

1. Proxysmart manual.

Hello, I have worked a lot with residential proxies, including building my own network from scratch. I already have the code ready to be deployed, on Linux boxes where USB modems can be attached, up to 21 modem per box. Then these proxies can be sold to the customers. IP rotation is supported.

1. Basic configuration.

Variables are set `/etc/proxysmart/conf.txt`, but default values are in `/etc/proxysmart/source`, don't edit the latter file.

Each variable has brief description.

2. Adding a new modem,

- remove PIN from the modem's SIM card and plug in the modem into USB port or USB hub.
- Check whether your Huawei modem requires authentication, and if it does, set its admin password to `admin123`. Basically to the value of `$DEFAULT_HILINK_ADMIN_PASSWORD` variable. Otherwise many functions will not work, and its IMEI will be detected similarly to 2-1.1.2

if you know its IMEI.

In that case you will assign some specific proxy credentials to it.

- detect IMEI for the modem (check its back side).
- edit `/etc/proxysmart/map.txt` , add new line with new modem, IMEI and proxy ports, login & pass. See 1st line in the file with fields names.
- run `proxysmart.sh reset_gently`. It will detect it & apply settings.
- run `proxysmart.sh show_status` to confirm new creds applied.

if you don't know its IMEI.

In that case some random proxy credentials will be assigned.

- make sure `GENERATE_RANDOM_PROXY_CREDS=1` in `/etc/proxysmart/conf.txt` . It is the default setting.
- run `proxysmart.sh reset_gently`. It will detect new modem & apply random proxy credentials to it. Autogeneration allocates ports in a fully random manner. Autogeneration is only useful to demonstrate the modems are working. It is not supposed to be used on a regular basis.
- run `proxysmart.sh show_status` to confirm new modem is detected. It will have random creds.
- if you want to make them static, edit `/etc/proxysmart/map.txt`, add the modems there, and then run `proxysmart.sh apply_settings_for_a_modem_by_imei_raw Imei` where Imei is Imei of the modem. Then proxy credentials from the `map.txt` will be applied
- run `proxysmart.sh show_status` to confirm new creds applied.

3. Proxy credentials for new modems

When adding new modems to `/etc/proxysmart/map.txt`, please use

- unique HTTP ports from 8001 to 8999,
- unique SOCKS ports from 5001 to 5999.

If you want different ports ranges, update `firewall.conf` accordingly.

- please use unique nicknames like `dongleXXX` or whatever else. Don't use nicknames like `randomXXX`, that are assigned automatically.

4. Where is WebUI

One of

- <http://localhost:8080/>
- http://LAN_IP:8080/
- http://VPS_IP:8080/

By default login/password are `proxy` / `proxy`.

5. How to use proxies

- If proxy ports are forwarded via remote cloud VPS: then the proxies can be used from all over the Internet, by that VPS IP and proxy port numbers.
- From the same LAN where multimodem server is located: by the server's LAN IP and proxy port numbers.

6. Get list of all modems & their external IPs

Run: `proxysmart.sh show_status` for table-alike output.

7. Reconfigure all modems & proxies.

Run: `proxysmart.sh reset_complete`

It is done after reboot automatically by a Cron job.

8. How to change proxy credentials for a modem. How to rename a modem.

- Edit the map file (`/etc/proxysmart/map.txt`), set new port or password or nickname for a modem.
- Then either run `proxysmart.sh apply_settings_for_a_modem_by_imei_raw 999999999999` or:
- click APPLY in the WebUI

9. Reset (change) IP on a modem.

The options are below.

From command line.

- Run: `proxysmart.sh reset_quick_nick dongle1`

Where `dongle1` is a Dongle "nickname" that is seen from output of `proxysmart.sh show_status`

From Web API.

- check `examples/web-api.txt`

How to rotate a modem periodically?

Install a Cron job. Edit a file `/etc/cron.d/proxysmart`, add a line (or uncomment a commented line..)

```
*/* * * * * root run-one /usr/local/bin/proxysmart.sh reset_quick_nick dongle3
```

so that a modem with the Nickname `dongle3` is rotated every 10 min.

Repeat for each modem you want to rotate periodically.

10. How many modems can I run on a single computer?

Hi , technically it depends on how powerful this PC is, and how intensively proxies are used.

- Raspberry PI - 4 proxies (roughly)
- a miniPC (Intel NUC or similar) - up to 10
- a Laptop like Core i5 - up to 30.

Also it depends on what Plan you buy.

Also it depends on USB configuration, for maximum number of modems:

- disable USB3.0 in BIOS
- use USB2.0 hubs

11. How to set TTL and why?

In some cases custom TTL must be set in order to have Cell Operator think we are not using the modem in hotspot \ tethering mode. I.e. we don't share its data. By default Linux OS has `ttl = 64`. To change Cell Operator perception of the situation, we want to set it +1 i.e. 65.

Edit `/etc/proxysmart/conf.txt` and set `CUSTOM_TTL_SET=1` and `CUSTOM_TTL_VALUE=65` and regenerate settings.

12. How to set MTU and why?

In some cases different MTU values connect with different types of ISP's. You may want to change it.

Mtu can be only lowered. E.g. if you have MTU 1390, you can set 1340. Not opposite.

- Option 1. One value for all modems.

Edit `/etc/proxysmart/conf.txt` and set `CUSTOM_MTU_SET=1` , `CUSTOM_MTU=1410`.

- Option 2. Individual values for modems.

The same as above , but also edit `/etc/proxysmart/per_modem_conf.yaml` and add `mtu` value for some modems that need custom value.

13. How to set extra settings for a modem.

Those are optional and are set in YAML file `/etc/proxysmart/per_modem_conf.yaml`

- allowed customers IP's who are not required to type in proxy password (IP-based auth). Those are set in `white_list` array.
- bandwidth (speed) limit. Values are in bits per second. Set them in `bandlimin` and `bandlimout`. E.g. for 2/2 mbps it will be 2000000/2000000.
- `DENIED_SITES_ENABLE` (1 or 0) and `DENIED_SITES_LIST` (array of blocked sites patterns), see examples .
- `bw_quota` , value is in Megabytes
- `mtu` , `ttl`

After changing the file, apply setting for the modem you changed settings for.

14. How can I prevent access to modems web interface via proxy?

Edit `/etc/proxysmart/conf.txt` and set

```
PROXY_ADMIN_ENABLE=1
PROXY_ADMIN_LOGIN=admin
PROXY_ADMIN_PASS=papapa
```

And regenerate configs. So only **admin** user is allowed to use modems web interfaces, and normal proxy users are not.

15. How to set monthly traffic quota per modem?

Edit `/etc/proxysmart/per_modem_conf.yaml` and add `bw_quota` value for some modems that need custom value.

E.g. a line is below, where the modem that IMEI has **2000 Megabytes** monthly quota, from begin to the end of the month. It is applied to both Upload and Download.

```
- { imei: 7777777777777777, bw_quota: 2000 }
```

or

```
-
  imei: 7777777777777777
  bw_quota: 2000
```

16. How to make my proxies Open (i.e. not requiring authentication)

Set `OPEN_PROXIES=1` in `conf.txt` and regenerate all configs.

Note, when proxy ports are forwarded via a VPS, the proxies are available to any internet user. Use it with caution.

17. Get monthly/daily proxy usage.

Run: `proxysmart.sh bandwidth_report_json`, you will see these columns:

Also reports are stored in `/var/lib/3proxy/reports/`. Files are named like `report.$IMEI.YYYY.MM.DD`

Or (if `IPTABLES_COUNTERS=1` in `conf.txt`) you can run

```
proxysmart.sh get_counters_imei IMEI START_TIME END_TIME
```

where `START_TIME END_TIME` is one of `HH:MM`, `now`, `YYYYMMDD` or a UNIX epoch-time, e.g.

```
proxysmart.sh get_counters_imei 866467049999949 12:00 now
{
  "in": "11227626",
  "out": "106423"
}
```

18. How to get current number of connections for a modem?

Run a command

```
ss -o state established | grep -c :8038
```

But change 8038 with HTTP port of a desired proxy

19. How to read SMS from a modem.

You have these options.

1. Browse to the modem IP (it is shown as GW in `proxysmart.sh show_status`) through the proxy. Click SMS button.
2. run `proxysmart.sh list_sms_for_a_modem_by_imei_json 999999999999999` i.e. IMEI of required modem.
3. Click SMS in the WebApp

20. How to change WebUI password for <http://localhost:8080/>

By default it is set to `proxy / proxy`. The password sits on the server's folder `/etc/nginx/`. It Can be updated from the Terminal , with the command as follows:

```
sudo htpasswd -b /etc/nginx/htpasswd proxy NewAwesomePassword999999
```

Then it will ask for password for current Ubuntu user.

21. OS Spoofing

Os Spoofing is used to simulate other OS TCP fingerprints, in most cases make it look like Windows.

How to enable OS Spoofing?

It applies to all modems at once.

- update the code , run `./install_pkgs.sh` , `./install_files.sh` from the root of the repo.
- set `OS_SPOOF=1` in `/etc/proxysmart/conf.txt`.
- optionally set `OSGENRE` and `DETAILS_P0F`, like done in `/etc/proxysmart/source`. Defaults are spoof to Windows.
- run: `proxysmart.sh reset_complete`

How to test OS Spoofing ?

Visit one of these websites (IP checkers) through a proxy. Find something like "OS TCP fingerprints".

- <http://witch.valdikss.org.ru/>
- <https://thesafety.us/>
- <https://Whoer.net> , extended results
- <https://browserleaks.com/ip>

What OS can I spoof?

Just Windows. It is very experimental.

22. Performance tuning

When >10 modems are added, and when modem list is generated slowly, play with `MAX_PARALLEL_WORKERS_STATUS` variable, e.g. set it to 2 or 4. On faster CPU's it can be set to 12.

Also try to disable OS TCP reporting, i.e. set `ENABLE_VALDIK` to 0. It will also make modem list generation faster.

23. How to get more cellular IP's?

Sometimes 3G has another IP pool, but speeds are lower. You can set random Auto(4G),3G rotation method. Check `source` file and adjust `conf.txt`.

24. What if a modem connected via 3G or 2G, and I want 4G?

Rotate its IP.

25. I want to add extra users to a proxy, in addition to those defined in 'map.txt'

Add them to `per_modem_conf.yaml`, check the template. Basically each modem may have an array of extra users with `user:password` definition.

E.g. here 2 extra users are added for a modem with 862329049849999

```
- { imei: 862329049849999 , extra_users: [ john : jjj23 , martin: mmm23 ] }
```

Then apply setting for the modem.

2. Project description

1. project architecture (clients, servers, websites),

- onsite: box with Ubuntu, USB hub and modems
- remote: VPS with proxy ports (optional)

2. what is hosted where and from which repository,

Online services are used:

- <http://ip.tanatos.org/ip.php> which is simple PHP script that returns visitor's IP. It is used to detect whether a modem is really online. Can be replaced with one of <https://ifconfig.co> or similar, but I was not happy with their reliability, they are down sometimes. The URL is defined in `/etc/proxysmart/conf.txt`.
- <http://witch.valdikss.org.ru/> : used for detecting p0f and MTU

Software used to build the box:

- 3proxy 0.8.13 <https://github.com/z3APA3A/3proxy/> ; forwards requests from clients to modems; does authentication; provides HTTP/SOCKS services
- hlcli <https://github.com/kenshaw/hilink> ; gathers info from Huawei modems
- HiPi <https://metacpan.org/pod/HiPi> ; perl module, API for gathering info from Huawei modems and resetting IP's on them
- J2cli <https://pypi.org/project/j2cli/> ; Python module for building 3proxy conf files
- Imgur image hosting for uploading screenshots
- everything else is in official Ubuntu repo (APT).

3. how all these elements communicate with each other,

- Ubuntu box initializes modems, creates Linux networking routing for each.
- 3proxy instance is started for each.
- Proxy ports are exposed to VPS.

4. what are common points of failure and how to deal with them,

- a modem is not recognized => insert it into Win/Mac and check there. Probably there is an issue with SIM.
- IP can't be rotated => check whether you completed setup of the modem in Huawei Web Gui (192.168.8.1) on Windows/Mac and set its admin password to `admin`

5. how to update each and every project element, what is the workflow, how to build and deploy it, what tools to use,

- 3proxy - I am not seeing a reason to update it, because newer releases may bring some incompatibility, but anyways: <https://3proxy.ru/howtoe.asp#GCCUNIX> or check README in the project Github.
- Ubuntu version - it is better not update it, i.e. stick with same version. Security updates are good to apply.

3. CLI API

1. show status

Show full status of all modems, table (slower).

```
# proxysmart.sh show_status
```

NICK	N	DEV	MODEL	IMEI	HTTP	LOCAL_IP	GW	EXT_IP	ONLINE	CELL:MODE	MSG
dongle1	0	modem0	E3372h-320	862329099999999	8001	192.168.8.100	192.168.8.1	46.216.113.63	yes	MTS BY:LTE	
dongle2	1142	modem1142	E3131	352221099999999	8002	192.168.8.100	192.168.8.1		no	:NO_SERVICE	

items TOTAL 2

Show brief status of all modems, table, (faster)

NICK	N	DEV	IMEI	HTTP	LOCAL_IP	GW	EXT_IP	ONLINE	MSG
cdongle2	77	modem77	862329099999999	8002	192.168.8.100	192.168.8.1	46.216.152.241	yes	
Client5	93	modem93	352221099999999	8004	192.168.0.100	192.168.0.1	46.56.186.34	yes	

Show full status of all modems , json

```
# proxysmart.sh show_status_json
```

```
[
  {
    "MSG" : "",
    "N" : "0",
    "modem_details" : {
      "HUB_ID" : "1-1",
      "HUB_PORT" : "1-1",
      "IMEI" : "899999999999999",
      "MODEL" : "E3372h-320",
      "NICK" : "dongle1"
    },
    "net_details" : {
      "CELLOP" : "MTS BY",
      "ConnectionStatus" : "(901) DATA:connected",
      "CurrentNetworkType" : "(101) LTE",
      "DEV" : "modem0",
      "EXT_IP" : "46.216.113.63",
      "GW" : "192.168.8.1",
      "IS_ONLINE" : "yes",
      "LOCAL_IP" : "192.168.8.100",
      "SimStatus" : "(1) valid SIM card",
      "VALDIK" : "Detected OS = Linux 2.2.x-3.x [generic];MTU = 1420;Network link = generic tunnel or VPN;PTR test = Probably home user;Fin",
      "workmode" : "LTE"
    },
    "proxy_creds" : {
      "HTTP_PORT" : "8001",
      "LOGIN" : "alice",
      "PASS" : "cool",
      "SOCKS_PORT" : "5001"
    },
    "redirector_status" : {
      "ActiveState" : "active",
      "NRestarts" : "0",
      "SubState" : "running",
      "UPTIME" : "2min 6s"
    }
  },
  {
    "MSG" : "",
    "N" : "1142",
    "modem_details" : {
      "HUB_ID" : "1-3",
      "HUB_PORT" : "3-1",
      "IMEI" : "352228888888888",
      "MODEL" : "E3131",
      "NICK" : "dongle2"
    }
  }
]
```

```

    },
    "net_details" : {
        "CELLOP" : null,
        "ConnectionStatus" : "(902) DATA:disconnected",
        "CurrentNetworkType" : "(0) NO_SERVICE",
        "DEV" : "modem1142",
        "EXT_IP" : null,
        "GW" : "192.168.8.1",
        "IS_ONLINE" : "no",
        "LOCAL_IP" : "192.168.8.100",
        "SimStatus" : "(255) SIM card is missing",
        "VALDIK" : null,
        "workmode" : "unknown"
    },
    "proxy_creds" : {
        "HTTP_PORT" : "8002",
        "LOGIN" : "alice",
        "PASS" : "cool",
        "SOCKS_PORT" : "5002"
    },
    "redirector_status" : {
        "ActiveState" : "active",
        "NRestarts" : "13",
        "SubState" : "running",
        "UPTIME" : "1s"
    }
}
}
]

```

Show status for a single modem, JSON

```

# proxysmart.sh show_single_status_json dongle111
[
  {
    "IS_LOCKED" : "false",
    "MSG" : "",
    "N" : "115",
    "modem_details" : {
        "HUB_ID" : "1-1",
        "HUB_PORT" : "3",
        "IMEI" : "8999999999999999",
        "MODEL" : "E3372h-320",
        "NICK" : "dongle111",
        "UDEV UPTIME" : "1212172",
        "UPTIME" : "14 days + 43.883333 minutes"
    },
    "net_details" : {
        "CELLOP" : "MTS BY",
        "ConnectionStatus" : "901, DATA:connected OK",
        "CurrentNetworkType" : "(101) LTE",
        "DEV" : "modem115",
        "EXT_IP" : "46.216.224.164",
        "GW" : "192.168.8.1",
        "IS_ONLINE" : "yes",
        "LOCAL_IP" : "192.168.8.100",
        "SIGNAL_STRENGTH" : "4",
        "SimStatus" : "(1) valid SIM card",
        "VALDIK" : "Detected OS = Linux 2.2.x-3.x [generic];MTU = 1420;Network link = generic tunnel or VPN;PTR test = Probably home user;Fin",
        "workmode" : "LTE"
    },
    "proxy_creds" : {
        "HTTP_PORT" : "8004",
        "LOGIN" : "alice",
        "PASS" : "cool",
        "PROXYSTDLINE_LAN" : "192.168.100.2:8004:alice:cool",
        "PROXYSTDLINE_WWW" : "forwarding_disabled",
        "SOCKS_PORT" : "5004"
    },
    "redirector_status" : {
        "MSG" : "redirectors disabled globally"
    }
  }
]

```

2. full reconfiguration

```

# proxysmart.sh reset_complete

= old fake default route deleted
= adding faked default route
= restoring default linux TTL
net.ipv4.ip_default_ttl = 64
=start_connections

= lock acquired on DEV modem0
== [add_individual_dev] generating config for DEV=modem0 N=0 HUB_ID=1-1 HUB_PORT=1-1
= found IP 192.168.8.100 on modem0
= got GW=192.168.8.1
= got IMEI=8999999999999999
= got model E3372h-320

```

```

= got NICK=dongle1
= start 3proxy config generation for N=0
= applying new settings: DEV modem0, N 0, IMEI 899999999999999, nick dongle1, http_port 8001, socks_port 5001, auth: alice / cool
= starting proxy 0 on modem0
= starting redirector@0 on modem0
= lock released on DEV modem0

= lock acquired on DEV modem1142
= [add_individual_dev] generating config for DEV=modem1142 N=1142 HUB_ID=1-3 HUB_PORT=3-1
= found IP 192.168.8.100 on modem1142
= got GW=192.168.8.1
= got IMEI=352228888888888
= got model E3131
= got NICK=dongle2
= start 3proxy config generation for N=1142
= applying new settings: DEV modem1142, N 1142, IMEI 352228888888888, nick dongle2, http_port 8002, socks_port 5002, auth: alice / cool
= starting proxy 1142 on modem1142
= starting redirector@1142 on modem1142
= lock released on DEV modem1142

= finding a live modem for setting as fallback default gateway
= testing modem0
= got EXT_IP 46.216.113.63
= marking that modem as LIVE, as tested
= setting fallback default gateway via modem dongle1 // modem0 // 192.168.8.1

it took 10 seconds to prepare the system

all found modems initialized. after 5 sec delay the status will be shown

```

NICK	N	DEV	MODEL	IMEI	HTTP	LOCAL_IP	GW	EXT_IP	ONLINE	CELL:MODE	MSG
dongle1	0	modem0	E3372h-320	899999999999999	8001	192.168.8.100	192.168.8.1	46.216.113.63	yes	MTS BY:LTE	
dongle2	1142	modem1142	E3131	352228888888888	8002	192.168.8.100	192.168.8.1		no	:NO_SERVICE	

```

items TOTAL 2

```

3. apply setting for a modem by IMEI

JSON output

```

# proxysmart.sh apply_settings_for_a_modem_by_imei 868723023562406
{
  "debug" : "= lock acquired on DEV modem0,= start 3proxy config generation for N=0,= applying new settings: DEV modem0, N 0, IMEI 868723023562406",
  "message" : "",
  "result" : "success"
}

```

Plain text output.

```

proxysmart.sh apply_settings_for_a_modem_by_imei_raw 359999999999999
= lock acquired on DEV modem93
= start 3proxy config generation for N=93
= applying new settings: DEV modem93, N 93, IMEI 359999999999999, nick Client5, http_port 8004, socks_port 5004, auth: alice / cool
= found ALLOWED_CLIENT_IPS=22.22.22.22,22.22.22.11
= got BANDLIMIN 12222
= got BANDLIMOUT 1444444
= got BW_QUOTA 20
= extra users detected: myuser1 : mypassword1,myuser2 : mypassword2
= purging old MTU rules from Iptables for modem N=93
deleted rule 8 from mangle/OUTPUT
= purging old MTU rules from Iptables for modem N=93
= adding MTU rules to Iptables for modem N=93 MTU=1400 MSS=1360
= starting redirector@93 on modem93
= lock released on DEV modem93

```

4. reset IP on a modem

Args: IMEI or NICKNAME.

JSON output:

```

# proxysmart.sh reset_modem_by_imei 899999999999999
{
  "message" : "external ip changed from 46.216.188.74 to 46.216.113.63",
  "ext_ip" : "46.216.113.63",
  "result" : "success",
  "debug" : "= lock acquired on DEV modem0,= resetting DEV modem0 on HUB_ID 1-1, HUB_PORT 1-1 local IP 192.168.8.100 N 0 GW 192.168.8.1 IMEI 899999999999999"
}

```

Plain text output:

```

# proxysmart.sh reset_quick_nick Client5
= lock acquired on DEV modem93
= resetting NICK Client5 DEV modem93 on HUB_ID 1-1.2, HUB_PORT 3 local IP 192.168.0.100 N 93 GW 192.168.0.1 IMEI 359999999999999

```



```

= external IP is 46.56.178.172
=stopping redirector N 93
...
=DNS test attempt 2/7 to DNS server 1.1.1.1
Checking/setting forced routing config (skip with /etc/proxysmart/altnetworking.sh -s ...)
Applying net_cls class identifier 0x0010093 to cgroup cgproxy93
Unset reverse path filtering for interface "all"
Unset reverse path filtering for interface "modem93"
DNS OK - 0.092 seconds response time (gmail.com. 276 IN TXT "globalsign-smime-dv=CDYX+XFHUw2wmL6/Gb8+59BsH31KzUr6c1L2BPvqKX8=") |time=0.092426
= passed
= restarting proxy@93 to definitely drop old connections..
= starting redirector N 93
=now detect EXT_IP
= external IP is 46.56.181.222
= purging old MTU rules from Iptables for modem N=93
deleted rule 9 from mangle/OUTPUT
= purging old MTU rules from Iptables for modem N=93
= adding MTU rules to Iptables for modem N=93 MTU=1400 MSS=1360
=save report:
start_time=2022-05-29@21:14:43 end_time=2022-05-29@21:15:13 total_time=27 old_ip=46.56.178.172 new_ip=46.56.181.222 target_mode=auto
= lock released on DEV modem93

```

5. reboot a modem

Args: Nickname or IMEI.

```

# proxysmart.sh reboot_modem dongle61_us
and
# proxysmart.sh reboot_modem 899999999999999

```

6. Run speedtest on all modems at once

```

# proxysmart.sh speedtest all
[
  {
    "IMEI" : "352228888888888",
    "N" : "1142",
    "NICK" : "dongle2",
    "test" : {
      "msg" : "some_error"
    }
  },
  {
    "IMEI" : "899999999999999",
    "N" : "0",
    "NICK" : "dongle1",
    "test" : {
      "download" : "5.9mbps",
      "share" : "http://www.speedtest.net/result/11130520118.png",
      "upload" : "12.3mbps"
    }
  }
]

```

7. report bandwidth

On a single modem. Args: NICKNAME or IMEI.

```

# proxysmart.sh bandwidth_report_json 869076043182393
[
  {
    "IMEI" : "869076043182393",
    "NICK" : "dongle2",
    "bandwidth_bytes_day_in" : "3482408",
    "bandwidth_bytes_day_out" : "460261",
    "bandwidth_bytes_month_in" : "18163459",
    "bandwidth_bytes_month_out" : "2929636",
    "bandwidth_bytes_yesterday_in" : "3924623",
    "bandwidth_bytes_yesterday_out" : "625495"
  }
]

```

On all modems:

```

# proxysmart.sh bandwidth_report_json_all
[
  {
    "IMEI" : "352228888888888",
    "NICK" : "dongle2",
    "bandwidth_bytes_day_in" : "1202",
    "bandwidth_bytes_day_out" : "322",
    "bandwidth_bytes_month_in" : "10729051",
    "bandwidth_bytes_month_out" : "689922",
    "bandwidth_bytes_yesterday_in" : null,

```

```

    "bandwidth_bytes_yesterday_out" : null
  },
  {
    "IMEI" : "899999999999999",
    "NICK" : "dongle1",
    "bandwidth_bytes_day_in" : "5254",
    "bandwidth_bytes_day_out" : "3866",
    "bandwidth_bytes_month_in" : "19502452",
    "bandwidth_bytes_month_out" : "1376472",
    "bandwidth_bytes_yesterday_in" : null,
    "bandwidth_bytes_yesterday_out" : null
  }
]

```

8. reset bandwidth counter on a modem

```

# proxysmart.sh bandwidth_reset_counter dongle4
{"result":"success","debug":null}

```

9. list sms on a modem

```

# proxysmart.sh list_sms_json 869086046197801
[
  {
    "Date" : "2021-07-08 14:05:23",
    "Content" : "Your free month has started. Keep referring friends to get more free months. https://smarty.co.uk/dashboard",
    "Index" : "40001",
    "Phone" : "SMARTY"
  },
  {
    "Date" : "2021-07-12 10:23:47",
    "Content" : "621036 is your SMARTY login verification code.\n\nREMEMBER: We won't ask you for this over the phone. Don't share it with a",
    "Index" : "40002",
    "Phone" : "SMARTY"
  }
]

```

10. send sms

Plain output:

```

# proxysmart.sh send_sms_raw 899999999999999 +375293511066 "ypa ypa 333"
= Logging in with admin:admin123
= preparing token
= Logged in
= Sending the following message to {+375293511066}: {ypa ypa 333}
= preparing token
= SENT OK
= Logging OUT
= preparing token
= RESPONSE=OK

```

JSON output:

```

# proxysmart.sh send_sms_json 899999999999999 +375293511066 "ypa ypa 333"
{
  "debug" : "= Logging in with admin:admin123,= preparing token,= Logged in ,= Sending the following message to {+375293511066}: {ypa ypa 333}",
  "result" : "success"
}

```

11. send ussd

Plain output:

```

# proxysmart.sh send_ussd_raw 899999999999999 '*100#'
= Logging in with admin:admin123
= preparing token
= Logged in
= sending USSD *100#
= preparing token
= SENT OK
= getting response. attempt 1
= preparing token
= not yet response received
= getting response. attempt 2
= preparing token
= not yet response received
= getting response. attempt 3
= preparing token
= OK response received
Zapros nedostupen na vashem TP, naberite *120#vyzov.
= Logging OUT

```

```
= preparing token
= RESPONSE=OK
```

JSON output:

```
# proxysmart.sh send_ussd_json 8999999999999 '*100#'
{
  "RESPONSE" : "Zapros nedostupen na vashem TP, naberite *120#vzvov.",
  "debug" : "= Logging in with admin:admin123,= preparing token,= Logged in ,= sending USSD *100#,= preparing token,= SENT OK,= getting respo",
  "result" : "success"
}
```

12. get bandwidth counters from a modem

Iptables counters must be configured (IPTABLES_COUNTERS=1) for that.

```
# proxysmart.sh get_counters_imei 866999999999949 12:00 now
{
  "in": "11227626",
  "out": "106423"
}
```

4. WEB API

1. Web API description.

localhost:8080 is the URL that Proxysmart Web-App sits on. Basically it is the URL that you can browse Proxysmart WebApp with.

It can be also \$LanIP:8080 or when you forward HTTP port 8080 to you static Home IP, it will became as well YourStaticIP:8080 . Use any endpoint that suits best.

Also attach proper username:password (the -u parameter).

2. List all modems (full status, slow)

Request:

```
curl 'http://localhost:8080/modems' -u proxy:proxy
```

Response:

```
{
  "message": null,
  "modems": [
    {
      "cellMode": "LTE",
      "cellOp": "A1 BY",
      "dev": "modem115",
      "extIp": "46.56.228.215 ",
      "imei": "8999999999999999",
      "model": "E3372h-320",
      "modemIp": "192.168.8.1",
      "nick": "random7",
      "proxyCreds": [
        "http: 28007:def:def",
        "socks: 25007:def:def"
      ],
      "redirectorStatus": "redirectors disabled globally",
      "simStatus": "(1) valid SIM card",
      "statusMessage": "",
      "valdik": "Detected OS = Linux 2.2.x-3.x [generic];MTU = 1400;
        Network link = Probably IPsec or other VPN;
        PTR test = Probably home user;Fingerprint and OS match. No proxy detected ;
        No OpenVPN detected."
    }
  ],
  "success": true
}
```

3. List all modems (brief status, fast)

Request:

```
curl localhost:8080/apix/show_status_brief_json -u proxy:proxy
```

Response:

```
[
{
  "MSG": "",
  "N": "172",
  "IS_LOCKED": "false",
  "modem_details": {
    "NICK": "dongle2",
    "IMEI": "352228888888888"
  },
  "net_details": {
    "DEV": "modem172",
    "GW": "192.168.8.1",
    "LOCAL_IP": "192.168.8.100",
    "EXT_IP": "46.216.112.104",
    "IS_ONLINE": "yes"
  },
  "proxy_creds": {
    "HTTP_PORT": "8003",
    "SOCKS_PORT": "5003",
    "LOGIN": "alice",
    "PASS": "cool",
    "PROXYSTDLINE_LAN": "192.168.100.6:8003:alice:cool",
    "PROXYSTDLINE_WWW": "forwarding_disabled"
  },
  "redirector_status": {
    "MSG": "redirectors disabled globally"
  }
}
]
```

4. Single modem status

Request:

(either by IMEI or Nickname)

```
curl http://localhost:8080/apix/show_single_status_json?arg=dongle111 -u proxy:proxy -Ss -v
curl http://localhost:8080/apix/show_single_status_json?arg=8999999999999999 -u proxy:proxy -Ss -v
```

Response:

```
[
{
  "IS_LOCKED" : "false",
  "MSG" : "",
  "N" : "115",
  "modem_details" : {
    "HUB_ID" : "1-1",
    "HUB_PORT" : "3",
    "IMEI" : "8999999999999999",
    "MODEL" : "E3372h-320",
    "NICK" : "dongle111",
    "UDEV UPTIME" : "1212291",
    "UPTIME" : "14 days + 45.866667 minutes"
  },
  "net_details" : {
    "CELLOP" : "MTS BY",
    "ConnectionStatus" : "901, DATA:connected OK",
    "CurrentNetworkType" : "(19) LTE",
    "DEV" : "modem115",
    "EXT_IP" : "46.216.224.164",
    "GW" : "192.168.8.1",
    "IS_ONLINE" : "yes",
    "LOCAL_IP" : "192.168.8.100",
    "SIGNAL_STRENGTH" : "4",
    "SimStatus" : "(1) valid SIM card",
    "VALDIK" : "Detected OS = Linux 2.2.x-3.x [generic];MTU = 1420;
    Network link = generic tunnel or VPN;PTR test = Probably home user;
    Fingerprint and OS match. No proxy detected ;No OpenVPN detected.",
    "workmode" : "LTE"
  },
  "proxy_creds" : {
    "HTTP_PORT" : "8004",
    "LOGIN" : "alice",
    "PASS" : "cool",
    "PROXYSTDLINE_LAN" : "192.168.100.2:8004:alice:cool",
    "PROXYSTDLINE_WWW" : "forwarding_disabled",
    "SOCKS_PORT" : "5004"
  },
  "redirector_status" : {
    "MSG" : "redirectors disabled globally"
  }
}
]
```

5. Reset (change) IP on a modem.

Request:

(either by IMEI or Nickname)

```
curl http://localhost:8080/apix/reset_modem_by_imei?IMEI=8999999999999999 -u proxy:proxy
curl http://localhost:8080/apix/reset_modem_by_nick?NICK=dongle22 -u proxy:proxy
```

Response:

```
{
  "debug" : "...",
  "ext_ip" : "46.216.248.48",
  "message" : "external ip changed from 46.216.225.112 to 46.216.248.48",
  "result" : "success"
}
```

6. Apply settings for a modem

Request:

```
curl http://localhost:8080/modem/settings -d imei=8623290999999999 -u proxy:proxy
```

Response:

```
{
  "message": "Result: success, message: ",
  "success": true
}
```

7. Reboot a modem

Request:

(either by IMEI or Nickname)

```
curl http://localhost:8080/apix/reboot_modem_by_imei -d IMEI=860493043888886 -u proxy:proxy
curl http://localhost:8080/apix/reboot_modem_by_nick -d NICK=dongle2 -u proxy:proxy
```

Response:

```
{
  "debug" : "...",
  "message" : "new external ip cannot be detected",
  "result" : "failure"
}
```

or

```
{
  "debug" : "...",
  "ext_ip" : "172.58.172.255",
  "message" : "external ip changed from 172.58.172.251 to 172.58.172.255",
  "result" : "success"
}
```

ETA: ~ 1.5 minute

8. Send SMS

Request:

```
curl 'http://localhost:8080/modem/send-sms' -u proxy:proxy \
  --data-urlencode 'imei=8999999999999999' \
  --data-urlencode 'phone=+375293511066' \
  --data-urlencode "sms=txt txt fff"
```

Response:

```
{"message": "Result: success", "success": true}
```

9. Send USSD and read response

Request:

```
curl 'http://localhost:8080/modem/send-ussd' -u proxy:proxy \
  --data-urlencode 'imei=8999999999999999' --data-urlencode 'ussd=*100#'
```

Response:

```
{
  "RESPONSE": "Zapros nedostupen na vashem TP, naberite *120#vyzov.",
  "debug": "...",
  "result": "success",
  "success": true
}
```

10. Read SMS from a modem

Request:

```
curl 'http://localhost:8080/modem/sms/8623298888888888?json=1' -u proxy:proxy -Ss | json_pp
```

Response:

```
{
  "data" : [
    {
      "Content" : "Вам звонили: +33333333370 в 10:45 22/07.",
      "Date" : "2020-07-22 14:59:35",
      "Index" : "40001",
      "Phone" : "+33333333370"
    },
    {
      "Content" : "Добро пожаловать в Открытый Интернет! Ваш интервал: 0-100МБ. Стоимость - 1,80 руб. Списание оплаты - 1-го числа каждого",
      "Date" : "2021-02-27 00:53:11",
      "Index" : "40002",
      "Phone" : "MTS.BY"
    },
    {
      "Content" : "Hh",
      "Date" : "2021-07-16 20:32:11",
      "Index" : "40042",
      "Phone" : "+375293511066"
    }
  ],
  "success" : true
}
```

11. Read bandwidth stats from a modem

Request:

```
curl localhost:8080/apix/bandwidth_report_json?IMEI=8999999999999999 -u proxy:proxy
```

Response:

```
[
  {
    "IMEI" : "8999999999999999",
    "NICK" : "dongle111",
    "bandwidth_bytes_day_in" : "2945",
    "bandwidth_bytes_day_out" : "2314",
    "bandwidth_bytes_month_in" : "62859",
    "bandwidth_bytes_month_out" : "49559",
    "bandwidth_bytes_yesterday_in" : "5048",
    "bandwidth_bytes_yesterday_out" : "3984"
  }
]
```

12. Read bandwidth stats from a modem, arbitrary time interval

Request:

```
curl http://localhost:8080/apix/get_counters_imei -d IMEI=866000000999999 -d START=12:00 -d END=now -u proxy:proxy
```

Response:

```
{ "in": "11227626", "out": "106423" }
```

13. Read bandwidth stats from all modems

Request:

```
curl localhost:8080/apix/bandwidth_report_json_all -u proxy:proxy
```

Response:

```
[
  {
    "IMEI" : "8999999999999999",
    "NICK" : "dongle111",
    "bandwidth_bytes_day_in" : "2945",
    "bandwidth_bytes_day_out" : "2314",
    "bandwidth_bytes_month_in" : "62859",
    "bandwidth_bytes_month_out" : "49559",
    "bandwidth_bytes_yesterday_in" : "5048",
    "bandwidth_bytes_yesterday_out" : "3984"
  },
  {
    "IMEI" : "862329041089999",
    "NICK" : "dongle111",
    "bandwidth_bytes_day_in" : "1295",
    "bandwidth_bytes_day_out" : "1234",
    "bandwidth_bytes_month_in" : "16259",
    "bandwidth_bytes_month_out" : "49259",
    "bandwidth_bytes_yesterday_in" : "5018",
    "bandwidth_bytes_yesterday_out" : "3294"
  }
]
```

14. Reset bandwidth stats for a modem

Request (by IMEI or nickname):

```
curl localhost:8080/apix/bandwidth_reset_counter?arg=dongle111 -u proxy:proxy -Ss -v
curl localhost:8080/apix/bandwidth_reset_counter?arg=2727233671671676 -u proxy:proxy -Ss -v
```

Response:

```
{"result":"success","debug":null}
```

15. Reset a modem via USB

Request either

- by network interface e.g. modem77
- by Nickname
- by IMEI

```
curl localhost:8080/apix/usb_reset_modem_json?arg=modem77 -u proxy:proxy
curl localhost:8080/apix/usb_reset_modem_json?arg=dongle22 -u proxy:proxy
curl localhost:8080/apix/usb_reset_modem_json?arg=8688888888888889 -u proxy:proxy
```

Response:

```
{"USB_RESET_METHOD": "uhubctl",
 "debug": ".....",
 "result": "ok"}
```

5. Mongodb integration

Instead of defining modems details in map.txt , you can use MongoDB.

Mongodb contains a collection `modems` with elements, 1 element = 1 modem.

Mandatory fields are

- IMEI
- name
- http_port
- socks_port
- proxy_login
- proxy_password

Create a file `modems.json` with 2 modems :

```
{
  "IMEI": "869076044374692",
  "name": "dongle4",
  "http_port": "8004",
  "socks_port": "5004",
```

```

"proxy_login": "mokos",
"proxy_password": "rQ1h6J",
"white_list": [
  "78.140.162.201",
  "78.140.162.202"
],
"bandlimin":1000000,
"bandlimout":1000000,
"DENIED_SITES_ENABLE": 1
"DENIED_SITES_LIST": [
  "bad.com",
  "/*.bad.com"
],
"bw_quota": 2000,
"mtu": 1400,
"extra_users": [ { "myuser1": "mypassword1" }, { "myuser2": "mypassword2" } ]
}
{
"IMEI": "869076042472821",
"name": "dongle5",
"http_port": "8005",
"socks_port": "5005",
"proxy_login": "kileq",
"proxy_password": "Jdh27dh"
}

```

Install MongoDB and database

```

apt install mongodb mongo-tools
mongo
> use proxysmart
> db.createUser( { user: "proxysmart", pwd: "zwKoI118BF", roles: [ { role: "readWrite", db: "proxysmart" } ] })
> exit

```

Then import the collection to the DB

```

mongoimport --uri=mongodb://proxysmart:zwKoI118BF@localhost:27017/proxysmart -c modems < modems.json --drop

```

Update mongodb uri in `/etc/proxysmart/conf.txt`

Set `DB_BACKEND=mongo` there

Regenerate all config files:

```

proxysmart.sh reset complete

```

So it will detect modems and look up for values from MongoDB.

6. Installation

1. Install DEB package

Install a fresh Ubuntu 20.04 or 18.04.

First, install a prereq package `sudo apt install gpg`

Get a DEB package from Developer. Or download from

- X86_64, for a PC\laptop: https://pathos.tanatos.org/proxysmart/proxysmart-latest_amd64.deb
- Arm64, for a Raspberry PI: https://pathos.tanatos.org/proxysmart/proxysmart-latest_arm64.deb

Then `sudo dpkg -i X.deb` Where X.deb is a package file you downloaded.

It will tell what to do next (run 2 files).

```

sudo /usr/lib/proxysmart/install_pkgs.sh
sudo /usr/lib/proxysmart/install_webapp.sh

```

After that check License section.

2. Building DEB package from source code

only make sense when you have got the source code

In a folder of source code run

```

sudo ./build_deb.sh

```


after building, find a file `../proxysmart*deb` and `sudo dpkg -i ../proxysmart*deb`

Then check section above "Install DEB package"

7. License

1. Demo license

Installation is shipped with default **demo** license.

It allows you to run proxy on 1 modem.

In order to run more modems, ask the developer for an extra license, send him the MachineData field from `proxysmart.sh license_status` output and he will issue new license and you will install it.

2. New license installation

You will be given the **license** and **license signature**. Both are sequences of numbers and characters. Then submit both either via WEB gui or CLI:

submitting via CLI

run commands

```
proxysmart.sh submit_license LICENSE
proxysmart.sh submit_license signature LICENSE SIGNATURE
```

submitting via WEB GUI

Open <http://localhost:8080> , unwrap License section and type in the keys & submit.

3. Restoring demo license.

If your paid license expired or broken: find files

- `/usr/share/doc/proxysmart/examples/license.txt`
- `/usr/share/doc/proxysmart/examples/license.txt.sig`

copy them to `/etc/proxysmart/` .