URLhaus | Data exposed through Real Time Intelligence Feed v. publicly available data

URLhaus provides context-rich signal from abuse.ch, informing of malicious URLs being used for malware distribution.

Users can access this data via the Real Time Intelligence Feed, or via publicly available APIs. These access methods provide different metadata outputs, as highlighted in this document.

The Real Time Intelligence Feed is the only access method to receive all observed threat signals.

Metadata field	Metadata description	Real Time Feed	Publicly available data
URL Additions			
`_idx`	An integer representing the incremental number of the message.	✓	✗
`_ts`	The Unix timestamp, indicating when the message was received by the real time infrastructure.	✓	✓*
`uuid`	An internal, unique identifier for the message. This is the property that should be used to dedupe the incoming flows where necessary.	✓	✗
`type`	Defines the type of message.	✓	✓
`id`	Represents the ID of the URL in the URLhaus database. It uniquely identifies the specific URL tracked. It also can be used to assemble the HTTP link to the URLhaus record page (https:// urlhaus.abuse.ch/url/`id`/).	✓	✓
`url`	Is the added URL.	✓	✓
`host`	The host associated with this URL (extracted from the URL).	✓	✓
`url_status`	Is a string that represents the status of the URL. Possible values are 'online', 'offline', and 'unknown'. 'unknown' is reported when the URL has not yet been checked by URLhaus.	✓	✓
`anonymous`	Is a boolean value indicating if the reporter of the URL wants to stay anonymous.	✓	✓
`reporter`	Is the handle of the reporter of the URL or 'null' if it should be anonymous. Currently, the handle equals the Twitter handle of the reporter. After migration to a new authentication system for abuse.ch, this handle will change to one from abuse.ch's own authentication platform.	✓	✓
`tags`	Are a list of tags associated with the added URL, as shown in URLhaus. Tags are "free field" and defined by the reporter (submitter) for the URL.	✓	✓
URL Removals			
`_idx`	Is an integer representing the incremental number of the message.	✓	✗
`_ts`	Is the Unix timestamp, indicating when the message was received by the realtime infrastructure.	✓	✗
`uuid`	Is an internal, unique identifier for the message. This is the property that should be used to dedupe the incoming flows where necessary.	✓	✗
`type`	Defines the type of message.	✓	✗
`id`	Represents the ID of the URL in the URLhaus database. This is needed to assemble the HTTP link to the URLhaus record page.	✓	✗
`url`	Is the URL being added.	✓	✗
`removal_note`	Is a text string, human-readable, that describes why the URL has been removed.	✓	✗
URL Changes			
`_idx`	Is an integer representing the incremental number of the message.	✓	✗
`_ts`	Is the Unix timestamp, indicating when the message was received by the real time infrastructure.	✓	✗
`uuid`	Is an internal, unique identifier for the message. This is the property that should be used to dedupe the incoming flows where necessary.	✓	✗
`type`	Defines the type of message.	✓	✗
`id`	Represents the ID of the URL in the URLhaus database. This is needed to assemble the HTTP link to the URLhaus record page (https:// urlhaus.abuse.ch/url/`id`/).	✓	✗
`url`	Is the URL being modified.	✓	✗
`field`	Shows which field has been changed. Fields currently supported are: `tag`, `url_status`	✓	✗
`value`	Is the new value of the affected field.	✓	✗
`action`	This represents what action happened to the `field`. The action could be `add`, `remove` or `change`.	✓	✗
New file download			
`_idx`	Is an integer representing the incremental number of the message.	✓	✗
`_ts`	Is the Unix timestamp, indicating when the message was received by the real time infrastructure.	✓	✓
`uuid`	Is an internal, unique identifier for the message. This is the property that should be used to dedupe the incoming flows where necessary.	✓	✗
`type`	Describes the type of this message and is always "file_download".	✓	✗
`sha256_hash`	Is the SHA256 hash of the file.	✓	✓
`md5_hash`	Is the MD5 hash of the file.	✓	✓
Observed payloads			
`_idx`	Is an integer representing the incremental number of the message.	✓	✗
`_ts`	Is the Unix timestamp, indicating when the message was received by the real time infrastructure.	✓	✓
`uuid`	Is an internal, unique identifier for the message. This is the property that should be used to dedupe the incoming flows where necessary.	✓	✗
`type`	Describes the type of this message and is always "payload_observed".	✓	✗
`id`	Represents the ID of the URL in the URLhaus database. This is needed to assemble the HTTP link to the URLhaus record page (https:// urlhaus.abuse.ch/url/`id`/).	✓	✗
`url`	Is the full URL from which the file was downloaded.	✓	✓
`mime_type`	Is the Multipurpose Internet Mail Extensions (MIME) type of the payload received.	✓	✗
`file_type`	Is the result of the Unix "file" command (not to be confused with the content-type header from the webserver).	✓	✓
`file_ext`	Is the guessed file extension (or 'null', if not available).	✓	✗
`file_size`	Is the size (in bytes) of the payload received.	✓	✗
`file_name`	Is the filename as extracted from the HTTP `Content-Disposition` header in the response. It's 'null' if the info is not available.	✓	✗
`md5_hash`	Is the MD5 hash of the payload received.	✓	✓
`sha256_hash`	Is the SHA256 hash of the payload received.	✓	✓
`imphash`	Is the imphash of the payload received.	✓	✗
`ssdeep`	Is the ssdeep of the payload received.	✓	✗
`tlsh`	Is the tlsh of the payload received.	✓	✗
`telfhash`	Is the telfhash of the payload received.	✓	✗
`malware`	This is the malware family.	✓	✓
Payload changes			
`_idx`	Is an integer representing the incremental number of the message.	✓	✗
`_ts`	Is the Unix timestamp, indicating when the message was received by the real time infrastructure.	✓	✗
`uuid`	Is an internal, unique identifier for the message. This is the property that should be used to dedupe the incoming flows where necessary.	✓	✗
`type`	Describes the type of this message and is always "payload_change".	✓	✗
`md5_hash`	Is the MD5 hash of the payload received.	✓	✗
`sha256_hash`	Is the SHA256 hash of the payload received.	✓	✗
`field`	Shows the affected field where the change occurred. Currently, only malware is supported.	✓	✗
`value`	Is the new value of the affected field.	✓	✗
`action`	This represents what action happened to the `field`. The action could be `add`, `remove` or `change`.	✓	✗

*** Comparable data for this field is provided via publicly available data, however the field names are not an exact match.**
If you require the comparable field name, please speak with your Spamhaus contact.