NAME

yarrp — high-speed active IPv4/IPv6 topology prober

SYNOPSIS

yarrp [**-hvQT**] [**-b** *bgp_rib*] [**-c** *tr_count*] [**-i** *target_file*] [**-m** *max_ttl*] [**-n** *nbr_ttl*] [**-o** *outfile*] [**-r** *rate*] [**-I** *interface*] [**-s** *sequential*] [**-S** *seed*] [**-p** *dst_port*] [**-t** *tr_type*] [**-M** *dst_mac*] [**-G** *src_mac*] [*subnet(s)*]

DESCRIPTION

yarrp (Yelling at Random Routers Progressively) is a high-speed active traceroute-style topology discovery tool. To achieve its high probing rates, yarrp is stateless and randomizes the order of probed destinations and TTLs. By spreading probes, yarrp distributes load and attempts to avoid various forms of rate-limiting. Yarrp supports both IPv4 and IPv6 and can send probes of any transport type (TCP, UDP, or ICMP).

OPTIONS

The set of IPv4 or IPv6 destination targets to probe may be specified in one of three ways:

subnet(s)
: Specify subnet(s) on command-line. A target in each /24 (IPv4), or each /48 (IPv6), of the subnets will be probed.

-i *target_file*
: Input list (one address per line) of explicit targets

-Q
: Internet-wide scanning. Probes a target in all /24 IPv4, or all /48 IPv6, addresses (use with caution).

The general options are as follows:

-h
: print command line options and a synopsis of each.

-v
: enable verbose output

-T
: test mode (default: off)

-o *outfile*
: output ytr file for probing results (default: output.yrp)

-r *rate*
: set packet per second probing rate (default: 10pps)

-t *tr_type*
: set probe type: TCP_ACK, TCP_SYN, UDP, ICMP (default: TCP_ACK)

-c *tr_count*
: set number of traces to issue (default: unlimited)

-m *max_ttl*
: set maximum TTL (default: 32)

-n *nbr_ttl*
: enable neighborhood enhancement and set local neighborhood TTL (default: off)

-S *seed*
: set permutation random seed (default: timestamp)

-p *dst_port*
: use specified transport destination port (default: 80)

-s
: send probes sequentially (default: random)

-b *bgp_rib*
: read BGP RIB (Potaroo text format) (default: none)

The IPv6-specific options are as follows:

-I *interface*
network interface to use, required

-t *tr_type*
set probe type: ICMP6, UDP6, TCP6_SYN, TCP6_ACK, required

-M *dst_mac*
MAC address of gateway router, required only if auto discovery fails

-G *src_mac*
MAC address of source, required only if auto discovery fails

OUTPUT

yarrp writes probe responses to the specified output file in a delimited ASCII format as they are received, one response per line. Because yarrp randomizes its probing, results will be similarly randomized. To determine all of the responses for a single target destination, it is necessary to filter and collate responses. The included yrp2warts.py utility performs this reconstitution and produces output in the standard warts binary format.

EXAMPLES

The command:

yarrp -i targets -o test.yrp -r 100 -Z

will send TCP_ACK topology probes in a randomly-permuted order to the IPv4 targets in file "targets" at a rate of 100pps, and write results to file "test.yrp".

The command:

yarrp -o scan.yrp -t ICMP -v -m 16 -Z 205.155.0.0/16

will send ICMP topology probes in a randomly-permuted order to all destinations within the prefix 205.155.0.0/16, from TTL 1 to 16 at the default rate of 10pps. Verbosity is switched on so that yarrp will report probe and response data to stdout. The results will be written to the file "scan.yrp".

The command:

yarrp -o scan2.yrp -t ICMP -Z -b bgptable.txt 1.0.0.0/8

will send ICMP topology probes in a randomly-permuted order to all destinations within the prefix 1.0.0.0/8, if the destination has a route in the BGP routing table "bgptable.txt". The routing table file must be plain-text in Potaroo format (the most recent table is available from https://bgp.potaroo.net/as6447/bgptable.txt). The results will be written to the file "scan2.yrp".

The command:

yarrp -t UDP6 -I eth0 -i targets6 -o test6.yrp -Z

will send UDP probes in a randomly-permuted order to the set of IPv6 targets in the file "targets6", and write the results to the file "test6.yrp".

SEE ALSO

`yrp2warts.py`(1)

R. Beverly, *Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery*, Proc. ACM/SIGCOMM Internet Measurement Conference 2016.

E. Gaston, *High-frequency mapping of the IPv6 Internet using Yarrp*, NPS Master's Thesis (http://hdl.handle.net/10945/52982), 2017.

AUTHORS

yarrp is written by Robert Beverly <rbeverly@cmand.org>. Ionut Luculescu contributed support for IPv4 UDP probing. Eric Gaston contributed support for IPv6 probing.