Automating certificate management with ACME

The traditional way of managing SSL certificates

In many companies, even those with more up-to-date IT systems, SSL certificates are managed using "old-style", essentially manual, procedures. When there are many certificates to be managed, as is the case in large private companies and national or regional public bodies, manual management is very time-consuming (and therefore costly), as well as presenting other disadvantages, including increased risk of error and the inability to renew certificates quickly, when required. Ultimately, managing SSL certificates manually wastes resources, is inefficient and even presents severe security risks, as recent studies also demonstrate [1] [2].

Let's take into account that installing *each* SSL certificate (whether for the first time or a renewal) normally requires several steps, including:

- accessing the server;
- generating a key pair;
- generating the corresponding CSR;
- sending the CSR to the CA, possibly via the company PKI management team (if there is one);
- preparing the response to any challenges from the CA to validate the domains (*);
- responding to the authentication verification request from the CA (*);
- downloading and installing the certificate on the server;
- backing up the private key and associated certificate;
- tracking the operations performed.

(*) When the company is enabled to act as an "Enterprise RA", these steps can be completed once and then skipped for a period of time (e.g. 12 months).

Some of these operations require the correct execution of rather "exotic" and rarely used commands, so must be carried out by trained systems engineers; mistakes are nevertheless still quite common. In addition, these operations require different procedures depending on the web server software (e.g. Apache, Nginx, Microsoft IIS, etc.) or the device used (e.g. firewall, loadbalancer, etc.). What's more, performing these operations with the necessary attention to detail is very labour-intensive and time-consuming when there are large numbers of certificates to manage. The cost of manual management is also dependent on the fact that the different certificates used within the company often have different expiry dates.

Automating the process with ACME

Clearly, in order to overcome the disadvantages of managing certificates manually, some level of automation is required, in line with an increasingly "DevOps" approach to IT management.

Various CAs, including Actalis, offer customers the opportunity to request certificates by invoking proprietary APIs. This possibility already results in a significant automation of the process, albeit with some limitations associated with the specific features of the protocol. To offer greater interoperability and make integration easier, however, it makes more sense to adopt an "open" interface. A number of protocols have been designed and standardized to automate the management of certificates, but one notable example has been particularly successful in recent years, thanks to its great effectiveness with SSL certificates and the availability of numerous open source implementations: we're talking about the ACME protocol.

ACME (Automated Certificate Management Environment) is a standard protocol, described in the **RFC8555** [3] publication, which lets you request and manage SSL certificates much faster and more easily than with the traditional manual management performed by a system administrator. In fact, the ACME protocol means that servers can obtain certificates completely automatically. And there's more: in most cases you can even automate the installation and renewal of certificates.

The ACME protocol typically works as follows (simplified description):

1. *once only*, an appropriate software agent (an **"ACME client"**) is installed on the web server and configured as desired (specifying, for example, which CA URL to use, as well as other options);
2. the ACME client "registers" with the CA (creates a new "**account**"); at this stage, a key pair is created that will be used to sign any subsequent request to the CA;
3. the ACME client sends the CA a certificate "**order**", containing the list of domains to be validated and the preferred validation method;
4. the CA responds to the ACME client with one or more "**challenges**" (depending on whether there are one or more domains to be validated) of the requested type;
5. the ACME client **automatically publishes** the responses to the challenges via the appropriate method (e.g. on the web server or on the domain);
6. the CA queries the web server and/or domain to verify that the response to the challenges is as expected;
7. the ACME client **generates** a **key** pair for the web server and a corresponding **CSR**, which it sends on to the CA;
8. the CA **generates the certificate** and returns it to the ACME client, along with the certification chain;
9. (typically) the ACME client **installs the certificate** on the web server automatically, which means that it is immediately operational;
10. (typically) the ACME client prepares the **automatic renewal** of the certificate, scheduling a specific task for thi.

(The details of the different steps may vary depending on the ACME client being used.)

All of these processes are carried out automatically and in "real" time with the sole exception of the "setup" phase (installing and configuring the ACME client, as well as the first command executed). The ACME client is also scheduled for renewals, usually automatically after the first command is executed. This eliminates almost all the work normally involved in manually requesting and installing SSL certificates, which can mean considerable savings when a company has multiple certificates to manage.

What type of certificates can it be used for?

Using the ACME protocol, all the different types of certificate can be obtained: **single domain**, **multi-domain** ("SAN") and **wildcard**. The actual possibilities depend on the particular CA used as well as the validation method (i.e. the type of "challenge") that you decide to use.
Likewise, the ACME protocol itself does not limit the level (DV, OV, EV) of certificate: that said, to get OV/EV certificates, the CA must obviously be able to *authenticate* the ACME call, so that it can assign it to a specific, previously validated and authorized organization. The technical details of the authentication methods may vary depending on the particular CA used. The same requirement also applies to DV certificates for paid services.

How to use ACME in practice

First of all, you need to decide which ACME client to use. There is an extensive choice: there are in fact numerous ACME clients, usually open source, for every conceivable environment. The options available include not only ready-to-use client software, but also lots of libraries that let you develop *your own* ACME client with your preferred programming language (C/C++, C#, Java, Python, etc.).

Certbot is usually recommended for Linux environments: it's definitely the most powerful and popular client. Below is an example of a certificate request based on this client.
Certbot is also available for Windows environments, but to date this offers less functionality than the Linux version. Other possible options for Windows environments are **win-acme**, **AutoACME**, etc.
Below are instructions on how to **invoke certbot** to request an SSL certificate for the domain `example.com` by contacting your CA at <URL>, assuming the web server is based on apache:

```
certbot --apache --server <URL> -d example.com
```

Renewal essentially works in the same way, simply by specifying the **renew** command, but without having to send **certbot** the information provided when first issued, as this will have been stored:

```
certbot --apache renew
```

Normally however, you don't need to execute this command manually, because after installing the first **certbot** certificate, it schedules itself for renewal, automatically creating a "cron job".

Supporto ACME di Actalis

The Actalis CA supports the ACME protocol for Enterprise customers.

Using the “Enterprise RA” web application, customers can create an ACME end-point according to their own requirements, and use it through the ACME client of their choice.

Notably, Actalis supports:

1. RSA keys between 2048 and 4096 bits
2. ECDSA P256 and P384 keys
3. Multiple ACME URLs per customer
4. HTTP-01 Challenge type
5. DNS-01 Challenge type

N.B.: It is not possible to guarantee support for all current ACME customers. Actalis tests primarily with Certbot and Acme4J clients. A *testing environment* is also available, offering companies/bodies who are interested in ACME the chance to try it out.

Riferimenti

[1] https://info.keyfactor.com/the-impact-of-unsecured-digital-identities-ponemon-report
[2] NIST SPECIAL PUBLICATION 1800-16
[3] https://tools.ietf.org/rfc/rfc8555.txt