Certificates

https://campus.barracuda.com/doc/45026235/

A signed certificate is a digital identity document that enables both server and client to authenticate each other. Certificates are used with HTTPS protocol to encrypt secure information transmitted over the internet. A certificate can be generated or procured from a third party Certificate Authority (CA). Generated certificates can be self-signed or signed by a trusted third-party CA. A certificate contains information such as user name, expiration date, a unique serial number assigned to the certificate by a trusted CA, the public key, and the name of the CA that issued the certificate.

To Create a Certificate

URL: /v1/certificates			
Method: POST			
Description: Creates a self-signed certificate with the given values.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
key_type	String	Optional	The key/algorithm used in the certificate. The values include: • rsa • ecdsa **Note**: By default, **key_type** is *rsa*. If the key used in the certificate is ECDSA, then specify *ecdsa* as **key_type**.
common_name	Alphanumeric	Yes	The domain name (DN) of the web server for which you want to generate the certificate.
san_certificate	Alphanumeric	Yes	The Subject Alternative Names (SAN) that needs to be associated with the certificate. The values include: • DNS • Email • URI • IP **Example**: • **DNS**: barracuda.yourdomain.com • **URI**, Enter a Fully Qualified Domain Name (FQDN). Example: http://www.barracuda.com • **Email**: rwatson@barracuda.com • **IP**: 192.168.7.1
country_code	Alphabetic	Yes	The two-letter country code of the location of the organization.

state	Alphabetic	Optional	The full name of the state or province of the location of the organization.
city	Alphabetic	Optional	The full name of the locality (city) where the organization is located.
organization_name	Alphanumeric	Optional	The legally registered name of the organization or company.
organization_unit	Alphanumeric	Optional	The department or unit within the organization.
key_size	Enumeration	Yes	The private key size for the certificate in bits. The enumerated values include: • 1024 • 2048 • 4096
curve_type	Enumeration	Optional	The elliptic curve. The enumerated values include: • secp256r1 • secp384r1 • secp512r1
allow_private_key_export	String	Yes	Specify whether to lock the Private Key corresponding to this certificate or not. The values include: • yes • no Normally, certificates are downloaded in PKCS12 format which includes the Private Key and Certificate. When a key is locked, you can only download the certificate in PEM format. Also, you cannot take a backup when the Private Key is locked. **Note**: This option is valid only for created and uploaded (generated and signed by a trusted CA) certificates.

Request:

curl http://10.11.28.179:8000/restapi/v1/certificates -u'eyJldCI6IjEzNDg3MTYzNzkwIiwicGFzc3dvcmQiOiJkODA0NjU3ZWE0NjFlZDJjMmI0YjNiYjVm\nNmJkN2I0ZSIsInVzZXIiOiJhZG1pbiJ9\n:' -X POST -H Content-Type:application/json -d'{"name":"Certificate13","common_name":"barracuda.yourdomain.com","country_code":"US","state":"California","city":"Campbell","organization_name":"BarracudaNetworks","organization_unit":"Engineering","key_size":"1024","allow_private_key_export":"yes","san_certificate":["IP:10.11.19.76","DNS:mydomain","Email:sd@gmail.com","URI:https://example.org/absolute/URI/with/absolute/path/to/resource.txt"],"key_type":"ecdsa","curve_type":"secp256r1"}'

Response:

{"id":"certificate1","token":"eyJldCI6IjE0ODcxNDQ5MjQiLCJwYXNzd29yZCI6IjFlMjJmMjI0ZGQzNzFhN2VjYTc1OGE2OWY1\nYzM5OWJmIiwidXNlciI6ImFkbWluIn0=\n"}

Request:

curl http://10.11.28.179:8000/restapi/v1/certificates -u'eyJldCI6IjEzNDg3MTYzNzkwIiwicGFzc3dvcmQiOiJkODA0NjU3ZWE0NjFlZDJjMmI0YjNiYjVm\nNmJkN2I0ZSIsInVzZXIiOiJhZG1pbiJ9\n:' -X POST -H Content-Type:application/json -d'{"name":"Certificate3","common_name":"barracuda.yourdomain.com","country_code":"US","state":"California","city":"Campbell","organization_name":"BarracudaNetworks","organization_unit":"Engineering","key_size":"1024","allow_private_key_export":"yes","san_certificate":["IP:10.11.19.76","DNS:mydomain","Email:sd@gmail.com","URI:https://example.org/absolute/URI/with/absolute/path/to/resource.txt"],"key_type":"ecdsa","curve_type":"secp256r1"}'

Response:

{"id":"Certificate3","token":"eyJldCI6IjE0ODcxNDQ5MjQiLCJwYXNzd29yZCI6IjFlMjJmMjI0ZGQzNzFhN2VjYTc1OGE2OWY1\nYzM5OWJmIiwidXNlciI6ImFkbWluIn0=\n"}

To Upload a Signed Certificate

URL: /v1/certificates?upload=signed			
Method: POST			
Description: Uploads the given signed (pem or pkcs12) certificate.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
type	String	Yes	Select the certificate type. The values include: • pkcs12 • pem
key_type	String	Optional	The key/algorithm used in the certificate. The values include: • rsa • ecdsa **Note**: By default, **key_type** is *rsa*. If the key used in the certificate is ECDSA, then specify *ecdsa* as **key_type**.
signed_certificate	String	Yes	The path and name of the signed certificate file that needs to be uploaded.

assign_associated_key	String	Conditional	The values include: • yes – If the CSR corresponding to this certificate was generated on the Barracuda Web Application Firewall. • no – Upload the private key corresponding to this certificate in the "*key*" field. **Note**: Required ONLY when the certificate being uploaded is in PEM format.
key	String	Conditional	The path and name of the corresponding private key for the signed certificate being uploaded. **Note**: Required ONLY when the certificate being uploaded is in PEM format.
intermediary_certificate	String	Conditional	The path and name of the intermediary CA certificate file that needs to be uploaded. **Note**: If your certificate is signed by a trusted CA, the certificate should be uploaded in the following order: • Leaf certificate • Intermediate certificate(s) • Root CA certificate This is required ONLY when the certificate being uploaded is in PEM format.
allow_private_key_export	String	Yes	Determines whether to export the private key corresponding to the certificate or not. The values include: • yes – To export the private key corresponding to the certificate. • no – To lock the private key. In this case, the certificate can be downloaded only in PEM format, and backup of system configuration cannot be taken.
password	Alphanumeric	Conditional	The password used to generate the PKCS #12 token for the signed certificate being uploaded. **Note**: Required ONLY when the certificate being uploaded is PKCS12 Token.

Example: Uploading a Signed Certificate in PEM Format

Request:

curl -i -F name=cert10 -F signed_certificate=@/home/gireesh/RestAPI/abc_bc_com.crt -F key=@/home/gireesh/RestAPI/abc_bc_com_key.pem -F assign_associated_key=no -F

key_type=rsa -F type=pem -F allow_private_key_export=yes http://10.11.25.108:8000/restapi/v1/certificates?upload=signed -u 'eyJldCI6IjE0NzQwMTg5NjciLCJwYXNzd29yZCI6IjJhMWViMDhmNTdlOTY2NjRiZTE4Y2VhOWRh\nMWJmZjA5IiwidXNlciI6ImFkbWluIn0=\n:'

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Fri, 03 Jul 2015 10:46:10 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive
{"id":"cert10","token":"eyJldCI6IjE0NzQwMTg5NjciLCJwYXNzd29yZCI6IjJhMWViMDhmNTdlOTY2NjRiZTE4Y2VhOWRh\nMWJmZjA5IiwidXNlciI6ImFkbWluIn0=\n"}

Example 1: Uploading a Signed Certificate in PKCS12 Token Format

Request:

curl -i -F name=Cert3 -F signed_certificate=@/home/gireesh/RestAPI/Barracuda.p12 -F type=pkcs12 -F key_type=rsa -F allow_private_key_export=yes -F password='password1231' http://10.11.25.108:8000/restapi/v1/certificates?upload=signed -u 'eyJldCI6IjE0NzQwMTgyNzEiLCJwYXNzd29yZCI6IjU5NmI5MThkZDNlNzMyNzdmZmQ2NmY3ZWZh\nMmE2Y2QyIiwidXNlciI6ImFkbWluIn0=\n:'

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Tue, 19 Nov 2013 12:31:56 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

{"id":"Cert3","token":"eyJldCI6IjE0NzQwMTgyNzEiLCJwYXNzd29yZCI6IjU5NmI5MThkZDNlNzMyNzdmZmQ2NmY3ZWZh\nMmE2Y2QyIiwidXNlciI6ImFkbWluIn0=\n"}

Example 2: Uploading a Signed Certificate in PKCS12 Token Format

Request:

```
curl -i -F name=cedr -F type=pkcs12 -F signed_certificate=@/root/raj_ssl/cert/ecdsa1.p12 -F
key_type=ecdsa -F password=123456 -F allow_private_key_export=yes
http://10.11.25.107:8000/restapi/v1/certificates?upload=signed -u
'eyJldCI6IjE0Mzg5MzU5NzAiLCJwYXNzd29yZCI6Ijg0YTg0YzRkMDlhYWIzZmEwOGEyNmU1ZDg4\n
YzRjMTNkIiwidXNlciI6ImFkbWluIn0=\n:'
```

Response:

```
HTTP/1.1 201
Server: BarracudaHTTP 4.0
Date: Fri, 24 Jul 2015 11:21:04 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
{"id":"cedr","token":"eyJldCI6IjE0Mzg5MzY4NjliLCJwYXNzd29yZCI6ImQxYjYxMGRlZGI1OGRiYzY1
MTJiYzcxYmM2\nMDI4MDFiIiwidXNlciI6ImFkbWluIn0=\n"}
```

To Upload a Trusted (CA) Certificate

Use this API to upload a Certificate Authority's (CA) certificate, a trusted certificate that acts as a root CA certificate for authenticating the client certificates. Any client certificate signed by the trusted certificate is valid and allowed access without further validation.

URL: /v1/certificates?upload=trusted			
Method: POST			
Description: Uploads the given trusted CA certificate.			
Parameter Name	**Data Type**	**Mandatory**	**Description**

Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
trusted_certificate	String	Yes	The path and name of the trusted CA certificate that needs to be uploaded.

Example:

Request:

curl -i -F name=Trusted_Cert -F trusted_certificate=@/home/certs/rootca.pem http://192.168.0.1:8000/restapi/v1/certificates?upload=trusted -u 'eyJldCI6IjEzODQyOTQyMzUiLCJwYXNzd29yZCI6IjQyZWNlN2JjMTc5MjlhMDZkMzZmZmI5NjYz\nODMyOTk0IiwidXNlciI6ImFkbWluIn0=\n:'

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Tue, 12 Nov 2013 06:46:11 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

{"id":"Trusted_Cert","token":"eyJldCI6IjEzODQyOTU3MDgiLCJwYXNzd29yZCI6ImRhNTU0OTFlNDY5Y2U0NDA4NjcxOTMzZGFj\nNzIyYWZkIiwidXNlciI6ImFkbWluIn0=\n"}

To Upload a Trusted Server Certificate

Use this API to upload a Certificate Authority's (CA) certificate, a trusted certificate that acts as a root CA certificate for authenticating back-end server certificates. Any back-end server certificate signed by the uploaded trusted certificate is valid and allowed access without further validation.

URL: /v1/certificates?upload=trusted_server
Method: POST

Description: Uploads the given trusted server certificate.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
trusted_server_certificate	String	Yes	The path and name of the trusted server certificate that needs to be uploaded.

Example:

Request:

curl -i -F name=Server_cert1 -F trusted_server_certificate=@/home/certs/rootca.pem http://192.168.0.1:8000/restapi/v1/certificates?upload=trusted_server -u 'eyJldCI6IjEzODQyOTQyMzUiLCJwYXNzd29yZCI6IjQyZWNlN2JjMTc5MjlhMDZkMzZmZmI5NjYz\nODMyOTk0IiwidXNlciI6ImFkbWluIn0=\n:'

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Tue, 12 Nov 2013 06:49:45 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

{"id":"Server_cert1","token":"eyJldCI6IjEzODQyOTU5NjEiLCJwYXNzd29yZCI6ImNjN2ZjOWNiNWQ3NTJlNDM1MGJiNjk2YmQz\nNzZlOGU0IiwidXNlciI6ImFkbWluIn0=\n"}

To Download a Signed Certificate

Use this API to download a signed certificate. For more information on certificates, refer to Certificate Management.

In the web interface of the Barracuda Web Application Firewall, the certificate is saved as a PKCS12 token (p12). Therefore, it is recommended to append **.p12** extension next to the certificate in the API call.

URL: /v1/certificates/{certificate_name}			
Method: GET			
Description: Downloads the given certificate.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
download	Binary	Yes	Determines whether the certificate needs to be downloaded or not. One (1) - to download the certificate.
encrypt_password	Alphanumeric	Yes	The password to save the certificate.

Example:

Request:

curl http://192.168.0.1:8000/restapi/v1/certificates/Cert1 -u 'eyJldCI6IjEzOTM1MDE3MTAiLCJwYXNzd29yZCI6IjU2YjliNGY2MzFlZjg5ZmU5Y2ZkNGZlNTYy\nNDIzODM5IiwidXNlciI6ImFkbWluIn0=\n:' -H Content-Type:application/json -X GET -o rft.p12 -G -d download=1 -d encrypt_password=123456

Response:

```
 % Total  % Received % Xferd Average Speed  Time  Time  Time  Current

                                Dload  Upload Total Spent Left  Speed

100 2485  0 2485   0 0 7102  0 699    0 --::--   --::-- --::-- 7223
```

To Download a Trusted (CA) Certificate or Trusted Server Certificate

Use this API to download a trusted (CA) certificate or trusted server certificate.

In the web interface of the Barracuda Web Application Firewall, a trusted (CA) certificate or trusted sever certificate is saved in PEM format. Therefore, it is recommended to append **.pem** extension next to the certificate in the API call.

URL: /v1/certificates/{certificate_name}

Method: GET			
Description: Downloads the given certificate.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
download	Binary	Yes	Determines whether the certificate needs to be downloaded or not. One (1) - to download the certificate.

Example:

Request:

curl http://192.168.0.1:8000/restapi/v1/certificates/server_cert1 -u 'eyJldCI6IjEzOTM1MDM1NDYiLCJwYXNzd29yZCI6ImYwMGMwMzM1OTI2YzExNTYzZTRlN2Y1ZWI0\nZTc3MTRhIiwidXNlciI6ImFkbWluIn0=\n:' -H Content-Type:application/json -X GET -o raj.pem -G -d download=1

Response:

```
 % Total  % Received % Xferd Average Speed  Time  Time  Time  Current

                      Dload   Upload Total Spent Left  Speed

100 1334  0 1334 0    0 7102  0 1537   0 --::--   --::-- --::-- 1543
```

To Retrieve Certificates

URL: /v1/certificates /v1/certificates/{certificate_id}			
Method: GET			
Description: Lists all certificates if “certificate_id” is not specified.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
parameters	Alphanumeric	Optional	Any specific parameter name that needs to be retrieved. See *Example 3*.

Example 1:

Request:

curl http://192.168.0.1:8000/restapi/v1/certificates -u

'eyJldCI6IjEzODYxNzAzNTIiLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTIy\nODljMzRjIiwidXNlciI6ImFkbWluIn0=\n:' -X GET

Response:

{"parameters":null,"object":"Certificates","data":[{"expiry":"Dec 1 06:06:16 2023 GMT\n","common_name":"barracuda.yourdomain.com","services":"No Service","private_key":"exportable","name":"Cert_cr_1","type":"created_certificate"},{"expiry":"Dec 1 06:06:25 2023 GMT\n","common_name":"waf4.bc.com","services":"ss1","private_key":"exportable","name":"cert_cr_2","type":"created_certificate"},{"expiry":"Dec 1 06:06:34 2023 GMT\n","common_name":"waf.bc.com","services":"No Service","private_key":"exportable","name":"cert_cr_3","type":"created_certificate"},{"expiry":"Dec 1 06:07:02 2023 GMT\n","common_name":"adc.bc.com","services":"No Service","private_key":"not_exportable","name":"cert_cr_4","type":"created_certificate"},{"expiry":"Dec 31 23:59:59 2013 GMT\n","common_name":"gdfews-globalenergy-stg.gdfsuez.com","services":"No Service","private_key":"not_exportable","name":"chained_6","type":"uploaded_certificate"},{"expiry":"Jul 25 12:04:51 2014 GMT\n","common_name":"wafqa.net","services":"No Service","private_key":"not_exportable","name":"cert9","type":"uploaded_certificate"},{"expiry":"Dec 31 23:59:59 2013 GMT\n","common_name":"gdfews-globalenergy-stg.gdfsuez.com","services":"No Service","private_key":"exportable","name":"chained_68","type":"uploaded_certificate"},{"expiry":"Jul 25 11:57:11 2014 GMT\n","common_name":"CN","services":"No Service","name":"ca2","type":"trusted_certificates"},{"expiry":"Jan 22 13:22:28 2016 GMT","common_name":"wafqa-1","services":"N/A","name":"svr_cert2","type":"trusted_server_certificates"}],"token":"eyJldCI6IjEzODYyNzk0MDMiLCJwYXNzd29yZCI6IjBjZDcyYzkzZWEzOGFkMDExMjE4OGQ2MDBl\nMjkxMTgwIiwidXNlciI6ImFkbWluIn0=\n"}

Example 2:

Request:

curl http://192.168.0.1:8000/restapi/v1/certificates/Cert1 -u 'eyJldCI6IjEzODYxNzAzNTIiLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTIy\nODljMzRjIiwidXNlciI6ImFkbWluIn0=\n:' -X GET

Response:

{"expiry":"Dec 1 06:06:16 2023 GMT\n","common_name":"barracuda.yourdomain.com","services":"No Service","private_key":"exportable","name":"Cert_cr_1","type":"created_certificate","token":"eyJldCI6IjEzODYyNzk0MDYiLCJwYXNzd29yZCI6IjRiYmYyMDQyNTQ5M2I2Yjc3MDU1ZjY3MWE3\nZDFh

MTQ0IiwidXNlciI6ImFkbWluIn0=\n"}

Example 3:

Request:

curl http://192.168.0.1:8000/restapi/v1/certificates -u 'eyJldCI6IjEzODYxNzAzNTIiLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTIy\nODljMzRjIiwidXNlciI6ImFkbWluIn0=\n:' -X GET -G -d parameters="name,type"

Response:

{"parameters":"name,type","object":"Certificates","data":[{"name":"Cert_cr_1","type":"created_certificate"},{"name":"cert_cr_2","type":"created_certificate"},{"name":"cert_cr_3","type":"created_certificate"},{"name":"cert_cr_4","type":"created_certificate"},{"name":"chained_6","type":"uploaded_certificate"},{"name":"cert9","type":"uploaded_certificate"},{"name":"chained_68","type":"uploaded_certificate"},{"name":"ca2","type":"trusted_certificates"},{"name":"svr_cert2","type":"trusted_server_certificates"}],"token":"eyJldCI6IjEzODYyNzk0MDQiLCJwYXNzd29yZCI6IjFlYTg3MjljZGQ3NGIwZWIzMjhhY2E1MDJj\nMmYxMWUyIiwidXNlciI6ImFkbWluIn0=\n"}

To Delete a Certificate

URL: /v1/certificates/{certificate_id}
Method: DELETE
Description: Deletes the given certificate.

Example:

Request:

curl http://192.168.0.1:8000/restapi/v1/certificates/Cert1 -u 'eyJldCI6IjEzODYxNzAzNTIiLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTIy\nODljMzRjIiwidXNlciI6ImFkbWluIn0=\n:' -X DELETE

Response:

{"msg":"Successfully deleted","token":"eyJldCI6IjEzODYyNzk0NzciLCJwYXNzd29yZCI6ImRiODM5NDE4NGE1YmVlMWY5NDE3ZDM5OTI5\nYjExZTE4IiwidXNlciI6ImFkbWluIn0=\n"}