Update #4 - Automated Backups for BookStack on VPS - Secure & Scheduled

Update #4

As part of hardening and maintaining my public BookStack instance hosted on a VPS, I implemented a secure, automated backup system using shell scripting and cron. This ensures my database and uploaded files are regularly archived and protected against data loss - without exposing sensitive information in scripts.

The Setup

- **VPS**: Ubuntu 22.04 LTS (Hostinger)
- **Web Stack**: Apache, MySQL, PHP 8.2 (LAMP stack)
- **Application**: BookStack (latest stable)
- **Backup Destination**: `/opt/bookstack_backups/` on the VPS
- **Security Enhancements**:
  - `fail2ban` active
  - SSH hardening complete
  - UFW configured to allow only necessary services

The Process

1. **Created a Shell Script**
   - A custom script (`/usr/local/bin/bookstack-backup.sh`) was written to:
     - Dump the MySQL `bookstack` database

    - Archive important BookStack directories (such as `/var/www/bookstack/public/uploads` and `/var/www/bookstack/storage/uploads`)
    - Rotate old backups by removing any older than 7 days
2. **Secured MySQL Credentials**
    - Rather than placing the database password in the script (which is insecure), credentials were stored securely in the root user's `~/.my.cnf` file:

```
[client]
user=bookstack
password=your_db_password
```

3. **Tested the Script**
    - Verified manual execution of the script:
        - Confirmed `.sql` dumps and `.tar.gz` files were created correctly
        - Ensured proper permissions and ownership of backup files
4. **Scheduled Daily Cron Job**
    - Added the script to root's crontab to run automatically at **2:00 AM** each day:

```
0 2 * * * /usr/local/bin/bookstack-backup.sh
```

Sample Output

Backup files generated daily look like:

- `/opt/bookstack_backups/bookstack_2025-05-09.sql`
- `/opt/bookstack_backups/bookstack_files_2025-05-09.tar.gz`

What I Learned

- Avoid using `mysqldump` passwords in scripts — `~/.my.cnf` is a secure alternative.
- Giving the database user `PROCESS` privilege was necessary to prevent dump errors.
- Always verify permissions and script execution manually before automating via cron.
- Storing backups under `/opt/` keeps them cleanly separated from web content.

Revision #2
Created 9 May 2025 21:21:27 by Nate Nash
Updated 7 June 2025 00:14:24 by Nate Nash