Anthony PETRICCA *– BTS SIO Option A SISR*

ACADÉMIE
DE GRENOBLE
Liberté
Égalité
Fraternité

lycée saint marc

Réalisations en milieu professionnel en alternance chez Toolog (2 ans)

Développement d'outils métiers internes & sécurisation Web

0 - Sommaire :

Sommaire

1 – Ajout outils sur l'intranet :

- 1.1 Solution développée en PHP

Mise en place de 2 solutions développées en PHP pour l'intranet de l'entreprise à la demande des chefs d'équipes de manière à faciliter le travail des opérateurs et à gagner du temps. J'ai développé et testé en préprod pour chaque scénario.

Aperçu de la solution sur le portfolio.

- 1.2 Mail envoyé aux chefs d'équipes

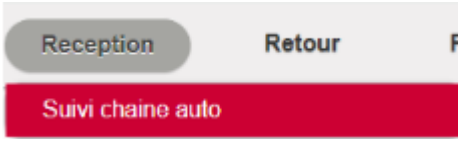
Voici le mail (version anonymisé) pour présenter la solution aux chefs d'équipes :

Bonjour à tous,

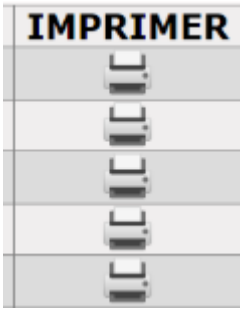
A la demande de certains d'entre vous et pour faciliter la réédition d'étiquettes RFID,

Nous avons mis à jour deux menus sur *intranet.fr* :

- *intranet.fr/chaineauto.php* dans Réception ➔ Suivi chaine auto :



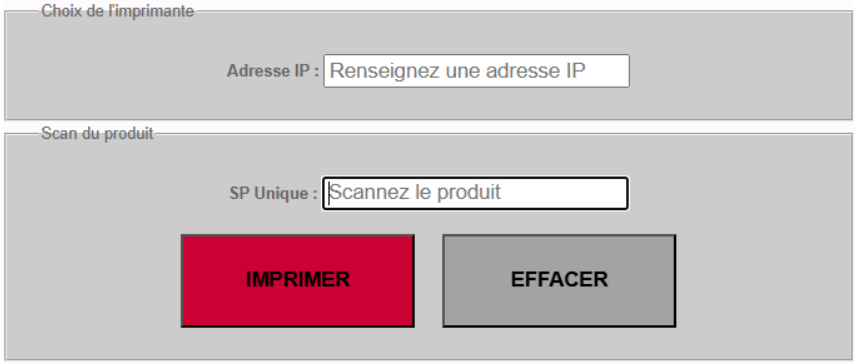
Dans ce menu vous trouverez une colonne supplémentaire à la fin de chaque ligne avec un logo d'imprimante pour imprimer l'étiquette correspond à sa ligne sur l'imprimante RFID à droite sur le bureau chaîne auto (*10.28.X.X*).



Attention, si l'étiquette à rééditer contient un code **PH** et/ou **CO**, pensez à garder la pièce de côté pour les contrôles nécessaires car le code ne sera pas réédité, exactement comme lorsque vous utilisez speed pour la réédition.

-*intranet.fr/reedition_zpl.php* dans Préparation/Rangement :

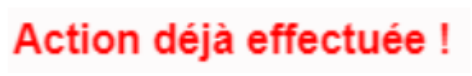
Ce menu permet de rééditer une étiquette produit (RFID) de forme standard sur l'imprimante RFID de votre choix à condition que le papier utilisé soit le bon évidemment (par exemple l'imprimante du bureau chaîne auto convient parfaitement si elle est connectée).



Pour l'utiliser il faut dans l'ordre qui suit, mettre une **adresse IP d'imprimante RFID** au format standard, renseignez **le SP unique** du produit puis "Imprimer". Si vous souhaitez réimprimer une nouvelle étiquette ou la même, il faut actualiser la page et **faire attention aux messages** qui indiquent le statut de l'impression :

(Vous pouvez récupérer votre étiquette)

ou



(Recommencer si vous venez d'imprimer une étiquette, c'est une sécurité pour éviter de sortir plusieurs étiquettes)

Erreur de connexion à l'imprimante : 10.28.8.999
Vérifiez le statut de connexion de l'imprimante.

ou *(L'adresse IP de l'imprimante est incorrecte ou l'imprimante est injoignable/non connecté)*

Si vous avez des questions sur le fonctionnement, n'hésitez pas à me contacter.

Cordialement,

Anthony PETRICCA, Service Informatique TOOLOG.

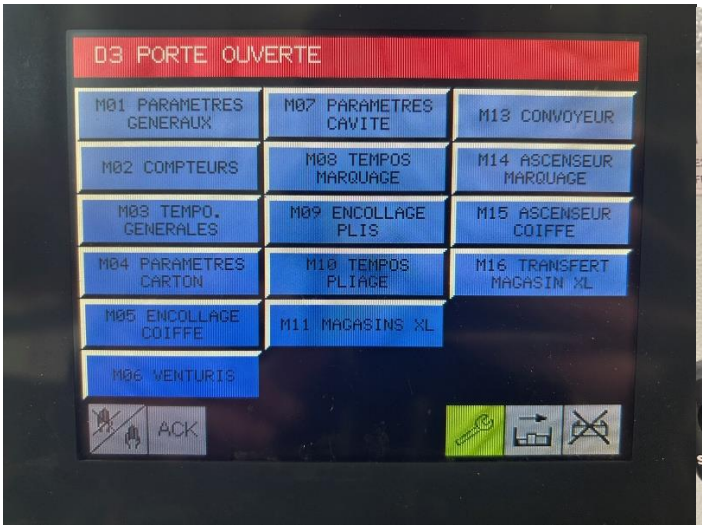
2 – Projet coiffe neutre (intranet + physique) :

- 2.1 Besoins exprimés & Objectifs

Le siège nous a demandé de mettre en place les coiffes neutres en plus des coiffes nommées Spartoo pour les colis destinés aux magasins. Ainsi le but était de régler les fermeuses cartons 1 & 2, les seules avec plusieurs choix de magasins (tapis où l'on dispose les coiffes). Pour éviter les erreurs des opérateurs lors d'un défaut sur un colis, je devais mettre en place une solution sur l'intranet permettant de vérifier si le colis doit être fermé avec une coiffe neutre ou non.

- 2.2 Problématique principale

Physiquement, les fermeuses sont configurées par leur boitier de l'entreprise B+, il fallait pour chaque fermeuses trouver quel était le numéro informatique du magasin correspondant au numéro physique qui n'était pas nécessairement le même. Ensuite, il fallait associer ce numéro dans le script pour savoir, lors de la descente des commandes, quelle coiffe prendre automatiquement selon le colis.



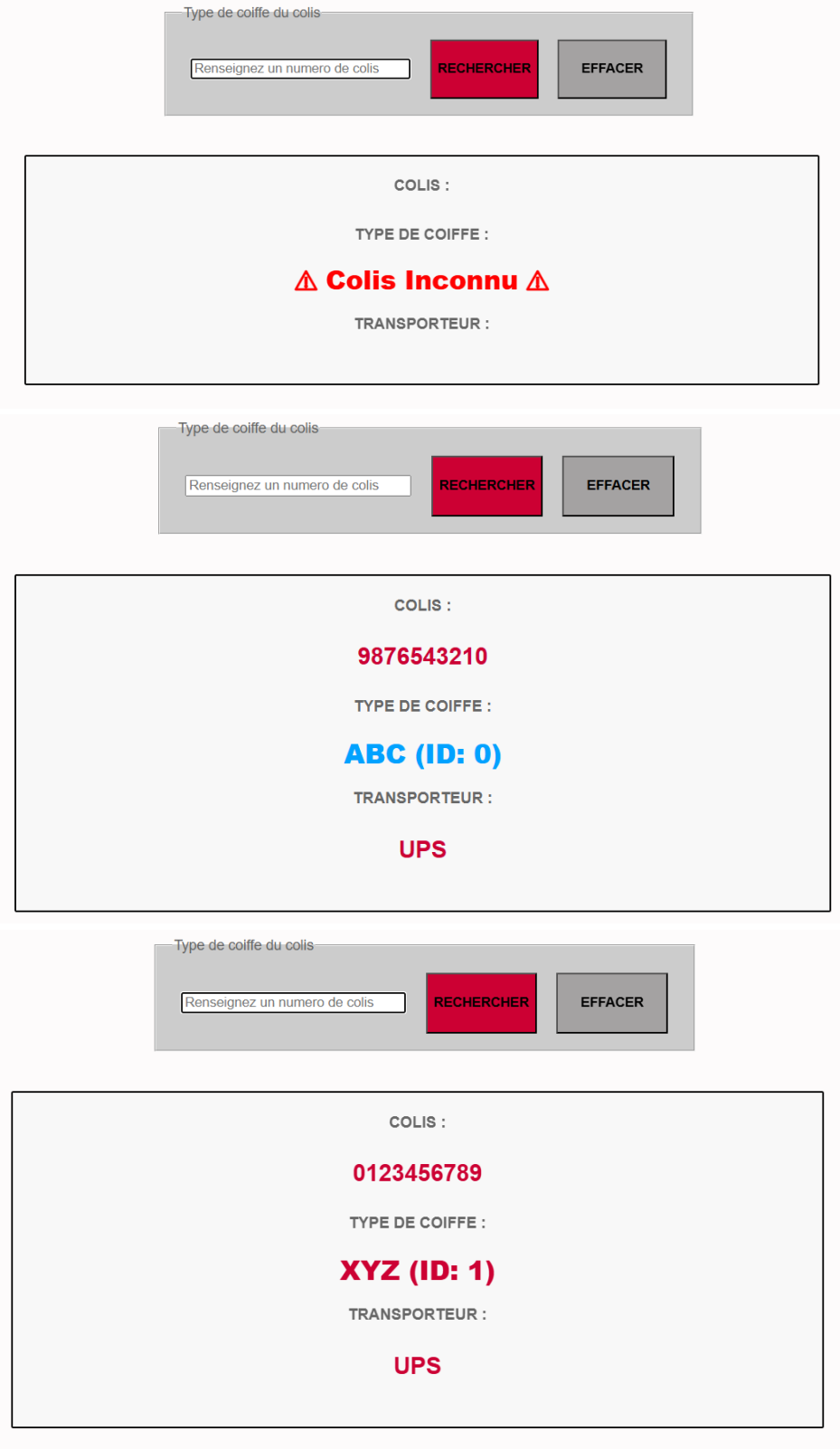
Ci-dessous la fermeuse n°2 contenant 75% de coiffes Spartoo et 25% de coiffes neutres :

Lors de la mise en place physique on a plusieurs sorties de colis car les opérateurs et nous-même, avons encore du mal à savoir comment poser les coiffes neutres étant donné qu'elles n'ont pas de distinction visuelle :

- 2.3 Résultat final (anonymisé)

ID: ABC: 0, 2, 3 => Coiffes Spartoo

ID: XYZ: 1 => Coiffes Neutre



3 – Sécurisation Web :

- 3.1 Sécurisation contre injection SQL

Pour cela, j'ai demandé à ChatGPT un script qui injecte les commandes que je lui ai donnée pour tenter plusieurs formes d'injection SQL (Code malveillant exécuté sur des sites mal sécurisés de manière à accéder au contenu de la base de données).

Script python fonctionnant avec URL du site si le site est en http :

```python
import requests
from bs4 import BeautifulSoup
import logging

# Configuration
URL = "http://"  # Remplace par l'URL du site
FIELDS = ["search", "comment", "email"]  # Liste des champs à tester

# Configuration des logs
logging.basicConfig(filename="sql_test.log", level=logging.INFO, format="%(asctime)s - %(levelname)s - %(message)s")

# Charges SQL à tester (adapté aux différents SGBD)
PAYLOADS = [
    "' OR '1'='1",  # Authentification bypass
    "' UNION SELECT 1,2,3 --",  # UNION SELECT
    "' AND 1=CONVERT(int,@@version) --",  # Erreur SQL (SQL Server)
    "' AND SLEEP(5) --",  # Delay (MySQL, MariaDB)
    "' WAITFOR DELAY '0:0:5' --",  # Delay (SQL Server)
    "' OR 1=1 --",  # Test classique sur champs divers
    """' OR 'a'='a' --""",  # Alternative auth bypass
    "' AND (SELECT COUNT(*) FROM information_schema.tables) > 0 --",  # Exfiltration basique
]

def test_sql_injection():
    session = requests.Session()

    for field in FIELDS:
        for payload in PAYLOADS:
            data = {f: "test" for f in FIELDS}  # Valeurs par défaut
            data[field] = payload  # Injection SQL

            response = session.post(URL, data=data)

            if is_vulnerable(response.text):
                message = f"⚠️ Le champ '{field}' est potentiellement vulnérable avec : {payload}"
                print(message)
                logging.warning(message)
            else:
                message = f"✅ Le champ '{field}' semble sécurisé contre : {payload}"
                print(message)
                logging.info(message)

def is_vulnerable(html):
    """Analyse la réponse pour détecter une faille SQL"""
    erreurs_sql = [
        "You have an error in your SQL syntax",  # MySQL / MariaDB
        "Microsoft OLE DB Provider for SQL Server",  # MSSQL
        "Unclosed quotation mark after the character string",  # MSSQL
        "Warning: mysql_fetch",  # MySQL
        "ORA-00933: SQL command not properly ended",  # Oracle
    ]
    return any(erreur in html for erreur in erreurs_sql)

if __name__ == "__main__":
    logging.info("Début du test SQL Injection")
    test_sql_injection()
    logging.info("Fin du test SQL Injection")
```

J'ai tenté également à la main sur les champs de renseignements « input » la commande suivante : **' OR 1=1 –**

Si le résultat était une erreur SQL voir même que j'arrivais à extraire des données, il fallait sécuriser le code PHP. Pour cela, j'ai restreint les champs aux nombres de charactères nécessaires avec un input **maxlenght** et j'ai chargé la requête SQL avant dans le code de manière à vérifier que le résultat demandé est bien la réponse à une requête précise et non pas un **Select ***. Il m'a fallu vérifier chaque PHP de l'intranet et le sécuriser.

- 3.2 Sécurisation contre XSS

Le XSS (Cross-site Scripting) est également une manière d'attaquer un site en injectant du code malveillant côté client du style JavaScript, VBScript… Pour vérifier si nos pages étaient sensibles à ce type d'attaque. On a utilisé une machine montée spécialement en Debian 12 pour exécuter l'outils open-source XSStrike disponible sur GitHub. Une fois exécuté on a pu voir les parties faibles de notre intranet et corriger les failles.