Smooth CA certificate change

Stefan Kronawithleitner
CWNE Essay

I. Introduction

802.1X CA Certificate changes are always a pain in wireless, especially in a BYOD environment. As a university, most of our devices are not centrally managed. At the same time, our main WLAN is eduroam, which connects students securely and worldwide, so it is crucial to validate the certificate and not send authentication data to Man-in-the-Middle-APs.

Our old server certificate was issued by a public certificate authority. CWSP studies and WLPC talks taught me this is not ideal for multiple reasons:

- We are giving the power to issue certificates in our name to someone else and have to trust a third party to grant this right only to us
- Due to contracts, we had to switch the CA every six years. This meant we had to change every time this happened, leading to a 'flag day' where everyone had to reconfigure their devices to continue connecting securely. This process also took a toll on our service desk, as they had to assist all students simultaneously.

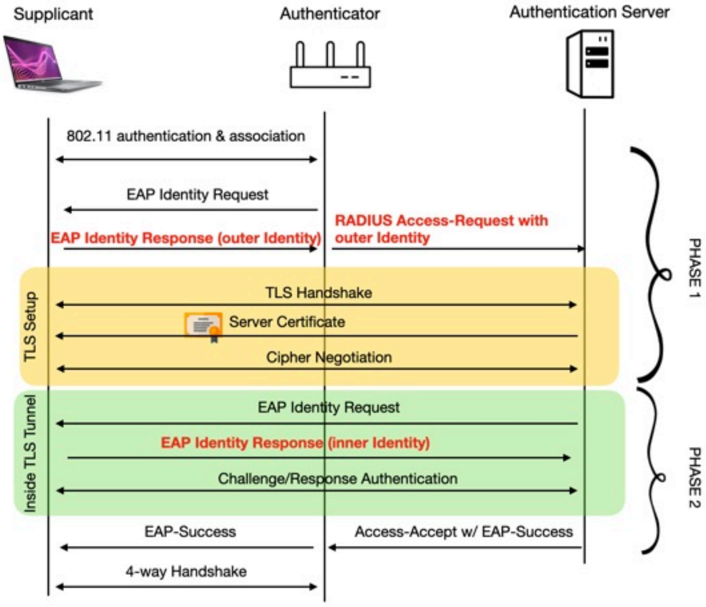
II. Looking for a solution

To avoid switching CAs so often, I decided to generate our own private CA to issue certificates for 802.1X in the future, stretching the time to only change for technical reasons - for example, if the CA cert requires longer keys with more bits or different hashing algorithms. While the certificate itself still has a short validity, the CA certificate was generated to be valid for 20 years.

The private CA solved the problem of having to change so often. The issue that remained was the flag day, and I started to look for a possible solution to make this more smooth.

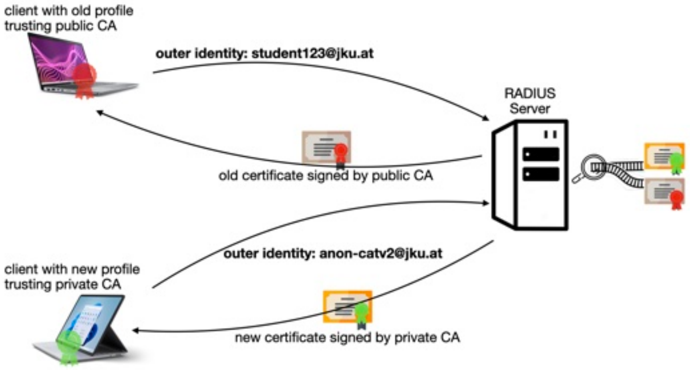
One idea was to cross-sign certificates, but a quick search revealed that most clients do not support this. The lack of support on iOS and macOS made this a non-starter. Another idea: Can I reference multiple CA certificates? Our onboarding tool allowed this, so I tested this feature. But while more clients seemed to work with this, Android phones did not support this or needed a relatively new Android version at the time. This would leave us with some clients that can have a smoother transition and some that don't, which is not ideal. So I kept looking for a better way and found the possibility of a kind of certificate "track switch."

This method takes advantage of the fact that tunneled EAP methods like EAP-PEAP have an outer- and inner identity. The client is supposed to send a "bogus" username in Phase 1 before establishing the TLS tunnel; the actual username and password are sent in the tunneled method during Phase 2. Default configurations of most operating systems send the same username for both, but an anonymized username for the outer identity is recommended.


Supplicant
Authenticator
Authentication Server
802.11 authentication & association
EAP Identity Request
EAP Identity Response (outer Identity)
RADIUS Access-Request with outer Identity
TLS Setup
TLS Handshake
Server Certificate
Cipher Negotiation
Inside TLS Tunnel
EAP Identity Request
EAP Identity Response (inner Identity)
Challenge/Response Authentication
EAP-Success
Access-Accept w/ EAP-Success
4-way Handshake
PHASE 1
PHASE 2

Simplified PEAP diagram highlighting outer and inner identity in Phase 1 and 2

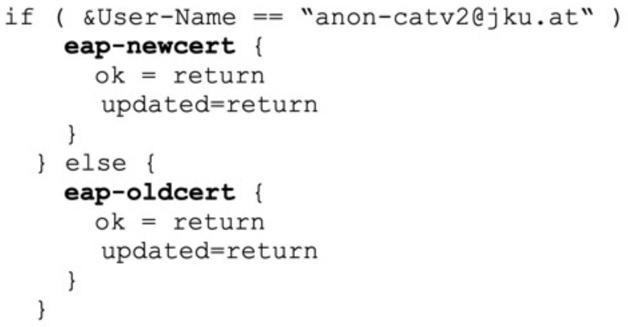
The certificate switch uses this to its advantage. As the client sends the outer identity before the RADIUS server sends the TLS tunnel certificate, we can discriminate with the information we receive. The goal is to send the new certificate (signed by our CA) when a specific outer identity is sent and the old certificate if any other identity is sent. The exact outer identity is set using the onboarding tool. As the particular username would only be set when the new profile by the onboarding tool is installed, it would trust our CA in this case, by which the certificate is signed.


client with old profile
trusting public CA
outer identity: student123@jku.at
RADIUS
Server
old certificate signed by public CA
outer identity: anon-catv2@jku.at
client with new profile
trusting private CA
new certificate signed by private CA

Different certificate presented to clients depending on outer identity

III. Testing and Implementing

I tested this using the usual clients, and as the only unique thing about this is a specific outer identity, it is supported by all the major clients we use (Windows, macOS, Linux / wpa_supplicant, iOS, and Android). All clients in my testbed supported this. The testbed I built strongly focused on Android devices with different versions, as I suspected that if I encountered problems, the diverse nature of Android manufacturers would play a role. The configuration on the server side was done using FreeRADIUS, where the scripting language “unlang” allowed this trick:



```
if ( &User-Name == “anon-catv2@jku.at“ )
  eap-newcert {
   ok = return
   updated=return
  }
 } else {
  eap-oldcert {
   ok = return
   updated=return
  }
 }
```

unlang configuration

As an added bonus, I used the outer identity as a versioning feature - including a version number in it - to see how far a new profile has spread with future changes.

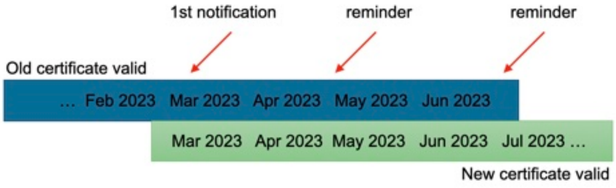
So people who had not yet installed the new profile - and thus did not send the specific outer identity - will see the old certificate as before, with no change.

People who installed the new profile will get the new certificate.

As the tests were all positive, the plan was set for the migration:

- make production certificates
- configure production FreeRADIUS with certificate switch
- e-Mail students to reinstall their profile with the onboarding tool in the next four months

This allows everyone to reconfigure their devices on their own time, and we can send reminders as the deadline approaches.



Smooth transition

IV. Conclusion

There were a few support cases of people wondering if they had to do this, suspecting phishing, or ignoring the e-mails. But all in all, this went very smoothly, and we successfully avoided a flag day.

After the old certificate expired, the extra configuration for the certificate switch was removed, and the new certificate—issued by our own CA—is now the default.

As the CA has a long duration and renewed certificates work without issues with correctly configured clients, I expect not to need this feature for a while. But if there will be another switch for technical reasons, this will be an excellent tool to keep changes of this magnitude smooth.