**Image for the blog background:**

<https://user-images.githubusercontent.com/18099289/77641182-eb620c80-6f5b-11ea-9be5-02639c94b141.png>

**Title:**

How a simple file can prevent your name to end on Twitter alongside “Contact me urgently!”.

**Abstract:**

# Once upon a time in InfoSec world…

Our story begins on a Friday evening, an InfoSec folk pass an order on a food web portal in order to take a romantic break with his sweet heart. He selects dishes and click on the “*Checkout”* button, however, instead of receiving the expected checkout page, he receives the following “nicely” SQL error page:

|  |
| --- |
| You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near “**Sushi**”. |

Our folk said to himself: “*OK, I will drop an email with the print screen alongside actions performed to the security contact point in order to inform them of the issue and how to reproduce it…*”

# Where is Wally?

Our folk take a look at the homepage then contact page and so on…No way to find the “security” contact point, only a general contact form is available.

Our folk want to enjoy the planned romantic break, so, he filled the contact form with the information summary without any print screen, the contact form was not allowing file attachment, and clicked on the “*Send”* button. After, he retry to pass its order, removing this time the character triggering the SQL error, and achieve to finalize it with success.

For him this event was over….As you can guess, its messages about the security issue end in /dev/null…

# Why a generic contact cannot fulfill the job?

A generic contact form (source [1]), most of the time, is designed to transfer public/non-sensitive information from a prospect to the marketing or sales department of a company. Conversely, when a person notify a company about a security vulnerability, the person want to fulfill the following objective:

1. Confidentiality of the information as well as the transfer of it, in order to only disclose the vulnerability to people that are in charge of the security. The goal is to give the information to people in position to take effective decision regarding the vulnerability analysis and remediation.
2. Responsiveness from the company regarding the fact that they have received and understand the problem. Sometime, the vulnerability was already discovered and exploited by an attacker so it become a race against time to close the hole…

A generic contact form do not fulfill the two objective above because it not ensure the confidentiality of the information (and its transfer) and initiate an “internal ping-pong” game in the target company between the recipient behind the contact form and the team expected for handle security incident.

It is for these reasons that a visible direct contact point to the team handling security incident is important.

At Excellium, when we discover a vulnerability and we notify the owner of the software using a by non-specialized channel, we are often involved despite us, in the famous “ping-pong” game mentioned above. It became a real hell to make the information reach the right department/person in within days instead of weeks/months…

Real response received, by our CSIRT (source [13]), from a vendor following an email notification regarding a critical vulnerability:

“*You send us a protected archive alongside your mail? We thought it was spam from which your email was permanently deleted.*”

# Security.txt file to the rescue

As this situation, occur often, a group of people decided to launch an initiative in order to address this issue. The objective was to define a way to indicate how to report a security issue in a standardized and simple way.

The **security.txt** Internet draft was born (source [2]).

# What is this security.txt file?

As described on its homepage (source [3]), it is a text file (**text/plain** Internet media type) located in one of the following locations:

* Recommended location: **/.well-known/security.txt**
* Fallback: **/security.txt**

It contains the following information: source [3]

|  |  |  |
| --- | --- | --- |
| Field | Required | Description |
| **Contact** | Yes | A link or e-mail address for people to contact you about security issues. |
| **Expires** | Yes | The date and time when the content of the security.txt file should be considered stale. |
| **Encryption** | No | A link to a key which security researchers should use to securely talk to you. |
| **Acknowledgments** | No | A link to a web page where you say thank you to security researchers who have helped you. |
| **Preferred-Languages** | No | A comma-separated list of language codes that your security team speaks. |
| **Canonical** | No | The URLs for accessing your security.txt file. |
| **Policy** | No | A link to a policy detailing what security researchers should do when searching for or reporting security issues. |
| **Hiring** | No | A link to any security-related job openings in your organization. |

Sample security.txt file containing only the required fields:

|  |
| --- |
| Contact: mailto:emergency@excellium-services.com  Expires: Sun, 1 Jan 2025 00:00 +0100 |

Sample security.txt file containing more fields:

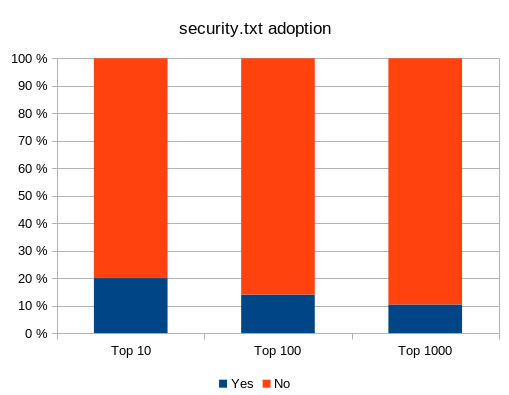
|  |
| --- |
| Contact: mailto:emergency@excellium-services.com  Expires: Sun, 1 Jan 2025 00:00 +0100  Encryption: https://excellium-services.com/assets/EMERGENCY\_PKEY.asc  Preferred-Languages: en,fr  Canonical: https://excellium-services.com/.well-known/security.txt  Policy: https://excellium-services.com/emergency-service/  Hiring: https://excellium-services.com/career-opportunities/ |

# Security.txt adoption

## Worldwide

Based on the chart below, from April 2020 (source [4]), in which an InfoSec folk analyzed the usage of the **security.txt** file against the Top Alexa 1 million most visited sites (source [5]), notes with astonishment that adoption of this feature, at the Internet level, is very low. Only 10% to 20% of sites were having such file.

It is perhaps due to a lack of knowledge/visibility of the initiative/project…



Source : https://community.turgensec.com/wp-content/uploads/2020/04/word-image.png

Statistics at the Internet level is interesting, however, what is the situation in our small and beloved country?

## Luxembourg

Based on the study mentioned in the previous section (source [4]), we decided to look at the adoption of the **security.txt** file on the LU domains.

To achieve that, we have leveraged the *Certificate Transparency* log (source [6]), as data sources, in order to extract a list of “**.lu**” domains.

In order to extract as much as possible records, the database free access offered by CRT.SH (source [7]) was used instead of the web API.

Basically, the following command line could be used to extract the records from the database (source [8]) as a text file.

The docker image of **postgres** (source [9]) was used in order to had access to the **psql** client without any third-party tools installation:

|  |
| --- |
| $ docker run -it --rm postgres psql -P pager=off -P footer=off -U guest -d certwatch --host crt.sh -c " select name\_value from certificate\_and\_identities where name\_type='san:dNSName' and right(name\_value, 3) = '.lu' limit 200" > data.txt |

However, as there are limitations in terms of execution time allowed for a query, a quick PowerShell script (source [10]) was created to retrieve the data by blocks of records.

Once the list of LU domains was generated, we have created a script (source [11]) in order to verify the presence of the **security.txt** file on the domains extracted.

**You have LU domains then relax:** Only safe HTTP classic GET requests, with a maximum of four requests by domain, were performed, so, we behaved ethically during this analysis.

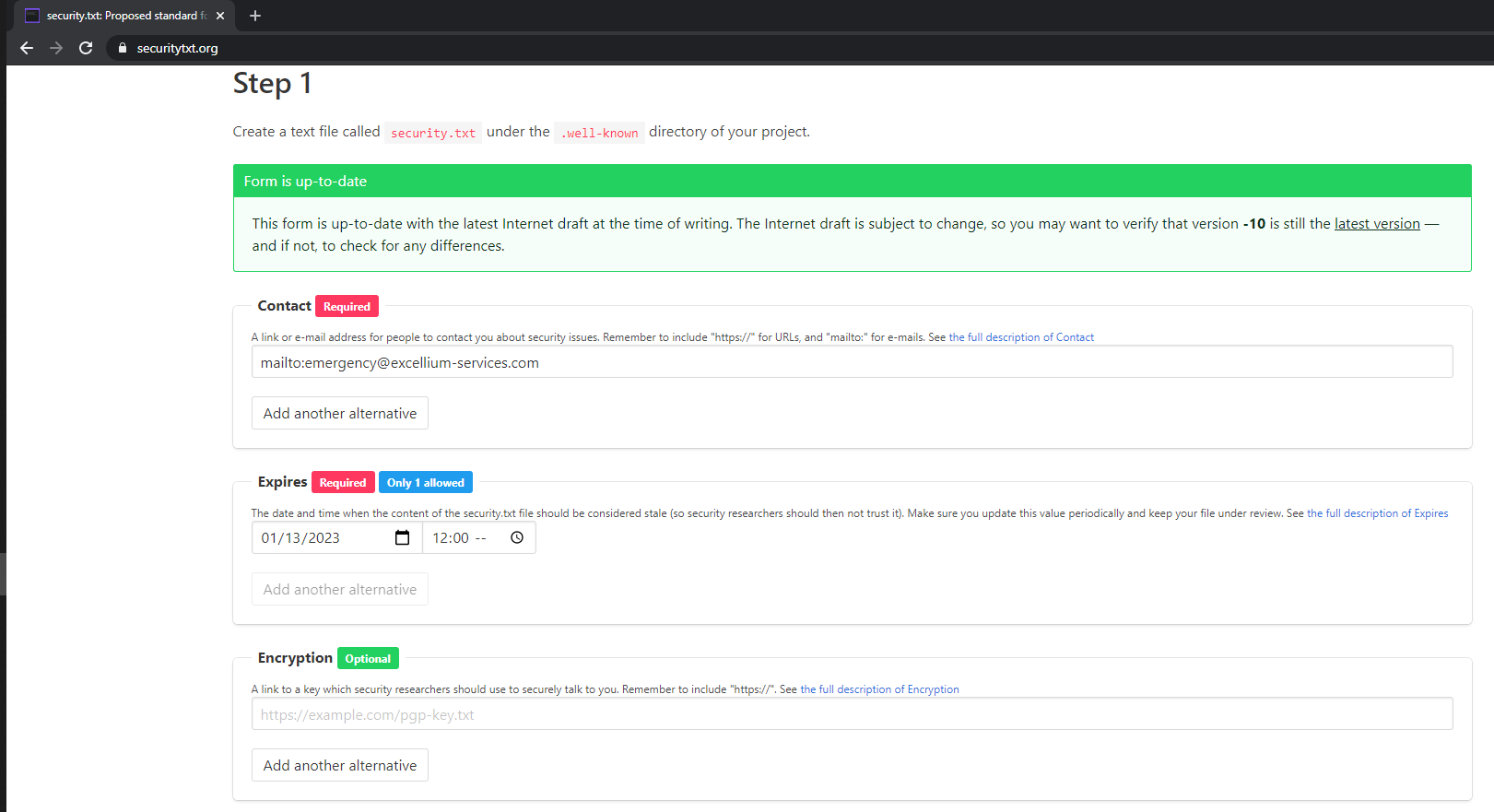
The result of the execution of the script was the following:

|  |
| --- |
| $ python generate-stats.py  [+] Prepare the list of domains...  1096 domains selected.  [+] Initialize DB...  [+] Process the list...  Testing domain: www.eurizoncapital.lu  [+] 1096 domains tested - Results:  ABSENT : 1096 |

Unfortunately, they were following the one from the study at Internet level; Luxembourgish domains are not better informed about this initiative…Therefore, this blog post seemed to be a good idea ☺

# It is never too late to add a security.txt file…

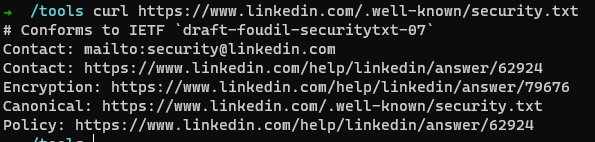
The team behind the project has anticipated this question. They have created an online tool to generate such file:



Therefore, the creation procedure is simple:

1. Fill the form for the fields that you want specify.
2. Click on the button “*Generate security.txt file*”.
3. Copy the content into a file named **security.txt** (thank you captain obvious☺).
4. Place the file in the location “**/.well-known/security.txt”** on your main web site.
5. Check that the file is reachable using an anonymous point of view (see below).

|  |
| --- |
| $ curl https://[YOUR\_DOMAIN]/.well-known/security.txt |

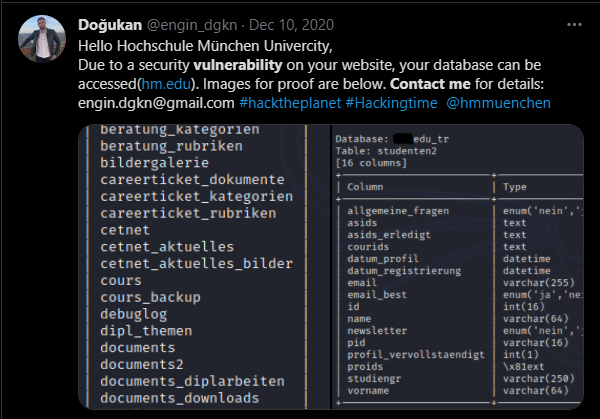


# The final word

Even if the current usage of the **security.txt** file is not wide spread, it is important to deploy such file on your main web site. This measure allow security researcher or other people to contact you discretely and securely if something goes wrong with one of your public exposed assets.

It will prevent you, at the same time, to be confronted with the following kind of advertising on social networks (source [12]):





# References

1. <https://www.autopolis.lu/fr/contact>
2. <https://tools.ietf.org/html/draft-foudil-securitytxt>
3. <https://securitytxt.org/>
4. <https://community.turgensec.com/security-txt-progress-in-ethical-security-research/>
5. <http://s3.amazonaws.com/alexa-static/top-1m.csv.zip>
6. <https://certificate.transparency.dev/>
7. <https://www.randori.com/enumerating-subdomains-with-crt-sh/>
8. <https://github.com/crtsh/certwatch_db/blob/master/sql/create_schema.sql>
9. <https://hub.docker.com/_/postgres>
10. [SCRIPT\_POWERSHELL]
11. [SCRIPT\_PYTHON]
12. <https://twitter.com/search?q=vulnerability%20contact%20me%20&src=typed_query>
13. <https://excellium-services.com/services/cert-xlm/>