

Installation d'Oxidized

Oxidized est un outil de sauvegarde de la configuration des équipements réseau qui a été développé pour remplacer Rancid. Oxidized est codé en Ruby et il prend en charge la collecte de configuration pour plus de 90 types de systèmes d'exploitation réseau.

Oxidized est installé sur un serveur sous Debian 9 et est paramétré pour sauvegarder la configuration de 4 switchs HP Procurve et un routeur PfSense. Les configurations seront stockées sous forme de fichiers sur le serveur Oxidized.

Installation :

```
# apt update
# apt upgrade
# apt install ruby ruby-dev libsqlite3-dev libssl-dev pkg-config cmake
libssh2-1-dev libicu-dev zlib1g-dev g++
```

Installation d'Oxidized.

```
# gem install oxidized
```

Installation des scripts codés en Ruby permettant l'interaction entre les équipements réseaux et Oxidized.

```
# gem install oxidized-script
```

Installation de l'interface web.

```
# gem install oxidized-web
```

Ajout de l'utilisateur oxidized.

```
# useradd oxidized
```

Modification de l'utilisateur pour empêcher qu'il puisse se connecter à la machine.

```
# chsh -s /usr/sbin/nologin oxidized
```

Création des répertoires nécessaire au fonctionnement d'Oxidized.

```
# mkdir -p /opt/oxidized/{output,.config/oxidized/}
```

On définit le répertoire /opt/oxidized comme répertoire home pour l'utilisateur oxidized.

```
# usermod -m -d /opt/oxidized oxidized
# echo "OXIDIZED_HOME=/opt/oxidized" | tee --append /etc/environment
```

On définit l'utilisateur oxidized en tant que propriétaire du dossier /opt/oxidized et tous les fichiers et dossiers qui y sont présents.

```
# chown -R oxidized :oxidized /opt/oxidized
```

Création du fichier `oxidized.service` dans le dossier `/lib/systemd/system/` pour pouvoir manager Oxidized avec `systemd`.

```
# nano /lib/systemd/system/oxidized.service
```

Ajout du contenu suivant dans le fichier :

```
# /lib/systemd/system/oxidized.service
[Unit]
Description=Oxidized - Network Device Configuration Backup Tool
After=network-online.target multi-user.target
Wants=network-online.target

[Service]
ExecStart=/usr/local/bin/oxidized
User=oxidized
KillSignal=SIGKILL

[Install]
WantedBy=multi-user.target
```

On démarre le service Oxidized et on l'active :

```
# systemctl start oxidized.service
# systemctl enable oxidized.service
```

On obtient le fichier `/opt/oxidized/.config/oxidized/config` que l'on a paramétré comme ceci :

```

username: adminSSH
password: azerty2QWERTY
model: procurve, pfsense
interval: 3600
use_syslog: true
log: /opt/oxidized/.config/oxidized/logs/
debug: true
rest: 192.168.200.5:8080
threads: 30
timeout: 20
retries: 3
prompt: !ruby/regexp /^([\w.@-]+[#>]\s?)$/
next_adds_job: false
pid: "/opt/oxidized/pid"

input:
  default: ssh, telnet
  debug: true
  ssh:
    secure: false

output:
  file:
    directory: /opt/oxidized/output/configs

source:
  default: csv
  csv:
    file: /opt/oxidized/.config/oxidized/router.db
    delimiter: !ruby/regexp /:/
    map:
      name: 0
      ip: 1
      model: 2
      group: 3
      vars_map: {}

model_map:
  hp: procurve
  pfsense: pfsense
vars: {}

groups:
  switch:
    username: adminSSH
    password: mdp
  router:
    username: adminSSH
    password: mdp

models: {}

```

On crée le fichier `router.db` pour mettre les informations des équipements.

```
# touch /opt/oxidized/.config/oxidized/router.db
# chown oxidized:oxidized /opt/oxidized/.config/oxidized/router.db
```

On indique les informations des équipements qui correspondent avec le fichier config.

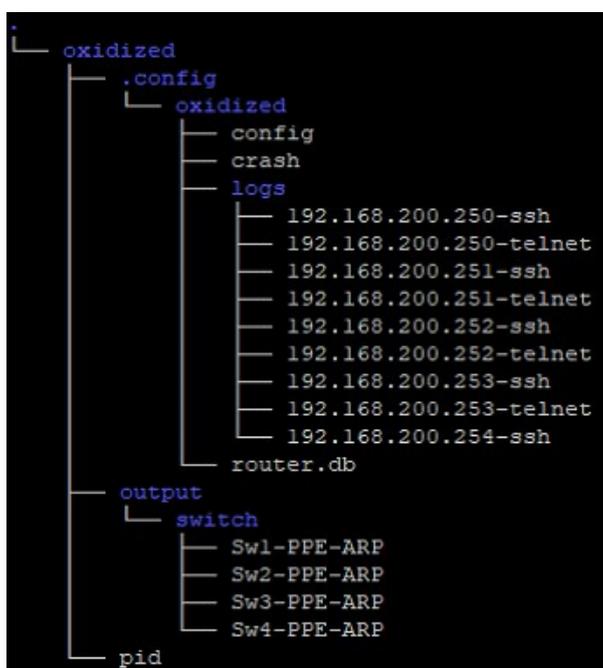
```
#name:ip:model:group
Sw1-PPE-ARP:192.168.200.253:hp:switch
Sw2-PPE-ARP:192.168.200.252:hp:switch
Sw3-PPE-ARP:192.168.200.251:hp:switch
Sw4-PPE-ARP:192.168.200.250:hp:switch

pfSense:192.168.200.254:pfSense:router
```

On redémarre maintenant le service Oxidized :

```
# systemctl restart oxidized.service
```

Voici l'arborescence d'Oxidized.



On voit grâce à l'interface web les équipements indiqués dans le fichier router.db, l'état des équipements et la date de la dernière sauvegarde.