PROGRAMME DE FORMATION BACHELOR CYB3R XP

Cahier des Charges de la MSPR « Développement et sécurité informatique »

COMPÉTENCES ÉVALUÉES :

- Concevoir et maintenir des scripts d'acquisition et de persistance des données
- Exploiter des données collectées
- Concevoir et déployer une plateforme centralisée de consultation des données
- Établir une communication sécurisée au sein d'un réseau local
- Assurer la continuité d'un service en mode déconnecté
- Sécuriser les échanges entre sites
- Appliquer les fondamentaux de la cybersécurité en environnements systèmes & réseaux
- Mettre en place une journalisation et une observabilité efficaces
- Communiquer et défendre une proposition technique en contexte client

PHASE 1 : PRÉPARATION DE CETTE MISE EN SITUATION PROFESSIONNELLE RECONSTITUÉE

- ***Durée de préparation*** *: 54 heures*
- ***Mise en œuvre*** *: Travail d'équipe constituée de 3 apprenants*

PHASE 2 : PRÉSENTATION ORALE COLLECTIVE + ENTRETIEN COLLECTIF

- ***Objectif*** *: mettre en avant et démontrer que les compétences visées par ce bloc sont bien acquises*
- ***Moyen*** *: L'équipe utilise un support de présentation*
- ***Durée*** *totale par groupe : 50 mn se décomposant comme suit :*
  - *20 mn de soutenance orale par l'équipe.*
  - *30 mn d'entretien collectif avec le jury (questionnement complémentaire).*
- ***Jury d'évaluation*** *: 2 personnes (binôme d'évaluateurs) par jury*

I - CONTEXTE

NFL IT – National Football League Information Technology est une société de services numériques (SSII) créée au début des années 2020 par deux associés français. Elle fournit des prestations d'infogérance, d'audit, de conseil et d'intégration orientées infrastructures informatiques pour des organisations multisites à forts enjeux de disponibilité.

Historiquement, NFL IT s'est spécialisée dans l'accompagnement des équipes de football américain (gestion d'équipes, statistiques, applicatifs métiers), domaine que les fondateurs connaissent bien pour y avoir exercé comme développeurs chez l'éditeur de référence du secteur.

Depuis 2020, NFL IT est titulaire d'un contrat d'exclusivité de plusieurs millions de dollars avec la National Football League (NFL), qui regroupe 32 franchises. L'entreprise compte plus de 100 collaborateurs (environ 30 techniciens/ingénieurs itinérants, 30 personnels support, et ~40 fonctions supports). Le siège est situé à Kansas City (Missouri) ; les clients sont répartis sur tout le territoire américain.

Pour ses besoins d'hébergement, NFL IT s'appuie également sur un datacenter en France (Roubaix) pour certaines applications clients et internes. Dans la continuité de cette stratégie, l'entreprise initie le programme "Seahawks monitoring", destiné à réduire les interventions sur site, industrialiser la supervision et améliorer la maintenabilité du parc chez les franchises, avec une perspective d'extension vers la ligue européenne (ELF).

Enjeu business : aujourd'hui, l'absence d'outils unifiés de maintenance et de visibilité entraîne des déplacements coûteux et des délais de résolution élevés. NFL IT porte donc une feuille de route visant à standardiser la collecte d'informations techniques, centraliser la supervision, et outiller le support pour intervenir à distance lorsque c'est possible.

II – CAHIER DES CHARGES

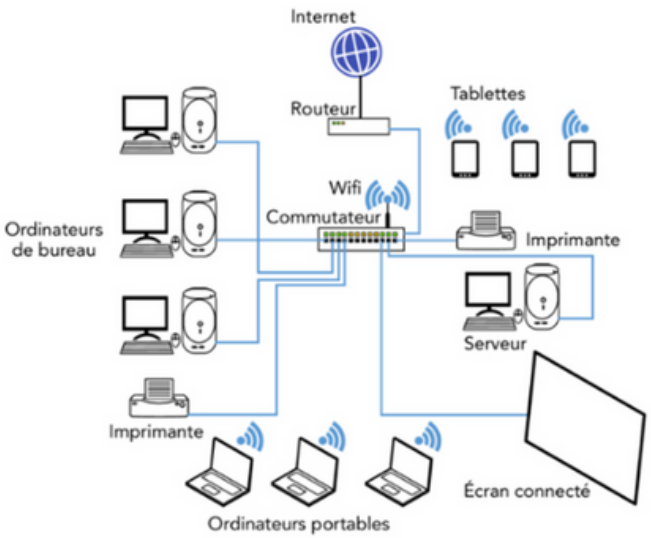
Seahawks Monitoring a pour objectif de standardiser la remontée d'informations essentielles depuis chaque franchise et de fournir au support une vue d'ensemble fiable. L'objectif métier est de réduire les interventions sur site, d'accélérer les diagnostics de niveau 1/2 et d'améliorer l'expérience des équipes locales en limitant les interruptions et les échanges itératifs.

Pour qui ?

- Bénéficiaires principaux : équipes Support N1/N2 et ingénieurs systèmes/réseaux de NFL IT.
- Parties prenantes côté client : Direction des Opérations (pilotage), Responsable Support (priorisation), Référents techniques des franchises (relais terrain).

Cette première étape vise à mettre en place un dispositif unifié de collecte et de consultation des informations de base nécessaires au support (IP/ID hôte, nb équipements détectés, synthèse dernier scan, latence moyenne, version…), avec un cadre d'exploitation simple et facilement déployable sur l'ensemble des sites. Elle constitue le socle du programme : des évolutions (télémaintenance, haute disponibilité, intégrations avancées) pourront être envisagées dans des phases ultérieures.

Exemple de la structure du réseau local d'une franchise



À l'issue de cette phase, NFL IT doit disposer d'un moyen opérationnel pour :
- **Voir rapidement l'essentiel** de la situation technique d'un site,
- **Prioriser les interventions** à distance ou sur place,

III – LES BESOINS EXPRIMÉS

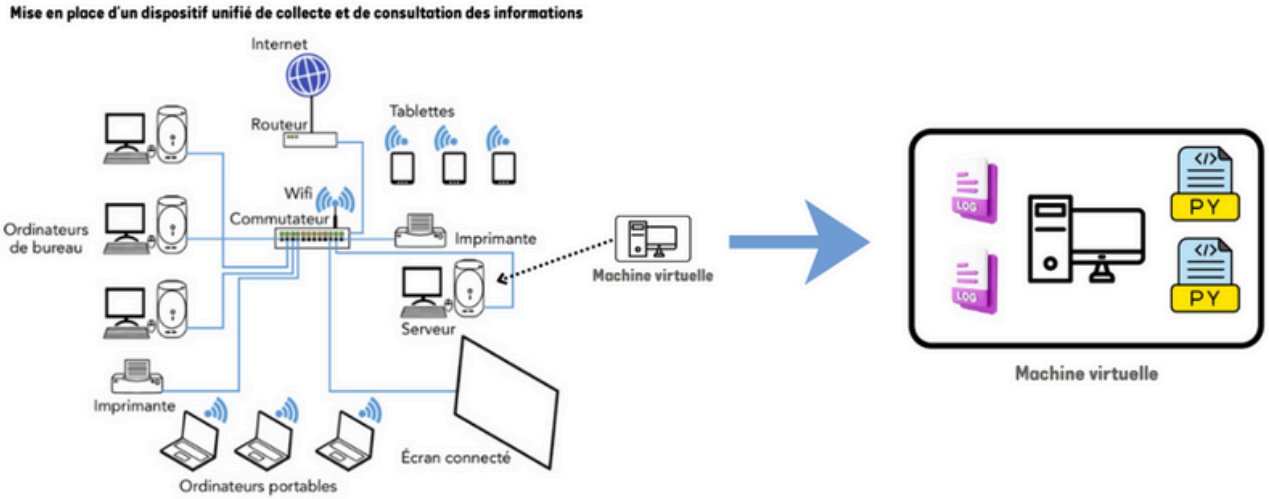
1) Objectifs attendus (métier & SI)

- Réduire les déplacements sur site en fournissant au support une visibilité minimale standardisée sur chaque franchise.
- Accélérer les diagnostics N1/N2 par une collecte régulière d'informations techniques pertinentes.
- Disposer d'un point de consultation central permettant de visualiser rapidement l'état d'un site.

2) Périmètre fonctionnel (ce que doit faire la solution)

2.1 Côté site (franchise) – « Seahawks Harvester »

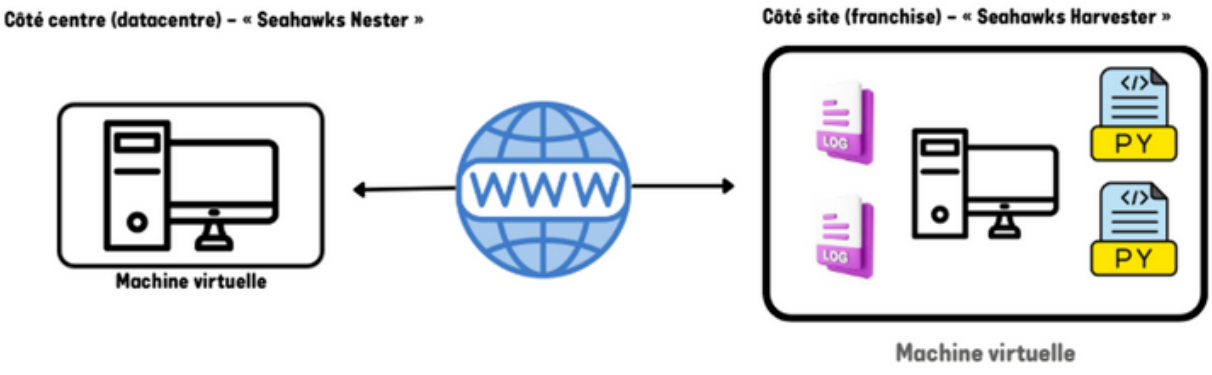
- Collecte réseau raisonnée depuis la franchise (découverte d'hôtes/ports et mesures simples de connectivité). L'usage de scripts Python est attendu, avec possibilité d'utiliser la librairie python-nmap.



- Tableau de bord local simple affichant a minima : IP/nom de la VM, nombre d'équipements détectés, résultat du dernier scan et latence WAN moyenne, ainsi que la version du composant.

2.2 Côté centre (datacentre) – « Seahawks Nester »

- Mise à disposition d'une application de consultation (unique, hébergée au datacentre de Roubaix) permettant : liste des sondes, état connecté/déconnecté, tableau de bord d'une sonde, accès au dernier rapport.



Remarque de périmètre : la prise en main à distance et la haute disponibilité relèvent d'autres lots MSPR (non demandés ici).

3) Contraintes fonctionnelles

- Simplicité d'usage pour techniciens N1/N2 : interface locale sobre et claire ; la complexité est cachée par des scripts.
- Autonomie locale : le Harvester fonctionne même sans connexion permanente au serveur ; la dernière mesure reste consultable localement.
- Échanges Harvester→Nester : format structuré

4) Contraintes techniques

- Site : exécution prioritaire dans une VM Linux (ou conteneur), OS au choix ; Python 3.x requis.
- Non-inclus dans ce lot : tunnel sécurisé, prise de main distante, gestion avancée des utilisateurs, CRUD de sondes.

5) Contraintes opérationnelles

- Déploiement simple et reproductible (VM clonable/déplaçable entre hyperviseurs).
- Évolutivité du programme : la solution constitue un socle pour des fonctionnalités futures (télémaintenance, HA).

6) Exigences de sécurité (socle à intégrer dès cette phase)

Niveau opérationnel attendu, compatible avec un environnement scripts Python / VM.

- Principe du moindre privilège sur les composants déployés (exécution non-root lorsque possible).
- Gestion de la configuration et des secrets : pas de mots de passe en clair dans les scripts ; stockage externalisé ou chiffré ; rotation simple documentée.
- Intégrité & traçabilité : chaque rapport et binaire est identifiable (horodatage, version) ; journaux structurés (événements, scans, erreurs) pour diagnostic.
- Échanges : usage d'un format structuré ; la conception du dialogue et des modalités d'échange est formalisée (dépôt de fichiers).
- Éthique & périmètre : collectes limitées aux environnements de lab/franchise autorisés ; pas de tests intrusifs. (La télémaintenance et le tunnel sécurisé ne sont pas requis à ce stade.)

7) Critères d'acceptation (vue client)

- Le socle d'informations défini en III.2 est collecté, présenté localement et consultable au centre (dernier rapport).
- La solution est exploitable par le support (N1/N2), sans expertise développement.

IV – LES LIVRABLES

À l'issue de la mission, NFL attend une série de livrables clairement identifiés, qui valideront l'atteinte des objectifs (cf. III.1), garantiront l'exploitabilité par le support N1/N2 et sécuriseront la mise en service.
Chaque livrable est numéroté, décrit et assorti de critères d'acceptation mesurables afin d'éviter toute ambiguïté d'interprétation.

Livraison finale : archive .zip (+ accès lecture au dépôt Git le cas échéant).

1. Un rapport de travail

- Contenu attendu : un document détaillé retraçant la démarche, les choix technologiques motivés, l'organisation de l'équipe (tâches, rôles, workflow) et les preuves associées. Ce rapport doit permettre à un lecteur externe (équipe support/qualité) de comprendre rapidement ce qui a été fait, pourquoi, comment, et avec quels impacts.
- Format : PDF, structuré et numéroté.

2. Composant "Seahawks Harvester" (franchise)

- Contenu attendu :
  - Scripts Python opérationnels,
  - Tableau de bord local minimal affichant les indicateurs convenus,
  - Fichier local contenant les données
  - Fichier VERSION (identifiant version + date/heure).

- Format : dépôt (structure lisible)
- Exécution sans connexion au centre (dernier résultat consultable) ; génération des fichiers en local versionnés contenant les indicateurs requis (IP/ID hôte, nb équipements détectés, synthèse dernier scan, latence moyenne, version).

3. Plateforme "Seahawks Nester" (centre)

- Contenu attendu : Application de consultation permettant
  - De lister les sondes (état + dernier contact)
  - D'obtenir le détail d'une sonde avec accès au dernier rapport (lecture directe),
- Format : dépôt

4. Runbook d'exploitation N1/N2 (franchise & centre)

- Contenu attendu : procédures simples illustrées : lancer une collecte, consulter le résultat local, vérifier la version, accéder au dernier rapport au centre, bascule "plan B" en cas d'échec.
- Format : PDF (5–8 pages), captures d'écran annotées.
- Opérable par un technicien N1/N2 sans compétence développement ; étapes numérotées, critères d'issue ("succès / échec / action suivante").

5. Support de soutenance

En complément des livrables, l'équipe projet devra préparer un support de présentation destiné à la soutenance finale devant le client (public technique). Ce support devra synthétiser les principaux éléments du travail réalisé : démarche suivie, difficultés rencontrées, solutions mises en place, résultats obtenus et perspectives.

Il est important de souligner que l'évaluation de cette MSPR repose sur la combinaison des trois éléments suivants :

- la qualité du travail réalisé au cours du projet,
- la pertinence et l'exhaustivité des livrables remis
- et la capacité de l'équipe à présenter, justifier et valoriser ce travail lors de la soutenance orale.

Les équipes devront donc s'assurer que la soutenance reflète bien l'ensemble des compétences attendues, en démontrant à la fois la maîtrise technique et la capacité à communiquer efficacement auprès d'un client professionnel.

V – RESSOURCES FOURNIES

Afin de permettre à l'équipe projet de mener à bien la mission, NFL mettra à disposition un ensemble de ressources techniques et documentaires. Ces éléments constituent la base de travail commune et devront être exploités et enrichis par les apprenants.

1. Assistance et périmètre

Dans le cadre de ce projet pédagogique, l'équipe projet n'aura aucun **contact direct NFL**. Le cahier des charges constitue la seule expression officielle du besoin. Toute demande de clarification devra être traitée avec l'encadrant pédagogique, jouant le rôle du client.

2. Webographie

Python

- Linkedin Learning Développement Python
- Logging HOWTO (officiel) – tutoriel de base pour configurer et utiliser logging. Python documentation
- Logging Cookbook (officiel) – recettes avancées (handlers, dictConfig, rotation…). Python documentation
- Référence API logging (officiel) – description complète du module. Python documentation
- PEP 8 – Guide de style Python – conventions pour un code lisible. Python Enhancement Proposals (PEPs)

Gestion de versions

- Git (livre en ligne) – Git
- Comparatif des workflows Git (Atlassian) – centralisé, feature-branch, Gitflow. Atlassian
- GitHub Actions (docs) – automatiser builds/tests/déploiements. GitHub Docs

Qualité & logs

- OWASP Logging Cheat Sheet – bonnes pratiques de journalisation de sécurité. cheatsheetseries.owasp.org
- OWASP Logging Vocabulary – vocabulaire standard pour les événements de sécurité. cheatsheetseries.owasp.org
- Python JSON Logger (docs) – formater les logs Python en JSON (ingestion ELK/Cloud). Nhairs
- Python Guide – Logging – synthèse pédagogique des options de logging. Guide du Routard Python

Sécurité – bases (auto-apprentissage)

- OWASP Cheat Sheet Series (index) – portail des fiches pratiques sécurité. cheatsheetseries.owasp.org
- OWASP Secrets Management Cheat Sheet – gestion des secrets (stockage, rotation). cheatsheetseries.owasp.org
- CIS Critical Security Controls v8 – priorisation des contrôles essentiels. CIS
- CIS Controls v8.1 (téléchargement) – version la plus récente (guide PDF). CIS

Architectures de consultation centralisée

- Flask – Tutoriel officiel – créer un petit serveur Python pour exposer des rapports. flask.palletsprojects.com
- NGINX – Servir du contenu statique – héberger des rapports statiques simplement. Documentation NGINX
- Google Cloud Storage – Héberger un site statique – alternative "sans serveur" pour rapports. Google Cloud