Jesse Russell
IA462 – Advanced OS Security
Christopher Krieger
Assignment Five

Design and Deploy a Two-Tiered Certificate Infrastructure

For my lab environment, the first thing I did was generate the root certificate authority for the rest of the certificates to build upon. To begin, I downloaded all the code modules from the class GitHub repository and placed them in my Windows Host box downloads folder, which can be accessed from my VM network. Using the CentOS 8 machine I set up previously, I then ran the 'ssl-genroot.sh' script to generate the certificate, changing the info inside to match my personal contact info. From there, I named this certificate "IA462" and proceeded to take the default options for the configuration until I was prompted to input a common name for it, which I named "Root Certificate Authority F21". This allows us to sign new certificates as well as verify that incoming certificates are authentic. Once that certificate was generated by the CentOS 8 box, I then copied that certificate to the domain controller windows server box in order to import and implement the certificate later.

From the DC virtual machine running Windows Server, I now had to create a group policy that would apply the certificate trust to all the machines on the domain. This way, the computers on the network will know to trust these certificates, since they are verifiably signed by the authority I created. To do this, I navigated to the Group Policy Management console and added a new GPO to the Group Policy Objects folder. This new GPO is called CA-CERT-Deploy. I disabled the user configuration just because I will not use it for this policy. Found under Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Polices > Trusted Root Certificates in the Group Policy Management Editor, the root certificate that was created earlier will be imported with all defaults taken. Once that certificate is imported, this GPO can be linked to the IA462.com domain, causing it to go into effect for every object in that domain. I figured this would be a good spot to verify that the group policy implementation was successful, which was done by using gpupdate and checking the local certificates for "Root Certificate Authority F21" under certificates > Trusted Root Certificates > Root Certificate Authority F21. I was able to see the certificate in this folder, which means that the Group Policy was successful.

Now that the group policy is a verified success, I can move onto generating an intermediate certificate for the PFSense box to manage. This was achieved via running the ssl-geninter.ssh' script on CentOS 8. First, I was prompted to input the root certificate, which was "IA462" (The one I configured earlier). I was also prompted for the name of this certificate, and I chose "PFSense-Inter" in an attempt to make the naming scheme as descriptive as possible. I then chose "PFSense Intermediate Signing Certificate" as the common name for the certificate. This script generates both a certificate and a private key, so both will need to be copied to the DC upon completion. From the DC after copying those files, PFSense needs to be configured to handle signing the certificate. Upon logging into the PFSense box browser interface, I was greeted with a certificate error, as per usual when logging into this interface. This message will be cleared after the next login because the certificate will be signed and trusted.

In PFSense, the Certificate Authorities are edited through System > Certificate Manager > CAs > Edit. The imported CA on the PFSense box will be called Root-CA and will handle the keys for the "Root Certificate Authority F21" created in the first step. The only configuration this step needs is to add the certificate data that was generated for the root, which I saved in the temp directory upon generation. The Root-CA does not need the private key entered. Once that data is saved, the PFSense Intermediate Certificate can be configured. This will be called "PFSense-Intermediate-Certificate" and follows the same process as the Root-CA, with a few extra steps. For the Intermediate CA, the private key needs to be unencrypted and put into the "Private Key Data" text field. This can be done by running "openssl rsa -in PFSense-Inter.key.pem -out PFSense-Inter.key.pem.nopass" and hitting enter. The command uses RSA to create an authorized version for the PFSense box to use without restriction. The PFSense-Intermediate-Certificate will use the serial number of 1 and will be added to the Operating System Trust Store. Both the "Root-CA" and "PFSense-Intermediate-Certificate" authorities are external certificates, and we need to create an internal certificate that will update the authority so we will not get that certificate error message we received earlier. This certificate is called "PFSense Web Certificate" and will be valid for 365 days. For the attributes of this certificate, I added the "pfsense.ia462.com" and "pfsense" hostnames, as well as the IP addresses of both the LAN and WAN interfaces on the box. Though the certificate is now created, it still needs to be chosen as the SSL certificate for the web interface, which can be done by going to Advanced setup and selecting the "PFSense Web Certificate" from the dropdown in the SSL/TLS Certificate field. I saved those settings and verified that the web interface now has a valid certificate by logging out, accessing the page again, and checking the certificate in my browser. I no longer received the certificate error, and I can see that the certificate is properly configured, including that it was issued to pfsense.IA462.com.

Returning to the DC, next we will install the Active Directory Certificate Services. To install, open Server Manager (if it's not already open) and click manage > Add Roles and Features and take the defaults until the "Select Server Roles" window is displayed. Here I selected "Active Directory Certificate Services," which should be the first option. On the "Role Services" screen, I enabled both "Certification Authority Web Service," "Certification Authority Web Enrollment", and "Network Device Enrollment Service." While I waited for the features to install, I created a service account for the certificate management process to be administrated through, called SRV_ADCS. I also created a new group for the account to reside in, called APP-ADCS, and added the DC server to it. Since I had already set up a GPO called FW-APP-ADCS, I just needed to link it to the group I had created. Under the FW-APP-ADCS GPO, the IIS-IUSRS group needs to be added to the restricted groups folder, and the SRV_ADCS account needs to be added to the IIS-IUSRS group. From here, the delegation tab needs to have APP-ADCS added, which should take the permissions specified in the GPO. The ADCS has finished installing on the DC, so that will now need to be configured. The CA will be set up as an Enterprise CA as well as a Subordinate CA. I then created a new private key with the length of 4096 and named the CA "Intermediate Certificate Authority". This will generate a request file that will need to be signed by the parent server and returned to the DC. Using the command from line 25 of "ssl-geninter.sh", we will generate a new windows certificate using the same process as before while including the request file this time. That certificate was copied to the DC, just as before, but was imported under the Public Key Infrastructure along with the PFSense intermediate key. Next, the CA certificate must be installed under the Certificate Authority menu. Then the windows key can be chosen for installation. This may throw an error that can be fixed using the steps detailed in the "Issues" section of this document. Once that error is taken care of, is the time to start the service by right clicking it and clicking "Start service." You will then have configure the ADCS again, and configure the roles that were not configured in the first set up. Here, the window prompts you for a service account, which is where the service account we created earlier comes into play. I was able to sign in with the credentials and continue the process. From this point, I just took the defaults until it was configured.

The last thing I had to do was set up some certificate templates so that the Windows Clients can automatically renew the certificates through a GPO. I set up one for the Domain Controller, Domain Controller Authentication, and the Web Server. All of these templates are running on Windows Server 2016 and allowed the proper accounts and groups to auto enroll in their respective CAs. In the GPO, I enabled the auto renewal rules that allows the process the be automated.

Issues

One problem I ran into was when I tried to install the CAs on PFSense, I had accidentally set them up as new certificates, instead of CAs. This is a simple mistake to fix, it just requires the process to be done again. This can be tedious at times.

In order to configure the ADCS on DC, your account needs to be a part of the Domain Admins and Enterprise Admins groups. This ensures your account has the proper permissions for all of the actions the configuration will take.

When I Copied the code from the SSL command in line 25, I had accidentally copied the wrong line from the code, which ended up giving me errors. Upon inspection, I found the script was copied wrong, and fixing the mistake solved the issue.

Generating the Windows certificate threw an error that I had not put the country in. I solved this issue by modifying the conf file and making those checks optional.

When adding the service account to the ADCS configuration, make sure that the account is in the IIS-IUSRS group so it can properly complete the configuration. Not doing so will throw an error. For some reason, the account did not add to the group earlier and it had to be done manually.

When installing the Windows key, it may throw an error that says something along the lines of how it can't be trusted. The error is thrown because the certificate needs a Certificate Revocation List (CRL). One solution is copying the code from the GitHub repo and altering it to fit my configuration. Once those CRLs point to the IA462.com repo, the certificate must be revoked and signed again. The Serial will also need to be changed back to what it was originally, as it will increment the number as a new serial if not changed. Then, in the DC, the CRL needs to be added to the DNS forward lookup zone linked to the CentOS8 machine. As it turns out, this was not the issue in my or the walkthrough's case. The issue lied inside the permissions of the /opt/ directory. This needed to be changed to 777.