WMC GLOBAL

WORLDWIDE BRAND AND CUSTOMER PROTECTION AT SCALE

COMPROMISED CREDENTIAL RECOVERY API DOCUMENTATION

V2.0 11.22.2024

WMC GLOBAL

TABLE OF CONTENTS

GETTING STARTED WITH COMPROMISED CREDENTIAL RECOVERY

Phishing websites utilize various techniques to evade detection, capture credential data, and extricate that data without being caught. The attack types have no uniform data structure, requiring the API's flexible designs to search vast amounts of compromised data.

The Compromised Credential Recovery (CCR) API locates compromised customer, employee, and 3rd party vendor data enabling the exploration of pre-dark web phishing victim data and helping fraud teams understand what information was obtained in an attack. This may include data to gain access to digital banking, SIM swapping, and/or bypassing two-factor authentication (2FA). The API integrates with downstream systems to automatically reset compromised account passwords or engage additional account security protocols, protecting users at speed. With Compromised Credential Recovery, you can:

- Identify compromised account data weeks or months before it is sold publicly on dark web marketplaces.
- Augment existing processes for fighting Payment and Credit Card Fraud
- Obtain access to compromised victim information via API
- Received details like URL, impersonated brand, and date/time when the credential was found
- Attain a detailed picture of threat actors and credential phishing
- Receive rapid return of credentials to minimize risk exposure
- Gain a deeper understanding of threat actor tactics, techniques, and procedures (TTPs)

Search for partial matches of email, credit card number, IP address, account number, phone number, and many other content formats to identify specific entries. The Compromised Credential Recovery system periodically rechecks active attacks for new credentials. Search for the latest version of a captured file or review multiple versions to identify updates.

WMC Global Website

...

DOCUMENT STATUS

This document supports v4.0 for the Compromised Credential Recovery API. The API and documentation are subject to change.

CCR DATA DICTIONARIES

DATA DICTIONARY INPUT PARAMETERS

PARAMETER NAME	DESCRIPTION	DATA TYPE
x-api-key	User API key; a unique identifier used to authenticate a user issued by WMC Global	string

DATA DICTIONARY SEARCH PARAMETERS

PARAMETER NAME	DATA TYPE	REQUIRED?	DESCRIPTION	EXAMPLE
brand	string	No	Indicate the impersonated brand.	`WMC`
content	string	No	Allow to query over the content of the file. Support regex (place the expression between "/") (Regular expressions use the Lucene syntax, which differs from more standardized implementations and are applied to the terms in the field (i.e. tokens), not the entire field.)	`*WMC*`
credit_card	string	No	Credit card extracted	`444401230`

			from the content of the file.	4560789
credit_card_bin	string	No	Credit card bins extracted from the content of the file.	401190
datetime_filter	string	No	Date range to be consulted. Format: YYYY-MM-DD or logical dates like now / now-#(h|d|y) Valid parameters are “gt” (greater than), ”gte” (greater than or equal),”lt” (less than) and “lte” (less than or equal). The default value is to search all data.	{‘gte’: ‘2023-04-08’, ‘lte’: ‘now-8h’}
email	string	No	Search for emails extracted from the content of the file.	user@wmc.com
file_name	string	No	Path to the victim file.	Htttp_phishing.com/filename.txt
filter	string	No	Select the columns that you want to consult. Available columns are: brand, content, credit_card, credit_card_bin, date_time, email,	“filter”: “date_time, file_name, id, source”

			file_name, file_part, id, ip, last_version_found, phish_id, url, source. The default value includes: "id","brand","date_time","file_name","phish_id","url", "last_version_found","file_part"	
ip	string	No	IPs extracted from the content of the file.	`192.168.1.1`
last_version_found	string	No	If "true" it will filter the files to search only over the last version of each file, if "false" it will search over all versions (including the last one).	`FALSE`
match_phrase	string	No	Filter files by a phrase that must exist in the content field (Supports whitespaces).	`WMC GLOBAL`
phish_id	string	No	The unique PhishFeed URL identifier the record was found to be associated with when applicable.	`33010101`
size_filter	int	No	Items to be returned. The default value is 10	`100`
			Indicate the source of the file, if it comes from a	`site`

source	string	No	live phising site ("site") or private conversation / forum data ("forum").	
url	string	No	The phishing attack URL where the credential file was found when applicable.	`https://phishing.com`

DATA DICTIONARY FILE PARAMETERS

PARAMETER NAME	DATA TYPE	REQUIRED?	DESCRIPTION	EXAMPLE
id	string	yes	Unique identifier of a file (Do not include "_p_#).	`046d5b54-425*`
size_filter	int	No	Number of items to be returned (In case you don't know how many file parts exist).	5
first_part	int	No	First part of the file to be consulted (This column is not needed if you want to start from the first part of the file.)	1
last_part	int	N0	Last part of the file to be consulted. (This column is not needed if you want to end at the last part of the column.)	3

CCR EXAMPLES

NOTE

When filtering columns, the fields “email,” “credit_card,” “credit_card_bin,” and “ip” support “.lines,” this will return the lines from the file in which the value appears. If a value appears multiple times on the same row, the row will appear that number of times on the list. The columns “brand,” “content,” “credit_card,” “credit_card_bin,” “email,” “file_name,” “ip,” “phish_id,” and “url” support wildcards.

EXAMPLE: `{“filter”: “brand”, “id”, “credit_card.lines”, “phish_id”, “credit_card”: “44858_ OR 45360_” ,”datetime_filter”: {“gte”: “2023-04-08”,”lte”: “now-8h”},size_filter: 100 }`

For performance implications, adding additional wildcards to any search has an impact on the performance and likelihood of a timeout. Similarly, the date range can also impact performance. In general, alphanumeric searches are less performant than numeric, such as credit_card or ip. As a best practice, if there are timeouts with a particular given query that is being attempted, the recommendation is to pare down the number of wildcards (if applicable), adding in additional filtering to remove extraneous results, or reduce the date/time range of the query.

For the date/time range, to perform a search for a single day as opposed to a range of dates, a combination of both gte and lte filters with the same day is required.

EXAMPLE: `"datetime_filter": {"gte": "2024-04-10", "lte": "2024-04-10"}`

EXAMPLES OF QUERIES USING /SEARCH ENDPOINT:

Get all email records with the domain “@domain.com” over the last 30 days:

```JSON
{
"filter": "date_time, file_name, source, id, email.data",
"email":  "*@domain.com",
"datetime_filter": {"gte": "now-30d"},
"size_filter": "1000"
}
```

EXAMPLE OF RETURNED DATA:

```JSON
[
{
"date_time": "2024-07-17T20:30:58+00:00",
"file_name": "filename.txt",
"source": "forum",
"id": "id123",
"email": [
{"data": "email@domain.com"},
{"data": "email@domain.com"}
  ]
}
]
```

USE REGEX TO FIND POTENTIAL PHONE NUMBERS OVER THE LAST 30 DAYS:

```JSON
{
"filter": "date_time, file_name, id, source",
"content": "/\\d{3}\\d{3}\\d{4}/",
"datetime_filter": {"gte": "now-30d"},
"size_filter": "1000"
}
```

EXAMPLE OF RETURNED DATA:

```JSON
[
{
"date_time": "2024-07-31T11:14:28+00:00",
"file_name": "filename.txt",
"source": "forum",
"id": "id123",
"content": {
"total_content_matched": 4,
"matched_content": [
"123456789",
"987654321",
"741852963",
"963852741"
]
}
}
]
```

COMBINATION OF OR LOGIC FOR CREDIT_CARDS, AS WELL AS EMAILS WITH THE DOMAIN “@DOMAIN.COM” OVER THE LAST 30 DAYS:

```JSON
{
"filter": "date_time, file_name, source, id",
"email":"*@domain.com",
"credit_card":"1* OR 2* OR 3*",
"datetime_filter": {"gte": "now-30d"},
"size_filter": "10000"
}
```

EXAMPLE OF RETURNED DATA:

```
JSON
[
{
"date_time": "2024-07-11T12:38:55+00:00",
"file_name": "filename.txt",
"source": "forum",
"id": "id123",
"email": [
{"data": email@domain.com }
],
"credit_card": [
{"data": "1234567890123456"},
{"data": "2345678901234567"},
{"data": "3456789012345678"}
]
}
]
```

EXAMPLE OF A QUERY USING /FILE ENDPOINT:

```
JSON
{
    "id": "0c6c2d84-37f2",
    "first_part": 1,
    "last_part": 3
}
```

EXAMPLE OF RETURN FROM THE /FILE ENDPOINT:

```
JSON
{
"start_line": 81708,
"first_part": 1,
"last_part": 2,
"content": "Files content",
"id": "046d5b54-4250-4763-b07c-457bc596ab97"
}
```

CCR RESPONSES

BASE URL: api.wmcglobal.com/recovsearch/v4

SCHEME: HTTPS

API: POST /search

API: POST /file

CONTENT TYPE: application/JSON

RESPONSE CODE	DESCRIPTION
200	success

RESPONSE CODE	DESCRIPTION
403	error

JSON

```
{
  "message": "Forbidden"
}
```

RESPONSE CODE	DESCRIPTION
413	error

JSON

```
{
  “total_records”: 10000,
  “message”: “Response
  Payload is too large”
}
```

RESPONSE CODE	DESCRIPTION
500	error

JSON

```
{
  "message": "A system error has occurred; please try again later or contact support"
}
```