



Jak skonfigurować scentralizowane uwierzytelnianie (ssh i sudo) za pomocą oprogramowania FreeIPA w systemie FreeBSD.



Krzysztof Toczyski

kt@a-ng.eu



Czym jest FreeIPA cz. 1

Jest to darmowy system zarządzania tożsamością (Identity management system), który ma nam zapewnić łatwość obsługi w następujących czynnościach:

- I – Identyfikacja (Identity)
- P – Polityka (Policy)
- A – Audyt (Audit)



Czym jest FreeIPA cz. 2

FreeIPA jest projektem otwartym rozwijanym przez firmę Red Hat, który zawiera następujące komponenty:

- 389 Directory Server (LDAP)
- MIT Kerberos
- NTP
- DNS (Bind)
- Dogtag certificate system
- SSSD (System Security Services Daemon)
- i inne (m. in. Apache)



Co nam zapewni FreeIPA

- rejestruje użytkowników centralnie
- potwierdza tożsamość użytkowników
- przechowuje i udostępnia dane identyfikacyjne użytkowników systemom autoryzującym uprawnionym do ich otrzymania
- umożliwia zablokowanie konta użytkownika na jego żądanie
- zapewnia integralność, autentyczność i poufność danych identyfikacyjnych i uwierzytelniających użytkownika
- zapewnia synchronizację czasu systemowego z czasem UTC



Konfiguracja FreeBSD jako klienta FreeIPA do SSH i Sudo

To nie będzie łatwa droga ;)

Co potrzebujemy:

1. Działający serwer FreeIPA
2. Działający serwer DNS (bardzo ważne)
3. Zbudować pakiety
4. Zainstalować pakiety
5. Konfiguracja systemu
6. Konfiguracja pakietów
7. Konfiguracja FreeIPA



1. Serwer FreeIPA

Instalujemy go na zalecanej dystrybucji Linuxa

- Polecam Fedore (najnowsze wersje FreeIPA)
- Nie polecam Centosa ;) (stare wersje FreeIPA)

Krótki opis instalacji:

yum install ipa-server

Dodajemy nazwę serwera do **/etc/hosts** w postaci krótkiej jak i FQDN

Instalujemy z serwerem DNS albo bez.

W wersji bez musimy dodać rekord A i PTR w naszego serwera DNS

ipa-server-install --enablemkhomedir

Musimy mieć jakąś domenę. Nie polecam używać z końcówką **.local** (używana w mDNS i bonjour w Linuxie)

Resztę załatwi za nas kreator.



2. Serwer DNS

- Musimy posiadać w sieci działający serwer DNS z serwera FreeIPA albo własny
- Każdy host musi bezwzględnie posiadać rekord A i PTR, inaczej nie będzie nam to działać
- Przy własnym serwerze musimy dodać rekordy z tego polecenia

```
ipa dns-update-system-records --dry-run
```




3. Budowa pakietów cz. 1

Mamy dwie opcje:

1. Pakiety budujemy na systemie na którym planujemy uruchomić połączenie do serwera FreeIPA
2. Pakiety budujemy w naszym prywatnym repo (Poudriere), a później instalujemy je przez pkg install. Opcja zalecana 😊



3. Budowa pakietów cz. 2

- Do `/etc/make.conf` dodajemy:

```
WANT_OPENLDAP_SASL=YES
```

```
WITH_GSSAPI=YES
```

Budujemy takie pakiety z następującymi opcjami:

- `security/sss`: Enable SMB (Builds and installs the IPA provider) << opcja konieczna, inaczej nie zbuduje nam modułu IPA dla sssd. Zostanie też zainstalowana Samba, której nie użyjemy.
- `security/sudo`: Enable SSSD backend
- `security/cyrus-sasl2-gssapi`
- `security/pam_mkhomedir`



4. Instalacja pakietów

```
pkg install -y -r NASZE.REPO.PL cyrus-sasl-gssapi  
sssd sudo pam_mkhome
```



5. Konfiguracja systemu

- **Tworzymy katalogi**

```
mkdir -p /usr/local/etc/ipa /var/log/sss /var/run/sss/private /var/db/sss
```

- **Ściągamy certyfikat CA z serwera IPA**

```
wget -O /usr/local/etc/ipa/ca.crt http://FreeIPA/ipa/config/ca.crt
```

- **Generowanie pliku KEYTAB dla hosta**

Logujemy się na serwer FreeIPA
ssh [root@FreeIPA.nasza.domena](ssh://root@FreeIPA.nasza.domena)

- **Logujemy się do Kerberos:**

```
kinit admin  
Password for admin@NASZA.DOMENA:
```

- **Dodajemy hosta:**

```
ipa host-add FreeBSD.nasza.domena
```

Powinniśmy otrzymać:

```
-----  
Added host "FreeBSD.nasza.domena"  
-----  
Host name: FreeBSD.nasza.domena  
Principal name:  
    host/FreeBSD.nasza.domena@NASZA.DOMENA  
Principal alias:  
    host/FreeBSD.nasza.domena@NASZA.DOMENA  
Password: False  
Keytab: False  
Managed by: FreeBSD.nasza.domena
```

- **Generujemy plik keytab dla hosta:**

```
ipa-getkeytab -s FreeIPA.nasza.domena -p  
    host/FreeBSD.nasza.domena@NASZA.DOMENA -k  
    /root/FreeBSD.keytab
```

```
Keytab successfully retrieved and stored in:  
    /root/FreeBSD.keytab  
exit
```

- **Ściągamy keytab na host lokalny:**

```
scp root@FreeIPA.nasza.domena:/root/FreeBSD.keytab  
    /usr/local/etc/ipa/krb5.keytab
```



6. Konfiguracja pakietów

Co potrzebujemy skonfigurować?

1. OpenLDAP
2. Kerberos
3. SSSD
4. nsswitch
5. pam.d
6. NTPdate
7. SSH



6.1 Konfiguracja OpenLDAP

```
vi /usr/local/etc/openldap/ldap.conf
```

```
BASE    dc=nasza,dc=domena  
URI     ldap://FreeIPA.nasza.domena
```

```
#SIZELIMIT    12  
#TIMELIMIT    15  
#DEREF        never
```

```
SASL_MECH GSSAPI  
SASL_REALM NASZA.DOMENA  
ssl start_tls  
TLS_CACERT /usr/local/etc/ipa/ca.crt
```



6.2 Konfiguracja Kerberos

[libdefaults]

```
default_realm = NASZA.DOMENA
default_keytab_name = FILE:/usr/local/etc/ipa/krb5.keytab
default_tkt_enctypes = aes256-cts des-cbc-crc aes128-cts arcfour-hmac
default_tgs_enctypes = aes256-cts des-cbc-crc aes128-cts arcfour-hmac
dns_lookup_realm = false
dns_lookup_kdc = false
rdns = false
ticket_lifetime = 24h
forwardable = yes
```

[realms]

```
NASZA.DOMENA = {
    kdc = FreeIPA.nasza.domena:88
    master_kdc = FreeIPA.nasza.domena:88
    admin_server = FreeIPA.nasza.domena:749
    default_domain = nasza.domena
    pkinit_anchors = FILE:/usr/local/etc/ipa/ca.crt
}
```

[domain_realm]

```
.nasza.domena = NASZA.DOMENA
nasza.domena = NASZA.DOMENA
```

[logging]

```
kdc = FILE:/var/log/krb5/krb5kdc.log
admin_server = FILE:/var/log/krb5/kadmin.log
kadmin_local = FILE:/var/log/krb5/kadmin_local.log
default = FILE:/var/log/krb5/krb5lib.log
```



6.3 Konfiguracja SSSD

vi /usr/local/etc/sss/sss.conf

```
[domain/nasza.domena]
#debug_level = 9
cache_credentials = True
krb5_store_password_if_offline = True
krb5_realm = NASZA.DOMENA
ipa_domain = nasza.domena
id_provider = ipa
auth_provider = ipa
access_provider = ipa
ipa_hostname = FreeBSD.nasza.domena << nazwa
    lokalnego hosta
chpass_provider = ipa
ipa_server = _srv_, FreeIPA.nasza.domena
ldap_tls_cacert = /usr/local/etc/ipa/ca.crt
krb5_keytab = /usr/local/etc/ipa/krb5.keytab
```

```
[sss]
services = nss, pam, ssh, sudo
config_file_version = 2
domains = nasza.domena
```

```
[nss]
filter_users = root,toor
homedir_substring = /home/users/%u
```

```
[pam]
```

```
[sudo]
#debug_level = 0x3ff0
```

```
[ssh]
```




6.3 Konfiguracja SSSD

- Nadanie odpowiednich uprawnień dla pliku konfiguracyjnego
`chmod 600 /usr/local/etc/sss/sss.conf`
- Wpisanie SSSD do `rc.conf`
`echo sss_enable="YES" >> /etc/rc.conf`

FreeBSD jest trochę ułomne jeżeli chodzi o grupy w SUDO:

With this setup, sudo rules regarding host groups won't work on our FreeBSD host, although sudo rules regarding separate hosts will work. FreeIPA keeps host groups in netgroups, and we configured netgroup: files in `nsswitch.conf` file; so we need to create a file from which FreeBSD will read information about host groups. We can create and periodically update such a file by running a special script via cron, which is a tool for running scripts in periodic intervals.

- Tworzymy plik:



6.3 Konfiguracja SSSD

```
vi /root/bin/sudo.sh
```

```
#!/bin/sh
```

```
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin  
export PATH
```

```
progname=$(basename $0)  
tmpf=$(mktemp)
```

```
old_krb5_ccache=${KRB5_CCACHE}  
KRB5_CCACHE=/tmp/_hostgroups_access.ccache.$$  
export KRB5_CCACHE  
kinit -k -t /usr/local/etc/ipa/krb5.keytab  
host/FreeBSD.nasza.domena@NASZA.DOMENA
```

```
trap "rm -f $tmpf" EXIT
```

```
ldapsearch -Y GSSAPI -LLL -H "ldap://FreeIPA.nasza.domena" \  
-b 'cn=hostgroups,cn=accounts,dc=idm,dc=wan' \  
'(objectClass=ipahostgroup)' cn member \  
|
```

```
while read line; do  
if [ "$line" = "" ]; then  
if [ "$members" != "" ]; then  
echo "$groupname \"\" >>$tmpf  
for host in $members; do  
echo "($host, -, fxcorp) \"\" >>$tmpf  
done  
echo "" >>$tmpf  
fi  
fi
```

```
groupname=""  
members=""  
continue  
fi
```

```
key=${line%*: *}  
value=${line##*: }
```

```
if [ "$key" = "dn" ]; then  
continue  
elif [ "$key" = "cn" ]; then  
groupname=$value  
elif [ "$key" = "member" ]; then  
host=${value%%,cn*}  
host=${host##fqdn=}  
members="$members $host"  
fi  
done
```

```
if [ ! -s "$tmpf" ]; then  
echo "$progname: refusing to install an empty file, bailing" >&1  
exit 1  
fi
```

```
install -m 0644 -o root -g wheel $tmpf /etc/netgroup  
rc=$?
```

```
if [ $rc -ne 0 ]; then  
echo "$progname: error installing /etc/netgroup (rc = $rc)" >&2  
exit 2  
fi
```

```
kdestroy  
KRB5_CCACHE=${old_krb5_ccache}  
export KRB5_CCACHE
```

```
exit 0
```



6.3 Konfiguracja SSSD

```
vi /root/bin/sssd.sh
```

```
#!/bin/sh  
service sssd restart
```

- Nadajemy uprawnienia:

```
chmod 744 sudo.sh  
chmod 744 sssd.sh
```

- Dodajemy do crona:

```
vi /etc/crontab
```

```
*/5 * * * * root /root/bin/sudo.sh >/dev/null 2>&1  
*/5 * * * * root /root/bin/sssd.sh >/dev/null 2>&1
```

- Po uruchomieniu pliku **sudo.sh** powinniśmy zobaczyć coś takiego:

```
SASL/GSSAPI authentication started  
SASL username: host/FreeBSD.nasza.domena@NASZA.DOMENA  
SASL SSF: 56 SASL data security layer installed.
```

- Uruchomienie demona SSSD

```
service sssd start
```



6.4 Konfiguracja nsswitch

```
vi /etc/nsswitch.conf
```

```
group: files sss << przerabiamy
group_compat: nis
hosts: files dns
#netgroup: compat << haszujemy
networks: files
passwd: files sss << przerabiamy
passwd_compat: nis
shells: files
services: compat
services_compat: nis
protocols: files
rpc: files
sudoers: sss files << dodajemy
netgroup: files << dodajemy
```



6.5 Konfiguracja pam.d

```
vi /etc/pam.d/system
```

```
# auth
auth          sufficient      pam_opie.so          no_warn no_fake_prompts
auth          requisite      pam_opieaccess.so   no_warn allow_local
auth          sufficient      pam_krb5.so         no_warn try_first_pass
#auth        sufficient      pam_ssh.so          no_warn try_first_pass
auth          sufficient      /usr/local/lib/pam_sss.so use_first_pass
auth          required       pam_unix.so         no_warn try_first_pass nullok

# account
#account     required       pam_krb5.so
account     required       pam_login_access.so
account     required       pam_unix.so
account     required      /usr/local/lib/pam_sss.so ignore_unknown_user ignore_authinfo_unavail

# session
#session     optional       pam_ssh.so          want_agent
session     required       pam_lastlog.so      no_fail
session     required      /usr/local/lib/pam_mkhomedir.so mode=0700

# password
#password    sufficient      pam_krb5.so         no_warn try_first_pass
password    sufficient      /usr/local/lib/pam_sss.so use_authtok
password    required       pam_unix.so         no_warn try_first_pas
```



6.5 Konfiguracja pam.d

```
vi /etc/pam.d/sshd
```

```
# auth
auth          sufficient      pam_opie.so          no_warn no_fake_prompts
auth          requisite      pam_opieaccess.so   no_warn allow_local
auth          sufficient      pam_krb5.so          no_warn try_first_pass
#auth         sufficient      pam_ssh.so           no_warn try_first_pass
auth          sufficient      /usr/local/lib/pam_sss.so use_first_pass
auth          required        pam_unix.so          no_warn try_first_pass

# account
account       required        pam_nologin.so
#account      required        pam_krb5.so
account       required        pam_login_access.so
account       required        pam_unix.so
account       required      /usr/local/lib/pam_sss.so ignore_unknown_user ignore_authinfo_unavail

# session
#session      optional        pam_ssh.so           want_agent
session       required        pam_permit.so
session       required      /usr/local/lib/pam_mkhomedir.so mode=0700

# password
#password     sufficient      pam_krb5.so          no_warn try_first_pass
password      sufficient      /usr/local/lib/pam_sss.so use_authtok
password      required        pam_unix.so          no_warn try_first_pass
```



6.6 Konfiguracja NTPdate

Synchronizacja czasu dla Kerberos.

```
vi /etc/rc.conf
```

```
ntpdate_enable="YES"
```

```
ntpdate_hosts="FreeIPA.nasza.domena"
```



6.7 Konfiguracja SSH

- SSH Klient

```
vi /etc/ssh/ssh_config
```

```
GSSAPIAuthentication yes
```

- SSH Serwer

```
vi /etc/ssh/sshd_config
```

```
#ChallengeResponseAuthentication yes
```

```
GSSAPIAuthentication yes
```

```
UsePAM yes
```




Uwagi

- FreeBSD 11.X nie wspiera zmiany hasła w LDAP via kerberos za pomocą **passwd**. Jest to problematyczne przy pierwszym logowaniu i przy resecie hasła przez administratora (związane z założeniem twórców FreeIPA). Przy tworzeniu konta i resecie hasła użytkownik i tak musi zmienić je na swoje.

Password expired. Change your password now.

Current password:

Password:

Password:

Permission denied (publickey,gssapi-with-mic,keyboard-interactive).

passwd: Sorry, `passwd' can only change passwords for local or NIS users.

Możemy to ominąć na kilka sposobów:

- Wchodzimy na stronę serwera FreeIPA np. <https://FreeIPA.nasza.domena/ipa/ui> i logujemy się za pomocą swojego użytkownika i hasła. Zostaniemy poproszeni o zmianę hasła.
- Zmieniamy hasło na Linuxie, który podpięty jest do serwera FreeIPA. Linux nie ma z tym problemów.
- Podobno są skrypty na FreeBSD, które to w jakiś sposób poprawiają, ja ich nie znalazłem.
- Możemy sami poprawić **passwd**, żeby zaczął to wspierać :D



7. Konfiguracja FreeIPA

Krótki pokaz konfiguracji, dodawania polityk w FreeIPA (wersja offline).

Od czego zaczynamy:

1. Stworzenie użytkownika
2. Stworzenie HBAC Rules
3. Stworzenie reguł Sudo



Host-Based Access Control (HBAC) Rules

Ustawienia

Refresh Revert Save Actions

General

Nazwa reguły jump_hosts

Opis Users in the "NOC" group can access only to jump_hosts only via SSH.

Who

Kategoria użytkowników, do których zastosowywana jest reguła: Anyone Specified Users and Groups

<input type="checkbox"/> Użytkownicy	Usuń + Dodaj
<input type="checkbox"/> Grupy użytkowników	Usuń + Dodaj
<input type="checkbox"/> noc	

Accessing

Kategoria komputerów, do których zastosowywana jest reguła: Any Host Specified Hosts and Groups

<input type="checkbox"/> Komputery	Usuń + Dodaj
<input type="checkbox"/> Grupy komputerów	Usuń + Dodaj
<input type="checkbox"/> jump_hosts	

Via Service

Kategoria usług, do których zastosowywana jest reguła: Any Service Specified Services and Groups

<input type="checkbox"/> Usługi	Usuń + Dodaj
<input type="checkbox"/> sshd	
<input type="checkbox"/> sudo	



Reguły Sudo

General

Nazwa reguły: test

Sudo order:

Opis:

Options

<input type="checkbox"/> Sudo Option	Usuń Dodaj
--------------------------------------	--

Who

Kategoria użytkowników, do których zastosowywana jest reguła: Anyone Specified Users and Groups

<input type="checkbox"/> Użytkownicy	External	Usuń Dodaj
<input type="checkbox"/> Grupy użytkowników		Usuń Dodaj
<input type="checkbox"/> admins		

Access this host

Kategoria komputerów, do których zastosowywana jest reguła: Any Host Specified Hosts and Groups

<input type="checkbox"/> Komputery	External	Usuń Dodaj
<input type="checkbox"/> Grupy komputerów		Usuń Dodaj
<input type="checkbox"/> jump_hosts		
<input type="checkbox"/> ftp_hosts		
<input type="checkbox"/> dns_hosts		

Run Commands



Reguły Sudo

Options

<input type="checkbox"/> Sudo Option	Usuń Dodaj
--------------------------------------	--

Who

Kategoria użytkowników, do których zastosowywana jest reguła: Anyone Specified Users and Groups

<input type="checkbox"/> Użytkownicy	External	Usuń Dodaj
<input type="checkbox"/> Grupy użytkowników		Usuń Dodaj
<input type="checkbox"/> admins		

Access this host

Kategoria komputerów, do których zastosowywana jest reguła: Any Host Specified Hosts and Groups

<input type="checkbox"/> Komputery	External	Usuń Dodaj
<input type="checkbox"/> Grupy komputerów		Usuń Dodaj
<input type="checkbox"/> jump_hosts		
<input type="checkbox"/> ftp_hosts		
<input type="checkbox"/> dns_hosts		

Run Commands

Command category the rule applies to: Any Command Specified Commands and Groups

Allow

<input type="checkbox"/> Polecenia zezwolone sudo	Usuń Dodaj
<input type="checkbox"/> /usr/bin/su	
<input type="checkbox"/> /bin/su	
<input type="checkbox"/> Sudo Allow Command Groups	Usuń Dodaj



Linki

Resetowanie haseł

- Self-Service Password Reset

<https://github.com/pwm-project/pwm/>

- Self-Service user account management tools

<https://github.com/ubccr/mokey>



Dziękuję za uwagę 😊

Q&A