Stage de première année



Le CIRISI est situé à l'arsenal de Cherbourg, à côté de DCNS. Il fait partie de la Marine et gère les équipements informatiques des bateaux militaires et des sémaphores. Leur zone d'action est très étendue et ils sont souvent amenés à ce déplacer. Je travaille de 8h à 11h30 et de 13h15 à 17h, du Lundi au Vendredi, et ce pendant 5 semaines qui s'étendent du 23 Mai au 24 Juin.

Projet

Pendant ces 5 semaines de stage, mon projet est de créer une interface capable d'analyser et de détecter si un badge qui passe sur une badgeuse est reconnue ou non, et de l'afficher sur une page accessible depuis un navigateur. J'ai créé un script Python qui permet de télécharger le fichier de log directement depuis la badgeuse et de l'analyser, puis d'écrire dans une base de donnée les numéros, l'heure et la date de passage des badges inconnus. Sur une page PHP, j'ai une liste déroulante avec l'adresse IP et le numéro des badgeuses, ce qui permet à l'utilisateur de choisir ou il veut récupérer les badges inconnus. Un bouton valider nous renvoie vers une page PHP ou sont affichés tous les renseignements.

Première Semaine

Découverte de l'entreprise et du projet. Création d'un script Python qui permet de ce connecter sur la badgeuse grâce à l'adresse IP choisie sur la page web, et de télécharger le fichier de log pour ensuite l'analyser et envoyer les badges inconnus dans une base de donnée. Après plusieurs heures de recherche, je trouve enfin le moyen de me connecter sur la badgeuse. Pour ce connecter, il faut saisir un mot de passe et un login, ce que je n'avais pas réussi à faire, mais après de nombreuse recherches j'ai enfin trouvé le moyen de me connecter et de télécharger le fichier de log:

```
import urllib.request
password_mgr = urllib.request.HTTPPasswordMgrWithDefaultRealm()
top_level_url = "http://192.168.0.250"
password_mgr.add_password(None, top_level_url, "ethanium", "ethanium")
handler = urllib.request.HTTPBasicAuthHandler(password_mgr)
opener = urllib.request.build_opener(handler)
urllib.request.install_opener(opener)
urllib.request.urlretrieve("http://192.168.0.250/Temp/Ethanium_out.log","Ethanium_out.log")
```

Ensuite, je me suis attelé à créer une fonction qui permet d'analyser le fichier de log, et de ressortir les entrées inconnues.

```
chaine = "inconnu"
validation = "Badgeage"

fichier = open("D:\Badges\Ethanium_out.log","r")
for ligne in fichier:
    if chaine in ligne:
        heure = ligne[11:19]
        print (ligne)
        fichier = open("D:\Badges\Ethanium_out.log","r")
        for ligne in fichier:
            if validation in ligne:
                if heure in ligne:
                    print(ligne)

fichier.close()
```

On peux voir que la variable chaîne contient le mot « inconnu », c'est parce que la ligne qui contient un badge inconnu contient le mot « inconnu ».

Pour avoir le numéro de badge, il faut aussi récupérer la ligne juste au dessus, j'ai donc rentré un mot dans la variable validation qui est présent sur toutes les lignes, j'ai choisi arbitrairement le mot « Badgeage ». Il faut faire très attention aux minuscules et aux majuscules, si le mot dans le dossier contient une majuscule et que l'on ne la met pas dans le script, celui-ci ne va pas marcher.

Ensuite je demande au script d'ouvrir le fichier téléchargé précédemment, avec son emplacement, puis je le fais regarder ligne par ligne si le mot « inconnu » est présent. Si il y est, alors la variable heure récupère les entre 11 et 19 caractères afin d'avoir uniquement l'heure du badgeage, puis ensuite le script vérifie si à l'heure ou il y'a a eu le badgeage inconnu le mot «Badgeage » est présent (il l'est toujours, c'est juste une façon plus rapide de récupérer la ligne contenant le numéro du badge), et d'afficher le résultat.

Cependant, pour le moment la badgeuse n'est pas raccordée au serveur de badge, elle ne peux pas analysée par elle même si un badge est inconnu ou non.

Pour cela j'ai donc installé sur un serveur un ESXi de chez VMWARE, qui me permet de récupérer un clone du serveur de badge officiel et de l'avoir sur mon serveur, pour pouvoir faire des tests avec des faux badges qui ne sont pas connus par le serveur.

ESXi est un hyperviseur qui permet de créer des machines virtuelles sur un serveur, et de pouvoir les contrôler sur un poste client équipé du logiciel ViClient.

En attendant d'avoir une copie du serveur de badge, je me suis attelé à faire une page web, et d'améliorer mon script afin que celui-ci enregistre dans une base de donnée l'heure, la date et le numéro du badge inconnus. J'ai donc installé un xampp sur ma machine pour pouvoir afficher du PHP, puis dans mon script j'ai utilisé la librairie pymysql.cursors qui permet d’enregistrer dans une base de donnée Mysql les valeurs d'une variable.

J'ai ensuite créé une page web, qui permet de choisir la badgeuse et un bouton valider qui envoie sur une page PHP avec les résultats.

```
<HTML>
<HEAD>
<meta charset="UTF-8">
<TITLE>Gestion des badges</TITLE>
</HEAD>
<BODY>
<DIV ALIGN="center">
<H2>Gestion des badges</H2>

<?php
mysql_connect("localhost", "root", "");
mysql_select_db("python");
$res = mysql_query("SELECT ip, lieu FROM badgeuse");
echo "<form action='resultat.php' method='POST'>
<table><tr><td>Choisissez votre badgeuse:
<select name='choix'>";
while ($row=mysql_fetch_array($res))
{
    echo"<option>$row[ip], $row[lieu]</option>";
}
$_POST['choix'];
echo"</select>
</td>
</tr>
</table>";
?>

<br/>
<input type="submit" value="Valider"/>

</DIV>
</BODY>
</HTML>
```

Ce code nous affiche donc la page suivante :



Une fois que l'on clique sur le bouton, on arrive sur la page suivante qui exécute le script Python et affiche ce qui est dans la base de donnée:

```
<HTML>
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("python");
?>
<HEAD>
<meta charset="UTF-8">
<TITLE>Gestion des badges</TITLE>
</HEAD>
<BODY>
<DIV ALIGN="center">
<H2>Gestion des badges</H2>
<h3>Vous avez choisi la badgeuse: </h3>
<?php
$choix = $_POST['choix'];
echo $choix;
?>
<br/>
<?php
$ip = explode(",",$choix);
$adresse = $ip[0];
echo $adresse;
?>
<br/>
<br/>
<P>Voici la date, l'heure et le numéro de badge: </P>
<?php
$tmp = exec("d:\Python\python.exe Script.py $adresse", $output);
?>

</form>

<?php
$reponse = mysql_query("SELECT date, heure, numero FROM badge");
while($donnees = mysql_fetch_array($reponse))
{
echo $donnees['date']."\r\n";
echo $donnees['heure']."\r\n";
echo $donnees['numero']."<br/>";
}
mysql_close();
?>
<INPUT type="button" value="Retour" onClick="window.history.back()">
</DIV>
</BODY>
</HTML>
```

Deuxième Semaine

En discutant avec mon maitre de stage, il c'est avéré que l'heure et la date ne sont pas dans le bon format, en effet, dans ma base de donnée celles ci sont en varchar (en chaine de caractère), il faudrait mieux les mètres en datetime afin de facilité l'enregistrement. En effet, le problème majeur de mon script est qu'il enregistre à chaque fois toutes les informations qui sont présentes dans le fichier de log, la base de donnée est donc remplie de doublons. Pour éviter cela, il faut vérifier que la valeur de ma variable Python est supérieure à la dernière entrée dans ma base de donnée, comme cela on pourra juste ajouter les valeurs qui ne sont pas dans la base, et ainsi éviter les doublons.

J'ai aussi améliorer mon script Python afin que celui-ci puisse récupérer l'adresse IP qui est utilisée dans le navigateur, et ainsi l'utiliser dans une variable qui va permettre de ce connecter sur l'adresse IP de la badgeuse souhaitée.

J'ai pu aussi partir avec des membres de l'équipe à bord d'un bateau militaire, Le Pluvier qui avait un problème informatique, je n'ai pu qu'observer.



Je n'ai malheureusement pas pu terminer cette deuxième semaine de stage, j'ai attrapé la grippe, ce faisant je n'ai pas pu venir.

Troisième Semaine

Retour après la maladie, je reprends et améliore mon script Python pour que celui-ci n'ajoute dans la base de donnée uniquement les nouvelles valeurs, afin d'éviter les doublons. Je passe mes valeurs au format datetime, pour que je puisse les comparer plus facilement.

```
resultdate = datetime.strptime(resultdate, "%d:%m:%Y%H:%M:%S")
```

Je fais la même manipulation avec la date qui est trouvée dans le fichier de log téléchargé. Vu que ces deux variables sont du même type, je peux maintenant les comparer. Pour que les données soient envoyées dans la base de donnée, il faut que la valeur trouvée dans le fichier de log soit **strictement** supérieure à la dernière valeur qui est trouvée dans ma base de donnée. Maintenant, il n'y a plus de doublons, et plus de problèmes de dates, ni d'heure. Nous sommes allés au Groupement Plongeur Démineur (GPD) afin de changer une tour, mais une personne m'a fait visité et m'a expliqué le fonctionnement des démineurs et m'a présenté le matériel d'intervention. Ils ont un musé avec des équipements de guerre.



Ensuite, nous sommes allés dans un sémaphore pour régler un problème de disque dur, et j'ai pu discuter avec des personnes qui y travaillent. Ils surveillent l'activité des bateaux dans un secteur précis, et indiquent si il y a des problèmes.



Avancement du script et mise en place d'un CSS pour rentre la page esthétique.

Quatrième Semaine

N'ayant pas voulu faire du JavaScript afin de faire page permettant de voir en temps réel et en continu les logs des badgeuses, la suite de mon projet est de créer un nouveau script Python qui permettrais d'avoir le nombre

de badges détectés (qu'ils soient inconnus ou non) par badgeuse, et ainsi pouvoir faire des statistiques.

Je suis monté à bord de l'Abeille Liberté à cause d'un problème de boite de messagerie, qui à été réglé une fois rentré à l'atelier.



Découverte d'un bug dans mon premier programme, certaines fois les horaires de passage étaient décalés d'une seconde dans le fichier de log, ce qui n'enregistre pas les données dans ma base de donnée. De plus, si la badgeuse n'est pas en ligne, lors du scan total de mon script, le programme va s'interrompre à cause d'une erreur, il n'aura pas réussi à ce connecter. Il faut donc que je gère les erreurs dans le cas ou la badgeuse n'est pas en ligne, alors on passe à celle d'après.

Petit temps mort, j'ai aidé des personnes de l'équipe à remettre en configuration usine deux switchs CISCO, voici la procédure pour le faire: procedure_remise_etat_usine_switch_cisco.pdf

Débogage et suite des deux scripts Python.

Cinquième Semaine

Installation de ProxMox sur un serveur pour le présenter à mon maitre de stage.

installation_proxmox.pdf

Installation d'une machine virtuelle Debian et une Windows 7 (avec les pilotes VirtIO).

installation_driver_virtio_windows_7_proxmox.pdf

Documentation sur le logiciel Openfiler afin de mettre en place un SAN qui servira de stockages à deux ESXi. Je laisse tombe Openfiler car il n'est plus mit à jours et est relativement obsolète. Je choisis donc FreeNAS qui est encore d'actualité. Installation de FreeNAS sur un serveur.

installation_freenas.pdf

Le dernier jours, nous nous sommes rendus dans la sémaphore de la Hague à cause d'un problème téléphonique, c'était un problème de filtre, les communications enregistrées contenaient trop de bruit pour être exploitées.



Puis, l'après-midi, nous avons étés à l'aéroport de Maupertus pour dépanner un disque dur. Nous sommes ensuite rentrés à l'atelier, ce qui conclut mon stage au sein du CIRISI.



Voici le script Python que j'ai pu réaliser pendant ces 5 semaines, ainsi que les pages PHP et ma base de donnée:

badgeuse.zip

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/etudiants/2016/anthony.dupont/stage1

Last update: **2019/08/31 14:21**