ADRAR FORMATION

Mis en place du 802.1x

MARAVAL Liam
09/12/2024

Ce document sera décomposé en plusieurs chapitres :

1. Contexte : Définition des besoins ainsi que le choix d'une solution en réponse à la demande client.
2. Procédure technique : Procédure technique de mise en place de la solution
3. Proposition de gestion du Wifi Guest par vouchers
4. Conclusion

Table des matières

1. Contexte 2
1.1 Demandes du client 2
1.2 Evolution de l'infrastructure 3
1.3 Choix des solutions 5
1.4 Sécurisation des accès 5
2. Procédure technique 6
2.1 Configuration du point d'accès Wifi 7
2.2 Configuration du serveur NPS 8
2.3 Test clients Wifi 19
2.3.1 Test VLAN 10 Stagiaires 19
2.3.2 Test VLAN 20 Administratif 22
2.3.3 Test VLAN 30 Guest 24
3. Proposition de gestion du VLAN Guest par vouchers 25
3.1 Contexte 25
3.2 Proposition de solution 26
3.3 Sécurisation des accès 27
3.4 Mise en place de la solution 28
3.4.1 Configuration de l'OPNSense 28
3.4.2 Configuration du service Flask 32
3.4.3 Configuration du serveur Apache2 33
3.4.4 Test de la solution 34
4. Conclusion 36
5. Annexes 37
5.1 Documentations et références 37

1. Contexte

1.1 Demandes du client

Nous sommes sollicités par l'ADRAR afin d'améliorer la sécurité de son réseau Wifi. En effet, l'ADRAR souhaite mettre en place un système de contrôle et de journalisation des connexions Wifi sur le réseau. Voici leurs demandes :

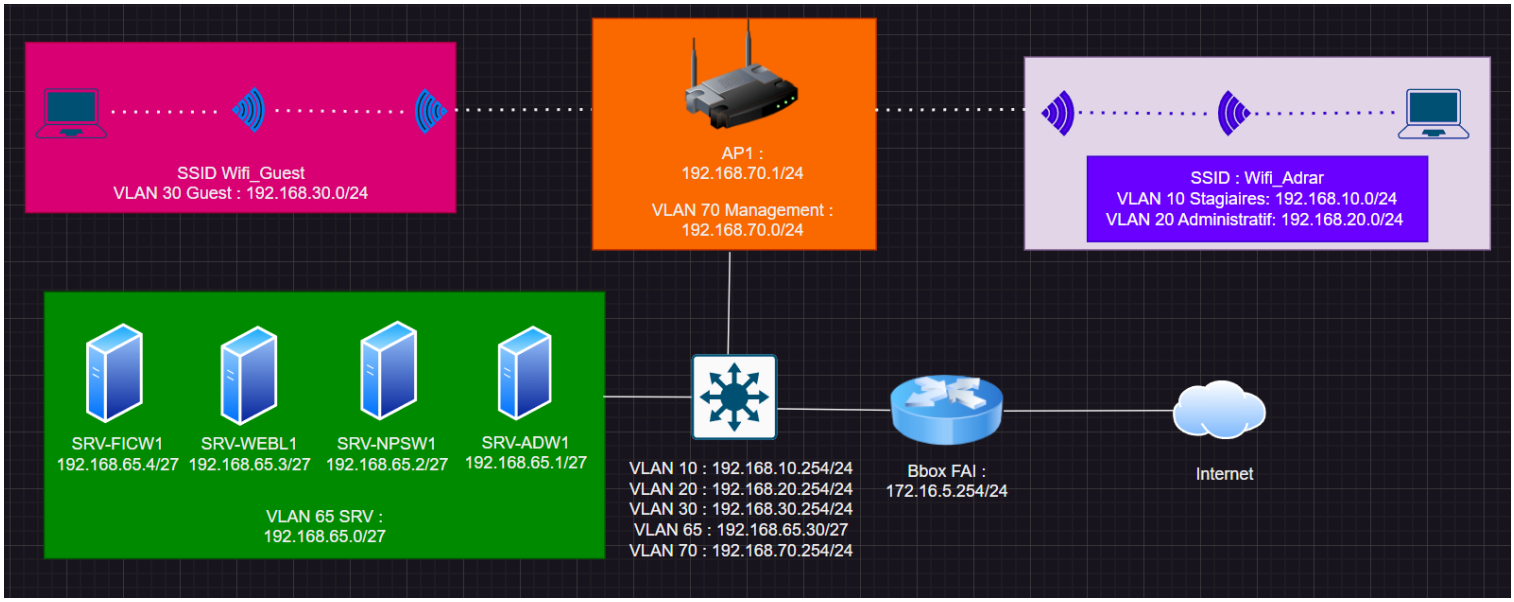
- Journaliser toutes les connexions Wifi
- Configurer 2 SSID « Administratif » et « Formation »
- Masquer le SSID Administratif
- Mettre en place un système d'attribution de VLAN automatique en fonction du groupe d'appartenance
- Sécuriser et journaliser le wifi Guest

Voici l'infrastructure en place actuellement :

- Une sortie internet via une Bbox en 172.16.5.254/24
- Un serveur Web Apache2 Debian 12
- Un serveur de fichiers Windows Server 2019
- Un serveur Active Directory/DNS avec un domaine « **adrar.local** »
- Un réseau global pour les postes utilisateurs et les serveurs
- Un point d'accès Wifi Cisco WAP371

1.2 Evolution de l'infrastructure

Voici le schéma de la nouvelle infrastructure :



Nous allons commencer par segmenter le réseau en 5 réseaux distincts :

- Un VLAN 10 **« Stagiaires »** pour les stagiaires en 192.168.10.0/24
- Un VLAN 20 **« Administratif »** pour les employés en 192.168.20.0/24
- Un VLAN 30 **« Guest »** pour les invités en 192.168.30.0/24
- Un VLAN 65 **« Serveurs »** pour les serveurs en 192.168.65.0/27
- Un VLAN 70 **« Management »** pour administrer les équipements en 192.168.70.0/24

En ce qui concerne la diffusion des SSID, l'ADRAR souhaitait deux SSID dont un qui affecterait automatiquement un VLAN en fonction du groupe d'appartenance de l'utilisateur.
Nous avons plutôt opté pour un SSID unique sur lequel l'attribution des VLANs se fera dynamiquement en fonction des groupes aussi bien pour les formateurs que les stagiaires.
Cela permet d'améliorer l'expérience utilisateur, simplifier la gestion et l'administration au quotidien tout en garantissant une segmentation sécurisée et efficace.

Ensuite, nous allons ajouter un serveur Radius qui permettra d'effectuer de l'authentification 802.1x.
Le protocole 802.1X permettra le contrôle d'accès réseau. Dans notre contexte, il va permettre un processus d'authentification sécurisé :

- L'utilisateur s'authentifie avec ses identifiants AD
- Le point d'accès WiFi agit comme authenticateur initiant le processus d'authentification avec l'utilisateur.
- Le serveur NPS valide les identifiants auprès de l'Active Directory et lui attribue un ID VLAN en fonction de son groupe d'appartenance.

Cette solution répond aux recommandations de l'ANSSI, notamment la R12 du document NP_WIFI_NoteTech-1.pdf, en :

- Utilisant une authentification forte (802.1X)
- Séparant les réseaux par profil
- Permettant la traçabilité des connexions

La mise en place de cette solution va avoir plusieurs avantages notamment :

- ✓ Attribution automatique du VLAN selon le groupe utilisateurs
- ✓ Pas de partage de clé WPA commune
- ✓ Révocation immédiate possible des accès
- ✓ Journalisation de connexions

Concernant le Wifi invité, le point d'accès Cisco WAP371 ne permettant pas de mettre en place une gestion de codes temporaires (vouchers), nous allons créer un utilisateur **« Guest ».**
Les identifiants seront à fournir aux personnes externes au centre pour pouvoir se connecter et à changer régulièrement.
Celui-ci sera rattaché à un VLAN étanche et n'aura le droit d'effectuer uniquement des requêtes Web en dehors du LAN.

A l'avenir, nous recommandons d'utiliser des solutions qui permettraient une meilleure gestion du Wifi Invité comme :

- Cisco Meraki
- Ubiquiti UniFi
- Aruba Instant On

Voici les informations du serveur NPS qui va être ajouté :

NOM	IP	Role	OS
SRV-NPSW1	192.168.65.2/27	Serveur Radius Windows	Windows Server 2019

Afin d'optimiser les couts de la licence Windows, un rôle DHCP sera également porté par ce serveur.
La bonne pratique voudrait, comme le recommande l'ANSSI, de dissocier les rôles sur des serveurs différents mais l'infrastructure et les raisons budgétaires de l'ADRAR ne leur permettent pas pour le moment.
Toutefois, nous pourrons à l'avenir réaliser la migration du service vers un autre serveur si l'ADRAR le souhaite.

L'ADAR ne disposant pas de PKI nous allons ajouter le rôle ADCS sur le contrôleur de domaine existant.
Ce rôle va permettre de promouvoir notre DC en tant qu'autorité de certification.
Il sera donc possible de délivrer des certificats via des templates pour tous les serveurs/postes de travail faisant partie du domaine.

Les certificats générés par notre CA vont servir notamment pour notre serveur Radius afin de sécuriser les données d'authentification au serveur.

Concernant les nouvelles règles mises en place afin de sécuriser les flux, veuillez-vous référer au chapitre 1.4 Sécurisation des accès

1.3 Choix des solutions

Pour l'authentification RADIUS, nous avons opté pour le "Network Policy Server (NPS)" de Windows Server 2019 pour les raisons suivantes :

- Intégration native avec l'Active Directory existant
- Configuration simplifiée des stratégies d'accès
- Compatibilité optimale avec l'infrastructure Microsoft
- Journalisation détaillée des connexions

Pour la gestion des certificats, nous avons opté pour Active Directory Certificate Services (ADCS) pour les raisons suivantes :

- Intégration native avec l'Active Directory
- Simplification de la gestion des certificats et de leur cycle de vie
- Compatibilité optimale avec les systèmes Windows et services Microsoft
- Suivi et audit centralisé des certificats et des clés

1.4 Sécurisation des accès

Pour répondre à la demande client, nous allons donc mettre en place un serveur Radius comme vu ci-dessus.
Le protocole d'authentification utilisé sera le PEAP.
Celui-ci sert à authentifier des utilisateurs de manière sécurisée au travers d'un tunnel chiffré.
Il nécessite un certificat côté serveur afin de pouvoir créer un tunnel TLS et d'échanger les informations de connexion de manière sécurisée sur le réseau.

Dans le système d'authentification Radius, c'est le point d'accès Wifi qui va jouer d'intermédiaire pour authentifier les supplicants (les pc, tablettes, smartphone ect…) auprès du serveur Radius.

Les filtrages de flux sont actuellement effectués via des ACLs existantes sur le Switch L3.
Comme mentionné plus haut, les différents VLAN hériteront des règles de filtrage en place.
Cependant pour le réseau Wifi Guest nous allons mettre en place de nouvelles ACL :

Guest					
Action	Protocol	IP Source	Port source	IP Destination	Port Destination
Block	any	192.168.30.0/24	any	192.168.10.0/24	any
Block	any	192.168.30.0/24	any	192.168.20.0/24	any
Block	any	192.168.30.0/24	any	192.168.65.0/27	any
Block	any	192.168.30.0/24	any	192.168.70.0/24	any
Pass	TCP	192.168.30.0/24	any	any	80
Pass	TCP	192.168.30.0/24	Any	any	443
Pass	TCP	192.168.30.0/24	Any	any	53

Le trafic sera interdit vers tous les réseaux de notre LAN, comme le recommande l'ANSSI.
Afin que les clients DHCP puissent avoir une adresse dynamiquement sans avoir à passer par nos serveurs, nous allons mettre en place un pool DHCP sur notre Switch.
Cette solution nous permet de n'ouvrir aucun flux vers notre réseau serveurs (les flux d'authentification Radius se faisant depuis le client Radius qui est notre point d'accès).

Pour le VLAN de management nous ajoutons également ces ACLs pour autoriser l'authentification Radius:

Management					
Action	Protocol	IP Source	Port source	IP Destination	Port Destination
Pass	UDP	192.168.70.1/32	any	192.168.65.2/32	1812
Pass	UDP	192.168.70.1/32	any	192.168.65.2/32	1645
Pass	UDP	192.168.70.1/32	any	192.168.65.2/32	1813
Pass	UDP	192.168.70.1/32	any	192.168.65.2/32	1646
Pass	UDP	192.168.30.0/24	any	192.168.65.1/32	53

2. Procédure technique

Afin de ne pas surcharger la documentation et d'éviter les doublons, nous ne reviendrons pas sur certaines procédures déjà établies pour l'ADRAR dans de précédents projets.

Voici les prérequis techniques nécessaires à la mise en place de notre solution :

- ✓ Configuration de VLAN sur un Switch Cisco 3700
- ✓ Configuration des ports d'un Switch Cisco 3700 (Tagged VLAN pour le trunk avec l'AP, untagged VLAN pour les ports access)
- ✓ Configuration d'un pool DHCP sur Switch Cisco 3700
- ✓ Créer des ACL sur un Switch Cisco 3700

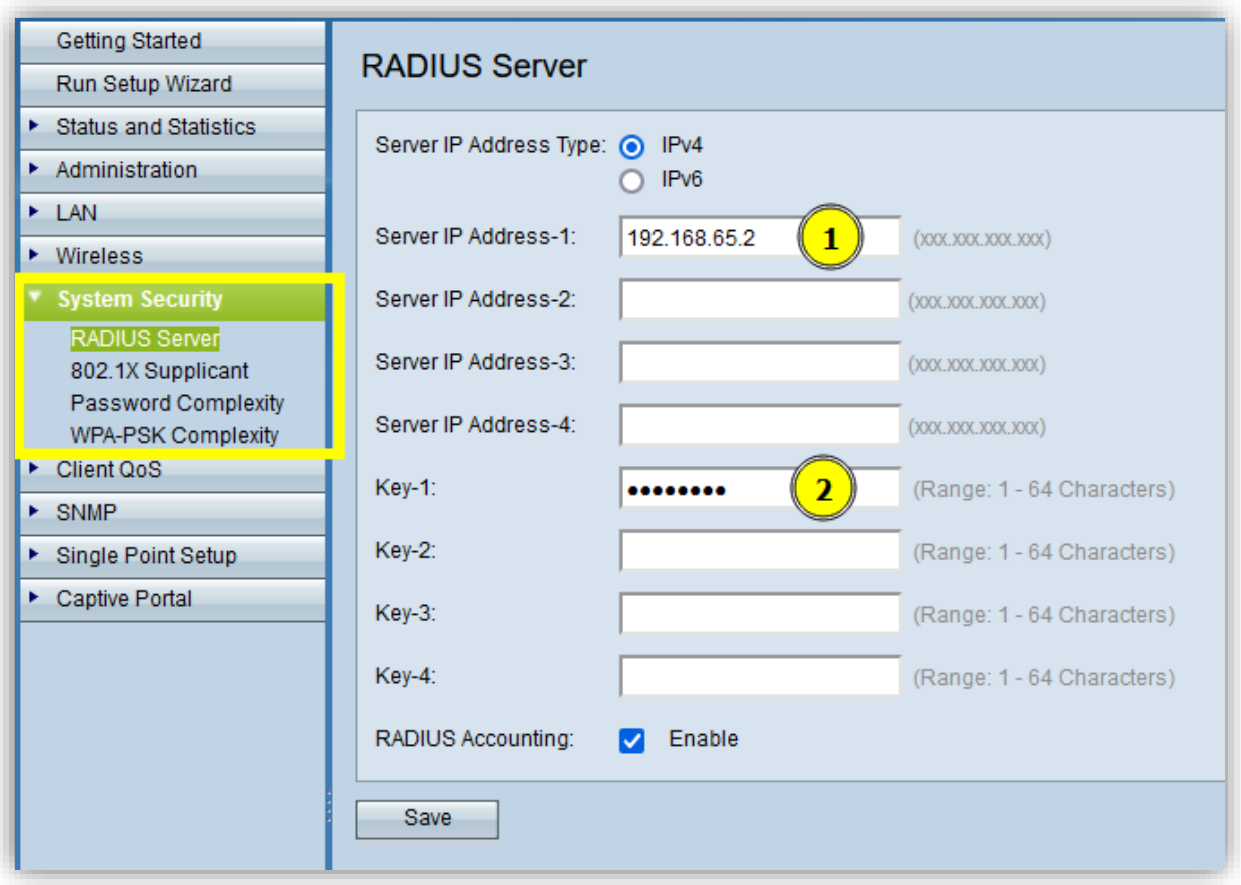
Pour plus de détails, merci de vous référer aux précédentes documentations rendues à l'ADRAR.

Concernant l'installation du rôle ADCS merci de vous référer à la documentation suivante :
https://www.it-connect.fr/adcs-creer-une-autorite-de-certification-racine-sous-windows-server/

2.1 Configuration du point d'accès Wifi

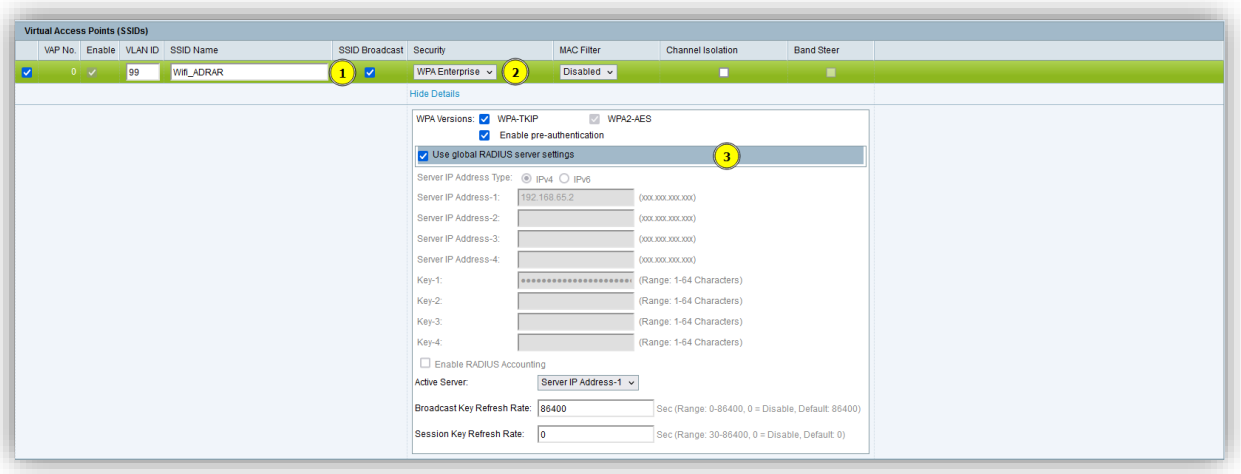
Nous commençons par le paramétrage de notre point d'accès Wifi nommé **« AP1 ».**
Celle-ci étant historique, les configurations de base sont déjà effectuées (nom, IP, domaine ect...)

Nous allons donc renseigner notre serveur Radius avec la PSK dans le menu suivant :



1. IP du serveur Radius
2. PSK définie entre le serveur Radius et le client

Nous configurons maintenant notre SSID :



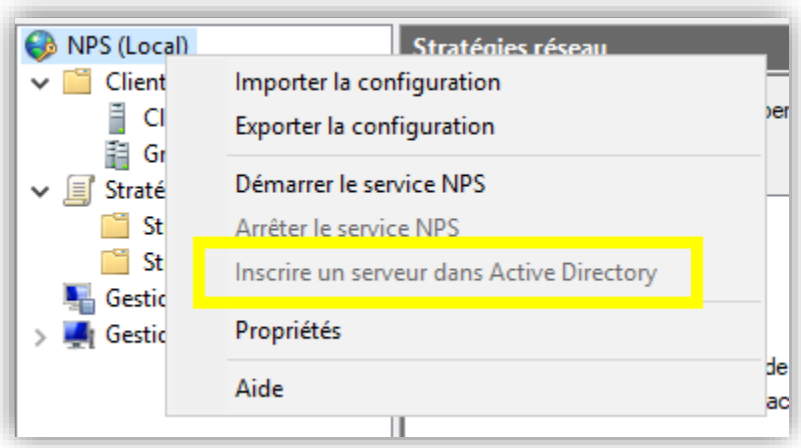
1. Nom du SSID avec l'ID du VLAN natif (en effet c'est le serveur Radius qui va définir le tag du VLAN pour les clients Wifi).
2. Mode d'authentification **« WPA_Entreprise »** via un Serveur Radius
3. Utiliser les paramètres Radius renseignés sur le point d'accès.

Notre SSID est maintenant configurée, si à l'avenir l'ADRAR souhaiterait masquer le SSID du réseau Wifi il suffirait de décocher la case **« SSID Broadcast »**

Une fois cela fait, nous allons configurer notre serveur NPS.

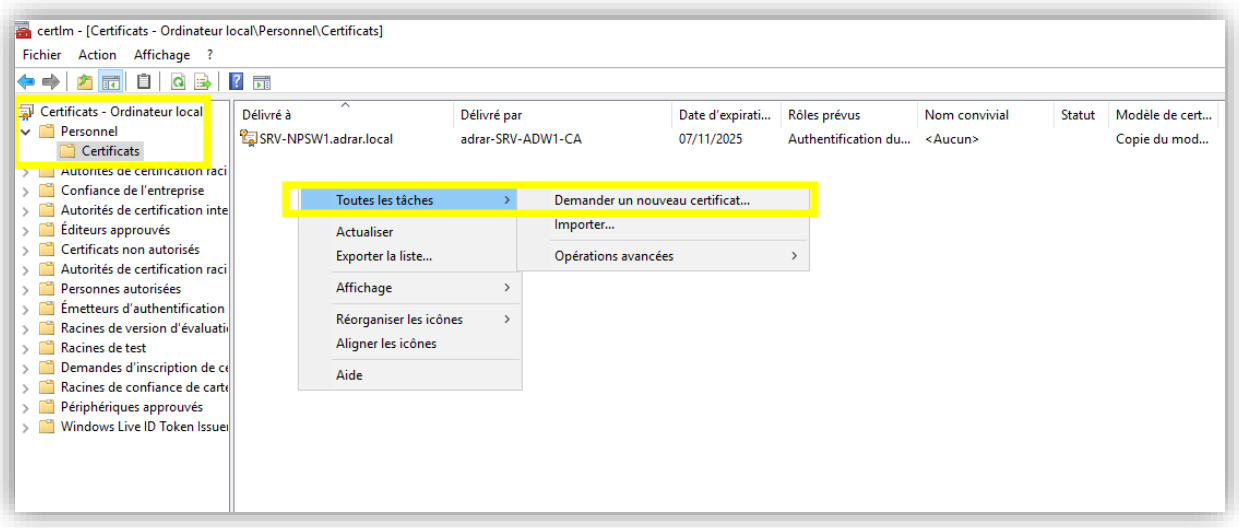
2.2 Configuration du serveur NPS

Une fois le rôle ajouté, nous commençons par ajouter notre certificat pour le serveur NPS.
Nous ouvrons la console **« NPS »** sur le serveur et cliquons sur la case suivante :

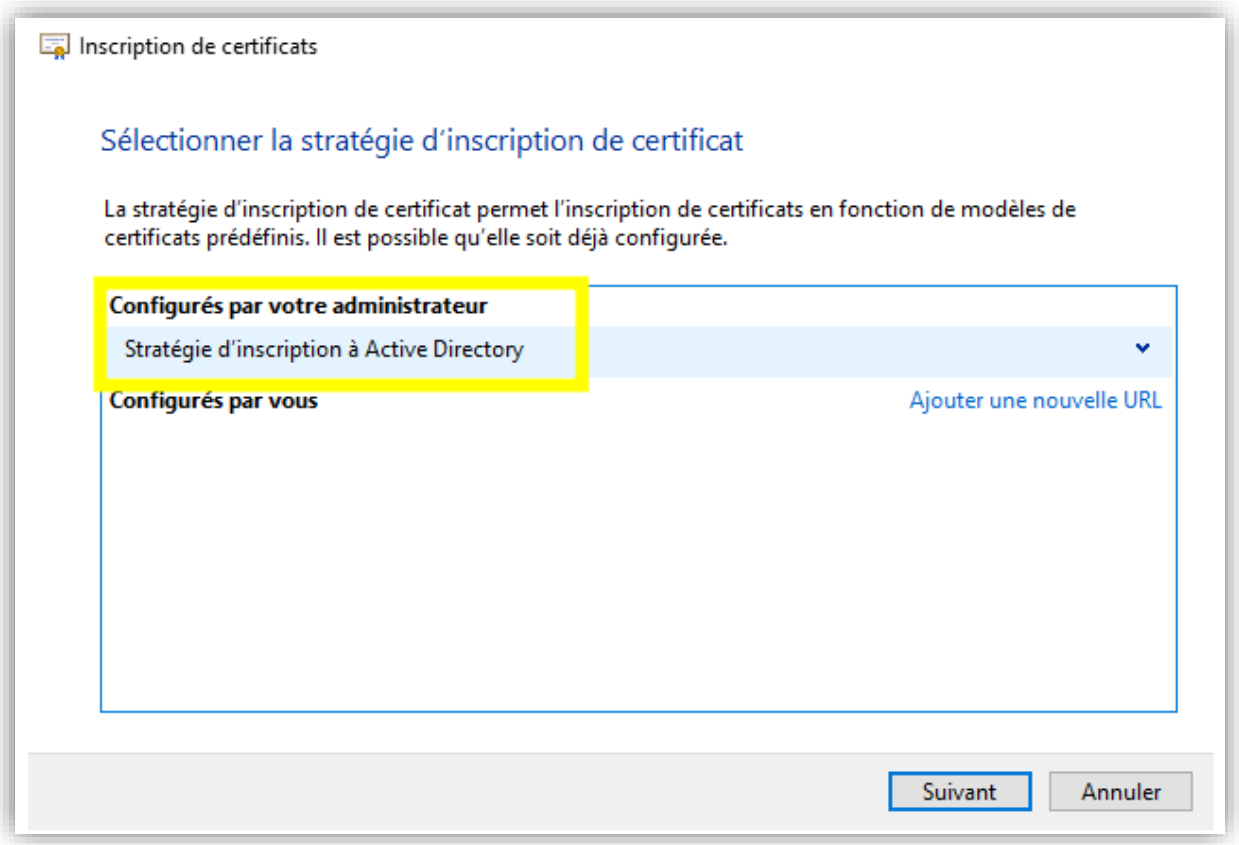


Cette option sert à rattacher notre serveur NPS à l'AD, ce qui va permettre de lui attribuer les bons groupes pour l'application des stratégies ainsi que pour demander notre certificat. (Sur l'exemple la case est grisée car le serveur est déjà inscrit auprès de l'AD)

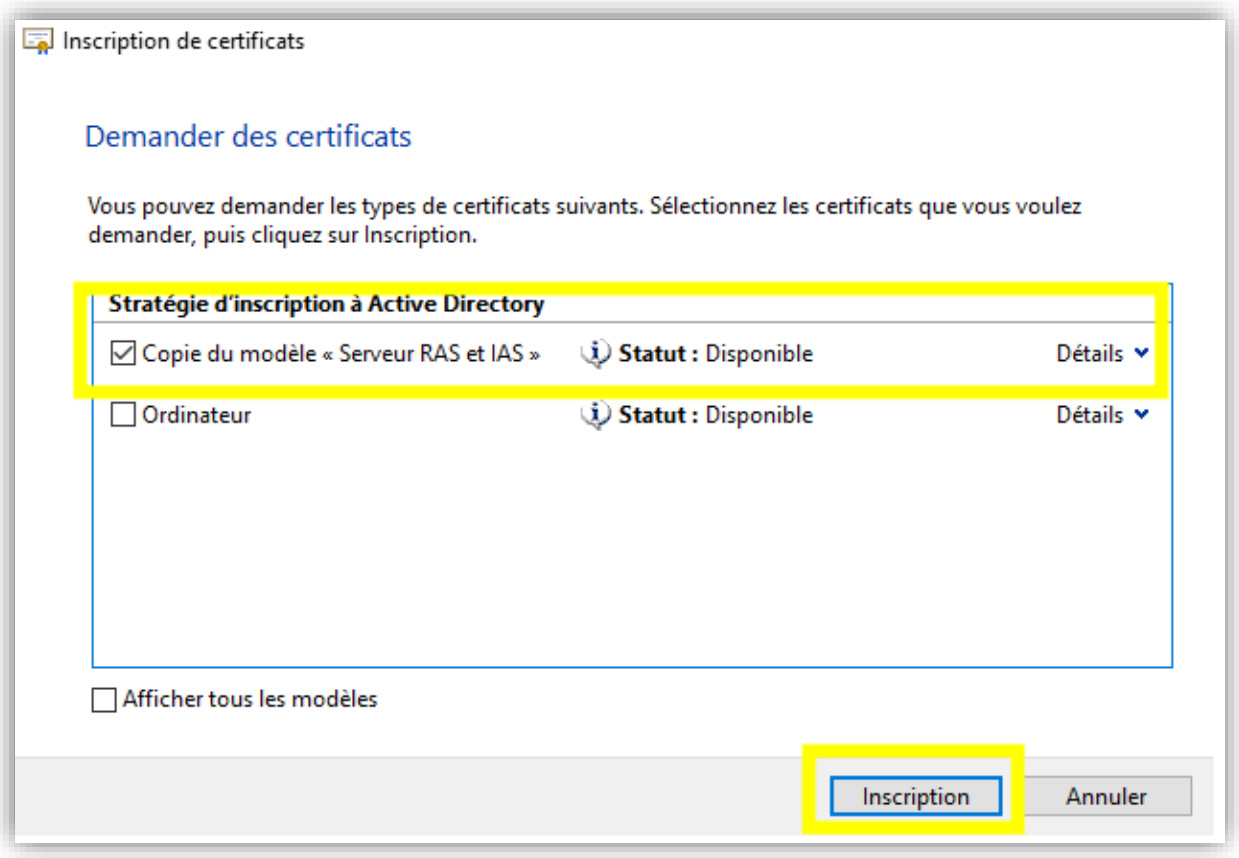
Nous allons maintenant nous rendre dans la console MMC pour générer notre certificat.



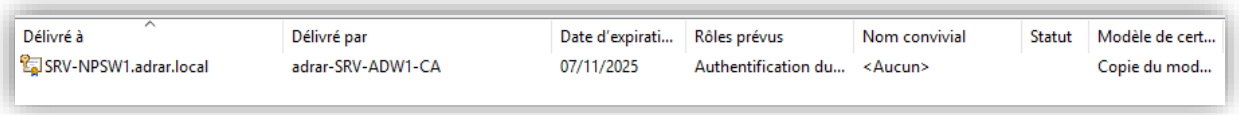
Une fois cela fait, nous choisissons de déployer le certificat via la stratégie d'inscription à l'AD :



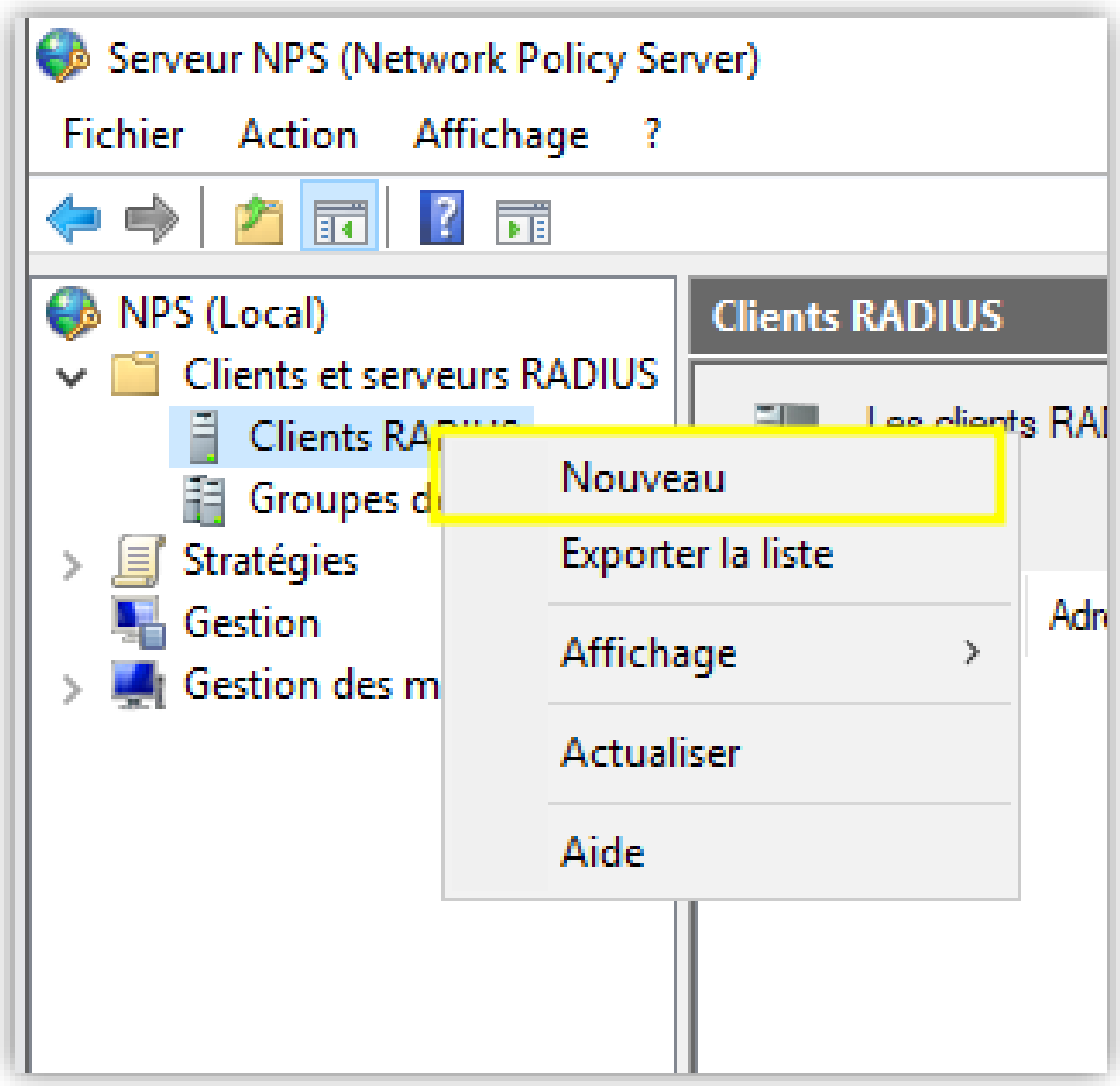
Nous choisissions le modèle serveur RAS et IAS qui correspond au modèle pour les serveurs NPS



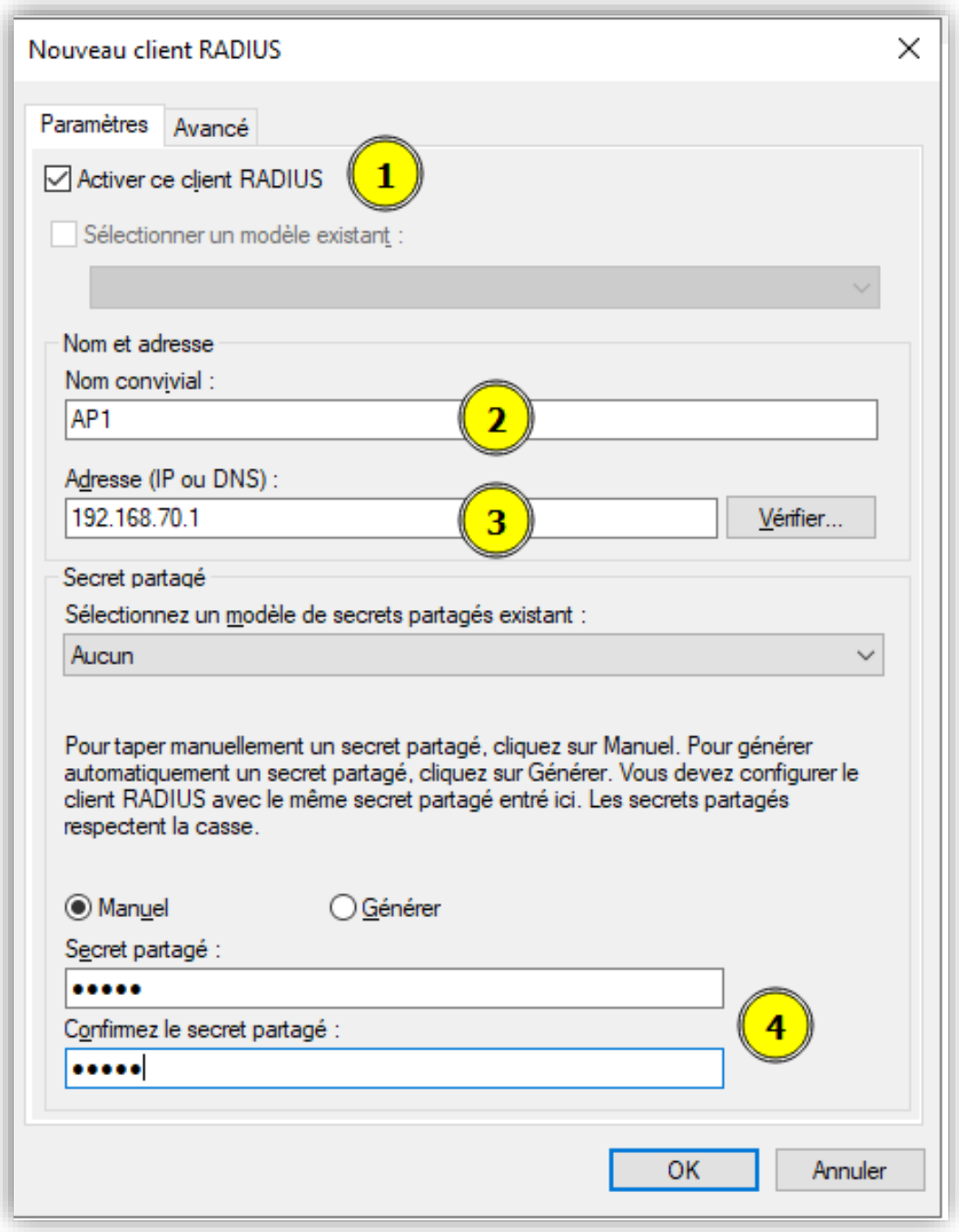
Notre certificat est maintenant généré :



Nous allons maintenant déclarer notre client Radius qui est l'AP1.
Nous allons dans le menu suivant :

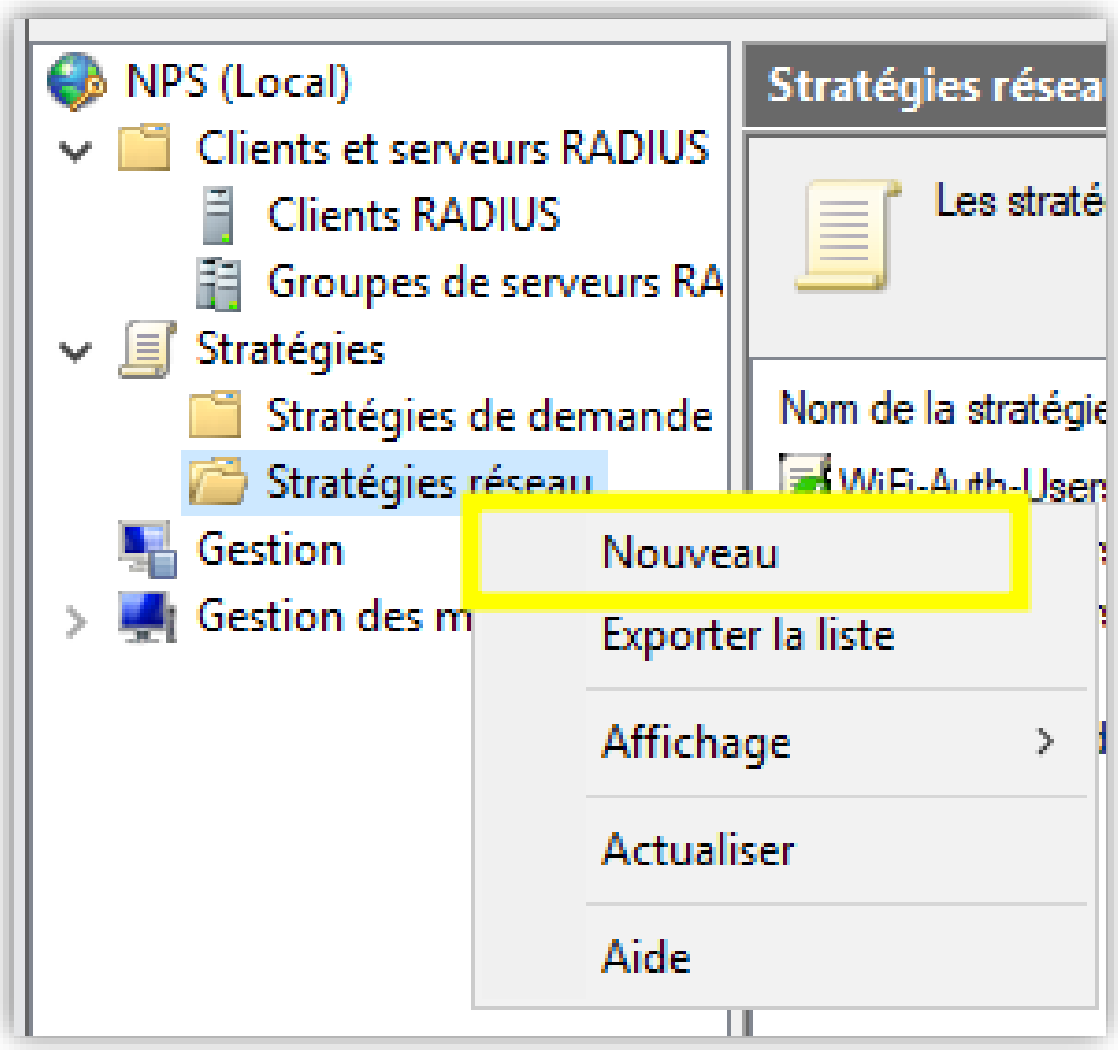


Puis nous renseignons les différents champs :

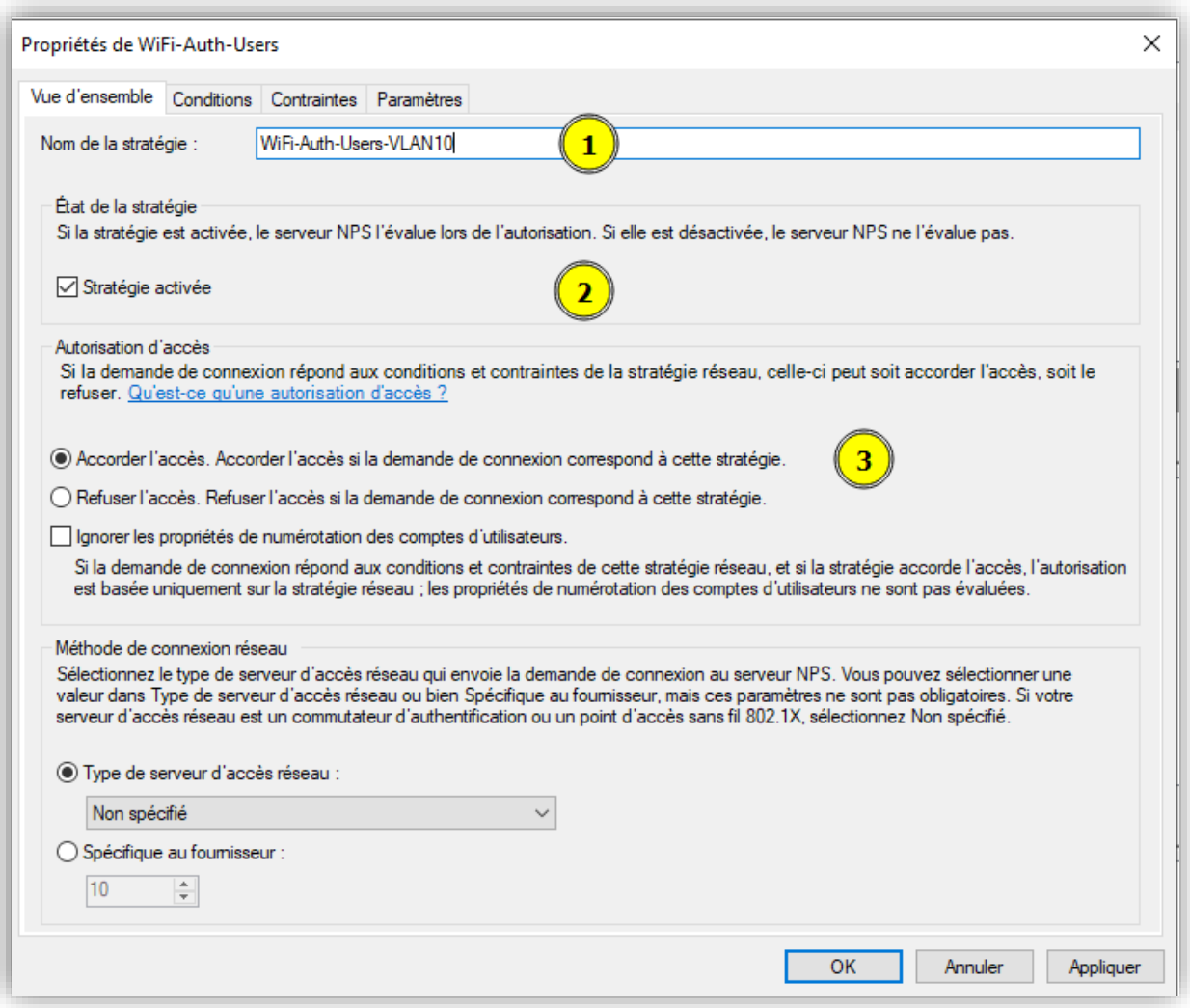


1. Activer le client Radius
2. Nom du client
3. Adresse IP du client
4. PSK définie entre le serveur et le client

Nous allons maintenant définir nos stratégies pour l'autorisation de connexion au réseau ainsi que l'attribution du VLAN dynamique.
Nous créons donc une nouvelle stratégie :

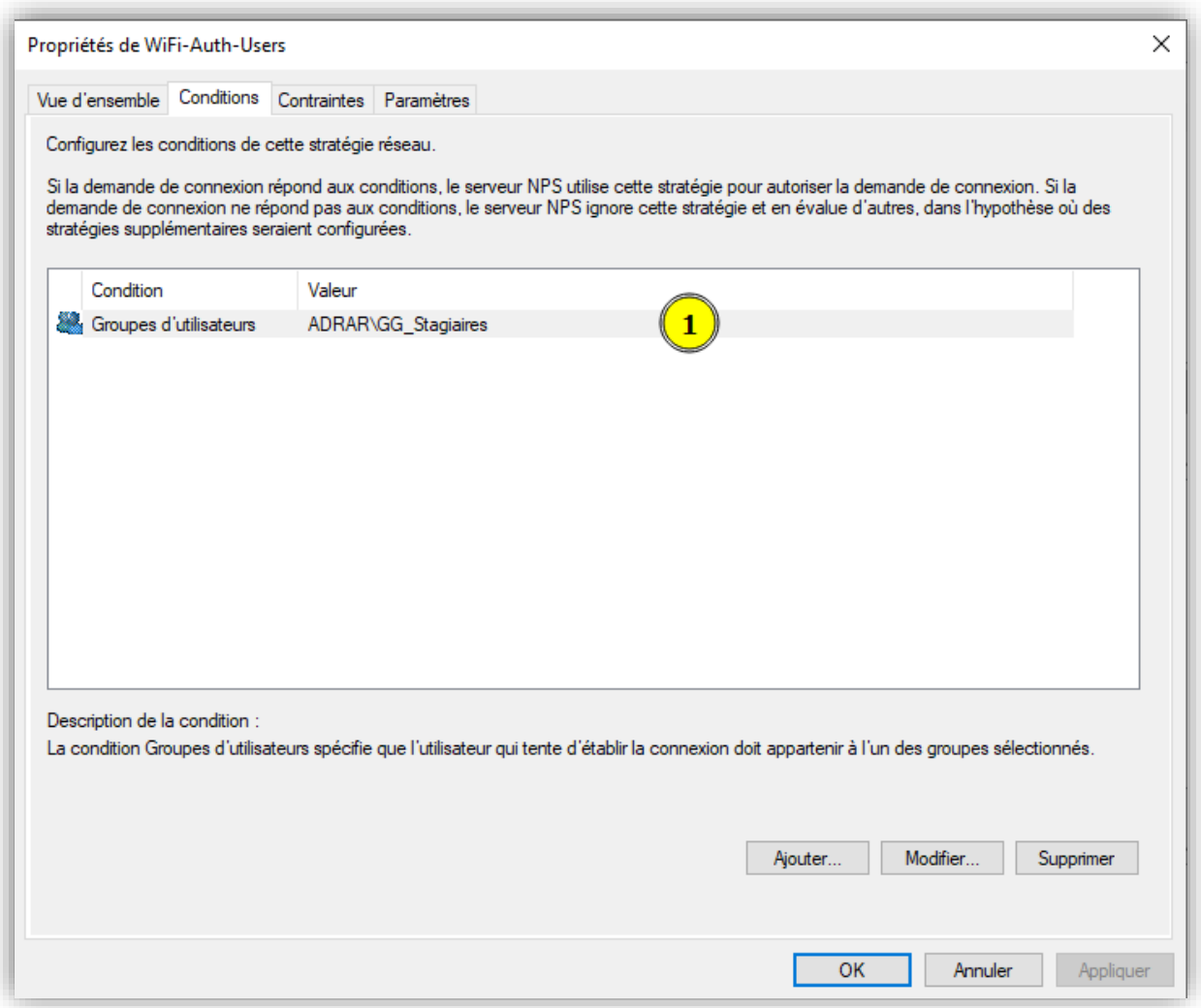


Nous renseignons les différents champs :



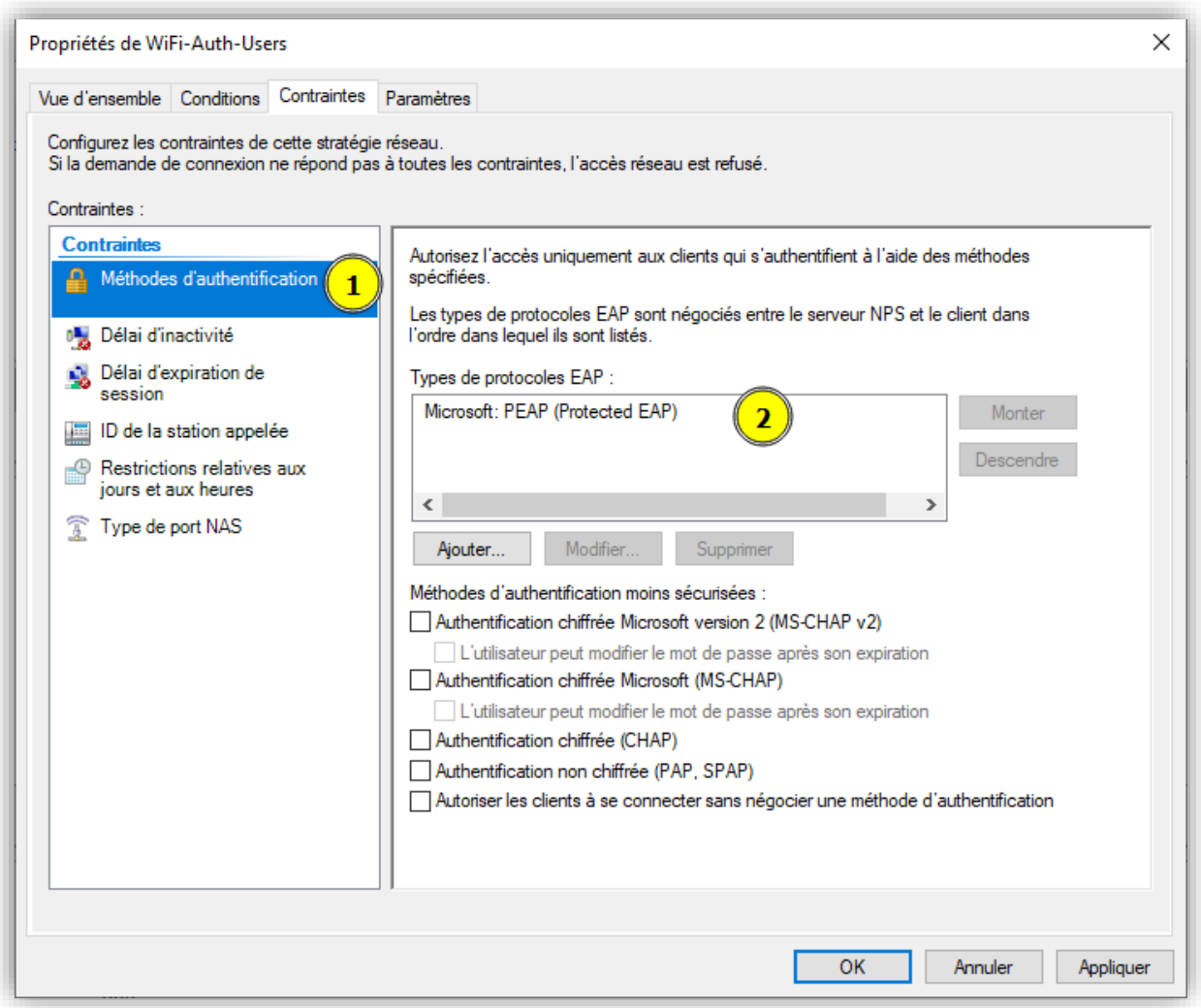
1. Nom de la stratégie
2. Activer la stratégie
3. Accorde l'accès réseau

Nous renseignons les conditions à respecter pour cette stratégie



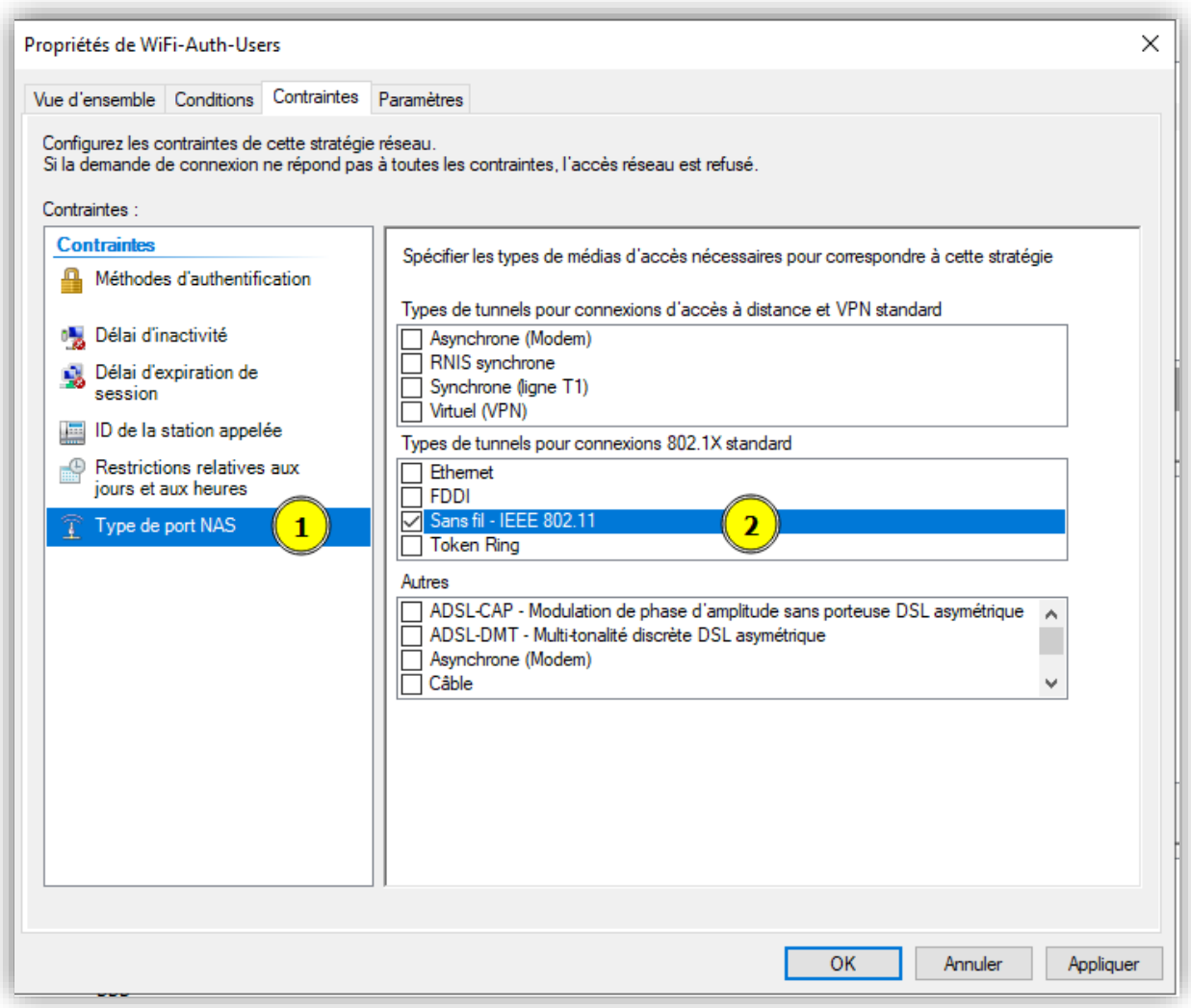
1. Condition d'appartenance au groupe pour accéder au réseau

Nous allons maintenant définir les méthodes d'authentification :



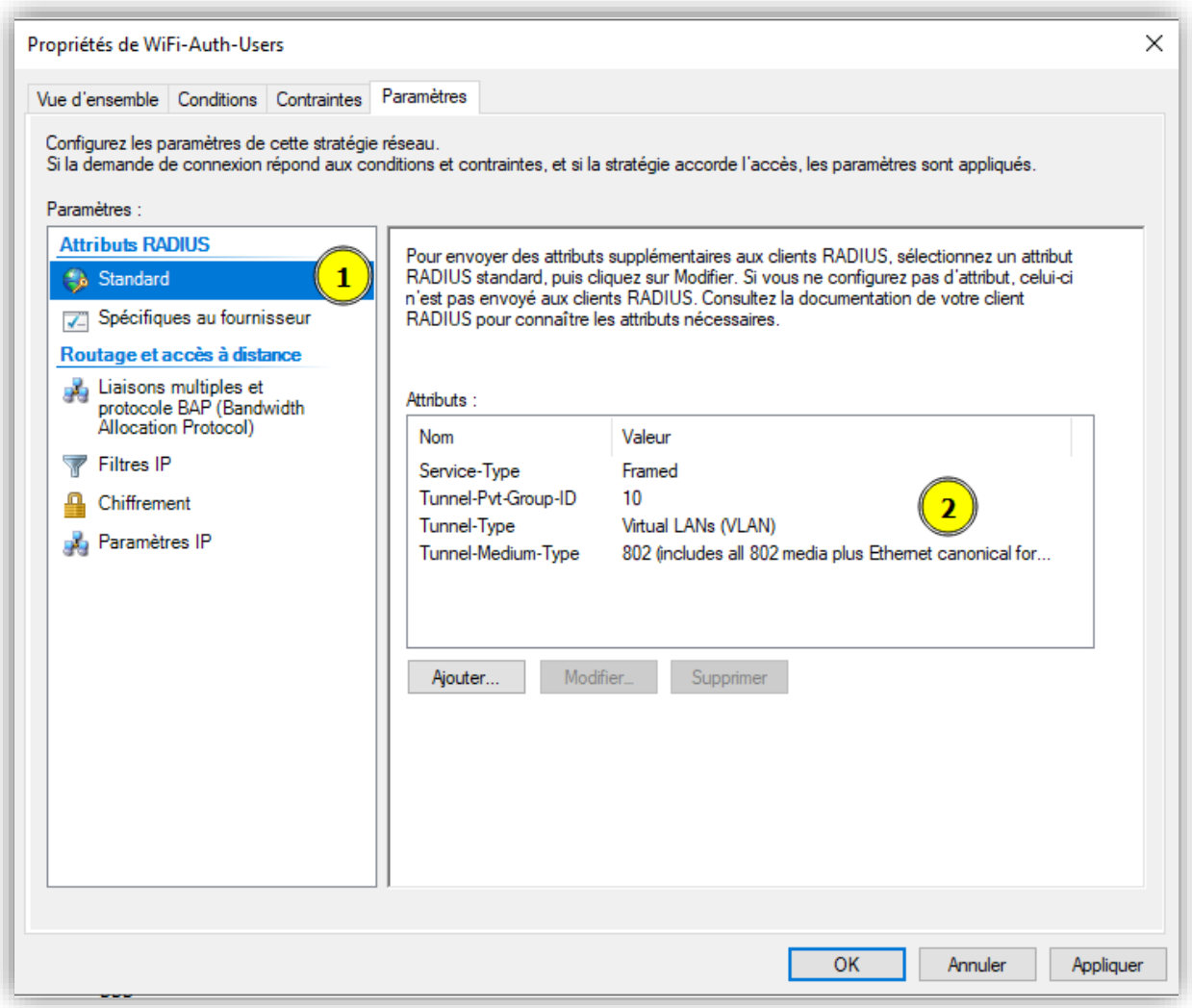
1. Méthode d'authentification
2. Définition de l'authentification via PEAP

Nous choisissons les médias concernés par notre stratégie :



1. Type de port concerné par la stratégie
2. Wifi via la norme 802.11 dans notre cas

Nous pouvons maintenant configurer les attributs Radius pour l'assignation du VLAN de façon dynamique :



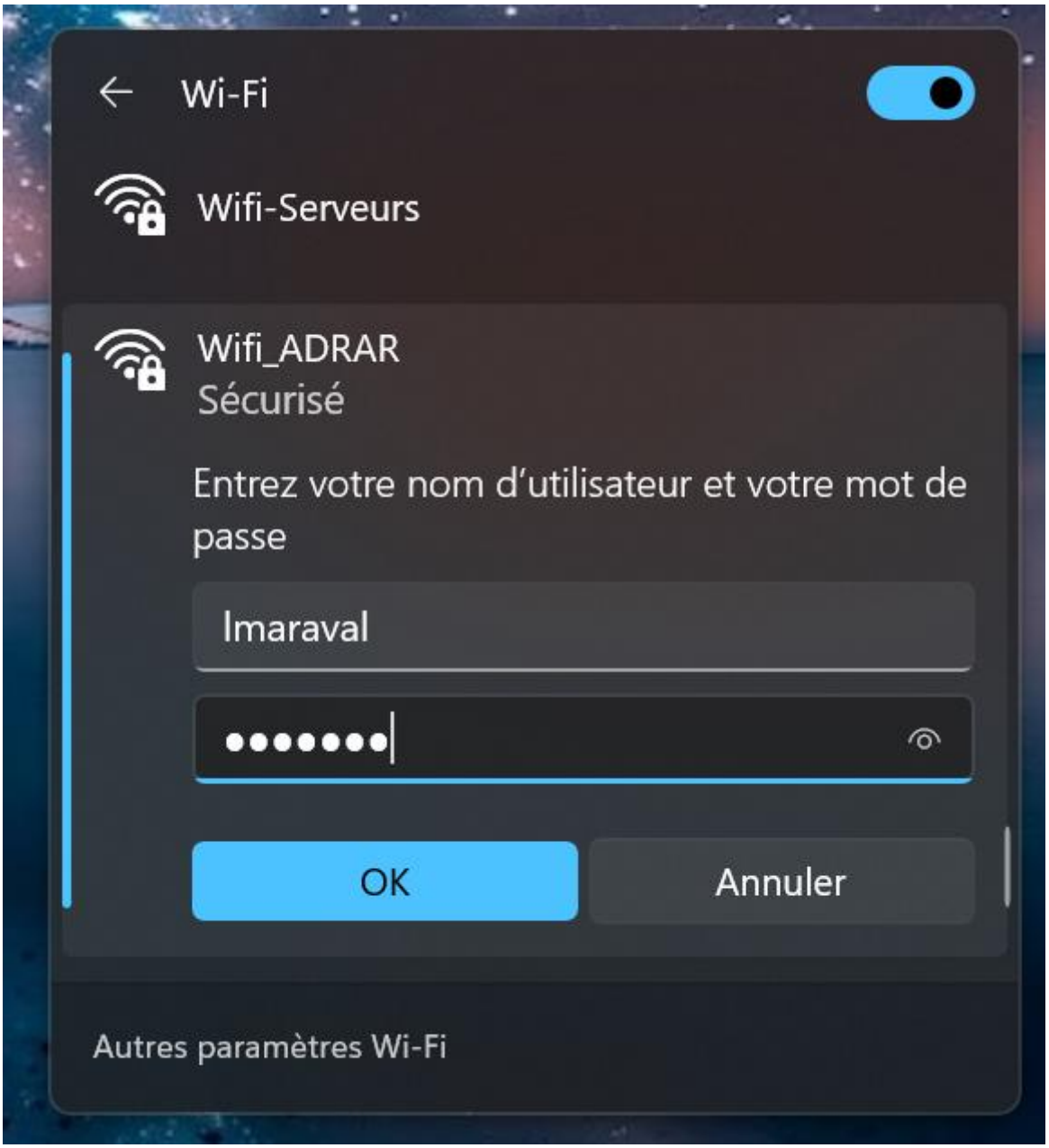
1. Attribut standard à configurer pour les VLANs
2. Attribution du VLAN 10 dès authentification

Nous effectuons les mêmes configurations pour les **« Wifi-Auth-Users-VLAN-20 »** et **« Wifi-Auth-Users-VLAN-30 »** en assignant respectivement les groupes **« GG_Formateurs »** et **« GG_Guest »**

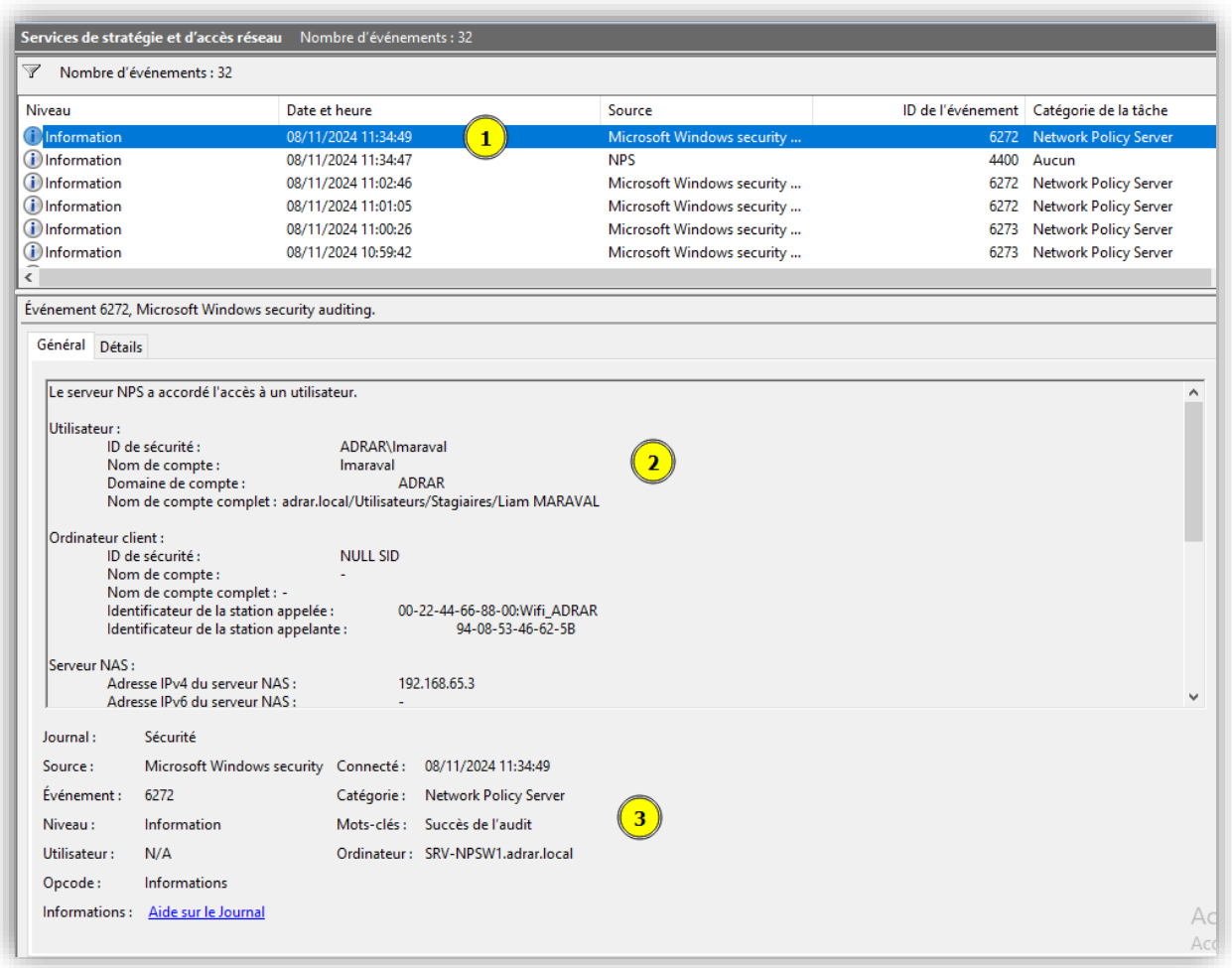
2.3 Test clients Wifi

2.3.1 Test VLAN 10 Stagiaires

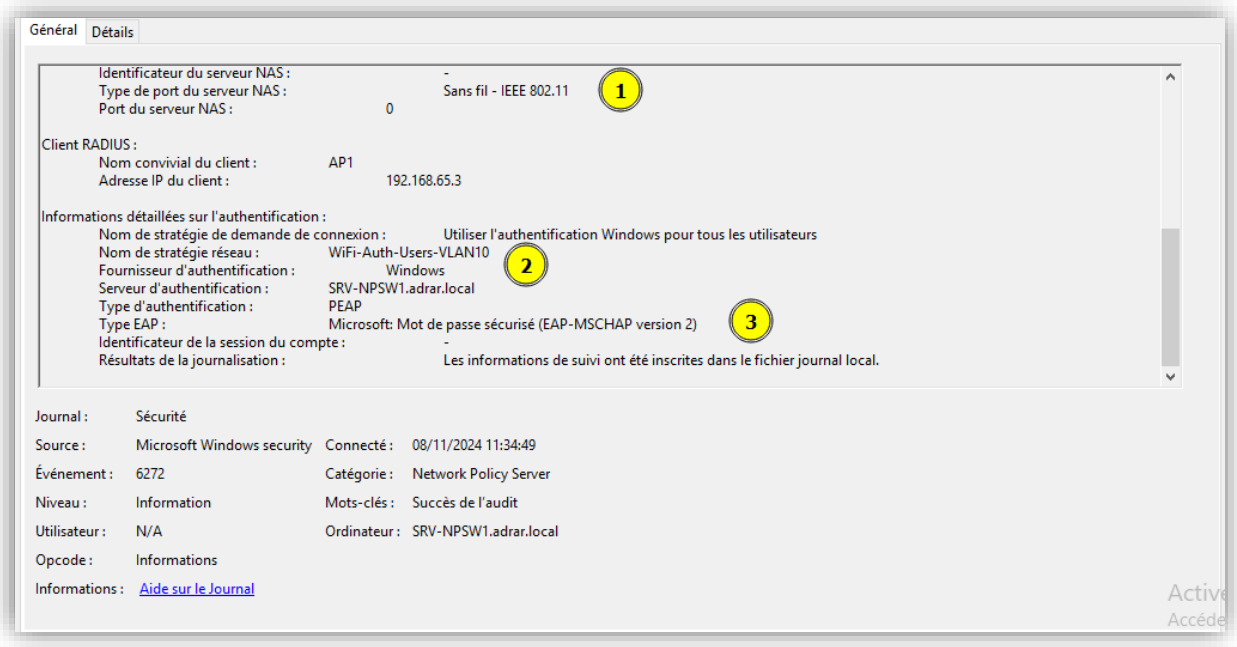
Nous allons maintenant effectuer les tests sur les différents utilisateurs.
Nous nous connectons au « Wifi_Adrar » et renseignons nos identifiants



Nous allons dans l'observateur d'événements NPS afin de valider le fonctionnement de la stratégie :

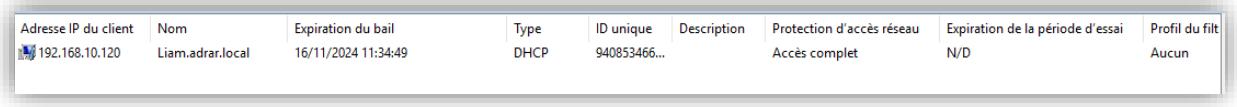


1. Log d'événement sur le NPS
2. Nom d'utilisateur
3. Autorisation de connexion via la stratégie



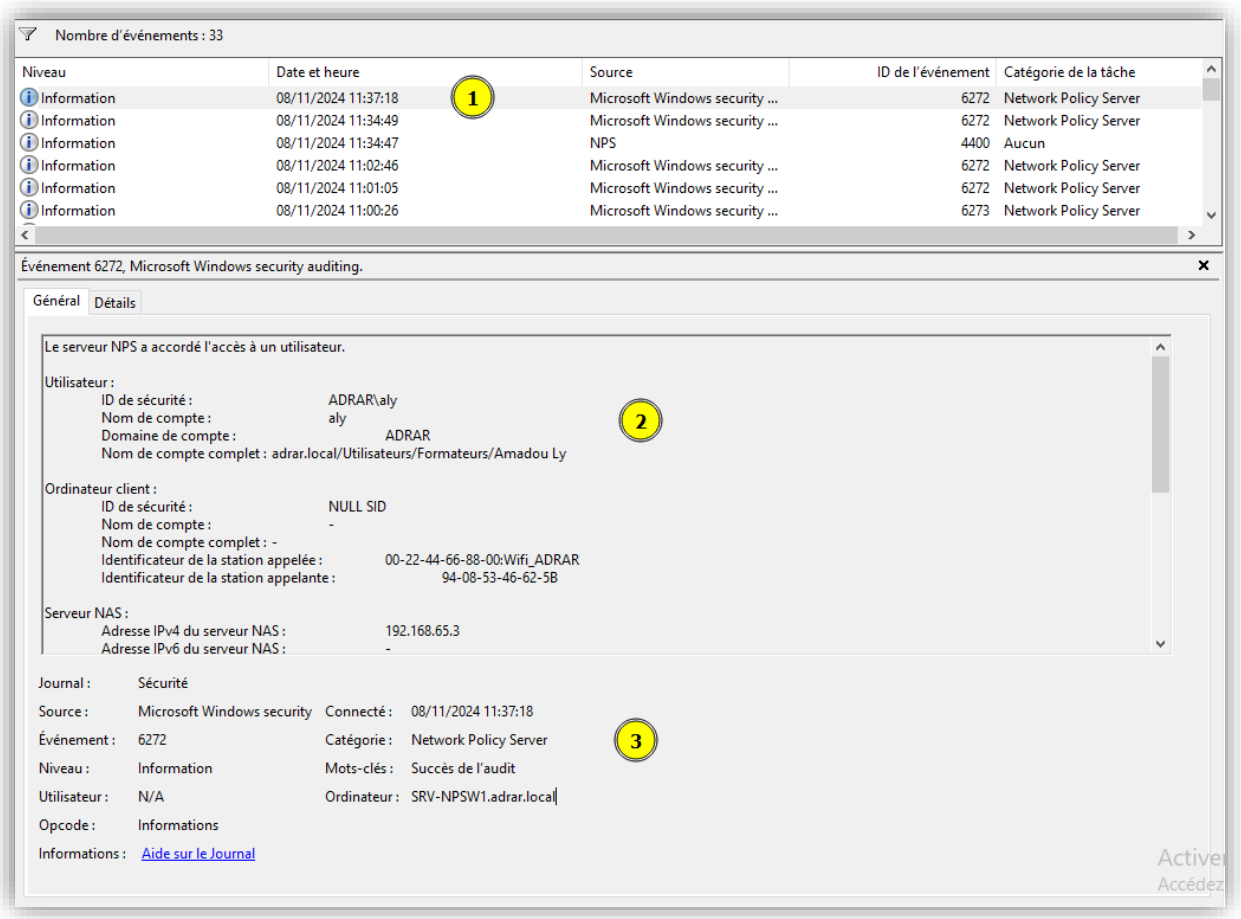
1. Connexion via le Wifi
2. Nom de la stratégie appliquée
3. Méthode d'authentification utilisée

Nous voyons ci-dessous l'IP assignée à notre client

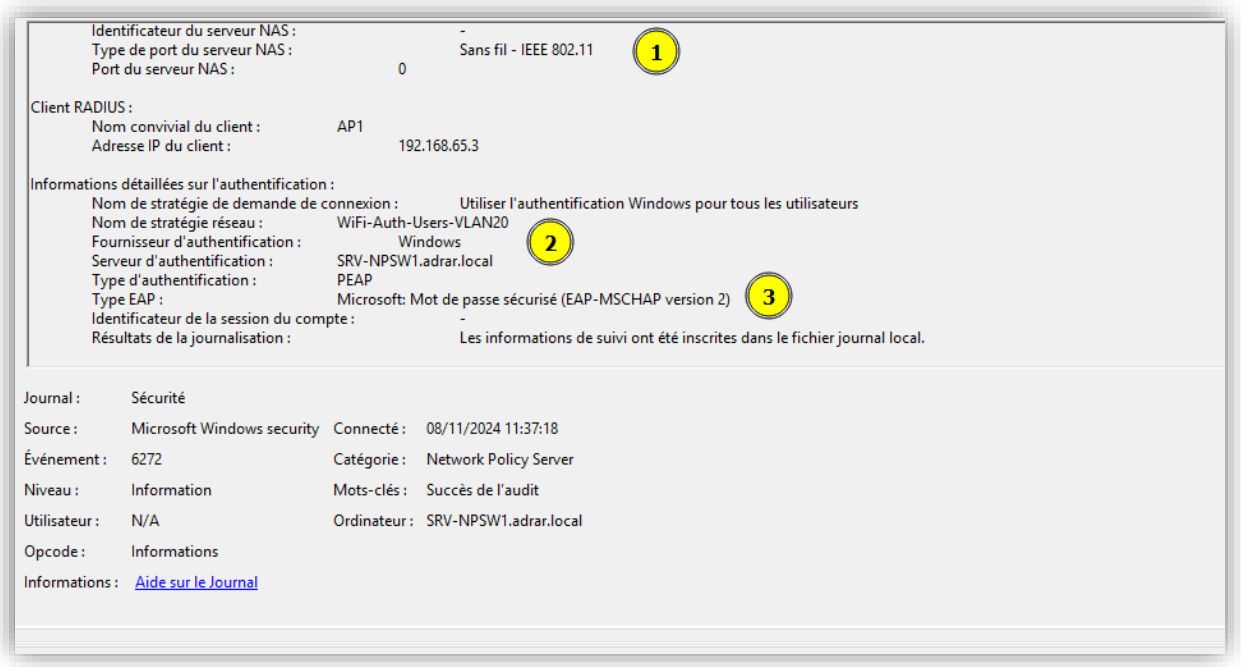


2.3.2 Test VLAN 20 Administratif

Nous effectuons les mêmes étapes de vérification pour le VLAN 20 Administratif mais avec un utilisateur appartenant au groupe **« GG_Formateurs »**

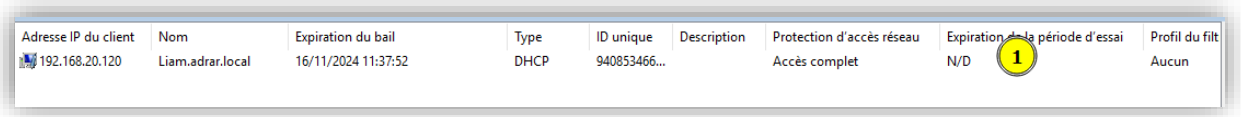


1. Log d'événement sur le NPS
2. Nom d'utilisateur
3. Autorisation de connexion via la stratégie



1. Connexion via le Wifi
2. Nom de la stratégie appliquée
3. Méthode d'authentification utilisé

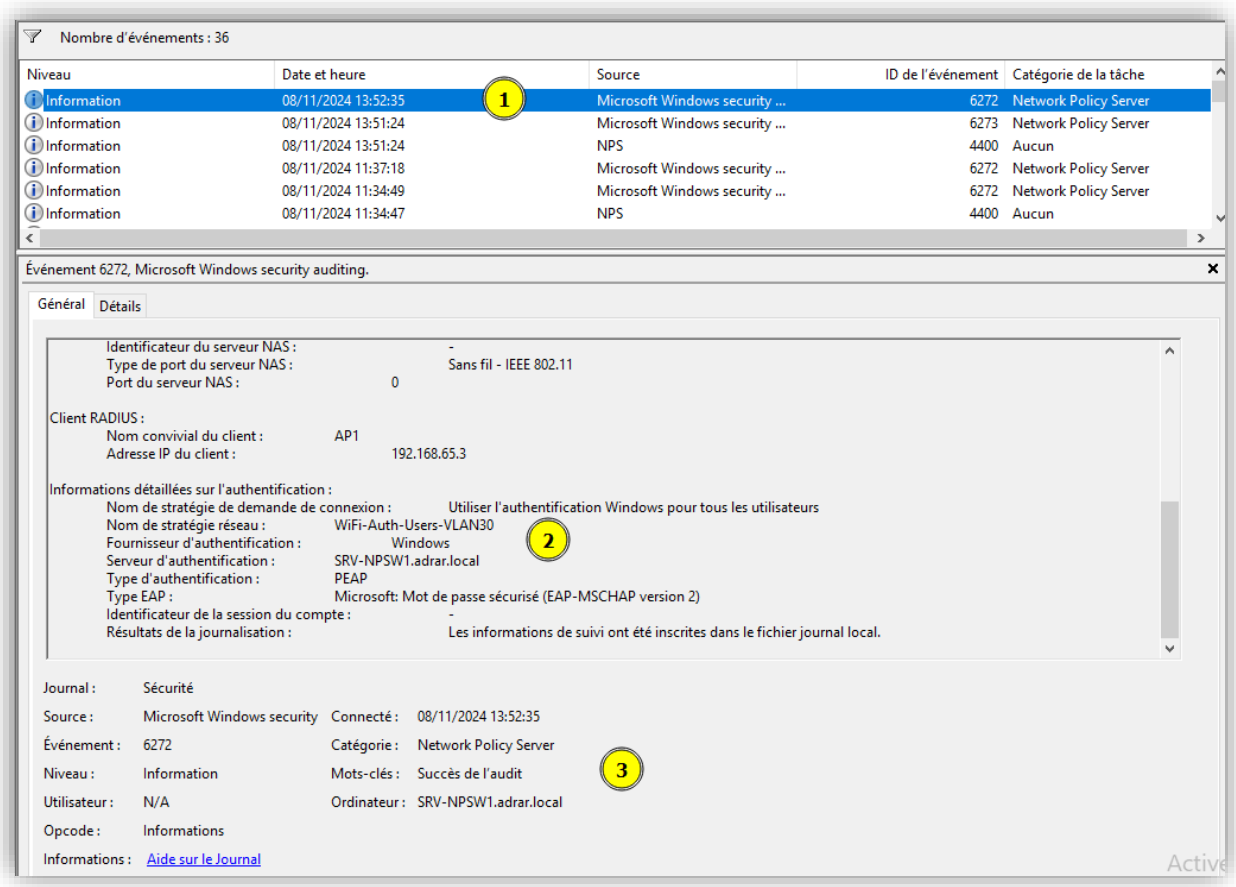
Nous voyons ci-dessous l'IP assigné à notre client



1. IP Assigné au poste

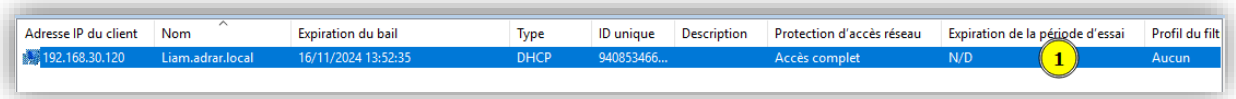
2.3.3 Test VLAN 30 Guest

Nous effectuons les mêmes étapes de vérification pour le VLAN 30 **« Guest »**



1. Log d'événement sur le NPS
2. Nom de la stratégie appliquée
3. Méthode d'authentification utilisé

Nous voyons ci-dessous l'IP assignée à notre client



1. IP assignée au poste

3. Proposition de gestion du VLAN Guest par vouchers

La proposition de solution via vouchers qui suis dans ce document n'est qu'une proposition de développement ! Afin de mettre celle-ci en PROD plusieurs éléments seraient à revoir afin de sécuriser nos applications.
Il faudrait revoir notamment le script Python afin de sécuriser les clés API via un Key Vault par exemple, mettre en place une gestion des erreurs plus élaborée, limiter les accès à notre API à certains utilisateurs ect…

La solution peut aussi être envisagée sans serveur Flask et Apache2, afin de n'utiliser que les vouchers et le portail captif sur l'Opnsense mais les vouchers devront être générés manuellement.

3.1 Contexte

Nous avons vu ci-dessus, la gestion du Wifi Guet en rattachant celui-ci à un VLAN étanche et l'authentification via un utilisateur « Guest » créé sur l'AD.

Une autre solution pour l'authentification consisterait à mettre en place un système de vouchers à attribuer aux utilisateurs externes de l'ADRAR. Ces vouchers donneraient un accès temporaire au réseau et auraient une durée de validité de 24h.

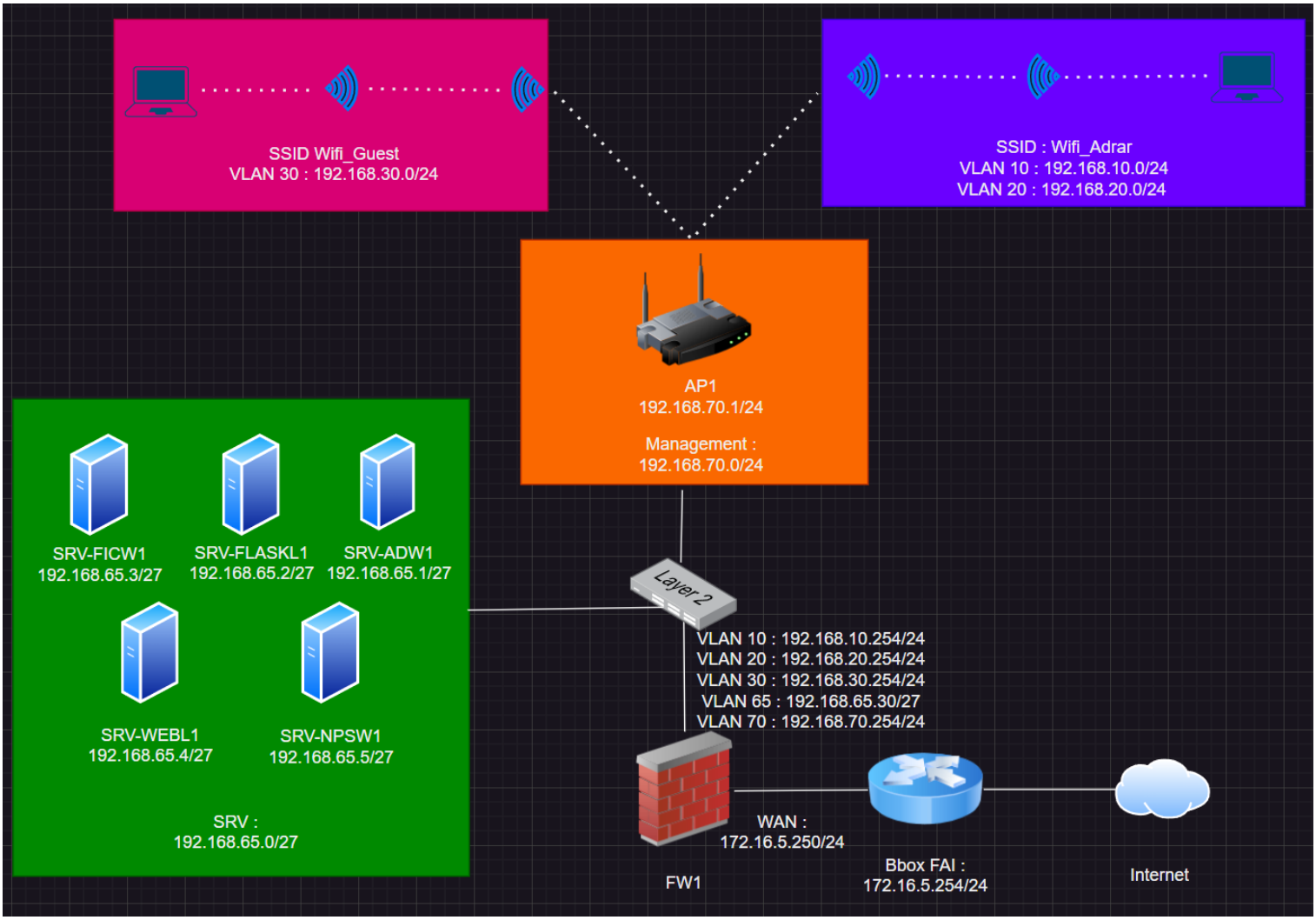
Cette solution nous permettrait de ne pas utiliser de compte utilisateur « Guest » tout en assurant une traçabilité accrue.
En effet, nous pourrions suivre l'attribution des vouchers à un utilisateur externe et consulter les sessions en cours avec les IP, MAC du dispositif se connectant au réseau.

Des points d'accès plus récents proposent déjà nativement ce service mais notre WAP371 en place à l'ADRAR ne nous le permet pas.

Pour cela, nous avons trouvé une solution alternative mais celle-ci pourrait être mise en place plus simplement via l'achat d'un point d'accès plus récent.

3.2 Proposition de solution

Voici le schéma de la nouvelle infrastructure :



Pour notre solution, nous allons rajouter plusieurs éléments :

- Un opnsense nommé FW1
- Un serveur debian 12 nommé SRV-FLASKL1

Le serveur Opnsense va servir pour le filtrage de flux mais aussi de portail captif pour les connexions via l'interface Guest (Wifi ou filaire).
Une fois connecté au réseau, une page s'ouvrira demandant un code d'accès.

Pour cela, nous allons également ajouter un serveur Vouchers sur l'OPNsense, celui-ci sera utilisé afin de générer les vouchers à attribuer aux utilisateurs externes.

Nous avons de plus ajouté quelques éléments à notre solution.
En effet, actuellement les vouchers devant être générés depuis l'OPNsense, cela nécessiterait l'intervention d'un administrateur pour se connecter et générer les codes. Cette solution n'est donc pas satisfaisante car elle rajoute à l'administrateur une tache à effectuer.

Pour cela, nous avons mis en place un serveur Debian qui va être utilisé pour héberger 2 services. Un serveur Apache2 et un service Flask.

Le serveur Apache2 va être utilisé afin de créer une interface Web pour la génération de vouchers. Celle-ci utilisera une API créée via Flask et configurée en Python.
L'API sera utilisée pour envoyer des requêtes POST à notre API Opnsense (native à Opnsense) afin de générer des vouchers de manière autonome.
De ce fait, la personne en charge de l'accueil à l'ADRAR pourra accéder à la page et générer les vouchers en autonomie. Au fil des attributions, l'hôte d'accueil de l'ADRAR pourra tenir à jour un fichier indiquant à quelle personne sont attribués les vouchers.

3.3 Sécurisation des accès

Dans cette nouvelle infrastructure, nous allons garder les règles précédemment définies mais qui seront reporté sur l'OPNSense.
Nous allons cependant effectuer quelques modifications pour le VLAN Guest notamment pour autoriser l'accès au portail captif :

Guest					
Action	Protocol	IP Source	Port source	IP Destination	Port Destination
Pass	TCP	192.168.30.0/24	Any	192.168.30.254/32	53
Pass	TCP	192.168.30.0/24	Any	192.168.30.254/32	8000 - 10000
Block	any	192.168.30.0/24	any	192.168.10.0/24	any
Block	any	192.168.30.0/24	any	192.168.20.0/24	any
Block	any	192.168.30.0/24	any	192.168.65.0/27	any
Block	any	192.168.30.0/24	any	192.168.70.0/24	any
Block	any	192.168.30.0/24	any	192.168.30.254/32	any
Pass	TCP	192.168.30.0/24	any	any	80
Pass	TCP	192.168.30.0/24	Any	any	443

Tous nos serveurs disposeront de certificats afin de chiffrer les échanges via SSL/TLS.

Comme mentionné plus haut, ceci est une proposition de DEV si l'on souhaitait mettre la solution en PROD, plusieurs points seraient à modifier afin de renforcer la sécurité de notre solution.
La solution d'utilisation uniquement via le portail captif et la génération de vouchers manuellement sur l'Opnsense peut également être envisagée afin de simplifier l'infrastructure.

3.4 Mise en place de la solution

Afin de ne pas surcharger la documentation et d'éviter les doublons, nous ne reviendrons pas sur certaines procédures déjà établies pour l'ADRAR dans de précédents projets.

Voici les prérequis techniques nécessaires à la mise en place de notre solution :

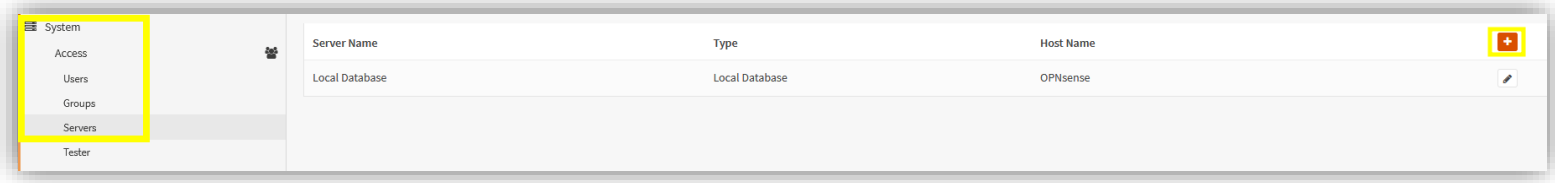
- ✓ Configuration de VLAN sur un Switch Cisco 3700
- ✓ Configuration d'un Pool DHCP sur Switch Cisco 3700
- ✓ Créer des ACL sur un Switch Cisco 3700
- ✓ Configuration de base d'un OpnSense (Nom, IP, VLAN, regle de pare feu ect…)

Pour plus de détails, merci de vous référer aux précédentes documentations rendues à l'ADRAR.

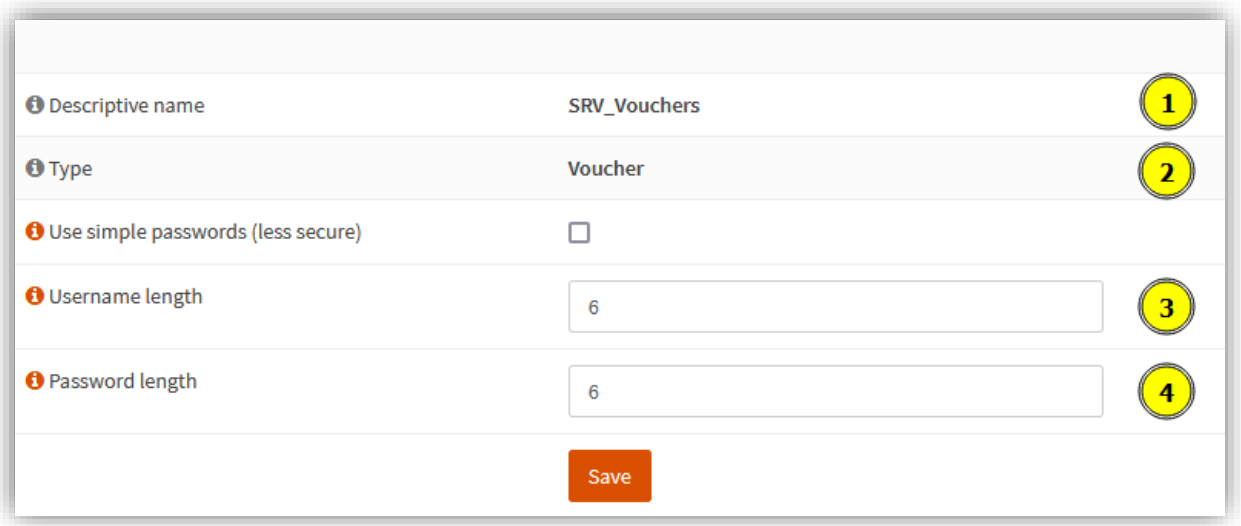
3.4.1 Configuration de l'OPNSense

Nous commençons par les configurations de base comme le nom, l'attribution des IP aux VLANs, nom de domaine ect…

Ensuite, nous allons créer notre serveur Vouchers dans le menu suivant :

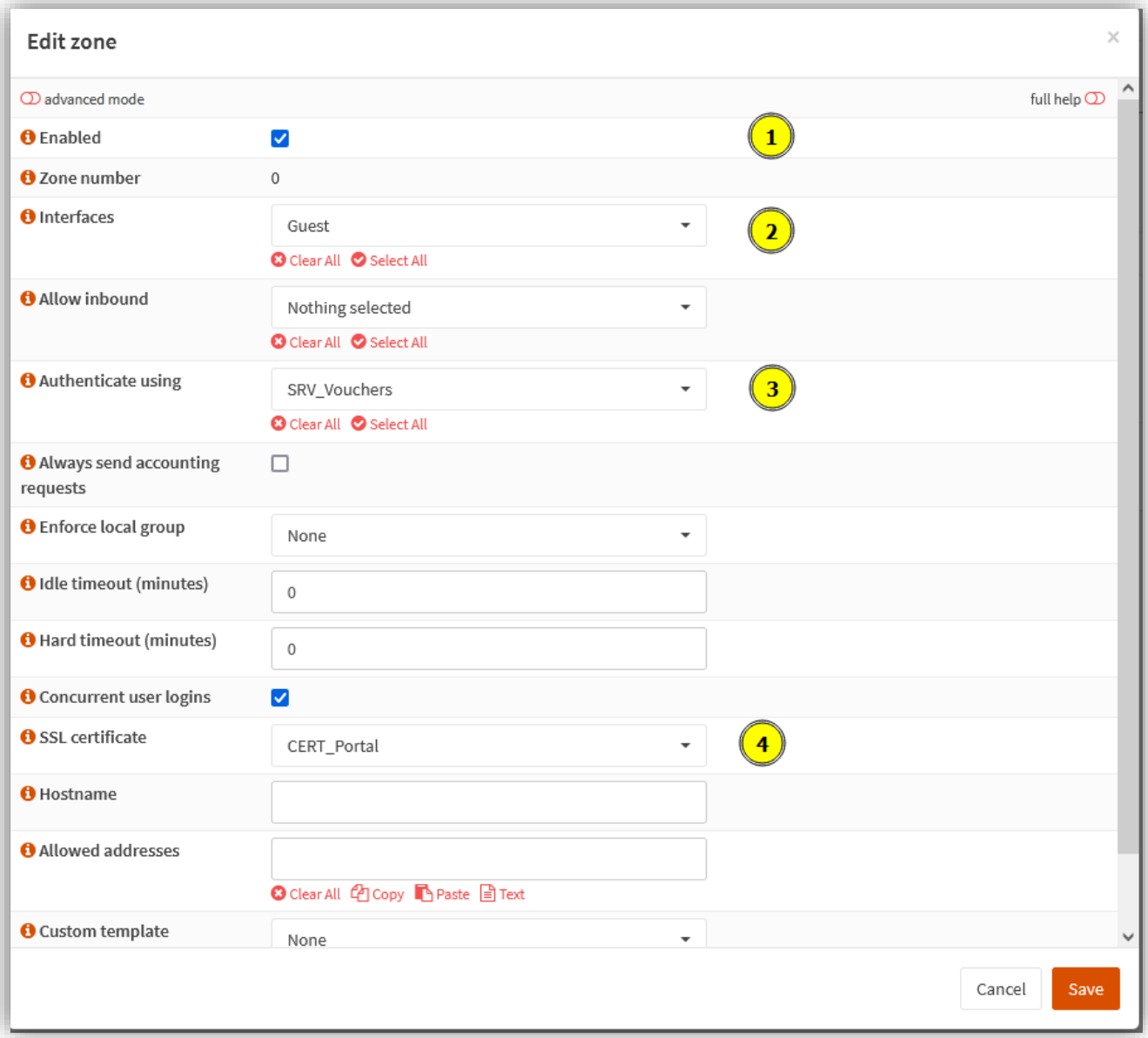


Nous renseignons nos paramètres :



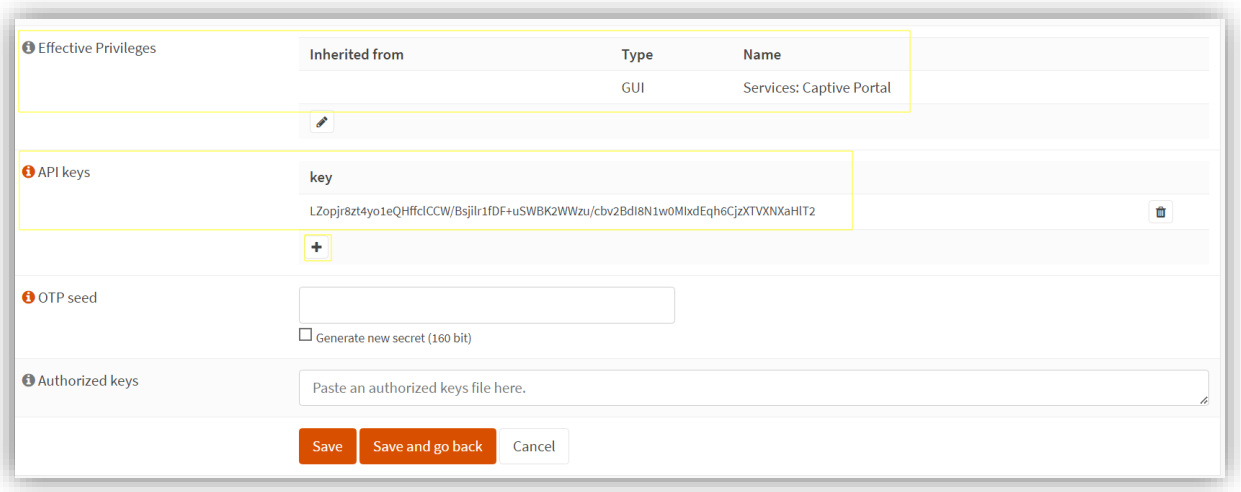
1. Nom du serveur
2. Type de serveur
3. Longueur du nom d'utilisateur
4. Longueur du mot de passe

Maintenant nous allons créer notre portail captif dans le menu suivant :



1. Nous activons le portail
2. L'assignons pour le VLAN **« Guest »**
3. Utilisation du serveur Vouchers pour l'authentification
4. Certificat SSL du portail

Une fois cela fait nous créons un utilisateur local sur l'Opnsense.
Nous pouvons maintenant générer notre clé API sur la page de l'utilisateur comme ceci :



Nous voyons ici la clé API, le secret associé est renseigné dans un fichier Excel téléchargé sur le poste en local lors de la génération (Ce secret n'est pas conservé en local sur le pare feu, il est donc important de bien le renseigner dans un Key Vault ou Key Pass)

3.4.2 Configuration du service Flask

Nous allons maintenant configurer Flask via un script Pyhton.
Ce script utilise le Framework Flask pour créer une API et générer des vouchers de manière automatisée.

```
from flask import Flask, request, jsonify
from flask_cors import CORS  # Importer le module CORS
import requests
from requests.auth import HTTPBasicAuth

app = Flask(__name__)

OPNSENSE_HOST = "https://192.168.30.254"
API_KEY = "API_KEY"
API_SECRET = "API_Secret"
PROVIDER = "SRV_Vouchers"
VALIDITY = 24

@app.route('/generate-vouchers', methods=['POST'])
def generate_vouchers():
    count = request.json.get('count', 1)  # Nombre de vouchers à générer

    url =
f"{OPNSENSE_HOST}/api/captiveportal/voucher/generateVouchers/{PROVIDER}"
    payload = {
        "provider": PROVIDER,
        "count": count,
        "validity": VALIDITY
    }

    try:
        response = requests.post(
            url,
            json=payload,
            auth=HTTPBasicAuth(API_KEY, API_SECRET),
            verify=False
        )

        if response.status_code == 200:
            return jsonify({"status": "success", "vouchers":
response.json()})
        else:
            return jsonify({"status": "error", "message": response.text}),
response.status_code
    except requests.exceptions.RequestException as e:
        return jsonify({"status": "error", "message": str(e)}), 500

if __name__ == '__main__':
    app.run(host='192.168.65.2', port=5000)
```

3.4.3 Configuration du serveur Apache2

Nous avons configuré un site Web « adrar.local » en HTTPS avec un certificat assigné au serveur.

Voici la page Web mise en place via le code .html suivant

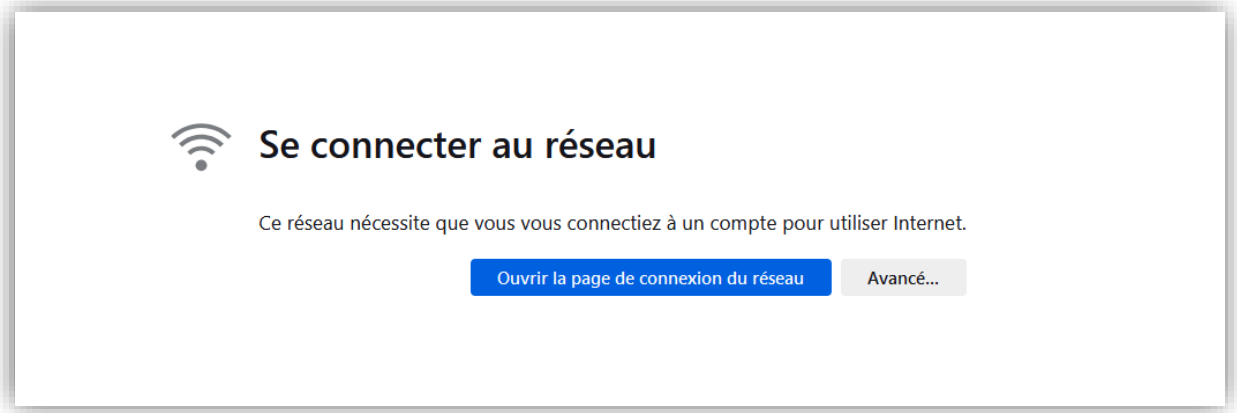
```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Générateur de Vouchers</title>
    <script>
        function generateVouchers() {
            const count = document.getElementById('count').value;

            fetch('http://192.168.65.2:5000/generate-vouchers', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({ count: count })
            })
            .then(response => response.json())
            .then(data => {
                if (data.status === 'success') {
                    let vouchers = data.vouchers;
                    let output = '<h2>Vouchers générés :</h2><ul>';
                    vouchers.forEach(voucher => {

                        output += `<li>Username: ${voucher.username} |
Password: ${voucher.password}</li>`;
                    });
                    output += '</ul>';
                    document.getElementById('vouchers').innerHTML = output;
                } else {
                    document.getElementById('vouchers').innerHTML =
`<p>Error: ${data.message}</p>`;
                }
            })
            .catch(error => {
                document.getElementById('vouchers').innerHTML = `<p>Error:
${error}</p>`;
            });
        }
    </script>
</head>
<body>
    <h1>Générateur de Vouchers</h1>
    <label for="count">Nombre de vouchers :</label>
    <input type="number" id="count" value="1" min="1">
    <button onclick="generateVouchers()">Générer des vouchers</button>
    <div id="vouchers"></div>
</body>
</html>
```

3.4.4 Test de la solution

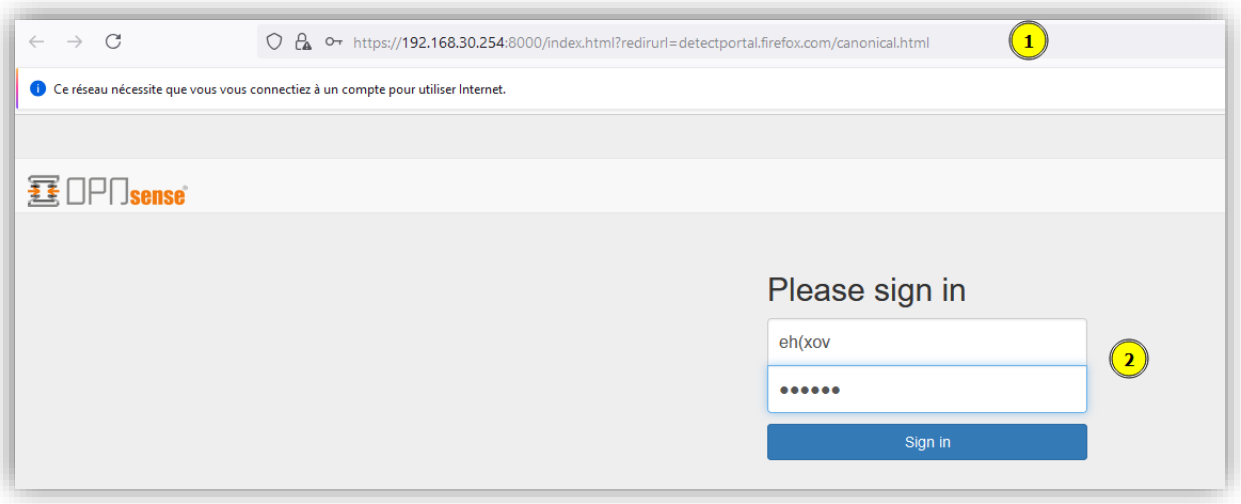
Nous allons maintenant effectuer nos tests.
Nous commençons par nous connecter à notre page Web et générer les vouchers :

1. IP du serveur Web
2. Nombre de vouchers à générer
3. Voucher généré à attribuer aux utilisateurs externes

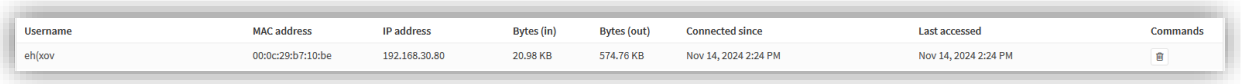
Maintenant que notre voucher est généré via notre site Web, nous allons nous rendre sur un poste connecté au VLAN 30 **Guest**.
Nous ouvrons le navigateur et voyons la page suivante :



Nous arrivons sur cette page et renseignons notre voucher généré précédemment :



Nous retrouvons maintenant la session de notre utilisateur :



4. Conclusion

En conclusion, ce projet de refonte du réseau Wi-Fi de l'ADRAR répond aux exigences de sécurité modernes et aux recommandations de l'ANSSI. La segmentation en VLANs distincts, couplée à l'authentification 802.1X via le serveur RADIUS, permet une gestion granulaire des accès et une traçabilité complète des connexions. Bien que la solution actuelle pour le Wi-Fi invité repose sur un compte "Guest" en raison des limitations matérielles, une évolution vers un système de vouchers via des équipements plus récents (Cisco Meraki, Ubiquiti UniFi ou Aruba Instant On) est recommandée pour l'avenir. L'architecture proposée offre un bon compromis entre sécurité, facilité de gestion et contraintes budgétaires, tout en restant évolutive pour les besoins futurs de l'établissement.

5. Annexes

5.1 Documentations et références

Configuration WAP 371: https://techexpert.tips/fr/access-point-cisco-fr/configuration-initiale-de-cisco-wap371/

Configuration SSID WAP371: https://www.cisco.com/c/fr_ca/support/docs/smb/wireless/cisco-small-business-100-series-wireless-access-points/smb5072-configuring-a-vap-on-the-wap351-wap131-and-wap371.html

Configuration serveur NPS : https://blog.matrixpost.net/set-up-a-radius-server-on-windows-server-2019-for-802-1x-wireless-connections/

Attribution de VLAN dynamique : https://www.expertnetworkconsultant.com/installing-and-configuring-network-devices/ieee-802-1x-authentication-and-dynamic-vlan-assignment-with-nps-radius-server/

Configuration du Wifi Guest via portail Opnsense : https://docs.opnsense.org/manual/how-tos/guestnet.html

Recommandation de l'ANSSI sur le Wifi : https://cyber.gouv.fr/publications/securiser-les-acces-wi-fi