Security Part 2

Before getting into any directions on specific tasks we will be running there is a few commands we must understand and have under our belt. Cronjob gives us many ways to automate and fortify out servers.

We will use crontab, it is a file that allows us to edit and schedule scripts or commands specified by the time we choose.

To view crontab use the command here:

**#crontab –l**

To edit the contents and enter commands/scripts and schedule times we will use this command:

**#crontab –e**

There is a certain syntax used for this editor. First entering in the time for the needed command/script to run, then the path to the specified command/script. In addition to any arguments you would like to use.

The time element will be valued in certain field such as this:

Day of week (0 – 7)

Month (1 – 12)

Day of the month (1 – 31)

Hour (0 - 23)

Minute (0 – 59)

Reference to crontab command: <https://www.digitalocean.com/community/tutorials/how-to-use-cron-to-automate-tasks-centos-8>

1. Take a snapshot of users every hour (Use a cron job for this) to see if there is any suspicious adding/removing of users

To view all users that are logged on we will need to run the “**sha1sum**” command and then “**cat”** the output. Which will display the contents of the specified file path and output a text file. To view users logged on we need to point to the /etc/passwd file. So for example the syntax will be:

**/etc/passwd >> /root/class/linux/user\_data.txt**

**/etc/passwd >> /root/class/linux/users.txt**

Since we want to do this every hour, we will add operators and the numbers from the values from above. So, it will look like something this:

**0 \*/1 \* \* \* /etc/passwd >> /root/class/linux/user\_data.txt**

**0 \*/1 \* \* \* /etc/passwd >> /root/class/linux/users.txt**

Reference for sha1sum command: <https://shapeshed.com/unix-sha1sum/>

1. Write a document that will show how to control what daemons run on boot and how to change that.  assume your audience is technically inclined, but not an expert.

To view and edit what services start and run-on boot we must view what daemons run already on boot. We will be using the **systemctl** command throughout this process. Let’s start by listing our services that are currently running by using this command:

**# systemctl list**

If there is a specified service we want to view, whether we want to make sure its enabled or disabled or just to see if it on our system we will run this command:

**# systemctl status “service name”**

Disabling a service

For example, we will check the status of sshd by using this command:

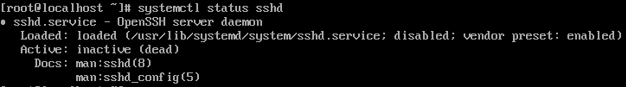
**# systemctl status sshd**

This displays if the service is loaded, active and the amount of memory usage it is using. We can now disable it by using this command:

**# systemctl disable sshd**



*Note: Disabling this service will now be disable until we re-enable the process, even after a reboot. Ensure you are not disabling a service that may be dependent on something else. This can break the server and not allow a correct reboot.*



Enabling a service

Now that we have disabled a service, to enable a service, we use the same command but change out disable to enable. For example, we will use this command:

**# systemctl enable “service name”**

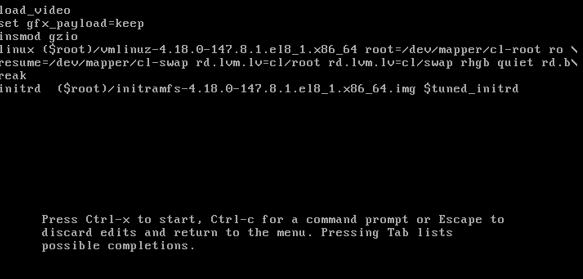
Reference to systemctl command: <https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units>

1. Find out how to boot into emergency mode for both your servers. Write a one page (or less) document on how to do that. Include 1 paragraph executive summary on why you might want to.

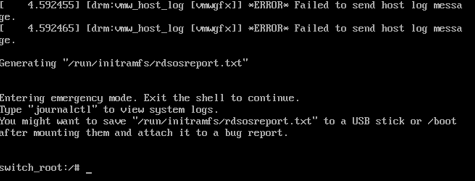
Emergency mode is like safe mode for a windows machine or another operating system but for our servers. We may use this if there is an issue with hardware or possibly forgetting our admin password for the server. We can also run repairs with emergency mode. You can only do this from initial boot of your server.

CentOS

1. Once you boot your server, press the “e” key to place the Virtual machine into edit mode which then display this screen:



1. On the fifth line you will see a string after resume. You want your string to look like the image above by adding “**rd.break**” at the end.
2. From here we can boot up our CentOS by following steps at the bottom that says **ctrl** and **x**
3. This will boot out CentOS server into emergency mode. It will look like the image below:



1. Once we have completed what we needed to complete in emergency mode we will reboot using this command:

**reboot –f**

Ubuntu

1. Once you boot the Ubuntu server you will press and hold down the **shift** key. This will bring up the following screen:



1. Following what it says on the bottom we will press the “**e”** key and enter edit mode like we did in CentOS.
2. Move down the menu and find the line ending in “**maybe-ubiquity**”
3. From here add **systemd.unit=emergency.target**



1. We will conclude by following the key press displayed in the bottom paragraph by pressing **f10**
2. To enter emergency mode on Ubuntu we need to enter in the root password or press **ctrl and d** to continue through the prompt
3. Once we have completed what is needed in emergency mode, we will issue the following command to exit out of emergency mode:

**systemctl default**