

STAGE DANS LE SERVICE INFORMATIQUE DU CHU DE BESANCON

Du 25 janvier au 19 mars 2021



Maitre de stage : Bernard Jacquot

Tuteur du lycée : Jean-Baptiste Aubry

Sommaire

Introduction.....	3
Développement.....	4
Présentation du CHU	4
Découverte de l'environnement de travail	6
Interventions opérationnelles	8
Simulation coupure de courant.....	8
Remplacement d'un cœur de réseau	8
Débrassage d'une baie	10
Mission : Filtrage des traps.....	11
Mission : Détection des mini-switchs sur le réseau	12
Mission : Installation nouveaux switchs.....	19
Mission : Dé câblage des baies inutilisées.....	20
Configuration de switchs.....	21
Installation de switchs dans les baies des nouveaux locaux	22
Conclusion	23

Introduction

Du 25 janvier au 19 mars j'ai pu effectuer mon stage au CHU de Besançon au site de Saint-Jacques à l'Arsenal dans le service informatique. Au cours de mon stage j'ai pu appliquer et approfondir les connaissances acquises au cours de mon BTS SIO et apprendre de nouvelles notions.

En effet mon maître de stage, Bernard Jacquot, étant responsable des équipes systèmes et réseaux, j'ai pu apprendre dans les meilleures conditions à son contact et à celui des autres membres de l'équipe.

Mon stage portant sur la configuration, optimisation de solutions de supervision réseau, m'a permis de découvrir le métier de technicien administrateur réseaux sur un grand réseau informatique comme celui du CHU de Besançon qui comporte plus de 900 équipements.

Ce stage m'a permis de voir l'utilité de chaque équipement et les différentes fonctionnalités de chacun ainsi que les différents problèmes qui surviennent et les méthodes pour les résoudre.

J'ai pu voir les différentes tâches que ce soit au niveau opérationnel où il faut se déplacer pour régler un souci d'un PC, de commutateurs ou encore de borne Wifi. Aussi, j'ai vu comment gérer un grand nombre d'équipement les répertorier, l'utilisation d'outils de supervision

Afin de présenter le déroulement de mon stage je vais d'abord présenter le CHU et mon environnement de travail quotidien à l'Arsenal avant de détailler les tâches que j'ai effectuées au cours de mon stage et conclure sur cette expérience très enrichissante en expliquant ce qu'elle m'a apporté en tant que futur technicien réseaux.

Développement

Présentation du CHU

Le CHU est composé de plusieurs bâtiments répartis en 2 sites principaux : Jean-Minjoz et Saint-Jacques, l'Institut de Formation de Professions de Santé (IFPS) et le Centre de réadaptation situés aux Tilleroyes. Les services du CHU qui était il y a quelques années exclusivement à Saint-Jacques, qui d'ailleurs inscrit en tant que bâtiment historique depuis 2012, est déplacée petit à petit vers l'hôpital Jean Minjoz depuis 2012 ; un déménagement qui devrait prendre fin horizon 2022-2023. En effet, la création de l'hôpital Jean Minjoz avait pour objectifs la remontée des services situés à l'hôpital Saint-Jacques Ces deux sites interconnectés sont séparés par 7km.



Minjoz à gauche et St-Jacques à droite de la carte

Le CHU dispose d'une capacité d'accueil de 1 400 lits et places d'hospitalisation. L'établissement compte 7 200 agents, personnels médicaux et non médicaux, ce qui fait de lui le plus gros employeur de la ville de Besançon.

Le site Jean-Minjoz a été ouvert en 1983 et on peut voir ci-dessous les bâtiments principaux qui sont reliés : le PCP, PCBio, bâtiment vert et le bâtiment gris.



Et le site Saint-Jacques avec l'hôpital à gauche et le service informatique à droite sur le site de l'Arsenal :



Le service informatique où j'ai travaillé était encore situé à l'Arsenal et le déménagement a eu lieu après mon départ. La majorité des équipements avaient déjà été déplacés et ils ne restaient que ceux nécessaires au fonctionnement du service qui gère l'ensemble du parc informatique.

Il n'est pas nécessaire d'être en physique pour la plupart des tâches car on n'accède aux équipements uniquement via SSH ou Telnet et on se branche en console uniquement pour la configuration initiale ou en cas de défaillance qui le requiert.

Le réseau est composé en plusieurs niveaux :

- Le cœur de réseau qui comme son nom l'indique est la base du réseau et est composé de plusieurs commutateurs de niveau 3.
- Les commutateurs de concentration donc de niveau 3 sur les différents sites connectés en double attachement au cœur de réseau.
- Les commutateurs de distribution donc de niveau 2 eux aussi connectés en double attachement aux commutateurs de concentration.

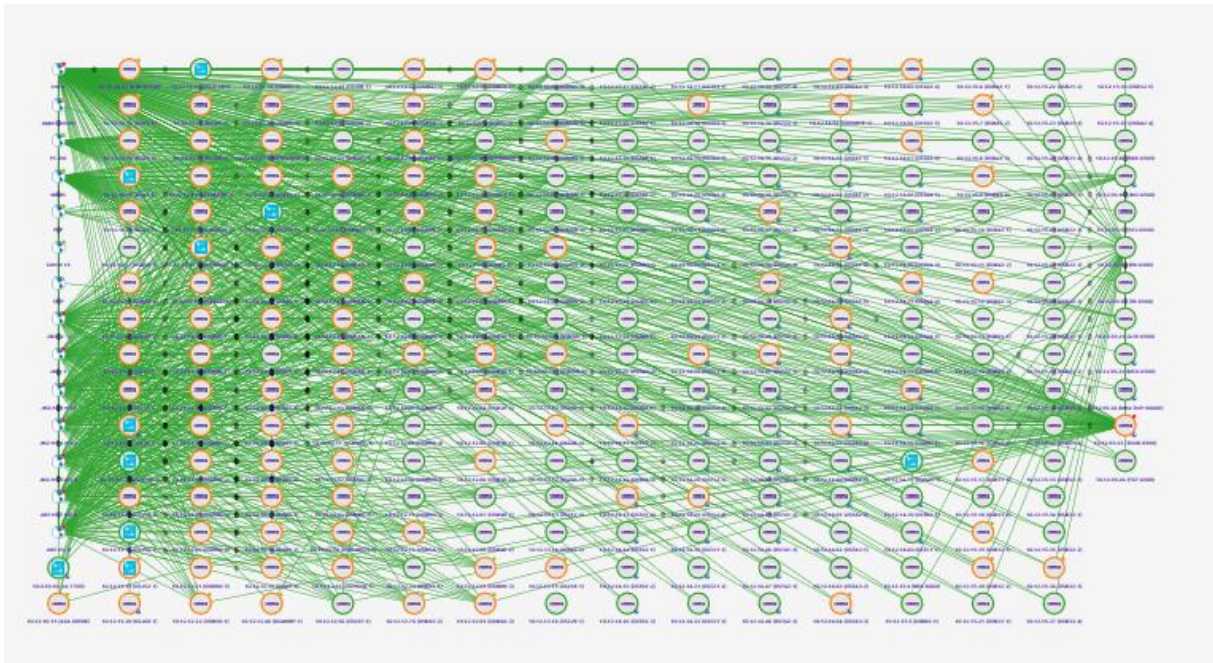
Les équipements sont des équipements Alcatel et j'ai pu manipuler 3 versions de commutateurs différentes : les 6600, les 6850 et les 6860 qui sont les commutateurs de distribution.

Découverte de l'environnement de travail

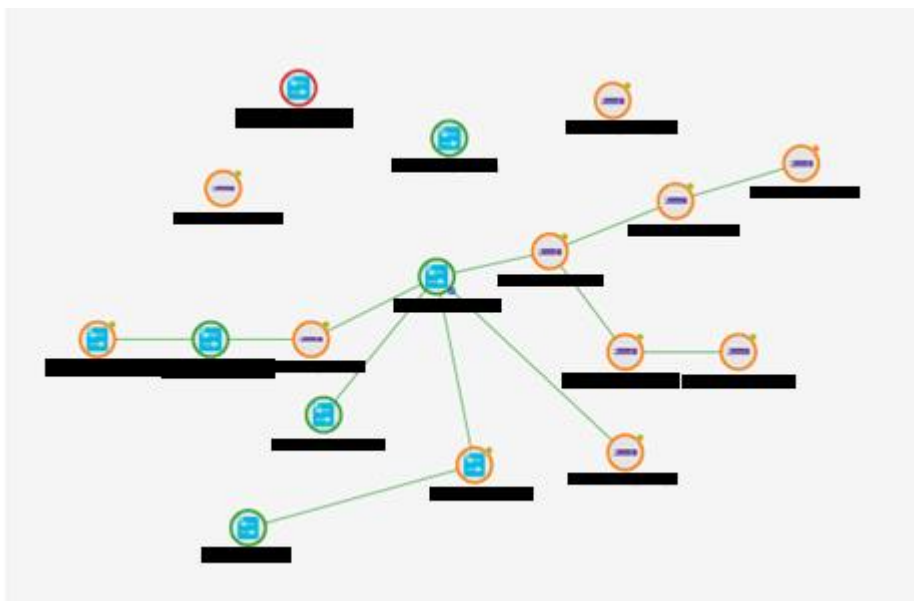
Dès mon premier jour j'ai configuré et connecter mon PC au réseau puis j'ai découvert l'outil de supervision du CHU qui est Omnivista car tous les équipements du réseau sont de la marque Alcatel et non Cisco et il y a donc un logiciel fournit pour la supervision.

Grâce à Omnivista, on peut observer une représentation de tout le réseau via une carte comme ci-dessous :

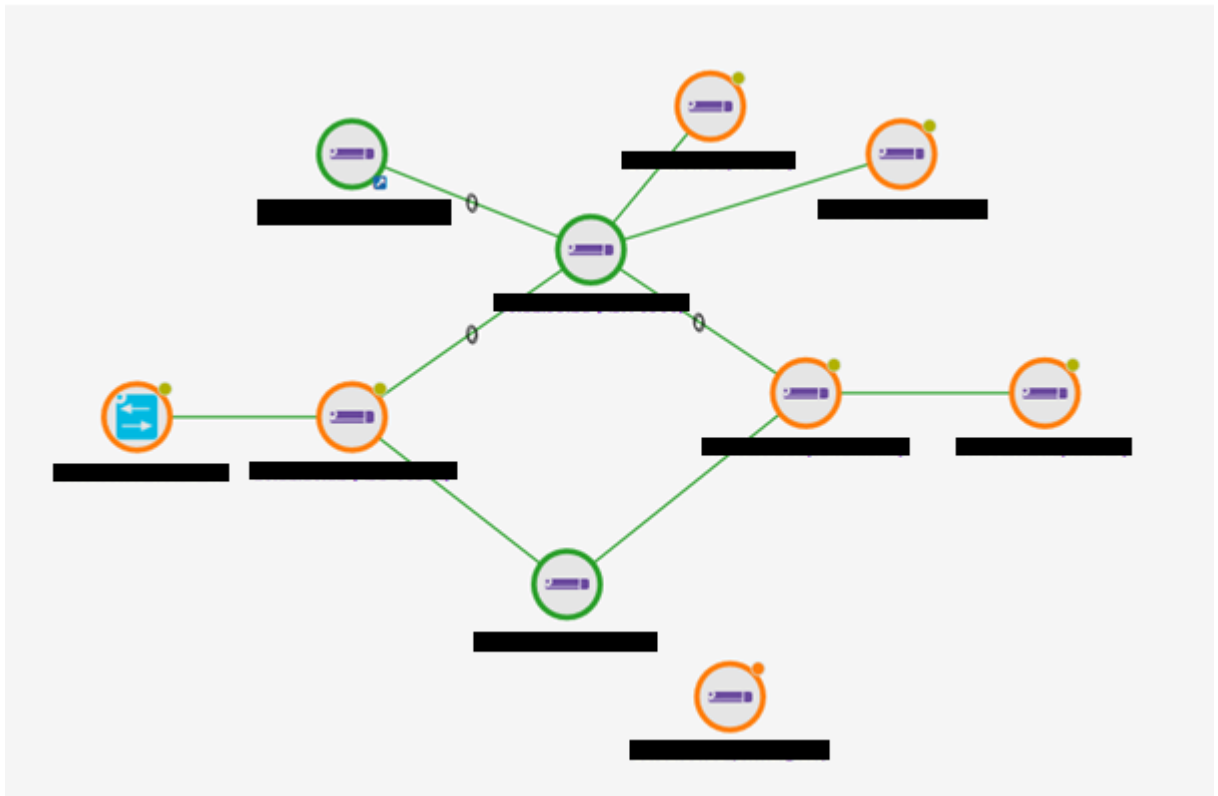
- On peut voir ici tous les commutateurs présents a sur le site Jean-Minjoz :



- Ici les commutateurs du site Saint-Jacques :



- Enfin les commutateurs de l'Arsenal qui est un site annexe de Saint-Jacques :



On peut voir que le commutateur qui n'est pas relié aux autres qui représente ma pile de switch où j'avais connecté mon PC mais après quelques tests j'ai dû changer de version de commutateur car ma version était trop ancienne.

Voici la première pile :



Interventions opérationnelles

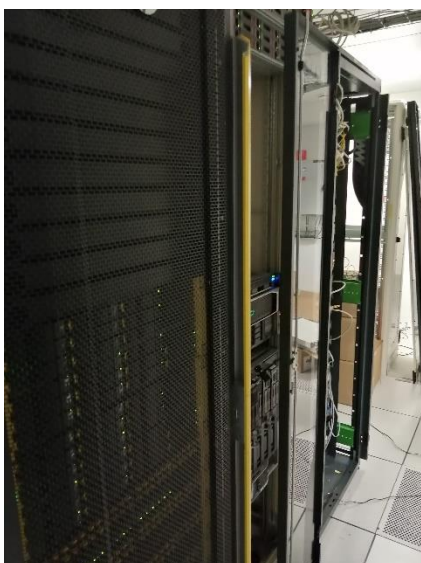
Simulation coupure de courant

Après un test de coupure de courant, il y a eu des problèmes avec des certains commutateurs qui n'ont pas redémarré correctement et grâce à Omnivista, on a pu les détecter et aller les reboot manuellement.

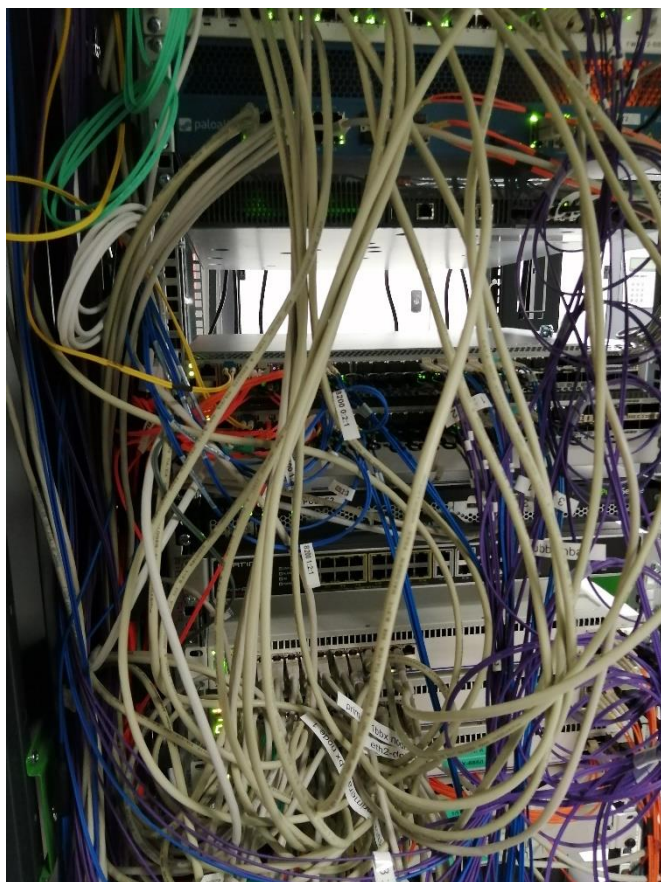
Remplacement d'un cœur de réseau

Il y a un système de redondance pour tous les équipements importants du réseau et il a fallu changer un des cœurs de réseau. Il a donc fallu l'installer et faire des branchements dans les différentes baies de l'hôpital pour que l'équipement soit fonctionnel.

J'ai pu voir la salle qui contenait des serveurs, des commutateurs Ci-dessous on peut les voir:



On peut également voir le câblage uniquement sur quelques commutateurs :



Débrassage d'une baie

Il n'y avait plus de ports disponibles dans la baie de brassage de la zone d'accueil du CHU pour brancher de nouveaux équipements, on nous a donc demandé de débrasser une partie.

J'ai donc repéré les ports qui n'étaient pas utilisés via une connections sh ou je regardais l'activité de chaque port des 5 piles de commutateurs pour savoir lesquels pouvaient être débranchés.

Ci-dessous on peut voir que l'interface 2/19 n'a pas eu d'activité contrairement à l'interface 2/20. J'ai fait cette commande pour vérifier sur tous les ports de la baie et j'ai relevé 165 ports inactifs.

```
OST03-1> show interfaces 2/19
Slot/Port 2/19 :
Operational Status : down,
Last Time Link Changed : WED JUL 31 07:10:36 ,
Number of Status Change: 0,
Type : Ethernet,
SFP/XFP : Not Present,
MAC address : e8:e7:32:05:90:a4,
BandWidth (Megabits) : - , Duplex : -,
Autonegotiation : 1 [ 1000-F 100-F 100-H 10-F 10-H ],
Long Frame Size(Bytes) : 9216,
Rx :
Bytes Received : 0, Unicast Frames : 0,
Broadcast Frames: 0, M-cast Frames : 0,
UnderSize Frames: 0, OverSize Frames: 0,
Lost Frames : 0, Error Frames : 0,
CRC Error Frames: 0, Alignments Err : 0,
Tx :
Bytes Xmitted : 0, Unicast Frames : 0,
Broadcast Frames: 0, M-cast Frames : 0,
UnderSize Frames: 0, OverSize Frames: 0,
Lost Frames : 0, Collided Frames: 0,
Error Frames : 0

OST03-1> show interfaces 2/20
Slot/Port 2/20 :
Operational Status : down,
Last Time Link Changed : MON FEB 15 16:49:33 ,
Number of Status Change: 501,
Type : Ethernet,
SFP/XFP : Not Present,
MAC address : e8:e7:32:05:90:a5,
BandWidth (Megabits) : - , Duplex : -,
Autonegotiation : 1 [ 1000-F 100-F 100-H 10-F 10-H ],
Long Frame Size(Bytes) : 1553,
Rx :
Bytes Received : 9497029587, Unicast Frames : 57518415,
Broadcast Frames: 2003948, M-cast Frames : 11454,
UnderSize Frames: 0, OverSize Frames: 0,
Lost Frames : 0, Error Frames : 3,
CRC Error Frames: 3, Alignments Err : 0,
Tx :
Bytes Xmitted : 70296560657, Unicast Frames : 70350736,
Broadcast Frames: 209053178, M-cast Frames : 30909470,
UnderSize Frames: 0, OverSize Frames: 0,
Lost Frames : 21, Collided Frames: 0,
Error Frames : 0
```

Nous sommes donc ensuite allés sur place afin de débrancher une partie de ces ports.

Mission : Filtrage des traps

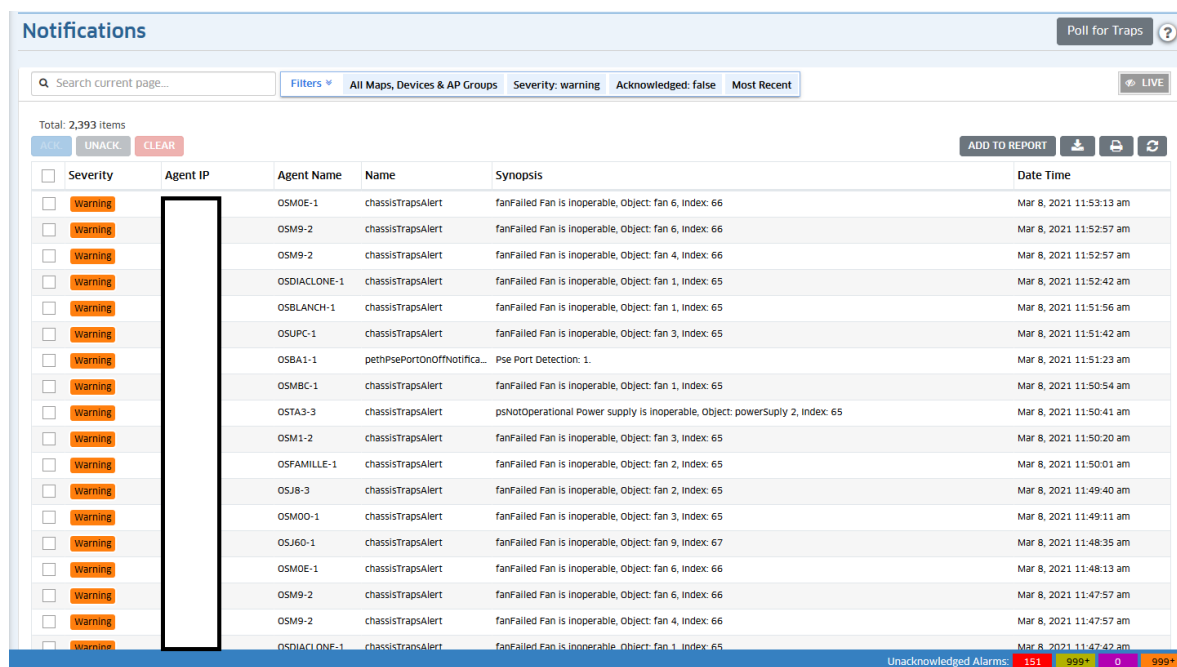
Les traps SNMP sont des informations envoyées en utilisant le protocole SNMP depuis un équipement supervisé vers un serveur de supervision ici Omnivista. Ce concept s'oppose au polling où le logiciel de supervision pose des questions aux équipements.

On peut donc voir les problèmes que rencontrent les équipements mais le problème est que les équipements remontent beaucoup d'informations qui ne sont pas forcément utiles.

J'ai donc été chargé de recenser les traps redondantes afin que l'on puisse définir lesquelles sont utiles ou non et ainsi pouvoir filtrer les traps inutiles.

J'ai donc créé un tableau Excel en inscrivant le nom des équipements qui revenaient régulièrement, le nom de la trap ainsi que son détail et je l'ai transmis à mon tuteur.

Ainsi on peut choisir quels filtres activer et chaque filtre a un numéro qui lui correspond et que l'on peut retrouver dans la documentation de chaque équipement.



The screenshot shows the 'Notifications' page in the Omnivista interface. At the top, there's a search bar and filter tabs: 'All Maps, Devices & AP Groups', 'Severity: warning', 'Acknowledged: false', and 'Most Recent'. Below the filters, it says 'Total: 2,393 items'. The main table has columns: 'Severity', 'Agent IP', 'Agent Name', 'Name', 'Synopsis', and 'Date Time'. The 'Agent IP' column is highlighted with a black box. The table lists various traps, mostly 'fanFailed Fan is inoperable' and 'Pse Port Detection: 1'. At the bottom, there's a status bar showing 'Unacknowledged Alarms: 151 999+ 0 999+'. There are also buttons for 'UNACK', 'CLEAR', 'ADD TO REPORT', and 'LIVE'.

Severity	Agent IP	Agent Name	Name	Synopsis	Date Time
Warning		OSMOE-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 6, Index: 66	Mar 8, 2021 11:53:13 am
Warning		OSM9-2	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 6, Index: 66	Mar 8, 2021 11:52:57 am
Warning		OSM9-2	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 4, Index: 66	Mar 8, 2021 11:52:57 am
Warning		OSDIACLONE-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 1, Index: 65	Mar 8, 2021 11:52:42 am
Warning		OSBLANCH-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 1, Index: 65	Mar 8, 2021 11:51:56 am
Warning		OSUPC-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 3, Index: 65	Mar 8, 2021 11:51:42 am
Warning		OSBA1-1	pethPsePortOnOffNotifica...	Pse Port Detection: 1.	Mar 8, 2021 11:51:23 am
Warning		OSMBC-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 1, Index: 65	Mar 8, 2021 11:50:54 am
Warning		OSTA3-3	chassisTrapsAlert	psNotOperational Power supply is inoperable, Object: powerSuply 2, Index: 65	Mar 8, 2021 11:50:41 am
Warning		OSM1-2	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 3, Index: 65	Mar 8, 2021 11:50:20 am
Warning		OSFAMILLE-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 2, Index: 65	Mar 8, 2021 11:50:01 am
Warning		OSJ8-3	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 2, Index: 65	Mar 8, 2021 11:49:40 am
Warning		OSMOD-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 3, Index: 65	Mar 8, 2021 11:49:11 am
Warning		OSJ60-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 9, Index: 67	Mar 8, 2021 11:48:35 am
Warning		OSMOE-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 6, Index: 66	Mar 8, 2021 11:48:13 am
Warning		OSM9-2	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 6, Index: 66	Mar 8, 2021 11:47:57 am
Warning		OSM9-2	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 4, Index: 66	Mar 8, 2021 11:47:57 am
Warning		OSDIACLONE-1	chassisTrapsAlert	fanFailed Fan is inoperable, Object: fan 1, Index: 65	Mar 8, 2021 11:47:42 am

Il y a 3 niveaux de notification en fonction de leur importance ici de la moins à la plus importante :

- Warning
- Major
- Critical

Mission : Détection des mini-switchs sur le réseau

On m'a demandé de détecter le nombre de mini-switchs qui sont connectés sur le réseau et sur quels ports ils sont présents car il y en a énormément et qu'ils engendrent souvent des problèmes.

J'ai particulièrement développé cette partie car cette mission m'a pris beaucoup de temps et j'ai beaucoup appris dans la logique à adopter pour les scripts.

J'ai donc récupéré les logs des adresses MAC présentes sur chaque port de chaque switch pour détecter ou il y en avait plusieurs en utilisant les scripts Omnivista pour envoyer un script avec les commandes correspondantes à chaque version de chaque switch Alcatel.

CLI Script

Send Script

Import Script

+

CLI Script List

Q

▼

Reset

Export to .csv

Print

Search ...

<input type="checkbox"/>	Filename	Description	Last Modified
<input checked="" type="checkbox"/>	mini_switch.script	<N/A>	Feb 9, 2021 2:19:26 PM
<input type="checkbox"/>	ntp snmp 6850.script	<N/A>	Dec 21, 2020 11:08:03 AM
<input type="checkbox"/>	shadmin_copy_certified_to_wo...	Copy Certified to Working(AOS...	Nov 11, 2020 4:40:14 AM
<input type="checkbox"/>	shadmin_copy_working_to_cer...	Copy Working to Certified (AO...	Nov 11, 2020 4:40:14 AM

Filename

mini_switch.script

Shared Admin Script

false

Description

<N/A>

Commands

show mac-address-table
show mac-learning
show system

J'avais ensuite les logs de 307 équipements présents dans Omnivista que j'ai exportés sur mon PC car j'ai créé mes scripts en fonction du résultat de ces logs qui comportent les commandes qui m'intéressent :

Logs

Q

▼

Reset

Export to .csv

Add to Report

Print

mini_switch|

<input type="checkbox"/>	Filename	IP Address	Name	Date
<input type="checkbox"/>	mini_switch-20210210_15-29-11.log	192.168.226.5	OSM62-1	Feb 10, 2021 3:29:22 PM
<input type="checkbox"/>	mini_switch-20210210_15-29-09.log	192.168.70.135	BIO-6850	Feb 10, 2021 3:29:19 PM
<input type="checkbox"/>	mini_switch-20210210_15-29-08.log	192.168.70.132	BIO-35	Feb 10, 2021 3:29:18 PM
<input type="checkbox"/>	mini_switch-20210210_15-29-08.log	192.168.70.131	BIO-3N	Feb 10, 2021 3:29:18 PM
<input type="checkbox"/>	mini_switch-20210210_15-29-06.log	192.168.70.130	BIO-25	Feb 10, 2021 3:29:17 PM
<input type="checkbox"/>	mini_switch-20210210_15-29-04.log	192.168.70.128	BIO-15	Feb 10, 2021 3:29:16 PM
<input type="checkbox"/>	mini_switch-20210210_15-29-05.log	192.168.70.129	BIO-2N	Feb 10, 2021 3:29:16 PM
<input type="checkbox"/>	mini_switch-20210210_15-28-57.log	192.168.70.127	BIO-1N	Feb 10, 2021 3:29:15 PM
<input type="checkbox"/>	mini_switch-20210210_15-28-55.log	192.168.70.124	BIO--15	Feb 10, 2021 3:29:13 PM
<input type="checkbox"/>	mini_switch-20210210_15-28-56.log	192.168.70.125	BIO-0N	Feb 10, 2021 3:29:13 PM
<input type="checkbox"/>	mini_switch-20210210_15-28-57.log	192.168.70.126	BIO--05	Feb 10, 2021 3:29:13 PM
<input type="checkbox"/>	mini_switch-20210210_15-28-52.log	192.168.70.120	TOIP-TB1-2	Feb 10, 2021 3:29:12 PM

Show: All

Showing All 307 rows

« < > »

Pour récupérer les informations qui m'intéresse dans chaque fichier j'ai dû créer un script avec PowerShell et donc apprendre un nouveau langage.

Dans ce script j'ai fait plusieurs parties en fonction des versions car il y en avait 3 différentes : les 6600, les 6850 et les 6860.

Ici 6860 :

```
OSDATACENTER-1> prompt string ->
-> prompt string ->
-> prompt string ->
-> no more
-> show mac-address-table
Legend: Mac Address: * = address not valid
```

Domain	Vlan/SrvId	Mac Address	Type	Protocol	Operation	Interface
VLAN	264	44:31:92:80:f6:15	learned	---	bridging	1/1
VLAN	264	5c:8a:38:4c:9d:76	learned	---	bridging	1/1
VLAN	306	5c:8a:38:4c:9d:78	learned	---	bridging	1/2
VLAN	306	d8:cb:8a:88:86:46	learned	---	bridging	1/2
VLAN	601	44:31:92:80:f6:14	learned	---	bridging	1/3
VLAN	601	5c:8a:38:4c:9d:79	learned	---	bridging	1/3

En effet, le résultat des commandes avait des différences et je ne pouvais pas trier les fichiers de la même manière pour tous.

Version	Colonne « Domain »	Colonne «Protocol »
6600	X	
6850		X
6860	X	X

```
15 | $chaine = Get-Content $file
16 | $reponse = $chaine | %{$_ -match "Domain"}
17 |
18 |
19 |
20 | if ($reponse -contains $true)
21 |
22 | {
23 |     $chaine2 = Get-Content $file
24 |     $reponse2 = $chaine2 | %{$_ -match "Protocol"}
25 |
26 |     if ( $reponse2 -contains $true)
27 |     {
28 |
29 |
30 |
31 |
32 |
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 | else
58 | {
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 | else
92 | {
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100 |
```

Pour chaque fichier j'ai donc gardé les lignes qui contenaient les informations sur les adresses MAC et je les ai mises dans un tableau en définissant les colonnes. J'ai ensuite récupéré le nom de chaque switch pour rajouter une colonne avec les noms.

6860 :

```
29 $content = Get-Content $file | where { $_ -clike "*VLAN*" } | foreach { ($_ -replace " +", ",") -replace "A,", ""}
30
31 $cols | Out-File $csv
32
33 $content | Out-File $csv -Append
34
35
36 $switchname = ((Get-Content $file | where {$_ -like "*Name*"} ) -replace " +Name: +", "") -replace ", ", ""
37 $result = Import-Csv -Path $csv | select *,@{Name='SwitchName';Expression={$switchname}}
38
```

6600 :

```
59 $content = Get-Content $file | where { $_ -clike "*VLAN*" } | foreach { ($_ -replace " +", ",") -replace "A,", ""}
60
61 $colonnes6600 | Out-File $csv6600
62
63 $content | Out-File $csv6600 -Append
64
65
66 $switchname = ((Get-Content $file | where {$_ -like "*Name*"} ) -replace " +Name: +", "") -replace ", ", ""
67 $result = Import-Csv -Path $csv6600 | select *,@{Name='SwitchName';Expression={$switchname}}
68
```

6850 :

```
94 $content = Get-Content $file | where { $_ -clike "*learned*" } | foreach { ($_ -replace " +", ",") -replace "A,", ""}
95
96 $colonnes | Out-File $csv5850
97
98 $content | Out-File $csv5850 -Append
99
100
101 $switchname = ((Get-Content $file | where {$_ -like "*Name*"} ) -replace " +Name: +", "") -replace ", ", ""
102 $result = Import-Csv -Path $csv5850 | select *,@{Name='SwitchName';Expression={$switchname}}
103
```

Ensuite j'ai dû enlever les lignes des adresses MAC des uplinks car sinon on aurait les adresses présentes sur les autres switches. Puis j'ai gardé les colonnes qui me donnaient des informations utiles.

6860 :

```
42 $array = $result | where { $_.Interface -notlike "O/*"}
43
44 $arraySorted = $array | Sort-Object -Property Interface, "Mac Address"
45
46 # $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1}
47 $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1} | foreach {
48     $int = $_.Name
49     $res = ($arraySorted | where { $_.Interface -eq $int} | Sort-Object -Property "Mac Address" -Unique)
50
51     if ($res.Count -ge 1) {
52         $res | select VlanId,"Mac Address",Interface,SwitchName | Export-Csv -Path $outCsv -NoTypeInfo -Append
53     }
54 }
55
```

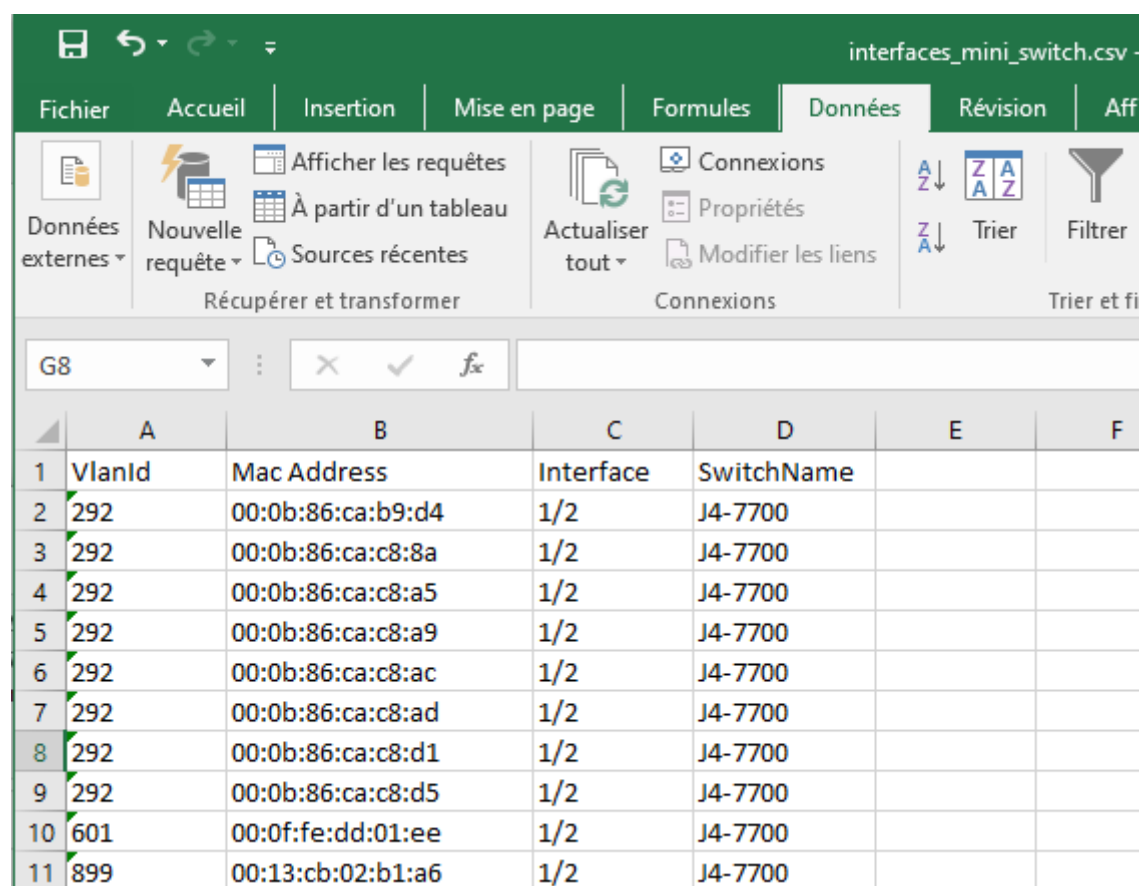
6600 :

```
72 $array = $result | where { $_.Interface -notlike "O/*"}
73
74 $arraySorted = $array | Sort-Object -Property Interface, "Mac Address"
75
76 $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1}
77 $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1} | foreach {
78     $int = $_.Name
79     $res = ($arraySorted | where { $_.Interface -eq $int} | Sort-Object -Property "Mac Address" -Unique)
80
81     if ($res.Count -ge 1) {
82         $res | select VlanId,"Mac Address",Interface,SwitchName | Export-Csv -Path $outCsv -NoTypeInfo -Append
83     }
84 }
85
```

6850 :

```
107 $array = $result | where { $_.Interface -notlike "0/*"}
108
109 $arraySorted = $array | Sort-Object -Property Interface, "Mac Address"
110
111 # $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1}
112 $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1} | foreach {
113     $int = $_.Name
114     $res = ($arraySorted | where { $_.Interface -eq $int} | Sort-Object -Property "Mac Address" -Unique)
115
116     if ($res.Count -ge 1) {
117         $res | select VlanId,"Mac Address",Interface,SwitchName | Export-Csv -Path $outCsv -NoTypeInformation -Append
118     }
119 }
```

Voici le résultat que j'ai eu avec la liste des adresses MAC sur chaque port de chaque switch.



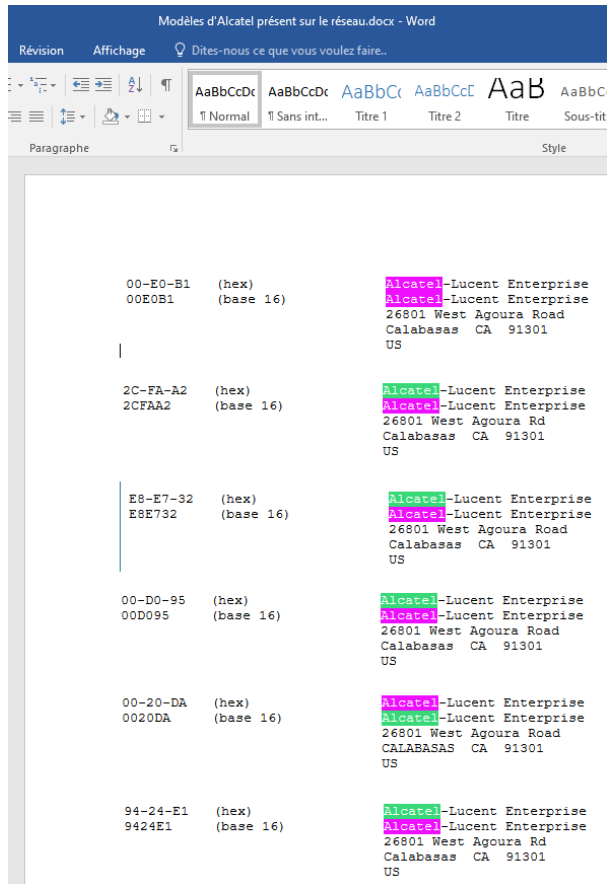
	A	B	C	D	E	F
1	VlanId	Mac Address	Interface	SwitchName		
2	292	00:0b:86:ca:b9:d4	1/2	J4-7700		
3	292	00:0b:86:ca:c8:8a	1/2	J4-7700		
4	292	00:0b:86:ca:c8:a5	1/2	J4-7700		
5	292	00:0b:86:ca:c8:a9	1/2	J4-7700		
6	292	00:0b:86:ca:c8:ac	1/2	J4-7700		
7	292	00:0b:86:ca:c8:ad	1/2	J4-7700		
8	292	00:0b:86:ca:c8:d1	1/2	J4-7700		
9	292	00:0b:86:ca:c8:d5	1/2	J4-7700		
10	601	00:0f:fe:dd:01:ee	1/2	J4-7700		
11	899	00:13:cb:02:b1:a6	1/2	J4-7700		

Et voici le script entier :

```
1 $File = ""
2 $csv = "U:\script\mac.csv"
3 $csv5850 = "U:\script\mac5850.csv"
4 $csv6600 = "U:\script\mac6600.csv"
5 $cols = "Domain,VlanId,Mac Address,Type,Protocol,Operation,Interface"
6 $colonnes = "VlanId,Mac Address,Type,Protocol,Operation,Interface"
7 $colonnes6600 = "Domain,VlanId,Mac Address,Type,Operation,Interface"
8 $outcsv = ".\script\interfaces_mini_switch.csv"
9
10
11
12 Get-ChildItem -Recurse -File -Path "U:\script\log" -Filter "*.log" | foreach {
13     $File = $_.FullName
14
15     $chaine = Get-Content $File
16     $reponse = $chaine | %{$_ -match "Domain"}
17
18
19     if ($reponse -contains $true)
20     {
21
22         $chaine2 = Get-Content $File
23         $reponse2 = $chaine2 | %{$_ -match "Protocol"}
24
25         if ($reponse2 -contains $true)
26         {
27
28             $content = Get-Content $File | where { $_ -clike "*VLAN*" } | foreach { ($_.replace " ", ",") -replace "\n", "" }
29
30             $cols | Out-File $csv
31
32             $content | Out-File $csv -Append
33
34
35             $switchname = ((Get-Content $File | where { $_ -like "*Name*" }) -replace " +Name: +", "" -replace ", ", ""
36             $result = Import-Csv -Path $csv | select *, @{Name='SwitchName'; Expression={$switchname}}
37
38             $result | ft
39
40
41             $array = $result | where { $_.Interface -notlike "0/*" }
42
43             $arraySorted = $array | Sort-Object -Property Interface, "Mac Address"
44
45             # $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1 }
46             $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1 } | foreach {
47                 $int = $_.Name
48                 $res = ($arraySorted | where { $_.Interface -eq $int } | Sort-Object -Property "Mac Address" -Unique)
49
50                 if ($res.Count -ge 1) {
51                     $res | select VlanId, "Mac Address", Interface, SwitchName | Export-Csv -Path $outcsv -NoTypeInfo -Append
52                 }
53             }
54         }
55     }
56 }
57 else
58 {
59     $content = Get-Content $File | where { $_ -clike "*VLAN*" } | foreach { ($_.replace " ", ",") -replace "\n", "" }
60
61     $colonnes6600 | Out-File $csv6600
62
63     $content | Out-File $csv6600 -Append
64
65
66     $switchname = ((Get-Content $File | where { $_ -like "*Name*" }) -replace " +Name: +", "" -replace ", ", ""
67     $result = Import-Csv -Path $csv6600 | select *, @{Name='SwitchName'; Expression={$switchname}}
68
69
70
71     $array = $result | where { $_.Interface -notlike "0/*" }
72
73     $arraySorted = $array | Sort-Object -Property Interface, "Mac Address"
74
75     # $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1 }
76     $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1 } | foreach {
77         $int = $_.Name
78         $res = ($arraySorted | where { $_.Interface -eq $int } | Sort-Object -Property "Mac Address" -Unique)
79
80         if ($res.Count -ge 1) {
81             $res | select VlanId, "Mac Address", Interface, SwitchName | Export-Csv -Path $outcsv -NoTypeInfo -Append
82         }
83     }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 {
93
94     $content = Get-Content $File | where { $_ -clike "*learned*" } | foreach { ($_.replace " ", ",") -replace "\n", "" }
95
96     $colonnes | Out-File $csv5850
97
98     $content | Out-File $csv5850 -Append
99
100
101     $switchname = ((Get-Content $File | where { $_ -like "*Name*" }) -replace " +Name: +", "" -replace ", ", ""
102     $result = Import-Csv -Path $csv5850 | select *, @{Name='SwitchName'; Expression={$switchname}}
103
104     $result | ft
105
106
107     $array = $result | where { $_.Interface -notlike "0/*" }
108
109     $arraySorted = $array | Sort-Object -Property Interface, "Mac Address"
110
111     # $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1 }
112     $arraySorted | Group-Object -Property Interface -NoElement | where { $_.Count -gt 1 } | foreach {
113         $int = $_.Name
114         $res = ($arraySorted | where { $_.Interface -eq $int } | Sort-Object -Property "Mac Address" -Unique)
115
116         if ($res.Count -ge 1) {
117             $res | select VlanId, "Mac Address", Interface, SwitchName | Export-Csv -Path $outcsv -NoTypeInfo -Append
118         }
119     }
120 }
121 }
122 }
```


Après ce résultat il me restait à trier les adresses MAC qui correspondaient à des switch Alcatel et qui faussait le résultat ainsi qu'à compter le nombre de mini-switchs.

J'ai donc fait une recherche pour connaître les 3 premiers octets des adresses MAC qui correspondaient aux switchs Alcatel afin de les exclure :



J'ai aussi enlevé tous les ports où il y avait plus de 8 adresses MAC car ils devaient correspondre à des ports d'uplinks :

```
1 $file = "U:\script\interfaces_mini_switch.csv"
2 $nbMiniSwitch = 0
3 $outCsv = "U:\script\tableau_mini_switch.csv"
4
5
6 Import-csv $file |
7 Group-Object -Property SwitchName, Interface | where {($_.Count -le 8) -and ( $_.Name -like "OS*" )} | foreach {
8     $alcatel = 0
9     foreach($line in $_.Group){
10         $mac = $line."Mac Address"
11         if ( ($mac -like "00:e0:81:*") -or ($mac -like "2c:fa:a2:*") -or ($mac -like "e8:e7:32:*") -or ($mac -like "00:d0:95:*") -or ($mac -like "00:20:da:*") -or ($mac -like "94:24:e1:*") ){
12             $alcatel = 1
13         }
14     }
15     if (-not $alcatel) {
16         $_
17     }
18 }
19 }
```

J'ai enlevé les ports qui correspondaient aux VLAN TOIP car il y a des PC branchés a des téléphones et donc il n'y a pas de mini-switchs dans ces cas.

```

20 foreach {
21     $oui = 0
22     foreach($line in $_.Group){
23         $vlan = $line."VlanId"
24         if (($vlan -like "22") -or ($vlan -like "26") -or ($vlan -like "62") -or ($vlan -like "234"))
25             $oui = 1
26     }
27 }
28 if (-not $oui) {
29     $nbMiniSwitch ++
30     $_
31 }

```

Après ce script j'avais donc la liste de tous les ports avec le nombre de mac address de chaque port ainsi que le nombre de mini-switchs.

465 mini-switchs

1	Name	Interfaces	Nombre de mac	
2	OSA1-1	3/22	2	
3	OSA2-1	1/10	3	
4	OSA2-1	1/2	2	
5	OSA3-1	1/2	2	
6	OSA3-1	1/20	3	
7	OSA3-1	1/23	2	
8	OSA3-1	2/20	3	
9	OSA3-1	2/6	3	
10	OSA3-1	3/19	2	
11	OSA3-1	3/24	8	
12	OSA3-1	3/7	4	
13	OSA3-1	3/8	3	
14	OSA3-1	4/19	3	
15	OSA3-1	5/22	3	
16	OSDIM-1	1/1	2	
17	OSDIM-1	1/12	2	
18	OSDIM-1	1/15	2	
19	OSDIM-2	1/10	2	
20	OSJ1-1	1/15	2	
21	OSJ1-1	1/17	5	
22	OSJ1-1	1/7	2	
23	OSJ1-1	2/11	3	
24	OSJ1-1	2/7	6	
25	OSJ1-1	3/12	2	
26	OSJ1-1	3/14	3	
27	OSJ1-1	3/16	2	
28	OSJ1-1	3/3	2	
29	OSJ2-1	1/13	2	
30	OSJ2-1	1/2	2	
31	OSJ2-1	2/18	2	

Mission : Installation nouveaux switchs

Il y a de nouveaux services qui arrivent dans un bâtiment et il faut une nouvelle pile de commutateur car il n'y a pas assez de ports. J'ai donc déballé, branché et configuré la pile de commutateur comme le stack précédent en mettant les ports qui concernaient le service déjà présent sur le switch 4.

J'ai donc mis la même adresse IP, la même passerelle, j'ai défini les VLAN et les VLAN par port, le nom du stack, l'environnement SNMP, les ports mobiles et démarrer le POE. J'ai ajouté la QoS, l'ULD (lien unidirectionnel et les traps sur les uplinks, la détection de loopback sur tous les ports, le protocole LLDP et les serveurs NTP.

```
-> show chassis
Local Chassis ID 1 (Master)
  Model Name:          OS6860E-P24,
  Module Type:         0x60e2205,
  Description:         Chassis,
  Part Number:         904059-90,
  Hardware Revision:   02,
  Serial Number:       JSZ201000553,
  Manufacture Date:    Mar  8 2020,
  Admin Status:        POWER ON,
  Operational Status:  UP,
  Number Of Resets:    5,
  MAC Address:         94:24:e1:3d:9a:d1

Remote Chassis ID 2 (Slave)
  Model Name:          OS6860E-P24,
  Module Type:         0x60e2205,
  Description:         Chassis,
  Part Number:         904059-90,
  Hardware Revision:   02,
  Serial Number:       JSZ201000529,
  Manufacture Date:    Mar  8 2020,
  Admin Status:        POWER ON,
  Operational Status:  UP,
  Number Of Resets:    5,
  MAC Address:         94:24:e1:3d:96:37

Remote Chassis ID 3 (Slave)
  Model Name:          OS6860E-P24,
  Module Type:         0x60e2205,
  Description:         Chassis,
  Part Number:         904059-90,
  Hardware Revision:   02,
  Serial Number:       JSZ201000537,
  Manufacture Date:    Mar  8 2020,
  Admin Status:        POWER ON,
  Operational Status:  UP,
  Number Of Resets:    5,
  MAC Address:         94:24:e1:3d:97:ff

Remote Chassis ID 4 (Slave)
  Model Name:          OS6860E-P24,
  Module Type:         0x60e2205,
  Description:         Chassis,
  Part Number:         904059-90,
  Hardware Revision:   02,
  Serial Number:       JSZ201000546,
  Manufacture Date:    Mar  8 2020,
  Admin Status:        POWER ON,
  Operational Status:  UP,
  Number Of Resets:    6,
  MAC Address:         94:24:e1:3d:9a:39
```

Ensuite nous sommes allés voir la baie de brassage où on pouvait mettre la pile puis nous l'avons installé dedans.

Malheureusement, nous n'avons pas pu connecter la pile au réseau directement car il n'y avait pas de fibre dans la baie. Nous avons donc demandé que quelqu'un intervienne pour tirer de la fibre jusqu'à cette baie et nous avons donc pu connecter la pile au réseau la semaine d'après.

Mission : Décâblage des baies inutilisées

Avec le déménagement qui se profilait, plusieurs baies étaient inutilisées et on m'a donc chargé d'enlever les câbles et de les trier pour toutes les baies inutilisées.

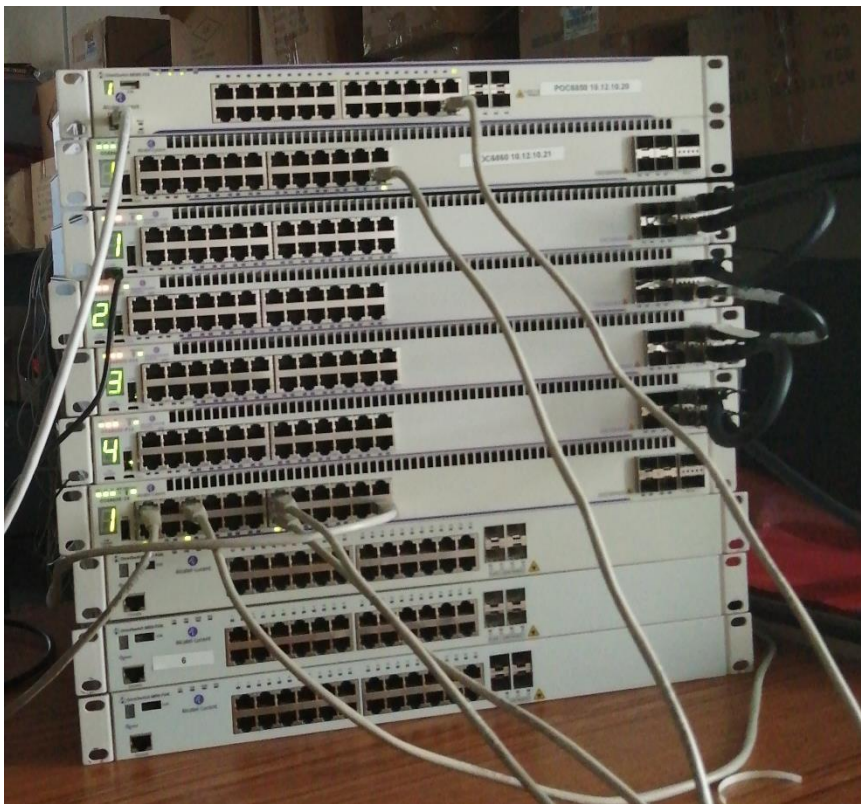
Configuration de switchs

J'ai configuré ma propre pile de switch avec

On m'a demandé de configurer des switchs à plusieurs reprises, je faisais une configuration simple avec les serveurs NTP, le serveur SNMP, l'adresse IP fournie, les VLAN demandés

Il pouvait y avoir 2 versions différentes : 6850 ou 6860.

Ci-dessous, de bas en haut, on peut voir ma première pile de commutateurs éteinte car plus utilisée, le commutateur qui me reliait au réseau, une pile de commutateur que j'ai configuré pour la dernière mission et d'autres commutateurs que l'on m'a demandé de configurer.



Installation de switchs dans les baies des nouveaux locaux

En prévision du déménagement je suis allé installer une dizaine de pile de switchs dans les baies du nouveau bâtiment en construction. On avait un ou deux cartons pour chaque baie avec une pile de deux ou trois switchs chacun.

On installait les switchs dans les baies, on branchait les alimentations, on branchait des SPF en fibre sur les premiers et derniers switchs de chaque pile et on branchait les switchs avec les câbles pour les stacker.

Conclusion

Ce stage est passé très vite grâce à l'équipe qui m'a très bien intégré et aux différentes missions qui m'ont été confiées et qui ont été très enrichissantes et qui m'ont permis d'accroître mes connaissances.

Toutefois ce stage m'a aussi fait réaliser que j'ai encore énormément à apprendre et qu'il me faudra redoubler d'efforts pour acquérir les connaissances et les compétences que requiert le métier de technicien réseau.