Backup, Restore, and Post Mortem

Adrianna Holden-Gouveia
Website: aholdengouveia.name
: aholdengouveia

Accessibility Notice

This document is also available in HTML format at:

Link: `https://aholdengouveia.name/LinuxAdmin/labexcercises/backups.html`

The HTML version provides enhanced accessibility features including keyboard navigation, screen reader support, responsive design, dark mode support, and high contrast options.

Ticket

Ticket: ACME-4518
Requested by: Management
Systems: Both report servers

Background

After last week's incident, management wants proof that Acme Corp can recover. A backup no one has tested isn't a backup. You will be working on the two servers you set up in the Server Setup lab: your Red Hat based server and your second server.

Learning Outcomes

By the end of this lab, you will be able to:

- Write a backup script that keeps a set number of backups and copies them to another server
- Schedule backups with cron
- Restore files from a backup and verify that they match the originals
- Write a post mortem for an incident

Requirements

1. Write a backup script using tar that backs up /etc, users' home directories, and your change log, compressed, with the date in the filename.
2. Keep only the last 7 backups, deleting older ones automatically.
3. Copy each backup to your other server, so it isn't stored only on the machine it came from.
4. Schedule your backup script with cron.
5. Restore drill: delete something important, such as a config file or a user's home directory, then restore it from your backup. Time how long it takes.
6. Write a post mortem for either your restore drill or the B0rk incident from the Scripting for Server Security Part 2 lab. Use the sample report from Google (linked under Resources) as your template.

Acceptance Criteria

- **Everything must be cited.** Every command, statement, and line of code in your scripts and documents must be cited, either with a link to a reference or with a note naming the previous class where you learned it, for example: `# Learned in CIS 117 Intro to Linux`. In scripts, put the citation in a comment next to the code. Anything without a citation will not count.
- The restore is proven with before and after screenshots.
- The restored files are checked against the originals, for example with `sha256sum`.
- Your cron schedule is shown working.
- Scripts are well commented and uploaded to GitHub.

- Every terminal screenshot shows your prompt with your username and hostname, and either the output of `date` or a note beside the terminal with your name and the term. Hostnames must match your server inventory from the Server Setup lab.
- Your GitHub repository shows your work over time: several commits across more than one day, each with a message describing the change, not a single upload of finished code.

Deliverables

Audience: a teammate who has to restore a server during an emergency. Assume technical knowledge but not expertise. Screenshots are helpful to go with your descriptions.

Make sure your deliverables include documents for BOTH servers, clearly labeled with what the document is and which server it's for.

- Your backup script, with a link to your GitHub where you've uploaded it.
- A restore procedure someone else could follow during an emergency, including how long your restore drill took.
- Your post mortem.

Resources

References, a video, a PowerPoint and some notes are available at my website Link: `https://www.aholdengouveia.name/LinuxAdmin/backups.html`

My video on how to use tar Link: `https://www.youtube.com/watch?v=x0EK6PmKRCU`

Sample post mortem report from Google Link: `https://web.archive.org/web/20220724194136/https://developers.googleblog.com/2013/05/google-api-infrastructure-outage_3.html`