Université de Bretagne Occidentale | Faculté des Sciences & Techniques

Compte rendu :

Sécurisation d'un réseau d'entreprise multi site

Formation : Télécommunications, Réseaux et Cybersécurité

ACKO Juste Fourier

2024-2025

Introduction générale

Le projet de sécurisation d'un réseau filaire d'entreprise, mené par l'Université de Bretagne Occidentale au sein du laboratoire LabSticc, vise à renforcer la sécurité des accès au réseau. En raison de la présence de nombreux chercheurs externes, la protection de la confidentialité des données et des ressources internes du laboratoire est primordiale. Le défi consiste à assurer une gestion rigoureuse des accès tout en préservant la flexibilité des utilisateurs.

Le projet se décline en quatre lots principaux :

1. La mise en place d'une architecture AAA (Authentication, Authorization, Accounting) permettant de contrôler l'accès au réseau ;
2. L'authentification des utilisateurs par certificat, pour assurer une identification sécurisée et transparente ;
3. La création de VLAN dynamiques, afin d'isoler les chercheurs extérieurs et de limiter leur accès aux ressources ;
4. La sécurisation des communications entre les sites du laboratoire via un VPN, garantissant la protection des informations échangées.

Ce projet, centré sur la sécurité et l'efficacité des infrastructures, représente une étape clé pour le laboratoire en matière de protection des données sensibles et de gestion des accès réseau.

Lot 1 : Architecture AAA

Pour renforcer la sécurité du réseau du laboratoire et prévenir toute connexion non autorisée, la mise en place d'une architecture AAA (Authentication, Authorization, Accounting) est essentielle. Plutôt que d'utiliser le protocole classique RADIUS, nous avons opté pour FreeDiameter, qui offre un chiffrement plus sécurisé et une meilleure gestion des sessions.

Pour faciliter le déploiement, le serveur d'authentification sera hébergé sur le serveur Ubuntu de l'entreprise, centralisant ainsi les différents services du laboratoire. De plus, la solution proposée tirera parti des équipements réseau déjà disponibles, notamment les switchs Cisco 2950, afin de limiter les modifications de l'infrastructure existante. Si ces équipements ne supportent pas entièrement l'architecture, des équipements subsidiaires pourront être intégrés, avec les coûts associés inclus dans le budget global du projet.

1. ***Installation de freeDiameter***

Avant de procéder à l'installation de FreeDiameter, il est important de s'assurer que l'environnement est prêt et que toutes les dépendances requises sont disponibles :

- Un serveur sous Linux (Ubuntu)
- Un serveur LDAP opérationnel pour l'authentification des utilisateurs
- Les dépendances nécessaires :

sudo apt update && sudo apt install cmake gcc make libssl-dev libpcre3-dev libidn11-dev

2. ***Installation de freeDiameter***

git clone https://github.com/FreeDiameter/freeDiameter.git

cd freeDiameter

mkdir build && cd build

cmake ..

make

sudo make install

3. ***Configuration de freeDiameter***

Avant de pouvoir utiliser FreeDiameter, il est nécessaire de configurer correctement ses paramètres afin d'assurer une communication sécurisée et une intégration fluide avec l'infrastructure existante.

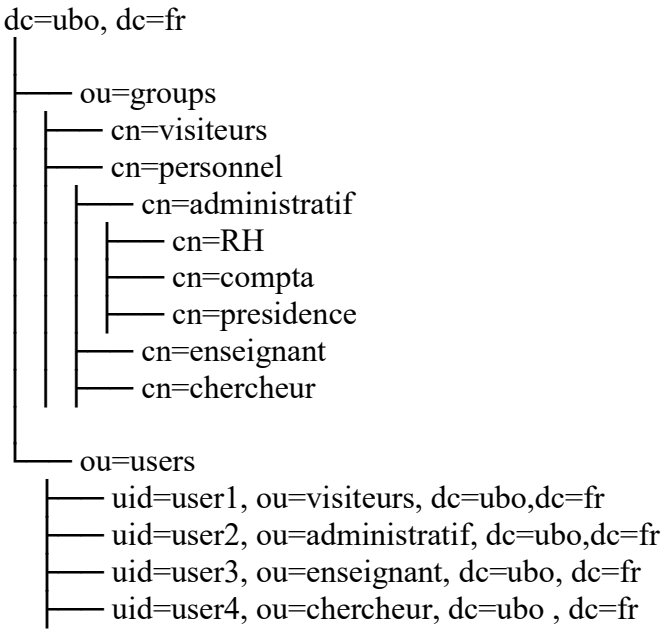
a. *Création du fichier de configuration (freediameter.conf)*

- Définir les identifiants du serveur
- Activer la connexion TLS
- Spécifier les peers (autres nœuds Diameter)

```
Identity = "ubo.fr";

Realm = "localhost";

TLS_Cred = "/root/myCA/ca.crt","/root/myCA/ca.key";

TLS_CA = "/root/myCA/ca.crt" ;

ConnectPeer = "client" {
ConnectTo = "192.168.10.2";
 NO_TLS;
Port = 3868;}; #connexion sans TLS mais on peut toutefois négocier une
connexion sur le secport 3869 selon les besoins de test
```

b. *Intégration de LDAP*

Nous avons créé une arborescence LDAP basée sur le modèle RBAC (Role-Based Access Control) afin de gérer les droits des utilisateurs de manière centralisée. Voici la structure de l'arborescence mise en place :



L'intégration avec LDAP n'étant pas native dans FreeDiameter, on a dû développer une extension spécifique basée sur les AVP (Attribute-Value Pairs) compilés en C pour permettre l'authentification et l'autorisation des utilisateurs via LDAP. Inséré dans le fichier de configuration de freeDiameter avec

LoadExtension = "/usr/local/etc.freeDiameter/extensions/ldap_users.fdx";

c. *Intégration de la classe vlan*

Dans cette section, nous détaillons le processus d'intégration de la classe VLAN dans l'annuaire LDAP en utilisant un schéma LDIF, en expliquant les étapes mises en œuvre pour structurer et organiser les informations au sein de l'arborescence LDAP.

```
dn: cn={4}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {4}vlan

dn: cn={5}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {5}vlan

dn: cn={6}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {6}vlan

dn: cn={7}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {7}vlan

dn: cn={8}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {8}vlan

dn: cn={9}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {9}vlan

dn: cn={10}vlan,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {10}vlan
```

Figure 1: Schéma de vlan dans l'arborescence

4. *Démarrage*

a. *Démarrage du serveur freeDiameter*

freediameterd -c /etc/freediameter/freediameter.conf

```
NOTI   libfdproto '1.6.0-1.5.0-128-g4c6785b' initialized.
INFO   libgnutls '3.7.3' initialized.
 DBG   Core state: 0 -> 1
NOTI   libfdcore '1.6.0-1.5.0-128-g4c6785b' initialized.
 DBG   Generating fresh Diffie-Hellman parameters of size 1024 (this takes some time)...
 DBG   Loading : /usr/local/etc/freeDiameter/extensions/ldap_users.fdx
NOTI   Chargement de l'extension LDAP Users...
NOTI   Tentative de connexion au serveur LDAP: ldap://localhost
NOTI   Connexion LDAP Etablie avec succès.
NOTI   Tentative d'authentification LDAP...
NOTI   Authentification LDAP réussie.
NOTI   AVP 'User-Password' introuvable, création en cours...
NOTI   Extension LDAP Users chargée avec succès.
NOTI   All extensions loaded.
NOTI   freeDiameter configuration:
NOTI     Default trace level .... : +1
NOTI     Configuration file ..... : /usr/local/etc/freeDiameter/freeDiameter.conf
NOTI     Diameter Identity ...... : ubo (l:3)
NOTI     Diameter Realm ......... : localdomain (l:11)
```

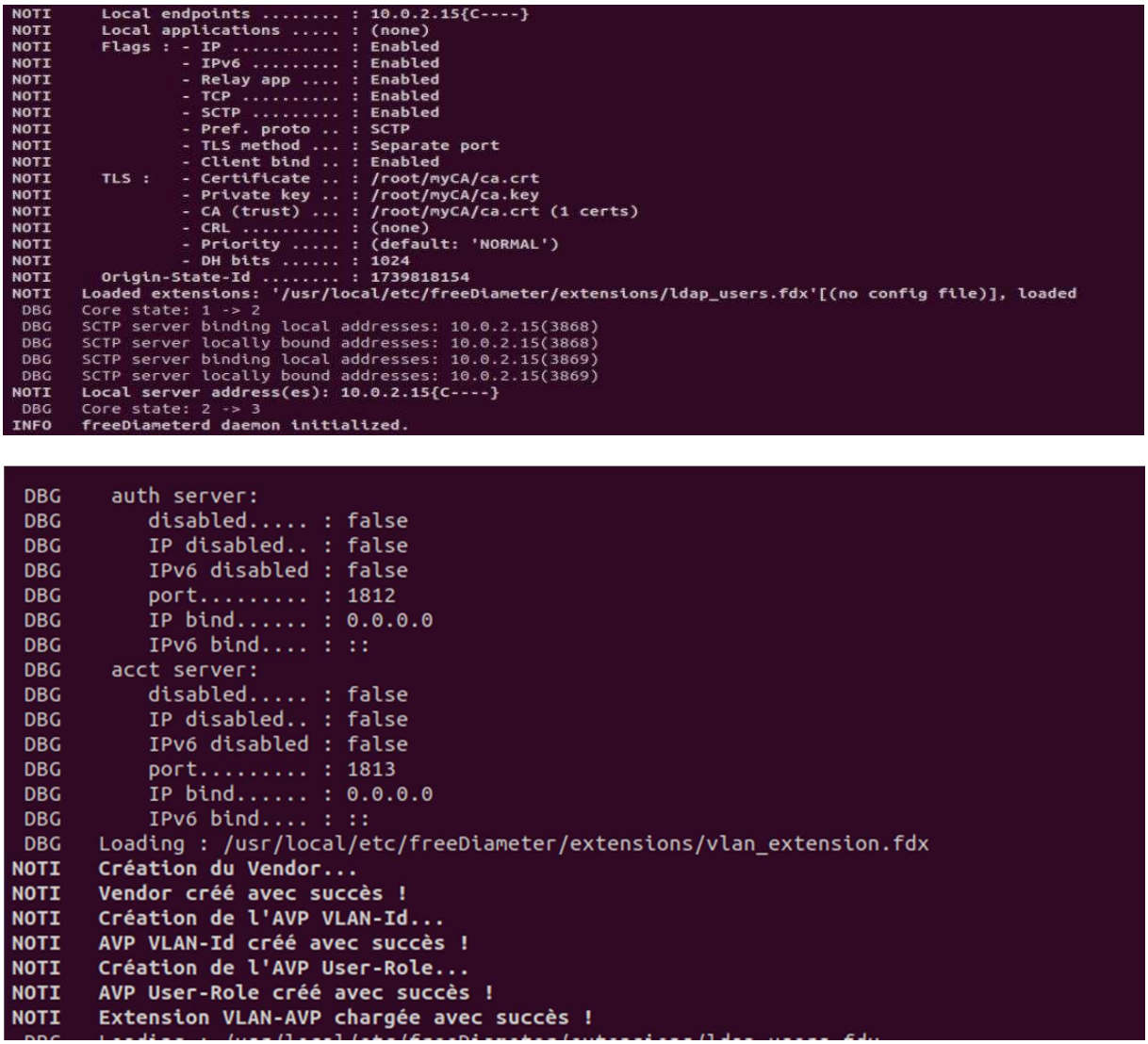


Figure 2: Lancement du serveur freeDiameter

L'installation et la configuration de FreeDiameter permettent d'établir une architecture AAA robuste et extensible. L'intégration avec LDAP facilite la gestion centralisée des utilisateurs et des permissions. Pour aller plus loin, il est possible d'ajouter d'autres extensions et d'optimiser la sécurité via des règles de pare-feu et TLS renforcé.

5. Tests

Le test de connexion entre le serveur freeDiameter et un client Diameter a été réalisé afin de vérifier l'établissement de la session et l'échange des messages CER. L'analyse des paquets capturés dans Wireshark permet d'observer les messages Diameter échangés, confirmant ainsi le bon fonctionnement de la communication. La création et la gestion des messages CER entre le client et le serveur ont été rendues possibles grâce aux bibliothèques de freeDiameter incluses dans freeDiameter, facilitant ainsi l'initialisation et la négociation des capacités entre les deux entités.

```
 DBG   Prepared 1 sets of connection parameters to peer client
 DBG   Connecting to TCP 192.168.10.2(3868)...
INFO   freeDiameterd daemon initialized.
 DBG   client: Connection established, {----} TCP,#8->192.168.10.2(3868)
NOTI   SND to 'client':
NOTI        'Capabilities-Exchange-Request'
NOTI          Version: 0x01
NOTI          Length: 148
NOTI          Flags: 0x80 (R---)
NOTI          Command Code: 257
NOTI          ApplicationId: 0
NOTI          Hop-by-Hop Identifier: 0x2744E68E
NOTI          End-to-End Identifier: 0x13CE58B4
NOTI           {internal data}: src:(nil)(0) rwb:(nil) rt:0 cb:(nil),(nil)((nil)) qry:(nil) asso:0 sess:(nil)
NOTI           AVP: 'Origin-Host'(264) l=11 f=-M val="ubo"
NOTI           AVP: 'Origin-Realm'(296) l=19 f=-M val="localdomain"
NOTI           AVP: 'Origin-State-Id'(278) l=12 f=-M val=1740398908 (0x67bc613c)
NOTI           AVP: 'Host-IP-Address'(257) l=14 f=-M val=192.168.10.1
NOTI           AVP: 'Vendor-Id'(266) l=12 f=-M val=0 (0x0)
NOTI           AVP: 'Product-Name'(269) l=20 f=-- val="freeDiameter"
NOTI           AVP: 'Firmware-Revision'(267) l=12 f=-- val=10600 (0x2968)
NOTI           AVP: 'Inband-Security-Id'(299) l=12 f=-M val='NO_INBAND_SECURITY' (0 (0x0))
NOTI           AVP: 'Auth-Application-Id'(258) l=12 f=-M val=4294967295 (0xffffffff)
 DBG   'STATE_WAITCNXACK'      -> 'STATE_WAITCEA'       'client'
```

```
NOTI   RCV from 'client':
NOTI        'Capabilities-Exchange-Answer'
NOTI          Version: 0x01
NOTI          Length: 188
NOTI          Flags: 0x00 (----)
NOTI          Command Code: 257
NOTI          ApplicationId: 0
NOTI          Hop-by-Hop Identifier: 0x2744E68E
NOTI          End-to-End Identifier: 0x13CE58B4
NOTI           {internal data}: src:client(6) rwb:(nil) rt:0 cb:(nil),(nil)((nil)) qry:0x73c488001240 asso:0 sess:(nil)
NOTI           AVP: 'Result-Code'(268) l=12 f=-M val='DIAMETER_SUCCESS' (2001 (0x7d1))
NOTI           AVP: 'Origin-Host'(264) l=14 f=-M val="client"
NOTI           AVP: 'Origin-Realm'(296) l=19 f=-M val="localdomain"
NOTI           AVP: 'Origin-State-Id'(278) l=12 f=-M val=1740399127 (0x67bc6217)
NOTI           AVP: 'Host-IP-Address'(257) l=14 f=-M val=192.168.10.2
NOTI           AVP: 'Vendor-Id'(266) l=12 f=-M val=0 (0x0)
NOTI           AVP: 'Product-Name'(269) l=20 f=-- val="freeDiameter"
NOTI           AVP: 'Firmware-Revision'(267) l=12 f=-- val=10600 (0x2968)
NOTI           AVP: 'Auth-Application-Id'(258) l=12 f=-M val=4294967295 (0xffffffff)
NOTI           AVP: 'Supported-Vendor-Id'(265) l=12 f=-M val=5535 (0x159f)
NOTI           AVP: 'Supported-Vendor-Id'(265) l=12 f=-M val=10415 (0x28af)
NOTI           AVP: 'Supported-Vendor-Id'(265) l=12 f=-M val=12345 (0x3039)
NOTI   peer client supports 0 applications but is a relay
NOTI   CONNECTED TO 'client' (TCP,soc#8):
```

Figure 3:Test de connexion côté serveur freeDiameter

```
 DBG   ubo: Connecting...
 DBG   'STATE_CLOSED' -> 'STATE_WAITCNXACK'    'ubo'
 DBG   Prepared 1 sets of connection parameters to peer ubo
 DBG   Connecting to TCP 192.168.10.1(3868)...
 DBG   ubo: Connection established, {----} TCP,#8->192.168.10.1(3868)
NOTI   SND to 'ubo':
NOTI        'Capabilities-Exchange-Request'
NOTI          Version: 0x01
NOTI          Length: 164
NOTI          Flags: 0x80 (R---)
NOTI          Command Code: 257
NOTI          ApplicationId: 0
NOTI          Hop-by-Hop Identifier: 0x3AE6DAA4
NOTI          End-to-End Identifier: 0x4AFDBDAF
NOTI           {internal data}: src:(nil)(0) rwb:(nil) rt:0 cb:(nil),(nil)((nil)) qry:(nil) asso:0 sess:(nil)
NOTI           AVP: 'Origin-Host'(264) l=14 f=-M val="client"
NOTI           AVP: 'Origin-Realm'(296) l=19 f=-M val="localdomain"
NOTI           AVP: 'Origin-State-Id'(278) l=12 f=-M val=1740399791 (0x67bc64af)
NOTI           AVP: 'Host-IP-Address'(257) l=14 f=-M val=192.168.10.2
NOTI           AVP: 'Vendor-Id'(266) l=12 f=-M val=0 (0x0)
NOTI           AVP: 'Product-Name'(269) l=20 f=-- val="freeDiameter"
NOTI           AVP: 'Firmware-Revision'(267) l=12 f=-- val=10600 (0x2968)
NOTI           AVP: 'Inband-Security-Id'(299) l=12 f=-M val='NO_INBAND_SECURITY' (0 (0x0))
NOTI           AVP: 'Auth-Application-Id'(258) l=12 f=-M val=4294967295 (0xffffffff)
NOTI           AVP: 'Supported-Vendor-Id'(265) l=12 f=-M val=12345 (0x3039)
 DBG   'STATE_WAITCNXACK'      -> 'STATE_WAITCEA'       'ubo'
NOTI   RCV from 'ubo':
NOTI        'Capabilities-Exchange-Answer'
NOTI          Version: 0x01
NOTI          Length: 148
```

Figure 4: Test connexion du client Diameter au serveur

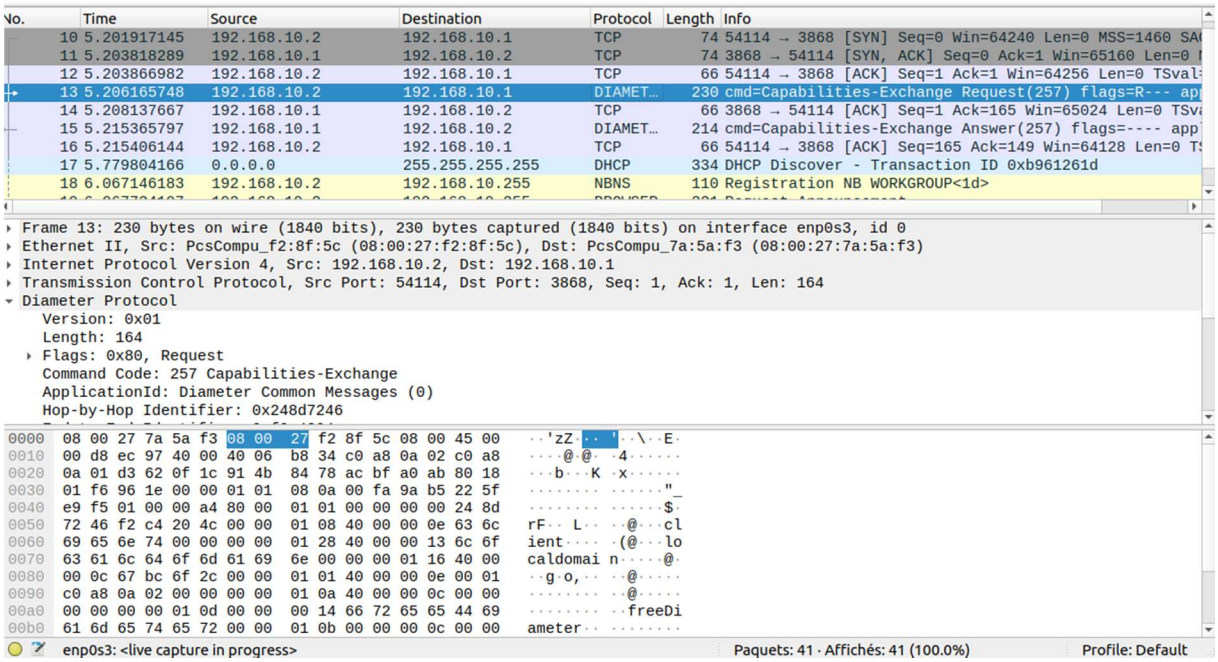


Figure 5: Capture des messages Diameter échangés

6. Authentification d'un client via le serveur Diameter

Pour vérifier l'authentification des utilisateurs via le serveur Diamter, un test a été effectué en tentant de se connecter avec l'utilisateur. L'objectif était d'évaluer la communication entre le switch et le serveur Diameter via Radius ainsi que la validation des identifiants.

Lors du test, un échange de messages RADIUS a eu lieu entre le switch et le serveur, comprenant l'envoi d'une requête d'authentification et la réception d'une réponse indiquant le statut de la demande. L'analyse de ces échanges permet d'observer le processus d'authentification et d'identifier d'éventuelles erreurs ou problèmes de configuration.

```
Switch#test aaa group radius Habib adminPass123 legacy
Attempting authentication test to server-group radius using radius

*Mar  1 01:17:41.691: AAA: parse name=<no string> idb type=-1 tty=-1
*Mar  1 01:17:41.691: AAA/MEMORY: create_user (0x3310C00) user='Habib' ruser='NULL' ds0=0 port='' rem_addr='NULL' aut
hen_type=ASCII service=LOGIN priv=1 initial_task_id='0', vrf= (id=0)
*Mar  1 01:17:41.691: RADIUS: Pick NAS IP for u=0x3310C00 tableid=0 cfg_addr=0.0.0.0
*Mar  1 01:17:41.691: RADIUS: ustruct sharecount=1
*Mar  1 01:17:41.691: Radius: radius_port_info() success=0 radius_nas_port=1
*Mar  1 01:17:41.691: RADIUS/ENCODE: Best Local IP-Address 192.168.40.10 for Radius-Server 192.168.40.8
*Mar  1 01:17:41.691: RADIUS(00000000): Send Access-Request to 192.168.40.8:1812 id 1645/9, len 57
*Mar  1 01:17:41.691: RADIUS:  authenticator 48 53 FD AB 1A 51 7D F5 - ED 7D DC 4E DA 74 35 28
*Mar  1 01:17:41.691: RADIUS:  NAS-IP-Address      [4]   6   192.168.40.10
*Mar  1 01:17:41.691: RADIUS:  NAS-Port-Type       [61]  6   Async                    [0]
*Mar  1 01:17:41.691: RADIUS:  User-Name           [1]   7   "Habib"
*Mar  1 01:17:41.691: RADIUS:  User-Password       [2]   18  *
*Mar  1 01:17:46.431: RADIUS: Retransmit to (192.168.40.8:1812,1813) for id 1645/9
*Mar  1 01:17:50.927: RADIUS: Retransmit to (192.168.40.8:1812,1813) for id 1645/9
*Mar  1 01:17:55.726: RADIUS: Retransmit to (192.168.40.8:1812,1813) for id 1645/9No authoritative response from any
server.
```

Figure 6: Log du switch après la tentative d'authentification

Lot 2 : Authentification par certificat

Pour garantir une sécurité renforcée sans compromettre l'expérience utilisateur, il est nécessaire de mettre en place une authentification fluide et efficace. Le système traditionnel de login/mot de passe étant peu adapté, l'authentification par certificats nominatifs devient indispensable. Afin de faciliter la gestion de ces certificats, une autorité de certification interne sera déployée sur un serveur Ubuntu, permettant ainsi une délivrance rapide des certificats. Le service technique disposera d'outils dédiés pour créer et révoquer ces certificats, et une formation sera proposée pour configurer efficacement les postes clients sous Windows 7 et Ubuntu, afin de garantir une intégration harmonieuse des nouveaux chercheurs au sein de l'infrastructure.

1. *Choix de la solution de l'autorité de certification*

Dans le cadre de notre projet, nous avons évalué deux solutions pour la gestion de l'autorité de certification (CA) et la génération de certificats : *EasyRSA* et *OpenSSL*. EasyRSA est une interface simplifiée, conçue pour faciliter la gestion des certificats avec des commandes prêtes à l'emploi. Son principal avantage est sa facilité d'utilisation, particulièrement adaptée pour des déploiements rapides et des besoins relativement simples, où une gestion automatisée et standardisée des certificats est suffisante. Cependant, cette simplicité vient avec des limitations en termes de flexibilité et de personnalisation.

En revanche, OpenSSL offre une plus grande flexibilité et un contrôle granulaire sur tous les aspects de la gestion des certificats. Bien qu'il soit plus complexe à configurer et à utiliser, il permet d'ajuster précisément les paramètres des certificats, comme l'ajout d'extensions spécifiques pour chaque utilisateur, ce qui est crucial dans notre contexte où chaque certificat doit être unique à chaque utilisateur et répondre à des exigences particulières de sécurité. Cette capacité de personnalisation, combinée à son large support dans l'industrie, a fait d'OpenSSL le choix idéal pour garantir que notre infrastructure PKI puisse s'adapter à des besoins évolutifs et complexes, tout en offrant une sécurité maximale.

Ainsi, malgré sa complexité, nous avons choisi d'utiliser OpenSSL pour ce projet en raison de ses fonctionnalités avancées et de la flexibilité qu'il offre dans la gestion des certificats, ce qui correspond parfaitement à nos exigences.

2. *Installer Openssl*

OpenSSL sera utilisé pour créer une autorité de certification et générer des certificats SSL.

```
sudo apt install openssl
```

3. *Créer un répertoire pour le CA*

Nous allons créer une structure pour héberger les fichiers de la CA :

```
mkdir ~/myCA
cd ~/myCA
mkdir certs crl newcerts private
touch index.txt
echo 1000 > serial
```

4. *Créer la clé privée de la CA*

La création de la clé privée de l'Autorité de Certification (CA) est une étape cruciale dans la mise en place d'une infrastructure de gestion des certificats sécurisée. Cette clé, qui doit rester confidentielle et protégée, est essentielle pour signer les certificats émis par la CA, garantissant ainsi leur authenticité et leur intégrité. La sécurisation de cette clé privée constitue le fondement de la confiance dans l'ensemble de l'écosystème de certificats.

openssl genrsa -aes256 -out private/ca.key.pem 4096

chmod 400 private/ca.key.pem

5. *Créer le certificat de la CA*

La création du certificat de l'Autorité de Certification (CA) représente une étape clé dans la mise en place d'une infrastructure de gestion des identités. Ce certificat, qui confirme l'identité de la CA, est vital pour signer et valider les certificats des utilisateurs et des dispositifs sur le réseau. En tant que certificat auto-signé, il joue un rôle crucial dans l'établissement de la confiance envers les certificats émis, garantissant ainsi la sécurité et l'intégrité des communications au sein de l'environnement informatique.

*openssl req –config /etc/ssl/openssl.cnf *

*-key private/ca.key.pem *

*-new -x509 -days 3650 -sha256 -extensions v3_ca *

-out certs/ca.cert.pem

chmod 444 certs/ca.cert.pem

Pendant ce processus, il faut fournir des informations telles que le nom de l'entreprise, le pays, unité, etc.

```
Country Name (2 letter code) [AU]:fr
State or Province Name (full name) [Some-State]:france
Locality Name (eg, city) []:brest
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ubo
Organizational Unit Name (eg, section) []:labsticc
Common Name (e.g. server FQDN or YOUR name) []:labsticc
Email Address []:espes@univ-brest.fr
root@acko:~/myCA#
```

Figure 7: création du certificat de la CA

6. *Créer la clé privée et la demande de signature de certificat (CSR) p*

Maintenant, nous allons générer une clé privée et un CSR pour le serveur

openss genrsa –out ca.key 2048

chmod 400 ca.key

Ensuite, créez une CSR avec cette clé :

openssl req -new -key ca.key -out ca.req

7. *Signer le certificat avec la CA*

openssl x509 -req -days 3650 -in ca.req -signkey ca.key -out ca.crt

```
root@acko:~/myCA# openssl x509 -req -days 365 -in ca.req -signkey ca.key -out c
a.crt
Signature ok
subject=C = fr, ST = france, L = brest, O = ubo, OU = labsticc, CN = labsticc,
emailAddress = espes@univ-brest.fr
Getting Private key
root@acko:~/myCA#
```

Figure 8: Signature du certificat

Maintenant, vous avez un certificat signé sur 10 ans par votre propre CA.

8. *Signature des certificats sur les routeurs*

La signature des certificats sur les routeurs a été effectuée afin d'assurer l'authentification et la sécurisation des échanges. L'analyse des certificats et des échanges TLS permet de valider leur bonne intégration et leur utilisation correcte dans les communications sécurisées.

```
R2#dir flash:
Directory of flash:/

    1  -rw-    50790652  Feb 22 1907 17:31:44 +00:00  c2800.bin
    2  -rw-        1012   Jan 1 2034 03:26:20 +00:00  running-config
    3  -rw-        1411   Jan 1 2034 18:01:30 +00:00  cacert.pem
    4  -rw-        1411   Jan 1 2034 18:21:02 +00:00  n
    5  -rw-        2118   Jan 1 2034 00:41:24 +00:00  my-ca-cert.pem
    6  -rw-        1598   Jan 1 2034 00:42:30 +00:00  my-R2-
    7  -rw-        1598   Jan 1 2034 00:44:48 +00:00  my-R2-cert-pem

64012288 bytes total (13193216 bytes free)
R2#sh crypto key mypubkey rsa
% Key pair was generated at: 00:31:58 UTC Jan 1 1970
Key name: R2.section.local
 Storage Device: private-config
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00845A85 617BCA6F
  BA1900FE BEB2DAA9 82DCDC97 86D6385C 15E53F52 E2938621 4128B48D 9AB514AE
  A29993AF A1E98348 A8D36257 ADB88AB6 F4BF2D4D 06E03099 0B020301 0001
% Key pair was generated at: 03:03:04 UTC Jan 1 1970
Key name: R2.section.local.server
Temporary key
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00B8B8D0 DE70EAAE
  4295AEB9 77443B00 53C4AD95 A1447C6D D471F8E4 E193DBD7 8141593A B6E00790
  A93C1A57 C1416906 04D608F8 239B7C2F C76DECAB F11119C7 A071086E A8E89C2C
  80D2D2F6 A4A1B370 F3B643F8 1E2ABAD1 FD7469CE FBBA36FB BB020301 0001
R2#
```

Figure 9: Signature des certificats sur les routeurs

Lot 3 : VLAN Dynamique

1. ***Création et transmission des AVP***

Les AVP (Attribute-Value Pairs) sont des éléments fondamentaux du protocole Diameter, définis dans la RFC 3588. Ils sont cohérents en des paires attribut-valeur qui transportent des informations spécifiques dans les messages Diameter, facilitant ainsi l'authentification, l'autorisation et la comptabilité (AAA) au sein des réseaux.

Dans le contexte de l'affectation dynamique des VLAN, les AVP jouent un rôle crucial. Lorsqu'un utilisateur s'authentifie sur le réseau, le serveur AAA utilise des AVP spécifiques pour transmettre des informations telles que l'ID du VLAN attribué à l'utilisateur. Cette approche permet une segmentation dynamique du réseau, où chaque utilisateur est automatiquement placé dans le VLAN correspondant à son profil ou à ses droits d'accès.

L'utilisation des AVP pour l'affectation dynamique des VLAN offre plusieurs avantages :

- **Flexibilité accrue** : Les administrateurs réseau peuvent définir des politiques d'accès granulaires, assurant que les utilisateurs sont placés dans des segments de réseau appropriés sans intervention manuelle.
- **Sécurité renforcée** : En isolant les utilisateurs dans des VLAN spécifiques basés sur leurs rôles ou niveaux d'autorisation

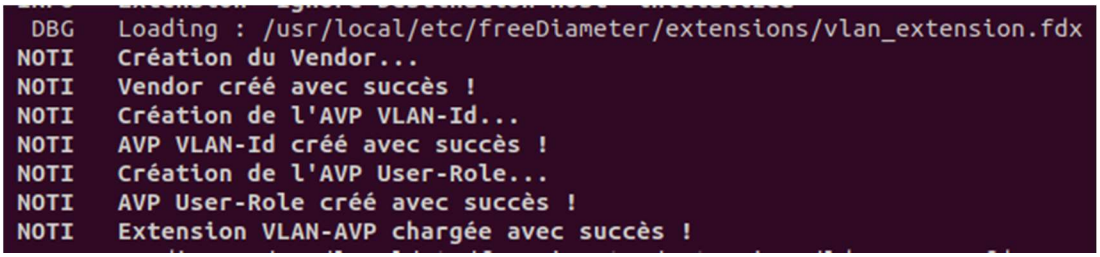


Figure 10: Chargement des AVP pour les vlans

2. ***Utilisation du 802.1X***

Le **protocole IEEE 802.1X** est une norme de contrôle d'accès au réseau qui fonctionne selon un modèle d'authentification par port. Il repose sur le protocole **EAP (Extensible Authentication Protocol)** pour l'échange des informations d'identification entre l'utilisateur et le serveur AAA qui est FreeDiameter.

a. ***Architecture du 802.1X :***

- **Supplicant (client)** : L'appareil utilisateur (PC) qui demande l'accès au réseau.
- **Authenticator (switch ou point d'accès Wi-Fi)** : L'équipement réseau intermédiaire qui contrôle l'accès au réseau.
- **Serveur d'authentification (AAA - FreeDiameter)** : Il vérifie les identifiants et renvoie une réponse d'autorisation ou de refus.

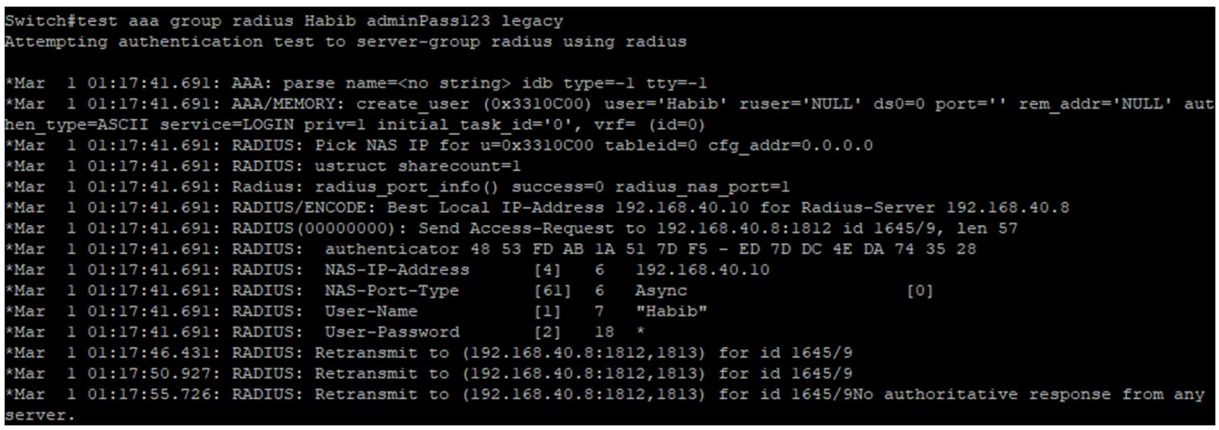


Figure 12: Authentification d'un client

b. *Affectation Dynamique des VLAN avec 802.1X*

Lorsqu'un utilisateur se connecte au réseau via un **switch compatible 802.1X**, l'authentification est gérée par le **serveur AAA**. Une fois l'utilisateur validé, le serveur AAA renvoie au switch un **attribut spécifique** (AVP) indiquant le VLAN auquel l'utilisateur doit être affecté.

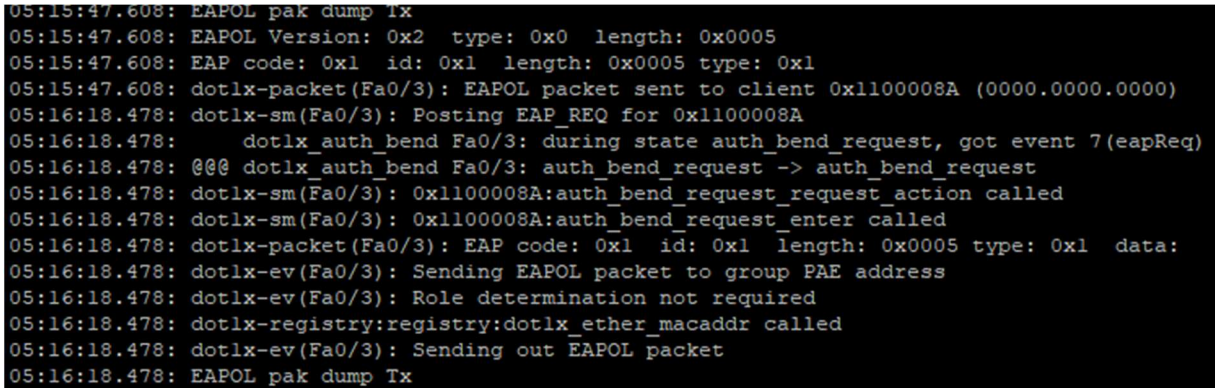


Figure 13: 802.X

- **Étapes du processus**
  - **L'utilisateur se connecte** et son appareil envoie une requête d'authentification 802.1X au switch.
  - **Le switch contacte le serveur AAA** (via Diameter) et transmet les identifiants de l'utilisateur.
  - **Le serveur AAA vérifie l'identité** et détermine le VLAN approprié en fonction des rôles ou groupes définis.
  - **Le switch affecte dynamiquement un VLAN** à l'utilisateur en se basant sur l'attribut VLAN reçu.
  - **L'utilisateur accède au réseau** uniquement au sein du VLAN qui lui est attribué

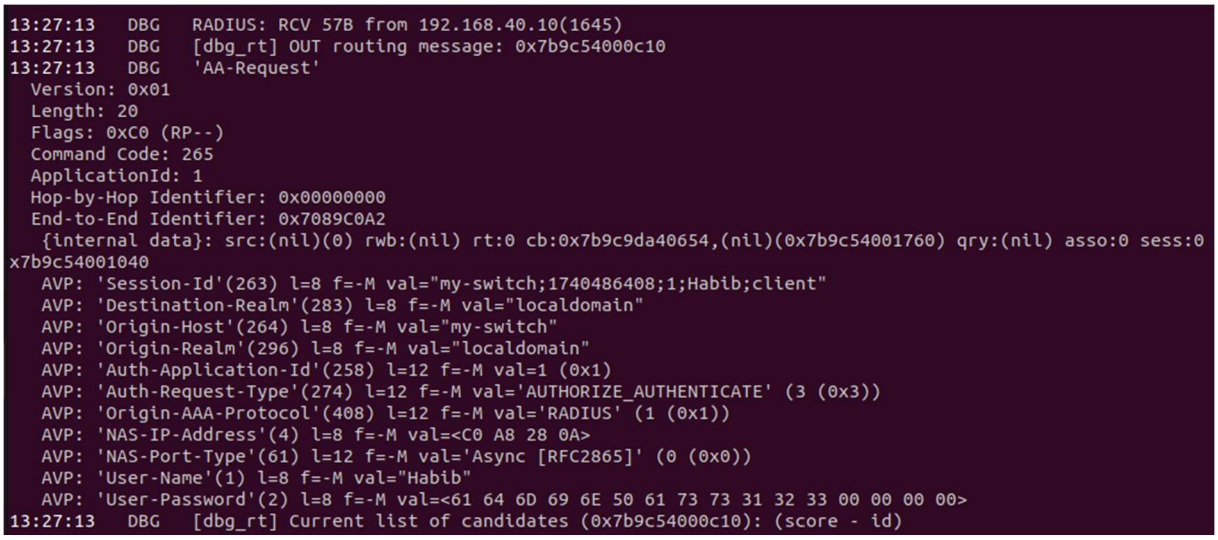


Figure 14: Logs de Diameter sur la tentative de connexion du client

3. *Mise en place du 802.1Q pour la propagation aux autres sites*

Le protocole IEEE 802.1Q est une norme qui permet l'étiquetage des trames Ethernet pour identifier les VLAN (Virtual Local Area Networks). Il est couramment utilisé pour propager plusieurs VLAN sur un seul lien physique, appelé "trunk", entre différents équipements réseau, tels que des commutateurs. Cette approche est particulièrement utile pour interconnecter des sites distants tout en maintenant une segmentation logique du réseau.

4. *Propagation des VLANs entre différents sites via le trunk des liens*

Lors de la connexion de plusieurs sites, il est essentiel de conserver la séparation des VLAN pour assurer une gestion efficace et sécurisée du réseau. Les liens trunk, configurés avec le protocole 802.1Q, permettent de transporter le trafic de plusieurs VLAN sur une seule liaison physique entre les sites. Chaque trame Ethernet est alors étiquetée avec un identifiant de VLAN (VLAN ID), garantissant que les données parviennent au VLAN approprié sur le site de destination.

5. *Avantages de l'utilisation du 802.1Q et des liens trunk pour la propagation des VLAN entre sites*

- **Efficacité** : Réduction du nombre de liaisons physiques nécessaires en regroupant le trafic de plusieurs VLAN sur un seul lien trunk.
- **Flexibilité** : Possibilité d'ajouter ou de supprimer des VLAN sans modifications physiques des connexions, facilitant ainsi l'évolution du réseau.
- **Cohérence** : Maintien de la segmentation logique du réseau à travers différents sites, assurant une politique de sécurité et de gestion uniforme.

Lot 4 : Sécurisation des communications inter-sites

Pour assurer la confidentialité, l'intégrité et l'authentification des échanges réseau, la mise en place d'IPSec (Internet Protocol Security) est essentielle. Cette solution permet de sécuriser les communications entre différents équipements du réseau, notamment pour les connexions VPN.

1. Présentation d'IPSec

IPSec (Internet Protocol Security) est un ensemble de protocoles permettant de sécuriser les communications sur un réseau IP en assurant l'authentification, l'intégrité et le chiffrement des données.
Les principales composantes d'IPSec sont :

- **AH (Authentication Header)** : assure l'authentification et l'intégrité des paquets, mais sans chiffrement.
- **ESP (Encapsulating Security Payload)** : fournit à la fois le chiffrement, l'authentification et l'intégrité des paquets.
- **IKE (Internet Key Exchange)** : protocole permettant l'échange de clés et la négociation des paramètres de sécurité entre les pairs.

Ces mécanismes permettent de créer des tunnels sécurisés pour la transmission des données sensibles.

2. Vérification et résultat de la configuration

Une fois la configuration effectuée, plusieurs commandes permettent de vérifier l'état du tunnel IPSec :

- **show crypto isakmp sa** : affiche l'état des associations de sécurité IKE.
- **show crypto ipsec sa** : présente les statistiques des sessions IPSec actives.

```
R2#show crypto ipsec sa

interface: Serial0/2/0
    Crypto map tag: mymap, local addr 181.23.12.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.10.10.0/255.255.255.252/0/0)
   remote ident (addr/mask/prot/port): (10.10.10.0/255.255.255.252/0/0)
   current_peer 181.23.13.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 181.23.12.2, remote crypto endpt.: 181.23.13.1
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/2/0
     current outbound spi: 0x0(0)
     PFS (Y/N): N, DH group: none

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

     outbound pcp sas:
R2#
```

Figure 15: Configuration de IPSec

La mise en place d'IPSec sur les routeurs permet de sécuriser efficacement les échanges de données entre sites distants. La configuration, associée aux captures d'écran, démontre que le tunnel est opérationnel et que le trafic est bien protégé.

Conclusion

Ce rapport a exploré la sécurisation d'un réseau filaire d'entreprise en mettant en place une architecture robuste reposant sur plusieurs mécanismes de contrôle et de protection. L'intégration d'une architecture AAA basée sur FreeDiameter a permis d'assurer une gestion centralisée des authentifications et des autorisations, garantissant ainsi un accès sécurisé au réseau. L'authentification par certificats, via une autorité de certification interne, a renforcé cette protection en offrant une méthode fiable d'identification des utilisateurs et des équipements.

Par ailleurs, la segmentation dynamique du réseau à l'aide de VLAN attribués en fonction des rôles des utilisateurs a optimisé l'organisation et la sécurité des accès, tout en améliorant la flexibilité du réseau. L'implémentation du protocole 802.1X a complété cette approche en rendant l'accès conditionnel à une authentification stricte. Enfin, la sécurisation des communications inter-sites a été assurée par IPSec, garantissant l'intégrité, la confidentialité et l'authenticité des échanges de données entre les équipements du réseau.

L'ensemble de ces solutions apporte une protection renforcée contre les accès non autorisés et les menaces potentielles, tout en maintenant une gestion efficace des utilisateurs et des ressources. Toutefois, certaines optimisations restent possibles, notamment en automatisant davantage la gestion des accès et des certificats, et en intégrant des stratégies de sécurité plus avancées.

Ce travail constitue ainsi une base solide pour le renforcement de la cybersécurité et ouvre la voie à de futures évolutions pour répondre aux enjeux croissants de la protection des infrastructures réseau.

Bibliographie

- https://github.com/freeDiameter/freeDiameter
- https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_8021x/configuration/xe-3se/3650/sec-user-8021x-xe-3se-3650-book/sec-ieee-8021x-vlan-assigned
- ietf.org/rfc/rfc4005.txt
- ietf.org/rfc/rfc3588.txt
- ietf.org/rfc/rfc4004.txt
- Diameter - Cisco
- https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_8021x/configuration/xe-3se/3850/sec-user-8021x-xe-3se-3850-book/sec-ieee-8021x-vlan-assign.html?#GUID-968934DE-8E01-430C-86B4-DAABC18EEF60
- https://reussirsonccna.fr/trunk-802-1q-et-isl-ce-quil-faut-savoir-pour-le-ccna/#:~:text=Le%20trunk%20normalis%C3%A9%20802.1Q,qui%20arrive%20sur%20le%20switch.