Hunting For Hackers with rkhunter

How simple tools can be used to scan your linux environment

Presenters:
Isabel Rodriguez
Joe Starr

What is rkhunter?

Rkhunter is a security Unix-based tool that scans for backdoors, rootkits, and possible local exploits. It helps security professionals and administrators to identify vulnerabilities and changes to the system that could jeopardize system security.
Rkhunter is open source software.

A “rootkit” is a malicious software designed to quietly gain access to your system as the root user.

Source: Xpert Assistant,
https://hackertarget.com/rkhunter-add-another-layer-to-your-security/

Rkhunter Features

- Open source software (`sudo apt install rkhunter`)
- Targeted to search for and find rootkits
- Is compatible with other antivirus programs
- Can search for hidden files and folders
- Checks default files that rootkits use
- Can be automated to run on it's own (we will demonstrate this later in the presentation)

Important:
rkhunter is not an antivirus software nor is it intended to replace any antivirus/antimalware programs. It only scans for the presence of rootkits and other potential vulnerabilities on your system.

Source: https://en.opensuse.org/Rootkit_Hunter#About_Rootkit_Hunter_(rkhunter)

Code Breakdown

```
#!/bin/bash 1

# Update rkhunter data 2
sudo rkhunter --update

# Run rkhunter scan 3
echo "rkhunter initiated... running scan -- please wait..."
if yes | sudo rkhunter --check | tee /home/sysadmin/Project4/rkhunter_scan_results.log && cat /var/log/rkhunter.log; then 4
  echo "rkhunter scan completed successfully. Check /home/sysadmin/Project4/rkhunter_scan_results.log for details." 5
else
  echo "rkhunter scan encountered an error. Check /home/sysadmin/Project4/rkhunter_scan_results.log for details." 6
  exit 1
fi
```

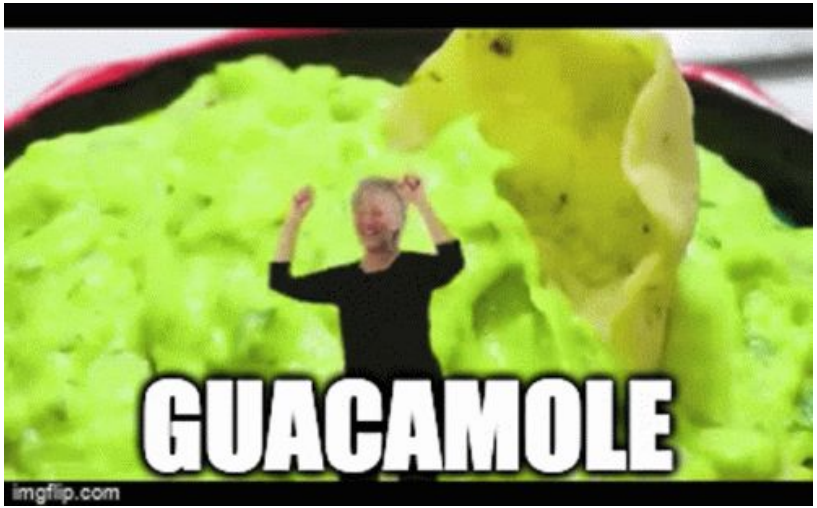
1. Indicates the script that should be executed.
2. The command updates rkhunter with the latest version. Running sudo ensures it has permissions to perform the update.
3. The echo command Informs the user that the rkhunter scan is starting.
4. This command runs the actual scan for vulnerabilities. The “if yes” command automatically selects yes for any prompts that occur during the scan. The tee command saves the output to our Project 4 file and displays it to the terminal.
5. If the scan completes without any errors, this message will indicate success and directs the user to the log file for details. The log file is also concatenated to display in the terminal after completion.
6. If there is an error during the scan, a message will display to inform the user and the script exits with a status of 1, indicating an error occurred.

Demonstration Breakdown

1. The code breakdown essentially walked us through what is to be expected…
2. Running the scan may take some time, depending on the size of the system.
3. After it runs, the results will display on the terminal or can be viewed in the log file. We wanted convenience for the user by displaying the results instantly instead of having to search for them.
4. The scan will notify us if it has completed successfully!

Demonstration

Proceed to Apache Guacamole



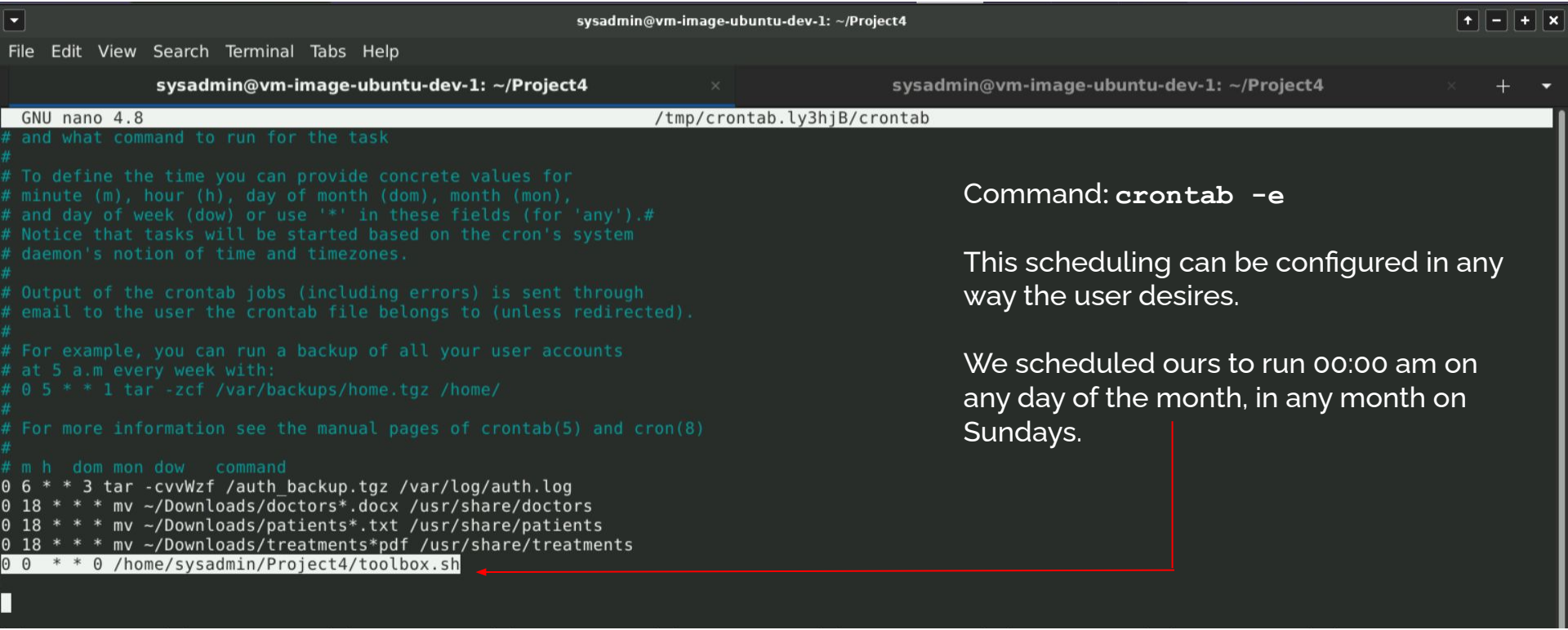
Summary - Crontab Scheduling

Many cybersecurity professionals can automate some of their tasks to achieve efficiency throughout their work days.

We wanted our script to be effective and be able to execute automatically in the background without having to manually execute our script each time.

We were able to automate this script by scheduling it in a crontab!

Summary - Crontab Scheduling



Command: **`crontab -e`**

This scheduling can be configured in any way the user desires.

We scheduled ours to run 00:00 am on any day of the month, in any month on Sundays.

Q & A

Thank you for your time!

Any questions?