



- 1. INTRODUCTION
 - 1.1 BSD License
 - 1.2 What is “Oblivion”?
 - 1.3 Features
 - 1.4 APIs used
- 2 Compatibilities
- 3 Oblivion Client
 - 3.1 Installation
 - 3.1.1 Windows
 - 3.1.2 Linux
 - 3.2 Configuration
 - 3.2.1 APIs
 - 3.2.2 G-Mail
 - 3.2.2.1 Enable notifications
 - 3.2.2.2 Configuring recipients
 - 3.2.2.3 Notification message
 - 3.2.2.3.1 Message with results
 - 3.2.3 Google Drive
 - 3.2.3.1 Changing destination folder
 - 3.2.4 Telegram
 - 3.2.4.1 Creating Bot
 - 3.2.4.2 Configuring Bot
 - 3.2.4.3 Enable notification
 - 3.2.4.4 Configuring recipients
 - 3.2.4.5 Notification message
 - 3.2.4.5.1 Message with results
 - 3.2.5 Encryption key

- 3.3 Analysis
 - 3.3.1 Configuring credentials
 - 3.3.1.1 Databases
 - 3.3.2 Configuring analysis
 - 3.3.2.1 Modules
 - 3.3.2.2 Formats
 - 3.3.2.3 Sending to Google Drive
 - 3.3.3 Results
- 3.4 History
 - 3.4.1 Clear history
- 3.5 Scheduling of analyzes
- 3.6 Decrypt
- 4 Oblivion Server
 - 4.1 Installation
 - Windows 4.1.1
 - Linux 4.1.2
 - 4.2 Configuration
 - 4.2.1 General
 - 4.2.1.1 Secret Key
 - 4.2.1.2 APIs
 - 4.2.1.2.1 Enabling APIs
 - 4.2.2 G-Mail
 - 4.2.2.1 Enable notifications
 - 4.2.2.2 Configuring recipients
 - 4.2.2.3 Notification message
 - 4.2.3 Google Drive
 - 4.2.4 Telegram
 - 4.2.4.1 Enable notifications
 - 4.2.4.2 Configuring recipients
 - 4.2.4.3 Notification message
 - 4.2.5 SSH
 - 4.2.5.1 SSH ec2
 - 4.2.6 AWS S3
 - 4.2.7 Databases

- 4.3 Requisition
 - 4.3.1 URL
 - 4.3.1.1 Examples
 - 4.3.2 Header
 - 4.3.2.1 Examples

INTRODUCTION -----

1.1 BSD 3-Clause License

Copyright (c) 2020, Gustavo Torroni
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and / or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2 What is “Oblivion”?

Oblivion is an open-source tool made in Python, which has functionality to store user data (passwords, emails, domains, URLs, IPs, CIDRs, Bitcoin Wallet, IPFS hashes etc.) and to monitor whether this data has been leaked.

It supports a graphical version for desktops, aimed at end users and a server version, which has API functionality.

1.3 Features

Common Website

Performs Web Scraper on sites most likely to disclose some type of data leak.

Indexed sites:

- Pastebin

Google Dorks

Through Web Scraper Oblivion performs queries on Google Search using Google Dorks to check past data leaks.

Word Lists

Performs Web Scraper on some word lists hosted on the GitHub server and evaluates if any user password is contained in them.

APIs

Checks for any leak related to user credentials through the consumption of three APIs (Intelx, Scylla and Have I Been Pwned).

CVEs (experimental)

Consumes the Circl CVE API to check the last 30 published CVEs and analyze whether any software present inside the host is vulnerable to these CVEs.

1.4 APIs used

- Scylla - free (<https://scylla.sh/>)
- Intelx - paid (<https://intelx.io/>)
- Have I Been Pwned - paid (<https://haveibeenpwned.com/>)
- Circl CVE Search– free (<https://www.circl.lu/>)

Compatibility -----

The versions mentioned below guarantee greater stability for Oblivion. The user can make use of other versions of Python / OS, however stability is not guaranteed:

Python

- ✓ 3.8.6

Operational systems

- ✓ Windows N10 1909
- ✓ Ubuntu Server 20.04 LTS

Some operating systems may experience a certain delay when processing API calls or sending results via SSH:

- Windows Server 2019
- Windows Server 2016

Oblivion Client -----

3.1 Installation

Python 3.8.6 installation is recommended prior to Oblivion installation:

(<https://www.python.org/downloads/release/python-386/>)

3.1.1 Windows

1. Download the Oblivion repository on GitHub (<https://github.com/loseys/Oblivion>)
2. Rename the “Windows” folder to “Oblivion”
3. Go to the “Oblivion Windows” folder and run the “install.py” file.
4. Execute OblivionClient.py

3.1.2 Linux


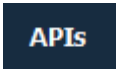
1. \$ git clone <https://github.com/loseys/Oblivion>
2. \$ mv Linux Oblivion
3. \$ cd Oblivion
4. \$ sudo python3.8 install.py
5. \$ sudo python3.8 OblivionClient.py

3.2. Configuration

The Oblivion Client settings apply to both the Windows system and the Linux system.

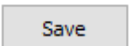
IMPORTANT: It is essential that for every configuration made Oblivion the application is restarted, otherwise the new settings will not be used.

3.2.1 APIs

1. Click on the gear icon 
2. Click on “APIs” in the left corner 
3. Place the corresponding API keys

Keys

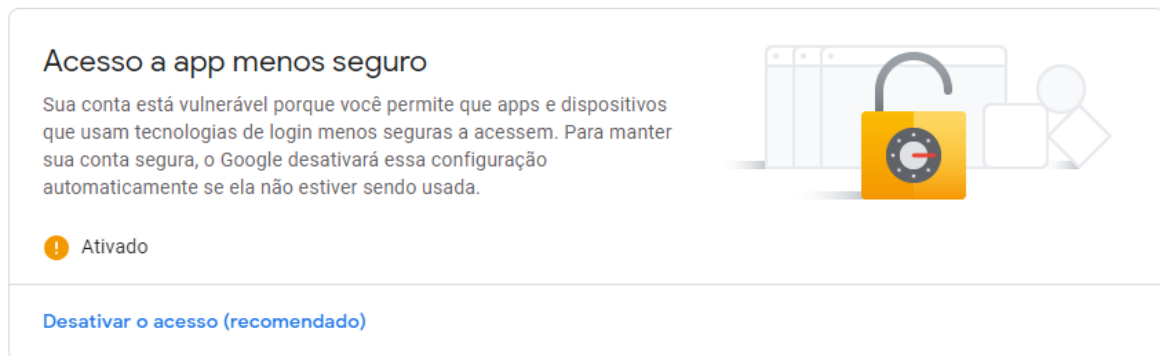
<input type="checkbox"/> Intelx	<input type="text"/>
<input type="checkbox"/> Have I Pwned	<input type="text"/>
<input type="checkbox"/> Scylla	<input type="text"/>

4. If you want to leave the API active you must check the Check Box for the API ☐
5. Click on “save” 

3.2.2 G-Mail

It is recommended that you use your own account to send notices, avoid using personal G-Mails. It is also important to note that the two-step verification must be disabled.

1. You must first activate the option “Access to less secure app” in the security options of your G-Mail (<https://myaccount.google.com/lesssecureapps>). It's import


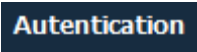


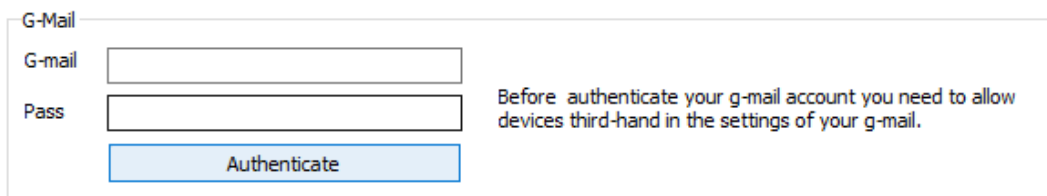
← Acesso a app menos seguro

Alguns apps e dispositivos usam tecnologias de login menos seguras, o que deixa sua conta vulnerável. Você pode desativar o acesso desses apps, o que recomendamos, ou ativá-lo se optar por usá-los apesar dos riscos. O Google desativará essa configuração automaticamente se ela não estiver sendo usada. [Saiba mais](#)

Permitir aplicativos menos seguros: ATIVADA



2. Click on the gear icon 
3. Click on “Authentication” in the left corner 
4. Enter your gmail account email and password






5. Click on “Authenticate”



6. Click on “save”

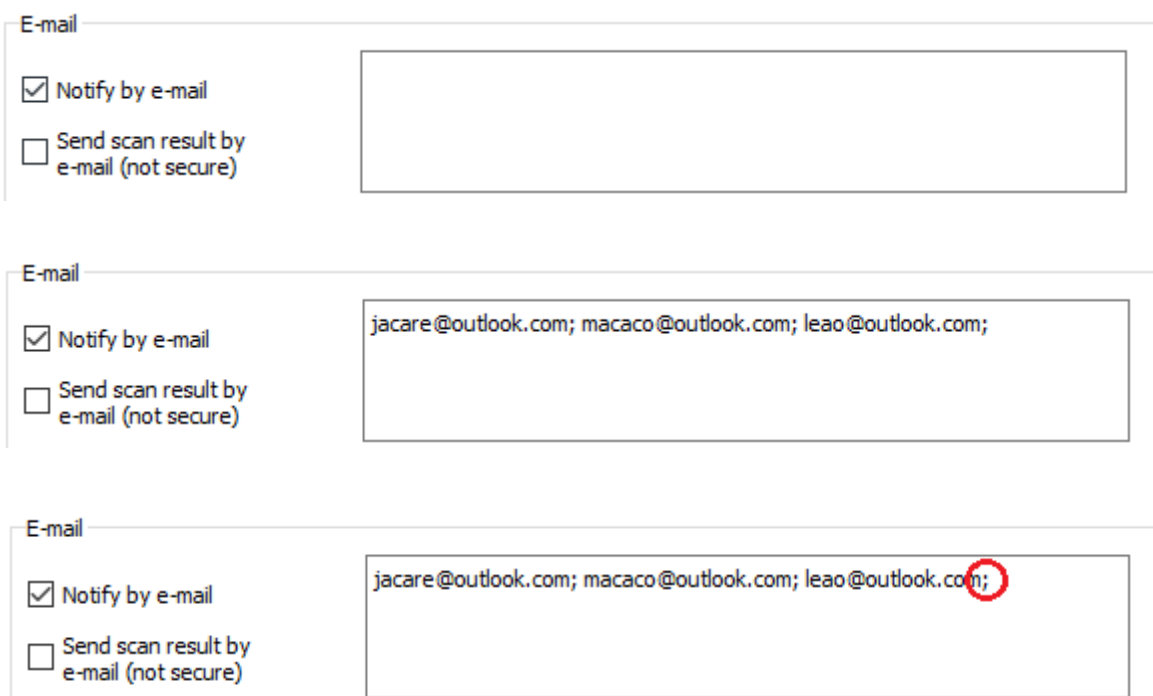
Save

3.2.2.1 Enabling notification

1. Go to Settings 
2. Click on "Notification" on the left 
3. To enable check the "Notify by e-mail" option  Notify by e-mail

3.2.2.2 Configuring recipients

1. Go to Settings 
2. Click on "Notification" on the left 
3. Enter the e-mails in the blank field, they must be separated by “;” and with


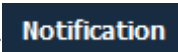


The image displays three sequential screenshots of a web interface for configuring email notifications. Each screenshot shows a form with the title 'E-mail' and two checkboxes: 'Notify by e-mail' (checked) and 'Send scan result by e-mail (not secure)' (unchecked). A text input field is present to the right of the checkboxes.

- Top Screenshot:** The text input field is empty.
- Middle Screenshot:** The text input field contains the email addresses 'jacare@outlook.com; macaco@outlook.com; leao@outlook.com;'.
- Bottom Screenshot:** The text input field contains the same email addresses, but the final semicolon at the end is circled in red, indicating it is a required character for proper formatting.

4. It is important that at the end it contains “;” to indicate the end of e-mails, otherwise Oblivion may present problems during the sending of e-mails

3.2.2.3 Notification message

1. Go to Settings 
2. Click on "Notification" on the left 
3. In the "E-mail" field you will have a text field with an HTML code. To change the message you must change this HTML code. If you are not familiar with HTML, we recommend not changing the code, just the message, so as not to trigger problems in sending the email.



E-mail

☒ Notify by e-mail

☐ Send scan result by e-mail (not secure)



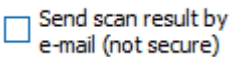
jacare@outlook.com; macaco@outlook.com; leao@outlook.com;

```
<h1 style="color: #5e9ca0;"><span style="color: #003366;">Oblivion</span></h1>
<h4 style="color: #2e6c80;"><span style="color: #ff0000;">Aten&ccedil;&atilde;o:</span></h4>
<p>Suas cred&ecirc;ncias podem estar em risco: acesse seu aplicativo Oblivion ou <a href="https://drive.google.com/drive/folders/1Zsnh9_79r3hCdGiHeV_sZWkVwckGPzv">dique aqui</a> para saber mais.</p>
<p>Caso esteja com algum problema, contate o administrador.</p>
<p>&nbsp;</p>
```

IMPORTANT: The use of accents within the HTML body is not recommended.

3.2.2.3.1 Message with result of the analysis

If you want the email to come with the results in the message field you must:

1. Go to Settings 
2. Click on "Notification" on the left 
3. In the "E-mail" field, check the option "Send scan result by e-mail" 

IMPORTANT: Very long analysis results may present a problem at the time of sending, due to the limitation of characters in the body of the email. It is advisable to avoid using this option.

3.2.3 Google Drive

1. Sign in to Google Developer (<https://developers.google.com/drive>)

2. Access the "Guides" tab 

3. Click on "Python" in the left menu

4. Go to "Enable Drive API"

Enable the Drive API

You're all set!

You're ready to start developing!

5. Click on "Download client configuration" on the final screen

DOWNLOAD CLIENT CONFIGURATION


6. Save the file and rename it to "client_secret.json"

7. Play at the root of the Oblivion folder

8. Go to your Google Drive and create a folder with any name and access it

9. Copy the folder ID contained in the URL

 https://drive.google.com/drive/folders/1Zsnh9_79r3hCdGiHeV_sZWWhKVwckGPzv

10. Go to the oblivion configuration menu 

11. Click on "Authentication" in the left corner

Authentication

12. paste the ID in the "ID folder" field

Google Drive

ID folder

Before authenticate your Google Drive you need to activate the API of your Google Drive account and put the token in root folder of Oblivion.

13. Click on "Authenticate"

14. A google page will open and you will need to log in with your Google Drive account

15. If you see a screen titled "This app has not been verified": click "advanced" and "Access Quickstart"

16. Go back to the Oblivion configuration menu

17. enter the Google Drive folder ID and click "Authenticate"


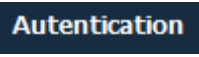
Google Drive

ID folder

Before authenticate your Google Drive you need to activate the API of your Google Drive account and put the token in root folder of Oblivion.

After that, it should contain a file named “Authentication.txt” at the root of your Google Drive. This file can be deleted without problems.

3.2.3.1 Changing destination folder

1. Go to the oblivion configuration menu 
2. Click on “Authentication” in the left corner 
3. In the “ID folder” field, enter the desired Google Drive folder ID

Google Drive

ID folder

Before authenticate your Google Drive you need to activate the API of your Google Drive account and put the token in root folder of Oblivion.

4. Click on “Authenticate”

5. Click on “save”


3.2.4 Telegram

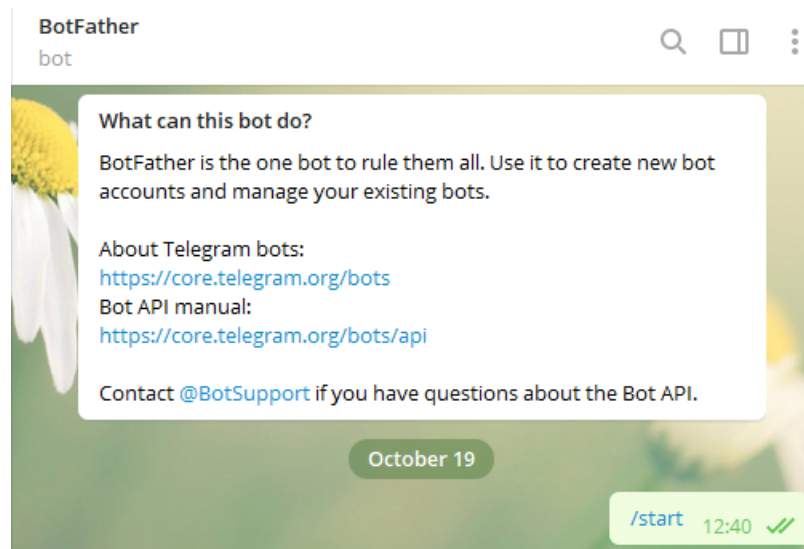
The Telegram Bot is made from BotFather and has no cost.

3.2.4.1 Creating Bot

1. Go to Telegram and search for BotFather



2. The official BotFather will have the checked symbol  next to the name
3. Type in the chat “/ start”



4. Type “/ newbot”
5. After that enter the name of your bot (there cannot be any bot with the same name)
6. Finally, BotFather will respond with your bot's Token, it is important that you keep that Token saved in a safe place (don't share the Token with anyone)

Use this token to access the HTTP API:

1353559052:AAH_XE32VQf

Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

3.2.4.2 Configuring Bot

1. Go to Settings 
2. Enter your Bot's Token in "Telegram bot"

Others

☒ Telegram bot

3. After that click on "save"

Save

4. Go to "Authentication" **Authentication**
5. Enter your Token in "API Bot"

Telegram

API bot


Authenticate

Before authenticate your Telegram bot you need to send a message to bot.

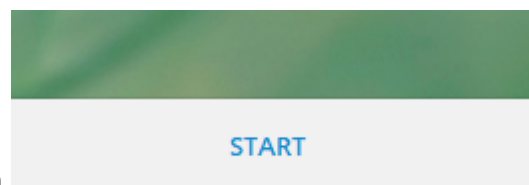
6. Click on "Authenticate"
7. After that click on "save"
8. Go to the Telegram search field and enter the name of your Bot

☰ ✕

Global search results

 **Oblivion**
@xOblivion_bot

9. Select the bot and click on the "START" option
10. After that send any message.



IMPORTANT: Your Telegram Bot will only notify users who clicked "START" and sent a message to him.

3.2.4.3 Enable notification

1. Go to settings



2. Click on "Notification"

Notification

3. In the "Telegram" section, check the option "Send scan result by Telegram"

☐ Notify by telegram

3.2.4.4 Configuring recipients

1. Go to the Telegram search field



get my id



Get My ID

2. Type "get my id" and select the first result
3. Send "get my id" in chat

4. The bot will send two IDs, copy "Your current ID"

Your user ID: 12345678
Current chat ID: 98765432 14:30

5. Go into setup



6. Click "Notification" in the left corner

Notification

7. In "Telegram" type the recipient IDs in the field on the left. It is important that IDs are separated by ";" and at the end of the IDs is contained ";", otherwise Oblivion may have problems sending the notification by Telegram.

Telegram



☐ Notify by telegram

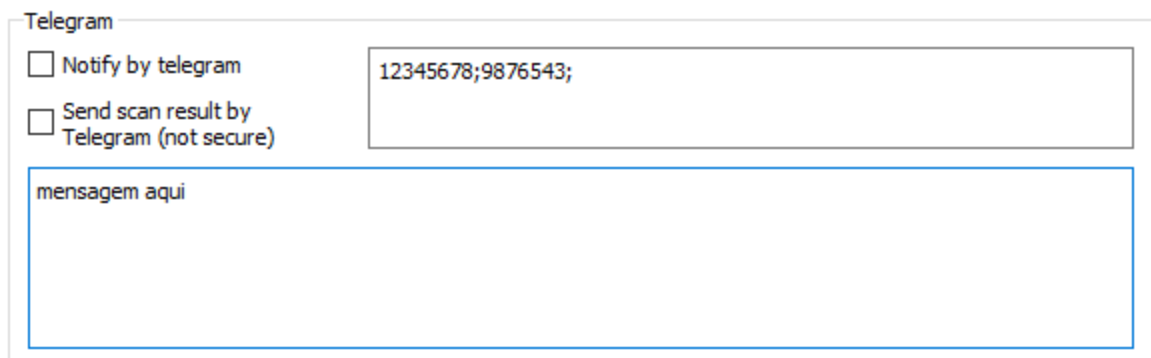
☐ Send scan result by Telegram (not secure)

12345678;9876543;

The sending of notices by the Bot telegram is done through the user's Telegram account ID.

3.2.4.5 Notification message

1. Go into setup 
2. Click "Notification" in the left corner 
3. In the “Telegram” area, type the desired message in the blank field



Telegram

☐ Notify by telegram

☐ Send scan result by Telegram (not secure)



12345678;9876543;

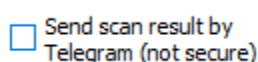
mensagem aqui

IMPORTANT: It is not recommended to use accents in the message

3.2.4.5.1 Message with results

To notify Telegram IDs with the results of the analyzes made:

1. Go into setup 
2. Click "Notification" in the left corner 
3. In the “Telegram” section, check the option “Send scan result by Telegram”



☐ Send scan result by Telegram (not secure)

IMPORTANT: Very long analysis results may present a problem at the time of sending, due to the limited character of the message body. It is advisable to avoid using this option.

3.2.5 Encryption key

By default Oblivion encrypts file with a standard key, it is highly recommended to change that key:

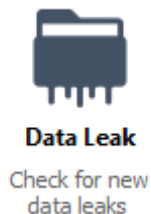
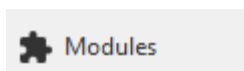
1. Access the root of the Oblivion folder
2. Access the “etc” folder
3. Open the file "key_crypt.txt"
4. Delete the default key and write a new key, without using spaces

IMPORTANT: If you have any problems with the new spelling, write a new key keeping the special characters and changing only the letters and numbers of the standard key, keeping the capital letters and capital letters.

3.3 Analysis

3.3.1 Configuring credentials

1. Go to modules in the left corner
2. Click on “Data Leak”



3. A screen will open to enter the credentials
4. Click on any cell and enter the credential
5. After that click on “save”

Passwords

Passwords column.

Emails

Email column

Documents

It will be the column where the user will put any type of data that is not a password or an email, for example an IP address, a domain etc.

3.3.1.1 Databases

Oblivion uses a “.db” database, it is located at the root of the Oblivion folder. It is important that the database file is named “data.db”.

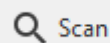
All data processing is done through SQLite.

The database format consists of:

- Data (Table)
 - email TEXT (column)
 - TEXT password (column)
 - TEXT document (column)

3.3.2 Configuring analysis

In “scan” located on the left you can access the scan part

A screenshot of the Oblivion configuration window. At the top, there are three tabs: "Configurations", "Modules", and "Format". The "Configurations" tab is selected. Below the tabs, there are two sections. The first section is titled "Scan" and contains a "Delay scan(s)" input field with the value "3,00" and a "Loop scan" checkbox. The second section is titled "Completion" and contains two checkboxes: "Close after conclude" and "Turn off after conclude".

Configurations Modules Format

Scan

Delay scan(s) 3,00

☐ Loop scan

Completion

☐ Close after conclude

☐ Turn off after conclude

Delay scan (s): time between analysis in loop mode

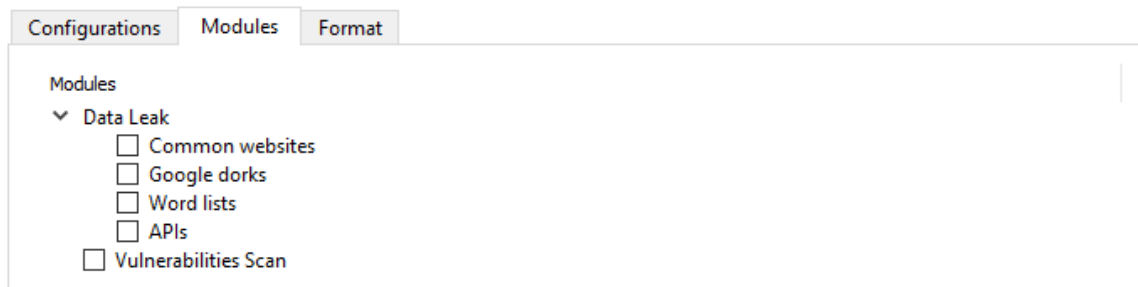
Loop scan: enable loop analysis, where Oblivion will scan until it finds any leaked credentials.

Close after concludes: close Oblivion after analysis is complete

Turn off after concludes: turn off Oblivion after completing the analysis

3.3.2.1 Modules

By clicking on “modules” you will have access to the modules and functionalities area:



For more information on the modules, access item 1.3.

3.3.2.2 Formats

If Oblivion detects any credential leak, it will generate a file with the report. The supported file types are:

- ✓ .txt
- ✓ .docx
- ✓ .pdf
- ✓ .xlsx
- ✓ .json
- ✓ .html
- ✓ .xml
- ✓ .db

To generate the desired file you must check the responsible checkbox.

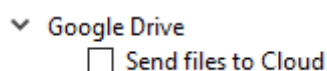
Raw data (.txt): generate raw data

Occult password: hide end of passwords with asterisks

Encrypt: generate the file in encrypted form

3.3.2.3 Sending to Google Drive

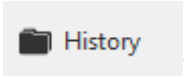
To send the results automatically to Google Drive check the option “Send file to Cloud”.



3.3.3 Results

All files with the results are in the “Oblivion” folder within the Documents folder.

3.4 History

In “History” in the left corner  will have all the analyzes performed by Oblivion.

3.4.1 Clear the history

Open the file “Oblivion/etc/logs/activity.txt” and delete all the content. It is important that you do not delete this file.

3.5 Scheduling of analyzes

Analysis scheduling is done by Windows Scheduler in the Oblivion Windows version and by Crontab in the Oblivion Linux version.

To schedule an analysis click on “Schedule” in the left corner  [Schedule](#).

1. Select the date and time

Date:

Time:

2. Enter the name of the analysis

Date:

Time:

3. Select the parameters (for more details, access items 3.3.2, 3.3.2.1, 3.3.2.2 and 3.3.2.3)


Results


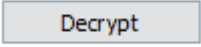
- ▼ Modules
 - ☐ Vulnerabilities Scan
 - ▼ Data Leak
 - ☐ Common websites
 - ☐ Google dorks
 - ☐ Word lists
 - ☐ APIs
 - ▼ Generate results
 - ▼ Local file
 - ☐ Raw data (.txt)
 - > ☐ .txt
 - > ☐ .docx
 - > ☐ .pdf
 - > ☐ .xlsx
 - > ☐ .json
 - > ☐ .html
 - > ☐ .xml
 - > ☐ .db
 - ▼ Google Drive

4. Click on “create”

Create

3.6 Decrypt

In the left corner click on “Decrypt” .

1. If you are receiving an encrypted file with a different key, you must put the key in the “Key” field (to learn more about the encryption key, see item 3.2.5)
2. Enter the path of the item in the “path” field or click  and select the file
3. click in 

Oblivion Server -----

4.1 Installation

Python 3.8.6 installation is recommended prior to Oblivion installation:

(<https://www.python.org/downloads/release/python-386/>)

4.1.1 Windows

1. Download the Oblivion repository on GitHub (<https://github.com/loseys/Oblivion>)
2. Rename the “Windows” folder to “Oblivion”
3. Go to the “Windows” folder and run the “install.py” file.
4. Execute OblivionServer.py

4.1.2 Linux installation

6. \$ git clone <https://github.com/loseys/Oblivion>
7. \$ mv Linux Oblivion
8. \$ cd Oblivion
9. \$ sudo python3.8 install.py
10. \$ sudo python3.8 OblivionServer.py

4.2 Configuration

4.2.1 General

To access general server settings go to:

`"/Oblivion/etc/serverx/config/config_server.py"`

api_return: enable or disable json return after making a request to the Oblivion server

db_name_f: name of the database file that the server will use, this file is located in the root of the Oblivion folder

SituationIntelx: enables or disables the use of the IntelX API on the Oblivion server

situationHaveIPwned: enables or disables the use of the Have I Been Pwned API on the Oblivion server

situationScylla: enables or disables the use of the Scylla API on the Oblivion server

id_f3: notification email

id_f4: notification email password

To change Oblivion's operating port, access the `"/Oblivion/OblivionServer.py"` file and change line 314:

A screenshot of a code editor showing two lines of Python code. The first line is `if __name__ == "__main__":` and the second line is `app.run(host='0.0.0.0', port="5000")`. The text is white on a dark blue background.

```
if __name__ == "__main__":  
    app.run(host='0.0.0.0', port="5000")
```

(image from line 314 of the file `"/Oblivion/OblivionServer.py"`)

To leave the server running locally, delete the `"host = '0.0.0.0'"`:

A screenshot of a code editor showing two lines of Python code. The first line is `if __name__ == "__main__":` and the second line is `app.run(port="5000")`. The text is white on a dark blue background.

```
if __name__ == "__main__":  
    app.run(port="5000")
```

(image from line 314 of the file `"/Oblivion/OblivionServer.py"`)

4.2.1.1 Secret Key

To make any type of request for the Oblivion Server, it will be necessary to pass a security key in the URL or in the Headers:

localhost: 5000/oblivion /<secret key>/<parameters>
"Key": "<secret key>"

Accessing the file `"/Oblivion/etc/serverx/config/keys.txt"` you can add the security key. It is interesting that the security key has uppercase strings, lowercase strings, letters and characters for security reasons.

Initially each key will be associated with a user. In addition, it is important that each key is placed on a different line, following the example below:

A screenshot of a terminal window showing the GNU nano 4.8 text editor editing the file keys.txt. The file contains three lines of alphanumeric strings: 8dAMM9sPfJDRhDYA9xgqC4Fk3Ks1D, kNS\$1w01ncAL2ns1fMC012Awqxqw0, and \$e0ns1IX5Ns8KSz1%1gPakOPQW1C@.

```
GNU nano 4.8 keys.txt
8dAMM9sPfJDRhDYA9xgqC4Fk3Ks1D
kNS$1w01ncAL2ns1fMC012Awqxqw0
$e0ns1IX5Ns8KSz1%1gPakOPQW1C@
```

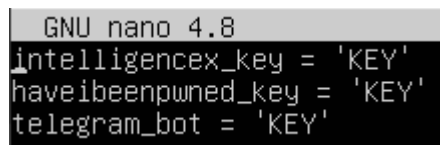
(image from the `"/Oblivion/etc/serverx/config/keys.txt"` file)

4.2.1.2 APIs

To configure the API keys you must first access:

`"/Oblivion/etc/api/keys_db.txt"`

Inside the file, you must replace `"KEY"` with your API key, keeping the single quotes at the beginning and end of the key

A screenshot of a terminal window showing the GNU nano 4.8 text editor editing the file keys_db.txt. The file contains three lines of code: intelligencex_key = 'KEY', haveibeenpwned_key = 'KEY', and telegram_bot = 'KEY'.

```
GNU nano 4.8
intelligencex_key = 'KEY'
haveibeenpwned_key = 'KEY'
telegram_bot = 'KEY'
```

(image from the `"/Oblivion/etc/api/keys_db.txt"` file)

4.2.1.2.1 Enabling APIs

By accessing “Oblivion/etc/serverx/config/config_server.py” you can define which APIs will be used by the server. In the “/API” field you can choose between “False” to disable the use of the API and “True” to activate the use of the API.

```
# /APIs
# Enables or disables the API.
situacaoIntelx = False
situacaoHaveIPwned = False
situacaoScylla = True
```

(image from the “/Oblivion/etc/serverx/config/config_server.py” file)

4.2.2 G-Mail

You must first activate the option “Access to less secure app” in the security options of your G-Mail (<https://myaccount.google.com/lesssecureapps>).



(image with the option “Access to less secure app” active)

4.2.2.1 Enable notifications

By accessing “Oblivion/etc/parameters.txt” you can activate simple and personalized notification. To activate or deactivate, simply change from “no” to “yes”.

```
GNU nano 4.8
email_notification:no
email_body:no
telegram_notification:no
telegram_body:no
```

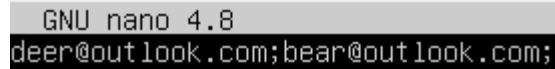
(image from the “/Oblivion/etc/parameters.txt” file)

email_notification: enable or disable sending email notification

email_body: enable or disable the sending of notification with the analysis results in the message body

4.2.2.2 Configuring recipient

In `"/Oblivion/etc/notification/email/emails.txt"` you can add the emails you want to notify. Each e-mail must contain `“;”` as a separator. After the last email, you must have `“;”`.



```
GNU nano 4.8
deer@outlook.com;bear@outlook.com;
```

(image from the `"/Oblivion/etc/notification/email/emails.txt"` file)

4.2.2.3 Notification message

To change the notification message, go to `"/Oblivion/etc/notification/email/body.html"`. The message will be sent in HTML format.



```
GNU nano 4.8 body.html Modified
<h1 style="color: #5e9ca0;"><span style="color: #003366;">Oblivion</span></h1>
<h4 style="color: #2e6c80;"><span style="color: #ff0000;">Aten&ccedil;&atilde;o:</span></h4>
<p>Suas cred&ecirc;ncias podem estar em risco: acesse seu aplicativo Oblivion ou <a href="https://>
<p>Caso esteja com algum problema contate o administrador.</p>
<p>&nbsp;</p>
```

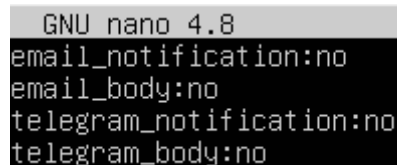
(image from the `"/Oblivion/etc/notification/email/body.html"` file)

4.2.4 Telegram

To learn more about creating a bot for Telegram, access item 3.2.4.1.

4.2.4.1 Enable notifications

In the file `"/Oblivion/etc/parameters.txt/"` you will find the following content:



```
GNU nano 4.8
email_notification:no
email_body:no
telegram_notification:no
telegram_body:no
```

(image from the `"/Oblivion/etc/parameters.txt/"` file)

telegram_notification: enable or disable Telegram notification

telegram_body: enable or disable sending analysis results by message

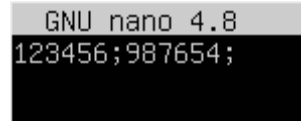
To activate or deactivate you must change from `“no”` to `“yes”`.

4.2.4.2 Configuring recipients

In “/Oblivion/etc/notification/users.txt” you can add the Telegram account IDs that will be notified. It is important to note that just placing the Telegram account number will not work, it needs to be the account ID.

To find out how to acquire the Telegram account ID, access item 3.2.4.4.

Each Telegram ID must be separated by “;”. At the end of all the IDs must contain a “;”.



```
GNU nano 4.8 users.txt
123456;987654;
```

(image from the “/Oblivion/etc/notification/users.txt” file)

4.2.4.3 Notification message

In the file “/Oblivion/etc/notification/message.txt” you can change the Telegram notification message. The use of accents in letters is not recommended.



```
GNU nano 4.8 message.txt
Atencao: suas credenciais podem estar em risco: acesse seu aplicativo Oblivion ou clique aqui (http
```

(image from the “/Oblivion/etc/notification/message.txt” file)

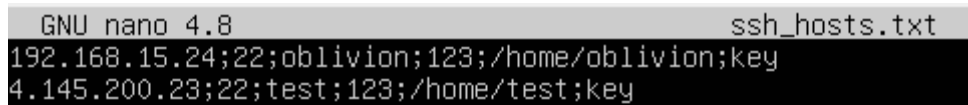
4.2.5 SSH

To add an SSH path, go to “/Oblivion/etc/serverx/config/ssh_hosts.txt”.

Syntax:

IP; PORT; LOGIN; PASSWORD; DESTINATION_FOLDER; SECRET_KEY

Each SSH path must have a Secret Key associated with it, to find out more about Secret Key see item 4.2.1.1



```
GNU nano 4.8 ssh_hosts.txt
192.168.15.24;22;oblivion;123;/home/oblivion;key
4.145.200.23;22;test;123;/home/test;key
```

(image from the “/Oblivion/etc/serverx/config/ssh_hosts.txt” file)

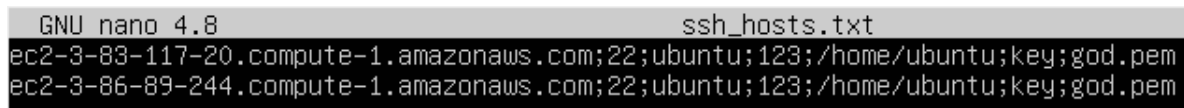
IMPORTANT: If the results are not being sent to the destination folder, use the user's home, for example “/home/test”.

4.2.5.1 SSH ec2

To add an SSH path to an AWS ec2 you first need to copy your .pem key to the “/Oblivion/etc/serverx/pems” directory. After that open the SSH hosts file “/Oblivion/etc/serverx/config/ssh_hosts.txt:

Syntax:

PUBLIC_DNS; PORT; LOGIN; PASSWORD; FOLDER_DESTINATION; SECRET_KEY;
NAME_PEM_FILE



```
GNU nano 4.8 ssh_hosts.txt
ec2-3-83-117-20.compute-1.amazonaws.com;22;ubuntu;123;/home/ubuntu;key;god.pem
ec2-3-86-89-244.compute-1.amazonaws.com;22;ubuntu;123;/home/ubuntu;key;god.pem
```

(image from the “/Oblivion/etc/serverx/config/ssh_hosts.txt” file)

In the case of the image above, the password was entered only for the purpose of Syntax, but it was not necessary.

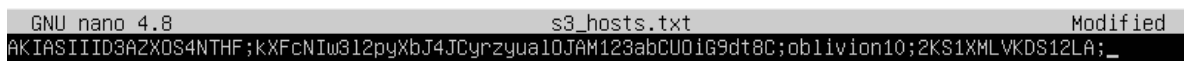
IMPORTANT: It is important that the public ec2 DNS is used. If the results are not being sent to the destination folder, use the user's home, for example “/ home / test”.

4.2.6 AWS S3

To add an S3 bucket go to “/Oblivion/etc/serverx/config/s3_hosts.txt”.

Syntax:

ACCESS_KEY; SECURITY_KEY; BUCKET_NAME; SECRET_KEY.



```
GNU nano 4.8 s3_hosts.txt Modified
AKIASIIID3AZXS4NTHF;kXFcNIw812pyXbJ4JCyrzyua10JAM123abCU0iG9dt8C;oblivion10;2KS1XMLVKDS12LA;_
```

(image from the “/Oblivion/etc/serverx/config/s3_hosts.txt” file)

Access key: bucket access key

Security key: bucket security key

Bucket name: bucket name

Each parameter must be separated by “;” and at the end it should present “;”. It is necessary that the customer has an associated Secret Key, to learn more about the Secret Key access item 4.2.1.1.

4.2.7 Databases

Oblivion uses a “.db” database, it is located at the root of the Oblivion folder. It is important that the database file is named “data.db”.

All data processing is done through SQLite.

The database format consists of:

- Data (Table)
 - email TEXT (column)
 - TEXT password (column)
 - TEXT document (column)

You can choose to either generate a database with the Oblivion Client or create your own .db file following the tables and columns above.

4.3 Requisition

4.3.1 URL

To make a request by URL you must follow the following Syntax:

`xxxx:port/oblivion/<security key>/<parameters>`

Parameter list:

- &EM "email@outlook.com" (uses a personalized email, must be in double quotes)
- &PW "password123" (uses a personalized password, must be in double quotes)
- &DM "www.teste.com" (uses personalized data, needs double quotation marks)
- &% loop (keep looping until you find some data leak)
- &common_web (use the Common Websites feature)
- &google_dorks (use the Google Dorks feature)
- &wordlists (use the Word List feature)
- &raw data (save raw data from results obtained)
- &api (use API functionality)
- &<file type>_f (generate file)
- &<file type>_ocult (generate file with semi hidden passwords)
- &<file type>_cript (generate encrypted file)
- &gdrive (send files to Google Drive)
- &aws_s3 (send files to AWS s3)
- ssh (send files over SSH)

Supported file types:

- ✓ .txt
- ✓ .docx
- ✓ .pdf
- ✓ .xlsx
- ✓ .json
- ✓ .html
- ✓ .xsl
- ✓ .db

Operation:

txt_f> generates a common .txt file

txt_f + txt_cript> generates an encrypted .txt file

txt_f + txt_cript + txt_ocult> generates an encrypted .txt file with occulted passwords

4.3.1.1 Examples

Using the APIs functionality. Generating a .txt file:

4.25.192.8:57000/oblivion/DRhDYA9xgqC4Fk3Ks1D/&api&txt_f

Using the Common Websites functionality and the APIs functionality. Generating the results in an encrypted .docx:

4.25.192.8:57000/oblivion/DRhDYA9xgqC4Fk3Ks1D/&api&common_web&docx_f&docx_cript

Using Word List functionality with a personalized password. Generating the results in a .db:

[4.25.192.8YS000/oblivion/DRhDYA9xgqC4Fk3Ks1D/&wordlists&PW"password123"&db_f](http://4.25.192.8YS000/oblivion/DRhDYA9xgqC4Fk3Ks1D/&wordlists&PW)

Using the APIs functionality. Generating the results in an .html and sending via SSH and to the S3 bucket:

4.25.192.8YS000/oblivion/DRhDYA9xgqC4Fk3Ks1D/&api&html_f&ssh&aws_s3

Using the APIs functionality. Generating the results in a txt file and sending it to Google Drive:

4.25.192.8YS000/oblivion/DRhDYA9xgqC4Fk3Ks1D/&api&txt_f&gdrive

4.3.2 Header

To make a request by URL you must follow the following Syntax:

```
h = {  
    'Key': '<Security_key>',  
    '<parameter>': '<condition>'  
}  
  
rr = requests.get ('http: //4.25.192.8:5000/oblivion/api/&', headers = h)  
print (rr.text)
```

Parameter list:

- '&Em': '{email@outlook.com}' (uses a personalized email, must be in double quotes)
- '&Pw': '{password123}' (uses a custom password, must be in double quotes)
- '&Dm': '{www.teste.com}' (uses personalized data, must be in double quotes)
- 'loop': 'True' (keep looping until you find some data leak)
- 'common_web': 'True' (use the Common Websites feature)
- 'google_dorks': 'True' (use the Google Dorks feature)
- 'wordlists': 'True' (use the Word List feature)
- 'raw_data': 'True' (save raw data from results obtained)
- 'api': 'True' (use API functionality)
- '<file type> _f': 'True' (generate file)
- '<file type> _ocult': 'True' (generate file with semi-hidden passwords)
- '<file type> _cript': 'True' (generate encrypted file)
- 'gdrive': 'True' (send files to Google Drive)
- 'aws_s3': 'True' (send files to AWS s3)
- 'ssh': 'True' (send files over SSH)

Supported file types:

- ✓ .txt
- ✓ .docx
- ✓ .pdf
- ✓ .xlsx
- ✓ .json
- ✓ .html
- ✓ .xsl
- ✓ .db

Operation:

txt_f> generates a common .txt file

txt_f + txt_cript> generates an encrypted .txt file

txt_f + txt_cript + txt_ocult> generates an encrypted .txt file with occulted passwords

4.3.2.1 Examples

Using the APIs functionality. Generating the results in a pdf file and sent via SSH and to the S3 bucket.

```
h = {  
  'Key': 'DRhDYA9xgqC4Fk3Ks1D',  
  'pdf_f': 'True',  
  'api': 'True',  
  'aws_s3': 'True',  
  'ssh': 'True'  
}
```

Using the Common Website functionality with a personalized email. Generating the results in an encrypted .pdf file.

```
h = {  
  'Key': 'DRhDYA9xgqC4Fk3Ks1D',  
  'pdf_f': 'True',  
  'pdf_cript': 'True',  
  'common_web': 'True',  
  '&Em': '{ example@gmail.com }'  
}
```