MBUM #2

Zarządzanie kopiami konfiguracji RouterOS

Jacek Rokicki

- w IT od 1998,
- entuzjasta systemów operacyjnych z rodziny Unix,
- projektowanie, budowa i utrzymanie wysoko dostępnych rozwiązań z wykorzystaniem FLOSS,
- z vendorem MikroTik od 2011,
- specjalizuję się w przełącznikach CRS, systemach VPN, IPv6 i protokołach dynamicznego routingu.

MikroTik MTCNA MikroTik MTCRE MikroTik MTCIPv6E MikroTik MTCINE

j.rokicki@c3net.pl

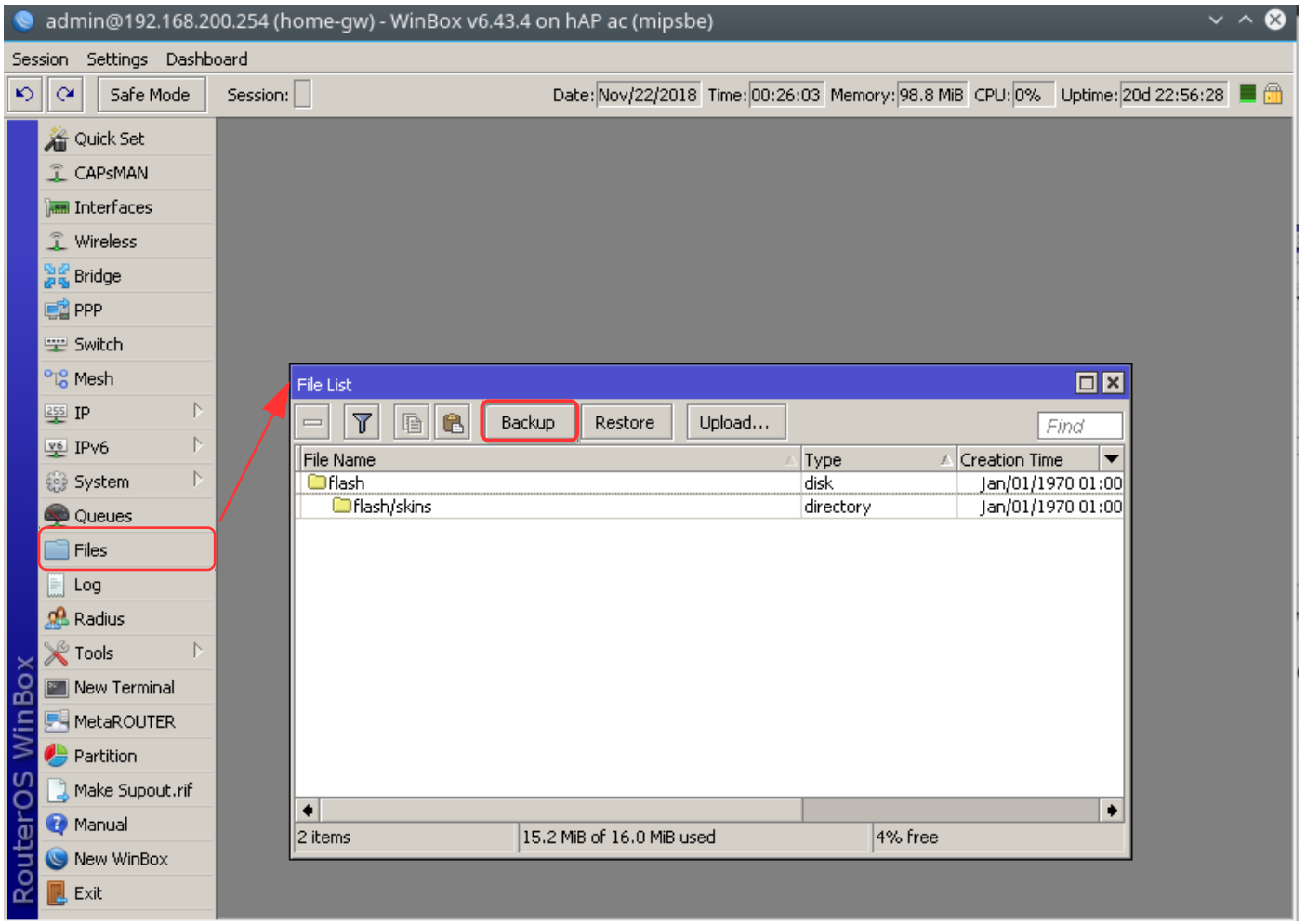
Agenda

- backup, czy i jak to robicie?
- krótko o Git i SSH,
- konfiguracja RouterOS-a,
- bezpieczeństwo raz jeszcze,
- jak utworzyć, pobrać i umieścić w repozytorium naszą konfiguracje,
- mamy konfiguracje w repozytorium, co można z tym zrobić?
- automatyzacja tworzenia kopii,
- co w razie problemów?
- pytania,
- zakończenie.

- **wbudowane w RouterOS narzędzia do wykonywania kopii konfiguracji**

/system backup save

/export file=



- zaprojektowany i napisany w 2005 przez twórcę Linuksa, rozwijany przez społeczność FLOSS.
- w odróżnieniu od innych systemów kontroli wersji, przechowuję dane nie jako listę zmian na plikach, a jako migawki stanów. To stanowi o jego wydajności.
- jako rozproszony system kontroli wersji, nie potrzebuję do pracy serwera odpowiadającego za utrzymanie repozytorium kodu.
- ma wbudowane mechanizmy spójności danych.

- **zdalne wykonywanie poleceń za pomocą SSH**

ssh [user@]hostname ‘’polecenie”

- **zarządzanie kluczami**

ssh-keygen

~/.ssh/id_rsa
~/.ssh/id_rsa.pub

- **dodatkowe opcje upraszczające życie**

-oStrictHostKeyChecking=no
-oConnectTimeout=3
-oPasswordAuthentication=no

- **konfiguracja w RouterOS**

- usługa SSH jest domyślnie włączona

- warto zmienić domyślny port i ograniczyć połączenia do konkretnego adresu lub sieci
/ip service set ssh port=10022 address=1.2.3.4/32

- prawdopodobnie trzeba dodać regułę FW na łańcuchu INPUT
*/ip firewall add action=accept chain=input comment=backup *
src-address=1.2.3.4/32 dst-port=10022 protocol=tcp

- dodajemy grupę z ograniczonymi uprawnieniami oraz użytkownika w tej grupie, importujemy klucz publiczny

```
/user group
add name=backup policy="ssh,read,write,policy,test,sensitive,!ftp,!local,!telnet,\
    !reboot,!winbox,!password,!web,!sniff,!api,!romon,!dude,!tikapp"
/user
add address=1.2.3.4/32 group=backup name=backup
ssh-keys import public-key-file=id-rsa.pub user=backup
```

- **zdalnie tworzymy zrzut konfiguracji i dodajemy go do repo**

- konfigurujemy naszą instancję git-a dodając informacje o sobie
  $ git config --global user.name "Jan Nowak"
  $ git config --global user.email jannowak@example.com

-inicjalizujemy repozytorium oraz tworzymy branch dla naszego MT
  $ git init
  $ git branch mikrotik1

- zdalnie tworzymy kopię konfiguracji w formie binarnej oraz tekstowej
  *$ ssh -p 10022 backup@mikrotik1 ‘’system backup save dont-encrypt=yes *
   name=config ; export file=config”

- pobieramy oba pliki konfiguracyjne
  $ scp -P 10022 -r backup@mikrotik1:config. .*

- dodajemy i zatwierdzamy w repo nasze nowe pliki konfiguracyjne
  $ git add .
  $ git commit -m ‘’konfiguracja mikrotik1”

- sprawdzamy co się stało
  $ git log
  commit c6ad3de1b45ab2012463a57c93c49e2d8378814c
  Author: Jan Nowak <jannowak@example.com>
  Date: Tue Nov 20 03:18:29 2018 +0100

      konfiguracja mikrotik1

- **podstawowe operacje na konfiguracji w repo**

- uzyskanie konfiguracji z konkretnego momentu
  $ git log
  $ git checkout hash
  lub
  $ git archive hash > backup.tar

- porównanie zmian między bieżącym, a poprzednim zatwierdzeniem
  $ git log -p -1

- porównanie konfiguracji między zatwierdzeniami
  $ git log
  $ git diff hash1 hash2

- powrót do poprzedniego miejsca
  $ git checkout -

- powrót do ostatniego zatwierdzenia w danym branchu
  $ git branch
  $ git checkout branch

- Defragmentacja
  $ git gc

- **automatyzacja tworzenia kopii**

- prosty skrypt np. w Bash-u

- wykonywanie zrzutu konfiguracji tylko gdy pojawiły się zmiany (export na stdout i porównanie z ostatnim zatwierdzeniem)

- użycie systemowego harmonogramu (cron) do cyklicznego wykonywania skryptu

$ crontab -e

*15 3 * * * $HOME/bin/skrypt.sh*

- **co w razie problemów?**

- przekierowywanie wyjścia kolejnych etapów do pliku-logu

- po wykonaniu skryptu, wysyłanie maila z logiem w załączniku
  $ mailx -A konto -s temat -a załącznik jannowak@example.com

- informowanie o błędach na podstawie słów kluczowych z loga

?

Dziękuję za uwagę