Package 'rIP'

May 29, 2019

Type Package

Title Detects Fraud in Online Surveys by Tracing, Scoring, and Visualizing IP Addresses

Version 1.2.0

Maintainer Ryan Kennedy <rkennedy@uh.edu>

Description Takes an array of IPs and the keys for the services the user wishes to use (IP Hub, IP Intel, and Proxycheck),
and passes these to all respective APIs. Returns a dataframe with the IP addresses (used for merging),
country, ISP, labels for non-
US IP Addresses, VPS use, and recommendations for blocking. The package
also provides optional visualization tools for checking the distributions. Additional
functions are provided to call each discrete API endpoint. The package and methods are detailed in the recent paper
Waggoner, Kennedy, and Clifford (2019) <doi:10.21105/joss.01285>.

Imports httr, utils, iptools, dplyr, graphics, amerika, jsonlite

URL http://joss.theoj.org/papers/10.21105/joss.01285

BugReports https://github.com/MAHDLab/rIP/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Suggests testthat (>= 2.1.0)

Author Ryan Kennedy [aut, cre],
Philip Waggoner [aut] (<https://orcid.org/0000-0002-7825-7573>),
Scott Clifford [ctb],
Bob Rudis [ctb] (<https://orcid.org/0000-0001-5670-2640>)

Repository CRAN

Date/Publication 2019-05-29 17:10:02 UTC

R topics documented:

`getIPinfo` *Detects Fraud in Online Surveys by Tracing, Scoring, and Visualizing IP Addresses*

Description

Cleans and processes an array of IP address data and, using keys for several IP check services, passes these data to the needed APIs. Returns visual and numerical information on the IP address, including the internet service provider (ISP) and whether it is likely a server farm being used to disguise a respondent's location.

Usage

```
getIPinfo(d, "i", "iphub_key", "ipintel_key", "proxycheck_key", plots = TRUE)
```

Arguments

`d`	Data frame where IP addresses are stored
`i`	Name of the vector in data frame, d, corresponding to IP addresses in quotation marks
`iphub_key`	User's IP Hub X-key in quotation marks
`ipintel_key`	User's email address (used as key for getipintel.net) in quotation marks
`proxycheck_key`	User's API key for proxycheck.io in quotation marks
`plots`	Logical argument. If TRUE, produces a barplot of potentially suspicious IP addresses. Default is TRUE.

Details

Takes an array of IPs and the keys for the services the user wishes to use (IP Hub, IP Intel, and Proxycheck), and passes these to all respective APIs. Returns a dataframe with the IP addresses (used for merging), country, ISP, labels for non-US IP Addresses, VPS use, and recommendations for blocking. The function also provides optional visualization tools for checking the distributions.

Value

ipDF A dataframe with the IP address, country code, country name, asn, isp, block, and hostname.

Note

Users must have active accounts and/or valid keys at iphub, ipintel, and/or proxycheck.

Examples

```
## Not run:
ip_hub_key <- "MzI2MTpkOVpld3pZTVg1VmdTV3ZPenpzMmhopIMkRMZQ=="
ipintel_key <- "useremailaddress"
proxycheck_key <- "MzI2MTpkOVpld3pZTVg1VmdTV3ZPenpzMmhod"
ipsample <- data.frame(rbind(c(1, "189.8.105.146"), c(2, "148.233.134.248")))
names(ipsample) <- c("number", "IPAddress")
getIPinfo(ipsample, "IPAddress", ip_hub_key, ipintel_key, proxycheck_key)

## End(Not run)
```

`getipintel` *Retrieve IP address metadata from GetIPIntel*

Description

Retrieve IP address metadata from GetIPIntel

Usage

```
getipintel(ip, flags = NULL, oflags = NULL,
  contact_info = getipintel_contact_info())
```

Arguments

`ip`	an IP address (length 1 character vector)
`flags, oflags`	a valid GetIPIntel flag specification (See: https://getipintel.net/free-proxy-vpn-tor-detectio #optional_settings)
`contact_info`	GetIPIntel requires supplying contact info with each API call. Presently, this takes the form of an email address. See `getipintel_contact_info()` for more information.

Author(s)

Bob Rudis (bob@rud.is)

References

https://getipintel.net/

Examples

```
## Not run:
getipintel("24.63.157.68")

## End(Not run)
```

`getipintel_contact_info` *Get or set GETIPINTEL_CONTACT_INFO value*

Description

The API wrapper functions in this package all rely on a GetIPInfo API contact info string residing in the environment variable `GETIPINTEL_CONTACT_INFO`. The easiest way to accomplish this is to set it in the `.Renviron` file in your home directory.

Usage

```
getipintel_contact_info(force = FALSE)
```

Arguments

`force`	Force setting a new GetIPIntel contact info string for the current environment?

Value

atomic character vector containing the PGetIPIntel contact info string

Author(s)

Bob Rudis (bob@rud.is)

References

https://getipintel.net/

iphub_api_key *Get or set IPHUB_API_KEY value*

Description

The API wrapper functions in this package all rely on a PacketTotal API key residing in the environment variable `IPHUB_API_KEY`. The easiest way to accomplish this is to set it in the `.Renviron` file in your home directory.

Usage

```
iphub_api_key(force = FALSE)
```

Arguments

`force`	Force setting a new IPHub key for the current environment?

Value

atomic character vector containing the IPHub api key

Author(s)

Bob Rudis (bob@rud.is)

References

https://iphub.info/api

iphub_check *Retrieve IP address metadata from IPHub*

Description

Retrieve IP address metadata from IPHub

Usage

```
iphub_check(ip, api_key = iphub_api_key())
```

Arguments

`ip`	an IP address (length 1 character vector)
`api_key`	an IPHub API key (see `iphub_api_key()`)

Author(s)

Bob Rudis (bob@rud.is)

References

https://iphub.info/api

Examples

```
## Not run:
iphub_check("24.63.157.68")

## End(Not run)
```

`proxycheck` *Retrieve IP address metadata from ProxyCheck*

Description

Pass in an IP address along with API key and any extra API query flags and retrieve metadata about the IP from ProxyCheck.

Usage

```
proxycheck(ip, ..., api_key = proxycheck_api_key())
```

Arguments

`ip`	an IP address (length 1 character vector)
`...`	ProxyCheck API query flage (see Details)
`api_key`	a ProxyCheck API key (see `proxycheck_api_key()`)

Details

You can specify values for any additional query flags via `...`. The package will be updated as the supported flags change. Current supported query flags are:

- `vpn`: (logical) VPN check on the IP Address and present the result to you.
- `asn`: (logical) ASN check on the IP Address and present you with the provider name, ASN, country, country isocode, city (if it's in a city) and lang/long for the IP Address.
- `node`: (logical) Will return node within our cluster answered your API call. This is only really needed for diagnosing problems with our support staff.
- `time`: (logical) Will return how long this query took to be answered by our API excluding network overhead.
- `inf`: (logical) When `FALSE` the query will *not* use the real-time inference engine. In the absence of this flag or if it's set to `TRUE` we will run the query through our real-time inference engine.

- `risk`: (logical) When `TRUE`, will return the risk score for this IP Address ranging from 0 to 100. Scores below 33 can be considered low risk while scores between 34 and 66 can be considered high risk. Addresses with scores above 66 are considered very dangerous.
- `port`: (logical) Will return the port number we saw this IP Address operating a proxy server on.
- `seen`: (logical) Will return the most recent time we saw this IP Address operating as a proxy server.
- `days`: (integer) Will restrict proxy results to between now and the amount of days specified. For example if you supplied `days=2` we would only check our database for Proxies that we saw within the past 48 hours. By default without this flag supplied we search within the past 7 days.
- `tag`: (string) The query result will be tagged with the message you supply.

Author(s)

Bob Rudis (bob@rud.is)

References

https://proxycheck.io/api/#introduction

Examples

```
## Not run:
proxycheck("24.63.157.68", asn = TRUE, risk = TRUE)

## End(Not run)
```

`proxycheck_api_key` *Get or set PROXYCHECK_API_KEY value*

Description

The API wrapper functions in this package all rely on a PacketTotal API key residing in the environment variable `PROXYCHECK_API_KEY`. The easiest way to accomplish this is to set it in the `.Renviron` file in your home directory.

Usage

```
proxycheck_api_key(force = FALSE)
```

Arguments

`force`	Force setting a new ProxyCheck key for the current environment?

Value

atomic character vector containing the ProxyCheck api key

Author(s)

Bob Rudis (bob@rud.is)

References

https://proxycheck.io/api/#introduction

Index