certbot-dns-rfc2136 Documentation

Release 0

Certbot Project

Jun 10, 2019

Contents:

The *dns_rfc2136* plugin automates the process of completing a dns-01 challenge (DNS01) by creating, and subsequently removing, TXT records using RFC 2136 Dynamic Updates.

CHAPTER 1

Named Arguments

`--dns-rfc2136-credentials`	RFC 2136 *credentials* INI file. (Required)
`--dns-rfc2136-propagation-seconds`	The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (Default: 60)

CHAPTER 2

Credentials

Use of this plugin requires a configuration file containing the target DNS server and optional port that supports RFC 2136 Dynamic Updates, the name of the TSIG key, the TSIG key secret itself and the algorithm used if it's different to HMAC-MD5.

Listing 1: Example credentials file:

```
# Target DNS server
dns_rfc2136_server = 192.0.2.1
# Target DNS port
dns_rfc2136_port = 53
# TSIG key name
dns_rfc2136_name = keyname.
# TSIG key secret
dns_rfc2136_secret = 4q4wM/2I180UXoMyN4INVhJNi8V9BCV+jMw2mXgZw/CSuxUT8C7NKKFs␣
↪AmKd7ak51vWKgSl12ib86oQRPkpDjg==
# TSIG key algorithm
dns_rfc2136_algorithm = HMAC-SHA512
```

The path to this file can be provided interactively or using the `--dns-rfc2136-credentials` command-line argument. Certbot records the path to this file for use during renewal, but does not store the file's contents.

Caution: You should protect this TSIG key material as it can be used to potentially add, update, or delete any record in the target DNS server. Users who can read this file can use these credentials to issue arbitrary API calls on your behalf. Users who can cause Certbot to run using these credentials can complete a `dns-01` challenge to acquire new certificates or revoke existing certificates for associated domains, even if those domains aren't being managed by this server.

Certbot will emit a warning if it detects that the credentials file can be accessed by other users on your system. The warning reads "Unsafe permissions on credentials configuration file", followed by the path to the credentials file. This warning will be emitted each time Certbot uses the credentials file, including for renewal, and cannot be silenced except by addressing the issue (e.g., by using a command like `chmod 600` to restrict access to the file).

2.1 Sample BIND configuration

Here's a sample BIND configuration for Certbot to use. You will need to generate a new TSIG key, include it in the BIND configuration and grant it permission to issue updates on the target DNS zone.

Listing 2: Generate a new SHA512 TSIG key

```
dnssec-keygen -a HMAC-SHA512 -b 512 -n HOST keyname.
```

Note: There are a few tools shipped with BIND that can all generate TSIG keys; `dnssec-keygen`, `rndc-confgen`, and `ddns-confgen`. Try and use the most secure algorithm supported by your DNS server.

Listing 3: Sample BIND configuration

```
key "keyname." {
  algorithm hmac-sha512;
  secret "4q4wM/2I180UXoMyN4INVhJNi8V9BCV+jMw2mXgZw/CSuxUT8C7NKKFs␣
↪AmKd7ak51vWKgSl12ib86oQRPkpDjg==";
};

zone "example.com." IN {
  type master;
  file "named.example.com";
  update-policy {
    grant keyname. name _acme-challenge.example.com. txt;
  };
};
```

Note: This configuration limits the scope of the TSIG key to just be able to add and remove TXT records for one specific host for the purpose of completing the `dns-01` challenge. If your version of BIND doesn't support the update-policy directive then you can use the less-secure allow-update directive instead.

CHAPTER 3

Examples

Listing 1: To acquire a certificate for `example.com`

```
certbot certonly \
  --dns-rfc2136 \
  --dns-rfc2136-credentials ~/.secrets/certbot/rfc2136.ini \
  -d example.com
```

Listing 2: To acquire a single certificate for both `example.com` and `www.example.com`

```
certbot certonly \
  --dns-rfc2136 \
  --dns-rfc2136-credentials ~/.secrets/certbot/rfc2136.ini \
  -d example.com \
  -d www.example.com
```

Listing 3: To acquire a certificate for `example.com`, waiting 30 seconds for DNS propagation

```
certbot certonly \
  --dns-rfc2136 \
  --dns-rfc2136-credentials ~/.secrets/certbot/rfc2136.ini \
  --dns-rfc2136-propagation-seconds 30 \
  -d example.com
```

CHAPTER 4

API Documentation

4.1 certbot_dns_rfc2136.dns_rfc2136

DNS Authenticator using RFC 2136 Dynamic Updates.

class certbot_dns_rfc2136.dns_rfc2136.**Authenticator**(**args*, ***kwargs*)

Bases: certbot.plugins.dns_common.DNSAuthenticator

DNS Authenticator using RFC 2136 Dynamic Updates

This Authenticator uses RFC 2136 Dynamic Updates to fulfull a dns-01 challenge.

classmethod add_parser_arguments(*add*)

Add plugin arguments to the CLI argument parser.

NOTE: If some of your flags interact with others, you can use cli.report_config_interaction to register this to ensure values are correctly saved/overridable during renewal.

Parameters **add** (*callable*) – Function that proxies calls to argparse.ArgumentParser.add_argument prepending options with unique plugin name prefix.

_setup_credentials()

Establish credentials, prompting if necessary.

_perform(*_domain*, *validation_name*, *validation*)

Performs a dns-01 challenge by creating a DNS TXT record.

Parameters

- **domain** (*str*) – The domain being validated.
- **validation_domain_name** (*str*) – The validation record domain name.
- **validation** (*str*) – The validation record content.

Raises **errors.PluginError** – If the challenge cannot be performed

_cleanup(*_domain*, *validation_name*, *validation*)
Deletes the DNS TXT record which would have been created by `_perform_achall`.

Fails gracefully if no such record exists.

Parameters

- **domain** (*str*) – The domain being validated.
- **validation_domain_name** (*str*) – The validation record domain name.
- **validation** (*str*) – The validation record content.

class certbot_dns_rfc2136.dns_rfc2136.**_RFC2136Client**(*server*, *port*, *key_name*, *key_secret*, *key_algorithm*)
Bases: object

Encapsulates all communication with the target DNS server.

add_txt_record(*record_name*, *record_content*, *record_ttl*)
Add a TXT record using the supplied information.

Parameters

- **record_name** (*str*) – The record name (typically beginning with '_acme-challenge.').
- **record_content** (*str*) – The record content (typically the challenge validation).
- **record_ttl** (*int*) – The record TTL (number of seconds that the record may be cached).

Raises **certbot.errors.PluginError** – if an error occurs communicating with the DNS server

del_txt_record(*record_name*, *record_content*)
Delete a TXT record using the supplied information.

Parameters

- **record_name** (*str*) – The record name (typically beginning with '_acme-challenge.').
- **record_content** (*str*) – The record content (typically the challenge validation).
- **record_ttl** (*int*) – The record TTL (number of seconds that the record may be cached).

Raises **certbot.errors.PluginError** – if an error occurs communicating with the DNS server

_find_domain(*record_name*)
Find the closest domain with an SOA record for a given domain name.

Parameters **record_name** (*str*) – The record name for which to find the closest SOA record.

Returns The domain, if found.

Return type str

Raises **certbot.errors.PluginError** – if no SOA record can be found.

_query_soa(*domain_name*)
Query a domain name for an authoritative SOA record.

Parameters **domain_name** (*str*) – The domain name to query for an SOA record.

Returns True if found, False otherwise.

Return type bool

Raises `certbot.errors.PluginError` – if no response is received.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index

Symbols

A

C

D