PassiveDNS Integration Guide

Are you looking to use the public PassiveDNS database API? Please see the simplified integration guide for external users.

Detailed API documentation

The Swagger API documentation is always up-to-date and lets you try out any query with your user session or an API-key.

Integration guide

Simple query

To perform a simple PDNS query, use

```
https://api.mnemonic.no/pdns/v3/<query>?parameters
```

where "<query>" is a DNS query or answer string to lookup, for example

```
https://api.mnemonic.no/pdns/v3/cnn.com?rrType=A
```

Query parameters

There are multiple query parameters that can be passed along:

Parameter	Possible values	Default value	Description
limit	[0-maxint]	25	Limit the number of returned values. The default value of .25 is set to avoid that a queried record with a huge resultset is streamed back to the client unecessarily. A value of "0" means that the client explicitly requests an unlimited resultset. The result will return a 412 error code if the limit is set to be more than the server allows. This is 10 000 for authenticated users, while public use has its limit set to 1000.
offset	[0-maxint]	0	Skip the initial <offset> number of values in the resultset. Together with the parameter "limit", this allows a client to perform pagination of the results.
rrClass	Any DNS record class	<any>	The most probable record class is IN. By default, this field is not filtered.
rrType	Any DNS record type	<any>	The most used record classes are A, AAAA, PTR, CNAME and MX, but any supported DNS record name can be used here. By default, this field is not filtered.
The following options are only meaningful for use with private PDNS data			
customerID	[0-maxint]	<any>	If the user is authenticated, and has access to private PDNS data, this filter can limit which customers to return data for. Only records from customers the user has access to will be affected by this filter. Public data from other customers are "anonymous", and can be filtered out using the parameter "includeAnonymous".

tlp	white,green,amber,red	<any>	If set, limit the returned records by TLP. For customers which the user does not have access to private data, only TLP white records will be returned. If the user is not authenticated, then also will only TLP white records be returned.
aggregate	true,false	true	The PDNS service contains data from multiple customers, and PDNS records are stored per customer. By default, any data returned by the service will be aggregated into one record per query/answer tuple. If aggregate is set to false, the service will return each non-anonymous record separately.
includeAnonymous	true,false	true	If true (default), any public data in the PDNS service will be included in the query results. If set to false, only data from customers the user has access to will be included.

Result format

The result format is JSON, and consists of a result container, and a number of results.

The result container has the following format

```
{
    "responseCode": 200, # the response code. Normal responses should
return HTTP code 200
    "count": 20,         # the total number of matching records. If this
value is lower than the imposed limit, the resultset is truncated!
    "limit": 25,         # the limit imposed on the query results (default
is 25). Use the limit parameter in the request to set a higher limit.
    "offset": 0,         # the offset applied on the query results
(default i 0)
    "currentPage": 1,    # the current "page" (calculated from limit
/offset)
    "size": 20,          # the size of the current resultset (should be
same as count or limit)
    "data": [],          # the list of query result objects
    "messages": [],      # any server messages
    "metaData": {},      # any server metadata}
```

Each query result has the following format:

```
{
    "rrclass": "in",                        # the DNS record class
    "rrtype": "a",                          # the DNS record type
    "query": "cnn.com.",                    # the DNS record query part
    "answer": "157.166.255.19",             # the DNS record answer part
    "firstSeenTimestamp": 1340308340000,  # the first registered timestamp
for this record
    "lastSeenTimestamp": 1377520248000,   # the last registered timestamp
for this record
    "maxTtl": 300,                          # the maximum TTL observed for
this record
    "minTtl": 300,                          # the minimum TTL observed for
this record
    "times": 675,                           # the number of times this
record has been observed
    "tlp": "white",                         # the TLP of this record. Public
records have TLP "white"
    "customer": null,                       # the customer owning this
record. If null, this is an aggregated or anonymous record.
    "createdTimestamp": 0,                  # not in use
    "lastUpdatedTimestamp": 0,              # not in use
}
```

Examples

Query	Description
`https://api.mnemonic.no/pdns/v3/cnn.com`	Query for any record concerning the domain "cnn.com" Includes both DNS records for queries for cnn.com and DNS records returning the value "cnn.com" Does not contain subdomains of cnn.com
`https://api.mnemonic.no/pdns/v3/cnn.com?limit=0`	Query for any record concerning the domain "cnn.com". Return all results, no matter how big the resultset is.
`https://api.mnemonic.no/pdns/v3/cnn.com?rrType=A`	Query for A-records for the domain "cnn.com"
`https://api.mnemonic.no/pdns/v3/cnn.com?rrType=A&rrType=MX`	Query for A-records and MX-records for the domain "cnn.com"
`https://api.mnemonic.no/pdns/v3/157.166.255.19`	Query for any record for the IPv4 address "157.166.255.19"
`https://api.mnemonic.no/pdns/v3/2a04:4e42:400::323`	Query for any record for the IPv6 address "2a04:4e42:400::323"

JSON query format

An alternative to a simple GET-string is a POST query with a JSON query format to the URL

```
https://api.mnemonic.no/pdns/v3/search
```

Example

```
curl -X POST https://api.mnemonic.no/pdns/v3/search -d '
{
  "query": "cnn.com",
  "rrClass": [
    "IN"
  ],
  "rrType": [
    "a"
  ],
  "limit": 0,
  "offset": 0
}
```

> The POST-query is currently permitting the same parameters as the GET query, but new future parameters or advanced/nested parameters may be added only to the JSON query format as an "advanced format".

Authenticated queries

The PDNS API is publicly available, and does not require authentication.

However, unauthenticated queries are limited to see only public data (TLP white), and are limited to 1000 requests per day (currently).

To access private data (granted that mnemonic is collecting PDNS data from your customer), you need to use an authenticated query.
Also, users with an extended PDNS resource limit must use authenticated queries to be able to use their quota.

To authenticate your query, you need to request an API key from mnemonic.

To use that API-key, add a HTTP header to your request:

```
Argus-API-Key: 1234/1/abcd1234ef012
```

Example:

```
curl -H "Argus-API-Key: 1234/1/abcd1234ef012" -X GET https://api.mnemonic.
no/pdns/v3/cnn.com
```

Resource limits

All users are subject to resource limitations.

- Unauthenticated users are limited to 100 requests per minute, and 1000 requests per day.
- Authenticated users are limited according to their agreement with mnemonic.

If you hit the resource limit, Argus will return a 402 error, with the following JSON response:

```
{
    "responseCode": 402,
    "data": null,
    "messages": [
        {
            "message": "Resource limit exceeded",
            "messageTemplate": "resource.limit.exceeded",
            "type": "ACTION_ERROR"
        }
    ],
    "metaData": {
        "millisUntilResourcesAvailable": 558
    }
}
```

The resource limit is calculated both per minute and per day.

- If you reach the per-minute (short-term) resource limit, you will typically be rejected for a short period, to save resources on our end.
- If you reach the per-day (long-term) resource limit, you will typically be rejected for a period up to 24 hours. This means you have exhausted the data quota granted by mnemonic.

If you find yourself reaching the per-day resource limit a lot, you may want to request a higher resource quota from mnemonic

Please use the metadata key "millisUntilResourcesAvailable" to let your client back off gracefully.

Submit PassiveDNS records in bulk operation

PassiveDNS records can be submitet to the collector by using the POST endpoint /record in the API and will accept a list of multiple records. The minimum information in these records are the fields 'query', 'answer', 'firstSeen', 'times', 'minTtl', 'maxTtl', 'tlp' and 'recordType', or the request will be considered invalid. If not given the field recordClass defaults to 'IN', while lastSeen will be the submission time unless a timestamp is given.

If any of the records are invalid, the default behavior is to fail the entire request and return the reasons why to the user. This can however be overwritten by using the field 'ignoreOnFailed', in which case the valid records are submitted while the invalid ones are ignored, though the response will contain a list of those that failed the validation and why.

It should also be noted that unless 'query' or 'answer' is written as IP addresses, they are expected to end with a period(.). If one is missing then the endpoint will add one before they are stored.

Below is a minimal request:

```
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/pdns/v3/record -d '{
  "records": [
    {
      "query": "192.168.10.15",
      "answer": "192.168.10.16",
          "firstSeen": 1576589261000,
          "times": 12,
          "minTtl": 1,
          "maxTtl": 2,
          "tlp": "white",
          "recordType": "aaaa"
    }
  ]
}'
```

You can check the swagger documentation for a more detailed description of the endpoint and the values that can be used in the Json object.

Another request using strings for 'query' and 'answer', and an ISO-8601 timestamp for 'firstSeen'

```
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/pdns/v3/record -d '{
  "records": [
    {
      "query": "mnemonic.no",
      "answer": "not-mnemonic.no.",
          "firstSeen": "2016-11-30T15:47:00Z",
          "times": 12,
          "minTtl": 1,
          "maxTtl": 2,
          "tlp": "white",
          "recordType": "aaaa"
    }
  ]
}'
```

You can check the swagger documentation for a more detailed description of the endpoint and the values that can be used in the Json object.

Submit PassiveDNS records in bulk operation

If your user has the submitPDNSRecord function then you can submit single or multiple PassiveDNS records to the collector by using the POST endpoint /record in the API. The minimum information in these records require are the fields 'query', 'answer', 'firstSeen', 'times', 'minTtl', 'maxTtl', 'tlp' and 'recordType', or the request will be considered invalid. If not given the field recordClass defaults to 'IN', while lastSeen will be the submission time unless a timestamp is given.

If any of the records are invalid, the default behavior is to fail the entire request and return the reasons why to the user. This can however be overwritten by using the field 'ignoreOnFailed', in which case the valid records are submitted while the invalid ones are ignored, though the response will contain a list of those that failed the validation and why.

It should also be noted that unless 'query' or 'answer' is written as IP addresses, they are expected to end with a period(.). If one is missing then the endpoint will add one before they are stored.

Below is a minimal request:

```
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/pdns/v3/record -d '{
  "records": [
    {
      "query": "192.168.10.15",
      "answer": "192.168.10.16",
          "firstSeen": 1576589261000,
          "times": 12,
          "minTtl": 1,
          "maxTtl": 2,
          "tlp": "white",
          "recordType": "aaaa"
    }
  ]
}'
```

You can check the swagger documentation for a more detailed description of the endpoint and the values that can be used in the Json object.

Another request using strings for 'query' and 'answer', and an ISO-8601 timestamp for 'firstSeen'

```
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/pdns/v3/record -d '{
  "records": [
    {
      "query": "mnemonic.no",
      "answer": "not-mnemonic.no.",
          "firstSeen": "2016-11-30T15:47:00Z",
          "times": 12,
          "minTtl": 1,
          "maxTtl": 2,
          "tlp": "white",
          "recordType": "aaaa"
    }
  ]
}'
```

You can check the swagger documentation for a more detailed description of the endpoint and the values that can be used in the Json object.