REST API Examples

https://campus.barracuda.com/doc/96011777/

This article provides a few examples on how to use the Barracuda Web Application Firewall REST APIs:

In this article:

Certificates

- To Upload a Trusted CA Certificate
- To Upload a Trusted Server Certificate
- To Upload a Self-Signed Certificate

Virtual Service

To Create a Virtual Service

Example: Creating a HTTP Service

Request:

```
curl -X POST "http://10.36.73.160:8000/restapi/v3.2/services " -H "Content-Type:
application/json" -u "eyJldCI6IiOGUxNDQ2OTM4ZjRkIn0=\n:" -d
{
"address-version": "IPv4",
"app-id": "app1",
"comments": "This is a demo service",
"ip-address": "10.36.73.154",
"mask": "255.255.255.0",
"name": "service_app1",
"port": "80",
"type": "HTTP"
}
```

Response:

```
{"id":"service_app1","token":"eyJ1c2ZXQiOiIxNjI2MDg1NDc3In0=\n","msg":"Successfully
created."}
```

Example: Creating a HTTPS Service

Request:

```
curl -X POST "http://10.36.73.160:8000/restapi/v3.2/services " -H "Content-Type:
application/json" -u "eyJldCkNjQzNjZiOGUxNDQ2OTM4ZjRkIn0=\n:" -d
' {
```

"address-version": "IPv4",
"app-id": "app1-https",
"comments": "This is a demo service",
"ip-address": "10.36.73.154",
"mask": "255.255.255.0",
"name": "service_app1_secure",
"port": "443",
"certificate": "test_cert",
"type": "HTTPS"
}'

Response:

{"id":"service_app1_secure","token":"eyJldCNlciI6ImFkbWluIn0=\n","msg":"Successfully created."}

To Retrieve a Virtual Service

Example:

Request:

curl -X GET "http://10.36.73.160:8000/restapi/v3.2/services/service_app1" -H "Content-Type: application/json" -u "eyJldCI6IjE2xNDQ2OTM4ZjRkIn0=\n:"

Response:

{
"data": {
"service_app1": {
"Captcha Settings": {
"recaptcha-domain": [

],
"recaptcha-type": "CAPTCHA"
},
"Instant SSL": {
"status": "Off",
"sharepoint-rewrite-support": "Off",
"secure-site-domain": [

]
},

"SSL OCSP": {
"responder-url": "",
"certificate": "",
"enable": "No"
},
"WaaS Account Details": {
"waas-account-id": "0",
"waas-account-serial": "1623473"
},
"Server": {

},
"mask": "255.255.255.255",
"type": "HTTP",
"session-timeout": "60",
"Caching": {
"status": "Off",
"ignore-request-headers": "No",
"cache-negative-response": "No",
"file-extensions": [

],
"ignore-response-headers": "No",
"min-size": "256",
"expiry-age": "60",
"max-size": "256"
},
"Comment Spam": {
"exception-patterns": [

],
"parameter": [

]
},
"ip-address": "10.36.73.154",
"Rule Group": {

},
"vsite": "default",
"status": "On"....."rate-control-pool": "NONE",
"mode": "Passive",
"web-firewall-policy": "default",
"client-ip-addr-header": "",
"trusted-hosts-group": "",
"trusted-hosts-action": "Default"

}
}
},
"object": "Service",
"token": "eyJ1c2VQiOiIxNjI2MDg1ODkyIn0=\n"
}

(The above response is snipped)

To Update a Virtual Service

In this REST API call, the parameters can be passed in a Simple JSON request or a Nested JSON request based on the parameters that needs to be modified. For information on JSON requests, see Request Syntax.

Example:

Request:

curl -X PUT "http://10.36.73.160:8000/restapi/v3.2/services/service_app1 " -H "Content-Type: application/json" -u "eyJldCI6IjGUxNDQ2OTM4ZjRkIn0=\n:" -d '{"enable-access-logs": "Yes"}'

Response:

{"id":"service_app1","msg":"Configuration updated","token":"eyJldCI6YjdiOTgzNjExIn0=\n"}

To Delete a Virtual Service

Example:

Request:

curl -X DELETE "http://10.36.73.160:8000/restapi/v3.2/services/service_app1" -u "eyJwYXNzQiOiIxNjI2MDg5NjcwIn0=\n:"

Response:

{"token":"eyJldCI6IjE2MiI6ImFkbWluIn0=\n","msg":"Successfully deleted."}

Real Server

To Create a Real Server

Example:

Request:

curl -X POST "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/servers " -H "Content-Type: application/json" -u "eyJldCI6IjNDQ2OTM4ZjRkIn0=\n:" -d

'{
"ip-address": "10.1.1.1",
"name": "real_server_1",
"identifier": "IP Address",
"port": "80",
"address-version": "IPv4",
"comments": "Real Server"
}'

Response:

{"msg":"Successfully created.","token":"eyJ1c2VyNjI2MDg3ODE5In0=\n","id":"real_server_1"}

To Retrieve all Real Servers for a Service

Example:

Request:

curl -X GET "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/servers " -u "eyJldCUxNDQ2OTM4ZjRkIn0=\n:"

Response:

{
"data": {
"real_server_1": {
"Out of Band Health Checks": {

"interval": "10",
"enable-oob-health-checks": "Yes"
},
"Load Balancing": {
"backup-server": "No",
"weight": "1"
},
"hostname": "",
"name": "real_server_1",
"ip-address": "10.1.1.1",
"identifier": "IP Address",
"port": "80",
"status": "In Service",
"Advanced Configuration": {
"max-spare-connections": "0",
"timeout": "300000",
"max-establishing-connections": "100",
"max-requests": "1000",
"source-ip-to-connect": "",
"max-keepalive-requests": "0",
"max-connections": "10000",
"client-impersonation": "No"
},
"SSL Policy": {
"enable-ssl-compatibility-mode": "No",
"validate-certificate": "Yes",
"enable-https": "No",
"enable-tls-1": "No",
"enable-ssl-3": "No",
"enable-tls-1-3": "No",
"enable-tls-1-1": "Yes",
"enable-http2-server": "No",
"client-certificate": "",
"enable-sni": "No",
"enable-tls-1-2": "Yes"
},
"Application Layer Health Checks": {
"method": "GET",
"url": "",
"status-code": "200",
"match-content-string": "",
"domain": "",
"additional-headers": [

]
},

"In Band Health Checks": {
"max-timeout-failure": "10",
"max-other-failure": "10",
"max-http-errors": "0",
"max-refused": "10"
},
"Connection Pooling": {
"enable-connection-pooling": "Yes",
"keepalive-timeout": "900000"
},
"address-version": "IPv4",
"resolved-ips": "",
"comments": "Real Server"
}
},
"object": "Server",
"token": "eyJldCI62MTRlZjQ1ZTEzIn0=\n",
"Service": "service_app1"
}

To Update a Real Server

Example:

Request:

curl -X PUT "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/servers/real_server_1 " -H "Content-Type: application/json" -u "eyJldCI6IQTM4ZjRkIn0=\n:" -d '{ "status": "Out of Service Maintenance" }'

Response:

{"token":"eyJldCI6IjEZjQ2YTFkIn0=\n","msg":"Configuration updated","id":"real_server_1"}

To Delete a Real Server

Example:

Request:

curl -X DELETE

"http://10.36.73.160:8000/restapi/v3.2/services/service_app1/servers/real_server_1 " -u "eyJldCI6M4ZjRkIn0=\n:"

Response:

{"token":"eyJ1c2xNjI2MDg4MzAwIn0=\n","msg":"Successfully deleted."}

Content Rule

To Create a Content Rule

Example:

Request:

curl -X POST "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules" -H "Content-Type: application/json" -u "eyJldCI6IjE2OTM4ZjRkIn0=\n:" -d

'{
"comments": "Content Rule 1",
"extended-match": "*",
"mode": "Passive",
"url-match": "/index.html",
"extended-match-sequence": "1",
"name": "Content_rule_1",
"web-firewall-policy": "default",
"host-match": "*"
}'

Response:

{"id":"Content_rule_1","msg":"Successfully created.","token":"eyJldCIBlNTU1NjdiIn0=\n"}

To Retrieve a Content Rule

Example:

Request:

curl -X GET
"http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1" -u
"eyJldCIxN4ZjRkIn0=\n:"

Response:

{
"data": {
"Content_rule_1": {
"extended-match": "*",
"app-id": "Content_rule_1",
"Compression": {
"compress-unknown-content-types": "No",
"status": "Off",
"min-size": "8192",
"content-types": [

]
},
"extended-match-sequence": "1",
"url-match": "/index.html",
"host-match": "*",
"comments": "Content Rule 1",
"web-firewall-policy": "default",
"Logging": {
"access-log": "Enable"
},
"status": "On",
"Advanced Client Analysis": {
"advanced-analysis": "Same as Service"
},
"Rule Group Server count": 0,
"name": "Content_rule_1",
"Caching": {
"cache-negative-responses": "No",
"max-size": "256",
"expiry-age": "60",
"min-size": "256",
"ignore-response-headers": "No",
"file-extensions": [

],
"ignore-request-headers": "No",
"status": "Off"
},

"Captcha Settings": {
"rg-recaptcha-domain": [

],
"recaptcha-type": "CAPTCHA"
},
"mode": "Passive",
"Load Balancing": {
"header-name": "",
"persistence-cookie-name": "persistence",
"persistence-method": "None",
"persistence-idle-timeout": "600",
"failover-method": "Load Balance",
"persistence-cookie-domain": "",
"parameter-name": "",
"source-ip-netmask": "",
"persistence-cookie-path": "",
"lb-algorithm": "Round Robin",
"cookie-age": ""
},
"Rule Group Server": {

}
}
},
"object": "Rule Group",
"token": "eyJ1c2VyIjNjI2MDg4NjY1In0=\n",
"Service": "service_app1"
}

To Update a Content Rule

Example:

Request:

curl -X PUT "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1" -H "Content-Type: application/json" -u "eyJldCI6IjE2DM4ZjRkIn0=\n:" -d '{"mode": "Active"}'

Response:

{"id":"Content_rule_1","msg":"Configuration updated","token":"eyJldCI6IjE2ImFkbWluIn0=\n"}

To Delete a Content Rule

Example:

Request:

curl -X DELETE
"http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1" -u
"eyJldCI6IjEViZTU0ZDg4ZDZlIn0=\n:"

Response:

{"msg":"Successfully deleted.","token":"eyJldCNlciI6ImFkbWluIn0=\n"}

Rule Group Server

To Create a Rule Group Server

Example:

Request:

curl -X POST
"http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1/cont
ent-rule-servers" -H "Content-Type: application/json" -u "eyJldCI6IZDg4ZDZlIn0=\n:" -d

'{
"identifier": "Hostname",
"name": "content_rule_server",
"hostname": "www.cudanet.com",
"port": "80",
"comments": "Server using the hostname identifier"
}'

Response:

{"id":"content_rule_server","token":"eyJldCI6IiMjkzZWQ5In0=\n","msg":"Successfully created."}

To Retrieve a Rule Group Server

Example:

Request:

curl -X GET
"http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1/content-rule-servers" -u "eyJldCI6IDg4ZDZlIn0=\n:"

Response:

{
"Service": "service_app1",
"Rule Group": "Content_rule_1",
"token": "eyJTZjIiwiZXQiOiIxNjI2MDg5MTYyIn0=\n",
"object": "Rule Group Server",
"data": {
"content_rule_server": {
"hostname": "www.cudanet.com",
"Load Balancing": {
"backup-server": "No",
"weight": "1"
},
"Out of Band Health Checks": {
"enable-oob-health-checks": "On",
"interval": "10"
},
"ip-address": "18.144.102.70",
"name": "content_rule_server",
"SSL Policy": {
"validate-certificate": "Yes",
"enable-ssl-compatibility-mode": "No",
"enable-sni": "No",
"enable-tls-1-2": "Yes",
"client-certificate": "",
"enable-ssl-3": "No",
"enable-http2-server": "No",
"enable-tls-1-3": "No",
"enable-tls-1-1": "Yes",
"enable-tls-1": "No",
"enable-https": "Off"
},
"Advanced Configuration": {

"client-impersonation": "No",
"max-keepalive-requests": "0",
"max-connections": "10000",
"source-ip-to-connect": "",
"max-establishing-connections": "100",
"max-requests": "1000",
"timeout": "300000",
"max-spare-connections": "0"
},
"status": "In Service",
"port": "80",
"identifier": "Hostname",
"comments": "Server using the hostname identifier",
"resolved-ips": "",
"address-version": "IPv4",
"In Band Health Checks": {
"max-timeout-failure": "10",
"max-other-failure": "10",
"max-http-errors": "0",
"max-refused": "10"
},
"Connection Pooling": {
"enable-connection-pooling": "Yes",
"keepalive-timeout": "900000"
},
"Application Layer Health Checks": {
"match-content-string": "",
"status-code": "200",
"url": "",
"method": "GET",
"additional-headers": [

],
"domain": ""
}
}
}
}

To Update a Rule Group Server

Example:

Request:

curl -X PUT "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1/content-rule-servers/content_rule_server" -H "Content-Type: application/json" -u "eyJldCI6Ij0ZDg4ZDZlIn0=\n:" -d '{"port": "8888"}'

Response:

{"msg":"Configuration updated","token":"eyJldCI6IzYmI0NzEwMDk2In0=\n","id":"content_rule_server"}

To Delete a Rule Group Server

Example:

Request:

curl -X DELETE "http://10.36.73.160:8000/restapi/v3.2/services/service_app1/content-rules/Content_rule_1/content-rule-servers/content_rule_server" -u "eyJldCI6iZTU0ZDg4ZDZlIn0=\n:"

Response:

{"msg":"Successfully deleted.","token":"eyJldCI6IjDNmZWE0ZDZjIn0=\n"}

Logs

The maximum number of logs that can be fetched at a time using Logs REST API is 1,00,000.

To Fetch Logs

By default, 20 logs are fetched at a time.

Example:

Request:

curl http://10.36.68.52:8000/restapi/v3.2/logs/web-firewall-logs -X GET -u "eyJ3MTMwMDYyIn0=\n:"

Response:

{
"token": "eyJwYXNzd29yZCc5ODM4In0=\n",
"object": "Web Firewall Logs",
"data": [
{
"client-ip": "99.66.160.6",
"severity": "Alert",
"proxy-port": 38008,
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"date": "2021-07-27",
"client-port": 38008,
"country-code": "US",
"session-id": "",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"service-port": 80,
"client-type": "Unknown",
"url": "/",
"service-app-id": "rule1",
"id": "17ae6f29352-f061ec01",
"attack-detail": "Key=\"username\" Length=\"8\"",
"host": "99.66.160.6",
"attack": "Max Key Length Exceeded",
"risk-score": 120,
"method": "POST",
"protocol": "HTTP",
"service-name": "service1",
"query-string": "\"-\"",
"rule": "service1:host_test",
"follow-up-action": "None",
"rule-type": "JSON Profile",
"service-ip": "99.66.160.6",
"client-risk-score": 0,
"authenticated-user": "",
"proxy-ip": "99.66.160.6",
"referer": "",

"time": "00:51:35.29",
"attack-category": "JSON Violations"
},
{
"service-ip": "99.66.160.6",
"rule-type": "JSON Profile",
"follow-up-action": "None",
"rule": "service1:host_test",
"service-name": "",
"query-string": "\"-\"",
"time": "00:51:35.29",
"referer": "",
"attack-category": "JSON Violations",
"proxy-ip": "99.66.160.6",
"authenticated-user": "",
"client-risk-score": 0,
"service-app-id": "rule1",
"url": "/",
"service-port": 80,
"client-type": "Unknown",
"date": "2021-07-27",
"client-port": 38008,
"country-code": "US",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"session-id": "",
"action": "LOG",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"severity": "Alert",
"proxy-port": 38008,
"client-ip": "99.66.160.6",
"method": "POST",
"protocol": "HTTP",
"risk-score": 180,
"host": "99.66.160.6",
"attack": "Max Value Length Exceeded",
"attack-detail": "Key=\"username\" Length=\"4\"",
"id": "17ae6f29352-f061ec01"
}
]
}

To Fetch a Log using the Log ID

Example:

Request:

curl http://10.36.68.52:8000/restapi/v3.2/logs/web-firewall-logs/17abee19ff6-12679b22 -X GET -u"eyJ1c2Vy3MTMwMDYyIn0=\n:"

Response:


```
{
"token": "eyJ1cYTY3NzYwIn0=\n",
"object": "Web Firewall Logs",
"data": [
{
"id": "17abee19ff6-12679b22",
"attack-detail": "Key=\"username\" Length=\"8\"",
"attack": "Max Key Length Exceeded",
"host": "99.66.160.6",
"risk-score": 120,
"method": "POST",
"protocol": "HTTP",
"client-ip": "99.66.160.6",
"proxy-port": 41726,
"severity": "Alert",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "DENY",
"date": "2021-07-19",
"client-port": 41726,
"country-code": "US",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"session-id": "",
"service-port": 80,
"client-type": "Unknown",
"service-app-id": "service1",
"url": "/",
"client-risk-score": 0,
"authenticated-user": "",
"proxy-ip": "99.66.160.6",
"time": "06:08:15.480",
"referer": "",
"attack-category": "JSON Violations",
"service-name": "service1",
"query-string": "\"-\"",
"rule": "service1:reco_6666cd76f96956469e7be39d750cc7d9_post",
"follow-up-action": "None",
```

"rule-type": "JSON Profile",
"service-ip": "99.66.160.6"
}
]
}

To Fetch a Parameter Value from a Log

Example:

Request:

curl http://10.36.68.52:8000/restapi/v3.2/logs/web-firewall-logs?parameters=service-name,time,date -X GET -u"eyJ1c2ODI3MTMwMDYyIn0=\n:"

Response:

{
"token": "eyJ1c2VzkzYTY3NzYwIn0=\n",

"object": "Web Firewall Logs",
"data": [
{
"date": "2021-07-27",
"service-name": "service1",
"time": "00:51:35.29"
},
{
"date": "2021-07-27",
"service-name": "",
"time": "00:51:35.29"
}
]
}

To Apply a Filter on Logs

Example:

Request:

curl http://10.36.68.52:8000/restapi/v3.2/logs/web-firewall-logs?filters=[{"field":"client-ip", "operator":"is equal to", "values":["99.66.160.6"]}] -X GET -u"eyJ1cODI3MTMwMDYyIn0=\n:"

Response:

{
"token": "eyJ1c2VyI3NzYwIn0=\n",
"object": "Web Firewall Logs",
"data": [
{
"service-ip": "99.66.160.6",
"follow-up-action": "None",
"rule-type": "JSON Profile",
"query-string": "\"-\"",
"service-name": "service1",
"rule": "service1:host_test",
"attack-category": "JSON Violations",
"time": "00:51:35.29",
"referer": "",
"proxy-ip": "99.66.160.6",
"client-risk-score": 0,
"authenticated-user": "",
"service-app-id": "rule1",
"url": "/",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"country-code": "US",
"session-id": "",
"client-port": 38008,
"date": "2021-07-27",
"client-type": "Unknown",
"service-port": 80,
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"client-ip": "99.66.160.6",
"proxy-port": 38008,
"severity": "Alert",
"protocol": "HTTP",
"method": "POST",
"risk-score": 120,
"attack-detail": "Key=\"username\" Length=\"8\"",
"attack": "Max Key Length Exceeded",
"host": "99.66.160.6",
"id": "17ae6f29352-f061ec01"
},
{

"risk-score": 180,
"protocol": "HTTP",
"method": "POST",
"id": "17ae6f29352-f061ec01",
"attack-detail": "Key=\"username\" Length=\"4\"",
"attack": "Max Value Length Exceeded",
"host": "99.66.160.6",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"country-code": "US",
"session-id": "",
"client-port": 38008,
"date": "2021-07-27",
"client-type": "Unknown",
"service-port": 80,
"url": "/",
"service-app-id": "rule1",
"client-ip": "99.66.160.6",
"severity": "Alert",
"proxy-port": 38008,
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"attack-category": "JSON Violations",
"time": "00:51:35.29",
"referer": "",
"client-risk-score": 0,
"authenticated-user": "",
"proxy-ip": "99.66.160.6",
"service-ip": "99.66.160.6",
"query-string": "\"-\"",
"service-name": "",
"rule": "service1:host_test",
"follow-up-action": "None",
"rule-type": "JSON Profile"
}
]
}

To Apply Multiple Filters on Logs

Example:

Request:

curl http://10.36.68.52:8000/restapi/v3.2/logs/web-firewall-logs?filters=[{"field":"client-ip",

"operator":"is equal to", "values":["99.66.160.6"]}, {"field":"attack-category", "operator": "in (comma-separated)", "values": ["Limits Violation","XSS Injections"]}] -X GET -u "eyJ1DI3MTMwMDYyIn0=\n:"

Response:

{
"object": "Web Firewall Logs",
"data": [
{
"id": "17ad24ea025-b16e7850",
"attack-detail": "type=\"cross-site-scripting\" pattern=\"script-tag\" token=\"<script>\" header=\"test\" value=\"<script></script>\"",
"host": "99.66.160.6",
"attack": "Cross-Site Scripting in Header",
"risk-score": 140,
"method": "GET",
"protocol": "HTTP",
"client-ip": "99.66.160.6",
"proxy-port": 39710,
"severity": "Alert",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "DENY",
"date": "2021-07-23",
"client-port": 39710,
"session-id": "",
"country-code": "US",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"service-port": 80,
"client-type": "Unknown",
"service-app-id": "service1",
"url": "/",
"client-risk-score": 0,
"authenticated-user": "",
"proxy-ip": "99.66.160.6",
"referer": "",
"time": "00:40:06.54",
"attack-category": "XSS Injections",
"service-name": "service1",
"query-string": "\"-\"",
"rule": "service1:star-acl",
"follow-up-action": "None",
"rule-type": "Header ACL",
"service-ip": "99.66.160.6"
},

{
"url": "/",
"service-app-id": "service1",
"client-type": "Unknown",
"service-port": 80,
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"session-id": "",
"country-code": "US",
"date": "2021-07-19",
"client-port": 59208,
"action": "LOG",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"proxy-port": 59208,
"severity": "Alert",
"client-ip": "99.66.160.6",
"protocol": "HTTP",
"method": "GET",
"risk-score": 140,
"attack": "Cross-Site Scripting in Parameter",
"host": "99.66.160.6",
"attack-detail": "type=\"cross-site-scripting\" pattern=\"script-tag\" token=\"<script>\"
Parameter=\"id\" value=\"<script>alert(hi)</script>\"",
"id": "17abdef8c11-126bdc27",
"service-ip": "99.66.160.6",
"rule-type": "Global",
"follow-up-action": "None",
"rule": "security-policy",
"query-string": "id=<script>alert(hi)</script>",
"service-name": "",
"attack-category": "XSS Injections",
"referer": "",
"time": "01:43:50.676",
"proxy-ip": "99.66.160.6",
"client-risk-score": 0,
"authenticated-user": ""
},
{
"risk-score": 140,
"protocol": "HTTP",
"method": "GET",
"id": "17aaa07aabf-9997b68",
"attack-detail": "type=\"cross-site-scripting\" pattern=\"script-tag\" token=\"<script>\"
Parameter=\"id\" value=\"<script>alert(hi)</script>\"",
"attack": "Cross-Site Scripting in Parameter",
"host": "99.66.160.6",
"country-code": "US",

"session-id": "",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"client-port": 40418,
"date": "2021-07-15",
"client-type": "Unknown",
"service-port": 80,
"url": "/",
"service-app-id": "service1",
"client-ip": "99.66.160.6",
"severity": "Alert",
"proxy-port": 40418,
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"attack-category": "XSS Injections",
"referer": "",
"time": "04:57:47.71",
"authenticated-user": "",
"client-risk-score": 0,
"proxy-ip": "99.66.160.6",
"service-ip": "99.66.160.6",
"query-string": "id=<script>alert(hi)</script>",
"service-name": "service1",
"rule": "security-policy",
"follow-up-action": "None",
"rule-type": "Global"
},
{
"risk-score": 140,
"protocol": "HTTP",
"method": "GET",
"id": "17aaa078717-1098b55d",
"attack-detail": "type=\"cross-site-scripting\" pattern=\"script-tag\" token=\"<script>\"
Parameter=\"id\" value=\"<script>alert(hi)</script>\"",
"host": "99.66.160.6",
"attack": "Cross-Site Scripting in Parameter",
"session-id": "",
"country-code": "US",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"date": "2021-07-15",
"client-port": 40406,
"client-type": "Unknown",
"service-port": 80,
"service-app-id": "service1",
"url": "/",
"client-ip": "99.66.160.6",
"proxy-port": 40406,

"severity": "Alert",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"attack-category": "XSS Injections",
"referer": "",
"time": "04:57:37.947",
"authenticated-user": "",
"client-risk-score": 0,
"proxy-ip": "99.66.160.6",
"service-ip": "99.66.160.6",
"query-string": "id=<script>alert(hi)</script>",
"service-name": "",
"rule": "security-policy",
"follow-up-action": "None",
"rule-type": "Global"
}
],
"token": "eyJwYXNzd29MzgxMTA2In0=\n"
}

To Apply the Limit and Offset

Example:

Request:

curl http://10.36.68.52:8000/restapi/v3.2/logs/web-firewall-logs?limit=4&offset=2 -X GET -u "eyJ1c2VyIjTMwMDYyIn0=\n:"

Response:

{
"data": [
{
"risk-score": 120,
"method": "POST",
"protocol": "HTTP",
"id": "17ae3bb7701-e2614918",
"attack-detail": "Key=\"username\" Length=\"8\"",
"host": "99.66.160.6",
"attack": "Max Key Length Exceeded",
"date": "2021-07-26",
"client-port": 45912,

"session-id": "",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"country-code": "US",
"service-port": 80,
"client-type": "Unknown",
"url": "/",
"service-app-id": "rule1",
"client-ip": "99.66.160.6",
"severity": "Alert",
"proxy-port": 45912,
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"referer": "",
"time": "09:52:31.619",
"attack-category": "JSON Violations",
"client-risk-score": 0,
"authenticated-user": "",
"proxy-ip": "99.66.160.6",
"service-ip": "99.66.160.6",
"service-name": "service1",
"query-string": "\"-\"",
"rule": "service1:host_test",
"follow-up-action": "None",
"rule-type": "JSON Profile"
},
{
"time": "09:52:31.619",
"referer": "",
"attack-category": "JSON Violations",
"proxy-ip": "99.66.160.6",
"authenticated-user": "",
"client-risk-score": 0,
"service-ip": "99.66.160.6",
"follow-up-action": "None",
"rule-type": "JSON Profile",
"service-name": "",
"query-string": "\"-\"",
"rule": "service1:host_test",
"method": "POST",
"protocol": "HTTP",
"risk-score": 180,
"attack-detail": "Key=\"username\" Length=\"4\"",
"attack": "Max Value Length Exceeded",
"host": "99.66.160.6",
"id": "17ae3bb7701-e2614918",
"service-app-id": "rule1",

"url": "/",
"client-port": 45912,
"date": "2021-07-26",
"session-id": "",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"country-code": "US",
"service-port": 80,
"client-type": "Unknown",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"client-ip": "99.66.160.6",
"proxy-port": 45912,
"severity": "Alert"
},
{
"service-ip": "99.66.160.6",
"rule": "service1:host_test",
"service-name": "service1",
"query-string": "\"-\"",
"rule-type": "JSON Profile",
"follow-up-action": "None",
"referer": "",
"time": "07:37:23.999",
"attack-category": "JSON Violations",
"client-risk-score": 0,
"authenticated-user": "",
"proxy-ip": "99.66.160.6",
"service-port": 80,
"client-type": "Unknown",
"client-port": 38612,
"date": "2021-07-26",
"session-id": "",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"country-code": "US",
"url": "/",
"service-app-id": "rule1",
"proxy-port": 38612,
"severity": "Alert",
"client-ip": "99.66.160.6",
"action": "LOG",
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"risk-score": 120,
"method": "POST",
"protocol": "HTTP",
"id": "17ae33fc09e-10608df0",
"attack": "Max Key Length Exceeded",

"host": "99.66.160.6",
"attack-detail": "Key=\"username\" Length=\"8\""
},
{
"client-ip": "99.66.160.6",
"severity": "Alert",
"proxy-port": 38550,
"user-agent": "curl/7.15.3 (i686-pc-linux-gnu) libcurl/7.15.3 OpenSSL/0.9.8c zlib/1.2.7",
"action": "LOG",
"country-code": "US",
"session-id": "",
"client-fingerprint": "g_bcec223e740c7b5424b26a01c028ef91",
"date": "2021-07-26",
"client-port": 38550,
"client-type": "Unknown",
"service-port": 80,
"url": "/",
"service-app-id": "rule1",
"id": "17ae33ea5a7-d677625",
"attack-detail": "Key=\"username\" Length=\"8\"",
"host": "99.66.160.6",
"attack": "Max Key Length Exceeded",
"risk-score": 120,
"protocol": "HTTP",
"method": "POST",
"query-string": "\"-\"",
"service-name": "",
"rule": "service1:host_test",
"follow-up-action": "None",
"rule-type": "JSON Profile",
"service-ip": "99.66.160.6",
"authenticated-user": "",
"client-risk-score": 0,
"proxy-ip": "99.66.160.6",
"attack-category": "JSON Violations",
"time": "07:36:11.562",
"referer": ""
}
],
"object": "Web Firewall Logs",
"token": "eyJwYXNz3MzgxNjI1In0=\n"
}

Certificates

When uploading a certificate, the certificate and key must be converted to a Base64-encoded format to use it as a JSON value in v3.1 API.

To Upload a Trusted CA Certificate

Example

Request:

curl -X POST "http://10.36.73.160:8000/restapi/v3.2/certificates/trusted-ca-certificate " -H "accept: application/json" -u "eyJldCI6NGEwYzNiIn0=\n:barracuda" -H "Content-Type: application/json" -d '{ "certificate": "LS0tLSNSEF4Q3pBSk", "name": "TEST" }'

Response:

{"token":"eyJ1c2DU5NGNlIn0=\n","id":"TEST","msg":"Successfully created."}

To Upload a Trusted Server Certificate

Example:

Request:

curl -X POST "http://10.36.73.160:8000/restapi/v3.2/certificates/trusted-server-certificate" -H "accept: application/json" -u "eyJldCI6NGEwYzNiIn0=\n:barracuda" -H "Content-Type: application/json" -d '{ "name": "TRUSTEDSERVER", "certificate": "LS0tLS1VUFNSEF4Q3pBSk" }'

Response:

{"token":"eyJ1c2MDU5NGNlIn0=\n","id":"TRUSTEDSERVER","msg":"Successfully created."}

To Upload a Self-Signed Certificate

Example:

Request:

curl -X POST "http://10.36.73.160:8000/restapi/v3.2/certificates/signed-certificate " -H "accept: application/json" -u "eyJldCI6NGEwYzNiIn0=\n:barracuda" -H "Content-Type: application/json" -d '{ "certificate-password": "LS0tLS1VUFNSEF4Q3pBSk", "cert-type": "signed", "schedule-renewal-day": "15 days", "key-type": "RSA", "assign-associated-key": "", "signed-certificate": "LS0tLS1NSEF4Q3pBSk", "intermediary-certificates": [LS0tLS1NSEF4Q3pBSk], "auto-renew-cert": "Yes", "name": "TESTUPLOAD", "certificate-type": "PEM Certificate", "certificate-key": "LS0tLSNSEF4Q3pBSk", "allow-private-key-export": "Yes" }'

Response:

{"token":"eyJ1c2MDU5NGNlIn0=\n","id":"TESTUPLOAD","msg":"Successfully created."}