Section Overview

What You Will Learn

- PAM
- Linux account types
- Password security
- Shadow passwords

What You Will Learn

- Managing account expiry data
- Locking/unlocking accounts
- Monitoring authentication logs
- Multifactor authentication

Account Security

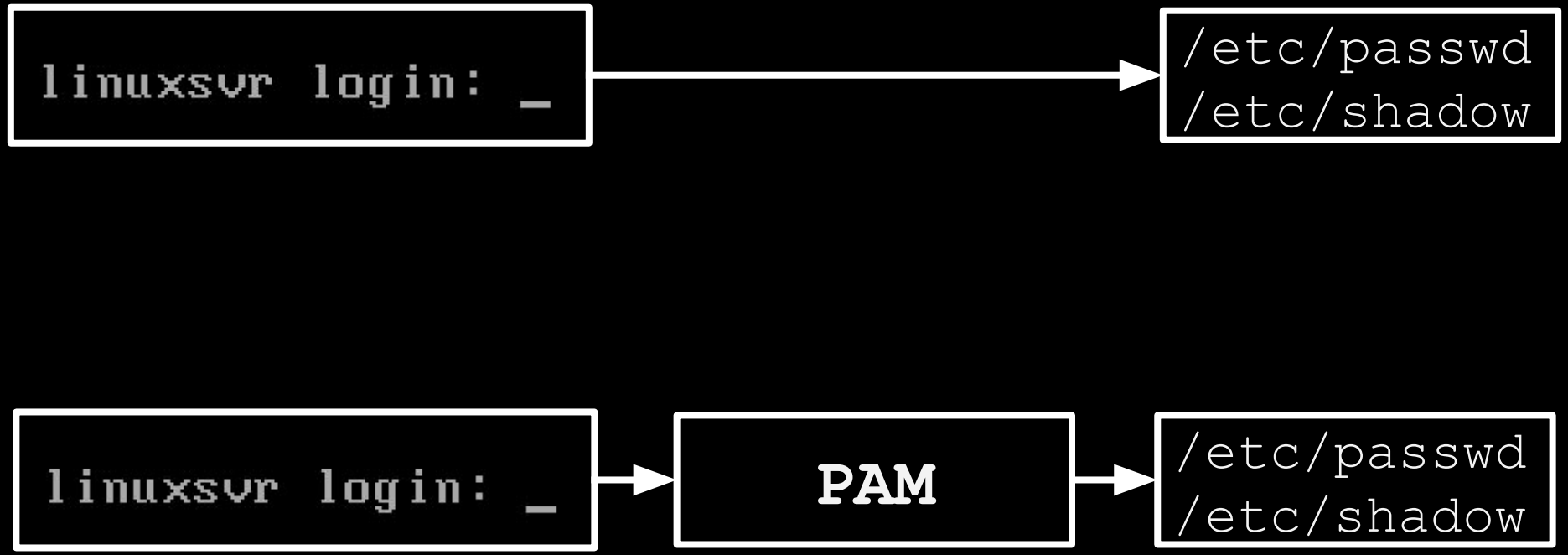
Linux Account Security

- It's easier to attack a system from the inside.
- Privilege escalation attacks are a threat.
- Keep unwanted users out.
- Secure accounts.

PAM

Pluggable Authentication Modules


linuxsvr login: _
/etc/passwd
/etc/shadow


linuxsvr login: _
/etc/passwd
/etc/shadow
linuxsvr login: _
PAM
/etc/passwd
/etc/shadow

linuxsvr login: _

/etc/passwd
/etc/shadow

linuxsvr login: _

PAM

PAM Configuration files

Location:

```
/etc/pam.d
/etc/pam.d/login
/etc/pam.d/sshd
```

Format:

```
module_interface control_flag module_name module_args
```

PAM Module Interfaces

- auth - Authenticates users.
- account - Verifies if access is permitted.
- password - Changes a user’s password.
- session - Manages user’s sessions.

PAM Control Flags

- required - Module result must be successful to continue.
- requisite - Like required, but no other modules are invoked.
- sufficient - Authenticates user if no required modules have failed, otherwise ignored.
- optional - Only used when no other modules reference the interface.
- include - Includes configuration from another file.
- complex control flags - attribute=value

PAM Configuration Example

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_unix.so nullok
auth       required     pam_nologin.so
account    required     pam_unix.so
password   required     pam_pwquality.so retry=3
password   required     pam_unix.so shadow \
                            nullok use_authtok
session    required     pam_unix.so
```

```
#%PAM-1.0
auth       required   pam_securetty.so
auth       required   pam_unix.so nullok
auth       required   pam_nologin.so
account    required   pam_unix.so
password   required   pam_pwquality.so retry=3
password   required   pam_unix.so shadow \
                          nullok use_authtok
session    required   pam_unix.so
```

```
#%PAM-1.0
auth       required   pam_securetty.so
auth       required   pam_unix.so nullok
auth       required   pam_nologin.so
account    required   pam_unix.so
password   required   pam_pwquality.so retry=3
password   required   pam_unix.so shadow \
                          nullok use_authtok
session    required   pam_unix.so
```

```
#%PAM-1.0
auth       required   pam_securetty.so
auth       required   pam_unix.so nullok
auth       required   pam_nologin.so
account    required   pam_unix.so
password   required   pam_pwquality.so retry=3
password   required   pam_unix.so shadow \
                          nullok use_authtok
session    required   pam_unix.so
```

```
#%PAM-1.0
auth       required   pam_securetty.so
auth       required   pam_unix.so nullok
auth       required   pam_nologin.so
account    required   pam_unix.so
password   required   pam_pwquality.so retry=3
password   required   pam_unix.so shadow \
                          nullok use_authtok
session    required   pam_unix.so
```

```
#%PAM-1.0
auth       required   pam_securetty.so
auth       required   pam_unix.so nullok
auth       required   pam_nologin.so
account    required   pam_unix.so
password   required   pam_pwquality.so retry=3
password   required   pam_unix.so shadow \
                          nullok use_authtok
session    required   pam_unix.so
```

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_unix.so nullok
auth       required     pam_nologin.so
account    required     pam_unix.so
password   required     pam_pwquality.so retry=3
password   required     pam_unix.so shadow \
                            nullok use_authtok
session    required     pam_unix.so
```

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_unix.so nullok
auth       required     pam_nologin.so
account    required     pam_unix.so
password   required     pam_pwquality.so retry=3
password   required     pam_unix.so shadow \
                            nullok use_authtok
session    required     pam_unix.so
```

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_unix.so nullok
auth       required     pam_nologin.so
account    required     pam_unix.so
password   required     pam_pwquality.so retry=3
password   required     pam_unix.so shadow \
                            nullok use_authtok
session    required     pam_unix.so
```

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_unix.so nullok
auth       required     pam_nologin.so
account    required     pam_unix.so
password   required     pam_pwquality.so retry=3
password   required     pam_unix.so shadow \
                                nullok use_authtok
session    required     pam_unix.so
```

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_unix.so nullok
auth       required     pam_nologin.so
account    required     pam_unix.so
password   required     pam_pwquality.so retry=3
password   required     pam_unix.so shadow \
                            nullok use_authtok
session    required     pam_unix.so
```

PAM Documentation

Configuration:

```
account   required   pam_nologin.so
session   required   pam_unix.so
```

Getting help:

```
man pam_nologin
man pam_unix
```

Linux Account Types

`root`, the superuser

- Root can do *anything*.
- Always has the UID of 0.

Account Security Demo

System accounts

- UIDs < 1,000
- Configured in /etc/login.defs
- `useradd -r system_account_name`

Normal User Accounts

- UIDs >= 1,000
- Intended for human (interactive) use

Password Security

- Enforce, not hope for, strong passwords.
- Use pam_pwquality, based on pam_cracklib.
  - Configuration File:
    `/etc/security/pwquality.conf`
  - PAM Usage:
    `password requisite pam_pwquality.so`
  - Module attributes:
    `man pam_pwquality`

Use Shadow Passwords

- /etc/passwd unencrypted:

```
root:$6$L3ZSmlM1H5:0:0:root:/root:/bin/bash
```

Use Shadow Passwords

- /etc/passwd unencrypted:

```
root:$6$L3ZSmlM1H5:0:0:root:/root:/bin/bash
```

- /etc/passwd with shadow passwords:

```
root:x:0:0:root:/root:/bin/bash
```

- /etc/shadow:

```
root:$6$L3ZSmlM1H5::0:99999:7:::
```

Converting Passwords

`pwconv` - convert to shadow passwords.

`pwunconv` - convert from shadow passwords.

/etc/shadow format

Username

Hashed password

Days since epoch of last password change

Days until change allowed

Days before change required

Days warning for expiration

Days before account inactive

Days since epoch when account expires

Reserved

Display user account expiry info with chage

`chage -l account` - Show account aging info.

```
$ chage -l jason
Last password change : Apr 01, 2016
Password expires     : never
Password inactive    : never
Account expires      : never
Minimum number of days between password change    : 0
Maximum number of days between password change    : 99999
Number of days of warning before password expires : 7
```

Change user account expiry info with chage

`-M MAX_DAYS` - Set the maximum number of days during which a password is valid.

`-E EXPIRE_DATE` - Date on which the user's account will no longer be accessible.

`-d LAST_DAY` - Set the last day the password was changed.

/etc/login.defs

```
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
PASS_MIN_LEN     5
PASS_WARN_AGE    7
```

Password History

PAM directive:

```
password required pam_pwhistory.so

# remember=N
```

Controlling Account Access

Locking and Unlocking accounts

```
passwd -l account
```

```
passwd -u account
```

Locking with `nologin` as the Shell

Example /etc/passwd entries:

```
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

Locking with `nologin` as the Shell

Example /etc/passwd entries:

```
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

Using `chsh`:

```
chsh -s SHELL ACCOUNT
chsh -s /sbin/nologin jason
```

Centralized Authentication

- Easy to manage users system-wide.
  - lock account everywhere
- Example authentication systems:
  - freeIPA
  - LDAP (openLDAP)
- Has drawbacks too.

Disable Logins

- pam_nologin module
  - Looks for /etc/nologin or /var/run/nologin
  - Disables logins and displays contents of nologin file.

Monitoring Authentication Logs

```
# last
jason  pts/0  10.11.12.13 Mon Feb 1 19:22 still logged in
jason  pts/0  10.11.12.14 Mon Feb 1 19:04 - 19:21  (00:16)
ralph  pts/0  www01       Mon Feb 1 19:04 - 19:04  (00:00)
root   pts/0  thor        Mon Feb 1 19:04 - 19:04  (00:00)
# lastb
jason  pts/1  10.11.12.14 Mon Feb 1 18:54 - 18:54  (00:00)
# lastlog
Username      Port      From             Latest
root          pts/0     thor Mon Feb  1 19:04:13 -0500 2016
```

Monitoring Authentication Logs

/var/log/messages

/var/log/syslog

/var/log/secure

/var/log/auth.log

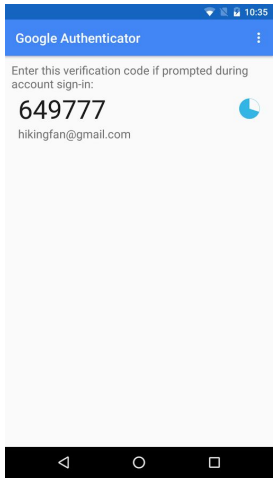
** Depends on syslog configuration. **

Intrusion Prevention with fail2ban

- fail2ban.
- Monitors log files.
- Blocks IP address of attacker.
- Automatic unban.
- Not just for Linux logins.

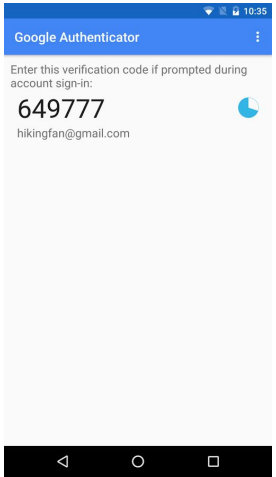
Multifactor Authentication

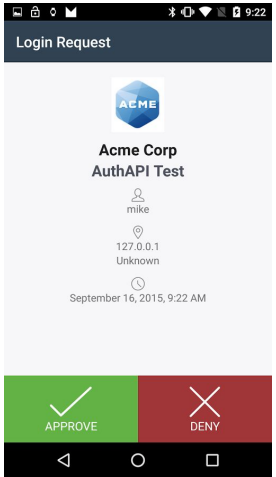
- Google Authenticator PAM module



Multifactor Authentication

- Google Authenticator PAM module
- DuoSecurity's pam_duo module

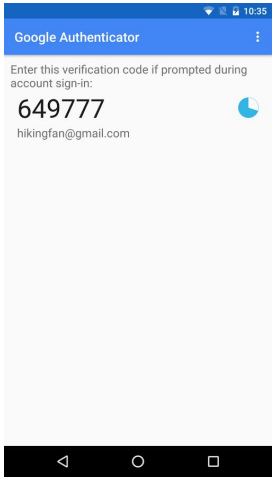


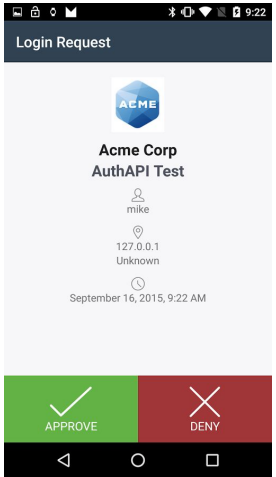


LinuxTrainingAcademy.com

Multifactor Authentication

- Google Authenticator PAM module
- DuoSecurity's pam_duo module
- RSA SecurID PAM module





Security by Account Type

Account Security - root

- Use a normal account for normal activities.
- Avoid logging in as root.
- Use sudo instead of su.
- Avoid using the same root password.
- Ensure only the root account has UID 0.

```
awk -F: '($3 == "0") {print}' /etc/passwd
```

Disabling root Logins

- pam_securetty

```
auth [user_unknown=ignore success=ok \
ignore=ignore default=bad] pam_securetty.so
```

  - /etc/securetty
    - console
    - tty1

Account Security Demo

Disabling SSH root Logins

```
/etc/ssh/sshd_config

PermitRootLogin no

systemctl reload sshd
```

System / Application Accounts

- Use one account per service.
  - web service (httpd), web service account (apache)
- Don’t activate the account.
- Don’t allow direct logins from the account.
  - sshd_config: DenyUsers account1 accountN
- Use sudo for all access.
  - `$ sudo -u apache apachectl configtest`

User Accounts

- One account per person.

Deleting Accounts

- Determine the UID
  - `id ACCOUNT`
- Delete their account and home directory
  - `userdel -r`
- Find other files that belong to them on the system.
  - find / -user UID
  - find / -nouser

Using and Configuring Sudo

sudo vs su

- “SuperUser Do” or “Substitute User Do”
- Use instead of the `su` command.
- Complete shell access with `su`.
- With `su` you need to know the password of the other account.
- Breaks the Principle of Least Privilege.
- Vague audit trail with `su`.

Sudo

- Fine grain controls.
- No need to share passwords.
- Clear audit trail.

Sudo configuration

- Sudo configuration live in `/etc/sudoers`.
- Use `visudo` to make changes.
  - Syntax checking
- Additional configuration in `/etc/sudoers.d`.
  - `visudo -f /etc/sudoers.d/file_name`
- Use `EDITOR` to control `visudo`.
  - `export EDITOR=nano`

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel ALL=(ALL)        ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel ALL=(ALL)        ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel  ALL=(ALL)       ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel ALL=(ALL)        ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel  ALL=(ALL)       ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel  ALL=(ALL)       ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root)  /sbin/apachectl
%web    web*=(root)      /sbin/apachectl
%wheel  ALL=(ALL)        ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel  ALL=(ALL)       ALL
```

Sudoers Format

User Specification Format:

```
user host=(run_as) command
```

Examples:

```
jason   webdev01=(root) /sbin/apachectl
%web    web*=(root)     /sbin/apachectl
%wheel  ALL=(ALL)       ALL
```

Sudo Authentication

- Sudo requires a user to authenticate.
- Default 5 minute grace period (timeout).
- You may not want to use a password.

NOPASSWD & PASSWD

```
apache web*=(root) NOPASSWD:/sbin/backup-web
```

```
NOPASSWD:/sbin/backup-web, PASSWD:/sbin/apachectl
```

Sudo Aliases

- User_Alias
- Runas_Alias
- Host_Alias
- Cmnd_Alias

Format:

Alias_Type NAME = item1, item2, ...

Sudo Aliases

```
User_Alias  WEBTEAM = jason, bob


WEBTEAM web*=(root)   /sbin/apachectl
WEBTEAM web*=(apache) /sbin/apachebackup
```

Sudo Aliases

```
Runas_Alias  WEBUSERS = apache, httpd


WEBTEAM web*=(WEBUSERS) /sbin/apachectl
```

Sudo Aliases

```
Host_Alias  WEBHOSTS = web*, prodweb01


WEBTEAM WEBHOSTS=(WEBUSERS) /sbin/apachectl
```

Sudo Aliases

```
Cmnd_Alias  WEBCMNDS = /sbin/apachectl


WEBTEAM WEBHOSTS=(WEBUSERS)WEBCMNDS
```

Sudo Aliases

```
User_Alias    WEBTEAM  = jason, bob
Runas_Alias   WEBUSERS = apache, httpd
Host_Alias    WEBHOSTS = web*, prodweb01
Cmnd_Alias    WEBCMNDS = /sbin/apachectl

WEBTEAM WEBHOSTS=(root) /sbin/apachebackup
WEBTEAM WEBHOSTS=(WEBUSERS) WEBCMNDS
```

Displaying the Sudo Configuration

- List commands you are allowed to run:

```
sudo -l
```

- Verbose listing of commands:

```
sudo -ll
```

- List commands another USER is allowed:

```
sudo -l -U user
```

Running Commands with `sudo`

- Run as root:

```
sudo COMMAND
```

- Run as USER:

```
sudo -u USER COMMAND
```

- Get a shell

```
sudo -s
```

```
sudo -s -u USER
```

Account Security Demo

Section Summary

Summary

- PAM
- Linux account types
- Password security
- Shadow passwords

Summary

- Managing account expiry data
- Locking/unlocking accounts
- Monitoring authentication logs
- Multifactor authentication
- Sudo