

استفاده از نسخه

جدید MTPROXY

تویتر نگارنده: Skynet\_Acc

## تفاوت با نسخه قبلی:

در نسخه جدید MTPROXY که در مرحله بتا از توسعه خود قرار دارد، تا ۱۶ بایت Padding تصادفی به انتهای پکت‌های آن افزوده میشود. در نسخه ۱ این پروتکل به دلیل ثابت بودن طول پکت‌ها شاهد شناسایی سرورها توسط تکنیک DPI بودیم، اما به لطف افزوده شدن Padding‌های تصادفی، طول پکت‌ها همواره متغیر خواهد بود و دیگر نمیتوان سرورهای MTPROXY را بوسیله اندازه طول پکت‌ها شناسایی و بلاک کرد.

برای اینکه بفهمیم یک سرور از نسخه جدید MTPROXY - که به Secure Intermediate یا Version D معروف است - استفاده میکند یا خیر، باید به Secret آن پراکسی نگاه کنیم. در جدول زیر تفاوت این دو نسخه آورده شده است:

نسخه پراکسی	1	D
طول Secret	۳۲ کاراکتر	۳۴ کاراکتر
شاخص Secret	---	حتماً با dd شروع میشود
ویژگی	سایز پکت‌ها ثابت است	سایز پکت‌ها متفاوت است

در زیر نحوه نصب و استفاده از نسخه جدید MTPROXY به شما آموزش داده خواهد شد. این اسکریپت بر روی توزیع‌های گنو/لینوکس زیر تست و با موفقیت نصب و اجرا شده‌اند:

1. Debian 9 (64-bit)
2. Ubuntu 16 (64-bit)
3. CentOS 7 (64-bit)

## پیشنیازها:

برای استفاده و راه‌اندازی این اسکریپت، نیازمند بسته‌های نرم‌افزاری زیر می‌باشیم، که در ادامه با نحوه نصب هر یک از این ابزارها آشنا میشویم:

1. git
2. یک ادیتور txt که من از nano استفاده میکنم.
3. Docker CE
4. Docker Compose

قبل از نصب هر یک از موارد فوق با اجرای دستورات زیر مطمئن میشویم که لیست پکیج‌های ما بروز هستند، سپس پکیج‌های نصب شده را به آخرین نسخه آپدیت میکنیم:

<b>Debian/Ubuntu</b>	sudo apt-get update sudo apt-get upgrade
<b>CentOS</b>	sudo yum upgrade

## نصب git:

برای کلون و دانلود کردن پروژه نیازمند نصب git بر روی سرور خود هستیم. این بسته معمولاً بطور پیشفرض بر روی سیستم‌عامل شما نصب است، اما اگر نصب نبود میتوانید با استفاده از دستور زیر آنرا نصب کنید. توجه داشته باشید که اجرای این دستور حتی در صورتی که git بر روی سیستم شما نصب باشد مشکلی ایجاد نمیکند و تنها آنرا به جدیدترین نسخه موجود آپدیت میکند (در صورت موجود بودن) و اگر آپدیتی در کار نبود تنها به شما پیامی مبنی بر از قبل نصب بودن git میدهد:

<b>Debian/Ubuntu</b>	sudo apt-get install git
<b>CentOS</b>	sudo yum install git

## نصب Docker در دبین و اوبونتو:

ابتدا با نصب بسته‌های زیر، این اجازه را به apt میدهیم تا بتواند از ریپازیتوری‌های HTTPS استفاده کند.

```
sudo apt-get install \  
apt-transport-https \  
ca-certificates \  
curl \  
gnupg2 \  
software-properties-common
```

سپس کلید GPG رسمی داکر را به apt اضافه میکنیم:

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

اکنون ریپازیتوری داکر را به apt اضافه میکنیم تا بتوانیم آنرا نصب و دانلود کنیم:

<b>Debian</b>	<pre>sudo add-apt-repository \ "deb [arch=amd64] https://download.docker.com/linux/debian \ \$(lsb_release -cs) \ stable"</pre>
<b>Ubuntu</b>	<pre>sudo add-apt-repository \ "deb [arch=amd64] https://download.docker.com/linux/ubuntu \ \$(lsb_release -cs) \ stable"</pre>

اکنون یکبار پکیج ایندکس apt را بروزرسانی میکنیم تا بتوانیم از ریپازیتوری اضافه شده استفاده کنیم:

```
sudo apt-get update
```

حال کفایست داکر را نصب کنیم:

```
sudo apt-get install docker-ce
```

پس از اجرای دستور فوق، برای اطمینان از نصب کامل میتوانید دستور زیر را اجرا کنید. اگر همه چیز درست نصب شده باشد، دستور فوق پس از دانلود image تست و اجرای آن، متون آزمایشی را چاپ میکند و از برنامه خارج میشود. این به آن معناست که داکر بدرستی نصب و قابل استفاده میباشد.

```
sudo docker run hello-world
```

## نصب Docker در CentOS:

ابتدا پکیجها و درایورهای مورد نیاز جهت افزودن ریپازیتوری به yum را با استفاده از دستور زیر نصب میکنیم:

```
sudo yum install -y yum-utils \  
device-mapper-persistent-data \  
lvm2
```

سپس ریپازیتوری داکر را به yum می‌افزاییم:

```
sudo yum-config-manager \  
--add-repo \  
https://download.docker.com/linux/centos/docker-ce.repo
```

اکنون میتوانیم با استفاده از دستور زیر داکر را نصب کنیم:

```
sudo yum install docker-ce
```

پس از نصب سرویس داکر را اجرا میکنیم:

```
sudo systemctl start docker
```

حال میتوانیم جهت اطمینان از نصب و کارایی داکر، دستور زیر را اجرا کنیم. در صورتی که همه چیز درست نصب و اجرا شده باشد، با اجرای دستور زیر یک image آزمایشی دانلود میشود، اطلاعاتی به شما نمایش داده میشود و سپس از آن خارج میشود:

```
sudo docker run hello-world
```

## نصب Docker Compose:

ابتدا docker-compose را دانلود میکنیم:

```
sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
```

سپس دسترسی اجرایی را به آن میدهیم:

```
sudo chmod +x /usr/local/bin/docker-compose
```

حال برای بررسی درستی کارکرد آن، دستور زیر را اجرا میکنیم. با اجرای این دستور باید نسخه docker-compose در کنسول چاپ شود:

```
docker-compose --version
```

## نصب nano:

برای ویرایش و کانفیگ MTPROXY نیازمند یک ادیتور متن هستیم. من از ادیتور nano استفاده میکنم که ساده، سبک و کاربردی است. برای نصب از دستور زیر استفاده میکنیم:

<b>Debian/Ubuntu</b>	sudo apt-get install nano
<b>CentOS</b>	sudo yum install nano

## نصب اسکریپت MTPROXY:

ابتدا با استفاده از git اسکریپت مورد نظر را از برنج master دانلود میکنیم:

```
git clone -b master https://github.com/alexbers/mtprotoproxy.git
```

سپس وارد دایرکتوری آن میشویم:

```
cd mtprotoproxy
```

اکنون با استفاده از دستور زیر فایل config.py را که مختص تنظیمات پراکسی است باز میکنیم و تغییرات دلخواه را در آن ایجاد میکنیم:

```
nano config.py
```

با اجرای دستور فوق محتویات زیر را مشاهده میکنید:

```
PORT = 3256

#name -> secret (32 hex chars(
USERS = {
    "tg": "00000000000000000000000000000000",
    "tg2": "0123456789abcdef0123456789abcdef"
}

#Tag for advertising, obtainable from @MTProxybot
#AD_TAG = "3c09c680b76ee91a4c25ad51f742267d"
```

عددی که مقابل PORT قرار میگیرد پورتهی است که پراکسی شما بر روی آن سرویسدهی خواهد کرد. USERS کاربران شما را مشخص میکند، شما میتوانید چندین سکرت مختلف برای کاربران مختلف برای پراکسی خود در نظر بگیرید، فرمت تعریف کاربر بصورت "SECRET" : "USERNAME" است. عبارتی که در SECRET قرار میدهید باید یک عبارت هگز ۳۲ کاراکتری باشد. با کلیک بر روی [این لینک](#) میتوانید یک عبارت تصادفی HEX تولید کنید و از آن در فایل کانفیگ استفاده کنید. اگر میخواهید کانال شما به عنوان اسپانسر در تلگرام کاربران نمایش داده شود، میتوانید کاراکتر # را از جلوی AD\_TAG حذف کنید و تگ کانال خود را که از ربات MTPROXY دریافت کردهاید، در محل مربوطه وارد کنید.

بعد از انجام تغییرات دکمه CTRL+X را فشار دهید، با فشردن دکمه Y ذخیره شدن تغییرات را تایید کنید و بدون تغییر در نام فایل دکمه ENTER را فشار دهید تا تغییرات شما بر روی فایل ذخیره شود.

به عنوان مثال من میخواهم پراکسی خود را بر روی پورت ۸۰۸۰ راه اندازی کنم. تنها یک سکرت میخواهم داشته باشم و آن بطور مثال aebd04e4d18dcb19fc201ac690d8bdef است. توجه داشته باشید که نام کاربری مهم نیست و هر چیزی میتواند باشد. در ضمن تعریف تنها یک سکرت به این معنی نیست که فقط یک نفر میتواند به سرور شما متصل شود، بلکه هزاران نفر میتوانند از آن سکرت برای برقراری ارتباط با پراکسی شما استفاده کنند. از آنجایی که نمیخواهم کانال خود را به عنوان اسپانسر معرفی کنم، به AD\_TAG کاری ندارم. بنابراین فایل کانفیگ من به شکل زیر خواهد بود:

```
PORT = 8080

#name -> secret (32 hex chars)
USERS = {
    "skynet": "aebd04e4d18dcb19fc201ac690d8bdef"
}

#Tag for advertising, obtainable from @MTProxybot
#AD_TAG = "3c09c680b76ee91a4c25ad51f742267d"
```

اکنون با اجرای دستور زیر کانتینر خود را میسازیم و آنرا اجرا میکنیم، بنابراین سرور ما آماده سرویس دهی MTPROXY خواهد بود:

```
docker-compose up --build -d
```

حال برای دیدن لینک پراکسی خود از دستور زیر استفاده میکنیم:

```
docker-compose logs
```

با اجرای دستور فوق، لینک پراکسی شما نمایش داده خواهد شد. توجه داشته باشید برای هر یوزر دو لینک نمایش داده خواهد که یکی از آنها عبارت را beta همراه خود دارد و سکرت آن با dd شروع میشود. با استفاده از این لینک میتوانید با استفاده از پروتکل جدید MTPROXY، از طریق تلگرام به سرور خود متصل شوید.

## مشکل در اجرای docker-compose:

اگر با اجرای دستور `docker-compose` با خطای `ERROR: Couldn't connect to Docker daemon` مواجه شدید، علت قرار نداشتن یوزر لینوکس شما در گروه کاربری `docker` است. برای رفع این مشکل از `sudo` استفاده کنید و یا دستور زیر را اجرا کنید:

```
sudo usermod -aG docker ${USER}
```

اکنون یکبار `logout` کنید و مجدداً به `SSH` خود متصل شوید. حال میتوانید از `docker-compose` استفاده کنید.