



**syslog-ng**

open source edition

31/03/2016


# La centralisation des Logs

Honvault Mickaël


# splunk® >

Mickaël Honvault

LYCEE DU PARC DE VILGENIS A MASSY(91)

Section : BTS SIO 2 <sup>ème</sup> année Pré-requis : Linux Commands	Sujet : La centralisation des Logs	
---	---------------------------------------	---

I.	Présentation .....	3
1.1.	Qui génère des logs ? .....	3
1.2.	Objectifs.....	3
II.	Contexte .....	4
III.	Infrastructure .....	5
IV.	Mise en place.....	5
1.1.	Cours Syslog-ng .....	5
1.1.1.	Objet source .....	6
1.1.2.	Les objets destinations .....	7
1.1.1.	Les objets Filter.....	8
1.1.2.	Les objets Log .....	10
1.2.	Client linux : Lamp -SSH .....	10
1.2.1.	Destination serveur Syslog-NG .....	10
1.2.2.	Déclaration du logging.....	11
1.3.	Splunk – SSH .....	11
1.3.1.	Déclarer source : .....	11
1.3.2.	Déclarer destination : .....	11
1.3.3.	Déclaration du logging.....	11
1.4.	Test .....	11
1.5.	Client Linux : lAmp.....	12
1.6.	Splunk – Apache2 .....	12
1.7.	MySQL.....	13
1.8.	Client Windows : Serveur AD.....	13
1.9.	Equipement Cisco .....	14
1.10.	Configuration Splunk-Cisco .....	15
V.	Serveur Syslog/Splunk .....	16
1.11.	Télécharger/Installer .....	16
1.12.	Ajout d'une source de données.....	20
VI.	Annexe.....	24
1.1.	Les différentes Facilités .....	24
1.2.	Les différents codes de gravités .....	25
1.3.	Rsyslog.....	26
1.3.1.	Client Rsyslog.....	26
1.3.2.	Serveur Rsyslog .....	28
1.3.3.	Test serveur Syslog .....	29

Section : BTS SIO 2 <sup>ème</sup> année Pré-requis : Linux Commands	Sujet : La centralisation des Logs	 Lycée Parc de Vilgénis
---	---------------------------------------	---

1.4.	Schéma des Alertes .....	30
------	--------------------------	----

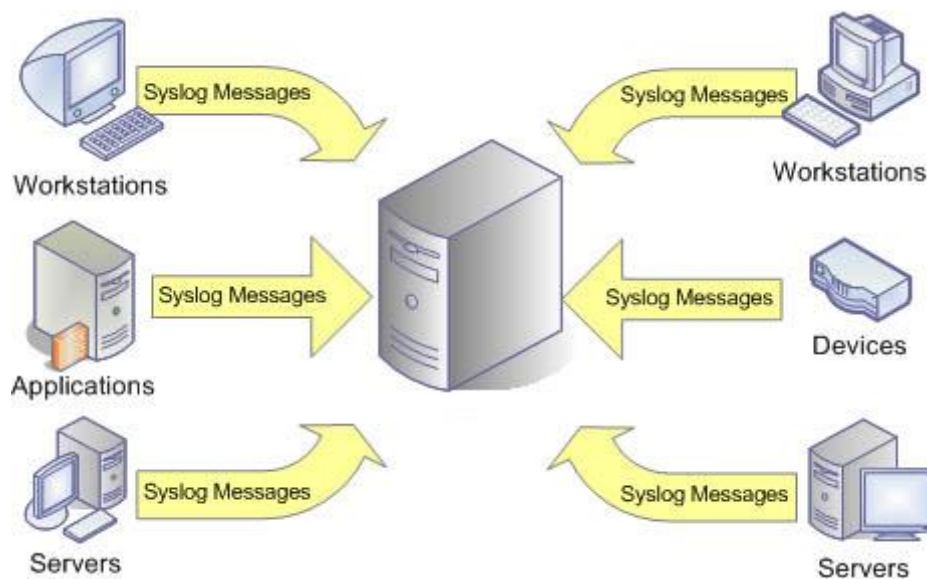
# I. Présentation

## 1.1. Qui génère des logs ?

Tout système informatique génère des logs ou autrement nommé des journaux, ces logs peuvent être aussi générés par le système qu'une application et celle-ci pourrait être codée par vous-même ! Mais il y a également les équipements qui génèrent des logs, switch, routeur, téléphone, firewall, ...

La question serait plutôt qui ne génère pas de log ? (... je n'ai pas la réponse à cette question ...)

## 1.2. Objectifs



Nous avons compris que tout le monde génère des logs, mais quel sont les buts de ces petits fichiers si important ?

- Dans le cas d'une analyse : Rassembler l'ensemble de ces journaux en un seul point permet de :
  - o Les retrouver plus facilement et éviter la multiplicité des agents distants
  - o Les scanner (faire des recherches dessus)
  - o Les comparer
- Dans le cas d'un crash : Si votre équipement à crasher, comment voulez-vous obtenir les logs ? (Comment ? pourquoi ? qui ? ... ?) Si vous ne centraliser pas les logs, vous vous asseyez sur des indices primordiaux pour rétablir votre système !
- Dans le cas d'un système de détection d'intrusion : Les logs en disent beaucoup plus sur le comportement de vos systèmes d'informations que votre antivirus, il se pourrait bien qu'une personne veuille exporter des communications d'un serveur vers un nouveau réseau. Comment voulez-vous le détecter ? Seule la possibilité d'une alerte sur le fait qu'un serveur échange des données avec un nouveau serveur peuvent attirer l'attention ... !

Durant vos recherches sur le net vous allez retrouver syslog, rsyslog, syslog-ng, php-syslog-ng, quelle est la différence entre tous ?

**Syslog** : C'est le protocole et également la première version du démon.

**Rsyslog** : C'est une version avancé de Syslog

**Syslog-ng : Next-Gen** (Next génération), les possibilités de cette plateforme sont bien plus facilement maintenable/manipulable car désormais tout est objet !

## II. Contexte

Dans notre cas, nous allons mettre en place 4 machines virtuelles :

- Client linux : LAMP (Debian)
- Client Cisco
- Client Windows : AD
- Serveur Syslog/Splunk (Debian)

**Penser à bien renommer vos machines sous linux : (/etc/hostname) et redémarrer votre VM et à vérifier que vos serveurs sont bien synchroniser sur le même serveur de temps.**

Je considère que vous êtes capable de trouver comment modifier le serveur de temps sous windows (ntp.unice.fr), voici la procédure sous Linux :

**Installer les paquet ntp, ntpdate :**

```
root@Splunk#apt-get install ntp ntpdate -y  
root@Splunk# service ntp stop
```

**Si toute fois vous doutez de votre fuseau horaire :**

```
root@Splunk:#dpkg-reconfigure tzdata
```

Editer le fichier de configuration /etc/ntp.conf :

```
#server 0.debian.pool.ntp.org iburst  
#server 1.debian.pool.ntp.org iburst  
#server 2.debian.pool.ntp.org iburst  
#server 3.debian.pool.ntp.org iburst  
server ntp.unice.fr
```

Lance la mise à jour et relancer le service ntp :

```
root@Splunk:# ntpdate-debian  
root@Splunk:# service ntp restart
```

Ou stopper les services ntp et taper la commande **ntpdate ntp.unice.fr** pour être positionner sur le serveur de temps de Nice puis relancer les services après avoir tapé : **ntpdate-debian**. Si toute fois le problème persiste : taper la commande suivante :

```
date -u 032922252016
```

03 = mois

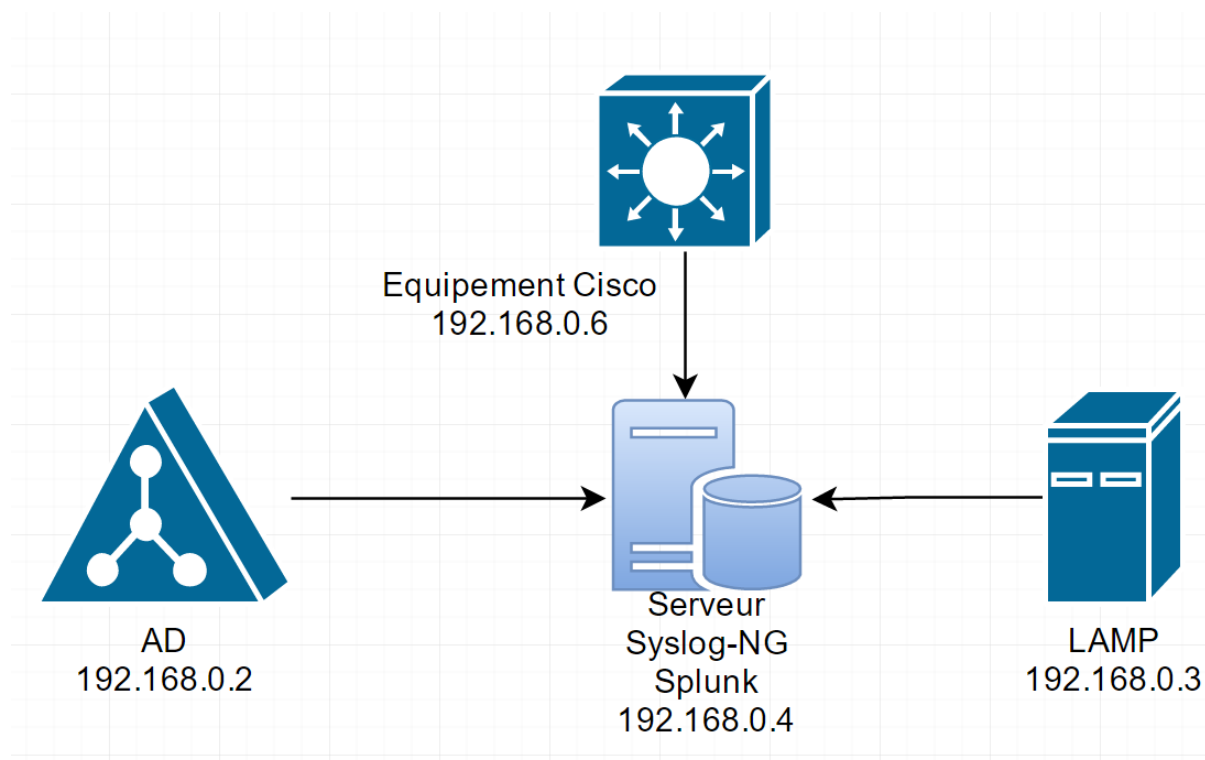
29 : jour

22 = heure

25 = minute

2016 = année

### III. Infrastructure



### IV. Mise en place

#### 1.1. Cours Syslog-ng

La première machine devra implémenter une architecture LAMP pour cela sur cette première machine nous allons installer les démons suivants :

- ssh
- apache
- mysql
- phpmyadmin
- syslog-ng

Nous allons commencer par accéder aux configurations (/etc/syslog-ng/syslog-ng.conf du serveur Syslog pour lui informer de l'existence du serveur Syslog-NG.

Le fichier syslog-ng.conf est découpé en quatre phases :

- Déclaration des objets sources
- Déclaration des objets destinations
- Déclaration des objets filtres
- Déclaration d'un log utilisant l'ensemble des objets sources, destinations et filtres

### 1.1.1. Objet source

La notion de source est le média d'arrivée d'un log, c'est-à-dire par quel moyen l'information va-t-elle arriver jusqu'au démon syslog-ng. Il en existe plusieurs sources tels que :

- Les flux locaux (= internal) // Récupération des logs générés par Syslog-ng
- Les flux réseaux (à travers les protocoles tcp et ou udp)
- Les flux provenant de socket (principalement unix) // Se positionne avant Syslog-ng dans la réception des logs
- ...

#### Déclaration d'une nouvelle source

```
source <identifier> { source-driver(params); source-driver(params); ... };
```

Exemple : Récupérer les sources de logs générés sur le socket /dev/log :

```
source s_localhost {  
    unix-stream("/dev/log");  
}
```

Dans la déclaration d'une source, il vous est possible de venir ajouter plusieurs sources, exemple :

```
source s_localhost {  
    internal();  
    unix-stream("/dev/log");  
}
```

Dans le cadre d'un serveur de centralisations des logs, les journaux arriveront à travers des flux tcp / udp, exemple d'utilisation :

```
source s_fluxTCP {  
    tcp(port(2514));  
}
```

```
source s_fluxUDP {  
    udp(port(514));  
}
```

Je peux également filtrer ma source avec l'adresse ip de l'émetteur est être plus précis :

```
source s_fluxLAMPClient {  
    udp(172.16.0.39 port(514));  
}
```

On constate bien à chaque fois le nom de l'objet est différent, il doit être clair on doit être capable à sa lecture de comprendre le rôle de l'objet. **Les objets sources sont toujours suffixés par s**

Faites bien attention certains équipements ne vous permettent pas de choisir le port, ni le protocole d'envoi, renseignez-vous auprès du constructeur de votre matériel pour configurer correctement votre serveur.

## 1.1.2. Les objets destinations

La déclaration de destination est très importante, car celle-ci va nous permettre de définir ce que nous allons faire de ce journal, voici quelques exemples d'utilisation :

- Stocker les logs dans des fichiers
- Transférer les logs vers un autre serveur
- ...

### **Déclaration d'une destination :**

```
destination <identifier> { destination-driver(params); destination-  
driver(params); ... };
```

Si on regarde notre fichier syslog-ng.conf, il existe déjà énormément de destination préconfigurée

```
destination d_auth { file("/var/log/auth.log"); };
```



Il est également possible de classer automatiquement les logs par année/mois/jours :

```
destination df_auth {  
    file("/var/log/$YEAR/$MONTH/$DAY/auth.log");  
    owner("root")  
    group("adm")  
    perm(0600)  
    create_dirs(yes));  
};
```

Ce qui nous permet d'obtenir un répertoire par année (ex : 2016) avec un répertoire mois à l'intérieur (ex : 03), avec un répertoire par jour (ex : 22) et l'ensemble des logs seront stocké à l'intérieur :

2016/03/22/auth.log

Nous aurions également pu construire une arborescence avec la chaîne '\$HOST' pour créer un répertoire par serveur.

**Ou vers un autre serveur :**

```
destination d_srvs syslog {tcp ("192.168.0.4" port(514)); };
```

### 1.1.1. Les objets Filter

Les objets filter permettent comme le nom l'indique d'appliquer des filtres à travers des expressions régulières ou la recherche de chaîne de caractère.

Le filtre est un peu la table de routage de syslog-ng. Un filtre permet d'identifier un "type" de logs pour ensuite lui appliquer une destination.

#### Déclaration d'un objet filter

```
filter <identifier> { expression; };
```

Un filtre peut être établi en fonction de la criticité et facilité d'un log ou encore filtrer sur le programme émetteur ou l'expression régulière

#### Exemple 1 :

```
filter f_firewall-PfSense{  
    match("Pfsense-IN")  
    or match("Pfsense-OUT")  
    or match("Pfsense-FWD");  
};
```

**Exemple 2 :**

```
filter f_demo_optimized_regexp {  
  
    program("demo_program") and  
  
    match("time error") and  
  
    match("is too large") and  
  
    match("set clock manually");  
};
```

**Exemple 3 :**

```
filter f_cp_ { host("10.28.88.4"); };
```

**Exemple 4 :**

```
filter f_iptables { match("^IPTABLES" value("MESSAGE")); };  
filter f_messages { not filter(f_iptables); };
```

**Exemple 5 :**

```
# Filter everything except regex keyword Shorewall  
filter f_shorewall { not match("regex" value("Shorewall")); };  
# Filter regex keyword Shorewall  
filter f_noshorewall { match("regex" value("Shorewall")); };  
  
filter f_grsecurity { match("^grsec" value("MESSAGE")); };
```

**Exemple 6 :**

<https://www.balabit.com/sites/default/files/documents/syslog-ng-ose-latest-guides/en/syslog-ng-ose-guide-admin/html/reference-filters.html>

**Exemple 7 :**

<http://eagain.net/articles/syslog-ng-chroot/>

## 1.1.2. Les objets Log

La déclaration des logs permet de faire le lien à travers les différents objets que nous avons créé, selon syslog-ng, un log est composé :

- D'une ou plusieurs sources
- D'un ou plusieurs filtres (optionnels)
- D'une ou plusieurs destinations

### Déclaration d'un objet log :

```
log {  
    source(s_localhost);  
    filter(f_auth);  
    destination(df_auth);  
};
```

## 1.2. Client linux : Lamp -SSH

Nous allons commencer à configurer notre client (file : **/etc/syslog-ng/syslog-ng.conf**) et ajouter une nouvelle source ainsi qu'une nouvelle destination. Vérifier qu'une source avec `system()` et `internal()` existe, sinon créez là.

### 1.2.1. Destination serveur Syslog-NG

Vous allez créer une destination (`d_srvSyslog`) de type `tcp` en direction de votre serveur Syslog (voir documentation création des destinations si besoin).

## 1.2.2. Déclaration du logging

Vous allez ajouter deux nouveaux logs :

- Un log SSH :
  - o Source : votre pc
  - o Filtre : Les comptes utilisateurs, afin d'envoyer les logs de connexion utilisateur SSH.
  - o Destination : Votre serveur de log

## 1.3. Splunk – SSH

### 1.3.1. Déclarer source :

Nous allons commencer par configurer notre client (file : `/etc/syslog-ng/syslog-ng.conf`) et ajouter une nouvelle source :

```
source s_network {  
    tcp(port(514));  
};
```

### 1.3.2. Déclarer destination :

La destination est déjà existante pour le SSH, cependant on aurait pu améliorer cette gestion en mettant en place une destination dans un répertoire précis du type :

`/Client/AAAA/MM/JJ/auth.log`

### 1.3.3. Déclaration du logging

Il nous reste à créer le lien entre notre source, notre filtre et notre destination :

```
log {  
    source(s_network);  
    filter(f_auth);  
    destination(ndf_auth);  
};
```

## 1.4. Test

Pensez à redémarrer vos services syslog-ng, puis depuis le poste client, générer des connexions ssh erroné et réussi, puis contrôler que les logs atterrissent bien dans votre serveur.

Si vous apercevez les logs, vous pouvez continuer vers la centralisation des logs pour apache2

## 1.5. Client Linux : lAmp

Depuis votre fichier de configuration de Syslog-NG

- Source Apache2

Vous allez créer une source (s\_apache2) de type fichier afin d'envoyer les logs d'erreur et d'appel de fichier vers notre serveur Syslog-NG. (Il vous sera nécessaire de créer une source avec plusieurs lignes pour les deux fichiers)

```
file("/var/log/monFichierLog " flags(no-parse));};
```

- Destination : Déjà existante (notre serveur syslog)
- Filtre : non nécessaire, car dans notre cas, nous prenons l'ensemble des fichiers
- Un log Apache
  - o Source : Votre pc
  - o Filtre : /
  - o Destination : Votre serveur de log

## 1.6. Splunk – Apache2

Analyser le fichier de log d'accès d'apache2, vous pouvez constater que sur une majorité des lignes, celle-ci est précédé par GET, nous allons réaliser un filtre simple. Toute entré en TCP avec contenant une ligne avec le mot GET sera envoyé vers un fichier isolé.

- Source : déjà existante
- Destination : déjà existante
- Filtre :

```
filter f_apache2 {match("GET");};
```

- Logging :

```
log {  
    source(s_network);  
    filter(f_apache2);  
    destination(d_apache2);  
};
```

## 1.7. MySQL

Je pense que vous avez compris le concept, je vous laisse en autonomie réfléchir sur comment mettre en place une gestion de log concernant MySQL !

## 1.8. Client Windows : Serveur AD

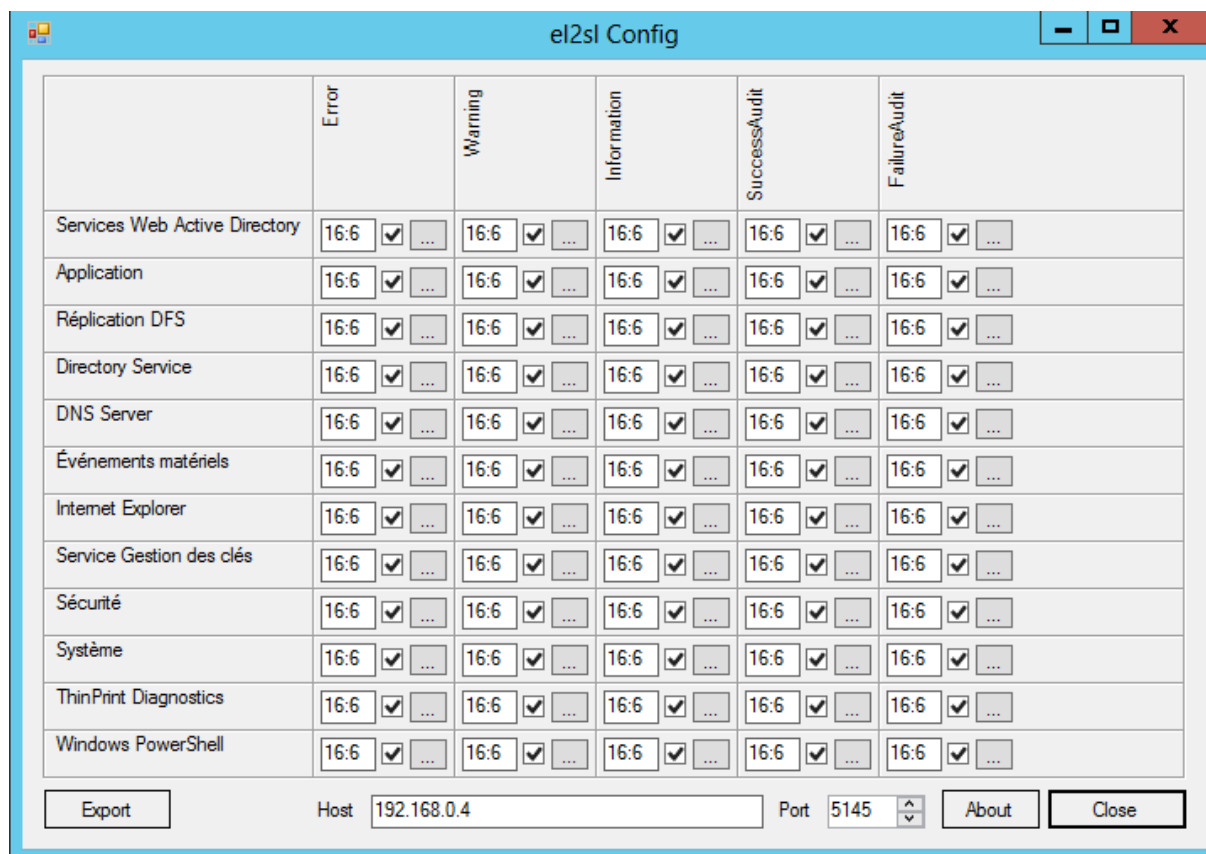
Pour cette partie, je vous laisse implémenter une machine Windows Serveur 2012 avec Active Directory, penser à renommer votre machine !

Il existe énormément d'outil pour envoyer nos logs vers un serveur de centralisation, dans notre cas nous allons utiliser el2sl.

Installer l'application el2sl (Event Log to SysLog) disponible à l'adresse suivante :

<https://sourceforge.net/projects/el2sl/files/el2sl/el2sl%20installer>

Lancer l'application :



	Error	Warning	Information	Success/Audit	Failure/Audit
Services Web Active Directory	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Application	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Réplication DFS	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Directory Service	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
DNS Server	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Événements matériels	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Internet Explorer	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Service Gestion des clés	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Sécurité	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Système	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
ThinPrint Diagnostics	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...
Windows PowerShell	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...	16:6 <input checked="" type="checkbox"/> ...

Export Host 192.168.0.4 Port 5145 About Close

Cocher quelques éléments clé, tel que les éléments liait à l'Active Directory, puis mettez en place une plateforme pour la réception des logs provenant de notre machine AD. (5145 = votre port UDP)

```
#L'objet source :
source s_network {
    tcp(port(5142));
    udp(port(5145));
};

#L'objet destination :
destination d_AD{
    file("/var/log/ad.log"
    owner("root")
    group("adm")
    perm(0600)
    create_dirs(yes));
};

# L'objet Log :
log { source(s_network); destination(bordel);};
```

Afin de générer des logs, vous pouvez créer un compte utilisateur ou encore renommer un compte utilisateur.

## 1.9. Equipement Cisco

Il est important de noter que l'horodatage, c'est à dire l'heure que vont avoir les journaux exportés a une importance particulière dans le système de centralisation des logs, nous allons donc commencer par là :

```
Cisco#clock set 02:08:00 march 29 2016
```

Ensuite on active l'horodatage des logs :

```
Cisco(config)#service timestamps
```

Puis on paramètre les informations sur le serveur qui recevra les logs :

```
Cisco(config)#logging 192.168.0.4 transport udp port 5142
```

Puis on précise la facility utilisé pour pouvoir réaliser un filtre :

```
Cisco(config)#logging facility local7
```

Nous allons également pouvoir définir les logs que nous souhaitons recevoir :

- 7 – **debugging** (nous allons utiliser celui-ci, le log le plus parlant).
- 6- **informational**
- 5 – **notifications**
- 4 – **warnings**
- 3 – **errors**
- 2 – **critical**
- 1 – **alerts**
- 0 – **emergencies**

```
Cisco(config)#logging trap debugging
```

## 1.10. Configuration Splunk-Cisco

```
# L'objet source :
source s_network {
    udp(port(5142)) ;
};

# L'objet destination :
destination d_cisco{
    file("/var/log/cisco.log"
    owner("root")
    group("adm")
    perm(0600)
    create_dirs(yes)) ;
};

# L'objet filter :
filter f_localCisco {facility(local7)};

# L'objet log :
log {
    source(s_network) ;
    filter(f_localCisco) ;
    destination(d_cisco) ;
};
```



## V. Serveur Syslog/Splunk

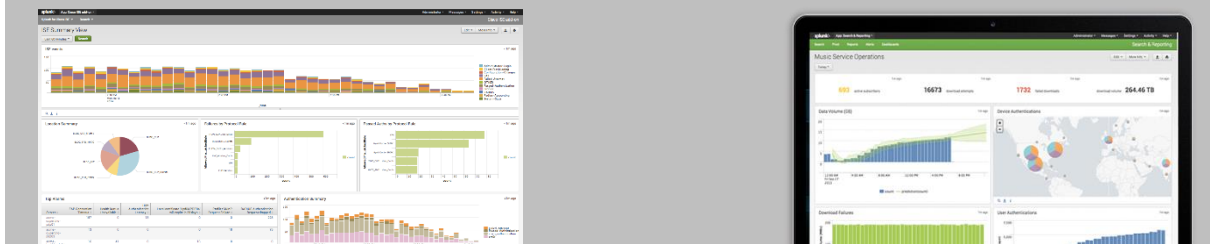
Une machine serveur sous Debian 8, avec pour particularité d'être le serveur de centralisation des logs (syslog-ng) et d'être la plateforme de traitement des logs.



**Splunk** est une entreprise américaine multinationale dont le siège est situé à San Francisco. Elle produit des **logiciels de collecte et d'analyse de données orientées "big data"**, accessibles via une interface web.

Splunk **indexe en temps réel des données issues de machines** (logs, web services, configurations, équipements télécom, GPS, capteurs,...). Les utilisations vont de la **sécurité** (corrélation, analytics, fraude...) à la **supervision d'infrastructure**, en passant par le **reporting métier**.

Celui-ci est concurrent à la suite d'outils SELKS (Suricata, **ElasticSearch**, **Logstash**, **Kibana**, Scirius), ou encore Onion, Plaso, Graylog2, ....



Commencer par installer ssh pour simplifier l'installation de la machine et favoriser les copier-coller.

Accéder à votre serveur via votre remote app préféré, il n'y aura que deux étapes pour la mise en place :

- Configuration du serveur Syslog
- Téléchargement, installation de Splunk

L'aspect utilisation de Splunk sera apprécié dans une autre partie du document.

### 1.11. Télécharger/Installer

#### Télécharger le paquet :

Aller sur le site de Slunk, rubrique téléchargement (Free Splunk), prenez une version entreprise, puis choisissez selon votre version de Linux une 64bits ou 32bits. (il vous faudra créer un compte pour accéder au chemin de téléchargement. Vous aurez alors accès à la page suivante :

**Your download is starting...**

Download not starting? Please use [this URL](#).

Got wget? [Get this URL](#).

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click [here](#) to select the entire command.

```
wget -O splunk-6.3.3-f44afce176d0-linux-2.6-intel.deb  
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86&platform=linux&version=6.3.3&product=splunk&filename=splunk-6.3.3-f44afce176d0-linux-2.6-intel.deb'
```

Education  
[Training and Certification](#)

Professional Services  
[Bring in the Experts](#)

Il ne vous reste plus qu'à copier-coller le lien via MRemote.

```
root@Splunk:/opt# wget  
http://download.splunk.com/products/splunk/releases/6.3.3/splunk/linux/splunk-6.3.3-f44afce176d0-linux-2.6-amd64.deb
```

### Installer le paquet Splunk :

```
root@Splunk:/opt# dpkg -i splunk-6.3.3-f44afce176d0-linux-2.6-intel.deb
```

### Activer le démarrage automatique (boot-start)

```
root@Splunk:/opt/splunk/bin# ./splunk enable boot-start
```

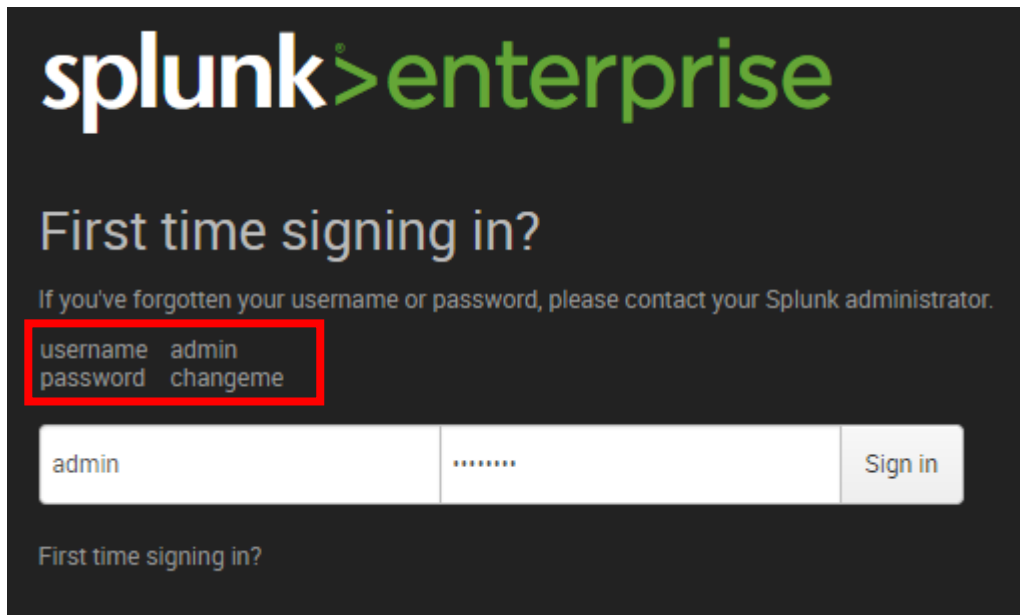
Ou sinon, vous pouvez utiliser le cron ou encore l'update-rc.d !

### Lancer le service Splunk :

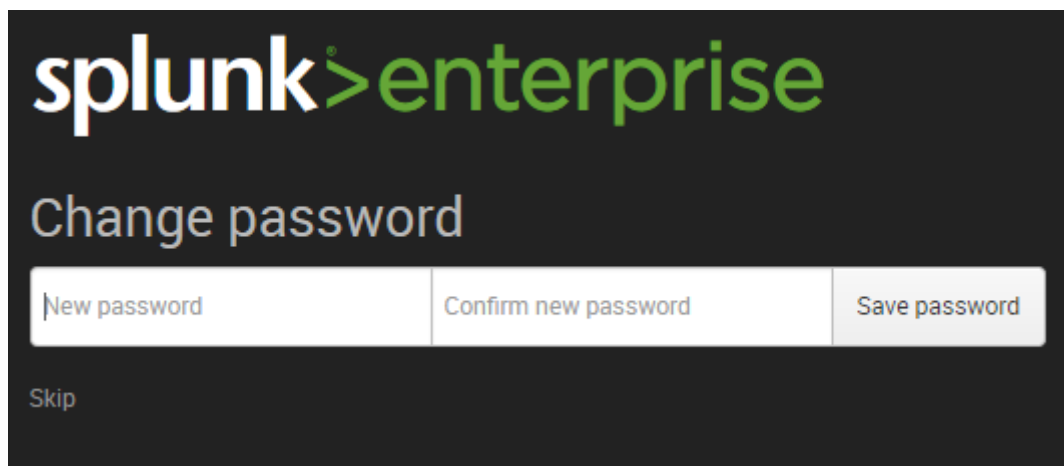
```
root@Splunk:/opt/splunk/bin# ./splunk start
```

Le port par défaut est le port : 8000 en http, il nous reste plus qu'à nous connecter et à commencer la configuration :

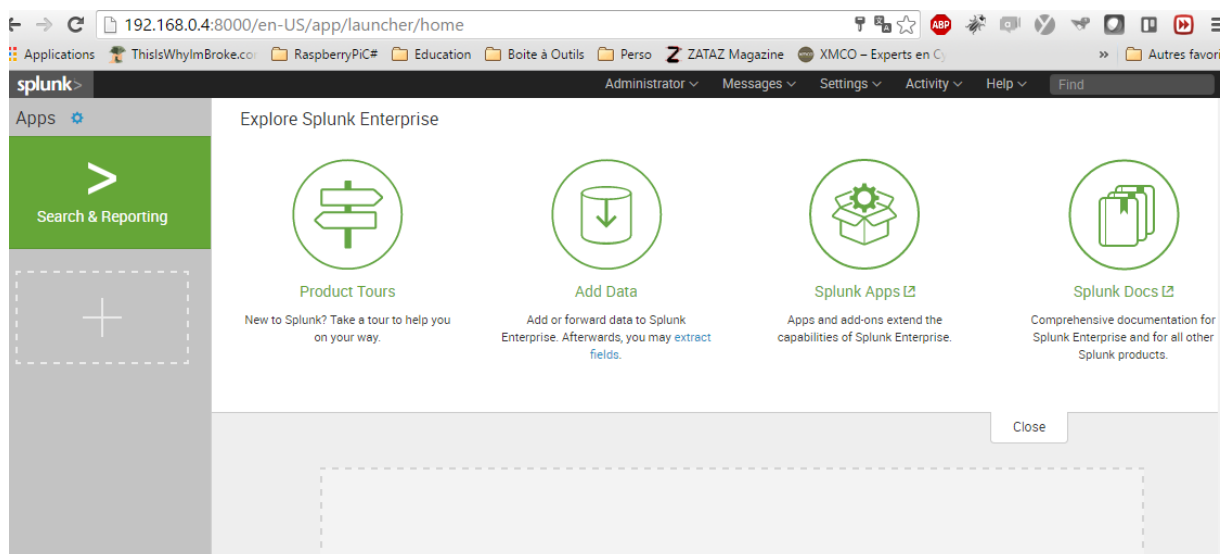
<http://192.168.0.4:8000/> avec les identifiants : admin/changeme



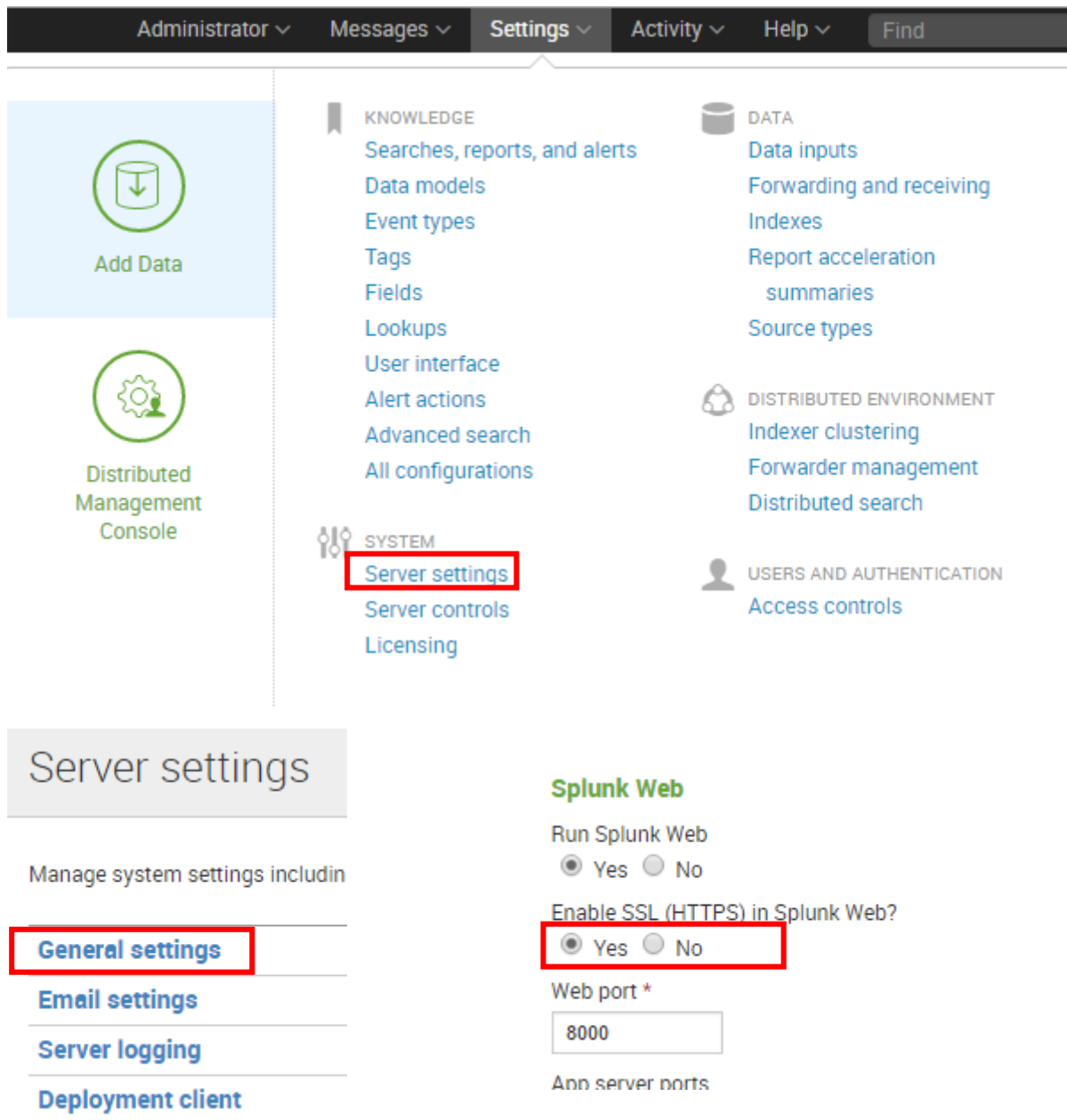
La plateforme vous propose de changer de mot de passe :



Nous voilà sur notre première connexion à la plateforme :



Vous allez pouvoir activer le HTTPS de la façon suivante :



Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

**KNOWLEDGE**  
Searches, reports, and alerts  
Data models  
Event types  
Tags  
Fields  
Lookups  
User interface  
Alert actions  
Advanced search  
All configurations

**DATA**  
Data inputs  
Forwarding and receiving  
Indexes  
Report acceleration summaries  
Source types

**DISTRIBUTED ENVIRONMENT**  
Indexer clustering  
Forwarder management  
Distributed search

**SYSTEM**  
**Server settings**  
Server controls  
Licensing

**USERS AND AUTHENTICATION**  
Access controls

## Server settings

Manage system settings including

- General settings**
- Email settings
- Server logging
- Deployment client

### Splunk Web

Run Splunk Web  
☒ Yes ☐ No

Enable SSL (HTTPS) in Splunk Web?  
☒ Yes ☐ No

Web port \*  
8000

Ann server ports

Si vous voulez passer l'application en Français : <http://192.168.0.4:8000/fr-FR/>

## 1.12. Ajout d'une source de données



### Présentations des produits

Nouveau sur Splunk ? Découvrez nos présentations.



### Ajouter des données

Ajouter ou transmettre des données à Splunk Enterprise. Par la suite, vous pourrez [extraire des champs](#).



### Splunk Apps [L2](#)

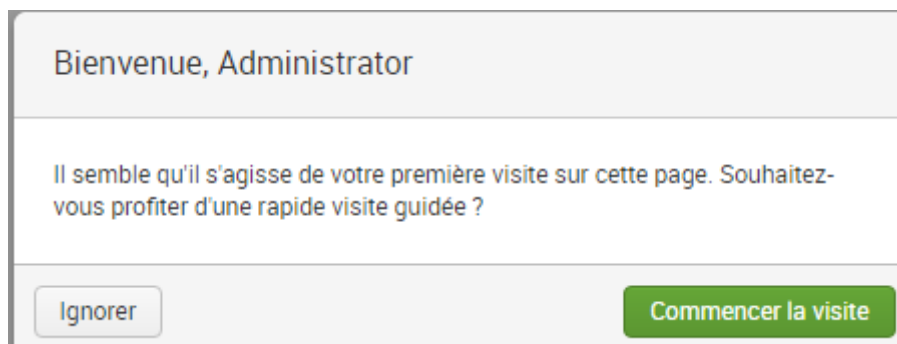
Les applications et les extensions développent les capacités de Splunk Enterprise.



### Splunk Docs [L2](#)

Une documentation complète pour Splunk Enterprise et les autres produits Splunk.

Cliquer sur ajouter des données, puis découvrez le fonctionnement de splunk avec la visite guidée :



### télécharger

des fichiers depuis mon ordinateur

Fichiers de logs locaux  
Fichiers structurés locaux (par ex : CSV)  
[Didacticiel pour ajouter des données](#)



### surveiller

des fichiers et des ports sur cet indexeur Splunk

Fichiers - WMI - TCP/UDP - Scripts  
Entrées modulaires pour des sources de données externes



### transmettre

des données à partir du forwarder Splunk

Fichiers - TCP/UDP - Scripts  
[M'aider à installer l'universal forwarder](#)

Ajouter des données

Sélectionnez une source Paramètres d'entrée Revoir Fait

**Fichiers et répertoires**  
Télécharger un fichier, indexer un fichier local, ou surveiller un répertoire entier.

**Collecteur d'événements HTTP**  
Configurer les variables que les clients pourront utiliser pour transmettre des données à l'aide du protocole HTTP ou HTTPS.

**TCP/UDP**  
Configurez Splunk pour écouter un port réseau.

**Scripts**  
Récupérez des données depuis un service API ou une base de données à l'aide d'un script.

Configurez cette instance pour écouter tous les ports TCP ou UDP pour capturer des données envoyées sur le réseau (telles que syslog). [En savoir plus](#)

TCP UDP

Port ?  
Exemple : 514

Remplace le nom de la source ? facultatif  
Hôte : port

Accepter uniquement une connexion de ? facultatif  
exemple : 10.1.2.3, !badhost.splunk.com, \*.splunk.com

Donner le nom de votre port udp : 5145 et laisser les autres éléments vides, nous allons récupérer les sources de notre AD et de notre routeur Cisco.

Dans les paramètres d'entrée, réaliser une recherche sur Syslog et scroller l'ascenseur pour trouver Syslog.

## Paramètres d'entrée

Vous pouvez également configurer des paramètres d'entrée supplémentaires pour cette entrée de données de la manière suivante :

### Type de source

Le type de source est l'un des champs par défaut que Splunk affecte à toutes les données entrantes. Il indique à Splunk le type de données dont vous disposez, de sorte qu'il est en mesure de les formater de manière intelligente pendant l'indexation. Et cela représente une manière de classer vos données afin que vous puissiez y effectuer facilement des recherches.

### Contexte de l'app

Les contextes d'application sont des dossiers à l'intérieur d'une instance Splunk, qui contiennent des configurations pour un cas d'utilisation spécifique ou un domaine de données. Les contextes d'application permettent d'accroître la capacité à gérer les définitions des entrées et types de sources. Splunk charge tous les contextes d'applications en fonction de règles de priorité. [En savoir plus](#)

### Hôte

Lorsque Splunk indexe des données, chaque événement reçoit une valeur d'« hôte ». La valeur d'hôte doit être le nom de l'appareil d'où provient l'événement. Le type d'entrée choisi détermine les choix de configurations disponibles. [En savoir plus](#)

Sélectionner ou nouveau

syslog

syslog

anaconda\_syslog

cisco\_syslog

linux\_messages\_syslog  
Format found within the Linux log file  
/var/log/messages

postfix\_syslog  
Output produced by the Postfix email  
server

sendmail\_syslog  
Output produced by the Sendmail email  
server

Personnaliser

Terminer votre ajout en sélectionnant la méthode IP :

Contexte de l'application

Méthode?

Index  [Créer un nouvel index](#)

Visualisation avant validation :

Ajouter des données

☒ Revoir ☐ Fait

Revoir

Types d'entrée	Port UDP
Numéro de port	5145
Remplace le nom de la source	S/O
Limiter à l'hôte	S/O
Type de source	syslog
Contexte de l'application	search
Hôte	(adresse IP du serveur distant)
Index	default

Retourner au menu principale et vous pouvez constater que vos journaux apparaissent désormais sur la plateforme :

[Recherche](#) [Pivot](#) [Rapports](#) [Alertes](#) [Tableaux de bord](#) [Search](#)

### Recherche

#### Comment chercher

Si vous ne connaissez pas bien la recherche dans Splunk ou si vous souhaitez en savoir plus, consultez l'une des ressources suivantes.

[Documentation](#) [Tutoriel](#)

#### Que chercher

490 Événements INDEXÉ

6 hours ago PREMIER ÉVÉNEMENT

6 hours ago DERNIER ÉVÉNEMENT

[Résumé des données](#)

Vous pouvez ensuite filtrer le tout par adressage hôte :

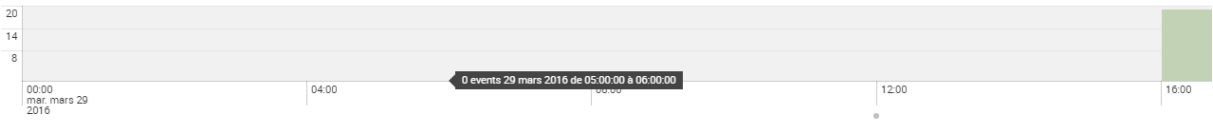
### Résumé des données

[Hôtes \(2\)](#) [Sources \(1\)](#) [Sourcetypes \(1\)](#)

Hôte		Nombre	Dernière mise à jour
192.168.0.2		549	29/03/16 16:16:07,000
192.168.0.6		8	29/03/16 16:15:48,000

[Événements \(4\)](#) [Patterns](#) [Statistiques](#) [Visualisation](#)

[Mettre en forme la chronologie](#) [Zoom arrière](#) [Zoom sur la sélection](#) [Annuler la sélection](#)



[Liste](#) [Format](#) [20 par page](#)

[Masquer les champs](#) [Tous les champs](#)

Champs sélectionnés

- host 1
- source 1
- sourcetype 1

Champs intéressants

- index 1
- linecount 1

i	Période	Événement
>	29/03/16 22:49:23,000	Mar 29 22:49:23 192.168.0.6 60: Oct 29 22:49:13.007: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
>	29/03/16 22:49:23,000	Mar 29 22:49:23 192.168.0.6 59: Oct 29 22:49:12.007: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
>	29/03/16 22:49:18,000	Mar 29 22:49:18 192.168.0.6 58: Oct 29 22:49:08.207: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
>	29/03/16 22:49:18,000	Mar 29 22:49:18 192.168.0.6 57: Oct 29 22:49:07.207: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

Il ne vous reste plus qu'à vous essayer à la conception de tableau de bord ! 😊

Amusez-vous bien !



## VI. Annexe

### 1.1. Les différentes Facilities

Codes de catégorie			
Code	Mot-clé	Description	
0	kern	kernel messages	Utilisé pour les messages concernant le kernel
1	user	user-level messages	Facilités par défaut quand aucune n'est spécifiée
2	mail	mail system	Utilisé pour les événements des services mail
3	daemon	system daemons	Utilisé par les différents processus systèmes et d'application
4	auth	security/authorization messages	Utilisé pour des événements concernant la sécurité ou l'authentification à travers des applications d'accès (type SSH)
5	syslog	messages generated internally by syslogd	
6	lpr	line printer subsystem	
7	news	network news subsystem	
8	uucp	UUCP subsystem	
9		clock daemon	
10	authpriv	security/authorization messages	Utilisé pour les messages relatifs au contrôle d'accès
11	ftp	FTP daemon	
12	-	NTP subsystem	
13	-	log audit	
14	-	log alert	
15	cron	clock daemon	
16	local0	local use 0 (local0)	
17	local1	local use 1 (local1)	

18	local2	local use 2 (local2)	
19	local3	local use 3 (local3)	
20	local4	local use 4 (local4)	
21	local5	local use 5 (local5)	
22	local6	local use 6 (local6)	
23	local7	local use 7 (local7)	Utilisé pour les messages du boot

Autres possibilités :

- \* : Désigne toutes les facilities, par soucis de simplicité c'est ce que nous avons spécifié lors de notre première règle de redirection des logs un peu plus haut
- **none** : Désigne aucune facilities

## 1.2. Les différents codes de gravités

Codes de gravité			
Code	Gravité	Mot-clé	Description
0	Emergency	emerg (panic)	Urgence, <a href="#">système</a> inutilisable
1	Alert	alert	Alerte. Intervention immédiate nécessaire
2	Critical	crit	Erreur critique pour le système.
3	Error	err (error)	Erreur de fonctionnement.
4	Warning	warn (warning)	Avertissement (une erreur peut intervenir si aucune action n'est prise).
5	Notice	notice	Événement normal méritant d'être signalé.
6	Informational	info	Pour information.
7	Debugging	debug	Message de débogage

## 1.3. Rsyslog

### 1.3.1. Client Rsyslog

Maintenant que tout est activé, accédons au fichier rsyslog.conf pour y activer la redirection des logs vers notre serveur. (Accéder au fichier /etc/rsyslog.conf)

Dans la partie règle vous pouvez ajouter la ligne suivante qui vous permettra de rediriger l'ensemble des logs sur votre serveur :

Pour transférer les logs en TCP : (attention au port utilisé, il faudra mettre le même sur le serveur)

```
*.* @@IP_SERVEUR:2514
```

Pour transférer les logs en UDP : (attention au port utilisé, il faudra mettre le même sur le serveur)

```
*.* @IP_SERVEUR:514
```

\* = Pour chaque facilité

. = Nous allons choisir

\* = les niveaux de gravités à transférer → \*.\* Tous les logs !

@ = UDP | @@ = TCP

IP\_Serveur = IP du serveur de centralisation des logs

2514/514 = Port de destination du serveur de centralisation des logs. (A voir dans la configuration du serveur)

Exemple :

```
*.* @172.16.0.38:514
```

Après avoir modifié ce fichier vous penserez à redémarrer le service rsyslog :

```
root@debian# service rsyslog restart
```

Il serait également intéressant de renommer le nom de votre machine linux (/etc/hostname, puis redémarrer !).

- **Rediriger ou copier selon la facilitie ou la priorité**

Sous Linux quand on parle de gestion de logs, les facilites sont des catégories dans lesquelles les logs vont se “ranger” afin de mieux les archiver et les trier. Parmi ces facilites, on retrouve par exemple (voir annexe « Les différentes Facilities »).

En plus de ces facilites, nous retrouvons pour chaque facilitie un niveau de gravité (appelé Priorité) qui va du plus grave à la plus simple information (voir annexe sur « Les codes de gravités »).

les logs qui nous intéressent et donc ceux que l’on va rediriger. Par exemple si l’on cherche à rediriger vers notre serveur de logs 172.16.0.38 uniquement les messages critiques et supérieurs concernant les mails sur le port UDP 514, on ajoutera la ligne suivante :

```
mail.err @172.16.0.38:514
```

On peut également rediriger tous les logs mails :

```
mail.* @172.16.0.38:514
```

On peut également saisir en une ligne plusieurs types de facilities et de priorité, on trouve par exemple dans le fichier de configuration par défaut les lignes suivantes :

```
*.=debug;\n    auth,authpriv.none;\n    news.none;mail.none    -/var/log/debug\n\n*.=info;*.=notice;*.=warn;\n    auth,authpriv.none;\n    cron,daemon.none;\n    mail,news.none        -/var/log/messages
```

On voit ici que toutes les priorités debug sont redirigées vers le fichier “/var/log/debug” et que toutes les priorités **info**, **notice** et **warn** seront dans “/var/log/messages”. Pour que ces filtres soient redirigés vers le serveur de logs, il suffit de spécifier l’IP du serveur ainsi que son port comme fait plus haut à la place du nom du fichier.

- **Rediriger les logs vers un dossier/fichier par host**

On peut également, pour faciliter la hiérarchisation et l’archivage de nos logs lorsque l’on a un grand nombre de client Rsyslog utiliser une arborescence avec un dossier/fichier par hôte plutôt que de mettre tous les logs dans le même fichier que le serveur de logs. On va pour cela utiliser une template que nous mettrons après le bloc “**RULES**” dans le fichier de configuration du serveur :

```
$template syslog,»/var/log/clients/%fromhost%/syslog.log"
```

On va ensuite appliquer ce template à tous les logs entrants :

```
*.* ?syslog
```

Il nous suffira ensuite de redémarrer notre service rsyslog puis de générer des logs depuis les clients. On se retrouvera alors avec un dossier `"/var/log/clients/"` contenant un dossier par IP/nom client et contenant respectivement un fichier `"syslog.log"` avec les logs de chaque client respectif, ce qui simplifie la recherche d'information dans les logs d'un client précis

### 1.3.2. Serveur RSyslog

Etant données que rsyslog est par défaut installer sur les machines debian, il faudra informer le démon de sa capacité à recevoir des messages provenant de différentes sources. Cette étape se configure dans le fichier `rsyslog.conf` :

```
root@Splunk:~# nano /etc/rsyslog.conf
```

Nous allons venir décommenter les lignes suivantes :

```
# provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514
```

Pour obtenir :

```
# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 2514
```

Il est important de faire attention aux ports que vous utilisez ! Si vous activez la réception TCP et UDP, votre serveur devra ouvrir deux sockets différentes, il a donc besoin de deux ports ! D'ailleurs, il faudra prendre en compte cette modification sur vos configurations de Syslog.

Il est tout à fait possible de choisir d'envoyer certains logs en TCP et d'autres en UDP en fonction de la criticité des logs.

### 1.3.3. Test serveur Syslog

Votre serveur de log est désormais prêt à recevoir des logs, vous pouvez vérifier que cela est fonctionnel.

**Première vérification :** Vérifier que les sockets sont bien ouvertes avec la commande :

```
root@Splunk:/# netstat -npltu
```

Résultat :

```
root@Splunk:/# netstat -npltu
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 383/rpcbind
tcp 0 0 0.0.0.0:2514 0.0.0.0:* LISTEN 839/rsyslogd
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 412/sshd
tcp 0 0 0.0.0.0:33686 0.0.0.0:* LISTEN 396/rpc.statd
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 677/exim4
tcp6 0 0 :::111 :::* LISTEN 383/rpcbind
tcp6 0 0 :::2514 :::* LISTEN 839/rsyslogd
tcp6 0 0 :::39955 :::* LISTEN 396/rpc.statd
tcp6 0 0 :::22 :::* LISTEN 412/sshd
tcp6 0 0 :::1:25 :::* LISTEN 677/exim4
udp 0 0 0.0.0.0:514 0.0.0.0:* 839/rsyslogd
udp 0 0 0.0.0.0:68 0.0.0.0:* 692/dhclient
udp 0 0 0.0.0.0:33119 0.0.0.0:* 396/rpc.statd
udp 0 0 0.0.0.0:111 0.0.0.0:* 383/rpcbind
udp 0 0 0.0.0.0:5799 0.0.0.0:* 692/dhclient
udp 0 0 0.0.0.0:982 0.0.0.0:* 383/rpcbind
udp 0 0 127.0.0.1:996 0.0.0.0:* 396/rpc.statd
udp6 0 0 :::514 :::* 839/rsyslogd
udp6 0 0 :::3126 :::* 692/dhclient
udp6 0 0 :::111 :::* 383/rpcbind
udp6 0 0 :::982 :::* 383/rpcbind
udp6 0 0 :::56812 :::* 396/rpc.statd
```

**Deuxième vérification :**

Vous pouvez générer des connexions réussi et erroné sur votre client et vérifier que les fichiers de log (/var/log/auth.log) sont bien alimenté.

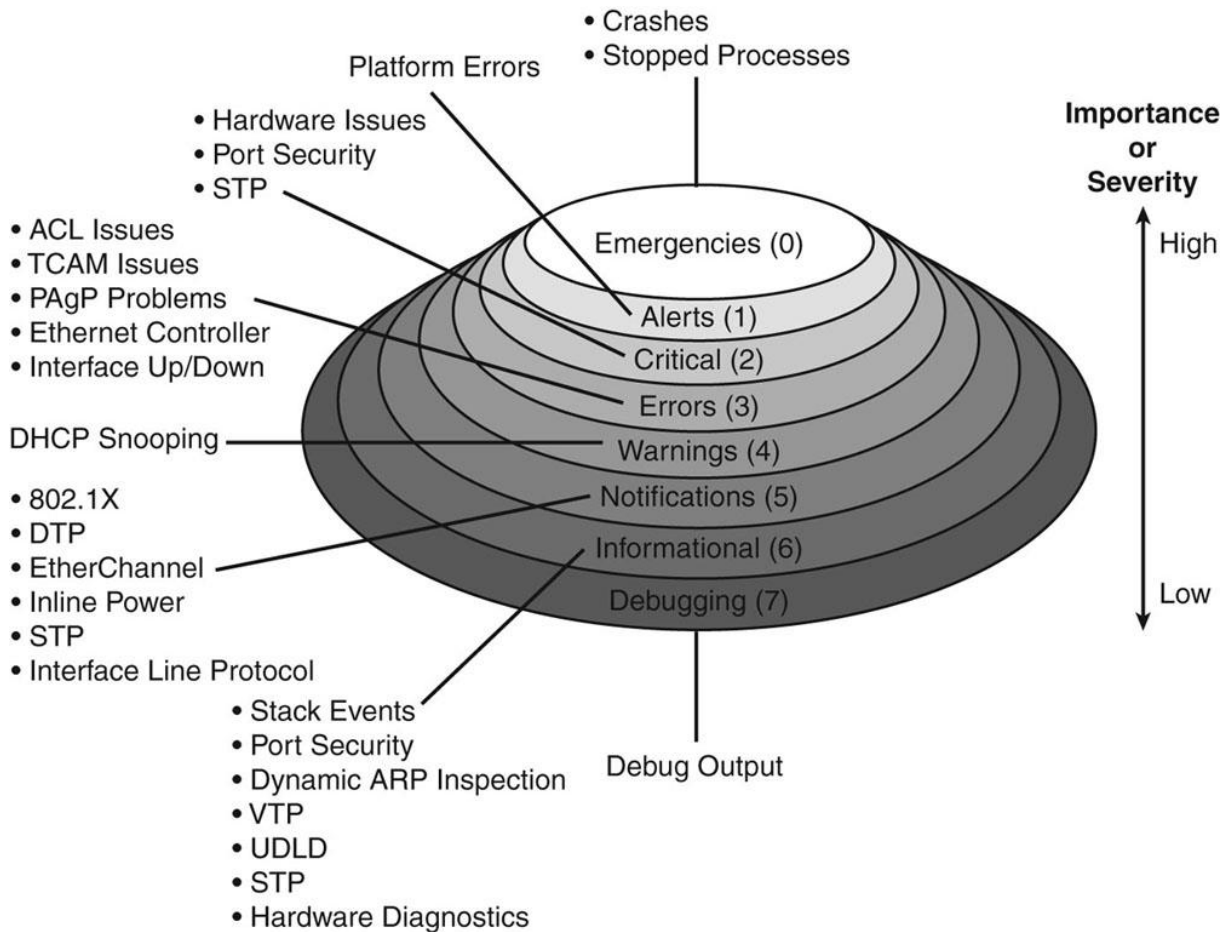
Résultat :

```
Mar 22 09:07:39 Splunk sshd[732]: pam_unix(sshd:session)
Mar 22 09:07:44 Splunk sshd[783]: Accepted password for
Mar 22 09:07:44 Splunk sshd[783]: pam_unix(sshd:session)
Mar 22 09:16:11 ClientLAMP sshd[1352]: Invalid user ro
Mar 22 09:16:11 ClientLAMP sshd[1352]: input_userauth_
Mar 22 09:16:11 ClientLAMP sshd[1352]: pam_unix(sshd:au
Mar 22 09:16:11 ClientLAMP sshd[1352]: pam_unix(sshd:au
Mar 22 09:16:13 ClientLAMP sshd[1352]: Failed password
Mar 22 09:16:15 ClientLAMP sshd[1352]: fatal: Read from
Mar 22 09:16:16 ClientLAMP sshd[1354]: Invalid user ro
Mar 22 09:16:16 ClientLAMP sshd[1354]: input_userauth_
Mar 22 09:16:16 ClientLAMP sshd[1354]: pam_unix(sshd:au
Mar 22 09:16:16 ClientLAMP sshd[1354]: pam_unix(sshd:au
Mar 22 09:16:18 ClientLAMP sshd[1354]: Failed password
```

Super ! Pour améliorer tout ça on peut gérer un fichier par client ! (Voir dans la partie configuration d'un hôte)

<http://www.it-connect.fr/centralisez-vos-logs-avec-rsyslog/>

## 1.4. Schéma des Alertes



Webographie :

[https://wiki.auf.org/wikiteki/Syslog-ng#Configuration\\_de\\_la\\_machine\\_.2BAOk-mettrice](https://wiki.auf.org/wikiteki/Syslog-ng#Configuration_de_la_machine_.2BAOk-mettrice)