Assignment 4 - Option One

For this assignment, we were assigned to set up a two-tier certificate authority architecture. This design introduced PKI services onto the network, which allowed for secure, certificate-based authentication, encryption, and communication between devices. The project's objective is to create a Root CA, an Intermediate Signing CA, and to deploy trusted certificates to both Windows and pfSense systems via Group Policy. A two-tier CA model offers better security by keeping the Root CA out of daily network usage, while intermediate CAs are allowed to issue operational certificates. It also emulates how large companies do PKI deployments, ensuring that the chain of trust of the certificates remains safe and manageable.

The lab environment maintained the same IP structure as the previous Active Directory deployment. The domain controller, AD-SERVER remained at 192.168.50.100 within the Windows LAN subnet of 192.168.50.100–192.168.50.200. The Linux system operated within the 192.168.100.0/24 subnet, with the pfSense firewall bridging between the two networks. Instead of deploying a new server, the Microsoft CA service was enabled on the existing domain controller set up in the previous lab. This is because this way it will allow easier AD management and distribution through Group Policy Objects. Following security best practices though, the Root CA portion was isolated and set to stay offline once an intermediate CA had been established.

Root Certificate Authority Configuration

The Root Certificate Authority (Root CA) serves as the highest trust anchor within the PKI hierarchy. It signs only subordinate or intermediate certificates and should never directly issue certificates to end users or devices. The Root CA was created as a standalone CA to maximize security and minimize exposure. Within the Windows Server Manager, the “Active Directory Certificate Services” role was installed, and the “Certification Authority” role service was selected. When prompted for setup type, “Standalone Root CA” was chosen, and a new private key was generated using RSA 2048-bit encryption. The Root CA was assigned the common name “LAB-ROOT-CA” and configured to use the SHA256 hashing algorithm with a 10-year validity period.

After installation, the self-signed certificate of the Root CA was exported and stored in a secure location on the host system. The resulting certificate was then copied onto removable storage, making it possible to transfer it later to the intermediate CA for chain establishment. Since the Root CA should never be exposed, after the initial setup, the Certification Authority service was stopped, and the system was disconnected from the network. All these steps have been followed according to the best practices for the enterprise deployment of Root CAs in offline and high-security environments.

Intermediate Signing CA Configuration for pfSense and Windows

With the Root CA configured and secured, the next step was to create an Intermediate Signing CA responsible for the issuance of all operational certificates within the domain. This CA was to be set up on the domain controller using the same Certificate Service role but installed as an enterprise CA. During the installation process, the system prompted me for a certificate request file, which was made and then signed by the root CA that is offline using the CA console. After signing, the new certificate was imported into the intermediate CA, which officially established the two-tier trust hierarchy.

Next, the “LAB-INT-CA” Intermediate CA was used to issue certificates for both the internal Windows systems and pfSense firewall. For pfSense, a new Server Certificate Template was created within the Certification Authority console that allowed the web management interface to be accessed using HTTPS with a trusted certificate. Using the pfSense web GUI, a CSR was generated and downloaded. The request was signed by the Intermediate CA, and the resultant certificate was imported back into pfSense under System -> Cert Manager -> Certificates. This replaced the self-signed default certificate that was in place, offering secure administrative access to pfSense without browser trust warnings.

Group Policy Deployment of Certificates

To distribute the trusted Root and intermediate CA certificates across the Windows environment, a new Group Policy Object named “Trusted Root and Intermediate CA” was created. Using the GP management console, I linked the GPO created to the workstations and servers that we set up and are using in the different subnets. Under computer configuration -> policies -> window settings -> security settings -> public key policies, both the Root and

Intermediate CAs were imported into their stores. This made sure that all domain joined systems can now automatically have trusted certificates issued within the PKI infrastructure.

Once the GPO had been created, it was applied through the command "gpupdate /force" on client systems. It was verified by opening the Certificates Management Console, certmgr.msc, and double-clicking on the "Trusted Root Certification Authorities" folder to display the Root and Intermediate CAs in the respective folders: "Trusted Root Certification Authorities" and "Intermediate Certification Authorities". HTTPS connections to pfSense and RDP over SSL confirmed trust in the certificates and validity. At this stage, the two-tiered CA was up and running, and all domain devices had trusted certificates.

Troubleshooting

The implementation of a two-tiered CA system required addressing a host of technical problems systematically. The first major problem was related to certificate signing at the Root CA level. An "Invalid or Unsupported Request" message was encountered in the attempt to sign the certificate request of the Intermediate CA. Further investigation showed this was because the certificate request file was generated with the default SHA1 hash algorithm, which did not match the SHA256 set for the Root CA. This problem was fixed by regenerating the request using SHA256, and thus signing was successful. This highlights one important point in ensuring consistency of cryptographic configurations across all tiers of the CAs.

Other than that, another problem was in the issuance of a certificate for pfSense. At first, the Certificate Signing Request, generated by pfSense, did not validate in the CA console because the key usage field was not supported. By default, the pfSense CSR template used “TLS Web Client Authentication” instead of “Server Authentication.” So, to fix this issue, the CSR had to be created again with the proper EKU value. Re-submitted, the request went through, and the Intermediate CA signed it. The obtained certificate was then imported into pfSense with no errors. The web interface of pfSense now operates over a trusted HTTPS certificate; at least, browsers do not show warnings anymore.

Finally, a challenge that kept arising had to do with certificate chain validation on a Windows client. After adding the Root and Intermediate CA via Group Policy, the workstations still showed "Untrusted Root" errors. These seemed to relate to the delays in Group Policy replication and computers offline not yet being updated. Running "gpupdate /force" and

restarting the workstations forced these to resynchronize with the domain controller, after which the chain of trust was properly presented. In production environments, this may be quite normal; therefore, either allow time for propagation or run startup scripts to manually verify.