URL Risk Adviser

Made by Moti Shaul

Contents

Introduction

The URL Risk Adviser script leverages data from VirusTotal to analyze the safety of URLs. By querying VirusTotal's API, the script provides users with valuable insights into the potential risks associated with a given URL.

The script assists users in proactively protecting themselves against malware, phishing attacks, and other online threats.

Once the user sets the API Key and provides the URL, the script fetches data about the URL. It then analyzes this data to generate a comprehensive report with relevant information, including a clear conclusion called the 'Risk Level'.

Additionally, the script offers the user the option to output the report to a text file alongside the output to the console window.

You can download the script at this link: https://github.com/liomoti/url_risk_adviser

Design

- **VirusTotal** (https://www.virustotal.com)
  VirusTotal is a free online service that analyzes files and URLs for potential malware threats. It aggregates results from various antivirus engines and other tools to provide comprehensive security assessments.

- **API Integration**
  I chose to use VirusTotal API as my cybersecurity-related API because its reliability and the good reputation, the ease of use and the quality of the data.

- **Data Analysis**
  The script is focused about 3 aspects from the information from VirusTotal:
  1. URL Category
  2. URL Https certificate
  3. URL antivirus scan results

  Based on those aspects the script is calculate the “Risk Level” measure.

- **Risk Level Calculation**
  There are 4 Risk Levels, and they are calculated as below:

Risk Level	Https Certificate	Category in blacklist categories	Antivirus scanned as suspicious	Antivirus scanned as malicious
High	-	-	-	✓
Medium	-	✓	✓	X
Low	X	✓	X	X
No Risk	✓	✓	X	X

  ✓ means exist
  X means not exist
  - means don’t matter if it exists or not

- **Reporting**
  The report contains:

  1. The detected URL
  2. The categories that the Webroot URLs mapping sites have categorized the site
  3. Information about the Https certificate
  4. Antivirus scans results
  5. “Risk Level” conclusion

  The report is printed to the console, the script offers the user the option to export the report to a text file.

❖ Security Controls

In this script, several security measures have been implemented to ensure the safety and integrity of the application:

- **Secure API Key Handling** - The script does not store the API Key by default in the .env file. Instead, after the user provides the API Key, they are prompted by the script if they want to securely save it in the .env file for future use. This approach minimizes the risk of unauthorized access to sensitive information and enhances overall security.
- **User Input Validation:** To prevent potential security vulnerabilities, the script employs regex validation on user input to ensure that the provided URL address adheres to the expected format.
- **Robust Error Handling Techniques:** Throughout the script, various programming techniques are employed to prevent crashes and provide appropriate feedback to the user in extreme cases.
    - Try-catch blocks
    - Checking fields in a JSON response in an un-crashing manner (using the 'get' function)

These security controls collectively contribute to the overall resilience and reliability of the script, helping to safeguard against potential threats and ensure a secure user experience.

Instructions

❖ Requirements

To execute the script, ensure that `Python` (version 3 or later) and `pip` (Python package installer) are installed on your system.

❖ Execute the script

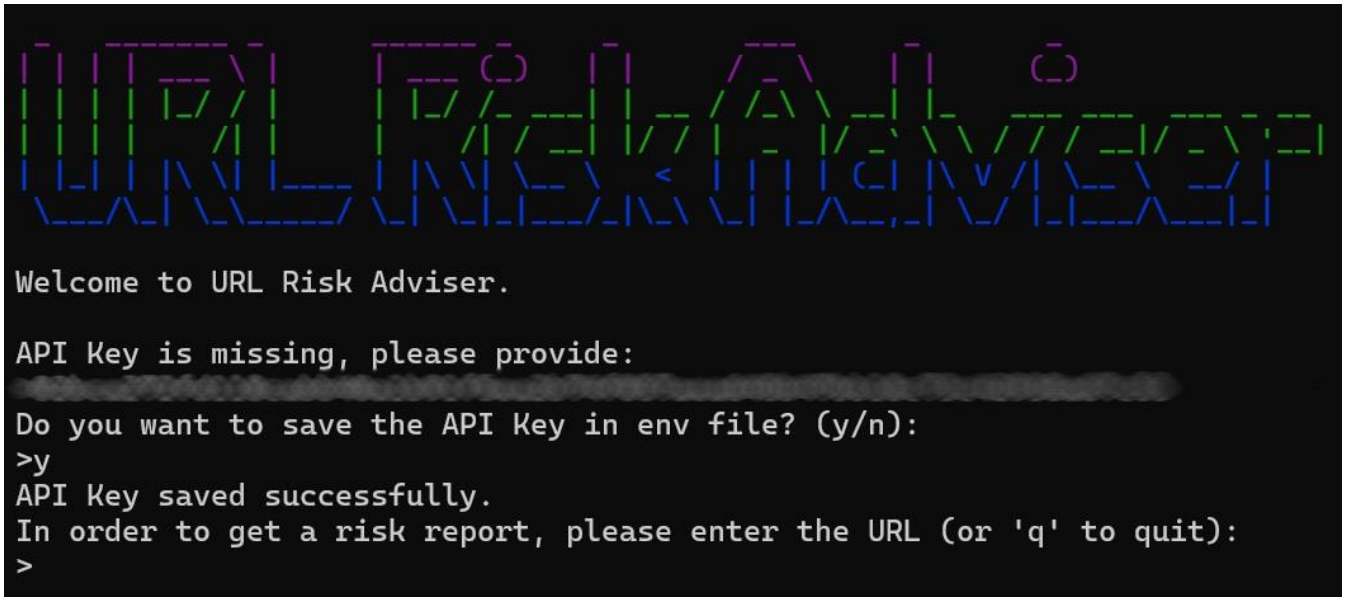
- **For Windows VM:**
    1. Navigate to the directory where the script files are located.
    2. Double-click on the file named "`run_script_windows.bat`" to execute the script.
- **For Linux VM:**
    1. Open a terminal window.
    2. Navigate to the directory where the script files are located.
    3. Run the command `chmod +x run_script_linux.sh` to give execution permissions to the shell script.
    4. Execute the script by running the command `./run_script_linux.sh`.

❖ First running

After clicking on the appropriate file to execute the script, you will see the URL Risk Adviser interface displayed on your screen.

Enter the API Key and press 'Enter', the script will ask if you want to save the API Key in the .env file for future uses.

In the example below I chose to save the API Key and entered 'y'.



❖ Usual use

Once you've entered the URL, the script will proceed to analyze the data and generate the report.
In the example below I entered www.google.com and chose to export the report to text file.
The text file report will be stored in the reports folder in the folder of the script files.

```
In order to get a risk report, please enter the URL (or 'q' to quit):
>www.google.com
Fetching data from VirusTotal about [www.google.com]...

            ------------------------------------
                Detected URL: www.google.com
            ------------------------------------
Categories: searchengines / search engines & portals / search engines / search engines and portals

        ----------------- HTTPS Certificate -----------------
Https certificate: Yes
Last https certificate date: 2024-04-09 01:06:35


        ------------- Antiviruses scan results -------------
Last analysis date: 2024-04-09 01:06:35
+-------------+-------------+-------------+------------+-----------+
|   malicious |   suspicious |   undetected |   harmless |   timeout |
+=============+=============+=============+============+===========+
|           0 |           0 |          19 |         71 |         0 |
+-------------+-------------+-------------+------------+-----------+

        ---------------------- Summary ----------------------
Risk Level: No Risk

Do you want to export the report to a text file? (y/n):
>y
Report successfully written to reports/report_google_09-04-2024_17-39-09.txt
In order to get a risk report, please enter the URL (or 'q' to quit):
>
```

Note that a valid URL can be without the ‘www’ prefix but must be in a valid form of URL:

```
In order to get a risk report, please enter the URL (or 'q' to quit):
>google
Invalid URL format! [ Example for valid URL: www.example.com or example.com ]

In order to get a risk report, please enter the URL (or 'q' to quit):
>google.
Invalid URL format! [ Example for valid URL: www.example.com or example.com ]

In order to get a risk report, please enter the URL (or 'q' to quit):
>.com
Invalid URL format! [ Example for valid URL: www.example.com or example.com ]

In order to get a risk report, please enter the URL (or 'q' to quit):
>
```

Report Samples

- **google.com – No Risk example**

```
                    ------------------------------------
                        Detected URL: www.google.com
                    ------------------------------------
Categories: searchengines / search engines / search engines and portals

            ---------------- HTTPS Certificate ----------------
Https certificate: Yes
Last https certificate date: 2024-04-09 01:06:35


            ------------- Antiviruses scan results ------------
Last analysis date: 2024-04-09 01:06:35
+-------------+--------------+--------------+------------+-----------+
|   malicious |   suspicious |   undetected |   harmless |   timeout |
+=============+==============+==============+============+===========+
|           0 |            0 |           19 |         71 |         0 |
+-------------+--------------+--------------+------------+-----------+

            ---------------------- Summary --------------------
Risk Level: No Risk
```

- **googele.com – High Risk example**

```
                    ------------------------------------
                        Detected URL: googele.com
                    ------------------------------------
Categories: Malicious, Phishing, Suspicious (alphaMountain.ai) / ads /
spyware and malware / Phishing and Other Frauds / parked domain

            ---------------- HTTPS Certificate ----------------
Https certificate: No


            ------------- Antiviruses scan results ------------
Last analysis date: 2024-04-05 07:28:42
+-------------+--------------+--------------+------------+-----------+
|   malicious |   suspicious |   undetected |   harmless |   timeout |
+=============+==============+==============+============+===========+
|           7 |            0 |           23 |         60 |         0 |
+-------------+--------------+--------------+------------+-----------+

            ---------------------- Summary --------------------
Risk Level: High Risk
```

❖ **israel.postal-fees.info – Medium Risk example**

```
                ------------------------------------
                    Detected URL: israel.postal-fees.info
                ------------------------------------
Categories: No data

            ---------------- HTTPS Certificate ----------------
Https certificate: Yes
Last https certificate date: 2024-03-26 18:12:08


            ------------- Antiviruses scan results ------------
Last analysis date: 2024-03-26 18:12:06
+-------------+--------------+--------------+------------+-----------+
|   malicious |   suspicious |   undetected |   harmless |   timeout |
+=============+==============+==============+============+===========+
|           0 |            1 |           27 |         62 |         0 |
+-------------+--------------+--------------+------------+-----------+

            ---------------------- Summary ---------------------
Risk Level: Medium Risk
```