, and as a standalone server

- Requires HTTPS connection, set up by default in FreeIPA 4.2, very easy to use (one line change on the client)
- Allows to obtain tickets from anywhere
- SSSD 1.12+
- GNOME project has enabled KDC proxy support in `https://account.gnome.org` to allow use of Kerberos credentials for SSH accounts for GNOME developers

VPN and Kerberos

OpenConnect client supports GSSAPI negotiation

- Fedora 22+ works out of the box

OpenVPN does not support GSSAPI negotiation

- to do since 2005

Could we enforce stronger authentication at a VPN edge?

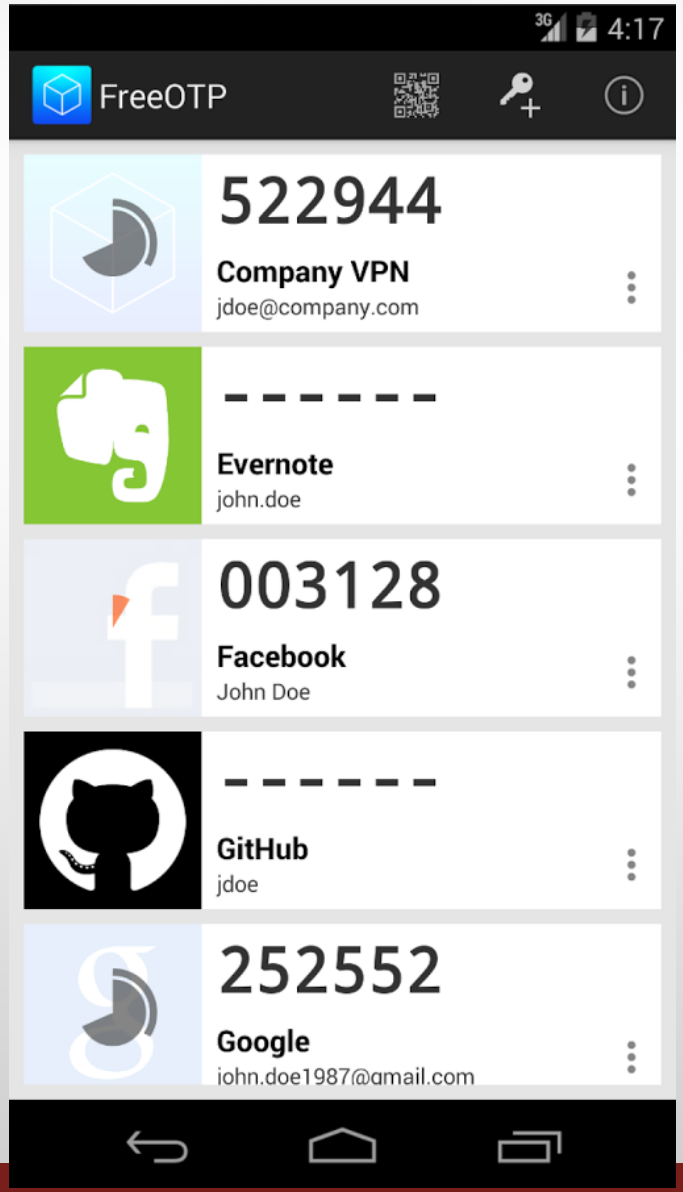
- yes, we are be able to do so with Kerberos 1.14
  - no practical implementation in FreeIPA yet

Two-factor authentication

FreeIPA 4.x supports 2FA natively

- Yubikey, FreeOTP client for Android and iOS, any HOTP/TOTP compatible software and hardware
- Two-factor authentication is enforced on Kerberos level
- Performs pre-authentication before issuing a ticket
- Authentication Indicators are in Kerberos 1.14
- Pre-authentication modules can say *how* tickets were issued

FreeOTP client for Android and iOS



Demo of interactive logon with 2FA

Let's create a token for a user and logon with 2FA via Yubikey

What was that?

1. One time password token was programmed to Yubikey and added for the user in FreeIPA

What was that?

1. One time password token was programmed to Yubikey and added for the user in FreeIPA
2. SSSD handles login and notices OTP pre-authentication support in Kerberos conversation

What was that?

1. One time password token was programmed to Yubikey and added for the user in FreeIPA
2. SSSD handles login and notices OTP pre-authentication support in Kerberos conversation
3. Login to the system is verified over public network using a proxy for Kerberos protocol

What was that?

1. One time password token was programmed to Yubikey and added for the user in FreeIPA
2. SSSD handles login and notices OTP pre-authentication support in Kerberos conversation
3. Login to the system is verified over public network using a proxy for Kerberos protocol
4. Kerberos ticket is obtained, first factor is provided by SSSD to GDM for unlocking GNOME passwords and keys storage (SeaHorse)

What was that?

1. One time password token was programmed to Yubikey and added for the user in FreeIPA
2. SSSD handles login and notices OTP pre-authentication support in Kerberos conversation
3. Login to the system is verified over public network using a proxy for Kerberos protocol
4. Kerberos ticket is obtained, first factor is provided by SSSD to GDM for unlocking GNOME passwords and keys storage (SeaHorse)
5. **Credentials were entered only once**

If Kerberos credentials are available, what can we do with them?

- Authenticate with GSSAPI against almost anything

If Kerberos credentials are available, what can we do with them?

- Authenticate with GSSAPI against almost anything
- Obtain SAML assertion for other web services (and more)

If Kerberos credentials are available, what can we do with them?

- Authenticate with GSSAPI against almost anything
- Obtain SAML assertion for other web services (and more)
- Use to access networking file systems

If Kerberos credentials are available, what can we do with them?

- Authenticate with GSSAPI against almost anything
- Obtain SAML assertion for other web services (and more)
- Use to access networking file systems
- Display properties of the available tickets

If Kerberos credentials are available, what can we do with them?

- Authenticate with GSSAPI against almost anything
- Obtain SAML assertion for other web services (and more)
- Use to access networking file systems
- Display properties of the available tickets
- Renew the ticket granting ticket (TGT)

If Kerberos credentials are available, what can we do with them?

- Authenticate with GSSAPI against almost anything
- Obtain SAML assertion for other web services (and more)
- Use to access networking file systems
- Display properties of the available tickets
- Renew the ticket granting ticket (TGT)
- Choose which Kerberos principal is in use

Authenticate with GSSAPI

Epiphany, the GNOME Web Browser, in GNOME 18:

- GSSAPI support is no more, depends on libsoup support

Authenticate with GSSAPI

Epiphany, the GNOME Web Browser, in GNOME 18:

- GSSAPI support is no more, depends on libsoup support
- libsoup has been dragging since 2009, bug #587145

Authenticate with GSSAPI

Epiphany, the GNOME Web Browser, in GNOME 18:

- GSSAPI support is no more, depends on libsoup support
- libsoup has been dragging since 2009, bug #587145
- WebkitGtk is unusable for SAML/OAuth2 interactions involving Kerberos

Authenticate with GSSAPI

Epiphany, the GNOME Web Browser, in GNOME 18:

- GSSAPI support is no more, depends on libsoup support
- libsoup has been dragging since 2009, bug #587145
- WebkitGtk is unusable for SAML/OAuth2 interactions involving Kerberos
- One cannot use Google apps with GSSAPI in Gnome Online Accounts

Authenticate with GSSAPI

Epiphany, the GNOME Web Browser, in GNOME 18:

- GSSAPI support is no more, depends on libsoup support
- libsoup has been dragging since 2009, bug #587145
- WebkitGtk is unusable for SAML/OAuth2 interactions involving Kerberos
- One cannot use Google apps with GSSAPI in Gnome Online Accounts
- No single sign-on with GSSAPI from GNOME applications using WebkitGtk to authenticate

Can we do better than this?

What was that?

- Tomáš Popela (Red Hat) and David Woodhouse (Intel) worked to fix `libsoup` and `WebkitGtk`
- This laptop is running an experimental build of them
- We logged into my FreeIPA server's Web UI
- Hopefully, the code will be in the next GNOME release

What does GSSAPI support open for use in GNOME Online Accounts?

- Single sign-on is the primary feature

What does GSSAPI support open for use in GNOME Online Accounts?

- Single sign-on is the primary feature
- Automated credentials renewal

What does GSSAPI support open for use in GNOME Online Accounts?

- Single sign-on is the primary feature
- Automated credentials renewal
- Automated token/assertion renewal for SAML/OpenID

What does GSSAPI support open for use in GNOME Online Accounts?

- Single sign-on is the primary feature
- Automated credentials renewal
- Automated token/assertion renewal for SAML/OpenID
- No need to store passwords locally (secure kiosks?)

Visualize

GNOME Online Accounts could show Kerberos ticket properties

- Ticket time validity, flags (forward, renewal)
- Authentication indicators
- Existing service tickets in the credentials cache and allow to remove them selectively
- Allow automatic ticket renewal if KDC permits it

Visualize

And choose between different Kerberos principals

- MIT Kerberos supports kernel keyring (1.12+) and directory-based (1.11+) storage of credentials
- Multiple Kerberos principals can be stored and used at the same time
- Only a single principal can be defined as “primary” for each Kerberos realm in the collection of credentials

Kerberos ticket renewal

- SSSD supports automatic Kerberos ticket renewal for single factor cases
  - Renewing 2FA tickets requires UI interaction triggered by expiry time
  - Automatic ticket renewal requires permission from KDC, visible as a ticket flag
- GNOME Online Accounts could integrate with SSSD in prompting for credentials (multiple factors) in 2FA case needed information could be provided via SSSD InfoPipe/AuthPipe

Better Kerberos in browsers

- Firefox Kerberos setup isn't nice
  - needs about:config manipulation
  - DNS domains associated with Kerberos realm could be discovered via DNS SRV records, prompted for confirmation once
- FreeIPA used to provide an extension to automate Firefox setup
  - Extension was generated locally for for each FreeIPA deployment to provide configuration details
  - not anymore: Firefox removed ability to provide non-publicly available extensions since version 43

Better Kerberos in browsers

- Chromium/Chrome
  - Have bugs for processing of `WWW-Authenticate: Negotiate` when Kerberos credentials are not available
  - On Linux only allows to configure Kerberos use through command line, poor user experience
- A fixed libsoup/WebkitGtk allows to always use GSSAPI if server advertises `WWW-Authenticate: Negotiate` over HTTPS
  - no need to configure anything in Epiphany
  - could be further confined with a user confirmation similar to how passwords are managed on first use

Better Kerberos in browsers

- GSSAPI flow is synchronous, needs better UI interaction to avoid hogging down other tabs
  - still major issue for many browsers

Any *practical* use of it?

What was that?

Ipsilon is an Identity provider that supports GSSAPI, SAML, OpenID, and other methods of authentication

- I set up Ipsilon to authenticate against my FreeIPA server

What was that?

Ipsilon is an Identity provider that supports GSSAPI, SAML, OpenID, and other methods of authentication

- I set up Ipsilon to authenticate against my FreeIPA server
- I set up Owncloud instance and created a simple application to do login via Ipsilon SAML

What was that?

Ipsilon is an Identity provider that supports GSSAPI, SAML, OpenID, and other methods of authentication

- I set up Ipsilon to authenticate against my FreeIPA server
- I set up Owncloud instance and created a simple application to do login via Ipsilon SAML
- Successfully logged-in users get created in Owncloud if they belong to a certain group in FreeIPA

What was that?

Ipsilon is an Identity provider that supports GSSAPI, SAML, OpenID, and other methods of authentication

- I set up Ipsilon to authenticate against my FreeIPA server
- I set up Owncloud instance and created a simple application to do login via Ipsilon SAML
- Successfully logged-in users get created in Owncloud if they belong to a certain group in FreeIPA
- No need to enter password if Kerberos credentials are available

What was that?

Ipsilon is an Identity provider that supports GSSAPI, SAML, OpenID, and other methods of authentication

- I set up Ipsilon to authenticate against my FreeIPA server
- I set up Owncloud instance and created a simple application to do login via Ipsilon SAML
- Successfully logged-in users get created in Owncloud if they belong to a certain group in FreeIPA
- No need to enter password if Kerberos credentials are available
- **Credentials were entered only once**

Oops, I “invented” Owncloud Enteprise Edition?

Better support for SAML in GNOME Online Accounts

GNOME Online Accounts doesn't support SAML for arbitrary provider

- One cannot setup own Owncloud account in GNOME without entering passwords
- Have to use separate Owncloud end-point for non-SAML logon

Certificates

FreeIPA 4.2 supports issuing x.509 certificates to users

FreeIPA 4.2 adds per-user vault to store keys and credentials wrapped into an encrypted blob

- authentication to password vaults is GSSAPI-based
- multiple clients can use unique public/private key pairs to derive their access to user's vault
- SSSD 1.13 allows to authenticate with certificates
- Certificates can come from any OpenSC and coolkey compatible devices

How *enterprisey* our home could become?

What benefits do we get by becoming *enterprisey* with FreeIPA and GNOME?

1. Control your own infrastructure

What benefits do we get by becoming *enterprisey* with FreeIPA and GNOME?

1. Control your own infrastructure
2. Improve user experience by reducing number of password/logon interactions

What benefits do we get by becoming *enterprisey* with FreeIPA and GNOME?

1. Control your own infrastructure
2. Improve user experience by reducing number of password/logon interactions
3. Profit?

Questions?