certbot-dns-rfc2136 Documentation

Release 0

Certbot Project

Mar 16, 2022
## CONTENTS:

1 Named Arguments 3

2 Credentials 5
   2.1 Sample BIND configuration 6

3 Examples 7

4 API Documentation 9

5 Indices and tables 11

Python Module Index 13

Index 15
The `dns_rfc2136` plugin automates the process of completing a `dns-01` challenge (DNS-01) by creating, and subsequently removing, TXT records using RFC 2136 Dynamic Updates.

**Note:** The plugin is not installed by default. It can be installed by heading to [certbot.eff.org](http://certbot.eff.org), choosing your system and selecting the Wildcard tab.
### NAMED ARGUMENTS

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--dns-rfc2136-credentials</code></td>
<td>RFC 2136 credentials INI file. (Required)</td>
</tr>
<tr>
<td><code>--dns-rfc2136-propagation-seconds</code></td>
<td>Number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (Default: 60)</td>
</tr>
</tbody>
</table>
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:

```plaintext
# Target DNS server (IPv4 or IPv6 address, not a hostname)
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/CSuxUT8C7NKKFst/
˓→AmKd7ak5lVWkgS112ib86oQRPkpDjg==
# 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 currentDateAmKd7ak51vWKgSl12ib86oQRPkpDjg==";
};

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. See the BIND documentation for details.
CHAPTER THREE

EXAMPLES

Listing 1: To acquire a certificate for example.com

```bash
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

```bash
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

```bash
certbot certonly \ 
--dns-rfc2136 \ 
--dns-rfc2136-credentials ~/.secrets/certbot/rfc2136.ini \ 
--dns-rfc2136-propagation-seconds 30 \ 
-d example.com
```
Certbot plugins implement the Certbot plugins API, and do not otherwise have an external API.
• genindex
• modindex
• search
C

certbot_dns_rfc2136, 1
INDEX

C
certbot_dns_rfc2136
   module, 1

M
module
certbot_dns_rfc2136, 1