Package: pingers (via r-universe)

October 24, 2024

Type Package

Title Identify, Ping, and Log Internet Provider Connection Data

Description To assist you with troubleshooting internet connection issues and assist in isolating packet loss on your network. It does this by allowing you to retrieve the top trace route destinations your internet provider uses, and recursively ping each server in series while capturing the results and writing them to a log file. Each iteration it queries the destinations again, before shuffling the sequence of destinations to ensure the analysis is unbiased and consistent across each trace route.

Version 0.1.1

Date 2018-10-17

Maintainer Jesse Vent <cryptopackage@icloud.com>

URL https://github.com/JesseVent/pingers

BugReports https://github.com/JesseVent/pingers/issues

Depends R (>= 3.4.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, stringr, tibble, tictoc, tidyselect, data.table, lubridate, plotly, reshape2

RoxygenNote 6.1.0

Repository https://jessevent.r-universe.dev

RemoteUrl https://github.com/jessevent/pingers

RemoteRef HEAD

RemoteSha 1fc9ddaa201de08c1187fa68ee3a3b2f0de04c70

Contents

`capture_logs` *Capture ISP network logs*

Description

Repeat capturing network logs with parameters you specify from `ping_capture` and `get_destinations`. This will output a csv file with your ping results displaying packet loss and average ping across the defined periods.

Usage

```
capture_logs(destinations = 9, pings = 50, log_path = NULL,
  sleep = NULL)
```

Arguments

`destinations`	Retrieve the first n addresses in your ISP destinations
`pings`	Number of times to ping server
`log_path`	Optional: The path and filename to save the result set
`sleep`	Optional: Seconds to sleep for throughout iterations

Value

csv file with captured network log information

Note

If the log_path parameter is not provided, it will default to saving a csv file in the current working directory called network_logs.csv prefixed with the current timestamp in the format '

Examples

```
## Not run:
capture_logs(destinations = 3, pings = 10, log_path = log, sleep = 20)

## End(Not run)
```

get_destinations *Get ISP destinations*

Description

Trace route and grab the top n servers to assist isolating issues with individual nodes for your ISP.

Usage

```
get_destinations(keyword = NULL, top_n = NULL,
  site = "google.com.au")
```

Arguments

keyword	Keyword to search for i.e. 'AAT'
top_n	Retrieve the first n addresses
site	Defaults to 'google.com.au' to trace route against

Value

dataframe with server and IP range

Examples

```
## Not run:
dest <- get_destinations(top_n = 3)
print(dest)

## End(Not run)
```

pingers_heatmap *Packet Loss Heatmap*

Description

Generates a heatmap that displays the packet loss hotspots on an hourly basis during the week.

Usage

```
pingers_heatmap(logs = NULL)
```

Arguments

logs	network_logs file

Value

highcharts heatmap

Examples

```
## Not run:
pingers_heatmap(net_logs)

## End(Not run)
```

`ping_capture` *Ping Server*

Description

Ping a server to capture response details

Usage

```
ping_capture(server, count)
```

Arguments

`server`	IP address or URL of server
`count`	Number of times to ping server

Value

dataframe with ping results

Examples

```
## Not run:
dest     <- get_destinations(top_n = 1)
ping_res <- ping_capture(dest$ip[1], 10)
print(ping_res)

## End(Not run)
```

shuffle *Shuffle dataframe rows randomely*

Description

Randomly reorder the rows of a dataframe

Usage

```
shuffle(data)
```

Arguments

data	dataframe to shuffle

Value

reordered dataframe

Examples

```
{
ordered_df <- tibble::tibble(V1=1:26,V2=letters)
shuffled_df <- shuffle(ordered_df)
}
```

Index