📖 Установка Oxidized

Инструкция по установке и настройке `Oxidized` на **Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-generic x86_64)**.

✍️ Материал из канала 👉 @netscripor

---

1. Подготовка системы (DNS и время)

У вас может быть либо 50-cloud-init.yaml либо 00-network-config.yaml или какой то другой...

```
sudo nano /etc/netplan/*.yaml
```

Добавляем DNS-серверы:

```
network:
  version: 2
  ethernets:
    ens18:
      dhcp4: true
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
```

Применяем:

```
sudo netplan generate
sudo netplan apply
```

Настройка времени и синхронизации:

```
sudo timedatectl set-timezone Europe/Moscow
sudo timedatectl set-ntp true
sudo systemctl restart systemd-timesyncd
```

---

2. Установка зависимостей

```
sudo apt update
sudo apt install -y ruby ruby-dev libsqlite3-dev libssl-dev pkg-config cmake
```

```
libssh2-1-dev libicu-dev zlib1g-dev g++ libyaml-dev nginx apache2-utils jq
git
```

3. Установка Oxidized и веб-интерфейса

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

Проверка

```
oxidized --version
```

4. Системный пользователь и структура каталогов

```
sudo useradd -r -s /usr/sbin/nologin -d /opt/oxidized oxidized
sudo mkdir -p /opt/oxidized/config /opt/oxidized/logs /opt/oxidized/crashes
sudo chown -R oxidized: /opt/oxidized
```

5. Git-репозиторий для хранения конфигураций

```
sudo -u oxidized git config --global user.name "oxidized"
sudo -u oxidized git config --global user.email "admin@example.com"
sudo -u oxidized git config --global user.timezone "Europe/Moscow"

sudo -u oxidized bash -c "cd /opt/oxidized/config && git init --bare
oxidized.git"
```

6. Начальная настройка Oxidized

Создаём конфиг:

```
oxidized
```

Он появится в `~/.config/oxidized/config`

Копируем его в боевую директорию:

```
sudo cp /home/netscripor/.config/oxidized/config /opt/oxidized/config/config
```

Редактируем:

```
sudo nano /opt/oxidized/config/config
```

Пример содержимого `config` :

```
resolve_dns: false
interval: 43200 #каждые 12 часов собирает конфиги
use_syslog: true
threads: 60
use_max_threads: true
timeout: 180
retries: 2
prompt: !ruby/regexp /^([\w.@-]+[#>]\s?)$/
rest: 127.0.0.1:8888
next_adds_job: false

vars:
  enable: YOUR_ENABLE_PASSWORD

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

map:
  verify_host_key: never
  utf8_encoded: true

output:
  default: git
  git:
    user: oxidized
    email: admin@example.com
    repo: "/opt/oxidized/config/oxidized.git"

source:
  default: csv
  csv:
    file: "/opt/oxidized/config/node.db"
    delimiter: !ruby/regexp /:/
    map:
      name: 0
      model: 1
      ip: 2
      port: 3
      username: 4
      password: 5
      group: 6

model_map:
  Cisco: ios
```

```
  mikrotik: routeros
  Juniper: junos
  Huawei: vrp
  Nexus: nxos

hooks:
  telegram_failure_alert:
    type: exec
    events: [node_fail]
    cmd: 'curl -X POST https://api.telegram.org/bot<TOKEN>/sendMessage -d
chat_id=<CHATID> -d text="#Oxidized failed to retrieve the config for
${OX_NODE_NAME} at ${OX_NODE_IP}"'
    timeout: 120
    async: true
    interval: 3600
```

7. Список устройств (`node.db`)

```
sudo nano /opt/oxidized/config/node.db
```

Пример содержимого:

```
r1:ios:192.168.1.1:22:admin:pass:cisco
r2:junos:192.168.1.2:22:admin:pass:juniper
r3:routeros:192.168.1.3:22:admin:pass:mikrotik
```

8. Сервис systemd

```
sudo nano /etc/systemd/system/oxidized.service
```

```
[Unit]
Description=Oxidized - Network Device Config Backup
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/usr/local/bin/oxidized
User=oxidized
KillSignal=SIGKILL
Environment="OXIDIZED_HOME=/opt/oxidized/config"
Restart=on-failure

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

Запуск:

```
sudo systemctl daemon-reload
sudo systemctl enable --now oxidized
journalctl -u oxidized -f
```

---

9. Web-интерфейс через Nginx (с HTTPS и Basic Auth)

9.1. Пароль доступа

```
sudo htpasswd -cb /etc/nginx/.htpasswd admin StrongPass123
```

9.2. Генерация самоподписанного SSL-сертификата:

```
sudo mkdir -p /etc/nginx/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
 -keyout /etc/nginx/ssl/oxidized.key \
 -out /etc/nginx/ssl/oxidized.crt \
 -subj "/C=RU/ST=Moscow/L=Moscow/O=Oxidized/CN=oxidized.local"
```

9.3. Конфиг Nginx (замените IP SERVER на ваш ip):

```
sudo nano /etc/nginx/sites-available/oxidized
```

```
server {
    listen 443 ssl;
    server_name IP SERVER;

    ssl_certificate /etc/nginx/ssl/oxidized.crt;
    ssl_certificate_key /etc/nginx/ssl/oxidized.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384';

    auth_basic "Protected Oxidized UI";
    auth_basic_user_file /etc/nginx/.htpasswd;

    location / {
        proxy_pass http://127.0.0.1:8888;
        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_set_header X-Forwarded-Proto $scheme;

        limit_except GET {
            deny all;
```

```
        }
    }
}

server {
    listen 80;
    server_name IP SERVER;
    return 301 https://$host$request_uri;
}
```

Активация:

```
sudo ln -s /etc/nginx/sites-available/oxidized /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl restart nginx
```

10. Проверка работоспособности

```
sudo systemctl status oxidized
curl http://127.0.0.1:8888/nodes | jq .
git --git-dir=/opt/oxidized/config/oxidized.git log --oneline
```

11. Бэкап конфигурации и нод с Telegram-уведомлением

Подготовка структуры:

```
sudo mkdir -p /opt/oxidized/{backups,scripts}
sudo chown -R oxidized: /opt/oxidized
```

Скрипт:

```
sudo nano /opt/oxidized/scripts/git-backup.sh
```

```
#!/bin/bash
cd /opt/oxidized || exit 1
# === КОНФИГУРАЦИЯ ===
BACKUP_DIR="/opt/oxidized/backups"
CONFIG_DIR="/opt/oxidized/config"
RETENTION_DAYS=7
TELEGRAM_BOT_TOKEN="XXXXX"  # замените!
TELEGRAM_CHAT_ID="-XXXXX"   # замените!

# === Бэкап ===
DATE=$(date +%F_%H-%M)
```

```
FILENAME="oxidized_${DATE}.tar.gz"
OUT="$BACKUP_DIR/$FILENAME"

mkdir -p "$BACKUP_DIR"

# Бэкапим ВСЮ директорию конфигурации целиком
tar czf "$OUT" -C "$CONFIG_DIR" . 2>/dev/null

# === Удаление старых архивов ===
DELETED=$(find "$BACKUP_DIR" -type f -name "*.tar.gz" -mtime
+$RETENTION_DAYS -print -delete)

# === Telegram сообщение ===
MESSAGE="📦 Oxidized backup создан: $FILENAME"
if [[ -n "$DELETED" ]]; then
  MESSAGE+="\n🗑️ Удалено старых бэкапов:\n$(basename -a $DELETED)"
fi

curl -s -X POST
"https://api.telegram.org/bot${TELEGRAM_BOT_TOKEN}/sendMessage" \
  -d chat_id="$TELEGRAM_CHAT_ID" \
  -d text="$MESSAGE"
```

Права:

```
sudo chmod +x /opt/oxidized/scripts/git-backup.sh
sudo chown oxidized: /opt/oxidized/scripts/git-backup.sh
```

systemd unit + таймер:

```
sudo nano /etc/systemd/system/oxidized-git-backup.service
```

```
[Unit]
Description=Oxidized Git Backup with Telegram Notification

[Service]
Type=oneshot
User=oxidized
ExecStart=/opt/oxidized/scripts/git-backup.sh
```

Таймер:

```
sudo nano /etc/systemd/system/oxidized-git-backup.timer
```

```
[Unit]
Description=Daily backup of Oxidized Git repo

[Timer]
OnCalendar=daily
Persistent=true

[Install]
WantedBy=timers.target
```

Активация:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable --now oxidized-git-backup.timer
```

Проверка:

```
sudo systemctl start oxidized-git-backup.service
systemctl list-timers | grep oxidized
journalctl -u oxidized-git-backup.service
```

🔄 Восстановление:

```
sudo systemctl stop oxidized
sudo rm -rf /opt/oxidized/config/*
sudo tar xzf /opt/oxidized/backups/oxidized_<дата>.tar.gz -C
/opt/oxidized/config/
sudo chown -R oxidized: /opt/oxidized/config
sudo systemctl start oxidized
```

🔗 Материал подготовлен для @netscripor