Oxidized установка и настройка

Сайт: https://github.com/ytti/oxidized/

Небольшая но очень полезная программа для централизованного резервного копирования конфигурационных файлов сетевого оборудования. Так же, что очень полезно, умеет отображать изменения в настройках - это позволяет быстро понять что именно изменялось.

В данном примере показана настройка для бэкапа Mikrotik, но вообще Oxidized поддерживает более 130 типов устройств.

Установка

Установка на Ubuntu 18.04

Убедимся что репозиторий universe подключен:

```
add-apt-repository universe
```

Устанавливаем зависимости:

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

Устанавливаем gems (если не нужно ставить какие-то части, отредактируйте строку):

```
gem install oxidized oxidized-script oxidized-web
```

Лучше всего хранить конфигурации в Git, поэтому ставим его:

```
apt install git
```

Создаем пользователя

Процесс желательно запускать под отдельным пользователем, создаем его:

```
useradd -m oxidized
```

Настраиваем git:

```
git config --global user.name "oxidized"
git config --global user.email "oxidized@MYDOMAIN.ru"
git init oxidized.git
```

Из-под пользователя запускаем программу первый раз чтобы создались необходимые каталоги

и файл конфигурации:

```
su - oxidized
oxidized
```

Прерываем работу нажатием Ctrl+C

Настраиваем конфигурационный файл

Пример файла конфигурации, находится в /home/oxidized/.config/oxidized/:

```
---
username: oxidized
password: $ecretP@$$Word
model: routeros
resolve_dns: true
interval: 3600
use_syslog: false
debug: false
threads: 30
timeout: 140
retries: 3
prompt: !ruby/regexp /^([\w.@-]+[#>]\s?)$/
rest: 127.0.0.1:8888
next_adds_job: false
remove_secret: true
vars: {}
groups: {}
models: {}
pid: "/home/oxidized/.config/oxidized/pid"
log: "/home/oxidized/.config/oxidized/log"
crash:
  directory: "/home/oxidized/.config/oxidized/crashes"
  hostnames: false
stats:
  history_size: 10
input:
  default: ssh
  debug: false
  ssh:
    secure: false
  ftp:
    passive: true
  utf8_encoded: true
output:
  default: git
  git:
    user: oxidized
    email: oxidized@MYDOMAIN.ru
```

```
    repo: "/home/oxidized/.config/oxidized/devices.git"
source:
  default: csv
  csv:
    file: "/home/oxidized/.config/oxidized/router.db"
    delimiter: !ruby/regexp /:/
    map:
      name: 0
      model: 1
      ip: 2
      port: 3
      username: 4
      password: 5
    gpg: false
model_map:
  juniper: junos
  cisco: ios
```

Описание наиболее важных переменных:

- username: oxidized - имя пользователя по умолчанию, под которым Oxidized бует подключаться к устройствам
- password: $ecretP@$$Word - пароль по умолчанию
- model: routeros - модель устройства по умолчнию
- interval: 3600 - интервал создания бэкапов в секундах
- timeout: 140 - тайм-аут подключения (по умолчанию 20), если есть задержки в сети - лучше увеличить
- remove_secret: true - удаляет секреты (ключи, пароли и прочее), подробнее для каждой из железок тут: https://github.com/ytti/oxidized/tree/master/lib/oxidized/model
- rest: 127.0.0.1:8888 - адрес и порт веб-сервиса, желательно так и оставить
- в секции input: значение default: ssh лучше исправить убрав telnet, т.к. это небезопасно
- в секции output: сохраняем все в git → default: git
- в секции source: файл с данными о оборудовании default: csv и далее путь до него, разделитель, назначение полей

Подробности директив можно найти тут: https://www.rubydoc.info/gems/oxidized/0.28.0 (проверьте версию программы!)

И тут много полезной информации: https://github.com/ytti/oxidized/blob/master/docs/Configuration.md

Файл с оборудованием

Создаем и редактируем файл: /home/oxidized/.config/oxidized/router.db

```
mikrotik01:routeros:10.10.10.1
mikrotik02:routeros:10.10.10.10:22:test:$ecretP@$$word
```

первая строка - наименование, тип оборудования, IP-адрес вторая строка - наименование, тип

оборудования, IP-адрес, порт, логин, пароль

Для первого устройства будут использоваться логин и пароль по умолчанию, прописанные в файле config

Настройка службы

Облегчим жизнь копированием файла управлением службы и включением автозапуска:

```
sudo cp /var/lib/gems/2.5.0/gems/oxidized-0.28.0/extra/oxidized.service
/lib/systemd/system/
sudo cp /var/lib/gems/2.5.0/gems/oxidized-0.28.0/extra/oxidized.init.d
/etc/init.d/oxidized
sudo systemctl enable oxidized.service
sudo systemctl start oxidized
```

Готово, можно запускать службу, проверить ее работу

Авторизация в браузере

По умолчанию Oxidized не имеет никакой авторизации, что позволяет подключаться к нему всем желающим.

Исправить это можно с помощью Reverse-proxy

Nginx

Устанавливаем:

```
apt install nginx
```

Копируем файл настроек:

```
sudo cp /var/lib/gems/2.5.0/gems/oxidized-0.28.0/extra/oxidized.nginx
/etc/nginx/sites-available/oxidized
```

Его содержимое:

```
server {
        listen 80;
        listen [::]:80;

        server_name oxidized.example.com;

        location / {
                proxy_pass http://127.0.0.1:8888/;
```

```
        }

        access_log /var/log/nginx/access_oxidized.log;
        error_log /var/log/nginx/error_oxidized.log;
}
```

и добавляем строки:

```
auth_basic “Username and Password Required”;
auth_basic_user_file /etc/nginx/.htpasswd;
location / {
   proxy_set_header Host $host;
   proxy_set_header X-Real-IP $remote_addr;
   proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
   proxy_pass http://127.0.0.1:8888/;}
```

Создаем файл с паролем:

```
sudo htpasswd /etc/nginx/.htpasswd username
```

Apache

Включаем модули:

```
a2enmod proxy
a2enmod proxy_http
```

Добавляем в ports.conf, где 10.10.10.1 - IP веб-сервера

```
# Oxidized port
Listen 10.10.10.1:8888
```

Создаем файл конфигурации /etc/apache2/sites-available/oxidized.conf:

```
<VirtualHost *:8888>

    ServerAdmin admin@MYDOMAIN.ru
    ServerName oxidized.MYDOMAIN.ru
    ServerAlias oxidized

    <Location />
        AuthType Basic
        AuthName "Username and Password Required"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Location>

    ProxyPass /  http://127.0.0.1:8888/
    ProxyPassReverse / http://127.0.0.1:8888/
```

```
    ErrorLog /var/log/apache2/oxidized_error.log
    CustomLog /var/log/apache2/oxidized_access.log combined

</VirtualHost>
```

Включаем сайт:

```
a2ensite oxidized.conf
```

Создаем файл с паролем:

```
sudo htpasswd /etc/apache2/.htpasswd username
```

Перезапускаем apache2

Настройка Mikrotik

Осталось настроить наши устройства Mikrotik:

- Создать пользователя под которым будет подключаться Oxidized: System → Users, права дать только «Read»
- Проверить работает ли сервис SSH (IP → Services и включить ssh)
- Открыть доступ к порту 22 в IP → Firewall → Filter Rules

Крайне желательно ограничить доступ по порту 22 (если вы его не используете для работы) и IP-адресу сервера с установленным Oxidized

oxidized, mikrotik, backup, configuration, бэкап конфигурации