

# Capture the Flag

Joël Gugger – Michael Caraccio – Nicolas Huguenin

WEM 2017 – MSE

# Sommaire

- Contexte et objectifs
- Fonctionnalités
- Données
- Technologies utilisées
- Indexation
- Front end
- Démonstration
- Améliorations possible

# Contexte

- WTF is a CTF?
  - Challenges Sécurité informatique
  - Crypto, Reverse Engineering, Web, ...
  - Online/Offline
- Site web : [ctftime.org](https://ctftime.org)
  - Annuaire de writeups
  - Solution aux challenges par les participants



# Objectifs du projet

- Faciliter la recherche lors de CTF
  - Recensement d'outils
  - Challenges similaires
  - Langages similaires



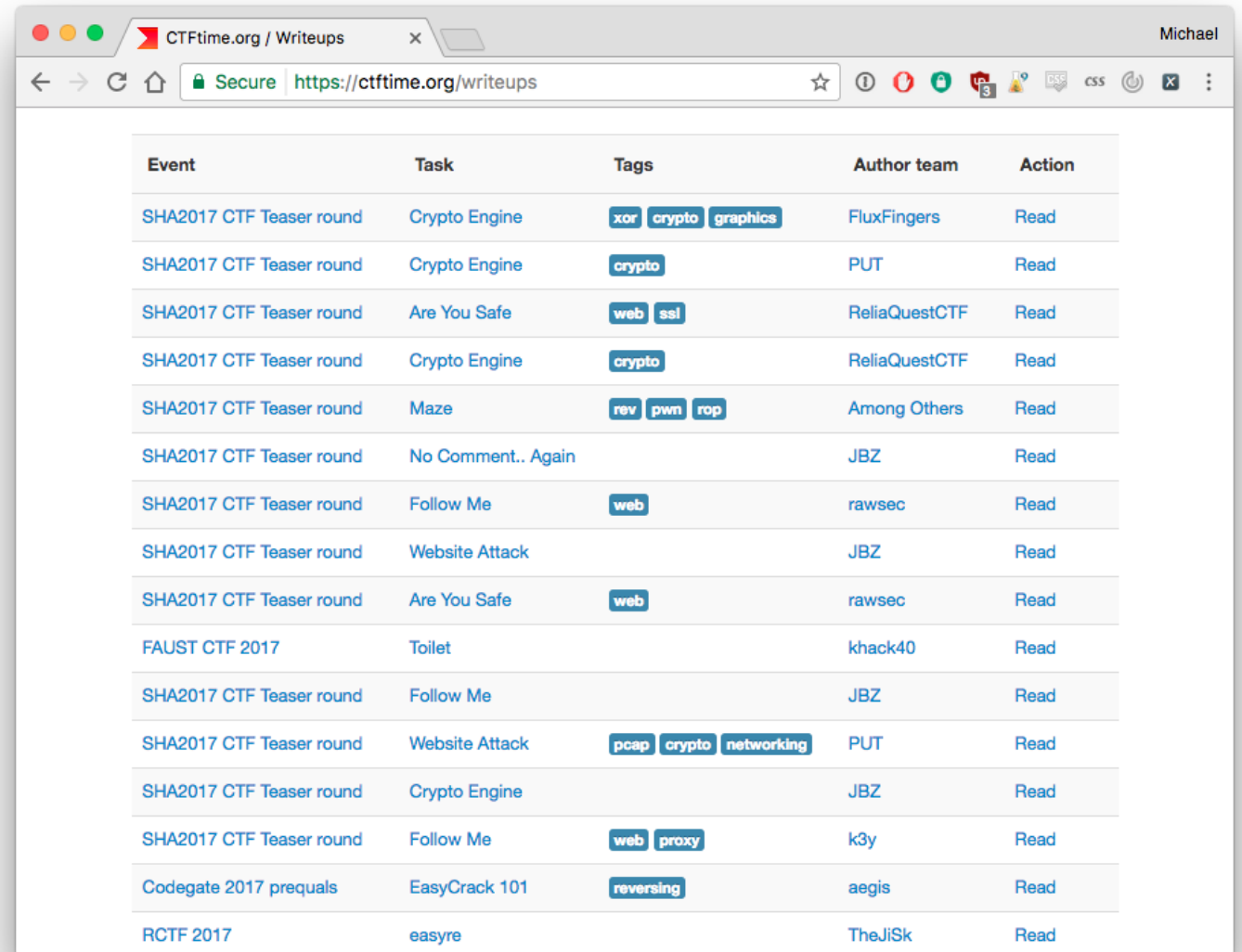
# Fonctionnalités

- Recherche par mots-clés
  - Langages
  - Writeups
  - Outils
- Recherche par catégorie
  - Web, Crypto, ...

# ctftime.org

## Catégories

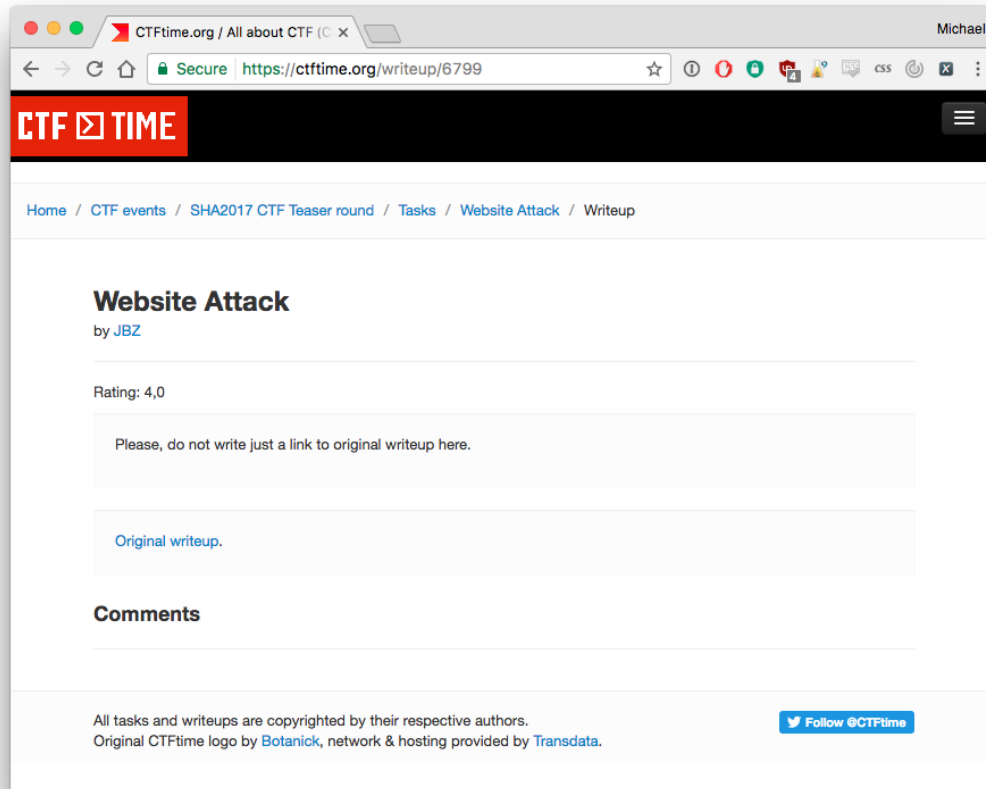
1. Web Hacking
2. Digital Puzzles
3. Cryptography
4. Steganography
5. Reverse Engineering
6. Binary Analysis
7. Mobile Security
8. Forensics
9. Cracking
10. Networking



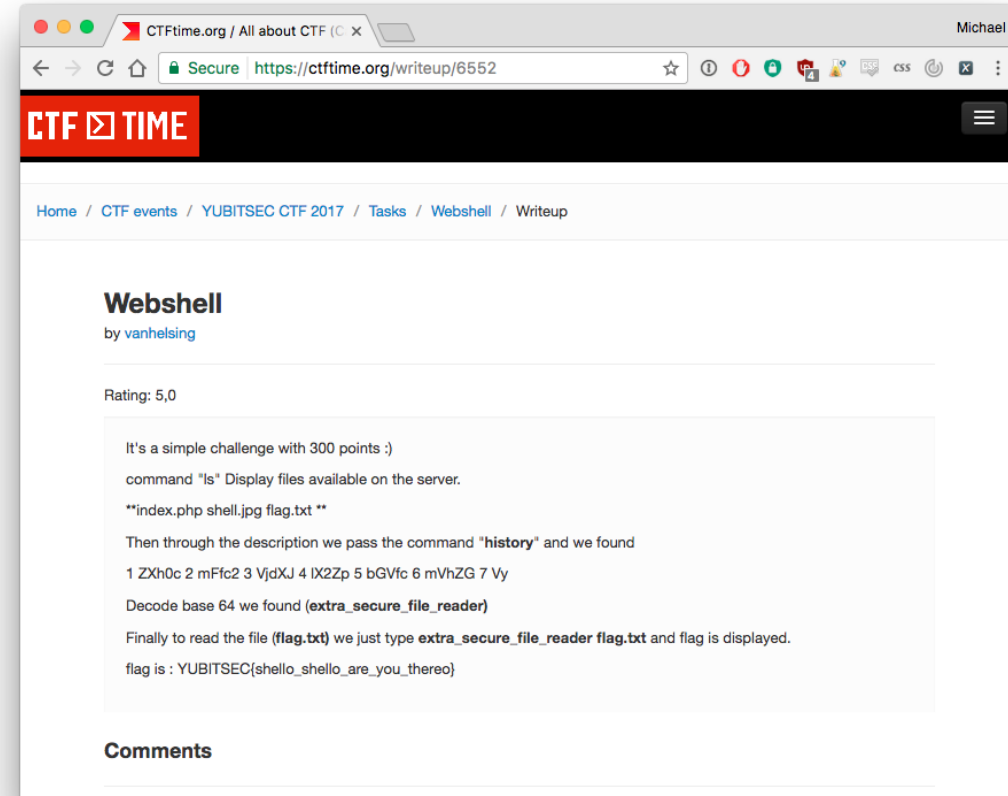
The screenshot shows a web browser window with the URL <https://ctftime.org/writeups>. The page displays a table of CTF events and tasks. The table has five columns: Event, Task, Tags, Author team, and Action. The data is as follows:

Event	Task	Tags	Author team	Action
SHA2017 CTF Teaser round	Crypto Engine	xor crypto graphics	FluxFingers	Read
SHA2017 CTF Teaser round	Crypto Engine	crypto	PUT	Read
SHA2017 CTF Teaser round	Are You Safe	web ssl	ReliaQuestCTF	Read
SHA2017 CTF Teaser round	Crypto Engine	crypto	ReliaQuestCTF	Read
SHA2017 CTF Teaser round	Maze	rev pwn rop	Among Others	Read
SHA2017 CTF Teaser round	No Comment.. Again		JBZ	Read
SHA2017 CTF Teaser round	Follow Me	web	rawsec	Read
SHA2017 CTF Teaser round	Website Attack		JBZ	Read
SHA2017 CTF Teaser round	Are You Safe	web	rawsec	Read
FAUST CTF 2017	Toilet		khack40	Read
SHA2017 CTF Teaser round	Follow Me		JBZ	Read
SHA2017 CTF Teaser round	Website Attack	pcap crypto networking	PUT	Read
SHA2017 CTF Teaser round	Crypto Engine		JBZ	Read
SHA2017 CTF Teaser round	Follow Me	web proxy	k3y	Read
Codegate 2017 prequals	EasyCrack 101	reversing	aegis	Read
RCTF 2017	easyre		TheJiSk	Read

# Contenu



Lien externe vers le  
writeup



Writeup interne à  
ctftime.org

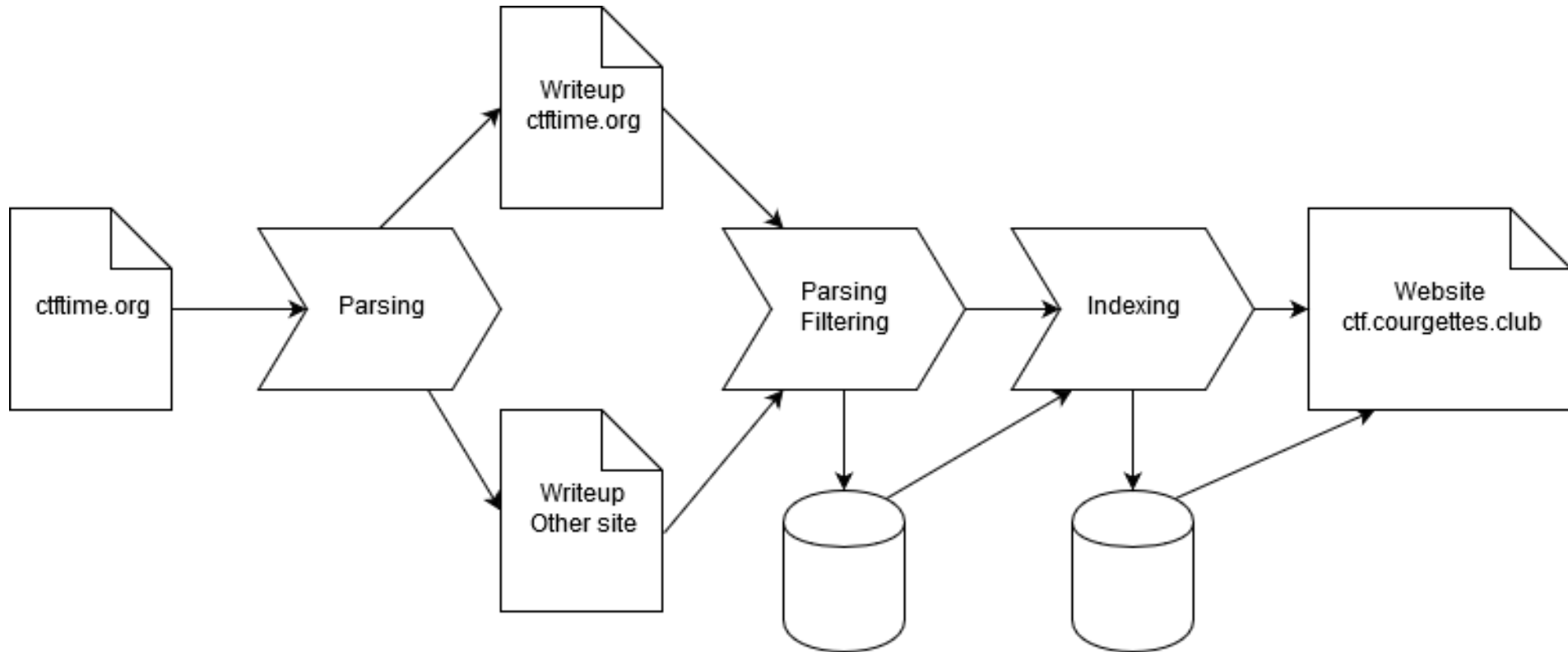
# Technologies utilisées

- Langage : Python
- Site web : Flask
- Indexing : Whoosh
- Front-end : Bootstrap





# Pipeline



# Parsing

- Ctftime.org

"title" : titre du projet

"author" : auteur

"tags" : tags

"event" : nom de la CTF

"url" : lien externe

- Meta :

"title", "og:title", "twitter:title" : titre du projet

"description", "og:description", "twitter:description" : courte description

"keywords" : mots-clef

- Contenu du writeup : contenu visible de la page html

# Indexing

// CTFTIME indexer

self._schema = Schema(id	=	ID	(unique=True),
text	=	TEXT	(analyzer=self._analyser, stored=True),
title	=	TEXT	()
author	=	TEXT	()
tags	=	KEYWORD	(lowercase=True),
event	=	TEXT	()
url	=	TEXT	()
language	=	TEXT	()
category	=	TEXT	()
tool	=	KEYWORD	(lowercase=True),
tag_title	=	TEXT	()
meta_title	=	TEXT	()
meta_description	=	TEXT	(analyzer=self._analyser),
meta_keywords	=	KEYWORD	(lowercase=True),
meta_og_title	=	TEXT	()
meta_og_description	=	TEXT	(analyzer=self._analyser),
meta_twitter_title	=	TEXT	()
meta_twitter_description	=	TEXT	(analyzer=self._analyser)
)			

// Tool indexer

self._schema = Schema(id	=	ID	(unique=True),
category	=	TEXT	(analyzer=self._analyserLower),
title	=	TEXT	(analyzer=self._analyserLower),
url	=	TEXT	()
description	=	TEXT	(analyzer=self._analyser))

# Indexing - Filters

- LowercaseFilter() : Lower case
- StopFilter() : Stop words
- CharsetFilter(accent\_map) : Replace special characters
- Requests + BeautifulSoup → Texte visible de la page html



# Front end et web service

- Website
  - `ctf.courgettes.club`
- Interaction avec un service web
  - 2 routes disponibles
    1. Liste des termes disponible dans l'indexe
    2. Recherche par
      - méthode de classement
      - mots clés
      - termes

# Queries / API

// API terms

<https://ctf.courgette.club/api/terms/<column>> (tags, tool, category, language)

```
[  
    "binary analysis",  
    "cracking",  
    "cryptography",  
    "exploits",  
    "forensics",  
    "misc",  
    "mobile security",  
    "networking",  
    "reverse engineering",  
    "steganography",  
    "web"  
]
```

// Python implementation

```
@app.route("/api/terms/<field>", methods=["GET"])  
def categories(field):  
    searcher = indexer.getIndex().searcher()  
    try:  
        res = [cat.decode("utf-8") for cat in list(searcher.lexicon(field))]  
        return jsonify(res)  
    except Exception:  
        return jsonify({'message': 'This field doesnt exists!'}), 426
```

# Queries / API

// API search

/api/search/**1**?scoring=**tfidf**&category=**cryptography**&language=**python**&tool=**requests**&query=insomni'hack

- Recherche de **insomni'hack**
- À la page **1**
- Avec la méthode de classement **TD IDF**
- Qui contient la catégorie **cryptography**
- Et le langage **python**
- Ainsi que l'outil **requests**

# Construction de la requête

// API search

```
/api/search/1?scoring=tfidf&category=cryptography&language=python&tool=requests&query=insomni'hack
```

// Python implementation

# Make the query with (i) just words, or (ii) both (iii) or just terms

if query\_text is not None and terms is not None:

    query = And([query\_text, terms])

elif query\_text is not None:

    query = query\_text

elif terms is not None:

    query = terms

else:

    raise Exception('No input provided!')

return self.\_searcher.search\_page(query, page,  
pagelen=20)

metadata = {

    'category': request.args.getlist('category'),

    'language': request.args.getlist('language'),

    'tool': request.args.getlist('tool'),

    'tags': request.args.getlist('tags'),

}

terms = None

terms\_list = []

for field, values in metadata.items():

    if len(values) > 0:

        terms\_list.append(And([Term(field, value) for value in values]))

if len(terms\_list) > 0:

    terms = And(terms\_list)



```
{"data": [
  {
    "category": [ "misc", "cryptography", ...],
    "language": [ "ruby", "python", "javascript " ],
    "meta_description": "", "meta_keywords": "",
    "meta_og_description":
    "Informations\nVersion\n\n\n\nBy\nVersion\nComment\n\n\n\n\nnoraj\n1.0\nCreation\n\n\n\n\nCTF\n\n\nName
: Insomni'hack teaser 2017\nWebsite : teaser.insomnihack.ch\nType : Online\nFormat : Jeopardy\nCTF Time :
link\n\nDescription",
    "meta_og_title": "Insomni'hack teaser - 50 - cryptoquizz - Misc",
    "meta_title": "", "meta_twitter_description": "", "meta_twitter_title": "",
    "score": 31.516846530352904,
    "tag_title": "", "tags": "",
    "title": "cryptoquizz",
    "tool": [ "requests" ],
    "url": "http://rawsec.ml/en/Insomnihack-2017-Teaser-cryptoquizz/"
  }
],
"metadata": {
  "found": 1,
  "hasMore": false,
  "on": 1
}}
```

Démonstration

# Améliorations

- Design et interface
  - Responsive
  - Sélection des termes/filtres
- API publique
  - Documentation
  - Publication
  - RESTful-isation

# Conclusion

- Projet fonctionnel et déployé
- Tests concluants
- Prise en main de Whoosh
- Mise en pratique de A à Z de l'indexing

Merci pour votre attention

Questions?