Practical Networking

Practical TLS – Lab Guide

A deep dive into SSL and TLS – the protocols that secure the Internet

Lab 4.2 – Certificate Revocation

There are three methods of verifying the revocation status of a certificate: **Certificate Revocation Lists** (**CRLs**), **Online Certificate Status Protocol** (**OCSP**), and **OCSP Stapling**. In this lab you will check all three methods for two different websites: once for a certificate which you know is revoked, and again for a certificate which you hope is not revoked.

Create working directory and acquire lab files

1. Navigate to the folder you created in Lab 0.0 named Practical-TLS
   - `$ cd Practical-TLS`

2. Create and navigate to a new directory for Lab 4.2 files:
   - `$ mkdir "LAB4.2 - Certificate Revocation"`
   - `$ cd "LAB4.2 - Certificate Revocation"`

3. There are no files to download for this lab. You will be retrieving them or creating them yourself

Lab Instructions

Select one website from this list of sites with known revoked certificates:

- `https://revoked.badssl.com`
- `https://revoked.grc.com`
- `https://revoked-ecc-ev.ssl.com`
- `https://revoked-ecc-dv.ssl.com`

Note: These websites all had revoked certificates as of the creation of this lab guide. If that has changed, please reach out to the Instructor.

Select one more website from an HTTPS site of your choosing. Consider your online bank, or your favorite blog, or your company's home page.

Visit both webpages you've selected using a web browser to see what errors (if any) are shown. Try visiting with different combinations of browsers or different combinations of mobile phones, tablets, and computers. Notice which combinations give you revocation errors and which do not.

Then follow the instructions for ***all three tasks*** below for **both** websites you've selected.

Checking revocation status using a Certificate Revocation List (CRL)

In the instructions below, replace every instance of **site.com** with the website you are testing.

1. Retrieve the Certificate:
   - `openssl s_client -connect site.com:443`
   - Copy the text from `---BEGIN CERTIFICATE---` through `---END CERTIFICATE---`
   - Paste it into a blank text file and save the file as: **`site.com-cert`**
2. Retrieve the Certificate's CRL Distribution Point:
   - `openssl x509 -in site.com-cert -noout -text`
3. Retrieve the Certificate's Serial Number:
   - `openssl x509 -in site.com-cert -noout -serial`
4. Download the CRL from the URL acquired in Step 2 and save it as **`crl-site.com.DER`**:
   - `wget http://CRL.DISTRIBUTION.POINT.URL/FROM-STEP-2.crl -O crl-site.com.DER`
   - *Note:*
     - *The "-O" saves the CRL as the given file name (capital letter O, not zero)*
     - *CRL is downloaded in DER format*
5. Extract the text content of the CRL and save it to a new file:
   - `openssl crl -inform DER -in crl-site.com.DER -noout -text > crl-site.com`
6. Inspect the contents of the CRL file you just downloaded: *[OPTIONAL - informational step]*
   - `ls -hl`
   - `wc crl-site.com`
   - `head -30 crl-site.com`
   - `tail -30 crl-site.com`
   - `cat crl-site.com | grep -A1 Reason | sort | uniq -c | sort -n`
7. Search the CRL for your certificate's serial number acquired in Step 3:
   - `cat crl-site.com | grep -C4 <Serial Number from Step 3>`
   - Note: The serial number should have uppercase letters and no colons, for example:
     `cat crl-site.com | grep -C4 099BE86F0345E7AE9E05E6B74369BD16`

If the last step returned something, then the certificate you were checking exists in the Certificate Revocation List (CRL) and has therefore been revoked. It should not be trusted.

If the prior command returned no output, then the certificate you were verifying is not revoked and can still be trusted.

Checking revocation status using Online Certificate Status Protocol (OCSP)

In the instructions below, replace every instance of **site.com** with the website you are testing.

1. Retrieve the Certificate:
   - *(Same as step 1 in the prior task, if you've already done this it does not need to be repeated)*
   - `openssl s_client -connect site.com:443`
   - Copy the text from `---BEGIN CERTIFICATE---` through `---END CERTIFICATE---`
   - Paste it into a blank text file and save the file as: **`site.com-cert`**
2. Retrieve the OCSP Responder Location:
   - `openssl x509 -in site.com-cert -noout -ocsp_uri`
   - *(Older versions of OpenSSL do not support the `-ocsp-uri` argument. The OCSP URI can also be found in the `Authority Information Access` extension of the Certificate)*
   - `openssl x509 -in site.com-cert -noout -text`
3. Retrieve the Issuer's Certificate URL location
   - Find the URL in the `CA Issuers` field of the `Authority Information Access` extension
   - `openssl x509 -in site.com-cert -noout -text`
4. Retrieve the Issuer's Certificate and convert it to PEM format:
   - `wget http://CA.ISSUERS.URL/FROM-STEP-3.crl -O ica-site.com.DER`
   - `openssl x509 -in ica-site.com.DER -inform DER -out ica-site.com`
5. Request the OCSP Certificate Status:
   - The command requires providing three items:
     - The OCSP URL acquired in step 2
     - The Issuer Certificate acquired in steps 3 and 4
     - The Certificate you are verifying acquired in step 1
   - `openssl ocsp -url http://OCSP.URI -issuer ica-site.com -cert site.com-cert`

The output of the command will tell you two things:

- **`Response verify OK`** - OCSP response successfully received and verified
- **`site.com-cert: good`** *or* **`revoked`** - the Certificate is not revoked, or is revoked

6. Additional arguments can be added to the command in Step 5 to extract additional information:
   - *(Each argument below should be appended to end of the command as Step 5)*
   - `... -no_nonce`
   - `... -req_text`
   - `... -req_text -no_nonce`
   - `... -resp_text`
   - `... -text`

Checking revocation status using OCSP Stapling

OCSP Stapling involves asking for the Certificate Status when you are requesting the Certificate itself. To mimic this behavior you will simply use the same **openssl s_client** command you have already been using with one additional argument: **-status**.

In the instructions below, replace every instance of **site.com** with the website you are testing.

1. Retrieve the Certificate and request the Certificate status:
   - `openssl s_client -connect site.com:443 -status`

Remember, only about 30%~ of Servers support OCSP stapling. You may have to try a few different websites before you can see a successful OCSP stapled response.

2. Force the negotiation of TLS v1.2 and request Certificate status
   - `openssl s_client -connect site.com:443 -tls1_2 -status`

This is what you will see if the Server does **not** support OCSP stapling:

```
 ...
OCSP response: no response sent
---
Certificate chain
 ...
```

*NOTE: This does **not** indicate whether the certificate is valid or revoked.*

This is what you will see if the Server **does** support OCSP stapling:

```
 ...
OCSP response:
======================================
OCSP Response Data:
    OCSP Response Status: successful (0x0)
 ...
    Cert Status: revoked
    Revocation Time: Aug 20 21:24:42 2020 GMT
    This Update: Oct 31 21:57:01 2020 GMT
    Next Update: Nov  7 21:12:01 2020 GMT

Signature Algorithm: sha256WithRSAEncryption

50:6a:6a:0c:6c:6b:6f:73:42:95:aa:c8:79:70:3e:59:
 ...
b9:64:71:ca:f2:ba:f8:f1:8b:ad:e7:f1:5d:37:3f:ef:
42:5c:19:66
======================================
---
Certificate chain
 ...
```

```
 ...
OCSP response:
======================================
OCSP Response Data:
    OCSP Response Status: successful (0x0)
 ...
    Cert Status: good
    This Update: Oct 31 23:33:01 2020 GMT
    Next Update: Nov  7 22:48:01 2020 GMT


Signature Algorithm: sha256WithRSAEncryption

b4:27:2b:dc:29:0d:b1:99:e6:b0:e1:3e:c8:96:c8:23:
 ...
6f:7a:87:e3:fc:4a:1a:46:c5:80:1a:e0:9c:d9:aa:95:
af:a9:
======================================
---
Certificate chain
 ...
```