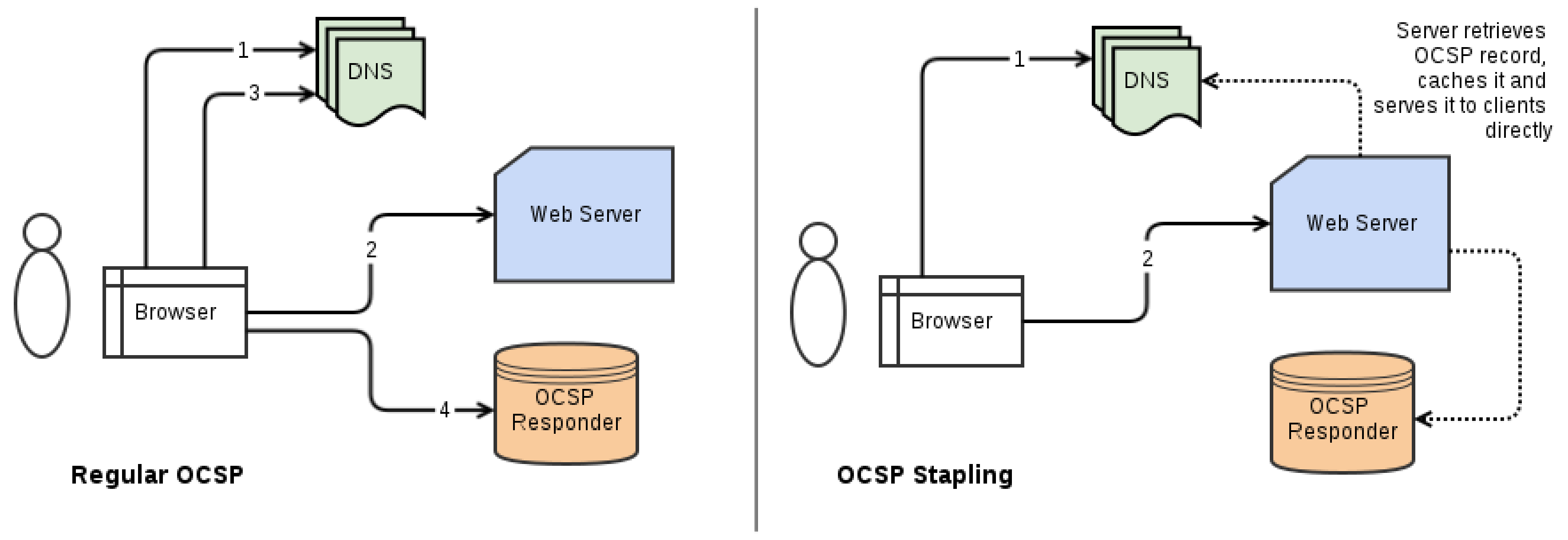
HAProxy OCSP stapling

4 minute read , Mar 30, 2015

The **Online Certificate Status Protocol (OCSP)** is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate. It is used by https clients (browsers) to confirm that the certificate sent by the server they have connected to is a valid one. Every time the client connects to a server it needs to contact the server's CA OCSP responder and check the validity of its certificate. An OCSP responder (a server typically run by the certificate issuer) may return a signed response signifying that the certificate specified in the request is 'good', 'revoked', or 'unknown'. If it cannot process the request, it may return an error code. It has several benefits over Certificate Revocation List (CRL) since an OCSP response contains less information and does not need to be parsed.

OCSP stapling, formally known as the TLS Certificate Status Request extension, is an alternative approach to the OCSP for checking the revocation status of X.509 digital certificates. It allows the presenter (the server) of a certificate to bear the resource cost involved in providing OCSP responses by appending ("stapling") a time-stamped OCSP response signed by the CA to the initial TLS Handshake, eliminating the need for clients to contact the CA.


DNS
1
3
2
4
Web Server
Browser
OCSP Responder
Regular OCSP
Server retrieves OCSP record, caches it and serves it to clients directly
OCSP Stapling

Setting HAP for OCSP stapling

HAProxy implements OCSP stapling since version 1.5.6. By attaching the CA signed OCSP response to the initial SSL/TLS hand shake, the servers speed up the process of the establishing the SSL connection since the client does not need to contact the CRL or the OCSP responder of the CA does saving significant time. In our case, we need to contact the OCSP responder of our CA (Digicert), save the response in a file and tell HAP to use that file for stapling. The process can be described in two steps:

1. Obtain the signed response from the OCSP authority

```
root@server:/etc/haproxy# openssl ocsp -issuer DigiCertSecureServerCA.pem -cert star_mydomain_com.crt -url http://ocsp.digicert.com -header Host ocsp.digicert.com -no_nonce -noverify -respout star_mydomain_com.crt.ocsp
star_mydomain_com.crt: good
    This Update: Mar 29 12:05:00 2015 GMT
    Next Update: Apr  5 12:20:00 2015 GMT
```

2. Dynamically update HAP with this information

```
root@server:/etc/haproxy# echo "set ssl ocsp-response $(/usr/bin/base64 -w 10000 /etc/haproxy/star_mydomain_com.crt.ocsp)" | socat stdio unix-connect:/run/haproxy/admin.sock
OCSP Response updated!
```

We can automate the process with running the following script `/usr/local/bin/ocsp_update.sh` as cronjob on daily basis:

```
#!/bin/bash

# Certificates path and names
DIR="/etc/haproxy"
CERT="star_mydomain_com.crt"

# Get the issuer URI, download it's certificate and convert into PEM format
ISSUER_URI=$(openssl x509 -in ${DIR}/${CERT} -text -noout | grep 'CA Issuers' | cut -d: -f2,3)
ISSUER_NAME=$(echo ${ISSUER_URI##*/} | while read -r fname; do echo ${fname%.*}; done)
wget -q -O- $ISSUER_URI | openssl x509 -inform DER -outform PEM -out ${DIR}/${ISSUER_NAME}.pem

# Get the OCSP URL from the certificate
ocsp_url=$(openssl x509 -noout -ocsp_uri -in ${DIR}/${CERT})

# Extract the hostname from the OCSP URL
ocsp_host=$(echo $ocsp_url | cut -d/ -f3)

# Create/update the ocsp response file and update HAProxy
openssl ocsp -noverify -no_nonce -issuer ${DIR}/${ISSUER_NAME}.pem -cert ${DIR}/${CERT} -url $ocsp_url -header Host $ocsp_host -respout ${DIR}/${CERT}.ocsp
[[ $? -eq 0 ]] && [[ $(pidof haproxy) ]] && [[ -s ${DIR}/${CERT}.ocsp ]] && echo "set ssl ocsp-response $(/usr/bin/base64 -w 10000 ${DIR}/${CERT}.ocsp)" | socat stdio unix-connect:/run/haproxy/admin.sock

exit 0
```

The script, apart from the last line that updates the HAP, is pretty generic and can be used with any PEM type certificate with some other services like Apache, Nginx etc. For the case of DigiCert the OCSP gets updated weekly so we can set a cronjob for the `root` user to update haproxy, for example:

```
# HAP OCSP
0 5 * * * /usr/local/bin/ocsp_update.sh
```

To test the OCSP stapling:

```
igorc@igor-laptop:~/Downloads$ echo quit | openssl s_client -connect server.mydomain.com:443 -status
CONNECTED(00000003)
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
verify error:num=19:self signed certificate in certificate chain
verify return:0
OCSP response:
======================================
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: 9071DB37EB73C8EFDCD51E12B634BA2B5AA0A692
    Produced At: Mar 29 12:05:00 2015 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 20A6948B01BEE825A43A455576661F55C1920633
      Issuer Key Hash: 9071DB37EB73C8EFDCD51E12B634BA2B5AA0A692
      Serial Number: 064E51056C1688FFD0A9172758797C49
    Cert Status: good
    This Update: Mar 29 12:05:00 2015 GMT
    Next Update: Apr  5 12:20:00 2015 GMT

    Signature Algorithm: sha1WithRSAEncryption
         3c:9e:39:45:51:c5:55:61:78:26:cc:3e:3d:78:44:b7:37:53:
         f8:09:25:a1:80:8a:2f:39:ca:db:1e:94:d5:69:47:f6:c6:cc:
         c2:74:7b:fb:0c:75:67:49:a2:5d:17:69:68:5b:43:fa:d7:27:
         53:76:24:13:e9:7f:7e:5b:da:25:9e:a7:ee:10:56:33:2e:f0:
         d6:1b:13:a7:78:fd:96:23:51:5f:5b:9e:33:55:d1:73:c7:6a:
         cd:50:30:8f:40:8c:bc:0e:5f:e7:14:04:fc:cd:88:d1:5f:e6:
         34:2a:41:22:92:29:c6:cc:b8:bb:26:1d:ef:4b:c1:17:35:be:
         c9:f4:19:11:d6:b7:70:18:72:19:b8:ad:b3:f6:be:c6:95:cd:
         97:80:04:ce:12:2d:37:00:db:a3:f0:25:4e:e1:38:f4:90:29:
         2d:e5:4e:6c:e6:a6:d8:02:eb:be:59:f3:55:7c:cb:73:73:2c:
         f9:9a:42:3f:43:7e:ee:6a:52:2f:46:3e:72:ee:ee:78:10:97:
         fc:97:c3:87:f6:9d:15:4e:3b:13:cb:37:db:0f:d8:1f:58:6c:
         c3:1a:45:94:c5:d8:d1:3f:dd:f4:82:c3:e2:a0:51:95:7f:37:
         1b:d3:28:af:bc:23:05:13:53:fe:dc:42:ae:7b:92:ec:2d:e7:
         0b:4a:56:aa
======================================
---
Certificate chain
 0 s:/C=AU/ST=New South Wales/L=Sydney/O=My Corporation Pty Ltd/CN=*.mydomain.com
   i:/C=US/O=DigiCert Inc/CN=DigiCert Secure Server CA
 1 s:/C=US/O=DigiCert Inc/CN=DigiCert Secure Server CA
   i:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root CA
 2 s:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root CA
   i:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root CA
...
```

We can see the server including the OCSP response in the TLS handshake.

Multiple certificates

In this case we pass multiple certificates to HAP installed under `/etc/haproxy/ssl.d/` directory. We need to slightly modify the script for this case:

```
#!/bin/bash

shopt -u nullglob

# Certificates path and names
SSL_DIR="/etc/haproxy/ssl.d"
DIR="/etc/haproxy/ssl.ocsp"
CERTS="${SSL_DIR}/*.crt"

for CERT in $CERTS; do
# Get the issuer URI, download it's certificate and convert into PEM format
ISSUER_URI=$(openssl x509 -in $CERT -text -noout | grep 'CA Issuers' | cut -d: -f2,3)
ISSUER_NAME=$(echo ${ISSUER_URI##*/} | while read -r fname; do echo ${fname%.*}; done)
[[ -d "$DIR"]] || mkdir -p $DIR
ISSUER_PEM="${DIR}/${ISSUER_NAME}.pem"
wget -q -O- $ISSUER_URI | openssl x509 -inform DER -outform PEM -out $ISSUER_PEM
```

```
# Get the OCSP URL from the certificate
ocsp_url=$(openssl x509 -noout -ocsp_uri -in $CERT)

# Extract the hostname from the OCSP URL
ocsp_host=$(echo $ocsp_url | cut -d/ -f3)

# Create/update the ocsp response file and update HAProxy
OCSP_FILE="${SSL_DIR}/${CERT##*/}.ocsp"
openssl ocsp -noverify -no_nonce -issuer $ISSUER_PEM -cert $CERT -url $ocsp_url -header Host $ocsp_host -respout $OCSP_FILE
[[ $? -eq 0 ]] && [[ $(pidof haproxy) ]] && [[ -s $OCSP_FILE ]] && echo "set ssl ocsp-response $(/usr/bin/base64 -w 10000 $OCSP_FILE)" | socat stdio unix-connect:/run/haproxy/admin.sock
done

exit 0
```

Obviously the only limitation is that all certificates have `.crt` extension. The `/etc/haproxy/ssl.ocsp` directory is where the CA certificates are being created during the process.

Tags: haproxy ocsp

Categories: Server

Updated: March 30, 2015

Previous Next

LEAVE A COMMENT