Instalace poštovního serveru Mailcow na Raspberry PI

tutorial viz: instalace Mailcow

Instalace poštovního serveru Mailcow na Raspberry PI

P?edpoklady pro indstalaci Mailcow na Raspberry PI s dockerem containerem

- Raspberry PI Server se systémem Raspbian 64bit

1. P?idejte Docker repository:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=arm64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

2. Nainstalujte Docker a Docker Compose:

```
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin
```

Ověřte verzi Dockeru:

```
docker --version
```

3. Stáhn?te Mailcow

Přejděte do složky `/opt` a stáhněte Mailcow:

```
sudo mkdir -p /opt/mailcow
cd /opt/mailcow
git clone https://github.com/mailcow/mailcow-dockerized.git mailcow
cd mailcow
```

4. Vygenerujte konfiguraci

```
./generate_config.sh
```

5. Tento skript vytvo?í soubor `mailcow.conf`, který m?žete upravit podle svých pot?eb:

```
nano mailcow.conf
```

Vyhledej řádek:

SKIP_UNBOUND_HEALTHCHECK=n

SKIP_LETS_ENCRYPT=n

A změň ho na:

SKIP_UNBOUND_HEALTHCHECK=y

SKIP_LETS_ENCRYPT=y

Restartuj kontejnery Mailcow

Aby se změny projevily, restartuj kontejnery:

```
docker-compose down
docker-compose up -d
```

6. Spušt?ní Mailcow

Stáhněte potřebné kontejnery a spusťte Mailcow:

```
docker-compose pull
docker-compose up -d
```

7. Ov??te, zda kontejnery b?ží:

```
docker ps
```

8. P?ístup k administraci

Otevřete svůj prohlížeč a přejděte na adresu, kterou jste nastavili v `mailcow.conf` (například `https://yourdomain.com`).

admin / moohoo

Zabezpe?ení Mailcow file2ban, p?ihlášení jen z CZ

Zabezpe?ení Mailcow file2ban, p?ihlášení jen z CZ

1. Nainstalovat geoip databázi:

```
apt install fail2ban
apt install geoip-database
apt install fail2ban geoip-database geoip-bin
```

Vytvoříme složku pro filtry, pokud neexistuje:

```
mkdir -p /etc/fail2ban/filter.d
```

2. Vytvořit nový filter pro fail2ban:

```
nano /etc/fail2ban/filter.d/non-cz.conf
```

S obsahem:

```
[Definition]
failregex = <HOST> .* [client .* ".*"$ !country=CZ
ignoreregex =
```

3. Vytvořit jail konfiguraci:

```
mkdir -p /etc/fail2ban/jail.d
nano /etc/fail2ban/jail.d/non-cz.conf
```

S obsahem:

```
[non-cz]
enabled = true
filter = non-cz
logpath = /var/log/auth.log
bantime = 86400
findtime = 600
maxretry = 1
```

4. Restartovat fail2ban:

```
systemctl restart fail2ban
```

Pravda, pojďme to udělat jednodušeji. Vytvořím vám jednoduchý skript, který uložíte do /root/check-ssh.sh a budete moct kdykoli spustit, abyste viděl, kdo se snažil připojit:

bash

```
cat > /root/check-ssh.sh << 'EOF'
#!/bin/bash
echo "=== Pokusy o SSH přihlášení ==="
journalctl | grep sshd | grep -i "failed\|invalid\|error" | tail -n 20
echo ""
echo "=== Blokované IP adresy ==="
iptables -L INPUT -n -v | grep DROP
EOF
```

Pak ho uděláte spustitelným:

bash

```
chmod +x /root/check-ssh.sh
```

A kdykoli budete chtít zkontrolovat pokusy o přihlášení, stačí napsat:

bash

```
/root/check-ssh.sh
```