WMC GLOBAL

WORLDWIDE BRAND AND CUSTOMER PROTECTION AT SCALE

KITINTEL
API DOCUMENTATION

v1.1 08.05.2024

WMC GLOBAL

TABLE OF CONTENTS

GETTING STARTED WITH KITINTEL

The KITIntel APIs are a set of static analysis tools for investigating and comparing phishing kit content within single or multiple kits. It can search file hashes and file content, retrieve content, and submit kits to KITIntel for cross-analysis.

A phishing kit is a package of software tools, often in the form of a compressed file, that makes it easier to launch phishing attacks and exploits. Phishing kits allow attackers to rapidly deploy and redeploy attacks. Phishing kits can target consumers, employees, financial institutions, marketplaces, and many more. Kits can deploy malware, collect credentials, detect bots, block IP ranges, generate QR codes, and update dynamically. Customers can use KITIntel to compare phishing kits, discover evidence about attackers and kit publishers, identify evasion techniques, and find new exploits. WMC Global collects phishing kits and associated data, supporting internal threat intelligence and external law enforcement initiatives. WMC Global is one of the largest providers of phishing kits to federal law enforcement. With KITIntel, you can:

- Search, retrieve content, and identify important metadata from over 80,000 unique phishing kits, and many millions of attack artifacts.
- Identify attackers using the same techniques targeting different brands.
- Uncover specific tactics, techniques, and procedures (TTPs) targeting customers and improve defenses.
- Gather attribution identifiers
- Submit known phishing URLs to the platform for inclusion in KITIntel for analysis.

DOCUMENT STATUS

This document supports v1.1 for the KITIntel API. The API and documentation are subject to change.

[WMC Global Website]

WMC GLOBAL

KIT/search DOCUMENTATION

Search phishing kits for specific content or hashes.

BASE URL: api.phishfeed.com/KIT/v1

SCHEME: HTTPS

API: POST /search

CONTENT TYPE: application/JSON

DATA DICTIONARY INPUT PARAMETERS

PARAMETER NAME	DESCRIPTION	SUPPORTS	EXAMPLES	DATA TYPE
x-api-key	Authentication token	Header		string

DATA DICTIONARY SEARCH PARAMETERS

PARAMETER NAME	DESCRIPTION	SUPPORTS	EXAMPLES	DATA TYPE

datetime_filter	Filter results based on the time of the phishing kit's discovery.		**RANGE EXAMPLE:**
			`"datetime_filter": {"gte": "2021-05-10","lte": "2021-06-01"}`
			TIME SCALE EXAMPLE:
			`"datetime_filter": {"gte": "now-30d"}`
				string

size_filter	Filter results on file size in bytes.		**RANGE EXAMPLE:** `"size_filter ": {"gte": 98502, "lte": 98502}`	string

			EXACT EXAMPLE: `"size_filter ": {"gte": 1000}`	

kit.size_filter	Filter search results based on phishing kit file size in bytes.		**RANGE EXAMPLE:**
			`"kit.size_filter": {"gte": 98502, "lte": 98502}`
			EXACT EXAMPLE:
			`"kit.size_filter": {"gte": 1000}`
				string

page_size	Limit the result count returned per page.		`"page_size": 30`	int
filter	Filter the fields returned by key.		`"filter": [`	array
content	Search the files or phishing kits for a specific content string.	- Wildcard * - Regex Search Documentation		string
kit.filetype	Search for a phishing kit package type.	- Wildcard * - Regex Search Documentation	`.zip .rar`	string
kit.kitname	Search for an exact phishing kit name.	- Wildcard * - Regex Search Documentation		string
kit.md5	Search for an MD5 hash of a phishing kit.	- Wildcard * - Regex Search Documentation		string
kit.sha256	Search for an SHA256 hash of a phishing kit.	- Wildcard * - Regex Search Documentation		string
kit.ssdeep	Search for a phishing kit using the ssdeep.	- Wildcard * - Regex Search Documentation		string
kit.uuid	Search for a phishing kit using the KIT UUID.	- Wildcard * - Regex Search		string

		Documentation		
filename	Search for an exact filename within a phishing kit.	- Wildcard * - Regex Search Documentation		string
filetype	Search or filter for an exact file type within a phishing kit.	- Wildcard * - Regex Search Documentation		string

fullfilename	Search for an exact file including the file path within a phishing kit.	- Wildcard *
		- Regex Search Documentation
			`"fullfilename": "*/Login/step1*"`
			
			`"fullfilename": "/Login/confirm.php"`
				string

md5	Search for an exact MD5 hash of a file within a phishing kit.	- Wildcard * - Regex Search Documentation		string
sha256	Search for an exact SHA256 hash of a file in a phishing kit.	- Wildcard * - Regex Search Documentation		string
ssdeep	Search for a file in a phishing kit using the ssdeep.	- Wildcard * - Regex Search Documentation		string
uuid	Search for a file in a phishing kit via unique identifier.	- Wildcard * - Regex Search Documentation		string

DATA DICTIONARY OUTPUT PARAMETERS

PARAMETER NAME	DESCRIPTION	DATA TYPE
total_count	The total number of entries found that match the search criteria.	int
count	The number of results returned per page.	int
scroll_id	After the initial query. Include the scroll_id to return the paginated response.	string

datetime	The time the phishing kit was discovered.	string
kit.md5	The MD5 hash of a phishing kit.	string
kit.uuid	A unique kit identifier.	string
kit.size	The size of the phishing kit measured in bytes.	int
kit.ssdeep	The fuzzy hash for the phishing kit.	string
kit.filetype	The phishing kit file extension.	string
kit.sha256	The SHA 256 hash of a phishing kit.	string
kit.kitname	The name of the phishing kit file.	string
file.filetype	The extension of the file within the phishing kit.	string
file.sha256	The SHA 256 hash of a file within a phishing kit.	string
file.fullfilename	The filename and path of the file in the phishing kit.	string
file.content	See a file's plain text content.	string
file.ssdeep	The fuzzy hash for a file in a phishing kit.	string
file.filename	The name of the file in the phishing kit.	string
file.size	The size of the file in the phishing kit measured in bytes.	int
file.uuid	The unique identifier of a file in a phishing kit.	string
file.md5	The MD5 hash of a file within a phishing kit.	string

Sample Query

JSON

```
{
  "content": "html",
  "fullfilename": "/.*ASSETS.[A-Z0-9]{10}.*/",
  "page_size": 1,
  "filter": [
    "datetime",
    "UUID",
    "fullfilename"
  ]
}
```

Responses

RESPONSE CODE	DESCRIPTION
200	Success

JSON


```
{
  "total_count": 7,
  "count": 1,
  "scroll_id": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1…QtTXJRV21FNC1YWUYxd0NfZw==",
  "results": [
    {
      "datetime": "2021-06-27T21:00:02Z",
      "file": {
        "fullfilename": "/lll/ASSETS-CQNLMYR0VOU5AXY04P7W/error_log",
        "UUID": "acf01c50-512f-45a3-a649-a8337a9fbe69"
      }
    }
  ]
}
```


RESPONSE CODE	DESCRIPTION
400	Error

JSON


```
{
  "message": "Could not parse request into json"
}
```

KIT/content DOCUMENTATION

Use the file UUID to request the URL to download the content of a file in a phishing kit. The download will be available for five minutes after the request is issued.

BASE URL: api.phishfeed.com/KIT/v1

SCHEME: HTTPS

API: POST /content

CONTENT TYPE: application/JSON

CONTENT INPUT DATA DICTIONARY

PARAMETER NAME	DESCRIPTION	SUPPORTS	EXAMPLES	DATA TYPE
uuid	Provide the file UUID to download the file.	Exact match only	`15454811-ead9-4a30-a7d4-bbdee35a399b`	string

CONTENT OUTPUT DATA DICTIONARY

PARAMETER NAME	DESCRIPTION	DATA TYPE
filename	The filename and path of the file in the phishing kit.	string
uuid	The unique identifier of a file in a phishing kit.	string
content	The URL where the file from the phishing kit can be downloaded. The link will expire in five minutes.	string

Sample Query

JSON

```
{
  "UUID": "15454811-ead9-4a30-a7d4-bbdee35a399b"
}
```

Responses

RESPONSE CODE	DESCRIPTION
200	Success

JSON

```
{
  "filename": "confirm.php",
  "UUID": "15454811-ead9-4a30-a7d4-bbdee35a399b",
  "content": "https://kit-intel-content.s3.amazonaws.com/15454811-ead9-4a30-a7d4-
bbdee35a399b.txt?
AWSAccessKeyId=AKIAWO7R6AKSF46LM35H&Signature=63Bjo3J%2F4YP5wjYUOdtZaopSd04%3D&Expires=1627
901392"
}
```

RESPONSE CODE	DESCRIPTION
400	Error

JSON

```
{
  "message": "Could not parse request into json"
}
```

KIT /submit DOCUMENTATION

Submit a phishing kit to KITIntel.

POST /SUBMIT

Submitting a phishing kit is a two-part process. Use /SUBMIT to add the name of the kit archive and receive a URL to upload the kit binary code via the /PUT method.

BASE URL: api.phishfeed.com/KIT/v1

SCHEME: HTTPS

API: POST /submit

CONTENT TYPE: application/JSON

STEPS TO UPLOADING A PHISHING KIT

To upload a ZIP kit file for analysis:

1. Send the kit name to /submit
2. PUT the zip binary to the returned URL

/SUBMIT INPUT DATA DICTIONARY

PARAMETER NAME	DESCRIPTION	EXAMPLES	DATA TYPE
file_name	Submit the phishing kit file name.	`16shop_new.zip`	string

/SUBMIT OUTPUT DATA DICTIONARY

PARAMETER NAME	DESCRIPTION	DATA TYPE
url	The URL that can be used for the phishing kit upload.	string

Sample Query

JSON

```
{
  "file_name": "16shop_2022.zip"
}
```

Responses

RESPONSE CODE	DESCRIPTION
200	Success

JSON

```
{
  "url": "https://kit-parser-search-api-test.s3.amazonaws.com/16shop_2022.zip?X-Amz-Algorithm=AWS4- HMAC-SHA256&X-Amz-Credential=AKIA4MUYUCLG2AYEC4WH%2F20210818%2Fus-east-1%2Fs3%2Faws4_request&X-Amz- Date=20210818T094806Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=d49501235d50df412e59fc2f2c8115e237ed4fae11d80f21c9de8dd40cb62c9f",
}
```

PUT /[URL returned from /CONTENT]

Submitting a phishing kit is a two-part process. Use /SUBMIT to add the name of the kit archive and receive a URL to upload the kit binary code via the /PUT method.

BASE URL: api.phishfeed.com/KIT/v1

SCHEME: HTTPS

API: PUT /[URL]

CONTENT TYPE: application/binary

/[URL returned from /CONTENT] SAMPLE QUERY

Sample Query

JSON

```
50 4b 03 04 14 00 08 00  08 00 ed 52 0b 53 00 00 00 00 00 00 00 00 1e 00  00 00 0b 00 20 00
74 65 73
74 69 6e 67 2e 74 78  74
```

Responses

RESPONSE CODE	DESCRIPTION
200	Success