Security Part 1 Ubuntu

Github : <https://github.com/trevortrites/Linux_Admin_Portfolio>

Importing and running these scripts on CentOS and Ubunutu is very similar except a few commands are a bit different.

Install inotify-tools

# sudo apt-get install inotify-tools

1. Write a Script to detect ip addresses trying to gain access, examples of things to pay attention to include all use between midnight and 6, all logins for a specific user, anything else you consider behavior that should send up a red flag.

Ensure the “**detect\_ip**” script is in **/root/class/linux/** directory. In addition to setting the permissions of the script to 755 so it has permission to run. To do this you will run:

**“chmod 755 detect\_ip”**

*SCRIPT IS LOCATED IN GITHUB BUT THIS IS THE SCRIPT HERE:*

#!/bin/sh

date >> /root/class/linux/network\_scan.txt

# This adds the date to the text file so we know when the output was created

nmap-sT -0 192.169.0.0/24 >> /root/class/linux/network\_scan.txt

# this redirects the detected IP addresses on my network to a txt file

Concerning behavior may include logins after normal hours, continuous login attempts, and abnormal ping requests.

1. Write script to detect changes to a specific directory.  Such as changes to /var/log or /etc/ think about using a diff here, or a hash.

Add the script **“dir\_change”** to your server. This is in the **/root/class/linux/** directory. Ensure permissions of the script are set to 755 so it has full permission to run. Use this command: **“chmod 755 dir\_change”**

*SCRIPT IS LOCATED IN GITHUB BUT THIS IS THE SCRIPT HERE:*

#!/bin/sh

date >> /root/class/linux /monitoring\_log.txt

# Append log file with date of collection.

sha1sum /etc/\* | sha1sum >> /root/class/linux/monitoring\_log.txt

inotifywait -m /etc >> /root/class/linux/monitoring\_log.txt

# This generates a output of changes in /etc/directory/, then rediects to a text file

I believe that the /etc/directory/ is the most crucial to keep an eye on because it includes the main structure of our server with main root file and passwords.

1. Monitor hidden files, root executables, and see if changes are made, who made them, and when they were changed.

Add script **“monitor\_script”** which is in the same directory as the script above in **/root/class/linux/** directory. Using the same permissions as well. Running this command:

**“chmod 755 monitor\_script”**

*SCRIPT IS LOCATED IN GITHUB BUT THIS IS THE SCRIPT HERE:*

#!/bin/sh

date >> /root/class/linux/monitor\_log.txt

# Append log file with date of collection.

stat /root/.?\* >> /root/class/linux/monitor\_log.txt

# List the statistics on the hidden files such as last modification date etc..

ls -ld /root/.?\* >> /root/class/linux/monitor\_log.txt

# List only the hidden files in root.

Monitoring hidden files and root executables is very important because if we have suspicious data on our system such as root kits, they will be detected with this script which we can then begin the process of removal. This can destroy our systems completely and even lock them up, or ourselves out.

From here we need to schedule these scripts to execute as cronjobs. This command will open the configuration file. We will schedule jobs with a select time and the script we want to use.

To open for editing use this command:

**# crontab –e**

Edit your crontab file and exit the file by committing the changes. We will do this by pressing ESC twice, then type “ **wq! “**. This writes, quits, and saves the file. If you want to make sure your edit worked, you can use this command:

**# crontab -l**

It should list the directories and script names for you output such as this:

./root/class/linux/detect\_ip

./root/class/linux/dir\_change

./root/class/linux/monitor\_script

A small paragraph or two explaining if there anything else you think you should have a script and/or cron job for relating to security?  why or why not?

I believe we should be running the script that we created for the security, hardening, and compliance. Running the script regularly will benefit us in many ways, because we can monitor our hardware and network health. Not just file structure, and IP traffic. One give away if our server is being attacked or something wrong is how much usage we may be using on our hardware. If it is higher than normal something may be wrong.