

SUJET WORKSHOP 3ÈME ANNÉE :

“REPARTIR À ZÉRO”

Les membres de l’équipe :

- GALLEY Hugo DevOps
- FLAHAUT Axel DevOps
- BESSE Thomas Dev IA
- FERNANDES Damien SysOps
- MÉRESSE Enzo SysOps

Tables des Matières

Introduction/Contexte.....3

Présentation du sujet.....3

Les contraintes.....4

Présentation de la solution.....5

Pratique Technique.....8

Conclusion.....14

Annexes.....17

Introduction / Contexte

Ce document s'inscrit dans le cadre du Workshop EPSI 2025-2026, dont l'objectif est de confronter les étudiants à une mise en situation concrète de conception et de déploiement d'une solution technique.
L'enjeu n'est pas seulement de mettre en place une infrastructure fonctionnelle, mais aussi de travailler en équipe, d'organiser les tâches et de livrer une solution documentée, sécurisée et opérationnelle.

Notre groupe a choisi d'orienter son projet vers la conception d'un intranet sécurisé, accessible uniquement depuis un réseau privé et protégé par des mécanismes de sécurité logiques (réseau, firewall, VLAN) et physiques (contrôle d'accès RFID).

Présentation du sujet

Dans le scénario proposé pour ce workshop, nous sommes projetés dans un univers où l'intelligence artificielle Ultron a déclenché une guerre numérique.
Son objectif : détruire tous les outils de communication et d'organisation utilisés par les humains. Réseaux compromis, serveurs corrompus, messageries réduites au silence… l'infrastructure classique ne tient plus.

Dans ce contexte de chaos numérique, notre mission était claire : reconstruire un système de communication interne, indépendant d'Internet, et résilient face aux attaques.
Nous avons donc imaginé et déployé une infrastructure intranet sécurisée, capable d'assurer à la fois :

- La sécurité logique, via une segmentation réseau (pfSense, VLAN, firewall) garantissant que les administrateurs et les collaborateurs disposent chacun d'un espace contrôlé
- La sécurité physique, via la construction d'une salle serveur protégée par un contrôle d'accès RFID.

Cette approche nous permet de proposer une solution robuste, où la survie de l'organisation ne dépend plus des réseaux mondiaux, mais d'un intranet privé, hébergé et protégé localement.

Les contraintes

Notre projet a donc été guidé par plusieurs contraintes :

Techniques :

- Utilisation de solutions Open Source.
- Interdiction d'utiliser des services Cloud controlés par Ultron.
- Priorité au mode offline afin de fonctionner en cas de panne Internet.

Fonctionnelles :

- La solution devait répondre à un besoin précis : permettre aux utilisateurs de continuer à communiquer via une messagerie interne et d'accéder à des ressources documentaires essentielles.
- Elle devait rester simple et accessible à tous, grâce à une interface claire et intuitive.

La problématique : Dans un contexte où les infrastructures mondiales sont détruites par Ultron et où la dépendance à Internet n'est plus envisageable, comment concevoir un intranet local capable de garantir la continuité des échanges et la protection des données essentielles, tout en restant simple d'utilisation et réplicable ?

Présentation de la solution

Objectif de la solution :

Face à la menace d'Ultron qui a détruit les outils de communication et de coordination, notre objectif était de concevoir un intranet autonome, sécurisé et accessible localement, permettant aux communautés de :

- Communiquer via une messagerie interne,
- Partager et consulter des ressources documentaires essentielles,
- Protéger physiquement le serveur hébergeant les données.

Notre solution ne dépend pas d'Internet ni de services centralisés compromis, ce qui lui permet d'être résiliente et durable dans un contexte hostile.

En combinant sécurité logique et physique, notre solution assure la résilience des communautés face à Ultron et leur permet de rester unies pour préparer la contre-attaque.

Le fonctionnement de la solution :

Ce projet de workshop combine autant la sécurité physique que la sécurité logique :

La sécurité physique :

- Conception et impression 3D d'une "salle serveur miniature".
- Intégration d'un système RFID basé sur Arduino qui permet de contrôler l'accès à la salle serveur.
- Lorsque le bon badge est scanné, la porte s'ouvre ; sinon la porte reste verrouillée. Cette protection empêche les accès non autorisés au Raspberry Pi qui contient les services critiques.

La sécurité logique :

- PfSense agit comme routeur et firewall :
    - Il nous permet de distribuer les IP avec son serveur DHCP intégré.
    - Segmentation VLAN.
    - Règles de filtrage appliquées (les admins accèdent à tous les équipements et les collaborateurs seulement la messagerie).
- Un switch HPE 1820 qui assure la gestion des vlans et la communication entre les équipements.
- Une borne Wi-Fi Planet qui diffuse deux réseaux distincts :

    - ADMIN_WS : accès total
    - COLLABORATEURS_WS : accès restreint
- Le Raspberry Pi qui héberge :
    - Notre messagerie interne (*Phantom*)
    - Notre intranet documentaire (Phantom Security)

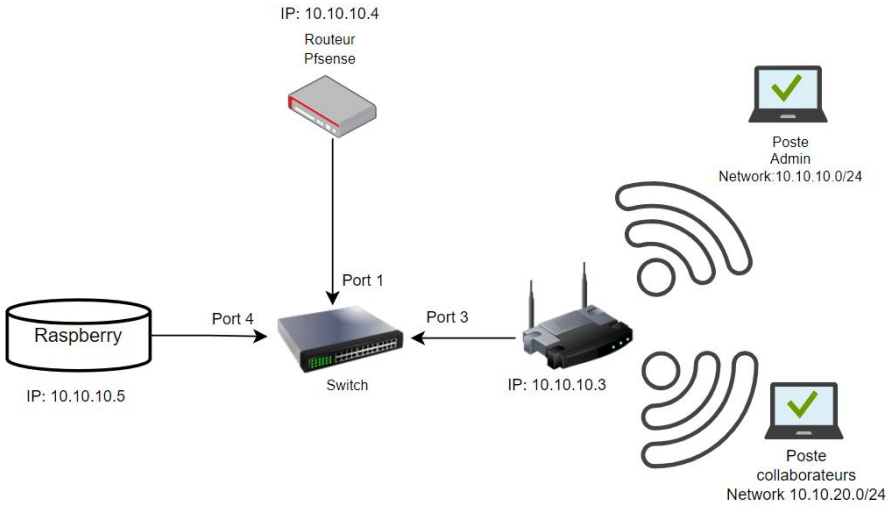
Le processus de conception et le matériel utilisé :

Matériel : Raspberry Pi 400 / borne Wi-Fi Planet WNAP-1120PE / Switch HPE 1820 Series / routeur pfSense / Arduino avec module RFID / imprimante 3D Creality K2 Plus / Matériel de bricolage

Logiciels et ressources Open Source : Draw.io / messagerie Phantom / librairies Arduino / CapCut / Nginx /pfSense

Méthodologie :

- Conception du schéma réseau
- Conception de la salle serveur
- Paramétrage et configuration du RFID
- Installation et configuration du Raspberry (messagerie + intranet)
- Installation et configuration des équipements réseaux (borne/routeur/switch)
- Tests de connectivité
- Réalisation des documents de rendu (présentation + rapport).



Légende 1 : Schéma de l'infrastructure réseau

Nous avons développé une application de messagerie Phantom. Voici une courte description de celle-ci **:**

Phantom est une application de messagerie chiffrée de bout en bout développée en Python, conçue pour assurer la confidentialité et la sécurité des échanges. Elle repose sur un backend construit avec FastAPI pour gérer les communications entre le frontend (React/Vite) et une base de données MySQL, tout en intégrant des mécanismes avancés de chiffrement comme AES-GCM pour protéger les messages et RSA pour l'échange sécurisé des clés. Côté client, IndexedDB permet un stockage local persistant, tandis que la communication en temps quasi-réel est assurée grâce à une technique de long polling. L'ensemble est pensé pour être facilement déployable via Docker, garantissant une mise en place rapide et portable.

Partie technique

Le code Arduino et RFID :



```
void loop() {

  if (RFID.isCard()) {
    if (RFID.readCardSerial()) {
      for (int i = 0; i < 5; i++) {
        UID[i] = RFID.serNum[i];
      }
    }

    if (UID[0] == MASTERKEY[0] &&
        UID[1] == MASTERKEY[1] &&
        UID[2] == MASTERKEY[2] &&
        UID[3] == MASTERKEY[3] &&
        UID[4] == MASTERKEY[4])
    {
      if (!estOuvert) {
        ouverture();
        estOuvert = true;
      }
    } else {
      if (!estOuvert) {
        mauvaiseCarte();
      }
    }

    RFID.halt();
    delay(500);
  }

  if (isClicked()) {
    fermeture();
  }
```

Légende 2 : code de la boucle principale de l'ouverture de la salle serveur

Le programme gère un système d'ouverture/fermeture à l'aide d'une carte RFID et d'un bouton :

- **Lecture de la carte RFID :**
  - Le code vérifie en permanence si une carte est détectée.
  - Si une carte est présente, il lit son identifiant unique (UID).
- **Vérification de l'UID :**
  - L'UID de la carte est comparé avec une clé maître enregistrée dans le programme (MASTERKEY).
  - Si les deux correspondent, l'accès est autorisé.
- **Gestion de l'ouverture/fermeture :**
  - Si la carte est correcte et que la porte n'est pas déjà ouverte, la fonction ouverture() est appelée.
  - Si la carte est incorrecte et que la porte est fermée, la fonction mauvaiseCarte() est exécutée.
  - Un bouton (via isClicked()) permet également de déclencher la fonction fermeture().

- **Temporisation :**

- De petits délais (delay) évitent les lectures trop rapides ou répétitives.

Le code Nginx avec l'intranet :

Objectif :

- Fournir un point d'accès centralisé aux ressources et services internes.
- Donner rapidement un accès rapide à notre messagerie sécurisée Phantom.
- Héberger une bibliothèque locale contenant manuels, procédures et documentation.
- Garantir un fonctionnement même sans connexion internet.

Configuration du serveur :

Nous avons utilisé un Raspberry Pi en serveur, installé Nginx comme serveur Web. Un vhost principal sert le portail sur l'adresse http://10.10.10.5

Page D'accueil de l'intranet :

- La page d'accueil est minimaliste et comporte deux gros boutons :

- Messagerie sécurisée → ouvre Phantom via un proxy Nginx sur le port 8080.

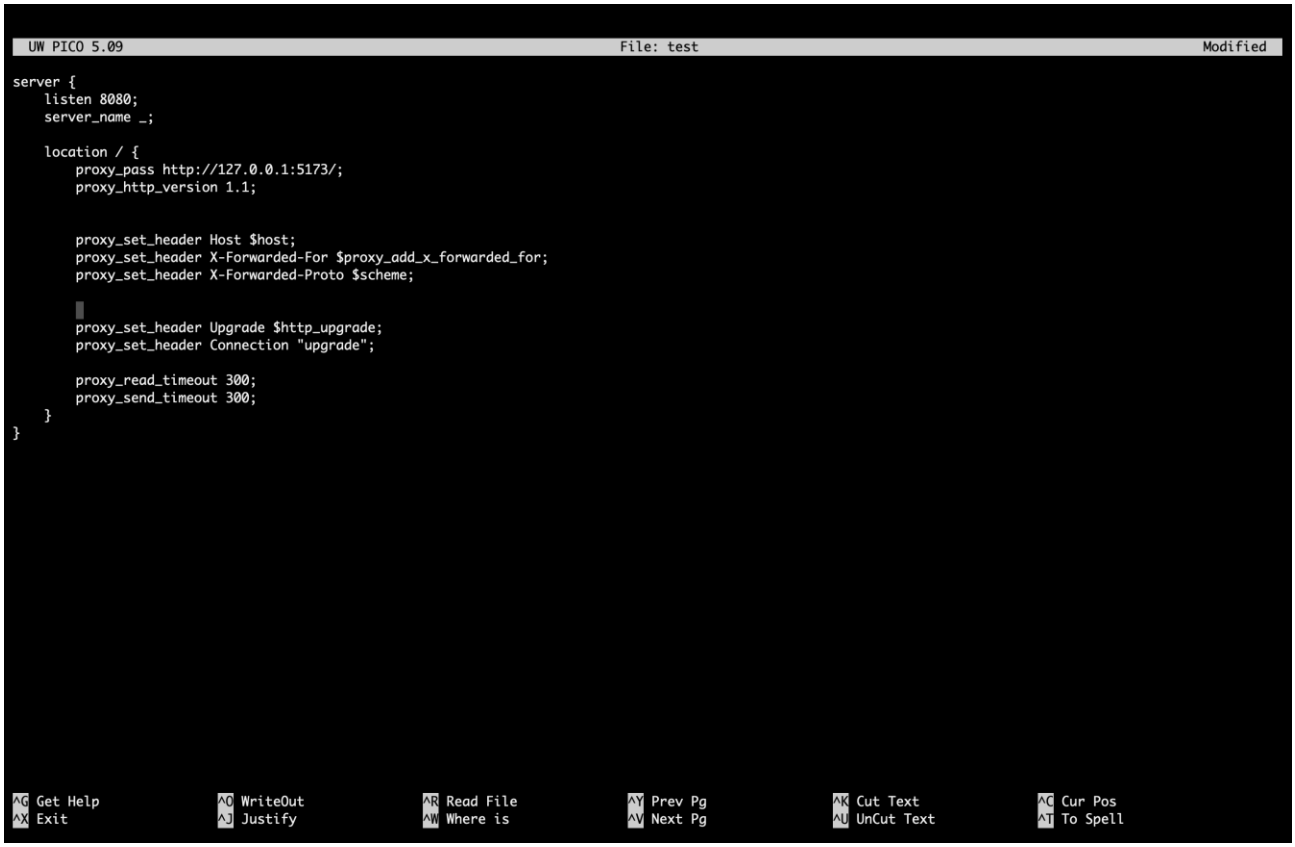
- Bibliothèque locale → redirige vers /docs/ pour accéder aux manuels et procédures.

Intégration de Phantom :

Nous avons ajouté un vhost Nginx dédié sur le port 8080 qui redirige tout le trafic vers le conteneur Docker Phantom (en HTTP sur 5173). Cela permet de centraliser l'accès à la messagerie via l'IP du serveur, sans connaître le port interne.

Extrait de configuration Nginx

Exemple du fichier /etc/nginx/sites-available/phantom-8080:



Légende 3 : Usage du reverse proxy pour que Docker soit exposer sur le réseau local

6. Résultat Final :

Accès unique : tout utilisateur du réseau peut aller sur http://10.10.10.5 et trouver :

La messagerie Phantom,

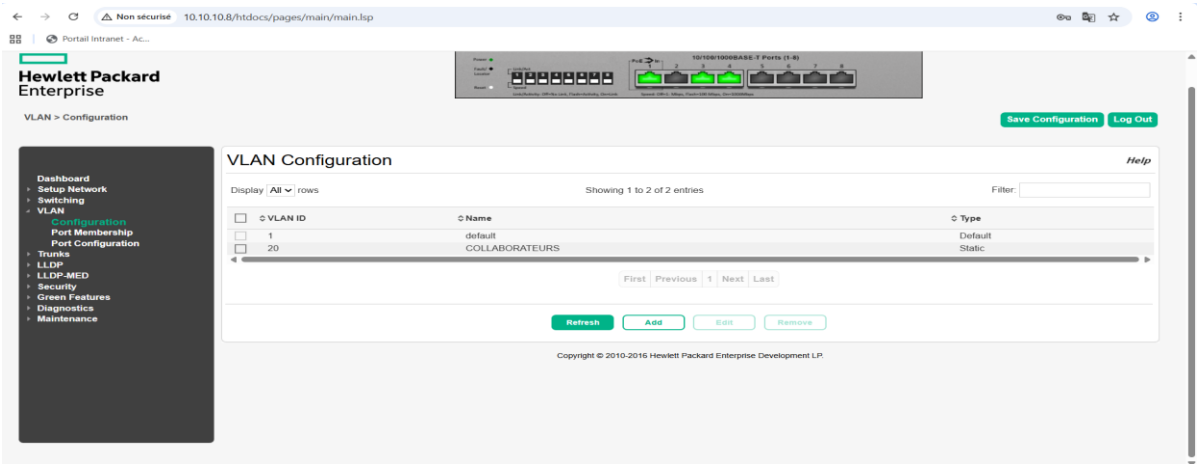
La bibliothèque locale avec documentation,

Et une page d'accueil claire.

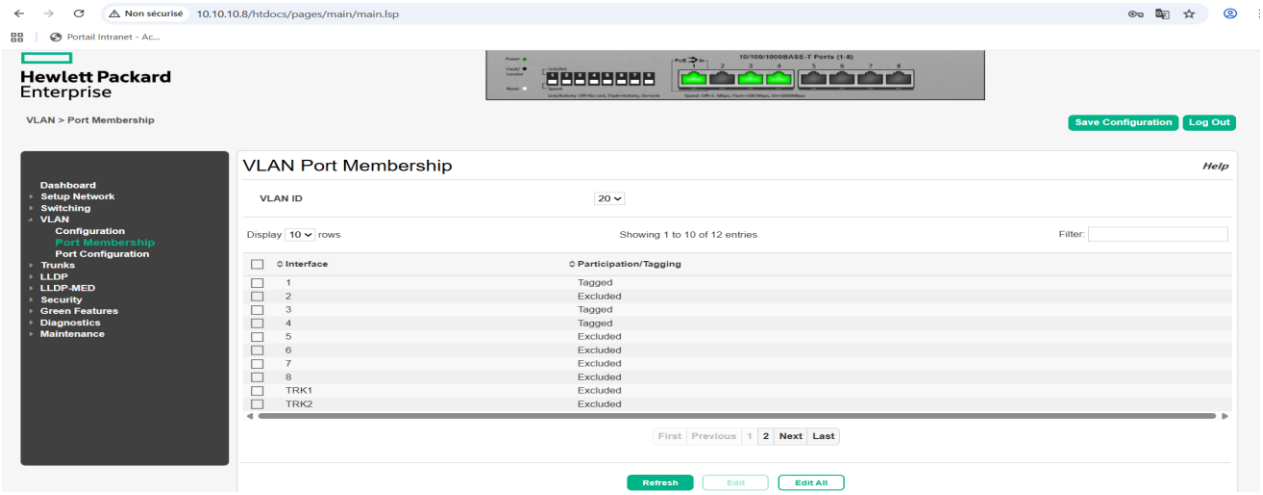
Résilience : même sans Internet, tout fonctionne car le contenu est statique et servi localement.

Séparation propre : Phantom reste dans son conteneur, mais est accessible de manière transparente via Nginx.

Configurations du switch :

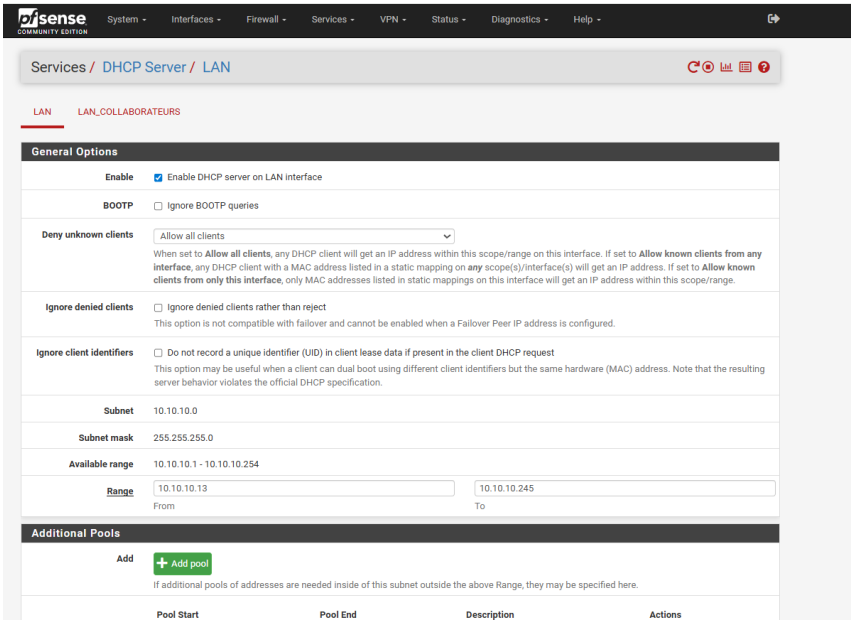


Légende 4 : Configuration des VLAN sur le switch

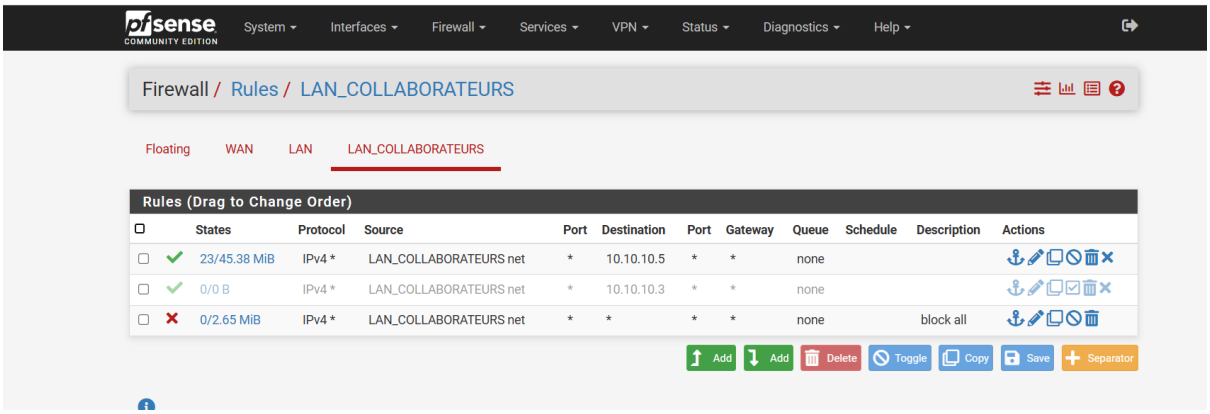


Légende 5 : Configuration du VLAN 20 sur les ports du switch

Configurations du pfSense :

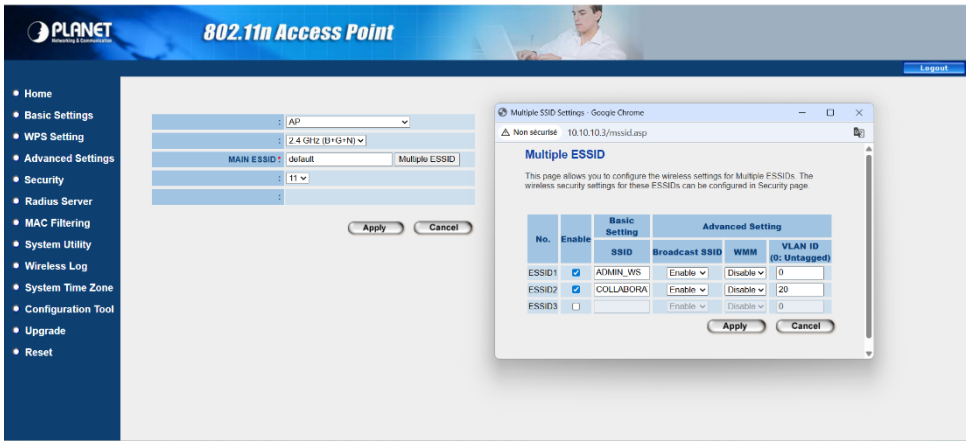


Légende 6 : Activation du DHCP pour les admins sur pfSense

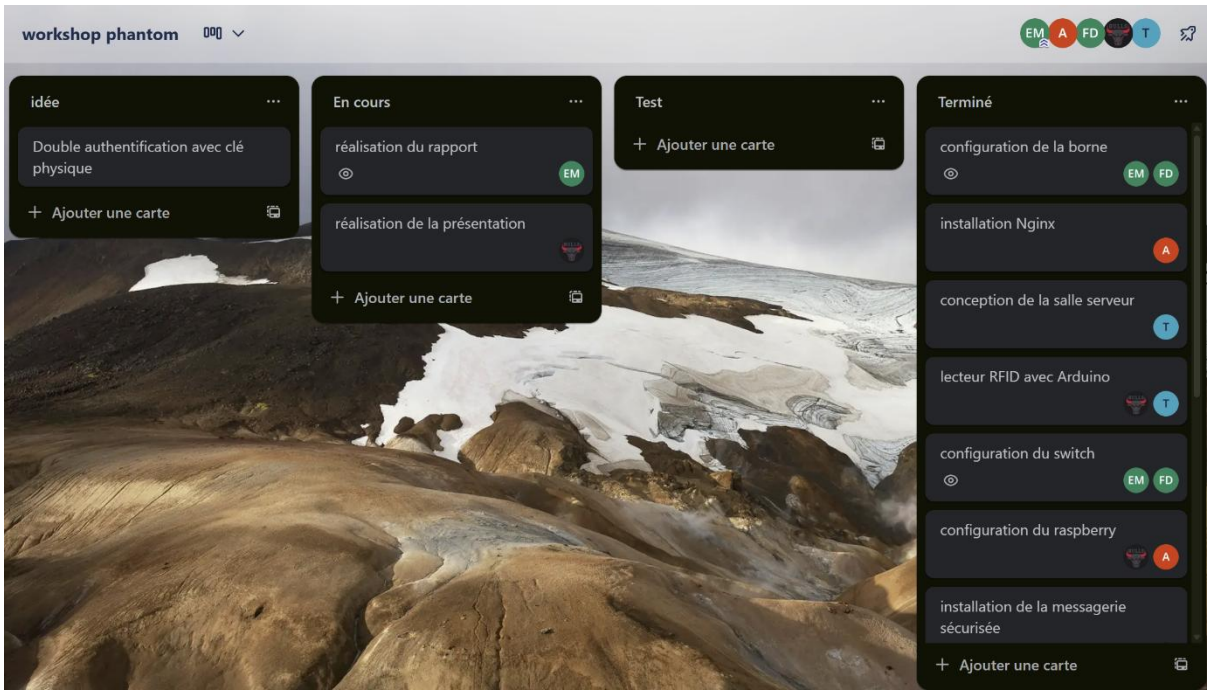


Légende 7 : Règle firewall qui empêche le passage du réseau 10.10.20.0/24 sur les appareils cruciaux

Configuration de la borne Wi-Fi :



Légende 8 : Configuration des accès de la borne wifi



Légende 9 : capture d'écran de notre organisation sur Trello

Lien Github du projet Phantom : https://github.com/Hugo-Galley/Phantom

Conclusion

Les perspectives de la solution :

Notre solution, bien que fonctionnelle et sécurisée, peut être renforcée et améliorée afin d'augmenter sa résilience et son niveau de sécurité.

Tout d'abord, il serait pertinent de mettre en place une redondance du serveur, en utilisant plusieurs Raspberry Pi synchronisés. Cela permettrait d'éviter tout point de défaillance unique et d'assurer une continuité de service même en cas de panne matérielle.

Ensuite, l'ajout d'une alimentation de secours garantirait le maintien des services essentiels en cas de coupure électrique, ce qui est crucial dans un contexte où la stabilité énergétique n'est pas toujours assurée.

Du côté réseau, l'intégration d'un système de détection d'intrusion (IDS/IPS) renforcerait la sécurité en permettant d'identifier et de bloquer rapidement toute tentative d'attaque. De plus, une supervision centralisée à l'aide d'outils tels que Zabbix ou Grafana offrirait une meilleure visibilité sur l'état du réseau et des services, facilitant ainsi la maintenance et l'anticipation des incidents.

Enfin, sur le plan applicatif, nous pourrions implémenter une authentification renforcée pour les utilisateurs, avec par exemple une double authentification reposant sur une clé physique (comme une YubiKey). Cette mesure limiterait fortement les risques d'accès non autorisés aux ressources critiques. Nous avons tenté d'implémenter cette idée mais celle-ci n'a pas pu être réalisée par manque de temps.

Ces évolutions permettraient d'élever encore le niveau de robustesse et de sécurité de notre solution, la rendant plus adaptée à un environnement hostile où la menace d'Ultron reste constante.

Les difficultés rencontrées :

Au cours de ce projet, nous avons été confrontés à plusieurs difficultés, aussi bien sur la partie réseau que sur le développement. Ces obstacles ont nécessité des recherches, des tests répétés et parfois des ajustements matériels ou logiciels pour parvenir à une solution fonctionnelle.

Difficultés réseau :

La première difficulté concernait la gestion du DHCP. Au départ, les clients connectés en Wi-Fi recevaient des adresses IP directement depuis la borne. Après migration vers pfSense pour centraliser la distribution, les postes ne recevaient plus d'adresse et basculaient en APIPA. La résolution a été de configurer correctement la borne en mode point d'accès, en laissant pfSense comme unique serveur DHCP.

La deuxième difficulté portait sur la segmentation réseau avec VLANs. La configuration du switch HPE 1820 et de la borne Wi-Fi n'était pas intuitive. Une mauvaise gestion du tagging/untagging empêchait les clients Collaborateurs d'accéder à leur passerelle, bien qu'ils reçoivent une adresse IP valide. Ce problème a été résolu en ajustant les ports trunk et access afin d'assurer une communication cohérente entre pfSense, le switch et la borne.

Limitations matérielles :

Nous avons également rencontré des contraintes matérielles. Certains équipements disponibles n'étaient pas suffisants pour couvrir tous nos besoins, tandis que d'autres présentaient des dysfonctionnements. Cela a ralenti nos phases de tests et nous a obligés à adapter notre infrastructure au matériel réellement fonctionnel.

Difficultés de développement :

Lors du déploiement de l'application sur le Raspberry Pi, plusieurs problèmes techniques se sont posés :

- HTTPS obligatoire : initialement, notre site n'était accessible qu'en HTTP, ce qui bloquait certains accès. Nous avons donc dû mettre en place le HTTPS afin de sécuriser les communications et poursuivre le développement.
- Problème de casse dans Docker : le conteneur du backend ne se déployait pas car il cherchait un dossier avec une mauvaise casse (majuscule/minuscule). Après correction des chemins, le déploiement a pu s'effectuer correctement.
- Conflits réseau avec Docker et ETH0 : nous avons eu des difficultés liées aux connexions multiples. L'interface ETH0 bloquait parfois le Wi-Fi, ce qui empêchait certains packages Python de se télécharger correctement. En désactivant l'interface au bon moment, nous avons pu contourner ce problème et finaliser l'installation.

Les apports de la semaine :

Cette semaine de workshop nous a permis de développer à la fois nos compétences techniques et notre travail en équipe. Sur le plan technique, nous avons acquis une meilleure maîtrise de la configuration réseau (pfSense, VLANs, switchs, bornes Wi-Fi), du déploiement de services sur un Raspberry Pi, et de l'intégration matérielle avec Arduino et RFID. Nous avons également appris à résoudre des problèmes concrets en conditions réelles, en adaptant nos choix aux contraintes matérielles et logicielles.

Sur le plan humain, ce projet nous a appris à collaborer efficacement, à répartir les rôles entre la partie réseau et développement, et à communiquer pour surmonter les difficultés. Enfin, nous avons pu renforcer notre créativité en combinant sécurité physique et logique dans une solution cohérente et innovante.

ANNEXES



Légende 10 : Installation du réseau et salle serveur



Légende 11 : Equipe 1 du Workshop 2025-2026