Cloud Engineering Assignment: Building a Hardened AWS AMI with Packer

Objective
Create a custom Amazon Linux 2023 AMI using HashiCorp Packer that implements CIS benchmark security controls and includes a configured Apache HTTP server.

Learning Outcomes
- Understand Infrastructure as Code principles using Packer
- Implement security hardening based on CIS benchmarks
- Automate AMI creation in AWS
- Practice bash scripting for system configuration

Prerequisites
- AWS Account with appropriate IAM permissions
- Packer installed locally (version 1.9+)
- AWS CLI configured with credentials
- Basic understanding of Linux and bash scripting

Assignment Tasks

Part 1: Project Setup (10 points)
1. Create a new directory for your Packer project
2. Initialize a Git repository to track your work
3. Create the following file structure:
   ```
   packer-ami-project/
   ├── template.pkr.hcl
   ├── scripts/
   │   ├── cis-hardening.sh
   │   └── httpd-setup.sh
   └── README.md
   ```

Part 2: Packer Template Configuration (25 points)
Create a `template.pkr.hcl` file with the following requirements:

1. **Source Configuration**
   - Use Amazon Linux 2023 as the base AMI
   - Instance type: t3.micro
   - SSH username: ec2-user
   - Region: us-east-2 (or your preferred region)

2. **Build Configuration**
   - Add appropriate tags to identify your AMI
   - Configure temporary security group for SSH access
   - Set AMI name with timestamp

3. **Provisioners**
   - Shell provisioner to run CIS hardening script
   - Shell provisioner to install and configure httpd

Part 3: CIS Benchmark Implementation (40 points)
Create `scripts/cis-hardening.sh` implementing these CIS Level 1 benchmarks:

Required Security Controls (implement at least 5):

1. **Filesystem Configuration**
   - Ensure /tmp is configured with nodev, nosuid, noexec options
   - Set permissions on /etc/passwd, /etc/shadow, /etc/group

2. **SSH Hardening**
   - Disable root login (PermitRootLogin no)
   - Set SSH Protocol to 2
   - Disable empty passwords (PermitEmptyPasswords no)
   - Set ClientAliveInterval and ClientAliveCountMax

3. **User Account Management**
   - Set password expiration policies (PASS_MAX_DAYS, PASS_MIN_DAYS)
   - Configure password complexity requirements

4. **System Auditing**
   - Enable and configure auditd service
   - Add basic audit rules for sensitive file monitoring

5. **Network Security**
   - Disable IPv6 if not needed
   - Enable TCP SYN cookies
   - Disable ICMP redirects

Your script should:
- Include comments explaining each security control
- Log all changes to `/var/log/cis-hardening.log`
- Be idempotent (can run multiple times safely)
- Check for errors and exit with appropriate codes

Part 4: HTTP Server Setup (15 points)
Create `scripts/httpd-setup.sh` that:

1. Installs Apache HTTP server (httpd)
2. Creates a custom index.html with:
   - Your name/team name
   - AMI creation date

   - List of CIS benchmarks implemented
3. Configures httpd to start on boot
4. Ensures httpd service is running
5. Configures basic security settings (ServerTokens, ServerSignature)

Part 5: Documentation (10 points)
Create a comprehensive README.md including:

1. **Project Overview**
   - Brief description of the project
   - List of CIS benchmarks implemented

2. **Prerequisites**
   - Required tools and versions
   - AWS permissions needed

3. **Usage Instructions**
   - How to validate the Packer template
   - How to build the AMI
   - How to test the resulting AMI

4. **Verification Steps**
   - How to verify CIS benchmarks were applied
   - How to access the web server
   - Security group requirements for testing

5. **Cleanup Instructions**
   - How to deregister the AMI
   - How to remove snapshots

Deliverables

Submit the following:
1. Complete Packer template (`template.pkr.hcl`)
2. All bash scripts in the `scripts/` directory
3. README.md with complete documentation
4. Screenshot or output showing successful AMI creation
5. Screenshot showing the httpd server responding with your custom page
6. Brief report (1-2 pages) describing:
   - Which CIS benchmarks you implemented and why
   - Challenges faced and how you overcame them
   - How you tested/verified your implementation

Evaluation Criteria

Criteria	Points

Packer template correctly configured	25
CIS benchmarks properly implemented	40
HTTP server setup and configuration	15
Documentation quality	10
Code quality and best practices	10
Total	**100**

Helpful Resources

- [Packer Documentation](https://developer.hashicorp.com/packer/docs)
- [CIS Amazon Linux Benchmark](https://www.cisecurity.org/benchmark/amazon_linux)
- [AWS EC2 AMI Documentation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html)
- [Packer AWS Builder](https://developer.hashicorp.com/packer/plugins/builders/amazon/ebs)

Testing Your AMI

After building:
1. Launch an EC2 instance from your custom AMI
2. SSH into the instance and verify CIS controls:
   ```bash
   # Check SSH configuration
   sudo grep -E "PermitRootLogin|Protocol|PermitEmptyPasswords" /etc/ssh/sshd_config

   # Check password policies
   grep -E "PASS_MAX_DAYS|PASS_MIN_DAYS" /etc/login.defs

   # Verify auditd is running
   sudo systemctl status auditd
   ```
3. Test the web server:
   ```bash
   curl http://<instance-public-ip>
   ```

Submission Deadline
[Instructor to specify]

Academic Integrity
This is an individual assignment. While you may discuss concepts with classmates, all code and documentation must be your own work.

Note: Remember to terminate any test instances and deregister AMIs after completing the assignment to avoid unnecessary AWS charges.

*****I have provided the bash script for CIS hardening for you so you do not have to go and find them. Please figure out how to implement this script into your packer code so that the image is hardened according to CIS Standards******

```
#!/bin/bash

################################################
# CIS Benchmark Hardening Script
# For Amazon Linux 2023
################################################

set -e

LOG_FILE="/var/log/cis-hardening.log"

# Function to log messages
log_message() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | sudo tee -a "$LOG_FILE"
}

log_message "========================================="
log_message "Starting CIS Hardening Process"
log_message "========================================="

################################################
# CIS 1.1.2 - Configure /tmp partition
################################################
log_message "CIS 1.1.2: Configuring /tmp with security options"

# Create systemd mount unit for /tmp
sudo bash -c 'cat > /etc/systemd/system/tmp.mount <<EOF
[Unit]
Description=Temporary Directory /tmp
ConditionPathIsSymbolicLink=!/tmp
DefaultDependencies=no
Conflicts=umount.target
Before=local-fs.target umount.target
```

```
[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,strictatime,noexec,nodev,nosuid

[Install]
WantedBy=local-fs.target
EOF'

sudo systemctl daemon-reload
sudo systemctl enable tmp.mount

log_message "✓ /tmp configured with noexec,nodev,nosuid"

################################################
# CIS 1.3.1 - Ensure AIDE is installed
################################################
log_message "CIS 1.3.1: Installing and configuring AIDE"

sudo dnf install -y aide
sudo aide --init
sudo mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz

log_message "✓ AIDE installed and initialized"

################################################
# CIS 5.2 - Configure SSH Server
################################################
log_message "CIS 5.2: Hardening SSH configuration"

# Backup original sshd_config
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup

# Apply SSH hardening settings
sudo bash -c 'cat >> /etc/ssh/sshd_config.d/99-cis-hardening.conf <<EOF
# CIS Benchmark SSH Hardening

# Disable root login
PermitRootLogin no

# Disable empty passwords
PermitEmptyPasswords no

# Set SSH protocol to 2 (default in modern SSH)
Protocol 2
```

```
# Enable strict mode
StrictModes yes

# Disable X11 forwarding
X11Forwarding no

# Set client alive interval (5 minutes)
ClientAliveInterval 300
ClientAliveCountMax 2

# Limit authentication attempts
MaxAuthTries 4

# Disable host-based authentication
HostbasedAuthentication no

# Disable password authentication (uncomment to enforce key-only)
# PasswordAuthentication no

# Log level
LogLevel INFO

# Use PAM
UsePAM yes
EOF'

log_message "✓ SSH hardened according to CIS benchmarks"

################################################
# CIS 5.4.1 - Set Password Expiration
################################################
log_message "CIS 5.4.1: Configuring password policies"

# Backup login.defs
sudo cp /etc/login.defs /etc/login.defs.backup

# Configure password aging policies
sudo sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS   90/' /etc/login.defs
sudo sed -i 's/^PASS_MIN_DAYS.*/PASS_MIN_DAYS   7/' /etc/login.defs
sudo sed -i 's/^PASS_MIN_LEN.*/PASS_MIN_LEN    14/' /etc/login.defs
sudo sed -i 's/^PASS_WARN_AGE.*/PASS_WARN_AGE   14/' /etc/login.defs

log_message "✓ Password expiration policies configured"

################################################
# CIS 5.4.4 - Ensure strong password policy
```

```
##############################################
log_message "CIS 5.4.4: Configuring password complexity requirements"

# Install PAM password quality module
sudo dnf install -y libpwquality

# Configure password quality requirements
sudo bash -c 'cat > /etc/security/pwquality.conf <<EOF
# Password Quality Requirements - CIS Benchmark
minlen = 14
dcredit = -1
ucredit = -1
ocredit = -1
lcredit = -1
EOF'

log_message "✓ Password complexity requirements configured"

##############################################
# CIS 4.1.1 - Configure auditd
##############################################
log_message "CIS 4.1.1: Configuring system auditing"

# Enable and start auditd
sudo systemctl enable auditd
sudo systemctl start auditd

# Add audit rules for sensitive files
sudo bash -c 'cat >> /etc/audit/rules.d/cis.rules <<EOF
# CIS Benchmark Audit Rules

# Monitor changes to system date and time
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time-change
-a always,exit -F arch=b64 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change

# Monitor user/group information
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity

# Monitor system network configuration
-w /etc/sysconfig/network -p wa -k system-locale
-w /etc/hosts -p wa -k system-locale

# Monitor changes to system mandatory access controls
```

```
-w /etc/selinux/ -p wa -k MAC-policy

# Monitor login and logout events
-w /var/log/lastlog -p wa -k logins
-w /var/run/faillock/ -p wa -k logins

# Monitor session initiation
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins

# Monitor changes to sudoers
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d/ -p wa -k scope
EOF'

# Reload audit rules
sudo augenrules --load

log_message "✓ Audit rules configured for sensitive file monitoring"

################################################
# CIS 3.2 - Network Parameters
################################################
log_message "CIS 3.2: Configuring network security parameters"

sudo bash -c 'cat >> /etc/sysctl.d/99-cis.conf <<EOF
# CIS Benchmark Network Security Settings

# Enable TCP SYN cookies
net.ipv4.tcp_syncookies = 1

# Disable IP forwarding
net.ipv4.ip_forward = 0
net.ipv6.conf.all.forwarding = 0

# Disable send packet redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Disable ICMP redirects
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0

# Disable secure ICMP redirects
```

```
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0

# Log suspicious packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Ignore ICMP ping requests
net.ipv4.icmp_echo_ignore_all = 0

# Ignore broadcast pings
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Enable reverse path filtering
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Disable IPv6 if not needed
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
EOF'

# Apply sysctl settings
sudo sysctl -p /etc/sysctl.d/99-cis.conf

log_message "✓ Network security parameters configured"

################################################
# CIS 1.4.1 - Set file permissions
################################################
log_message "CIS 1.4.1: Setting secure file permissions"

# Set permissions on critical system files
sudo chmod 644 /etc/passwd
sudo chmod 000 /etc/shadow
sudo chmod 644 /etc/group
sudo chmod 000 /etc/gshadow
sudo chmod 600 /etc/ssh/sshd_config

log_message "✓ Secure file permissions set on critical files"

################################################
# Additional Security Configurations
################################################
log_message "Applying additional security configurations"

# Disable unnecessary services
```

```
sudo systemctl disable debug-shell.service 2>/dev/null || true

# Set banner for SSH
sudo bash -c 'cat > /etc/issue.net <<EOF
************************************************************************
*
                            NOTICE TO USERS

This computer system is for authorized use only. Users (authorized or
unauthorized) have no explicit or implicit expectation of privacy.

Any or all uses of this system and all files on this system may be
intercepted, monitored, recorded, copied, audited, inspected, and
disclosed to authorized site, law enforcement personnel, as well as
authorized officials of other agencies, both domestic and foreign.
************************************************************************
*
EOF'

log_message "✓ Login banner configured"

##############################################
# Summary
##############################################
log_message "========================================="
log_message "CIS Hardening Completed Successfully"
log_message "========================================="
log_message "Implemented Controls:"
log_message "- /tmp partition hardening (noexec, nodev, nosuid)"
log_message "- AIDE file integrity monitoring"
log_message "- SSH server hardening"
log_message "- Password expiration policies"
log_message "- Password complexity requirements"
log_message "- System audit logging (auditd)"
log_message "- Network security parameters"
log_message "- Secure file permissions"
log_message "========================================="

echo "CIS hardening completed. Check $LOG_FILE for details."
```