George's Vacation - CSCG 2023

This writeup explains how I solved the challenge George's Vacation during the CSCG qualifiers of 2023.

- **Name:** George's Vacation
- **Category:** Misc
- **Difficulty:** Medium
- **Flag:** `CSCG{g30rg3_st4lk3r_4_r34l}`
- **Description:** George had an awesome vacation. Look here:

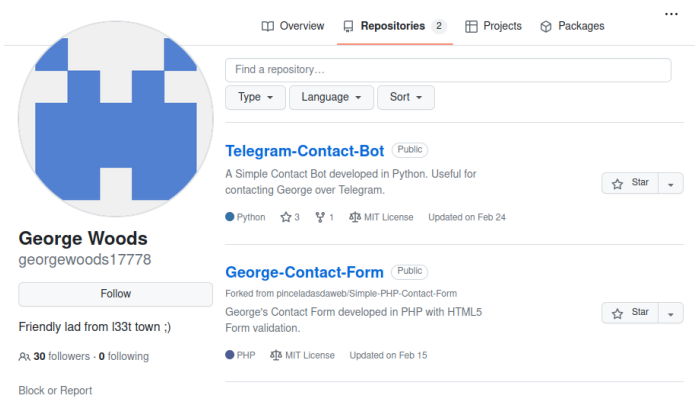
maybe you can find a secret he left behind?

This looks like a typical OSINT challenge, so let's figure out as much as we can about George.

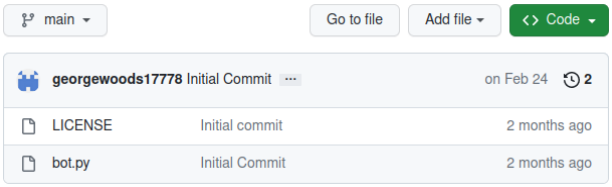
The first thing we notice is the sticker with QR codes on the right side of the picture. The bottom one is unreadable, but the top one encodes the following Telegram URL: https://t.me/george_contact_bot.

In addition to `/start`, the Telegram bot provides a menu with three commands: `/help`, `/message` and `/legal`. While `/message` seems interesting, the bot always sends the same reply, no matter what we send to George.

However, if we use `/legal`, the bot sends us a link to George's GitHub account!



George has two repositories: the source code of the Telegram bot and a PHP contact form. Let's look at the Telegram bot first. It has two commits: once which adds the license and one which adds the bot code. The `/message` command simply writes our message into a file, but there is no way that we can access this file. There are also no hidden commands. Seems like there is nothing useful here.




```python
def message(update, context):
    """Send a message when the command /message is issued."""
    time.sleep(1)
    logi = open("messages_for_george.txt","a")
    logi.write(update.message.text)
    logi.write("\n")
    logi.close()
    update.message.reply_text('Thank you for your message!')
```

The other repository is forked from `pinceladasdaweb/Simple-PHP-Contact-Form`. Because `pinceladasdaweb` has been active on GitHub for much longer, he is probably unrelated to the challenge. George has added two commits to his fork, but both are just some text changes, although one of them suggests that we should find his email address.

```
 7   7        'emails' => [
 8       -        'to'   => '',
 9       -        'from' => ''
     8   +        'to'   => '<todo updated on server>',
     9   +        'from' => '<todo update on server>'
10  10        ],
```

Git commits usually include the name and email address of their author. While GitHub doesn't show the email address in its UI, we can clone repositories locally to inspect them.

```
yannik@lenovo:~$ git clone https://github.com/georgewoods17778/George-Contact-Form
Cloning into 'George-Contact-Form'...
remote: Enumerating objects: 335, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 335 (delta 4), reused 11 (delta 3), pack-reused 315
Receiving objects: 100% (335/335), 62.50 KiB | 3.12 MiB/s, done.
Resolving deltas: 100% (120/120), done.
yannik@lenovo:~$ cd George-Contact-Form/
yannik@lenovo:~/George-Contact-Form$ git log
commit b7cd0a7a32ff1dcc84cc4c69d4d7a7b55e375ce7 (HEAD -> master, origin/master, origin/HEAD)
Author: George Woods <idontlikedoxxing@thusfakeemail.com>
Date:   Wed Feb 15 21:29:24 2023 +0100

    some more customizations

commit 871b089c2cb7ae54c223468f302e8ee05a457fb9
Author: George Woods <idontlikedoxxing@thusfakeemail.com>
Date:   Wed Feb 15 21:03:09 2023 +0100

    Customize my contact form

commit 5c0db9860349817f298ac6dc267f0d34e9dda653
Merge: 7a0227a e3ba092
Author: Pedro Rogério <pinceladasdaweb@hotmail.com>
Date:   Mon Oct 4 17:50:35 2021 -0300

    Merge pull request #44 from Watn3y/master
```

In the contact form repository, we see that the commits are made by `idontlikedoxxing@thusfakeemail.com`. This email address is obviously fake (the domain does not even exist), but it could be another hint that we have to find George's email address.

This is probably where most people get stuck. The Telegram bot repository is uninteresting, George does not show his email address on his profile, and has no activity on GitHub at all outside of his repositories. Seems like a dead end.

However, there is one more way to figure out what George has been doing: the GitHub API. Specifically, if we visit https://api.github.com/users/georgewoods17778/events we get a detailed list of his activities. One of them immediately stands out:


```
▼ 5:
    id:                 "27117944238"
    type:               "PushEvent"
  ▶ actor:              {…}
  ▶ repo:               {…}
  ▼ payload:
    ▼ commits:
      ▼ 0:
          sha:          "74e02d56239bee19da119f4a353bfef734645a18"
        ▼ author:
            email:      "idontlikedoxxing@thusfakeemail.com"
            name:       "George Woods"
          message:      "Message to my email"
          distinct:     true
        ▼ url:          "https://api.github.com/repos/georgewoods17778/George-Contact-Form/commits/74e02d56239bee19da119f
                        a353bfef734645a18"
    public:             true
    created_at:         "2023-02-15T20:09:55Z"
```

We have not seen this commit before. We follow the URL to get more details about the commit, and find that George added his Gmail address to the repository in this commit!

```
@@ -5,8 +5,8 @@
          'prefix' => '[Contact Form]'
      ],
      'emails' => [
-         'to'   => '',
-         'from' => ''
+         'to'   => 'george.woods17778@gmail.com',
+         'from' => 'george.woods17778@gmail.com'
      ],
      'messages' => [
          'error'   => 'There was an error sending your message, please try again later.',
@@ -20,4 +20,4 @@
          'message'  => 'Message',
          'btn-send' => 'Send'
      ]
-];
\ No newline at end of file
+];
```

Now, GHunt is a tool that scrapes Google APIs for information about a user. We use it as follows: `ghunt email george.woods17778@gmail.com`. Apparently George has uploaded two photos on Google Maps.

```
Maps data

Profile page : https://www.google.com/maps/contrib/107686015845164860810/reviews

[Statistics]
Photos : 2

[-] Reviews are private.
```

Let's check them out by visiting his profile page.

One of them contains the flag!

Summary

- The QR code on the right side of the image leads us to George's Telegram bot.
- The `/legal` command of the Telegram bot reveals his GitHub username.
- Through the GitHub API, we find a commit to one of his repositories that reveals his Gmail address.
- GHunt tells us that George published two photos on Google Maps.
- One of his photos contains the flag!

Mitigation

Apparently, George wants to keep his coding and personal life separate. He published his Gmail address on GitHub, and removed it from the commit history two minutes later. However, he wasn't aware that it is still visible through the GitHub API.

GitHub explains how sensitive information can be removed from a repository here, but it also provides an important warning:

> Once you have pushed a commit to GitHub, you should consider any sensitive data in the commit compromised. If you have committed a password, you should change it. If you have committed a key, generate a new one. Removing the compromised data doesn't resolve its initial exposure, especially in existing clones or forks of your repository.

This does not only apply to GitHub, but to any information that is published on the internet. Although some steps can be taken to remove information from the internet, one can never know for sure that the information is gone, and that no one was able to obtain a copy.

Therefore, to prevent this, one must make sure that sensitive information is never published on the internet. A good rule of thumb is: think before you act. Organizations can also train employees to be more careful, implement organizational security policies, and monitor outgoing traffic for sensitive information before it is transmitted.