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

MalwareBazaar provides context-rich signal from abuse.ch, providing intelligence on confirmed malware samples.

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
File Additions			
`_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. It's always 'file_addition'.	✓	✓*
`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.	✓	✓
`md5_hash`	Is the MD5 hash of the payload received.	✓	✓
`sha256_hash`	Is the SHA256 hash of the payload received.	✓	✓
`sha1_hash`	Is the SHA1 hash of the file.	✓	✓
`sha3_384_hash`	Is the SHA3-384 hash of the file.	✓	✓
`humanhash`	Is the human-readable hash. Provides human-readable representations of digests.	✓	✗
`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.	✓	✓
`gimphash`	Is the gimphash of the file.	✓	✓
`dhash_icon`	Is the dhash of the file icon.	✓	✓
`mime_type`	Is the Multipurpose Internet Mail Extensions (MIME) type of the payload received.	✓	✓*
`file_type`	Is the result from Unix "file" command.	✓	✓*
`file_ext`	Is the guessed file extension (or 'null', if not available).	✓	✓*
`malware`	This is the malware family.	✓	✓*
`tags`	Is a list of tags associated with this file.	✓	✓
`anonymous`	Is a boolean that indicates whether the submitter of this file wants to remain anonymous or not.	✓	✓
`reporter`	Is the abuse.ch handle of the submitter of this file (or 'null', if not available).	✓	✓
`origin_country`	Two letter Country code of the country from where the submission has been made.	✓	✓
`delivery_method`	Distributed via e-mail attachment.	✓	✓
`comment`	Is a comment from the reporter of the URL (or 'null', if not available).	✓	✓
File 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. It's always 'file_change'.	✓	✗
`md5_hash`	Is the MD5 hash of the payload received.	✓	✗
`sha256_hash`	Is the SHA256 hash of the payload received.	✓	✗
`sha1_hash`	Is the SHA1 hash of the file.	✓	✗
`sha3_384_hash`	Is the SHA3-384 hash of the file.	✓	✗
`field`	Shows the affected field where the change occurred (supported fields: `tag`, `malware`, `file_ext`).	✓	✗
`value`	Is the new value of the affected field.	✓	✗
`action`	Is an enumerated field that describes the action. May contain `add`, `remove`, `change`.	✓	✗
File 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 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. It's always 'file_removal'.	✓	✗
`md5_hash`	Is the MD5 hash of the payload received.	✓	✗
`sha256_hash`	Is the SHA256 hash of the payload received.	✓	✗
`sha1_hash`	Is the SHA1 hash of the file.	✓	✗
`sha3_384_hash`	Is the SHA3-384 hash of the file.	✓	✗
`removal_note`	Is a text string showing the removal note as inserted by the system or the remover.	✓	✗
YARA matches			
`_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`	UUID identifying this message.	✓	✗
`type`	Type of this message.	✓	✓*
`md5_hash`	MD5 hash of the affected file.	✓	✓
`sha256_hash`	SHA256 hash of the affected file.	✓	✓
`sha1_hash`	SHA1 hash of the affected file.	✓	✓
`sha3_384_hash`	SHA3-384 hash of the affected file.	✓	✓
`yara.rule_name`	Name of the matching YARA rule.	✓	✓*
`yara.author`	The author of the matching YARA rule.	✓	✓*
`yara.description`	Description of the matching YARA rule.	✓	✓
`yara.reference`	Reference of the matching YARA rule.	✓	✓
`yara.tlp`	Traffic Light Protocol (TLP) of the matching YARA rule.	✓	✓
Code Signing Certificate Blocklist (CSCB) additions			
`_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.	✓	✓*
`type`	Type of this message.	✓	✗
`type`	Type of this message.	✓	✓*
`subject_cn`	Subject Common Name (CN).	✓	✓
`issuer_cn`	Subject Common Name (CN).	✓	✓
`algorithm`	Algorithm used.	✓	✓
`valid_from`	Datetime from when this Code Signing Certificate is valid from.	✓	✓
`valid_to`	Datetime to when this Code Signing Certificate is valid to.	✓	✓
`serial_number`	Serial number of the Code Signing Certificate.	✓	✓
`thumbprint_algorithm`	Thumbprint algorithm.	✓	✓
`thumbprint`	Thumbprint.	✓	✓
`bl_reason`	Code Signing Certificate Blocklist (CSCB) listing reason.	✓	✓*
`malware_samples`	List of malware samples signed with this Code Signing Certificate.	✓	✗

*** 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.