Zdokonalení integrace SSSD a SUDO

Michal Šrubař
xsruba03@stud.fit.vutbr.cz

Obsah

1 Administrace linuxového systému **1**
- 1.1 su (Substitute User) . . . 1

2 SUDO (Super User DO) **1**
- 2.1 SUDO vs. su . . . 1
- 2.2 Distribution of the sudoers file among multiple systems . . . 2
- 2.3 Why is SUDO in LDAP better solution? . . . 2
- 2.4 SUDO on LDAP server . . . 2
- 2.5 Benefits of having sudoers on LDAP server . . . 2
- 2.6 Difference between LDAP sudoers and non-LDAP sudoers . . . 3
- 2.7 Drawbacks of having sudoers in LDAP . . . 3
- 2.8 Sudo a pluginy . . . 4
- 2.9 What is meant by LDAP server? . . . 4
- 2.10 ADD or NOT? . . . 4

3 FreeIPA **4**
- 3.1 Advantages the new schema brings . . . 4
- 3.2 How are SUDO rules stored in LDAP on IPA server? . . . 5

4 SSSD **5**
- 4.1 Benefits of the SSSD . . . 5

5 Proposed changes **5**

6 Jak to funguje teď **5**
- 6.1 compat plugin . . . 5
- 6.2 jak jsou uložena pravidla v IPA . . . 5
- 6.3 externí uživatele . . . 6

Hlavní problém, který moje BP řeší

FreeIPA používá pro uložení SUDO pravidel v LDAPu svoje schéma, které přináší mnoho vylepšení. **SSSD** ovšem toto schéma neumí zpracovat (FreeIPA nejprve používala legacy sudo schema pro uložení a pak se navrhlo nové schéma). FreeIPA musí nové schéma přeložit do legacy schématu (provádí to compat plugin), které sssd umí zpracovat. Převodem se ovšem ztrácí mnoho informací, které potom způsobují problémy a brzí to tak využití celého potenciálu.

Jelikož SSSD neumí to FreeIPA schéma, tak se musí dodatečně stahovat mnoho informací, což u rozsáhlé databáze způsobuje značné zpomalení. Úkolem není navrhnout nové LDAP schéma pro SUDO, ale upravit SSSD daemon a SUDO tak, aby podporovaly LDAP schéma, které používá IPA.

Hodně zjednodušeně bude výsledkem správný dotaz na LDAP adresář na IPA serveru, který pro daného klienta stáhne všechno potřebné.

1 Administrace linuxového systému

Na většině dnešních linuxových a Unix-like[1] operačních systémech provádí správu systému speciální uživatel zvaný **root**. Někdy také označován jako **superuser**. Tento uživatel má přístup ke všem souborům a může spouštět všechny příkazy operačního systému[2]. Ostatní uživatele systému toto právo nemají, až na tyto vyjímky:

- uživatel zná heslo uživatele root
- uživatel použije program, který mu dočasně poskytne práva uživatele root

1.1 su (Substitute User)

Tento program umožňuje libovolnému uživateli stát se dočasně jiným uživatelem. Vě většině případů je jiný uživatel právě uživatel root. To znamená, že program su může dát libovolnému uživateli, který zná heslo uživatele root, plnou kontrolu nad operačním systémem. In a nutshell sudo provides a way to give selective root access by user/machine/command.

2 SUDO (Super User DO)

Sudo allows to a user to execute one command as a different user. Security policies which defines which users are allowed to execute what commands are called *sudo rules*. These rules are stored in configuration file `/etc/sudoers`. The file should be edited only with tool *visudo* which provides automatic syntax checking.

2.1 SUDO vs. su

These are two different concepts of giving root access to a user.

- sudo allow limited access but su gives a user a full control over the system

[1] http://en.wikipedia.org/wiki/Unix-like
[2] http://www.linfo.org/root.html

- none of users need to know the root's password
- using sudo can be logged
- it's possible to define which users can execute which commands

2.2 Distribution of the sudoers file among multiple systems

SUDO doesn't have a native way to distribute *sudoers* file among multiple clients so the administrators in corporate environemnts faces a problem. How to distribute the sudo rules to all machines he administers? There is a few solutions:

1. Administrator can manualy distribute the *sudoers* file among the systems he administers with starndar UNIX tools such as *cron, scp, rsync* etc.
2. Use tools such as Puppet[3] which automaticaly watch and distribute files among multiple systems.
3. Use centralized database. If we do so administrator just insert sudo rule into database and all the clients can use it

2.3 Why is SUDO in LDAP better solution?

LDAP [4]is characterized as a "write-once-read-many-times"service. It's eminently suitable for maintaining data which are not changed very often. Sudo rules are set once by administrator of the system but accessed by users many times. Administrator will also want to edit the rules but not as often as clients will access it.

Directories are faster that databases because they don't require consistency as much as relational or transactional databases. It doesn't use transactions, locking or roll-backs[5].

2.4 SUDO on LDAP server

In most today's organization environments there is a LDAP server which can be used to centraly manage user's information. These information can be accessed with LDAP protocol. SUDO itself has buit-in support for proccessing sudo rules stored on LDAP server.

2.5 Benefits of having sudoers on LDAP server

1. if the sudo rules are centralized than the administrator doesn't have to worry about distributing them
2. Looking up sudo rules is faster because sudo no longer needs to read entire *sudoers* file. There are only a few LDAP queries per invocation.

[3]http://puppetlabs.com/puppet/what-is-puppet

[4]Lightweight Directory Access Protocol, which is an application protocol for querying and modifying directory services.

[5]www.zytrax.com/books/ldap/ch2/

3. If there is a typing error in *sudoers* than sudo won't start. With LDAP it's not possible to load data into the LDAP directory which does not conform the sudoers schema so the proper syntax is guaranted. Although you can still make a mistake in user name, host name or command. There is no need to use visudo[6].

2.6 Difference between LDAP sudoers and non-LDAP sudoers

- The biggest difference, according to the LDAP RFC, is that LDAP ordering of attributes is arbitrary which means that you can not expect that attributes are returned in any specific order. Let's suppose that we have these two rules in *sudoers* file:

```
adam ALL=/bin/cat /etc/shadow
adam ALL=!/bin/cat /etc/shadow
```

  There's an order in reading *sudoers* which means adam will not be able to print content of the */etc/shadow* file. The same rule in LDAP would look like this:

```
# LDAP equivalent of adam's␣sudo␣rules
dn: cn=role1,ou=Sudoers,dc=example,dc=com
objectClass: sudoRole
objectClass: top
cn: role1
sudoUser: adam
sudoHost: ALL
sudoCommand: /bin/cat
sudoCommand: !/bin/cat
```

  Since there's an arbitrary ordering of attributes there's no guarante of what sudo rules the client will receive as the last one. Which means we are not file or not. This is the reason why there is the `SudoOrder` attribute. Take a closer look at this!

- `User_Aliases, RunAs_Aliases and Cmnd_Aliases` are not supported
- `User_Aliases` can be replaced with groups and netgroups which can also be stored in LDAP
- `Cmnd_Aliases` are not needed becase it's possible to add more `sudoCommand` in one `sudoRole`
- `/etc/sudoers` file uses global default options but in LDAP it's possible to specify per-entry options.

Having sudoers in LDAP has also a few drawbacks. These are discussed in section 2.7.

2.7 Drawbacks of having sudoers in LDAP

- Administrator has to configure everything via text files. Sudo does not provide any GUI.
- Corporation don't want to maintain their users information a security policies separetly. They want to have those kind of information at one place so it can be link together.

[6] utility which provides save editing of the *sudoers* file

- If there are users information and sudo rules stored in LDAP than there is no relationship among them. rules.
- `User_Aliases, RunAs_Aliases and Cmnd_Aliases` are not supported
- `User_Aliases` can be replaced with groups and netgroups which can also be stored in LDAP
- `Cmnd_Aliases` are not needed becase it's possible to add more `sudoCommand` in one `sudoRole`
- `/etc/sudoers` file uses global default options but in LDAP it's possible to specify per-entry options.

2.8 Sudo a pluginy

Sudo Plugin API[7] umožňuje vytvoření vlastního modulu, který bude definovat vlastní správu politik. To jaké pluginy bude sudo používat je možné konfigurovat pomocí souboru */etc/-sudo.conf*.

2.9 What is meant by LDAP server?

Technically, LDAP is just a protocol that defines the method by which directory data is accessed. It also defines and describes how data is represented in the directory service.

2.10 ADD or NOT?

1. example of sudoers configuration and use?
2. Explain meaning of all `sudoRole` attributes and SUDOers container?

3 FreeIPA

FreeIPA uses LDAP directory to store information such as users information for authentication, dns records etc.

3.1 Advantages the new schema brings

- it's possible to disable a rule without deleting it
- Identities, Hosts and Commands uses DNs instead of string attributes

T FreeIPA používá vlastní schéma a proto musela napsat i vlastní plugin sss.

Výhody použití FreeIPA pro správu sudo pravidel - u sudo+openldap musím vytvořit schéma a ručně jej přidat mezi ostatní schémata + přidat o tom záznam do /etc/sldap.conf

[7] http://www.sudo.ws/sudo_plugin.man.html

3.2 How are SUDO rules stored in LDAP on IPA server?

IPA uses two containers for storing SUDO rules. A *sudoers* container which contains SUDO rules in legacy scheme which consists of these attributes:

- sudoHost
- sudoCommand
- sudoUser

Then there is a *sudo* container and this contains

4 SSSD

4.1 Benefits of the SSSD

unified configuration - instead of configuring the SSSD to perform account lookups and then configuring /etc/ldap.conf to perform sudo rules lookups, the user only configures one client piece - the SSSD. The SSSD also provides several advanced features that might not be available in other LDAP client packages, such as the support for server discovery using DNS SRV requests or advanced server fail over, which lets the admin define several servers that are tried in descending order of preference and then stick to the working server, by Kuba

5 Proposed changes

6 Jak to funguje teď

6.1 compat plugin

Ten compat plugin pracuje za běhu IPA nebo ty pravidla přeloží až, když o ně požádá SSSD?

Myslim ze cely compat strom se generuje v zavislosti na dotazech klientu. Tj na serveru je plugin, ktery vi, ze kdyz dojde dotaz na ou=sudoers,$DC, tak se ma interne zeptat do IPA sudo kontejneru a pokud neco dostane zpet, vlozi vysledek do compat pluginu, kde si ho precte klient co provadel search.

Klient nemusi byt jen sudo, muze to byt cokoliv, co se umi bavit LDAP protokolem.

6.2 jak jsou uložena pravidla v IPA

V kontejneru sudores sjou povolené pravidla v takovém formátu, které umí zpracovat SUDO. V sudo kontejneru jsou pravidla ve formátu, který používá IPA. Ten plugin to překládá za běhu a SSSD si to tahá ze sudoers kontejneru? Když potřebuje další informace, tak si je dodatečně stáhne?

Jo, myslim ze jsi to popsal dobre.

6.3 externí uživatele

Funguji momentálně ti externí uživatele. Pravidlo se mně stáhne v pořádku, ale sudo mně řekne, že uživatel není v SUDOers.

Externi uzivatele jsou hlavne na to, kdyz mas nektere uzivatele ulozene na jinem serveru nez IPA (typicky Active Directory). Nevim jestli jde pomoci externich uzivatelu vlozit treba uzivatele z /etc/passwd, ale spis bych rekl, ze ne.

Myslim ze tomuhle se v praci nemusis nijak venovat, max popsat, ze to tam je.

Možná struktura textu

- Problémy a nedostatky defaultního LDAP shématu, které SUDO používá
  - na uživatele se odkazuje pouze přes řetězec jména, není tam žádná provázanost s počítačem ze kterého dany uživatel je
  - pokud LDAP spravuje i uživatele, a je tam stejný uživatel jako na lokální mašině, tak tam vzniká nějaký problém s překrytím těch uživatelů
  - mezi uživateli a sudo pravidly neexistuje vazba, pokud odstraním uživatele, tak mně zůstane pravidlo s neexistujícím uživatelem
- Projekt FreeIPA
- Proč sudo ve FreeIPA (?granularita?)
- Proč má IPA vlastní SUDO LDAP schema a jaké výhody přináší
  - umožňuje např. povolit/zakázat pravidlo
- Jak to funguje teď
  - definice pravidla
  - jak se uloží v LDAPu
  - co tam bude jinak oproti tomu než bych to uložil do LDAPu s legacy schematem
  - konverze do legacy schématu když k tomu přistoupím přes SSSD
  - cachování
- Proč musí IPA nové schéma přeložit do legacy schematu než jej pošle SSSD
- Jaké problémy to přináší
- Jaké úpravy je nutné provést
  - u sudo bude třeba přenést něco z /lib do /usr (z ticketu: pe_task, cron)
- Vyzdvihnout všechny přednosti