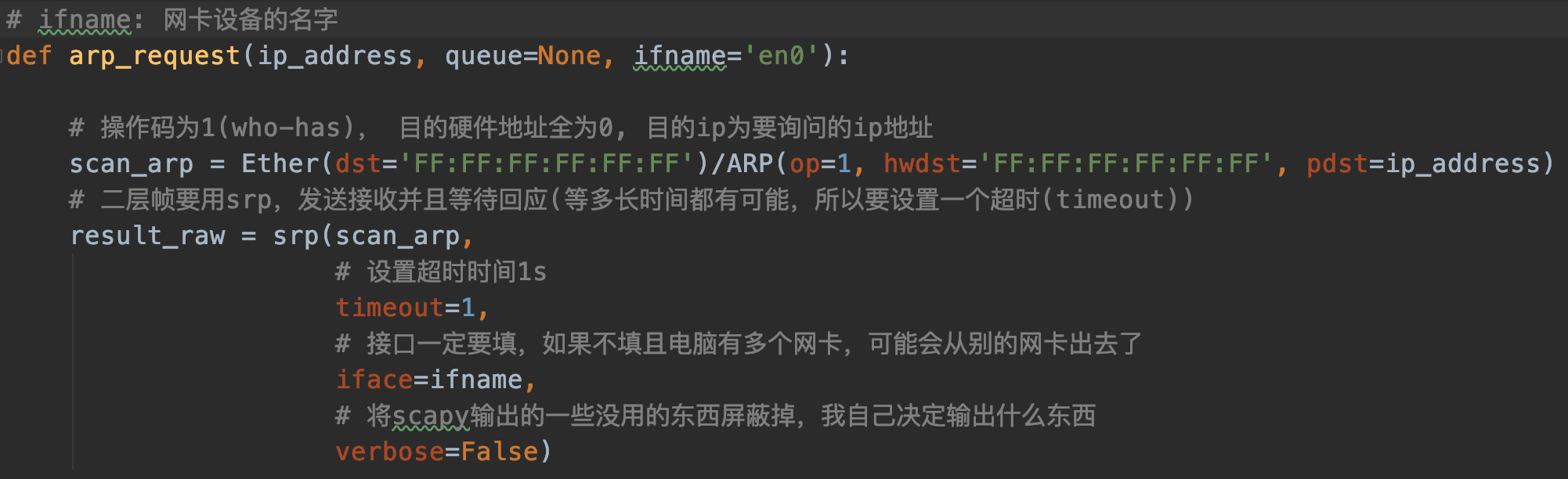
毕业设计思路

我的毕业设计是实现一个基于TCP/IP协议的具有扫描和检测功能的网络扫描器，主要分为三个模块，分别是主机探测模块，端口扫描模块和检测模块三个模块。其中的ip地址目前仅支持单个ip地址的输入或者网段的表示方式（由ipaddress模块实现）。

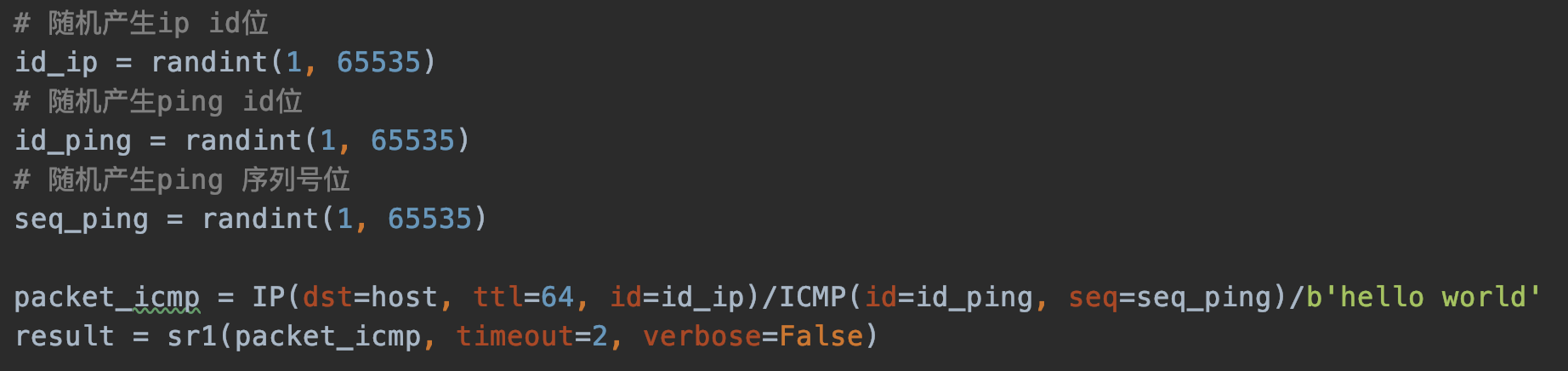
首先是主机探测模块，有arp探测，icmp探测和tcp探测。主要都是通过scapy构造包，通过解析回应报文来实现。其中arp和icmp是先实现一个，然后调用多进程来实现同时探测多个ip，tcp探测目前还没做。下图为我构造的arp探测的核心代码



使用多进程来实现探测多个



Icmp探测也类似，如果result不为空，说明主机存活。



Tcp中的SYN、FIN、NULL、XMAS探测我查资料发现这主要是用来探测端口开放情况的，如果用在判断目标主机是否存活的话，我打算随便设置一个端口, 然后发过去，如果有回应(不管是SYN+ACK还是ACK又或者RST、RST+ACK), 只要有回应, 就说明这个主机是存活的。

然后是端口扫描模块，SYN探测为发送一个带有SYN标识和端口号的TCP数据包给服务器，如果服务器这个端口是开放的，则会接受这个连接并返回一个带有SYN和ACK标识的数据包给客户端，随后客户端会返回带有RST标识的数据包而不是返回一个带有ACK和RST标识的数据包。如果目标端口处于关闭状态，则服务端会返回一个RST标识的数据包。代码如图。

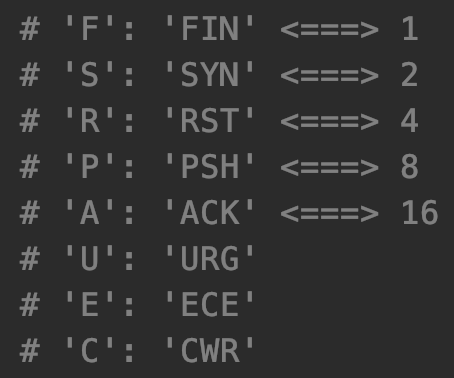


NULL扫描就是在发送的数据包中不设置任何标志位(tcp标志头是0)，如果目标端口是开放的则不会回复任何信息。如果目标端口关闭则会返回一个RST+ACK的数据包，代码如图。



FIN扫描与NULL扫描类似，只要把Null扫描发送的标志位改为F即可。

以下是标志位对应的简写



Xmas扫描是在发送的数据包中设置PSH,FIN,URG标志位，如果目标端口是开放的则不会回复任何信息。如果目标端口关闭则会返回一个RST+ACK的数据包



接下来的所有功能，我都还没实现。。。

先是udp扫描，扫描主机向目标主机的一个未打开的 UDP 端口发送一个数据包时，对方目标主机需要回复一个 ICMP“目的不可达”, 所以，如果对方不进行回复，就说明这个udp端口是开放的（前提是主机存活），但由于 UDP 和 ICMP 错误消息都不能保证到达，因此一次的扫描结果并不一 定能保证得出实际正确的结果，需要对目标主机进行多次的扫描，从而得到更加 准确的扫描结果。

随后是检测模块，因为端口服务和操作系统的识别实在是太麻烦了，所以暂时打算直接调用nmap。

Sql注入漏洞检测先是在要检测的url后边加

1. 一个单引号

2. And 1=1

3. And 1=2

4. and ‘1’ = ‘1

5. ' and '1'='2

等使页面出错，然后根据页面返回的关键字来判断所用的数据库

ftp弱口令检测主要就是暴力破解，首先用Python的ftplib库中的Ftp类做一个匿名登录，如果产生异常则弱口令扫描，首先读取字典文件（两个文件，一个用户名字典，一个密码字典，格式统一为一行只有一个用户名或者密码），然后循环读取两个文件中每一行数据作为用户名密码来尝试登陆，如果成功返回相应用户名密码，如果失败则不用管它，继续尝试其他用户名密码，直到所有全部失败，返回空，输出“用户名，密码未找到”

最后实现软件封装，做图形化界面