Domain Control Validation

Version History

1.0	Original version
1.01	Minor updates
1.02	MDCs now supported by email-DCV. HTTP and DNS CNAME supported via web-interface.
1.03	MDCs now support HTTP and CNAME mechanisms, re-issue conditions updated.

1. Introduction

All Comodo server certificates must have DCV (Domain Control Validation) performed on them.
DCV is a way to prove some level of control of a registered domain name.

DCV can be performed using any one of three methods:

- Email challenge-response
- Creation of a file on the domain's HTTP server
- Creation of a DNS CNAME record for that domain

2. Email Challenge-Response

This is the default method used for Comodo certificate orders.
When the order is placed, an email address is selected from a shortlist of acceptable options. An email is sent to that address, containing a unique validation code.
The email should be received by someone in control of the domain, where they can follow a link provided in the email and enter the validation code, thus proving domain control.

The list of acceptable email addresses for any given domain are:

- admin@
- administrator@
- hostmaster@
- postmaster@
- webmaster@
- Any email address that appears on the domain's WHOIS record, and is visible to our CA system.

Using via web-interface
The web-interface certificate request process will default to offering email challenge-response DCV. The list of acceptable email addresses will be displayed for selection, based on the common name extracted from the CSR.

Using via API
Once the CSR is received from the customer, the FQDN of the common name (CN) from the CSR must be extracted. The DecodeCSR API can be used, or any other method you have available.
The FQDN must then be passed to the GetDCVEmailAddressList API, which will return a complete list of acceptable email addresses for that FQDN.
ONLY email addresses returned from GetDCVEmailAddressList are acceptable. If an email address believed to be on the WHOIS record for the domain is not returned, this means our system was unable to extract it from a WHOIS query, and thus the address cannot be used.

Then, once a choice is made from the acceptable email address list, that address can be passed to the AutoApplySSL API, as the dcvEmailAddress parameter. Once the call is made to AutoApplySSL with this parameter, the DCV email will be sent.

Due to our caching of the WHOIS record result, the API call to AutoApplySSL must be made within 24 hours of the GetDCVEmailAddressList API call.
A call to GetDCVEmailAddressList is not required if the dcvEmailAddress selection is from the 5 'default' email addresses (i.e. not one extracted from the domains WHOIS record).

For multi-domain certificates, please see the information in Section 5.

3. HTTP Based DCV

HTTP based DCV requires that a HTTP server be running on port 80 of the FQDN requested as the common name of the certificate.
Two hashes of the CSR are generated before submission to Comodo. A simple text file is created on the HTTP server of the FQDN, with one hash as the filename, and one hash within the text file itself. Additionally a domain name is added to the text file, for expansion with future domain validation mechanisms.

For example:
A CSR is generated with the CN=www.example.com
The CSR is hashed using both the MD5 and SHA-1 hashing algorithms.

A text file is created, containing the SHA-1 hash and the domain 'comodoca.com' on the next line.

c7fbc2039e400c8ef74129ec7db1842c
comodoca.com

The file is then named in the format: <MD5 hash>.txt and placed in the root of the HTTP server, like so:

http://www.example.com/C7FBC2039E400C8EF74129EC7DB1842C.txt

Once the order is received by Comodo and the HTTP based DCV is specified, the Comodo CA system checks for the presence of the text file, and it's content. If the file is found and the hash values match, domain control is proven.

Using via web-interface
The hash values are calculated and presented via the web-interface during the order process. They are on the same screen as the DCV email-address options.
Both the MDC5 and SHA-1 hash values of the CSR are shown, and must be saved to the file served from your HTTP server as above before continuing with the order.

Using via API
The hashes are generated from the CSR before the order is submitted to Comodo.
The hashes MUST be generated from the DER-encoded (i.e. binary) version of the CSR – not the base64 PEM encoded version. Variations in the PEM encoding can cause differing hash values, whereas the hashes of the DER encoded version will remain constant.
The file must be created using the UPPERCASE formatting of the MD5 hash, as most HTTP servers are case-sensitive. The Comodo CA system will only look for the uppercase hash filename.
The file must be created with a .txt extension.
The SHA-1 hash within the file is case-insensitive.

The Comodo CA system will look for the file at both the FQDN provided in the CSR, as well as the 'base' registered domain name. Thus in the above example, the CA system will check for the file at 'www.example.com', and if the file is not found, it will also check 'example.com'.

When the AutoApplySSL call is made, an additional optional parameter must be specified to indicate use of HTTP based DCV.
This parameter is called 'dcvMethod' and must be set to the value (uppercase) 'HTTP_CSR_HASH'.

For multi-domain certificates, please see the information in Section 5.

4. DNS CNAME Based DCV

DNS based DCV requires the creation of a unique CNAME record, pointed back to Comodo.

Two hashes of the CSR are generated before submission to Comodo.
A CNAME DNS record is created under the FQDN that the certificate is requested for, in the format:

<MD5 hash>.FQDN CNAME *<SHA-1 hash>.comodoca.com*

For example, a certificate is requested for the FQDN www.example.com. The CSR has the hashes:
MD5: c7fbc2039e400c8ef74129ec7db1842c
SHA-1: 298a056d3e2f3018bda514defb18129dc5af459e

To perform DNS CNAME based DCV, the following DNS CNAME record must be created before submitting the order:

c7fbc2039e400c8ef74129ec7db1842c.www.example.com CNAME
298a056d3e2f3018bda514defb18129dc5af459e.comodoca.com

Then the request is submitted to Comodo, the presence of this CNAME DNS record is checked, and if found, domain control is proven.

Using via web-interface
The hash values are calculated and presented via the web-interface during the order process. They are on the same screen as the DCV email-address options.
Both the MDC5 and SHA-1 hash values of the CSR are shown, and must be added to DNS as a CNAME record as the above instructions show before continuing with the order.

Using via API
The hashes are generated from the CSR before the order is submitted to Comodo.
The hashes MUST be generated from the DER-encoded (i.e. binary) version of the CSR – not the base64 PEM encoded version. Variations in the PEM encoding can cause differing hash values, whereas the hashes of the DER encoded version will remain constant.
The DNS CNAME record is then created in the format above.

The Comodo CA system will look for the file at both the FQDN provided in the CSR, as well as the ‘base’ registered domain name. Thus in the above example, the CA system will check for the CNAME record at ‘www.example.com’, and if the record is not found, it will also check ‘example.com’.

When the AutoApplySSL call is made, an additional optional parameter must be specified to indicate use of HTTP based DCV.
This parameter is called ‘dcvMethod’ and must be set to the value (uppercase) ‘CNAME_CSR_HASH’.

For multi-domain certificates, please see the information in Section 5.

5. Notes

All certificate types (single, wildcard, MDC) can be validated with any of the three available DCV mechanisms. Multi-domain certificates can use a combination of any of the mechanisms for all FQDNs in the request.

Re-issuing

Re-issues of the certificates will require re-validation *unless* the re-issue is within 7 days of the original validation.

We now allow the re-issue to *not* require revalidation of already-validated FQDNs *if the same private key is used to generate the CSR for re-issue*. If a new private key is used to generate the CSR, then the order must have DCV re-performed by one of the available methods for all FQDNs in the request before the certificate can be issued.
This will also apply to re-issues that facilitate the addition or removal of domains for multi-domain certificates.

Re-sending DCV Emails

DCV emails can be resent from within the web-interface, or via the API ‘ResendDCVEmail’.
This will resend the DCV email for single-certificate orders, and for multi-domain certificate orders all outstanding (i.e. unvalidated) FQDNs the emails will be resent.

Multi-domain certificates

Multi-domain certificates (MDCs, UCCs) now require DCV for all orders. Any of the available mechanisms (email, HTTP and DNS CNAME) can be used. The web-based interface for this is provided once the order is placed. Simply login to your account and locate the order.
You will be presented with a screen that allows for selection of any valid email address to validate each FQDN in the certificate. Our system will run WHOIS lookups for all domains to provide the email addresses scraped from those records where possible. You can use the ‘Refresh’ button to update the data on screen – large numbers of FQDNs in a single certificate will take some time for all WHOIS records to be read.
The web interface can also be used to choose the HTTP or DNS CNAME mechanisms for each FQDN.

You can also remove some FQDNs from the certificate if they are unable to be validated, and issue the certificate with only the domains validated to that point.

Multi-domain certificate API Details

The API can be used to request and validate multi-domain certificates via email-based DCV.
The changes to the existing API and processes for ordering are:

- Instead of the ‘dcvEmailAddress’ parameter, there is now a ‘dcvEmailAddresses’ (note the plural) parameter.

This parameter accepts a list of email addresses to use for DCV. There must be one email address per FQDN in the ‘domainNames’ parameter, and they must be in exactly the same order.
Unlike single certificate orders via the API, our system will *not* reject orders because of incorrect DCV email addresses. They will be accepted, but no email will be sent. They can then be edited via the web-interface.
It is important to pass only valid email addresses for the DCV email address for each domainName.
Valid email addresses can be obtained by either using one of the default 5 (listed earlier in this document), or by calling the GetDCVEmailAddressList API for that domain.
It is also possible for you to perform and independent WHOIS lookup and send an email you can extract from the output to our API. However, please note that our system can only ‘see’ email addresses on a WHOIS lookup that is from the command-line. Email addresses visible from web-based WHOIS queries, or any that require human challenge-response systems (e.g. CAPTCHAS) will not be usable by our system, and the order will sit awaiting verification until you login and update the DCV email address via the web-interface.

Our system will attempt to send as few emails as possible. Where several FQDNs exist within the same registered domain name, one email will be sent

The API can also be used to validate domains using the HTTP and DNS CNAME mechanisms.
As above, the ‘dcvEmailAddresses’ parameter should be used. For each domain in the ‘domainNames’ parameter, in order, you can specify one of the following:

- A DCV email address as detailed in this document.
- The value ‘**HTTPCSRHASH**’ – and for this ‘domainName’ the HTTP DCV mechanism will be used.
- The value ‘**CNAMECSRHASH**’ – and for this ‘domainName’ the DNS CNAME DCV mechanism will be used.

In addition, if you wish to have *all* the domains validated by one of the alternative (HTTP or DNS CNAME) mechanisms, then simply pass a single value for the ‘dcvEmailAddresses’ parameter of:

- **ALLHTTPCSRHASH**
  or
- **ALLCNAMECSRHASH**

This should be the only value passed for the parameter, and will attempt to verify all domains via the same mechanism.