

Package ‘gdns’

September 17, 2018

Title Tools to Work with Google's 'DNS-over-HTTPS' ('DoH') 'API'

Version 0.3.1

Maintainer Bob Rudis <bob@rud.is>

Description To address the problem of insecurity of 'UDP'-based 'DNS' requests, 'Google Public DNS' offers 'DNS' resolution over an encrypted 'HTTPS' connection. 'DNS-over-HTTPS' greatly enhances privacy and security between a client and a recursive resolver, and complements 'DNSSEC' to provide end-to-end authenticated 'DNS' lookups. Functions that enable querying individual requests that bulk requests that return detailed responses and bulk requests are both provided. Support for reverse lookups is also provided. See <<https://developers.google.com/speed/public-dns/docs/dns-over-https>> for more information.

License AGPL

LazyData true

Encoding UTF-8

Depends R (>= 3.0.0)

Suggests testthat

Imports httr, stats, jsonlite, stringi

RoxygenNote 6.0.1.9000

NeedsCompilation no

Author Bob Rudis [aut, cre] (<<https://orcid.org/0000-0001-5670-2640>>)

Repository CRAN

Date/Publication 2018-09-17 14:10:06 UTC

R topics documented:

bulk_query	2
gdns	3
has_spf	3
is_soft_fail	4

query	4
resource_record_tbl	5
spf_ipv4s	6
split_spf	7

Index	8
--------------	----------

bulk_query	<i>Vectorized query, returning only answers in a data frame</i>
------------	---

Description

Vectorized query, returning only answers in a data frame

Usage

```
bulk_query(entities, type = 1, edns_client_subnet = "0.0.0.0/0")
```

Arguments

entities	character vector of entities to query
type	RR type can be represented as a number in [1, 65535] or canonical string (A, aaaa, etc). More information on RR types can be found here .
edns_client_subnet	The edns0-client-subnet option. Format is an IP address with a subnet mask. Examples: 1.2.3.4/24, 2001:700:300::/48. If you are using DNS-over-HTTPS because of privacy concerns, and do not want any part of your IP address to be sent to authoritative nameservers for geographic location accuracy, use edns_client_subnet=0.0.0.0/0. Google Public DNS normally sends approximate network information (usually replacing the last part of your IPv4 address with zeroes). 0.0.0.0/0 is the default.

Value

data.frame of only answers (use query() for detailed responses)

Note

this is a fairly naive function. It expects Answer to be one of the return value list slots. The intent for it was to make it easier to do bulk forward queries. It will get smarter in future versions.

References

<https://developers.google.com/speed/public-dns/docs/dns-over-https>

Examples

```
hosts <- c("rud.is", "r-project.org", "rstudio.com", "apple.com")
gdns::bulk_query(hosts)
```

`gdns`*Tools to Work with Google DNS Over HTTPS API*

Description

Traditional DNS queries and responses are sent over UDP or TCP without encryption. This is vulnerable to eavesdropping and spoofing (including DNS-based Internet filtering). Responses from recursive resolvers to clients are the most vulnerable to undesired or malicious changes, while communications between recursive resolvers and authoritative nameservers often incorporate additional protection.

To address this problem, Google Public DNS offers DNS resolution over an encrypted HTTPS connection. DNS-over-HTTPS greatly enhances privacy and security between a client and a recursive resolver, and complements DNSSEC to provide end-to-end authenticated DNS lookups.

Support for reverse lookups is also provided.

See <https://developers.google.com/speed/public-dns/docs/dns-over-https> for more information.

Author(s)

Bob Rudis (bob@rud.is)

`has_spf`*Test for whether a DNS TXT record is an SPF record*

Description

Test for whether a DNS TXT record is an SPF record

Usage

```
has_spf(sp*_rec)
```

Arguments

`spf_rec` a character vector of DNS TXT records

is_soft_fail	<i>SPF "all" type test</i>
--------------	----------------------------

Description

SPF "all" type test

Usage

```
is_soft_fail(spf_rec)
```

```
is_hard_fail(spf_rec)
```

```
passes_all(spf_rec)
```

Arguments

spf_rec	a character vector of DNS TXT records
---------	---------------------------------------

query	<i>Perform DNS over HTTPS queries using Google</i>
-------	--

Description

Traditional DNS queries and responses are sent over UDP or TCP without encryption. This is vulnerable to eavesdropping and spoofing (including DNS-based Internet filtering). Responses from recursive resolvers to clients are the most vulnerable to undesired or malicious changes, while communications between recursive resolvers and authoritative nameservers often incorporate additional protection.

To address this problem, Google Public DNS offers DNS resolution over an encrypted HTTPS connection. DNS-over-HTTPS greatly enhances privacy and security between a client and a recursive resolver, and complements DNSSEC to provide end-to-end authenticated DNS lookups.

Usage

```
query(name, type = "1", edns_client_subnet = "0.0.0.0/0")
```

Arguments

name	item to lookup. Valid characters are numbers, letters, hyphen, and dot. Length must be between 1 and 255. Names with escaped or non-ASCII characters are not supported. Internationalized domain names must use the punycode format (e.g. "xn--qxam").
------	--

If an IPv4 string is input, it will be transformed into a proper format for reverse lookups.

type	RR type can be represented as a number in [1, 65535] or canonical string (A, aaaa, etc). More information on RR types can be found here . You can use 255 for an ANY query.
edns_client_subnet	The edns0-client-subnet option. Format is an IP address with a subnet mask. Examples: 1.2.3.4/24, 2001:700:300::/48. If you are using DNS-over-HTTPS because of privacy concerns, and do not want any part of your IP address to be sent to authoritative nameservers for geographic location accuracy, use edns_client_subnet=0.0.0.0/0. Google Public DNS normally sends approximate network information (usually replacing the last part of your IPv4 address with zeroes). 0.0.0.0/0 is the default.

Details

To perform vectorized queries with only answers (and no metadata) use `bulk_query()`.

Value

a list with the query result or NULL if an error occurred

References

<https://developers.google.com/speed/public-dns/docs/dns-over-https>

Examples

```
query("rud.is")
query("example.com", "255") # ANY query
query("microsoft.com", "MX")
query("google-public-dns-a.google.com", "TXT")
query("apple.com")
query("17.142.160.59", "PTR")
```

resource_record_tbl	<i>An overview of resource records (RRs) permissible in zone files of the Domain Name System (DNS)</i>
---------------------	--

Description

A dataset containing the DNS resource record types, names, description and purpose

Usage

resource_record_tbl

Format

A data frame with 39 rows and 4 variables:

type numeric type of the resource record

name short name of the resource record

description short description of the resource record

purpose long-form description of the resource record purpose/function/usage

Source

https://en.wikipedia.org/wiki/List_of_DNS_record_types

spf_ipv4s	<i>SPF field extraction functions</i>
-----------	---------------------------------------

Description

Various helper functions to extract SPF record components.

Usage

spf_ipv4s(spf_rec)

spf_ipv6s(spf_rec)

spf_includes(spf_rec)

spf_ptrs(spf_rec)

spf_exists(spf_rec)

Arguments

spf_rec a character vector of DNS TXT records

`split_spf`*Split out all SPF records in a domain's TXT record*

Description

Given a vector of TXT records, this function will return a list of vectors of all the SPF records for each. If the given TXT record is not an SPF record, NA is returned (which makes it easy to skip with `purrr` functions).

Usage

```
split_spf(sp_f_rec)
```

Arguments

`spf_rec` a character vector of DNS TXT records

Index

*Topic **datasets**

resource_record_tbl, 5

bulk_query, 2

gdns, 3

gdns-package (gdns), 3

has_spf, 3

is_hard_fail (is_soft_fail), 4

is_soft_fail, 4

passes_all (is_soft_fail), 4

query, 4

resource_record_tbl, 5

spf_exists (spf_ipv4s), 6

spf_includes (spf_ipv4s), 6

spf_ipv4s, 6

spf_ipv6s (spf_ipv4s), 6

spf_ptrs (spf_ipv4s), 6

split_spf, 7