**API SCRIPT**

The customer has various options in Kaduu for accessing the query data:

* Via dashboard: data can be displayed and then exported (CSV, XML, DOCX etc)
* Via webhook
* Via REST API
* Via alerting (email)

The REST API allows you great flexibility in automation and integration into your existing processes and applications. You can use the API according to the official documentation <https://wiki.kaduu.ch/doku/lib/exe/fetch.php?media=kaduu-2.0.pdf>. In this document we illustrate how the API can be used with a customizable Python script which could assist you with few business use cases:

* **White labeled alerts:** You use Kaduu to regularly provide end customers or specific people with email alerts on certain topics (e.g. leaked data). To do this, you want to use an email with your design, company name and structure and also send this via your infrastructure.
* **Ticketing System Integration:** You want to integrate Kaduu into your existing environment and create tickets or incidents automatically in Splunk, Jira, Slack or similar systems.
* **Automated Output Parsing**: You want to store all results for certain search queries daily in an easy-to-process format (CSV, XLS, etc.) locally in a folder of your choice for analysis and further processing.
* **Reduce Workload in Alerting:** You want to save time when using Kaduu's email alerts: Instead of logging into the system for email alerts and researching the cause of the alerts, you want to receive the specific raw data of the alerts via email.

**Preparation**

Please make sure you have all the required modules installed. Please run

pip install -r requirements.txt

You will need the following modules:

* requests~=2.31.0
* urllib3~=1.26
* tldextract

Make also sure to run Python 3.11.4 or higher.

**How to use the script?**

Run the script either by giving the input in the terminal or operate it with an input file. Type "csv" as an option and then if your input file is in the same directory, type yourinputfile.csv. Please make sure you type the correct name of the input file.

Example:

Python v5.py

Enter 'manual' to input data manually or 'csv' to load from a CSV file: csv

Enter the path to your CSV file: input.csv

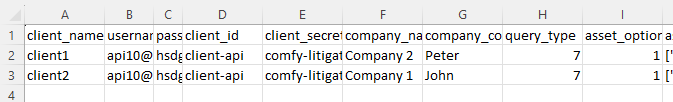
Before running the input file, put your username and password in the first two rows and adjust the different columns. For example enter one or multiple domains belonging to the same client under asset\_name without the subdomain.

**Using the input file (example\_input.csv)**

The script can operate with an input file. Here is the syntax for the different fields:

|  |  |  |
| --- | --- | --- |
| field | examples | usage |
| username | test@user.com | authentication: An email for control.leak.center access |
| password | Your\_password | authentication: A password for control.leak.center access |
| client\_id | client-api | Authentication. Is hardcoded and the same for all clients |
| client\_secret | comfy-litigate-embargo-forelimb | Authentication. Is hardcoded and the same for all clients |
| company\_name | United Security Providers | We only use it in HTML to create email, this field is not used in querying the API |
| company\_contact | John Smith | We only use it in HTML to create email, this field is not used in querying |
| query\_type | 7 | '0': Domains,'1': 'accounts','2': 'leaks','3': 'certificates','4': 'botnets','5': 'cc','6': 'ransomware','7'. From 0 to 7, where 7 is all |
| asset\_option | 1 | From 1 to 3: '1': 'Domain' (example.com), '2': 'Second-level domain' (example), '3': 'IP range' |
| asset\_name | ["juliusbaer.com","blkb.ch","basel.com"] | A list of values to perform the query (except CC). We recommend using domain names. Do not include the subdomain! |
| credit\_card\_owner | ['John'] OR [''Baer', 'Julius'] | A list of values to perform CC endpoint querying (The field credit\_card\_owner can be a single name (smith) or full name (john smith). |
| credit\_card\_number | ['555000\*'] OR [''555000\*', '555001\*'] | A list of values to perform CC endpoint querying. The field credit\_card\_number can be a partial number (like first 6 digits 543210\*)) |
| ransomware\_query | Hospital Basel | The clients name. A string that should be split and each word queried separately (Ransomware Endpoint only). We query if the name appears on threat actors ransomware sites. If you enter a name like "Hospital basel" we will query each word (hospital, basel) individually. |
| date\_range | 2023-01-30 TO \* | The date range for the search. use \* TO \* if no date range is required. The format is YYYY-MM-DD! |
| alert | 0 | If set to 1, use the script for daily queries. It automatically uses the current date range for the query and the date\_range values will be ignored, if set to 0, the date\_range will be used |
| alert\_email | client@email.com | Email adress where the alert is sent |
| sender\_email | mssp@sender.com | Sender Email Adress for the alert |
| sender\_password | Passw\*\*ord | Passwort to authenticate to SMTP port |
| smtp\_server | Mail.mssp.com | Mail Server to send email alerts |
| smtp\_port | 587 | Mail server port |
| supply\_chain | ["Hospital","basel","xplain"] | A string that should be split and each word queried separately (Ransomware only). If client name contains 2 values and supply\_chain contains 10 values, we should perform 10+2 queries |

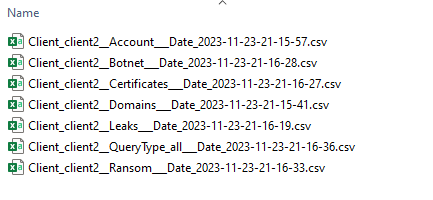
You can create multiple clients in your input file. Simply use 1 line per client:



The system will then append the put in separate directories:



Inside each folder you will find the output from the script:



**What are the API Endpoints integrated into the script?**

We didn’t integrate all API endpoints available, but just a few relevant ones from control.leak.center:

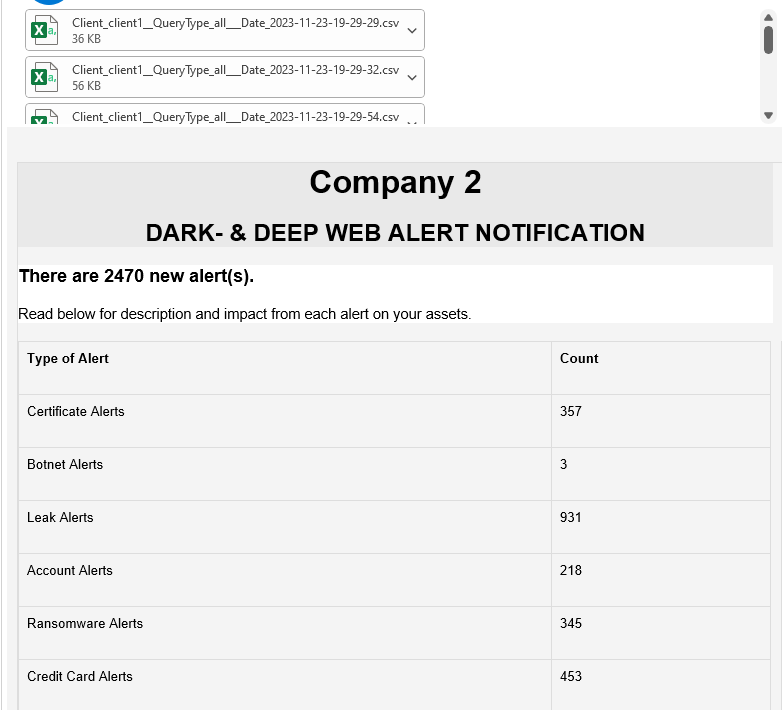
* 'domain': <https://app.leak.center/svc-saas/domain/search> <https://wiki.kaduu.ch/doku/doku.php?id=domain_database_search>
* 'accounts': <https://app.leak.center/svc-saas/account/search> , (account is just an extraction of leaks, we therefore recommend using leaks as it contains more data)
* 'leaks': <https://app.leak.center/svc-saas/leak/search> , <https://wiki.kaduu.ch/doku/doku.php?id=leak_search>
* 'certificates: <https://app.leak.center/svc-saas/certificate/search> <https://wiki.kaduu.ch/doku/doku.php?id=certificate_monitoring>
* 'botnets': <https://app.leak.center/svc-saas/bot/record/search> <https://wiki.kaduu.ch/doku/doku.php?id=bot_search>
* 'ransomware': <https://app.leak.center/svc-saas/ransomware/search> , <https://wiki.kaduu.ch/doku/doku.php?id=ransomware_site_monitoring>
* 'cc': <https://app.leak.center/svc-saas/cc/search>, <https://wiki.kaduu.ch/doku/doku.php?id=credit_card_search>

About Domain search:

* If your domain is less than 5 letters (example: sbb.com), we query the API endpoint with the domain like this “sbb”. You will only see different TLD’s as results like sbb.ch, sbb.online etc.
* If your domain is more than 5 letters (example: example.com), we query the API endpoint with the domain like this “\*example\*” using wildcards. You will all domains containing the clients domain like “getyour-example.com, greatexamples.com, example34.com.

**How to use the script for daily alerting?**

Alert Option “1” Use the Script for a daily Job. The script will ignore the date range, but fetch the current day and execute the query for the current day only. If alerting is set to 0: It will use the date range. The email will contain the results (you can choose if the results are attached to the email):



You can customize the HTML Email in the Python code following line 340:

html = """

<!DOCTYPE html>

<html>

<head>

<title>{{COMPANY\_NAME}} Security Alert</title>

<style>

body {

font-family: 'Arial', sans-serif;

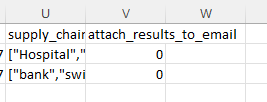
margin: 0;

padding: 0;

background-color: #f4f4f4;

…..

If you want the CVS files with the results attached to the email, please set the value to “1” in “attach\_results\_to\_email”. If not, set it to 0:



|  |  |
| --- | --- |
| Send Alerts, using the current day, no attachment of the output to the email: | |
| attach\_results\_to\_email = 0 | alert = 1 |
| Send Alerts, using the current day, attachment of the output to the email: | |
| attach\_results\_to\_email = 1 | alert = 1 |
| Send Alerts, using the date range, attachment of the output to the email: | |
| attach\_results\_to\_email = 1 | alert = 0 |
| Send no Alerts, just save the data locally | |
| attach\_results\_to\_email = 0 | alert = 0 |

**Running the script in Crontab (Linux)**

If you want to run the script in Crontab:

* **Locate Your Python Script:** Ensure your script (V1.py) is executable and located in a suitable directory. You might also want to make sure that the script has the appropriate permissions to execute. You can make it executable using the command: bash 🡪 chmod +x /path/to/V1.py
* **Find the Path to Your Python Interpreter**: You need to know the path to the Python interpreter that will run your script. You can find this with: bash 🡪 which python (Or if your script is written for Python 3 specifically: bash 🡪 which python3; This command will return a path like /usr/bin/python3)
* **Edit Your Crontab**: Open your crontab for editing by running: bash 🡪 crontab -e (This will open the crontab file in your default text editor)
* **Add a Cron Job:** In the crontab file, add a line that specifies when the script should run and what command should be executed. The general format is: \* \* \* \* command to execute. For daily execution, you might use something like: 0 0 \* \* \* /usr/bin/python3 /path/to/V1.py (This means the script V1.py will run at midnight every day. The five asterisks represent minute (0-59), hour (0-23), day of the month (1-31), month (1-12), and day of the week (0-7, where both 0 and 7 mean Sunday), respectively)
* **Save and Exit:** Save the crontab file and exit the editor. Your cron job is now scheduled. The cron daemon will automatically pick up the new job.
* **Check the Crontab List:** To make sure your cron job is listed, you can view your current cron jobs with: crontab -l
* **Logging**: If you want to log the output of your script, you can modify the cron job line like this: 0 0 \* \* \* /usr/bin/python3 /path/to/V1.py >> /path/to/logfile.log 2>&1 (This command will append the output of your script to logfile.log and include both standard output and standard error)
* **Environment Issues:** If your script depends on environment variables or a specific environment setup, you might need to either export the necessary variables directly in the crontab or run a shell script from cron that sets up the environment before running your Python script.
* Remember to replace /path/to/V1.py with the actual path to your Python script and /usr/bin/python3 with the path to the Python interpreter on your system.

**Potential issues**

The email might get filtered as it contains many special characters and an attachment with leak data

Error while sending email: (552, b'5.7.0 This message was blocked because its content presents a potential\n5.7.0 security issue. Please visit\n5.7.0 https://support.google.com/mail/?p=BlockedMessage to review our\n5.7.0 message content and attachment content guidelines. y22-20020a17090668d600b00a029145bbb9sm1167825ejr.3 - gsmtp')

Solution: Send without attachment. Set the value to “0” in attach\_results\_to\_email, but define alert value to 1.