Cyren-Data443 Malware URL Intelligence Feed

Cyren Threat InDepth is contextualized, correlated threat intelligence that allows security teams to gain a comprehensive and multi-dimensional view of evolving email-borne threats and make meaningful decisions to combat them. This high-fidelity, actionable intelligence is gathered by analyzing and processing billions of daily transactions across email content, suspicious files, and web traffic to provide unique, timely insights faster than other vendors.

Introduction

The automated daily analysis of billions of internet transactions in both web and email traffic - undertaken by Cyren's GlobalView™ Threat Intelligence Cloud - provides an unmatched view of phishing and fraud threats in real-time as they emerge. Cyren is able to extract URLs from this traffic by applying unique technology and algorithms, then analyze them within seconds and provide actionable information immediately.

Cyren's GlobalView™ accomplishes this by integrating techniques including Cyren's patented Recurrent Pattern Detection™ (RPD); real-time crawling by sophisticated machine-learning algorithms; heuristics and advanced detection logic; and cross- correlation with recent threat intelligence history. Expert security analysts fine-tune the detection logic to ensure continued detection accuracy. Cyren thus detects thousands of new malware URLs every day, providing valuable intelligence utilized by many of the world's leading security vendors and service providers, and adding to the millions of active malware URLs being monitored by Cyren at any given moment.

The malware URLs are gathered into separate files on the Cyren Datacenters, which are accessible via API and FTP. The purpose of this document is to describe the service from the subscriber's point of view, data structure and updating system.

Delivery Methods

Cyren offers two delivery methods for the Real-Time Malware URL feed. The first is fetching the feed entries via a REST API over HTTPS and the second is by pulling snapshot and delta files via FTP. Each method is explained in more detail below.

Delivery via "malware_urls" API

You can use the api-feeds.cyren.com "malware_urls" REST API to GET feed data over HTTPS. It works by using a script that continuously requests the newest entries to the feed starting from an "offset" and limited by a "count". This way you have the newest entries (or older entries if you choose) as well as control over how many results you receive on any request. Once you begin pulling entries, your dataset will begin to grow until it is current with the newest information of the malware URLs seen by Cyren. This architecture ensures you have the most up to date information from Cyren about each feed entry, but does require a "catch up" period when first deployed to build the initial dataset.

Below describes the process in further detail including examples.

> ⚠ A sample Python script is available for use in order to continuously query the API as described below. Please reach out to Cyren Support if you haven't received this script.

The API of the data provider is very simple with the following entry points accepting HTTP GET requests:

- `https://api-feeds.cyren.com/v1/feed/data`
- `https://api-feeds.cyren.com/v1/feed/info`

The following HTTP headers are expected:

- `Authorization` - must be used to provide the JWT token which will be provided to you as part of the onboarding process. i.e. `Authorization: Bearer <token>`
- `Accept-Encoding` - should be used to enable compression i.e. `Accept-Encoding: gzip`

Response Status Codes

Status Code	Message	Scenarios
200	Success	No errors in processing (depending on the parameters the response can be empty)
400	Bad Request	The request is missing a mandatory parameter, some parameter contains data which is incorrectly formatted
403	Forbidden	The request is not authorized to access data for the given feed
429	Too Many Requests	Too many requests were done in the configured time period
500	Internal Server Error	The service has encountered an unexpected situation and is unable to give a better response to the request
503	Service Unavailable	The service has encountered an unexpected situation and is unable to give a better response to the request

/v1/feed/data

Parameter	Meaning	Default	Examples
`feedId`	ID of the feed (required)		• `phishing_urls` • `ip_reputation` • `malware_urls` • `malware_files`
`format`	Output format: json or jsonl	`json1`	`json`
`count`	maximum number of records to return (1..100000)	`10000`	`20000`
`Offset`	minimal offset to start fetching		`42`
`startTime`	minimal timestamp to start fetching (in the ISO8601 format)		`2020-03-26T12:34:56.000Z`
`endTime`	maximum timestamp to stop fetching		`2020-03-26T12:34:56.000Z`

- The number of records returned is limited by the `count` parameter
- If the `startTime` parameter is set then all returned records will have timestamps equal or larger than `startTime`
- If the `endTime` parameter is set then all returned records will have timestamps less than `endTime`
- If the `offset` parameter is set then all returned records will have offset equal or larger than `offset`
- The first record returned has the minimal offset satisfying the conditions above

Notes for data connector developers:

- `jsonl` format allows to processes data in a streaming manner easier than `json`
- usage of `startTime` and `endTime` parameters is discouraged in data connectors
- employing offsets only is the easiest way to implement reliable data transport without data loss
- Pay attention that if the offset parameter is smaller than the offset of the oldest record present in the durable log then offset is automatically adjusted to the later

/v1/feed/info

Parameter	Meaning	Default	Examples
`feedId`	ID of the feed (required)		• `phishing_urls` • `ip_reputation` • `malware_urls` • `malware_files`

Examples

Fetching records of the feedId malware_urls starting with the offset 0 and maximum of two records returned in the jsonl format. Notice that the oldest record present in the durable log had the offset 264586 (in bold below) so the offset

```
> GET https://api-feeds.cyren.com/v1/feed/data?feedId=malware_urls&offset=0&count=2&format=jsonl
> Accept: */*
> Accept-Encoding: gzip
> Authorization: Bearer ...

< HTTP/1.1 200 OK
< Content-Length: 883
< Content-Type: application/jsonl
< Date: Thu, 09 April 2020 22:22:34 GMT
< Vary: Accept-Encoding

{"payload":{"action":"=","type":"url","identifier":"1b1b1bd9-7dcc-598a-b7fe-c9b42827084b","first_seen":"2020-03-24T21:33:41.000Z","last_seen":
{"payload":{"action":"=","type":"url","identifier":"2d8b72c8-4074-5f24-a955-6f21a4f4fd49","first_seen":"2020-03-16T11:40:56.000Z","last_seen":
```

Data is returned gzip-compressed with the `chunked Transfer-Encoding` for larger responses.
Getting the offsets range for the feed `malware_urls`

```
> GET https://api-feeds.cyren.com/v1/feed/info?feedId=malware_urls
> Accept: */*
> Accept-Encoding:gzip
> Authorization: Bearer <...

< HTTP/1.1 200 OK
< Content-Type: application/json; charset=utf-8
< Date: Thu, 09 Apr 2020 22:19:12 GMT
< Content-Length: 42
<
{"startOffset":264586,"endOffset":889302}
```

"malware_urls" API's JSON Structure

The following information is available for each URL entry within the feed:

⚠ Fields in **Bold** will always appear. If no data is available, some of the fields may not appear.

Name	Type	Description
payload	String	Contains the actual URL entry data described below
action	String	Contains values +/-/= • "+" adds a new record to the data set • "-" removes an existing record from the data set • "=" updates an existing record in the data set
type	String	Specifies the record type. All records are expected to be of the type "url"
identifier	String	A unique identifier of the record in Cyren database i.e. IP address or key
first_seen	Time stamp	Cyren first detection time in ISO 8601 compliant format [UTC]
last_seen	Time stamp	Cyren's most recent detection time in ISO 8601 compliant format [UTC]
detection		
category	Array	Detection category specifying malicious activity that the URL was involved in. Available categories: • malware • confirmed clean* *Note: Used to update URL/IP address category when it changes from suspected/malicious to confirmed clean
detection_ts	Time stamp	Most recent detection time for the observed category in ISO 8601 compliant format [UTC]
industry	Array	Detection category specifying the industry this URL can be associated. e.g. finance, e-commerce, etc.
meta		
port	Number	Server Port number
protocol	String	Distribution protocol
country_code	Array	Country abbreviation
relationships		
relationship_type	String	Description of the relationship of the threat to the URL
relationship_ts	Time stamp	The time the threat was detected by Cyren
sha256_hash/url_id/ip	String	Unique identifier of the related threat. Provides a pointer to the threat details according to your usage license. • If the entity type is a file then "sha256_hash" is displayed • If the entity type is a URL then "url_id" is displayed • If the entity type is an IP then "ip" is displayed
related_entity_category	String	Category of a threat related to the URL: e.g. malware
relationship_description	String	Description of malicious activity
detection_methods	Array	Cyren detection system(s) that identified a threat: e.g. Botnet detection, Active URL inspection, Advanced Threat Detection, Malware Detection etc.
url	String	The Uniform Resource Locator that Cyren considers or considered malicious
offset	Number	Offset number of entry
timestamp	Time stamp	Timestamp of when entry was fetched

Examples

Sample "phishing_urls" API entry returned:

```
{
    "payload": {
                "action": "=",
                "type": "url",
                "identifier": "b1b1bd9-7dcc-598a-b7fe-c9b42827084b",
                "first_seen": "2020-03-24T21:33:41.000Z",
                "last_seen": "2020-03-25T10:04:08.470Z",
                "detection": {
                    "category": [
                        "malware"
                    ],
                    "detection_ts": "2020-03-24T21:42:30.000Z",
                },
                "meta": {
                    "port": 803,
                    "protocol": "http"
                },
                "relationships": [
                    {
                        "relationship_type": "serves",
                        "relationship_ts": "2020-03-24T21:33:41.000Z",
                        "sha256_hash": "d508068aae1bc0117067a3baa1ea29c40f17289676e8b7d1de646faeb8917cd4",
                        "related_entity_category": "malware",
                        "relationship_description": "serves malware file"
                    },
                ],
```

```
                "detection_methods": [
                    "Malware detection"
                ],
                "url": "http://saborzuliano.com/index.php/images/stories/modules/images/stories/templates/images/modules/mod_ice_slideshow/js/
            },
    "offset": 264586,
    "timestamp": "2020-03-25T10:07:09.479Z"
}
```


Delivery via FTP

You can also choose to retrieve your feed via an FTP server that provides 3 types of files:

- Snapshot file - the service generates a single file once a day representing a snapshot of Cyren database at the time of its creation.
- Delta file - the service generates a new delta file every 5 minutes containing all new phishing URL's within that 5 minutes period
- Archive file – the service keeps the last months' worth of snapshots

Each day a new snapshot file is created replacing the previous snapshot file. Therefore, only a single snapshot file exists on the Datacenter for download at any given time. The snapshot file is used when the feed is started and each time that the feed mechanism on the subscriber's side is out of sync with the Datacenter. The content of the snapshot file should override any already-existing data on the database and thereby, override any existing data on the subscriber-side.

After a snapshot file was downloaded successfully, all successive delta files must be downloaded as well in order to ensure that the used data is up-to-date. From this point onwards the subscriber should poll the service every few minutes to download new delta files. Each time a new delta file is applied the database should be updated with the Next-Version value to maintain the correct order of updates.

The service always maintains in a FIFO (first in first out) queue of all information for the last day to be available to subscribers. The service makes sure that no matter when the last snapshot was taken, it will always have backed-up information for 24 hours from the current time.

In the dataurlphishing folder there are two main subfolders: snapshot and delta.

- Snapshot – folder contains the last snapshot file
- Delta – folder contains all the delta files for the last 24 hours

⚠ *Archive* contains the last months' worth of snapshots if needed

Snapshot File's JSON Structure

The snapshot file is named `data-malware-snapshot-YYMMDD.dat.gz` and it is an archived data file in JSON format. The file contains a snapshot of all malware URLs Cyren has observed in the past 30 days, and a new snapshot is generated every 24 hours.

The following reputation information is available for each URL:

⚠ Fields in **Bold** will always appear. If no data is available, some of the fields may not appear.

Name	Type	Description
type	string	Specifies the record type. All records are expected to be of the type "url"
identifier	string	A unique identifier of the record in Cyren database i.e. IP address or key
first_seen	time stamp	Cyren first detection time in ISO 8601 compliant format [UTC]
last_seen	time stamp	Cyren's most recent detection time in ISO 8601 compliant format [UTC]
detection		
category	array	Detection category specifying malicious activity that the URL was involved in. Available categories: • malware • confirmed clean* *Note: Used to update URL/IP address category when it changes from suspected/malicious to confirmed clean
detection_ts	time stamp	Most recent detection time for the observed category in ISO 8601 compliant format [UTC]
industry	array	Detection category specifying the industry this phishing URL can be associated. e.g. finance, e-commerce, etc.
meta		
port	number	Server Port number
protocol	string	Distribution protocol
country_code	array	Country abbreviation
Relationships		
relationship_type	string	Description of the relationship of the threat to the URL
relationship_ts	time stamp	The time the threat was detected by Cyren
sha256_hash/url_id/ip	string	Unique identifier of the related threat. Provides a pointer to the threat details according to your usage license. • If the entity type is a file then "sha256_hash" is displayed. • If the entity type is an IP then "ip" is displayed. • If the entity type is a URL then "url_id" is displayed.
related_entity_category	string	Category of a threat related to the URL: e.g. malware
relationship_description	string	Description of malicious activity
detection_methods	array	Cyren detection system(s) that identified a threat: e.g. Botnet detection, Active URL inspection, Advanced Threat Detection, etc.
URL	string	The Uniform Resource Locator that Cyren considers or considered malicious

Delta File's JSON Structure

The delta file is named `data-malware-delta-YYMMDDHH_X.dat.gz` and it is an archived data file in JSON format. The delta file is generated every 5 minutes and contains URL records that have been added, updated, or removed since the previous snapshot or delta. The delta files contain incremental updates where the "X" in the delta file name represents a sequential number. The delta file number is reset to "0" after each daily snapshot, and delta updates must be applied in sequence from low to high.
The delta file contains the same fields as the snapshot with an addition of the Action field.

Name	Type	Description
Action	string	Contains values +/-/= • "+" adds a new record to the data set • "-" removes an existing record from the data set • "=" updates an existing record in the data set

Examples


```
{
    "type": "url",
    "identifier": "b43df7c0-b04e-59dc-b2ea-a86680e1a579",
    "first_seen": "2020-03-11T00:28:03.000Z",
    "last_seen": "2020-03-11T01:00:25.813Z",
    "detection": {
        "category": [
            "malware"
        ],
```

```
        "detection_ts": "2020-03-11T00:31:06.000Z",
    },
    "meta": {
        "port": 80,
        "protocol": "http"
    },
    "relationships": [
        {
            "relationship_type": "serves",
            "relationship_ts": "2020-03-11T00:28:03.000Z",
            "related_entity_type": "file",
        "related_entity_identifier"": "56ef021f2f1306a002d8662c1b9c665bb704a1efe95dbbb07f72e19aa3b5087",
        "related_entity_category": "malware",
        "relationship_description": "serves malware file"
        }
    ],
    "detection_methods": [
        "Malware detection"
    ],
    "url": "http://p0rt666.blogsss.ru/"
}
```


Sample delta file record:


```
{
    "action": "+",
    "type": "url",
    "identifier": "9525a0de-6ee0-5bd3-9f19-027153404ba6",
    "first_seen": "2020-04-08T15:28:38.000Z",
    "last_seen": "2020-04-08T15:29:29.810Z",
    "detection": {
        "category": [
            "malware"
        ],
        "detection_ts": "2020-04-08T15:28:42.000Z"
    },
    "meta": {
        "port": 443,
        "protocol": "https"
    }, "relationships": [
        {
            "relationship_type": "serves",
            "relationship_ts": "2020-04-08T15:28:38.000Z",
            "related_entity_type": "file",
            "related_entity_identifier": "9e3272b57571e1415b760ad24e2ceb529ef136bcf9742ed95813edd2d5fc0dce",
            "related_entity_category": "malware",
            "relationship_description": "serves malware file"
        }
    ], "detection_methods": [
            "Malware detection"
    ],
    "url": https://newssummedup.com/summary/Gbajabiamila-confirms-suspension-of-22-78bn-external-loan-aaa
}
```