

به محیط خط فرمان لینوکس سرورتون وارد بشید.

برای کارمون نیاز به نصب این دو کتابخانه داریم که در ابونتو یا دبیان از دستور زیر استفاده می‌کنیم:

```
apt install git curl build-essential libssl-dev zlib1g-dev
```

```
vps-vpn-us.tlp - root@ - Bitwise xterm - root@aomk: ~
Last login: Tue Jul 11 02:51:09 2017
root@aomk:~# apt install git curl build-essential libssl-dev zlib1g-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5 gcc-5-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan2 libatomic1 libc-dev-bin libc6
  libc6-dev libcc1-0 libcilkrts5 libcurl3-gnutls libdpkg-perl libfakeroot libfile-fcntllock-perl
  libgcc-5-dev libgomp1 libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libssl-doc
  libssl1.0.0 libstdc++-5-dev libstdc++6 libtsan0 libubsan0 linux-libc-dev make manpages-dev
Suggested packages:
  binutils-doc cpp-doc gcc-5-locales debian-keyring g++-multilib g++-5-multilib gcc-5-doc
  libstdc++6-5-dbg gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-5-multilib
  libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan2-dbg liblsan0-dbg libtsan0-dbg
  libubsan0-dbg libcilkrts5-dbg libmpx0-dbg libquadmath0-dbg git-daemon-run | git-daemon-sysvinit
  git-doc git-el git-email git-gui gitk gitweb git-arch git-cvs git-mediawiki git-svn glibc-doc
  libstdc++-5-doc make-doc
The following NEW packages will be installed:
  binutils build-essential cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan2 libatomic1 libc-dev-bin libc6-dev
  libcc1-0 libcilkrts5 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-5-dev libgomp1
  libisl15 libitm1 liblsan0 libmpc3 libmpx0 libquadmath0 libssl-dev libssl-doc libstdc++-5-dev
  libtsan0 libubsan0 linux-libc-dev make manpages-dev zlib1g-dev
The following packages will be upgraded:
  curl gcc-5-base git libc6 libcurl3-gnutls libssl1.0.0 libstdc++6
7 upgraded, 39 newly installed, 0 to remove and 218 not upgraded.
Need to get 48.3 MB of archives.
After this operation, 154 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

را تایپ و اینتر را بزنید “Y” برای ادامه

استفاده می‌کنید، دستور زیر را وارد کنید RHEL یا CentOS اگر هم از

```
yum install openssl-devel zlib-devel
```

```
yum groupinstall "Development Tools"
```

از کیت‌هاب (Repository) کردن ریپازیتوری Clone

رو می‌توانید از طریق این ریپازیتوری کلن کنید. این رو حدود چند ماهه تست کردم و مشکلی نداره MTPProto فایل‌های اصلی

با دستور زیر ریپازیتوری مورد نظر را کلن کنید که این فایل‌ها به سرور شما منتقل شود:

```
git clone https://github.com/TelegramMessenger/MTPProxy
```

```
root@aomk:~# git clone https://github.com/TelegramMessenger/MTPProxy
Cloning into 'MTPProxy'...
remote: Enumerating objects: 194, done.
remote: Total 194 (delta 0), reused 0 (delta 0), pack-reused 194
Receiving objects: 100% (194/194), 303.45 KiB | 0 bytes/s, done.
Resolving deltas: 100% (82/82), done.
Checking connectivity... done.
```

شوید MTPROTO رو دریافت کردید، با دستور زیر وارد پوشه Done حال که به درستی محتویات به سرور شما منتقل شد و پیغام

```
cd MTPProxy
```

```
Checking connectivity... done.
root@aomk:~# cd MTPProxy
root@aomk:~/MTPProxy#
```

این دستور باید در مسیر پوشه: برای ساخت پوشه و فایلای باینری مورد نیاز برای تنظیمات پروکسی این دستور را وارد کنید (زده شود MTPProxy).

```
make && cd objs/bin
```

```
vps-vpn-us.tlp - root@ - Bitvise xterm - root@aomk: ~/MTPProxy/objs/bin
mon/proc-stat.o -o objs/common/proc-stat.o common/proc-stat.c
cc -O3 -std=gnu11 -Wall -mpclmul -march=core2 -mfpmath=sse -mssse3 -fno-strict-aliasing -fno-strict-overflow -fwrapv -DAES=1 -DCOMMIT="\2c942119c4ee340c80922ba11d14fb3b10d5e654\" -D_GNU_SOURCE=1 -D_F
ILE_OFFSET_BITS=64 -fpic -iquote common -iquote . -c -MP -MD -MF dep/common/kprintf.d -MQ objs/commo
n/kprintf.o -o objs/common/kprintf.o common/kprintf.c
cc -O3 -std=gnu11 -Wall -mpclmul -march=core2 -mfpmath=sse -mssse3 -fno-strict-aliasing -fno-strict-overflow -fwrapv -DAES=1 -DCOMMIT="\2c942119c4ee340c80922ba11d14fb3b10d5e654\" -D_GNU_SOURCE=1 -D_F
ILE_OFFSET_BITS=64 -fpic -iquote common -iquote . -c -MP -MD -MF dep/common/precise-time.d -MQ objs/
common/precise-time.o -o objs/common/precise-time.o common/precise-time.c
cc -O3 -std=gnu11 -Wall -mpclmul -march=core2 -mfpmath=sse -mssse3 -fno-strict-aliasing -fno-strict-overflow -fwrapv -DAES=1 -DCOMMIT="\2c942119c4ee340c80922ba11d14fb3b10d5e654\" -D_GNU_SOURCE=1 -D_F
ILE_OFFSET_BITS=64 -fpic -iquote common -iquote . -c -MP -MD -MF dep/common/cpuid.d -MQ objs/common/
cpuid.o -o objs/common/cpuid.o common/cpuid.c
cc -O3 -std=gnu11 -Wall -mpclmul -march=core2 -mfpmath=sse -mssse3 -fno-strict-aliasing -fno-strict-overflow -fwrapv -DAES=1 -DCOMMIT="\2c942119c4ee340c80922ba11d14fb3b10d5e654\" -D_GNU_SOURCE=1 -D_F
ILE_OFFSET_BITS=64 -fpic -iquote common -iquote . -c -MP -MD -MF dep/common/server-functions.d -MQ o
bjcs/common/server-functions.o -o objs/common/server-functions.o common/server-functions.c
cc -O3 -std=gnu11 -Wall -mpclmul -march=core2 -mfpmath=sse -mssse3 -fno-strict-aliasing -fno-strict-overflow -fwrapv -DAES=1 -DCOMMIT="\2c942119c4ee340c80922ba11d14fb3b10d5e654\" -D_GNU_SOURCE=1 -D_F
ILE_OFFSET_BITS=64 -fpic -iquote common -iquote . -c -MP -MD -MF dep/common/crc32.d -MQ objs/common/
crc32.o -o objs/common/crc32.o common/crc32.c
rm -f objs/lib/libkdb.a && ar rcs objs/lib/libkdb.a objs/common/crc32.o objs/common/pid.o objs/comm
on/sha1.o objs/common/sha256.o objs/common/md5.o objs/common/resolver.o objs/common/parse-config.o o
bjcs/crypto/aesni256.o objs/jobs/jobs.o objs/common/mp-queue.o objs/net/net-events.o objs/net/net-msg
.o objs/net/net-msg-buffers.o objs/net/net-config.o objs/net/net-crypto-aes.o objs/net/net-crypto-dh
.o objs/net/net-timers.o objs/net/net-connections.o objs/net/net-rpc-targets.o objs/net/net-tcp-conn
ections.o objs/net/net-tcp-rpc-common.o objs/net/net-tcp-rpc-client.o objs/net/net-tcp-rpc-server.o
objs/net/net-http-server.o objs/common/tl-parse.o objs/common/common-stats.o objs/engine/engine.o ob
js/engine/engine-signals.o objs/engine/engine-net.o objs/engine/engine-rpc.o objs/engine/engine-rpc-
common.o objs/net/net-thread.o objs/net/net-stats.o objs/common/proc-stat.o objs/common/kprintf.o ob
js/common/precise-time.o objs/common/cpuid.o objs/common/server-functions.o objs/common/crc32.o
cc -o objs/bin/mtproto-proxy objs/mtproto/mtproto-proxy.o objs/mtproto/mtproto-config.o objs/net/net
-tcp-rpc-ext-server.o objs/lib/libkdb.a objs/lib/libkdb.a -ggdb -rdynamic -lm -lrt -lcrypto -lz -lp
thread -lcrypto
root@aomk:~/MTPProxy/objs/bin#
```

حالا وقت تنظیمات پروکسی و ساخت سرویس‌های مورد نظر هست

که با دستور زیر انجام می‌شود (Secret) اتصال به سرورهای تلگرام برای ساخت فایل سکرت

```
curl -s https://core.telegram.org/getProxySecret -o proxy-secret
```

ساخته میشه proxy-secret فایل bin در واقع با این دستور در پوشه

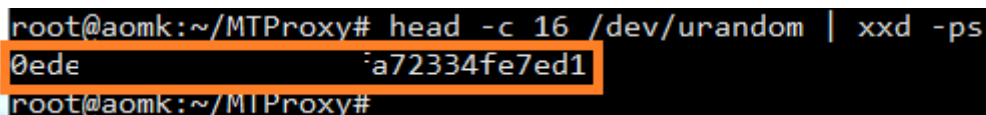
اتصال به سرورهای تلگرام برای ساخت فایل کانفیگ پروکسی که با دستور زیر انجام می‌شود:

```
curl -s https://core.telegram.org/getProxyConfig -o proxy-multi.conf
```

ساخته میشه proxy-multi.conf فایل bin در واقع با این دستور در پوشه

تولید سکرت اختصاصی که اون رو باید کپی و ذخیره کنید. با دستور زیر می‌تونید یک سکرت جنریت کنید:

```
head -c 16 /dev/urandom | xxd -ps
```



```
root@aomk:~/MTPProxy# head -c 16 /dev/urandom | xxd -ps
0ede a72334fe7ed1
root@aomk:~/MTPProxy#
```

مطابق تصویر بالا که ملاحظه می‌کنید، سکرت ساخته شد و اون رو انتخاب و کپی کنید در یک فایل تکست که بعداً لازم داریم.

اجرا کردن mtproto-proxy

با دستور زیر، در تنظیمات پروکسی، یوزرنیم، پورت یوزر، پورت لوکال و سکرت را ست می‌کنید:

```
./mtproto-proxy -u nobody -p 8888 -H 443 -S [secret] --aes-pwd proxy-secret proxy-multi.conf -M 1
```

سکرتی که مرحله قبل بهتون داد را جایگزین کنید، “[secret]” در دستور بالا، بجای

را بزنید cd bin و cd objs وارد کنید. برای اینکار objs/bin دقت داشته باشید که دستور بالا را باید در مسیر

اگر دستور بالا رو درست زده باشید، تصویر زیر برایتان ظاهر میشه.

```

root@aomk:~/MTPProxy# cd objs
root@aomk:~/MTPProxy/objs# cd bin
root@aomk:~/MTPProxy/objs/bin# ./mtproto-proxy -u nobody -p 8888 -H 443 -S 0ede8334fe7ed1
--aes-pwd proxy-secret proxy-multi.conf -M 1
[17906][2019-08-04 02:50:12.351040 local] Invoking engine mtproxy-0.01 compiled at Aug  4 2019 02:15:14 by
gcc 5.4.0 20160609 64-bit after commit 2c942119c4ee340c80922ba11d14fb3b10d5e654
[17906][2019-08-04 02:50:12.353879 local] config_filename = 'proxy-multi.conf'
[17906][2019-08-04 02:50:12.354551 local] creating 1 workers
[17907][2019-08-04 02:50:12.381283 local] Started as [198.143.180.27:8888:17907:1564901412]
[17907][2019-08-04 02:50:12.382193 local] configuration file proxy-multi.conf re-read successfully (752 by
tes parsed), new configuration active
[17907][2019-08-04 02:50:12.386258 local] main loop
[17906][2019-08-04 02:50:12.385090 local] Started as [198.143.180.27:8888:17906:1564901412]
[17906][2019-08-04 02:50:12.390133 local] configuration file proxy-multi.conf re-read successfully (752 by
tes parsed), new configuration active
[17906][2019-08-04 02:50:12.392100 local] main loop

```

ساخت لینک پروکسی برای اشتراک

تقریباً ۹۰ درصد کار پیش رفت و شما طبق الگویی که خود تلگرام در نظر گرفته، لینک پروکسی‌تون رو می‌سازید

```
tg://proxy?server=SERVER_NAME&port=PORT&secret=SECRET
```

باید سکریتی که در چند مرحله قبل “SECRET” سرورتون و همچنین بجای IP باید، “SERVER_NAME” در این لینک بجای ساختید را بزارید.

برای استارت خودکار MTPProxy.service مرحله آخر: پیکربندی و ساخت سرویس

توی این مرحله سرویس امتی‌پروکسی رو می‌سازیم که همیشه باید در حال اجرا باشه که پروکسی‌مون کار کنه. برای ساختش دستور زیر رو می‌زنیم

```
nano /etc/systemd/system/MTPProxy.service
```

```

[Unit]
Description=MTPProxy
After=network.target

[Service]
Type=simple
WorkingDirectory=/opt/MTPProxy
ExecStart=/opt/MTPProxy/mtproto-proxy -u nobody -p 8888 -H 443 -S 0ede4fe7ed1 -S
Restart=on-failure

[Install]
WantedBy=multi-user.target

```

با دستور بالا به محیط ادیتور لینوکس وارد می‌شویم که باید محتویات زیر را واردش کنیم:

```

[Unit]
Description=MTPProxy
After=network.target
[Service]
Type=simple
WorkingDirectory=/root/MTPProxy/objs/bin
ExecStart=/root/MTPProxy/objs/bin/mtproto-proxy -u nobody -p 8888 -H 443
-S [SECRET] --aes-pwd proxy-secret proxy-multi.conf -M 1
Restart=on-failure
[Install]
WantedBy=multi-user.target

```

خیلی مهمه که درست انتخاب بشه. مثلاً بعضی موارد این مسیر بجای اینکه در MTPProxy دقت کنید در دستورات بالا، مسیر پوشه `root/MTPProxy/objs/bin` در مسیر `opt/MTPProxy/objs/bin` باشه، در دستورات بالا اصلاحش می‌کنیم.

را بزنید X + وقتی دستورات بالا را در ادیتور لینوکس نوشتید، دکمه کنترل

دستور زیر را وارد نمایید:

```
systemctl daemon-reload
```

MTPROXY تست سرویس

دستورات زیر به کارتون میاد برای بررسی سرویس پروکسی

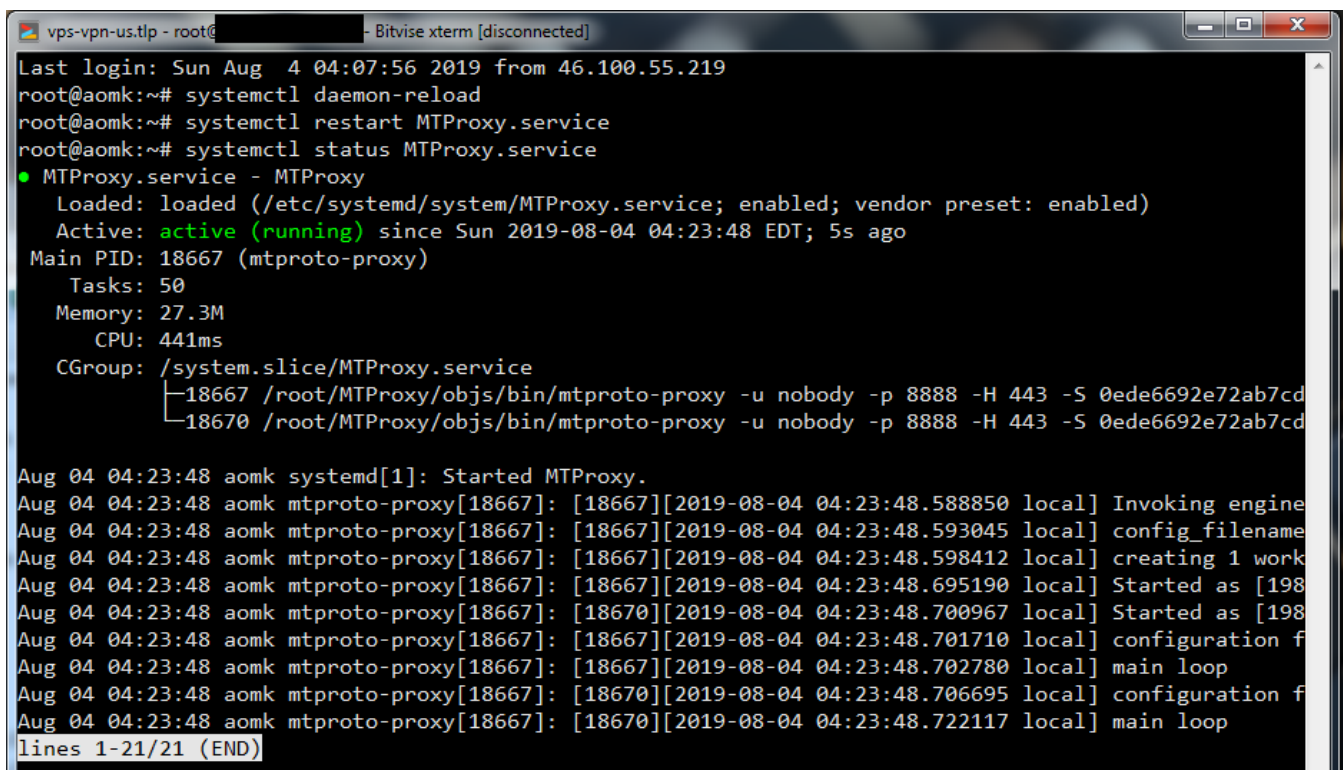
برای ریست کردن سرویس از دستور زیر

```
systemctl restart MTPROXY.service
```

برای بررسی کردن وضعیت سرویس

```
systemctl status MTPROXY.service
```

اگه همه چی رو درست انجام داده باشید با زدن دستور بالا، باید وضعیت مطابق تصویر زیر باشه و اروری دریافت نکنید



```
vps-vpn-us.tlp - root@ - Bitwise xterm [disconnected]
Last login: Sun Aug  4 04:07:56 2019 from 46.100.55.219
root@aomk:~# systemctl daemon-reload
root@aomk:~# systemctl restart MTPROXY.service
root@aomk:~# systemctl status MTPROXY.service
● MTPROXY.service - MTPROXY
   Loaded: loaded (/etc/systemd/system/MTPROXY.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2019-08-04 04:23:48 EDT; 5s ago
 Main PID: 18667 (mtproto-proxy)
    Tasks: 50
   Memory: 27.3M
      CPU: 441ms
   CGroup: /system.slice/MTPROXY.service
           └─18667 /root/MTPROXY/objs/bin/mtproto-proxy -u nobody -p 8888 -H 443 -S 0ede6692e72ab7cd
             18670 /root/MTPROXY/objs/bin/mtproto-proxy -u nobody -p 8888 -H 443 -S 0ede6692e72ab7cd

Aug 04 04:23:48 aomk systemd[1]: Started MTPROXY.
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18667][2019-08-04 04:23:48.588850 local] Invoking engine
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18667][2019-08-04 04:23:48.593045 local] config_filename
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18667][2019-08-04 04:23:48.598412 local] creating 1 work
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18667][2019-08-04 04:23:48.695190 local] Started as [198
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18670][2019-08-04 04:23:48.700967 local] Started as [198
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18667][2019-08-04 04:23:48.701710 local] configuration f
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18667][2019-08-04 04:23:48.702780 local] main loop
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18670][2019-08-04 04:23:48.706695 local] configuration f
Aug 04 04:23:48 aomk mtproto-proxy[18667]: [18670][2019-08-04 04:23:48.722117 local] main loop
lines 1-21/21 (END)
```

برای استارت خودکار این سرویس بعد هر بار ریبوت شدن سرور هم دستور زیر را وارد کنید

```
systemctl enable MTPROXY.service
```

ایجاد کردید. فقط صداشو در نیارید MTPROTO از الان به بعد دیگه میتونید لذت ببرید و به همین سادگی به سرور پروکسی تلگرام