UACME(1) Manual Page

NAME

uacme - ACMEv2 client written in plain C code with minimal dependencies

SYNOPSIS

uacme [**-a**|**--acme-url** *URL*] [**-b**|**--bits** *BITS*] [**-c**|**--confdir** *DIR*] [**-d**|**--days** *DAYS*] [**-f**|**--force**] [**-h**|**--hook** *PROGRAM*] [**-m**|**--must-staple**] [**-n**|**--never**] [**-s**|**--staging**] [**-t**|**--type** **RSA**|**EC**] [**-v**|**--verbose** …] [**-V**|**--version**] [**-y**|**--yes**] [**-?**|**--help**] **new** [*EMAIL*] | **update** [*EMAIL*] | **deactivate** | **newkey** | **issue** *DOMAIN* [*ALTNAME* …]] | **revoke** *CERTFILE*

DESCRIPTION

uacme is a client for the ACMEv2 protocol described in RFC8555, written in plain C code with minimal dependencies (libcurl and one of GnuTLS, OpenSSL or mbedTLS). The ACMEv2 protocol allows a Certificate Authority (https://letsencrypt.org is a popular one) and an applicant to automate the process of verification and certificate issuance. The protocol also provides facilities for other certificate management functions, such as certificate revocation. For more information see https://tools.ietf.org/html/rfc8555

OPTIONS

-a, --acme-url=*URL*

ACMEv2 server directory object *URL*. If not specified **uacme** uses one of the following:

https://acme-v02.api.letsencrypt.org/directory

production URL

https://acme-staging-v02.api.letsencrypt.org/directory

staging URL (see **-s, --staging** below)

-b, --bits=*BITS*

key bit length (default 2048 for RSA, 256 for EC). Only applies to newly generated keys. RSA key length must be a multiple of 8 between 2048 and 8192. EC key length must be either 256 (**NID_X9_62_prime256v1** curve) or 384 (**NID_secp384r1** curve).

-c, --confdir=*CONFDIR*

Use configuration directory *CONFDIR* (default */etc/ssl/uacme*). The structure is as follows (multiple *DOMAINs* allowed)

- *CONFDIR/private/key.pem*

  ACME account private key

- *CONFDIR/private/DOMAIN/key.pem*

  certificate key for *DOMAIN*

- *CONFDIR/DOMAIN/cert.pem*

  certificate for *DOMAIN*

-d, --days=*DAYS*

Do not reissue certificates that are still valid for longer than *DAYS* (default 30).

-f, --force

Force certificate reissuance regardless of expiration date.

-h, --hook=*PROGRAM*

Challenge hook program. If not specified **uacme** interacts with the user for every ACME challenge, printing information about the challenge type, token and authorization on stderr. If specified, **uacme** executes *PROGRAM* (a binary, a shell script or any file that can be executed by the operating system) for every challenge with the following 5 string arguments:

- *METHOD*

  one of **begin**, **done** or **failed**.

  - **begin**

    is called at the beginning of the challenge. *PROGRAM* must return 0 to accept it. Any other return code declines the challenge. Neither **done** nor **failed** method calls are made for declined challenges.

  - **done**

    is called upon successful completion of an accepted challenge.

  - **failed**

    is called upon failure of an accepted challenge.

- *TYPE*

  challenge type (for example **dns-01** or **http-01**)

- *IDENT*

  The identifier the challenge refers to

- *TOKEN*

  The challenge token

- *AUTH*

  The key authorization (for **dns-01** and **tls-alpn-01** already converted to the base64-encoded SHA256 digest format)

-m, --must-staple

Request certificates with the RFC7633 Certificate Status Request TLS

Feature Extension, informally also known as "OCSP Must-Staple".

-n, --never-create
By default **uacme** creates directories/keys if they do not exist. When this option is specified, **uacme** never does so and instead exits with an error if anything required is missing.

-s, --staging
Use Let’s Encrypt staging URL for testing. This only works if **-a, --acme-url** is **NOT** specified.

-t, --type=RSA | EC
Key type, either RSA or EC. Only applies to newly generated keys. The bit length can be specified with **-b, --bits**.

-v, --verbose
By default **uacme** only produces output upon errors or when user interaction is required. When this option is specified **uacme** prints information about what is going on on stderr. This option can be specified more than once to increase verbosity.

-V, --version
Print program version on stderr and exit.

-y, --yes
Autoaccept ACME server terms (if any) upon new account creation.

-?, --help
Print a brief usage text on stderr and exit.

USAGE

uacme [*OPTIONS* ...] **new** [*EMAIL*]
Create a new ACME account with optional *EMAIL* contact. If the account private key does not exist at *CONFDIR/private/key.pem* a new key is generated unless **-n, --never-create** is specified. A valid account must be created **before** any other operation can succeed. Any certificate issued by the ACME server is associated with a single account. An account can be associated with multiple certificates, subject of course to the rate limits imposed by the ACME server.

uacme [*OPTIONS* ...] **update** [*EMAIL*]
Update the *EMAIL* associated with the ACME account corresponding to the account private key. If *EMAIL* is not specified, the account contact email will be dropped.

uacme [*OPTIONS* ...] **deactivate**
Deactivate the ACME account corresponding to the account private key. **WARNING** this action is irreversible. Users may wish to do this when the account key is compromised or decommissioned. A deactivated account can no longer request certificate issuances and revocations or access resources related to the account.

uacme [*OPTIONS* ...] **newkey**

Change the ACME account private key. If the new account private key does not exist at *CONFDIR/private/newkey.pem* it is generated unless **-n, --never-create** is specified. The new key is then submitted to the server and if the operation succeeds the old key is hardlinked to *CONFDIR/private/key-TIMESTAMP.pem* before renaming *CONFDIR/private/newkey.pem* to *CONFDIR/private/key.pem*.

uacme [*OPTIONS* ...] **issue** *DOMAIN* [*ALTNAME* ...]

Issue a certificate for *DOMAIN* with zero or more *ALTNAMEs*. If a certificate is already available at *CONFDIR/DOMAIN/cert.pem* for the specified *DOMAIN* and *ALTNAMEs*, and is still valid for longer than *DAYS*, no action is taken unless **-f, --force** is specified. The new certificate is saved to *CONFDIR/DOMAIN/cert.pem*. If the certificate file already exists, it is hardlinked to *CONFDIR/DOMAIN/cert-TIMESTAMP.pem* before overwriting. The private key for the certificate is loaded from *CONFDIR/private/DOMAIN/key.pem*. If no such file exists, a new key is generated unless **-n, --never-create** is specified.

uacme [*OPTIONS* ...] **revoke** *CERTFILE*

Revoke the certificate stored in *CERTFILE*. Only certificates associated with the account can be revoked. If successful *CERTFILE* is renamed to *revoked-TIMESTAMP.pem*.

EXIT STATUS

0

Success

1

Certificate not reissued because it is still current

2

Failure (syntax or usage error; configuration error; processing failure; unexpected error).

EXAMPLE HOOK SCRIPT

The *uacme.sh* hook script included in the distribution can be used to automate the certificate issuance with *http-01* challenges, provided a web server for the domain being validated runs on the same machine, with webroot at /var/www

```
#!/bin/sh
CHALLENGE_PATH=/var/www/.well-known/acme-challenge
ARGS=5
E_BADARGS=85

if test $# -ne "$ARGS"
then
    echo "Usage: `basename $0` method type ident token auth"
1>&2
```

```
    exit $E_BADARGS
fi

METHOD=$1
TYPE=$2
IDENT=$3
TOKEN=$4
AUTH=$5

case "$METHOD" in
    "begin")
        case "$TYPE" in
            http-01)
                echo -n "${AUTH}" >
${CHALLENGE_PATH}/${TOKEN}
                exit $?
                ;;
            *)
                exit 1
                ;;
        esac
        ;;
    "done"|"failed")
        case "$TYPE" in
            http-01)
                rm ${CHALLENGE_PATH}/${TOKEN}
                exit $?
                ;;
            *)
                exit 1
                ;;
        esac
        exit 0
        ;;
    *)
        echo "$0: invalid method" 1>&2
        exit 1
esac
```

BUGS

If you believe you have found a bug, please create a new issue at https://github.com/ndilieto/uacme/issues with any applicable information.

AUTHOR

uacme was written by Nicola Di Lieto

COPYRIGHT

Version 1.0.20
Last updated 2019-10-03 20:44:42 CEST