Lab 11: When disaster strikes

Hands-on Unix system administration DeCal

2012-04-02, due 2012-04-09

For this lab, you will be using your OCF account and project group VM[1] . Instructions on accessing the project group VM are at the end of this lab.

Make yourself at home

Now that your project group has its own system, you want to create your own user account on it. It's generally bad practice to login as root all the time.

Provide two reasons why logging in as root might not be a good idea.

Creating your account

Let's access the project group VM (instructions are at the end of the lab). As root, run the command `adduser yourusername` to create your account, and follow the prompts.

Making yourself an administrator

As an alternative to logging in as root, many Unix-like systems advocate the use of sudo, a setuid program that allows some users to run programs with the security privileges of another user (often root).

What is a setuid binary? Why might the sudo program need to have the setuid permission?

Settings for the sudo program are set in `/etc/sudoers`. By default in Debian, users in the sudo group are allowed complete access to sudo. Users in the sudo group therefore have complete root access by means of sudo, and can impersonate any other user—this is not to be taken lightly.

Add yourself to the sudo group with the command `usermod -a -G sudo yourusername`. The `-a` option appends the sudo group to your list of groups, and the `-G` option adds sudo as a secondary group.

Locking down the superuser

From now on, you don't need to login as root or have access to root's password or private key. Simply prepend any command that needs to be run as root with the word sudo (e.g., `sudo make me a sandwich`[2]), or run the command `sudo -i` to obtain a root prompt without needing to login as root (this is not usually a good idea, unless if you are really lazy).

Let's change the root password and see who can login through SSH public key authentication. Run these commands one at a time. The hash symbol (#) and everything after it is a comment.

```
sudo -i # become root
passwd # change root's password
less ~/.ssh/authorized_keys
```

Describe the contents of `authorized_keys`. Who has access through SSH public key authentication? The `authorized_keys` file lists the public keys that can be used for logging in as that user.

[1] Virtual Machine, we have several "virtual computers" which run on a single physical computer

[2] `http://xkcd.com/149/`

Becoming conscious of security

A good sysadmin hopes for the best but assumes the worst. She or he is also highly devoted[3].

DoS attack

Assume the worst. Your fellow project group members are thugs who want to make sure you fail the class. They want to prevent you from finishing the decal project. Because you've thoughtfully limited root access (by changing the root password, removing their public key from the `authorized_keys` file, made yourself the only member of the sudo group, killed any processes that they may still be running as root), they can't login as root. Good!

They still have the user accounts they created earlier in this lab, just not root access via sudo or login.

If your project group members are mischievous, how might they perform a denial of service attack on the system to increase the system load and prevent you from getting anything done? (If you're stuck and can't think of anything, find out how to perform a fork bomb). Also, please don't actually do this... educational purposes only.

How might you prevent a fork bomb? Hint: `man limits.conf`. If you're still stuck, take a look at `/etc/security/limits.conf` on `tsunami.ocf.berkeley.edu`.

Setuid permissions

Why might setuid permissions be a concern? List at least one other setuid binary that exists on the system (bonus points if you list them all). Your work will be a lot easier if you use the `find` command (`man find`, or look on the web) with the `-perm` option.

Do logs not logarithms

The `rsyslog` daemon stores messages generated by the system in `/var/log`. By default in Debian, a copy of most messages is also included in `/var/log/syslog`. Take a look at `/var/log/syslog`.

Provide one example each of a syslog message containing a successful and unsuccessful login attempt. Authentication logging is by default in Debian stored in `/var/log/auth.log`.

Snooping with publicly-accessible information

Sometimes system logs make you suspicious of a user, but because of privacy concerns, you don't want to look at private information. So look at publicly-accessible information instead.

One of the facilitators wrote a script called `check` to aggregate some publicly-accessible information on a given OCF user. SSH into `tsunami.ocf.berkeley.edu` with your OCF account, and run `check` on one of your project group partners. See `check --help` for syntax.

What are two commands that `check` runs to "look" for information? Hint: run `how check`, how is another custom OCF script.

Bonus: look in CalNet

CalNet is an LDAP directory with both private and publicly-accessible information. Almost a year ago, OCF users started being associated with their CalNet UID numbers (retroactively to the mid-1990's), a unique identifier in CalNet. This allows online password resets[4] to be authenticated using CalNet.

Where available, `check` will report the associated CalNet UID number. You can search the CalNet directory for publicly-accessible attributes in that entry.

```
ldapsearch -x -H ldap://ldap.berkeley.edu -b dc=berkeley,dc=edu calnet_uid
```

[3] `http://xkcd.com/705/`

[4] `https://secure.ocf.berkeley.edu/account_tools/change_password`

Accessing project group VM

Instructions

Here are commands to run from a standard GNU/Linux machine that your private key identity is stored on. You should understand what they do (you were asked this question in the last lab).

```
exec ssh-agent bash
ssh-add ~/.ssh/identityfile
ssh -A ocfusername@coupdetat.ocf.berkeley.edu
ssh root@decalserver
```

where,

identityfile the corresponding private key to the one you provided to the facilitators

ocfusername your OCF username

decalserver your project group's assigned server, as indicated in the table below

Server assignments

Server	Project group
alpha	Abe's Avengers
bravo	Nooby Penguins
charlie	Tonight We Dine In Shell!
delta	I want an A
echo	aber
foxtrot	GN00bz
golf	assASCIIns
hotel	Honey Badger