Zdokonalení integrace SSSD a SUDO

Michal Šrubař
xsruba03@stud.fit.vutbr.cz

Obsah

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.

1.2 sudo (Super User DO)

Sudo umožňuje uživateli provést jeden příkaz jako jiný uživatel. Bezpečností politiky, které definují kteří uživatelé mohou spouštět jaké příkazy se nazývají **sudo pravidla**. Tyto pravidla jsou uloženy v souboru */etc/sudoers*. Sudo pravidla by měla být upravována výhradně pomocí utility **visudo**, která zároveň provádí kontrolu syntaxe tohoto souboru.

1.2.1 výhody sudo oproti su

Jedná se dva naprosto odlišné přístupy k tomu jak získat oprávnění uživatele root.

- sudo umožňuje limitovat přístup zatímco su může uživateli dát plnou kontrolu nad systémem

[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 SUDO

2.1 Distribute sudoers file among multiple systems

SUDO doesn't have 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 they 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. store sudo rules in centralized database such as LDAP[4].

2.2 Why is SUDO in LDAP better solution?

LDAP 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].

2.3 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.

Explain meaning of all sudo LDAP attributes?

2.4 Benefits of having sudoers on LDAP server

1. 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/

2. 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].

2.5 Difference between sudoers and LDAP

- 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.

- `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.

2.5.1 Centrálním uložiště

2.5.2 LDAP adresář

Ve většině firemních prostředí se dnes používá adresářových služeb k synchronizaci uživatelů, skupin a dalších sdílených informací. Sudo pravidla mohou být na tomto serveru také uložena. Přistoupit k nim je možné pomocí protokolu LDAP. Při použití adresářového serveru je možné sudo pravidla centrálně spravovat a globálně k nim přistupovat. Sudo ovšem takto nativně nepracuje proto je potřeba plugin. Sudo má plugin pro použití s openLDAP. Schéma v jakém jsou sudo pravidla na LDAP serveru uložena je pevně specifikováno a přináší následující nevýhody:

1. problém s tím překrýváním lokálních uživatelů
2. další

Pokud chce administrátor tento způsob správy politik využít, pak musí dané schéma dodržet i se všemi jeho nevýhodami.

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

2.6 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*.

2.7 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.

3 FreeIPA

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

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

section keywords: LDAP, IPA, SUDO legacy schema, IPA SUDO schema, container

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 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 Changes

section keywords: changes, proposal, sssd, sudo, ipa, ldap sudo scheme, sudoers container, sudo container

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

Možná struktura textu

- Smysl BP.
- Uživatelé, root, jak dát běžným uživatelům vyšší práva
- Sudo, su, alternativy
- Proč sudo?
- Proč sudo v LDAPu? (alternativy jak a kam ukládát sudo pravidla, např. puppet)
  - Složitost konfigurace a správy (všechno v textu, není GUI, které FreeIPA má)
  - Firma s mnoha zaměstnanci chce centrálně spravovat informace a politiky o svých uživatelích (chtěla by mít vše na jednom místě a né informace v ldap, sudo pravidla v LDAPu ale nebylo by to nijak propojeno, konfiguráky v puppetu, atd...)
  - sudo pravidla nejsou nijak propojena s ostatními informacemi, které mohou a často jsou v LDAPu uloženy (uživatelé, skupiny, atd)
- 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