



Infoblox Threat Intelligence Data Exchange Documentation

[API Getting Started Guide](#)

06/29/2021

Table of Contents

Overview	3
Credentials	4
API Access	4
User Interface Access	4
Querying Data.....	5
Query Active Threats	5
Query Data for a Single Threat Indicator	8
Query for a Specific IP Address	8
Query for a Specific Host Name.....	9
Query for a Specific URL	9
Query Data from a Particular Time	10
Get Threats for Time Period	10
Retrieve a List of Provider Organizations	12
Query Data from a Specified Organization.....	13
Threat Class APIs	16
Appendix A: Common HTTP Status Codes	17

Overview

This Getting Started Guide takes you through the steps required to query data via the Cloud Services Portal (CSP).

See the *Cloud Services Portal API Guide* for detailed information on the API calls and terms used in this document.

Credentials

API Access

The issuance of an API key is required to query and submit data to the platform via API. The API key is passed in the username field and is used for authentication. The password field should be set to an empty string.

Example of a Cloud Services Portal API token: <MY TOKEN>

User Interface Access

The issuance of a username and password is required to access the Cloud Services Portal (CSP) user interface at <https://csp.infoblox.com>. User interface access is not required for API access, but certain functions, including the creation of data profiles, are simpler to do with the user interface.

Example of Cloud Services Portal credentials:

Username: user@company.com

Password: *dk5seOg3TW46

Querying Data

These scenarios make use of multiple API calls to query available data or information, based on the parameters passed.

Query Active Threats

To access active threats available to your organization, use `tide/api/data/threats/state/`. If you don't specify a provider organization (using the "profile" query string parameter) then the search will be executed against all available data. You can specify multiple provider organizations by having multiple "profile" parameters.

To make samples a bit easier to use, the calls also specify the "rlimit" query string parameter. It's an optional parameter that limits the number of returned records.

Python

```
#note: install the 'requests' library first:
#pip install -U requests
import requests
from pprint import pprint

#note: replace this api_key value with your api key!
api_key = 'YOUR_API_KEY'
api_endpoint = 'https://csp.infoblox.com'
api_path = '/tide/api/data/threats/state'
url = '%s%s' % (api_endpoint, api_path)
params = {'rlimit': 2}(optional)

token = '<MY TOKEN>'

r = requests.get(url, headers={'Content-Type': 'application/json', 'Authorization': 'token {}'.format(token)})
print (r.status_code)
print (r.json())
# OR
#print (r.content)
```

Sample result

```
200

{u'dropped': False,
 u'dropped_record_count': 0,
 u'filtered_record_count': 2,
 u'record_count': 2,
 u'threat': [{u'batch_id': u'ffffffff-f343-11e3-897d-55530a829c6f',
              u'class': u'Exploit_Kit',
              u'detected': u'2017-06-13T15:42:06.000Z',
              u'dga': u'false',
```

```
u'domain': u'bomunykedafppw.info',
u'host': u'8uub.bomunykedafppw.info',

u'id': u'ffffffff-f342-11e3-897c-55530a829c6f',
u'imported': u'2017-06-13T21:42:54.429Z',
u'ip': u'',
u'origin': u'IID',
u'profile': u'IID',
u'property': u'Exploit_Kit_Angler',
u'target': u'',
u'threat_level': 1,
u'tld': u'info',
u'tlp': u'',
u'type': u'HOST',
u'up': u'true',
u'url': u''},
{u'batch_id': u'ffffffff-0c5b-11e4-913b-fb8aa419fdb',
u'class': u'Spam_Bot',
u'detected': u'2017-07-15T10:36:44.000Z',
u'domain': u'',
u'host': u'',
u'id': u'ffffffff-0c5b-11e4-913b-fb8aa419fdff',
u'imported': u'2017-07-15T20:06:57.174Z',
u'ip': u'1.26.31.136',
u'origin': u'OrgA',
u'profile': u'OrgA',
u'property': u'Bot Cutwail',
u'target': u'',
u'threat_level': 1,
u'tld': u'',
u'tlp': u'',
u'type': u'IP',
u'url': u''}}]
```

Curl

```
curl
`https://csp.infoblox.com/tide/api/data/threats/state?profile=EmergingThreats:Hos
tnames_Feed&class=APT,Bot&type=host&show_full_profiles=true&data_format=ndjson' -
H 'Authorization:Token token=<MY TOKEN>' | python -mjson.tool
```

Sample result

```
{
  "threat": [
    {
      "batch_id": "ffffffff-f343-11e3-897c-55530a829c6f",
      "class": "Exploit_Kit",
      "detected": "2017-06-13T17:24:26.000Z",
      "dga": "false",
      "domain": "real-bad-host.info",
      "host": "drawer.real-bad-host.info",
      "id": "ffffffff-f343-11e3-897c-55530a829cf6",
      "imported": "2017-06-13T21:42:54.429Z",
      "ip": "",
      "origin": "IID",
      "profile": "IID",
      "property": "Exploit_Kit_Nuclear",
      "target": "",
      "threat_level": 1,
      "tld": "info",
      "tlp": "",
      "type": "HOST",
      "up": "true",
      "url": ""
    },
    {
      "batch_id": "ffffffff-0c5a-11e4-913b-fb8aa419fdb",
      "class": "Spam_Bot",
      "detected": "2017-07-15T10:36:44.000Z",
      "domain": "",
      "host": "",
      "id": "ffffffff-0c5b-11d4-913b-fb8aa419fdb",
      "imported": "2017-07-15T20:06:57.174Z",
      "ip": "1.55.122.11",
      "origin": "OrgA",
      "profile": "OrgA",
      "property": "Bot Cutwail",
      "target": "",
      "threat_level": 1,
      "tld": "",
      "tlp": "",
      "type": "IP",
      "url": ""
    }
  ],
}
```

```
    "record_count": 2,  
    "filtered_record_count": 2,  
    "dropped_record_count": 0,  
    "dropped": false  
}
```

Query Data for a Single Threat Indicator

You can query the platform to retrieve all data for a single threat indicator, such as a particular IP address, host name, or URL.

The following requests will return threat records like the ones in the [Query Active Threats](#) example.

Query for a Specific IP Address

If you wanted to search for all instances of IP address 1.2.3.4 in csv format, you could submit the following curl request:

```
curl  
'https://csp.infoblox.com/tide/api/data/threats/state?type=ip&ip=1.2.3.4&data_format=csv' -H 'Authorization:Token token=<MY TOKEN>'
```

If you wanted to search for all instances of IP address 1.2.3.4 detected in the last day in csv format, you could submit the following curl request:

```
curl  
'https://csp.infoblox.com/tide/api/data/threats/state?type=ip&ip=1.2.3.4&period=1d&data_format=csv' -H 'Authorization:Token token=<MY TOKEN>'
```

If you wanted to search for all instances of IP address 1.2.3.4 which were reported as Zero Access Bots in csv format, you could submit the following curl request:

```
curl  
'https://csp.infoblox.com/tide/api/data/threats/state?type=ip&ip=1.2.3.4&property=bot_zeroaccess&data_format=csv' -H 'Authorization:Token token=<MY TOKEN>'
```

(A list of valid properties can be found at the API [/api/data/properties](#).)

Query for a Specific Host Name

If you wanted to search for all instances of host `example.com` in csv format, you could submit the following curl request:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/state?type=host&host=example.com
&data_format=csv' -H 'Authorization:Token
token=<MY TOKEN>'
```

If you wanted to search for all instances of host `example.com` imported in the last hour in csv format, you could submit the following curl request:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/state?type=host&host=example.com
&imported_period=1h&data_format=csv' -H 'Authorization:Token
token=<MY TOKEN>'
```

If you wanted to search for all instances of host `example.com` for threat class `Malware C2` in csv format, you could submit the following curl request

```
curl
'https://csp.infoblox.com/tide/api/data/threats/state?type=host&host=example.com
&class=Malware_C2&data_format=csv' -H 'Authorization:Token
token=<MY TOKEN>'
```

(A list of valid threat classes can be found at the API `/api/data/threat_classes`.)

Query for a Specific URL

If you wanted to search for all instances of URL `http://www.example.com`, you could submit the following curl request:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/state?type=url&url=http://ww
w.example.com&data_format=csv' -H 'Authorization:Token
token=<MY TOKEN>'
```

If you wanted to search for all instances of URL `http://www.example.com` detected since August 1, 2017 UTC, you could submit the following curl request:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/state?type=url&url=http://www
.example.com&from_date=2017-08-01T00:00:00Z&data_format=csv' -H
'Authorization:Token token=<MY TOKEN>'
```

If you wanted to search for all instances of URL `http://www.example.com` detected since August 1, 2017 UTC and targeting your company, you could submit the following curl request:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/state?type=url&url=http://www
.example.com&from_date=2017-08-
01T00:00:00Z&target=my%20company&data_format=csv' -H 'Authorization:Token
token=<MY TOKEN>'
```

Query Data from a Particular Time

You may want to regularly retrieve data with the same criteria. In this case, you'll probably want to query by time period.

For example, you want to check for host MalwareC2DGA threats every hour. You might create a cron job that submits:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/host/hourly?imported_period=1h&
class=MalwareC2DGA&data_format=csv' -H 'Authorization:Token
token=<MY TOKEN>'
```

Or you could save the date/time of your last retrieval and use it with `imported_from_date` and the suitable time period:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/host/daily?imported_from
_date=[last retrieval]&class=MalwareC2DGA&data_format=csv' -H
'Authorization:Token token=<MY TOKEN>'
```

Get Threats for Time Period

Returns threats submitted within the specified time period. Valid time periods are recent (30 minutes), hourly (90 minutes), daily (25 hours), weekly (7 days), and monthly (30 days).

Request

Request Endpoint

GET /data/threats/{type}

Request Body

N/A

Path Parameters

Parameter	Value	Data Type	Description
type	host, ip, or url	string	Type of threats to return
age	recent, hourly, daily, weekly, monthly	string	The age of threats to return. recent = 30 minutes, hourly = 90 minutes, daily = 25 hours, weekly = 7 days, monthly = 30 days

Query Parameters

Response

If the submission is successful, the HTTP code 200 (OK) will be returned with the list of Threat objects.

Example

Request using curl to return host records for the past day:

```
curl
'https://csp.infoblox.com/tide/api/data/threats/host/daily?data_format=ndjson' -H 'Authorization:Token
token=<MY TOKEN>'
```

Response:

```
{
  "threat": [
    {
      "id": "c2fe7b4b-1434-11e4-88e7-47366fc6a030",
      "type": "HOST",
      "host": "example.com",
      "domain": "example.com",
      "tld": "com",
      "profile": "IID",
      "origin": "IID",
      "property": "MalwareC2_Torpig",
      "class": "MalwareC2",
      "threat_level": 100,
      "detected": "2016-07-25T19:49:17.023Z",
      "imported": "2016-07-25T19:49:17.023Z",
      "dga": "false",
      "batch_id": "c2f9e76a-1334-11e4-88e7-47366fc6a010"
    }
  ],
}
```

Retrieve a List of Provider Organizations

Use `/api/admin/sharing/source_orgs` as follows to get the ID of the organizations providing data that is available to your organization.

Python

```
#note: install the 'requests' library first:
#pip install -U requests
import requests

#note: replace this api_key value with your api key!
api_key = 'YOUR_API_KEY'
api_endpoint = 'https://csp.infoblox.com'
api_path =
'/tide/admin/v1/resources/shared/dataprofiles'
url = '%s%s' % (api_endpoint,api_path)

token = '<MY TOKEN>'

r = requests.get(url,headers={'Content-
Type':'application/json','Authorization':'token {}'.format(token)})
print (r.status_code)
print (r.json())
# OR
#print (r.content)
```

Sample result

```
200

{u'status': u'success', u'code': 0, u'data': [u'IID']}
```

Curl

```
curl
'https://csp.infoblox.com/tide/admin/v1/resources/shared/dat
aprofiles' -H 'Authorization:Token
token=<MY TOKEN>'
| python -mjson.tool
```

Sample result

```
{
  "code": 0,
  "data": [
    "OrgA",
    "OrgB",
    "DemoOrg",
    "IID"
  ],
  "status": "success"
}
```

Query Data from a Specified Organization

To query data from a specified organization, use the state (active threats) or threats by age API.

For example, if you wanted to get a list of all active IP threats from OrgA, you would use the API:

```
/tide/api/data/threats/state/ip?profile=OrgA
```

If you wanted all IP threats submitted by OrgA in the last day, you would use the API:

```
/tide/api/data/threats/state/ip/daily?profile=OrgA
```

You must specify the name of the provider organization using the "profile" query string parameter. You can specify multiple provider organizations by having multiple "profile" parameters.

Python

```
#note: install the 'requests' library first:
#pip install -U requests
import requests
from pprint import pprint

#note: replace this api_key value with your api key!
api_key = 'YOUR_API_KEY'
api_endpoint = 'https://csp.infoblox.com'
api_path = '/tide/api/data/threats/ip/daily'
url = '%s%s' % (api_endpoint, api_path)
params = {'profile': ['OrgA', 'IID'], 'rlimit': 2}

token = '<MY TOKEN>'

r = requests.get(url, headers={'Content-Type': 'application/json', 'Authorization': 'token {}'.format(token)})
print (r.status_code)
print (r.json())
# OR
#print (r.content)
```

Sample result

```
200

{u'dropped': False,
 u'dropped_record_count': 0,
 u'filtered_record_count': 2,
 u'record_count': 2,
 u'threat': [{u'batch_id': u'fefefefe-f343-11e3-897c-55530a829c6f',
              u'class': u'ExploitKit',
              u'detected': u'2017-06-13T17:24:26.000Z',
              u'dga': u'false',
              u'domain': u'another-bad-host.info',
              u'host': u'drawer.another-bad-host.info',
              u'id': u'fefefefe-f343-11e3-fefe-55530a829c6f',
              u'imported': u'2017-06-13T21:42:54.429Z',
```

```

    u'ip': u'',
    u'origin': u'',
    u'profile': u'OrgA',
    u'property': u'ExploitKit_Nuclear',
    u'target': u'',
    u'threat_level': 100,
    u'tld': u'info',
    u'tlp': u'',
    u'type': u'HOST',
    u'up': u'true',
    u'url': u''},
  {u'batch_id': u'ad1798f7-fefe-11e3-fefe-55530a829c6f',
    u'class': u'ExploitKit',
    u'detected': u'2017-06-13T17:24:26.000Z',
    u'dga': u'false',
    u'domain': u'programrealty.info',
    u'host': u'draw.programrealty.info',
    u'id': u'ad257baa-f343-11e3-897c-fefefefefefe',
    u'imported': u'2017-06-13T21:42:54.429Z',
    u'ip': u'',
    u'origin': u'IID',
    u'profile': u'IID',
    u'property': u'ExploitKit_Nuclear',
    u'target': u'',
    u'threat_level': 100,
    u'tld': u'info',
    u'tlp': u'',
    u'type': u'HOST',
    u'up': u'true',
    u'url': u''}}}]

```

Curl

```

curl
'https://csp.infoblox.com/tide/api/data/threats?profile=OrgB&profile=IID&rlimit=2' -H 'Authorization:Token token=<MY TOKEN>' | python -
mjson.tool

```

Sample result

```

{
  "threat": [
    {
      "id": "ad257ba9-f343-11e3-897c-55530a829c6f",
      "type": "HOST",
      "host": "drawer.programrealty.info",
      "ip": "",
      "url": "",
      "domain": "programrealty.info",
      "tld": "info",
      "profile": "IID"
    }
  ]
}

```

```
"origin": "IID",
"property": "ExploitKit_Nuclear",
"class": "ExploitKit",
"threat_level": 100,
"target": "",
"detected": "2017-06-03T17:24:26.000Z",
"imported": "2017-06-13T21:42:54.429Z",
"dga": "false",
"up": "true",
"tlp": "",
"batch_id": "ad1798f7-f343-11e3-897c-55530a829c6f"
},
{
  "id": "ad257baa-f343-11e3-897c-55530a829c6f",
  "type": "HOST",
  "host": "draw.programreality.info",
  "ip": "",
  "url": "",
  "domain": "programreality.info",
  "tld": "info",
  "profile": "IID",
  "origin": "IID",
  "property": "ExploitKit_Nuclear",
  "class": "ExploitKit",
  "threat_level": 100,
  "target": "",
  "detected": "2017-06-03T17:24:26.000Z",
  "imported": "2017-06-13T21:42:54.429Z",
  "dga": "false",
  "up": "true",
  "tlp": "",
  "batch_id": "ad1798f7-f343-11e3-897c-55530a829c6f"
}
],
"record_count": 2,
"filtered_record_count": 2,
"dropped_record_count": 0,
"dropped": false
}
```

Threat Class APIs

Threat classes indicate the categories of threat, for example, phishing or spambots.

Get Threat Classes

Gets a list of threat classes.

Request

GET /data/threat_classes

Example

Request using curl:

```
curl
'https://csp.infoblox.com/tide/api/data/threat_classes' -H
'Authorization:Token token=<MY TOKEN>'
| python -mjson.tool
```

Response (with some detail removed for brevity):

```
{
  "threat_class" : [ {
    "link" : [ {
      "href" : "/data/threat_classes/MalwareDownload",
      "rel" : "self"
    } ],
    "id" : "MalwareDownload",
    "name" : "Malware Download"
  }, {
    "link" : [ {
      "href" : "/data/threat_classes/Spambot",
      "rel" : "self"
    } ],
    "id" : "Spambot",
    "name" : "Spambot"
  }, {
    "link" : [ {
      "href" : "/data/threat_classes/ExploitKit",
      "rel" : "self"
    } ],
    "id" : "ExploitKit",
    "name" : "Exploit Kit"
  }, {
    ...
  } ]
}
```


Appendix A: Common HTTP Status Codes

The 2xx range is returned for successful requests.

The 4xx range is returned due to errors made by the requestor.

The 5xx range is returned due to server errors.

The following is not an exhaustive list, but representative of the most likely codes returned by the platform.

Code	Reason	Description
200	OK	The request succeeded.
201	Created	The server created a new item, for example, a new threat batch. The server will generally return a Location URI in the response header indicating the location of the newly created item.
204	No Content	No content needed to be returned from the server, for example, if an entity has been deleted.
400	Bad Request	There was an error in the request due to bad syntax. There may be, for example, errors in the query parameters or in the body of the request. Check your syntax against the API documentation.
401	Unauthorized	The user has not submitted valid credentials. Make sure you are using the proper API key in your transmission.
403	Forbidden	The user does not have access to the requested resource.
404	Not Found	The server did not find a resource matching the request URI.
500	Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
503	Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.