CSIT Conference 2019, Yerevan, Armenia, September 23-27

SSL Certificate Deployment Automation Concept for ASNET-AM Network Services

Arthur Petrosyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
e-mail: arthur@sci.am

Gurgen Petrosyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
e-mail: gurgen@sci.am

Robert Tadevosyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
e-mail: robert@sci.am

ABSTRACT

This paper describes the concept proposed for the Academic Scientific Research Computer Network of Armenia (ASNET-AM) regarding the use of Certificate Deployment Automation. Security of most network services currently depends on the use of digital certificates. Taking into account the availability of free Let's Encrypt wildcard SSL certificates, the concept of implementing centralized certificate server is described. The goal is to provide centralized secure and automated free digital certificates service for multiple servers in ASNET-AM. Presented concept can be used for different types of network services, such as webservers, mailservers, etc.

Keywords

Networking, SSL, Let's Encrypt, Wildcard, Security, Certificate, Automation

1. INTRODUCTION

The Academic Scientific Research Computer Network of Armenia (ASNET-AM) is the National Research and Education Network (NREN) of Armenia. Created in 1994 and having over 20 years of experience in Networking and Information Technologies, ASNET-AM [1] provides various networking solutions to the Academic, Scientific, Research, Educational, Cultural and other organizations of Armenia, which are engaged in scientific and educational activities.

During past years several solutions were used at ASNET-AM for securing the network services, authenticating network servers and establishing secure sessions with end clients by use of digital certificates.

Among them were paid SSL certificates, issued and signed by a commercial Certificate Authorities (CA), as well as the Trusted Certificate Service (TCS) solution [2] from GEANT [3].

2. LET'S ENCRYPT SOLUTION

In mid 2016 new Let's Encrypt solution [4] [5] came on the scene. Let's Encrypt is a non-profit certificate CA run by Internet Security Research Group (ISRG) that provides X.509 certificates for Transport Layer Security (TLS) encryption at no charge. It's a free, automated, and open solution for securing network services. Digital certificates, provided by Let's Encrypt are valid for 90 days, during which of certificate renewal can take place. Renewal process can be automated to overcome manual work, such as creation, validation, signing and installation.

There is a rate limit to ensure fair usage by as many people as possible [6]. The main limit is 50 Certificates per Registered Domain (per week). A single certificate can contain up to 100 multiple names (SAN certificate) in it. For renewals there is a Duplicate Certificate limit of 5 per week. A certificate is considered a renewal (or a duplicate) of an earlier certificate if it contains exactly the same set of hostnames.

Let's Encrypt issues only domain-validated certificates [7], since they can be fully automated by the use of Automatic Certificate Management Environment (ACME) [8] communications protocol for automating interactions between CA and certificate requestors, allowing the automated deployment of public key infrastructure.

ACME v2 and wildcard certificates were introduced in March 2018. Wildcard certificates allow to secure an unlimited number of subdomains with a single certificate, and it can be done for free with Let's Encrypt now. It made this solution very interesting for us to implement, since a single shared "wildcard" certificate can be centrally obtained from Let's Encrypt and then distributed to several servers under this domain.

The requests for wildcard certificates require the modification of a Domain Name Service "TXT" record, verifying control over the domain, and it means that automation of DNS updates should also be implemented for that particular domain.

3. WILDCARD SSL CERTIFICATE DEPLOYMENT AUTOMATION

Automation of getting, installing and renewing the Let's Encrypt Wildcard SSL certificates can be effectively done by means of centralized certificate server, with an appropriate configuration. It should have a possibility to create requests for Let's Encrypt certificates, as well as publish the required "TXT" record in DNS zone of particular domain.

One of the ways to implement the centralized certificate server configuration is by using the Dehydrated Automation tool (script) [9]. It can do all the required steps to request, get and renew the certificate. Since the DNS zone needs to be modified for Wildcard SSL certificates, the same server can run a DNS service on it, to manage and dynamically update records of the appropriate "*_acme-challenge.*<*domain*>" zone (<domain> is considered to be "asnet.am" in the examples below). Thus, as a requirement for any domain to get the Wildcard SSL certificates through this centralized certificate server configuration, a zone delegation:

_acme-challenge.asnet.am. NS <centralized-nameserver>.

should be created in the main DNS infrastructure of that domain.

Configuration at the centralized certificate server should allow dynamic updates of "*_acme-challenge. asnet.am*" zone from the Dehydrated script. It can be done by a hook script able to do DNS-01 challenge [10]. Hook script example for BIND9 DNS server can be found in [11]. It uses "nupdate" to update the required zone and, thus, to prove the ownership of that domain.

After the BIND9 DNS server is configured with the required "*acme-challenge.asnet.am*" zones, the Dehydrated script should be configured with the required domain names to request Wildcard SSL certificate.

In */etc/dehydrated/config the* following lines should be present:

```
CHALLENGETYPE="dns-01"
DOMAINS_TXT="${BASEDIR}/domains.txt"
```

Next */etc/dehydrated/domains.txt* file should contain the domain names, for which Wildcard SSL certificate needs to be obtained. There are several possibilities for that, but we propose to use the following format:

```
*.asnet.am  asnet.am  >  star.asnet.am
```

This will create a certificate for '**.asnet.am*' with an alternative name '*asnet.am*' and store it in the directory *${CERTDIR}/star.asnet.am* i.e., it will support both wildcard certificate for subdomains and the plain domain name.

Initial run of Dehydrated script requires to register and accept terms of use:

```
dehydrated --register --accept-terms
```

Next, to obtain all certificates listed in */etc/dehydrated/domains.txt* file, it's required to run the Dehydrated script as follows:

```
dehydrated -c -k hook2
```

where *hook2* is the name of hook script to implement DNS-01 challenge [11].

There is also a possibility to force getting new certificates even if we already have valid ones by adding "*-x*" option:

```
dehydrated -c -k hook2 -x
```

4. DISTRIBUTION OF CERTIFICATES TO SEVERAL SERVERS

Several solutions to distribute the certificates to endnode servers can be used. One of them is the use of *rsync* tool over ssh, with public key authentication. In order to do that, each endnode server should have ssh public/private keypair generated and public key copied to centralized certificate server. This will allow rsync to run at each endnode and access centralized certificate server without password to obtain and update the particular certificate.

Since Let's Encrypt certificates are valid for 90 days, their renewal can be automated by simply running the Dehydrated script periodically at centralized certificate server using some scheduling solution like *cron*. The same can be done with running *rsync* at endnode servers. Thus, each endnode server will have the required up-to-date certificate and will still remain secure because it will contain only the copy of the certificate.

5. CONCLUSION

Described concept allows the creation of automated system to provision and update SSL certificates on multiple servers from one centralized certificate server. SSL certificates are being centrally obtained and updated from Let's Encrypt and then can be made securely and internally available at the centralized certificate server. Each endnode server can access that certificate server to get and use the certificate it requires. In case of ASNET-AM network services this approach is highly effective, since, it allows to use the same wildcard certificates to secure multiple services like, webhosting (HTTPS), Email (SMTPS/IMAPS/POP3S) and others. Given the fact that several domains are in use at ASNET-AM, the automation of the process mentioned above will increase the effectiveness of securing the network services.

REFERENCES

[1] The Academic Scientific Research Computer Network of Armenia (ASNET-AM) http://www.asnet.am

[2] TCS - Trusted Certificate Service https://www.geant.org/Services/Trust_identity_and_security/Pages/TCS.aspx

[3] GEANT - https://www.geant.org

[4] Let's Encrypt - free, automated and open CA https://en.wikipedia.org/wiki/Let%27s_Encrypt

[5] Let's Encrypt - free, automated and open CA https://letsencrypt.org/

[6] Let's Encrypt rate limits https://letsencrypt.org/docs/rate-limits/

[7] Domain-validated certificate https://en.wikipedia.org/wiki/Domain-validated_certificate

[8] Automatic Certificate Management Environment (ACME) communications protocol https://en.wikipedia.org/wiki/Automated_Certificate_Management_Environment

[9] Dehydrated Automation Script https://dehydrated.io

[10] DNS-01 challenge https://letsencrypt.org/docs/challenge-types/

[11] Dehydrated hook for DNS-01challenge via BIND9 https://github.com/arthur7373/dehydrated/wiki/Example-hook-script-using-for-_dns-01_-challenge-via-BIND9