Partie I: Mise en œuvre d'une PKI avec tinyCA2 & utilisation d'openssl

1. Installez tinyCA2.

2. Mettez en œuvre une chaine de confiance permettant d'émettre:
   - Des certificats d'authentification utilisateur;
   - Des certificats d'authentification serveur.

   La chaine de confiance devra posséder les caractéristiques suivantes:
   - Au moins une AC subordonnée non opérationnelle;
   - Une profondeur minimale de 3.

3. Générer des requêtes de certificats afin d'émettre:
   - Plusieurs certificats d'authentification utilisateur;
   - Plusieurs certificats d'authentification serveur.

4. Emettez les certificats.

5. Via openssl, décodez les certificats.

6. A l'aide des certificats et des clés privées, générez des PKCS12.

7. Via openssl, validez un certificat serveur, un certificat utilisateur.

8. Révoquez un certificat serveur, un certificat utilisateur.

9. Emettez les CRLs.

10. Vérifier que le numéro de série des de chaque certificat a bien été ajouté à la CRL.

11. Echangez chaine de confiance avec un autre groupe.

12. Faîtes une requête de certificat afin qu'au autre groupe puisse vous émettre un certificat client.

13. Décodez le certificat obtenu.

14. Validez le certificat obtenu.

15. Générez un PKCS12 depuis le certificat obtenu.

16. Chiffrez un fichier texte à l'aide du certificat obtenu puis fournissez le fichier chiffré au porteur du certificat utilisé pour chiffrer le document.

17. Une fois en possession du fichier chiffré, déchiffrez le.

18. Réalisez les mêmes opérations avec un fichier binaire.

19. A l'aide de la documentation openssl, signez un document (signature détachée) puis fournissez le document ainsi que la signature détachée à un autre groupe.

20. Une fois en possession du document ainsi que de sa signature détachée obtenu d'un autre groupe, vérifiez la signature.

21. Encodez un fichier en base64.

22. Décodez le fichier encodé en base64 et assurez-vous de l'intégrité de ce dernier.

➔ Vous connaissez à présent la grande majorité des incantations magiques les plus utilisées avec openssl.

Partie II: Apache & mod_ssl

Les opérations suivantes sont à réalisées sur une machine virtuelle linux sur laquelle **vous possédez le compte root**.

1. Installez apache.

2. Installez mod_ssl.

3. Installez php.

4. Vérifiez le bon fonctionnement d'apache.

5. Créez une page affichant les informations système via php (phpinfo).

6. Validez le bon fonctionnement d'apache et php.

7. A l'aide de la PKI mise en œuvre dans la partie I, émettez un certificat serveur pour votre instance d'apache (le CN du certificat doit obligatoirement contenir le FQDN de votre instance Apache).

8. Configurez mod_ssl pour votre instance Apache afin :
   - D'utilisez le certificat émis lors de l'étape précédente ;
   - D'activer le SSL.

9. Validez le bon fonctionnement de votre instance apache SSL.

10. Le navigateur se plaint. Pourquoi ?

11. Mettez en œuvre les actions correctives permettant de supprimer le 'Warning' obtenu lors de l'étape précédente et validez à nouveau le bon fonctionnement de votre instance apache SSL.

➔ **Votre instance apache SSL est à présent fonctionnelle.**

12. En utilisant la PKI mise en œuvre lors de la Partie I, générez un certificat d'authentification utilisateur.

13. Via openssl, générez un fichier PKCS12 depuis le certificat généré lors de l'étape précédente.

14. Installez le certificat d'authentification client dans votre navigateur.

15. Mettez en œuvre l'authentification forte par certificat sur votre instance apache SSL.

16. Validez le bon fonctionnement de l'authentification forte par certificat sur votre instance apache SSL.

17. Générez une CRL pour l'AC ayant émis le certificat d'authentification utilisateur.

18. Modifiez la configuration de mod_ssl afin de vérifier le statut de révocation des certificats d'authentification utilisateur.

19. Essayez à nouveau de vous authentifier fortement sur votre instance apache SSL. Cela fonctionne. Pourquoi ?

20. Révoquez le certificat d'authentification utilisateur utilisé.

21. Essayez à nouveau de vous authentifier fortement sur votre instance apache SSL. Cela fonctionne. Pourquoi ?

22. Mettez à jour la CRL configuré dans votre instance apache SSL.

23. Essayez à nouveau de vous authentifier fortement sur votre instance apache SSL. Cela fonctionne. Pourquoi ?

24. Redémarrez votre instance apache SSL.

25. Essayez à nouveau de vous authentifier fortement sur votre instance apache SSL. Vous n'avez plus accès. Pourquoi ?

26. Demander à un autre groupe de vous émettre un certificat d'authentification utilisateur.

27. Essayez de vous authentifiez en utilisant ce certificat. Vous n'avez pas accès. Pourquoi ?

28. Modifiez la configuration de votre instance apache SSL afin de pouvoir utilisé le certificat d'authentification obtenu depuis un autre groupe.

29. Essayez de vous authentifiez en utilisant ce certificat. Vous n'avez pas accès. Pourquoi ?

30. Demander au groupe vous ayant émis le certificat d'authentification utilisateur de vous fournir la CRL de l'AC ayant émis ce certificat.

31. Modifiez la configuration de votre instance apache SSL afin de consommer cette nouvelle CRL

32. Essayez de vous authentifiez en utilisant ce certificat. Cela fonctionne. Pourquoi ?

33. Emettez un certificat d'authentification serveur via votre PKI.

34. Générez un fichier PKCS12 depuis ce nouveau certificat.

35. Installez ce certificat dans votre navigateur.

36. Essayez de vous authentifiez en utilisant ce certificat. Cela fonctionne. Pourquoi ?

37. Modifiez la configuration de votre instance apache SSL afin de retreindre l'accès aux seuls certificats émis par votre AC émettant les certificats d'authentification utilisateur.

38. Essayez de vous authentifier fortement en utilisant le certificat d'authentification serveur. Vous n'avez pas accès. Pourquoi ?

39. Essayer de vous authentifier en utilisant le certificat d'authentification utilisateur émis par votre PKI. Cela fonctionne. Pourquoi ?

Question bonus : *Comment mettre en œuvre une gestion des accès plus fine sur une application php en utilisant comme 'principal' le DN du certificat d'authentification utilisateur ? Mettez en œuvre une solution minimaliste.*

Partie III: openvpn & certificate based authentication (mode solo)

L'objectif de cette partie consiste à mettre en œuvre une instance openvpn permettant au client de s'authentifier fortement par certificat. Utilisez les ressources à disposition sur Internet et n'hésitez pas à demander assistance si vous bloquez sur un point.

Partie III bis : Mise en œuvre de EAP-TSL sur freeradius (mode solo)

L'objectif de cette partie consiste à mettre en œuvre une instance radius (via freeradius) implémentant le protocole d'authentification EAP-TLS. Le bon fonctionnement du serveur radius permettrait, ultérieurement, de mettre en œuvre (la liste n'est pas exhaustive):

- Le 802.1X (switch authentifiant, accès sans fil) ;
- L'authentification pour les accès externes (VPN Ipsec, VPN SSL).

Utiliser les ressources à disposition sur Internet et n'hésitez pas à demander assistance si vous bloquez sur un point.

NB : *Pour tester le bon fonctionnement du protocole EAP-TLS, il sera nécessaire de compiler l'outil 'eapol_test' disponible dans les sources du supplicant 802.1x 'wpa_supplicant'.*