Andree Salvo

Southern New Hampshire University

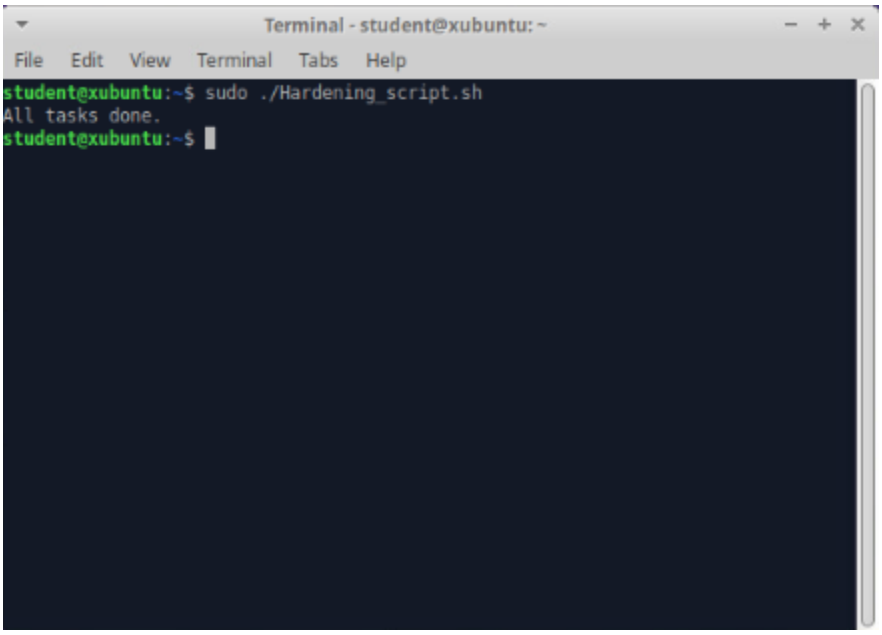
CYB 300 – 14668

Instructor: Jason Keltner

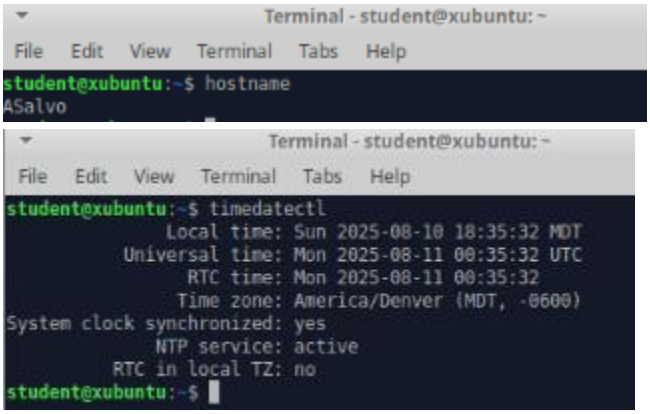
7-1 Project

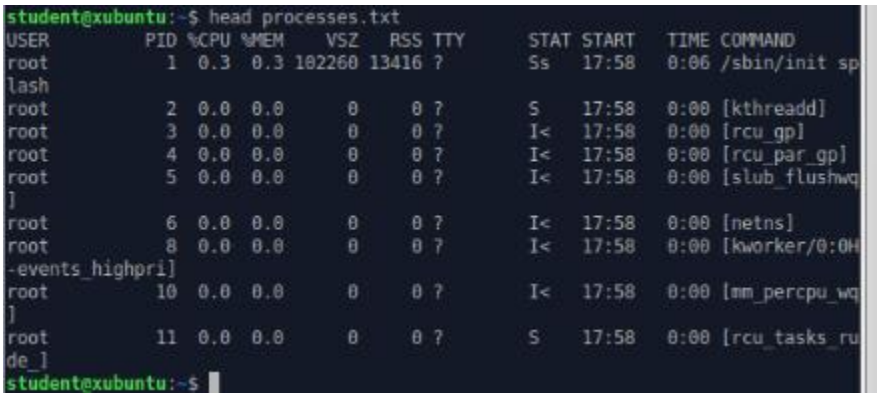
1. **Automated Hardening Scripts:** Compose a single executable script to automate hardening tasks to meet the requirements in the scenario.

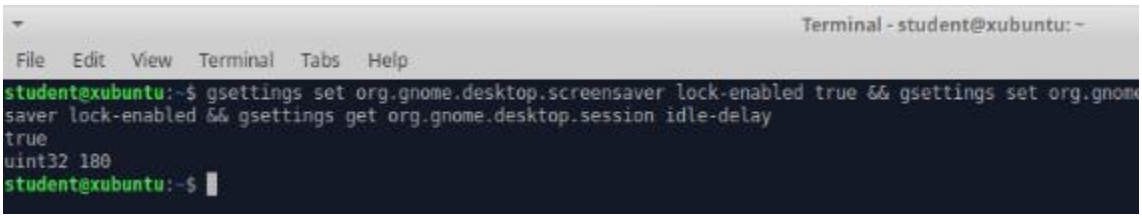
   a. Screenshot of a single **executable script** in the Linux shell environment

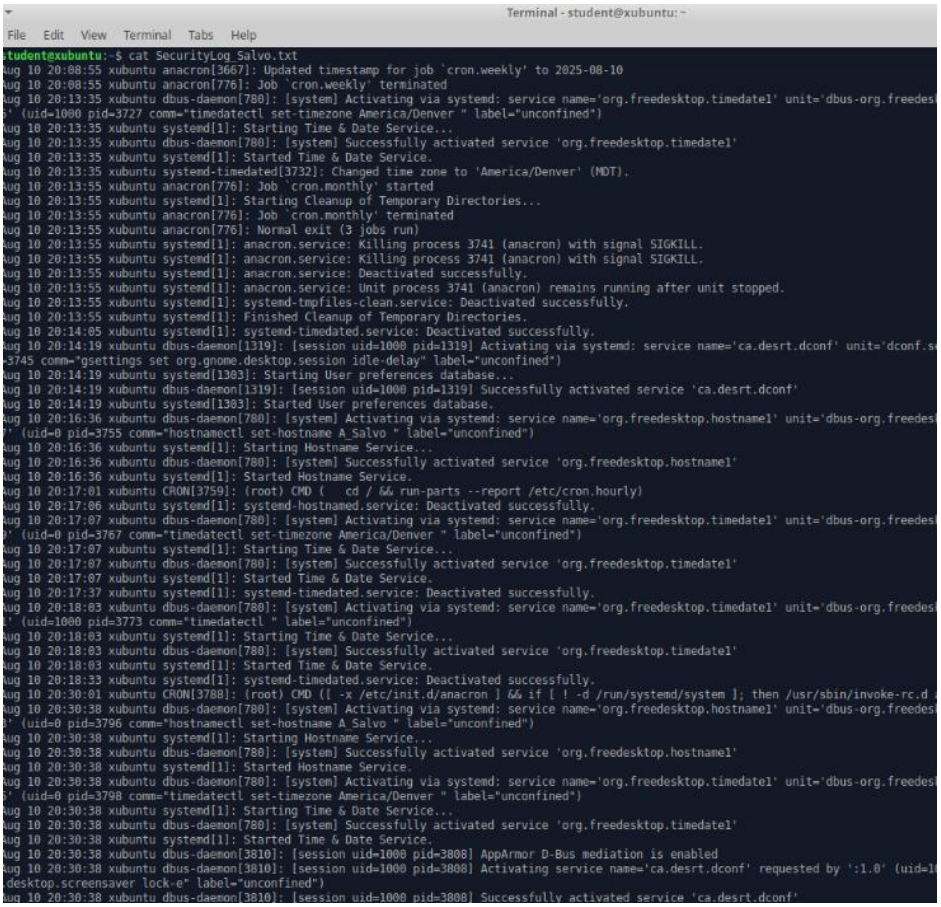


   b. Screenshots that **evidence** each requirement has been met







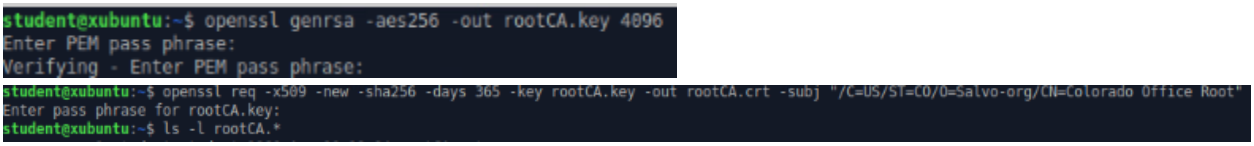


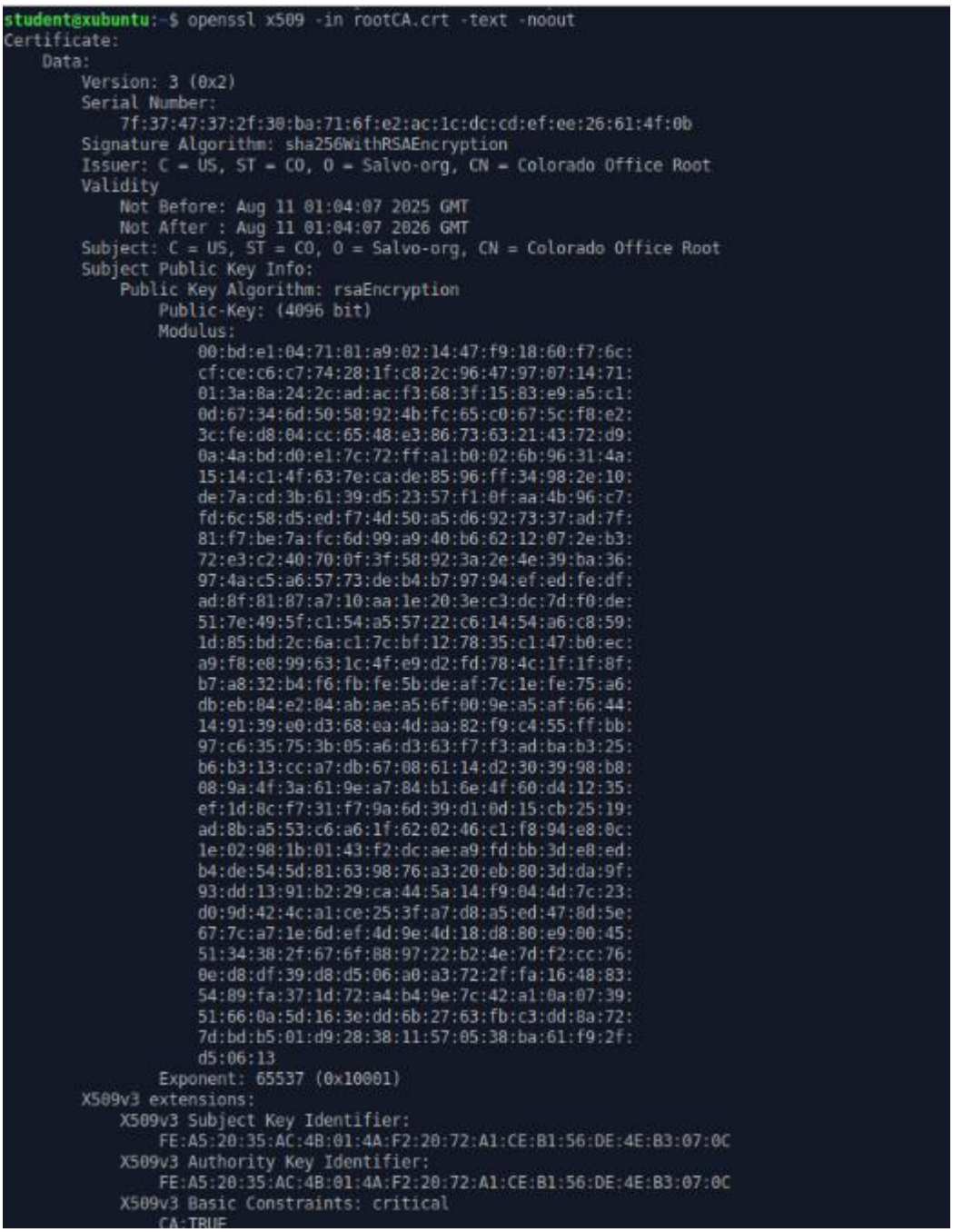
c. Aside from saving time, automated scripts make it a lot easier to keep things consistent and avoid mistakes that can happen when doing configurations by hand. This will let you

roll out the same secure settings to every system without having to repeat the work, and you can reuse them whenever you need to make updates. Also, having a script means you've got a clear record of what changes were made, which helps with audits and makes fixing issues down the road a lot simpler and easier.

2. Certificate Authority

a. Provide a screenshot of the OpenSSL commands to create a CA with settings that meet the organizational requirements





b. To create a Certificate Signing Request (CSR) for a server or workstation in the new location, the system administrator first generates a private key using openssl genrsa -out server.key 2048, which securely stores the cryptographic key on the machine. Next, they run openssl req -new -sha256 -key server.key -out server.csr -subj "/C=US/ST=CO/O=Salvo-Org/CN=server1.colorado.office.local" to generate the CSR,

embedding the organization and host details. The resulting .csr file is then securely transmitted to the Colorado Office Root CA administrator, who verifies the request against organizational policies, signs it with the CA's private key, and returns the issued certificate (server.crt). Finally, the server or workstation installs both the signed certificate and the CA's root certificate to establish a trusted, encrypted communication channel.

c. Using Public Key Infrastructure (PKI) ties into the security principles of **defense in depth** and **complete mediation**. First, Defense in depth comes from adding certificates and encryption as another layer on top of things like usernames and passwords, so there's more than one barrier keeping systems safe. Second, Complete mediation means every single connection is checked and validated through certificate verification, so nothing slips through without being reviewed. For the CIA triad, PKI helps with **confidentiality** by encrypting traffic, **integrity** by making sure data hasn't been tampered with through digital signatures, and **availability** by keeping connections secure and reliable so attacks don't take systems offline.

3. Hardening Systems

a. Discuss how to make the transition from industry guidelines to a baseline that is appropriate for your organization

Transitioning from industry guidelines to an organizational baseline starts with picking a trusted standard, like the NIST framework or CIS Benchmarks, as your starting point. You go through each control and figure out what applies best to your environment, removing anything that doesn't make sense for your systems and adding any extra

settings that address unique risks your organization faces. The goal here is to take the broad “baseline starting points” recommendations and customize them so they fit your actual network, applications, and workflows. Once these are tailored, the baseline is set, and you apply it consistently across all systems and keep it updated as threats, technology, or business needs change.

b. Create an operating system security-configuration checklist representing the elements used in Part I: Automated Hardening Scripts

I. Rename Computer > First Initial_Last Name (hostname status)

II. Change Time Zone > America/Denver (timedatectl)

III. List Running Processes > processes.txt (ls -l processes.txt and head processes.txt)

IV. Set idle lock time for screensaver to 3 min > (gsettings get org.gnome.desktop.session idle-delay and gsettings get org.gnome.desktop.screensaver lock-enabled)

V. Last 50 syslog entries > SecurityLog_Salvo.txt (wc -l SecurityLog_Salvo.txt and tail SecurityLog_Salvo.txt)

c. Why checklists are important

Security configuration checklists make it easier to keep systems consistent and reduce the chances of missing important steps. They turn policies into clear, repeatable actions that anyone on the team can follow, which is especially useful when multiple people are responsible for hardening systems. Checklists also make audits and troubleshooting simpler, because you have a record of what’s been applied and verified, and they help maintain compliance with internal standards or external regulations.