FreeIPA

Poziom: średnio **Dotyczy: FreeIPA 4.x na RHEL/Rocky 9** **Czas: ~30 min**

FreeIPA łączy w jednym miejscu cztery usługi, które w środowisku Windows zwykle stoją osobno — zaczynamy od tego, co dokładnie się w nim kryje, zanim przejdziemy do zarządzania użytkownikami.

Skróty użyte na tej stronie (13)

Skrót	Co znaczy
AD	Active Directory: katalog Microsoftu z kontami użytkowników, komputerów i grup w firmie
AD CS	Active Directory Certificate Services: rola Windows Server pełniąca funkcję firmowego urzędu certyfikacji
CA	Certificate Authority: urząd certyfikacji, który podpisuje certyfikaty
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
FQDN	Fully Qualified Domain Name: pełna nazwa hosta razem z domeną, np. docker01.ad.zielona.example
HBAC	Host-Based Access Control: reguły FreeIPA mówiące, kto może się zalogować na który serwer
KDC	Key Distribution Center: serwer Kerberosa wydający bilety logowania
LDAP	Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)
NSS	Network Security Services: biblioteka kryptograficzna Mozilli, w której bazie FreeIPA i Dogtag trzymają certyfikaty i klucze
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SSSD	System Security Services Daemon: usługa na kliencie linuksowym tłumacząca logowanie, sudo i sprawdzanie grup na zapytania do katalogu, czy to FreeIPA, czy Active Directory
TGT	Ticket Granting Ticket: główny bilet Kerberos wydawany przez kontroler domeny po zalogowaniu, na jego podstawie komputer dostaje bilety do konkretnych usług
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS

Fiszka

□□ ZMIENIA SYSTEM

Instalacja i dołączenie hosta do domeny: ipa-client-install

```
sudo ipa-client-install --domain=NazwaDomeny --server=ipa01.NazwaDomeny --mkhomedir
```

(gdzie `NazwaDomeny` to FQDN domeny FreeIPA, a flaga `--mkhomedir` każe systemowi tworzyć katalog domowy użytkownika przy pierwszym logowaniu, jeśli jeszcze nie istnieje)

Skrypt konfiguruje na hoście SSSD, Kerberos, NSS oraz wpisuje hosta do katalogu FreeIPA jako zaufanego klienta — po zakończeniu host rozstrzyga użytkowników domenowych identycznie jak lokalnych, przez tę samą warstwę `getent` / `id`.

□□ **FQDN** = Fully Qualified Domain Name: pełna nazwa hosta razem z domeną, np. docker01.ad.zielona.example · **SSSD** = System Security Services Daemon: usługa na kliencie linuksowym tłumacząca logowanie, sudo i sprawdzanie grup na zapytania do katalogu, czy to FreeIPA, czy Active Directory · **NSS** = Network Security Services: biblioteka kryptograficzna Mozilli, w której bazie FreeIPA i Dogtag trzymają certyfikaty i klucze · **ipa-client-install** = dołącz ten linux do istniejącej już domeny FreeIPA jako klienta

□□ TYLKO ODCZYT

Weryfikacja dołączenia — rozpoznanie użytkownika domenowego: id

```
id nazwa_usera@NazwaDomeny
```

Przykład:

```
uid=1487600123(nazwa_usera@ipa.zielona.example)
gid=1487600123(nazwa_usera@ipa.zielona.example)
groups=1487600123(nazwa_usera@ipa.zielona.example)
```

□□ **id** = pokaż użytkownika, jego numer UID oraz grupy, do których należy

□□ TYLKO ODCZYT

Pobranie biletu Kerberosa: kinit

```
kinit nazwa_usera
```

□□ **kinit** = Kerberos init: pobierz bilet logowania Kerberos

TYLKO ODCZYT

Sprawdzenie aktywnych biletów w bieżącej sesji: klist

```
klist
```

Przykład:

```
Ticket cache: KCM:0:1487600123
Default principal: nazwa_usera@IPA.ZIELONA.EXAMPLE

Valid starting     Expires            Service principal
09/22/26 08:14:02  09/22/26 18:14:02  krbtgt/IPA.ZIELONA.EXAMPLE@IPA.ZIELONA.EXAMPLE
```

Uważaj: nazwa domeny (realm) Kerberosa jest zawsze zapisywana wielkimi literami (`IPA.ZIELONA.EXAMPLE`), niezależnie od tego, jak zapisujemy FQDN domeny w reszcie konfiguracji — to konwencja Kerberosa, nie literówka w przykładzie.

IPA = Identity, Policy, Audit, czyli tożsamość, polityka i audyt: rozwinięcie nazwy FreeIPA, opisujące trzy warstwy, które projekt łączy w jednej instalacji · **FQDN** = Fully Qualified Domain Name: pełna nazwa hosta razem z domeną, np. docker01.ad.zielona.example · **klist** = Kerberos list: pokazuje bilety Kerberos zalogowanego użytkownika i czas ich ważności

TYLKO ODCZYT

Dodanie nowego użytkownika: ipa user-add

```
ipa user-add nazwa_usera --first=Imie --last=Nazwisko --email=nazwa_usera@NazwaDomeny
```

ipa = narzędzie wiersza poleceń do zarządzania FreeIPA (użytkownicy, grupy, reguły HBAC i sudo)

TYLKO ODCZYT

Dodanie nowej grupy: ipa group-add

```
ipa group-add NazwaGrupy --desc="Opis przeznaczenia grupy"
```

ipa = narzędzie wiersza poleceń do zarządzania FreeIPA (użytkownicy, grupy, reguły HBAC i sudo)

□□ ZMIENIA SYSTEM

Dodanie użytkownika do grupy: ipa group-add-member

```
ipa group-add-member NazwaGrupy --users=nazwa_usera
```

□□ **ipa** = narzędzie wiersza poleceń do zarządzania FreeIPA (użytkownicy, grupy, reguły HBAC i sudo)

□□ TYLKO ODCZYT

Odczyt członków grupy z poziomu systemu: getent

```
getent group NazwaGrupy
```

Przykład:

```
NazwaGrupy:*:1487600501:nazwa_usera,inny_user
```

□□ **getent** = get entries: odpytaj bazę systemową (użytkowników, grup, hostów) niezależnie od tego, czy to plik lokalny czy AD

□□ ZMIENIA SYSTEM

Synchronizacja grupy lokalnej z grupą FreeIPA: getent + usermod w pętli

```
getent group NazwaGrupyIPA | cut -d: -f4 | tr ',' '\n' | while read -r username; do
  sudo usermod -aG docker "$username"
done
```

Zdarza się, że aplikacja lokalna (np. silnik kontenerów) wymaga przynależności do lokalnej grupy systemowej, a listę uprawnionych osób chcemy utrzymywać centralnie w FreeIPA, a nie osobno na każdym hoście. Rozwiązujemy to pętlą odczytującą członków grupy domenowej i dopisującą ich do grupy lokalnej. Skrypt czyta listę członków grupy `NazwaGrupyIPA` z FreeIPA (czwarte pole wyniku `getent group`, rozdzielone przecinkami), rozbija je na osobne linie i dla każdej nazwy dopisuje użytkownika do lokalnej grupy `docker`. Uruchamiany cyklicznie (np. z crona) utrzymuje zgodność członkostwa lokalnego z grupą domenową bez ręcznej synchronizacji przy każdej zmianie personelu.

Uważaj: pętla tylko dodaje — nie usuwa z grupy lokalnej użytkowników, którzy zostali już z grupy domenowej wypisani. Pełna synchronizacja (również usuwanie) wymaga dodatkowego

porównania obecnego stanu grupy lokalnej z aktualną listą z FreeIPA, a nie samego dopisywania.

getent = get entries: odpytaj bazę systemową (użytkowników, grup, hostów) niezależnie od tego, czy to plik lokalny czy AD · **usermod** = zmień ustawienia istniejącego konta systemowego, na przykład przynależność do grupy

ZMIENIA SYSTEM

Utworzenie reguły HBAC: ipa hbacrule-add

```
ipa hbacrule-add NazwaReguly
ipa hbacrule-add-user NazwaReguly --groups=NazwaGrupy
ipa hbacrule-add-host NazwaReguly --hostgroups=NazwaGrupyHostow
```

HBAC = Host-Based Access Control: reguły FreeIPA mówiące, kto może się zalogować na który serwer · **ipa** = narzędzie wiersza poleceń do zarządzania FreeIPA (użytkownicy, grupy, reguły HBAC i sudo)

ZMIENIA SYSTEM

Utworzenie reguły sudo w FreeIPA: ipa sudorule-add

```
ipa sudorule-add NazwaReguly
ipa sudorule-add-user NazwaReguly --groups=NazwaGrupy
ipa sudorule-add-host NazwaReguly --hostgroups=NazwaGrupyHostow
ipa sudorule-add-allow-command NazwaReguly --sudocmds='/usr/bin/systemctl'
```

ipa = narzędzie wiersza poleceń do zarządzania FreeIPA (użytkownicy, grupy, reguły HBAC i sudo)

TYLKO ODCZYT

Żądanie certyfikatu dla usługi: ipa cert-request

```
ipa cert-request SciezkaDoPliku.csr --principal=HTTP/app01.NazwaDomeny
```

ipa = narzędzie wiersza poleceń do zarządzania FreeIPA (użytkownicy, grupy, reguły HBAC i sudo) · **HTTP** = Hypertext Transfer Protocol: protokół stron WWW

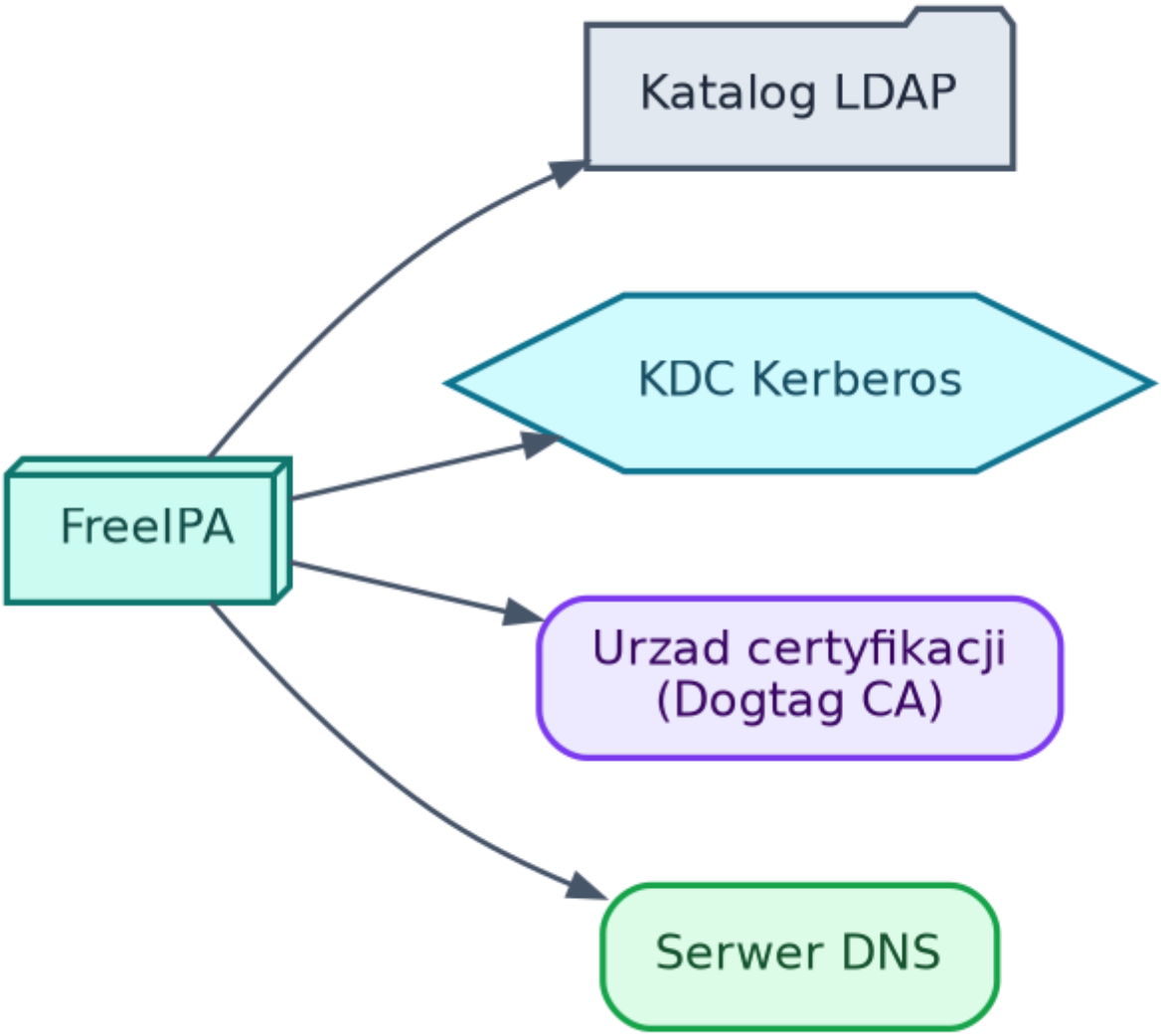
Z życia wzięte

Z ŻYCIA WZIĘTE

O 8:30 nowy pracownik czeka na dostęp do serwerów, a jego konto dopiero co dodane w FreeIPA nie może się jeszcze zalogować na `app01`. Tomek sprawdza `ipa hbacrule-find` i widzi, że reguła HBAC dla grupy „Serwery aplikacyjne" nie obejmuje jeszcze tej osoby, bo dodanie do grupy w katalogu nie jest tym samym co dodanie do reguły dostępu. Dopisuje go do właściwej grupy HBAC i logowanie zaczyna działać w ciągu kilku sekund, bez restartu czegokolwiek.

Czym jest FreeIPA

Pojęcie: FreeIPA. Otwartoźródłowy system zarządzania tożsamością, łączący w jednej instalacji katalog LDAP, ośrodek dystrybucji kluczy Kerberos (KDC), wewnętrzny urząd certyfikacji (CA) oraz serwer DNS — cztery usługi, które administrator Windows zna jako oddzielone role Active Directory (katalog, Kerberos, opcjonalnie AD CS, opcjonalnie DNS zintegrowany z AD), tutaj spięte we wspólną instalację i wspólny interfejs zarządzania.



FreeIPA łączy w jednej instalacji katalog LDAP, KDC Kerberosa, urząd certyfikacji i serwer DNS (kliknięcie otwiera pełny rozmiar)

FreeIPA a Active Directory. Oba systemy rozwiązują ten sam problem — scentralizowaną tożsamość i uwierzytelnianie Kerberos w sieci — ale FreeIPA jest natywnie zbudowana pod środowiska linuksowe (dołączanie hostów Linux, reguły `sudo` i `HBAC` opisane niżej), podczas gdy AD jest natywnie zbudowana pod Windows. Współistnienie obu w jednej organizacji realizuje się przez zaufanie międzydomenowe (trust), o którym więcej w sekcji na końcu tej strony — nie przez próbę zastąpienia jednego drugim tam, gdzie już działa.

Serwer i replika

Pojęcie: serwer FreeIPA. Host, na którym zainstalowana jest pełna usługa — katalog LDAP, KDC, CA, DNS — obsługujący żądania klientów i innych replik.

Pojęcie: replika. Kolejny serwer FreeIPA dołączony do tej samej domeny, synchronizujący pełną kopię katalogu z serwerem, z którego został utworzony — architektura wieloserwerowa (multi-master), więc każda replika przyjmuje zarówno odczyty, jak i zapisy, a zmiana wprowadzona na jednej propaguje się do pozostałych. Replikę stawiamy z dwóch powodów: odporności na awarię pojedynczego serwera oraz obniżenia opóźnień uwierzytelniania dla klientów w oddalonej lokalizacji sieciowej.

kinit i bilety Kerberosa

Pojęcie: kinit. Polecenie pobierające bilet Kerberosa (TGT — Ticket Granting Ticket) dla wskazanego użytkownika, na podstawie hasła albo klucza — od tego momentu do wygaśnięcia biletu użytkownik uwierzytelnia się w usługach zintegrowanych z Kerberosem bez ponownego podawania hasła.

Zarządzanie użytkownikami i grupami

Odczyt członków grupy z poziomu systemu

Pojęcie: getent. Polecenie odpytujące skonfigurowane źródła tożsamości systemu (lokalny `/etc/group`, oraz — na kliencie dołączonym do FreeIPA — katalog przez SSSD) w jednym, spójnym formacie, niezależnie od tego, skąd faktycznie pochodzi dana grupa czy użytkownik.

Reguły HBAC i sudo

Pojęcie: HBAC (Host-Based Access Control). Mechanizm FreeIPA decydujący, który użytkownik (albo grupa użytkowników) ma prawo się zalogować na który host (albo grupę hostów) — domyślna reguła `allow_all` zezwala każdemu na logowanie się wszędzie, co w praktyce zwykle zawężamy własnymi regułami zanim wdrożymy FreeIPA produkcyjnie.

Pojęcie: reguła sudo w FreeIPA. Reguła centralnie zarządzana, analogiczna do lokalnego wpisu w `/etc/sudoers`, ale definiowana raz w katalogu i propagowana na wszystkie hosty, których dotyczy — eliminuje ręczne utrzymywanie osobnego pliku `sudoers` na każdej maszynie z osobna.

Certyfikaty z wbudowanego CA

Pojęcie: wbudowany urząd certyfikacji FreeIPA (Dogtag CA). Urząd certyfikacji instalowany razem z serwerem FreeIPA, wystawiający certyfikaty dla hostów i usług w domenie bez potrzeby zewnętrznego CA — użyteczny do certyfikatów wewnętrznych (np. TLS między usługami w tej samej domenie), rzadziej stosowany do certyfikatów wystawianych publicznie dostępnym usługom, gdzie potrzebny jest urząd zaufany poza organizacją.

Zaufanie do domeny AD w skrócie

Pojęcie: zaufanie międzydomenowe (trust). Relacja skonfigurowana między domeną FreeIPA a domeną Active Directory, pozwalająca użytkownikom AD uwierzytelniać się w usługach zależnych od FreeIPA bez zakładania dla nich osobnych kont w katalogu FreeIPA. Konfiguracja wymaga roli AD Trust na serwerze FreeIPA (dodatkowy pakiet `ipa-server-trust-ad`) oraz koordynacji z administratorem domeny AD po drugiej stronie — to jest funkcja jednorazowego, świadomego wdrożenia integracyjnego, a nie coś ustawianego przy okazji zwykłej instalacji klienta.

Hosty dołączane wyłącznie do domeny AD (bez FreeIPA po ich stronie) opisujemy osobno na stronie Dołączenie do domeny AD przez realmd — mechanizm `realmd`/SSSD po stronie klienta jest analogiczny niezależnie od tego, czy katalogiem zarządzającym jest FreeIPA czy Active Directory.

Częste błędy

Pozostawienie domyślnej reguły HBAC allow_all na produkcji. Bez zawężenia reguł HBAC każdy użytkownik domenowy loguje się na każdy host dołączony do domeny — wygodne przy pierwszym wdrożeniu, ale niezgodne z zasadą najmniejszych uprawnień w środowisku produkcyjnym.

Traktowanie pętli synchronizującej grupę lokalną jako pełnej synchronizacji. Skrypt dopisujący członków grupy domenowej do grupy lokalnej bez logiki usuwania utrzymuje w grupie lokalnej też osoby dawno wypisane z grupy domenowej — wymaga to jawnego mechanizmu porównania stanu, nie samego dopisywania.

Mylenie wielkości liter w nazwie realm Kerberosa. Realm zapisany małymi literami w ręcznie edytowanym pliku `krb5.conf` jest częstą przyczyną niewyjaśnionych błędów uwierzytelniania — Kerberos rozróżnia wielkość liter w tym polu.

□□ Ćwiczenia

> Zadania sprawdzające.

Ćwiczenie 1 — dobór mechanizmu

Chcemy, żeby grupa administratorów zdefiniowana raz w FreeIPA miała prawo `sudo` bez hasła do polecenia restartu usług na wszystkich serwerach aplikacyjnych, bez edytowania pliku `sudoers` na każdym z nich osobno. Wskazujemy właściwy mechanizm.

Odpowiedź

Reguła sudo zarządzana centralnie w FreeIPA (`ipa sudorule-add` z przypisaną grupą użytkowników, grupą hostów i dozwoloną komendą) — to dokładnie mechanizm zaprojektowany po to, żeby uniknąć ręcznego utrzymywania osobnego pliku `sudoers` na każdym serwerze z osobna, przy zmianie propagowanej centralnie z jednego miejsca.

Powiązane: Dołączenie do domeny AD przez realmd · Gitea · NetBox

Źródła: dokumentacja FreeIPA, rozdział „Users, Groups, Hosts" i „Access Control" · `man ipa-client-install` · `man sudorule`

Revision #36
Created 2026-09-25 07:40:43 UTC by Tomasz Konieczny
Updated 2026-10-26 14:14:38 UTC by Tomasz Konieczny