

# socks proxy - socks5-server

Ein Docker Container mit einem socks proxy der auch Authentifizierung unterstützt

- [Installation](#)
- [Proxychains - Für Programme die keinen proxy unterstützen](#)

# Installation

Beschreibung:

Ein socks5 Proxy mit Authentifizierung

Erstellung der docker compose Datei und der .env Datei.

Vorraussetzung:

Docker installiert

```
apt install docker.io docker-compose curl net-tools
```

Installation

ein Verzeichnis erstellen im root Verzeichnis

```
mkdir /root/proxy
```

Dort eine Datei namens .env erstellen und folgenden Inhalt einfügen und anpassen wenn nötig

```
nano /root/proxy/.env
```

Inhalt

Wenn die Variablen proxy user und Passwort leer beleiben, funktioniert der Proxy ohne Authentifizierung

```
PROXY_USER=someuser  
PROXY_PASSWORD=somepass  
PROXY_PORT=1080
```

Nun die composer Datei

```
nano /root/proxy/docker-compose.yml
```

Inhalt

```
version: '3'
```

```
services:
```

```
  socks5-proxy:
```

```
    image: serjs/go-socks5-proxy
```

```
    container_name: socks5-proxy
```

```
    ports:
```

```
      - "${PROXY_PORT}:1080"
```

```
    environment:
```

```
      - PROXY_PORT=${PROXY_PORT}
```

```
    restart: unless-stopped
```

Nun mit

```
docker-compose up -d den proxy starten
```

nun kann der proxy mit dem vergeben Port verwendet werden

# Proxychains - Für Programme die keinen proxy unterstützen

Beschreibung:

Ein Programm das als Tunnel gilt, wenn das Programm selbst keinen proxy unterstützt. proxychain bekommt als parameter die binary von dem eigentlichen programm.

Installation

```
apt install proxychains
```

Nun die konfigurations anpassen

```
nano /etc/proxychains.conf
```

Dort den proxy hinzufügen

```
#die abarbeitung:
#proxychains wie der name schon sgat kann eine kette von proxys benutzen.
#hier die modie

#dynamic_chain
Dynamisch - Jede Verbindung wird über verkettete Proxys hergestellt.
Alle Proxys werden in der Reihenfolge, wie sie in der Liste erscheinen, verkettet.
Mindestens ein Proxy muss online sein, damit die Kette funktioniert.
(Tote Proxys werden übersprungen)
Andernfalls wird EINTR an die App zurückgegeben.

strict_chain
Strikt - Jede Verbindung wird über verkettete Proxys hergestellt.
Alle Proxys werden in der Reihenfolge, wie sie in der Liste erscheinen, verkettet.
```

Alle Proxys müssen online sein, damit die Kette funktioniert.  
Andernfalls wird EINTR an die App zurückgegeben.

```
#random_chain
```

Zufällig - Jede Verbindung wird über einen zufälligen Proxy  
(oder eine Proxy-Kette, siehe chain\_len) aus der Liste hergestellt.  
Diese Option ist gut, um dein IDS zu testen :)

```
#meisten nutzen wir die strict chain, bei 1 proxy in der liste es es sogar egal
```

```
#dns anfragebn auch über den proxy
```

```
proxy_dns
```

```
[ProxyList]
```

```
#beispiel proxy
```

```
#typ ip vom proxy port benutzername kennwort
```

```
# möcgliche typen socks4,socks5,http
```

```
socks5 192.168.1.1 1080
```

```
http 192.168.1.2 8080
```

```
socks5 192.168.1.3 1080 joe secretpassword
```

```
http 192.168.1.4 1080 joe secretpassword
```

## Aufruf

```
/usr/bin/proxychains <programmanme>
```

```
/usr/bin/proxychains firefox
```