



myLG

Information Security Inc.

# Contents

- About myLG
- Demo Setup
- Features
- Required dependency
- Installing myLG
- Using myLG
- References

# About myLG

- myLG is an open source software utility which combines the functions of the different network probes in one network diagnostic tool



**myLG, Command line Network Diagnostic Tool**

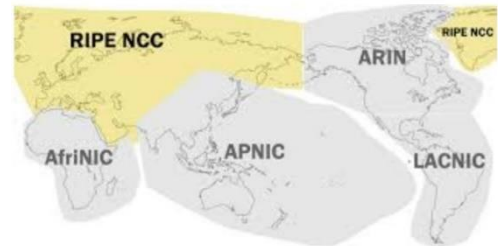
# Demo Setup

- Setup
- Kali Linux 2017

```
root@LUCKY64:/opt3# cat /etc/*rel*  
DISTRIB_ID=Kali  
DISTRIB_RELEASE=kali-rolling  
DISTRIB_CODENAME=kali-rolling  
DISTRIB_DESCRIPTION="Kali GNU/Linux Rolling"  
PRETTY_NAME="Kali GNU/Linux Rolling"  
NAME="Kali GNU/Linux"  
ID=kali  
VERSION="2017.2"  
VERSION_ID="2017.2"  
ID_LIKE=debian  
ANSI_COLOR="1;31"  
HOME_URL="http://www.kali.org/"  
SUPPORT_URL="http://forums.kali.org/"  
BUG_REPORT_URL="http://bugs.kali.org/"
```

# Features

- Popular looking glasses (ping/trace/bgp): Telia, Level3, NTT, Cogent, KPN
- More than 200 countries DNS Lookup information
- Local ping and real-time trace route
- Packet analyzer - TCP/IP and other packets
- Quick NMS (network management system)
- Local HTTP/HTTPS ping (GET, POST, HEAD)
- RIPE information (ASN, IP/CIDR)
- PeeringDB information



# Features

- Port scanning
- Network LAN Discovery
- Internet Speed Test
- Web dashboard
- Configurable options
- Direct access to commands from shell
- Support vi and emacs mode, almost all basic features
- CLI auto complete and history features



# Required dependency

- libpcap-dev

- LINUX

apt-get install libpcap-dev

```
root@LUCKY64: /opt # apt-get install libpcap-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpcap0.8-dev
The following NEW packages will be installed:
  libpcap-dev libpcap0.8-dev
```

# Required dependency

- golang

- LINUX

apt-get install golang

```
root@LUCKY64:~# apt-get install golang
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  golang-1.8 golang-1.8-doc golang-1.8-go golang-1.8-src golang-doc golang-go golang-src pkg-config
```



# Installing myLG

- Clone GitHub repository

```
root@LUCKY64:/opt3# go get github.com/mehrdadrad/mylg

root@LUCKY64:/opt3#
root@LUCKY64:/opt3# git clone https://github.com/mehrdadrad/mylg.git
Cloning into 'mylg'...
remote: Counting objects: 16244, done.
remote: Total 16244 (delta 0), reused 0 (delta 0), pack-reused 16244
Receiving objects: 100% (16244/16244), 28.42 MiB | 1.86 MiB/s, done.
Resolving deltas: 100% (4528/4528), done.
root@LUCKY64:/opt3# cd mylg/
root@LUCKY64:/opt3/mylg# ls
banner  data  Dockerfile  icmp  LICENSE  nms  packet  README.md  scan  speedtest  whois
cli     disc  http        lg    mylg.go  ns   peeringdb  ripe    services  ssh
```

# Installing myLG

- Build myLG

```
root@LUCKY64:/opt3/mylg# go build mylg.go
root@LUCKY64:/opt3/mylg#
root@LUCKY64:/opt3/mylg#
root@LUCKY64:/opt3/mylg#
root@LUCKY64:/opt3/mylg# ls
banner  data  Dockerfile  icmp  LICENSE  mylg.go  ns  peeringdb  ripe  services  ssh
cli     disc  http        lg    mylg     nms      packet  README.md  scan  speedtest  whois
```

# Using myLG

- Run myLG

```
root@LUCKY64:/opt3/mylg# ./mylg
=====
                    myLG
                    My Looking Glass
                    Free Network Diagnostic Tool
                    http://mylg.io
                    myLG v0.2.7 =====
local>
```

# Using myLG

- myLG help menu

```
local> help
Usage:
  The myLG tool, developed to troubleshoot networking situations.
  The vi/emacs mode, almost all basic features are supported. Press tab to see which options are available.

  connect <provider name>    connects to external looking glass, press tab to see the menu
  node <city/country name>   connects to specific node at current looking glass, press tab to see the available nodes
  local                      back to local
  lg                          change mode to external looking glass
  ns                          change mode to name server looking up
  ping                        ping ip address or domain name
  trace                       trace ip address or domain name (real-time w/ -r option)
  dig                         nameserver look up
  nms                         quick NMS - monitor device/server ports real-time
  whois                       resolve AS number/IP/CIDR to holder (provided by ripe ncc)
  hping                       ping through HTTP/HTTPS w/ GET/POST/HEAD methods
  scan                        scan tcp ports (you can provide range >scan host minport maxport)
  dump                        prints out a description of the contents of packets on a network interface
  disc                        discover all the devices on a LAN
  peering                     peering information (provided by peeringdb.com)
  web                         web dashboard - opens dashboard at your default browser

Please visit http://mylg.io/doc for more information
```

# Using myLG

- myLG hping

```
local> hping https://isec.ne.jp -trace -c 4
HPING isec.ne.jp (160.16.83.101), Method: HEAD, DNSLookup: 0.0017 ms
HTTP Response seq=0, proto=HTTP/1.1, status=200, time=1048.721 ms, connection=0.000 ms, first byte read=1047.682 ms
HTTP Response seq=1, proto=HTTP/1.1, status=200, time=994.251 ms, connection=0.000 ms, first byte read=993.201 ms
HTTP Response seq=2, proto=HTTP/1.1, status=200, time=850.786 ms, connection=0.000 ms, first byte read=849.812 ms
HTTP Response seq=3, proto=HTTP/1.1, status=200, time=726.849 ms, connection=0.000 ms, first byte read=726.376 ms

--- isec.ne.jp HTTP ping statistics ---
4 requests transmitted, 4 replies received, 0% requests failed
HTTP Round-trip min/avg/max = 726.85/831.49/1048.72 ms
HTTP Code [200] responses : [██████████] 100.00%
```

# Using myLG

- myLG port scanning

```
local> scan hackyourselffirst.troyhunt.com -p 1-1000
Scan hackyourselffirst.troyhunt.com (137.117.17.70) TCP ports 1-1000
please wait .
+-----+-----+-----+
| PROTOCOL | PORT | STATUS |
+-----+-----+-----+
| TCP      | 80   | Open   |
| TCP      | 443  | Open   |
| TCP      | 454  | Open   |
| TCP      | 455  | Open   |
+-----+-----+-----+
Scan done: 4 opened port(s) found in 96.366 seconds
```

# Using myLG

- myLG packet capture

```
local> dump help

usage:
  dump [filter expression] [options]
  * The expression consists of one or more primitives (Berkeley Packet Filter (BPF) syntax)

options:
  -c count      Stop after receiving count packets (default: 1M)
  -i interface  Listen on specified interface (default: first non-loopback)
  -w filename   Write packets to a pcap format file
  -d           Print list of available interfaces
  -t           Print without timestamp on each dump line.
  -x           Dump payload in hex format
  -s keyword   Search keyword at payload
  -n           Don't convert host addresses to names
  -nc         Shows dumps without color

Example:
  dump tcp and port 443 -c 1000
  dump !udp
  dump -i eth0
  dump -w /tmp/my pcap
  dump tcp -s verisign -x
```

# Using myLG

- myLG packet capture

```
local> dump -s http -x
Interface: eth0, capture size: 6144 bytes
21:13:22.314 IPv4/TCP :56770 > 137.117.17.70:80(http) [P.], win 229, len: 94
00000000 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a |GET / HTTP/1.1.|
00000010 48 6f 73 74 3a 20 68 61 63 6b 79 6f 75 72 73 65 |Host: hackyourse|
00000020 6c 66 66 69 72 73 74 2e 74 72 6f 79 68 75 6e 74 |lffirst.troyhunt|
00000030 2e 63 6f 6d 0d 0a 55 73 65 72 2d 41 67 65 6e 74 |.com..User-Agent|
00000040 3a 20 63 75 72 6c 2f 37 2e 35 30 2e 31 0d 0a 41 |: curl/7.50.1..A|
00000050 63 63 65 70 74 3a 20 2a 2f 2a 0d 0a 0d 0a |ccept: /*/*....|

21:13:22.383 IPv4/TCP 137.117.17.70:80(http) > :56770 [P.], win 514, len: 4024
00000000 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d |HTTP/1.1 200 OK.|
00000010 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 |.Cache-Control: |
```



# Using myLG

- myLG web dashboard



The image shows a terminal window on the left and a web browser window on the right. The terminal shows the following commands and output:

```
local>
local>
local> web
opening default web browser ...
local> 
```

The web browser window displays the myLG web dashboard. The address bar shows the URL `127.0.0.1:8080/#/`. The browser's bookmark bar includes links to `Offensive Security`, `Kali Linux`, `Kali Docs`, `Kali Tools`, `Exploit-DB`, `Aircrack-ng`, `Kali Forums`, `NetHunter`, `Most Visited`, and `Offensive`. The main content area of the dashboard includes:

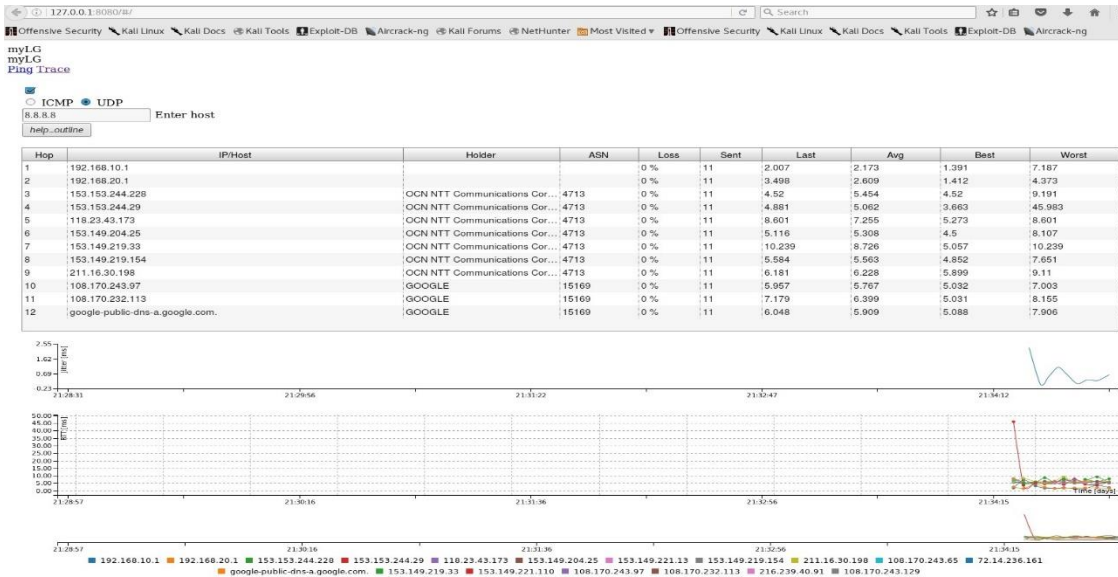
- myLG
- myLG
- [Ping Trace](#)
- ICMP  UDP
- Enter host
- [help\\_outline](#)

Below the form is a table with the following columns:

Hop	IP/Host	Holder	ASN	Loss	Sent	Last	Avg	Best	Worst

# Using myLG

- myLG web dashboard



# References

- Kitploit  
<http://www.kitploit.com/2016/11/mylg-network-diagnostic-tool.html>
- Kali Linux  
<https://www.kali.org/>
- Looking Glass server  
[https://en.wikipedia.org/wiki/Looking\\_Glass\\_server](https://en.wikipedia.org/wiki/Looking_Glass_server)