Wesley Marshall

Cloud Engineering Assignment: Building a Hardened AWS AMI with Packer

09/29/2025 - 10/18/2025

The objective of this project was to create a custom Amazon Linux 2023 AMI using HashiCorp Packer that implements CIS benchmark security controls and includes a configured Apache HTTP server. In the process of completing this project, a hierarchy file structure will be created to contain the packer template and scripts used to implement CIS benchmarks and configure Apache HTTP server. This packer template will be validated and an AWS EC2 instance created using the packer AMI. A Git repository will be created to track the work. The

In this project I used these

1. Filesystem Configuration: 1.1.1.1 to prevent execution and device abuse in temp space.
2. SSH Hardening: 5.28 to block direct root login over SSH
3. User Account Management: 5.4.1.1 to enforce password rotation and aging policies
4. System Auditing: 4:1.1.1 to enable logging of sensitive file access
5. Network Security: 3.3.1 to reduce attack surface by disabling unused protocol

I faced several problems, of which one was an error when trying to install packer where there were duplicate files. I used co-pilot to help troubleshoot. I faced another issue when trying to login to GitHub from VS CLI. I discovered that not able to use a password. In this case I had to make a keygen. Numerous issues were encountered when trying to run packer build, all of which were related to syntax, coding, and format of the scripts. I had to make many changes to the packer template, CIS, and HTTPD codes. I used to co-pilot to help navigate.

For testing and verification of my implementation, I used Automated and Manual validation. Automated validation was performed using packer validate and packer build. Verified CIS controls via `auditd`, `sshd_config`, and `httpd.conf` inspection. Manual validation was performed by SSH into hardened AMI and confirmed by no root login, Apache served branded page only, disabled filesystems not mounted. Used `curl`, `ss`, and `systemctl` to confirm service states and port bindings.

As a result of this project, I now have a better aptitude and proficiency in Linux, Packer, EC-2 and the components of Infastructure as well as an understanding of AWS EC-2

generation, LVM’s, and AMI’s. I understand the components of a packer template including CIS bencmarks and HTTPD Apache servers. I now have a better working knowledge of GitHub and how to add, commit and push from my local directory and how to document with README and screenshots.