OSSDEVCONF-2017 // Калуга

Поддержка смарт-карт во FreeIPA

Александр Боковой

Sr. Principal Software Engineer, Red Hat / FreeIPA и Samba Team

FreeIPA

FreeIPA

- Централизованное хранение пользовательской информации
- Централизованные правила доступа к приложениям
- Централизованная аутентификация и авторизация

Kerberos

- FreeIPA предоставляет реалм Kerberos
  - Каждый пользователь и служба в домене FreeIPA пристутствуют в Kerberos
  - Аутентификация пользователей возможна с использованием одного (пароль) или нескольких факторов
- Kerberos поддерживает модель идентификаторов аутентификации
  - идентификаторы добавляются средствами модулей пре-аутентификации
  - модуль otp поддерживает RFC 6560
  - токен передается в открытом виде внутри шифрованного канала FAST
  - верификация токена осуществляется запросом на RADIUS сервер
- FreeIPA реализует встроенный механизм проверки токенов OTP
  - для встроенных токенов поддерживаются RFC 4226 (HOTP) и RFC 6238 (TOTP)

Kerberos FAST

- Канал FAST в Kerberos использует любой имеющийся билет
  - `kinit user`
  - `kinit -T some_ccache new_user`
- SSSD автоматически использует ключ машины для первого шага
  - обычный пользователь не имеет доступа к `/etc/krb5.keytab` и ключу `host/host.name`
- Для обычного пользователя нужен какой-то механизм получения FAST

Kerberos PKINIT

- PKINIT – использование сертификатов для пре-аутентификации
  - RFC 4556
  - Успешная пре-аутентификация с сертификатом позволяет KDC выписать билет пользователю
- Anonymous PKINIT
  - использование “анонимного пользователя” для пре-аутентификации
  - опирается на цепь доверия сертификатов KDC
  - полученный билет можно использовать *только* для построения канала FAST

Kerberos PKINIT

- FreeIPA 4.5 вводит поддержку PKINIT
- Серверы FreeIPA автоматически выпускают сертификаты для KDC
  - С CA можно использовать для полноценного PKINIT
  - В CA-less автоматически поддерживается только Anonymous PKINIT
  - ... и только на самих серверах FreeIPA (для логина в Web UI по паролю)

PKINIT на стороне клиента

- SSSD умеет обрабатывать смарт-карты
- Информация о соответствии сертификата пользователю берется из FreeIPA
- SSSD автоматически получает билет Kerberos при использовании смарт-карты
- Почему не pam_pkcs11?
  - не масштабируется на большие внедрения
  - слишком много кода внутрь приложения (стандартная проблема PAM)
  - нужно дорабатывать механизм соответствия сертификатов пользователям

Соответствие сертификатов

- Правила соответствия сертификатов пользователям
  - традиционно: CN, uid, хэш сертификата, весь сертификат, наличие OID в расширениях
- Огромный разброс правил между реальными заказчиками
  - “для интерактивного логина используем сертификат 3 со смарт-карты, для ssh – сертифифкат 2, для подписи - 1”
  - “сравнивать CN в LDAP порядке”, “сравнивать CN в x.509 порядке”
- В результате, мы написали гибкий механизм описания правил соответствия сертификатов:
  - `https://docs.pagure.org/SSSD.sssd/design_pages/matching_and_mapping_certificates.html`

FreeIPA

- Правила соответствия задаются централизовано
- Могут применяться как для пользователей FreeIPA, так и для пользователей из AD
- Могут быть описаны в ID View для конкретной группы машин

SSSD

- SSSD полностью заменяет pam_pkcs11 с точки зрения функционала
- Обработка смарт-карты происходит на бэкенде
  - в приложении, вызывающем PAM стек, нет никакого дополнительного кода
- Поддерживаются и ролевые учетные записи
  - один сертификат для входа во множество учетных записей
  - много сертификатов для входа в одну учетную запись
- Интеграция с GNOME

Спасибо за внимание!