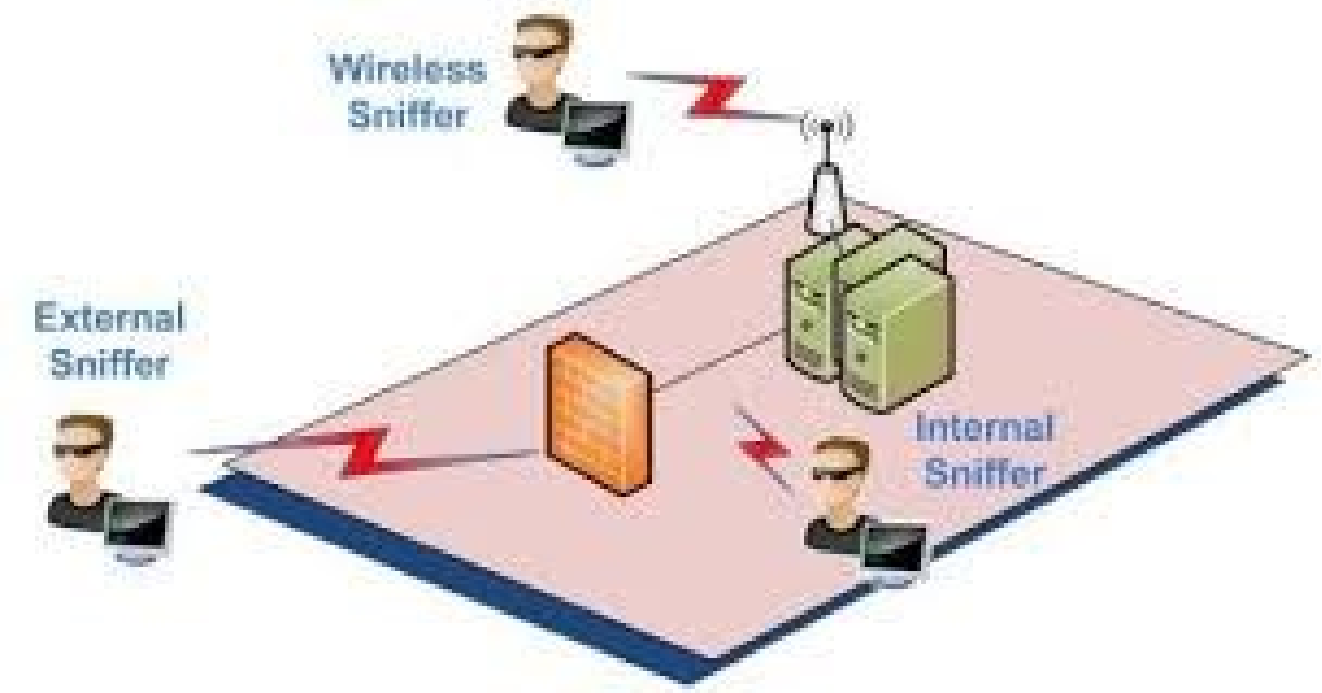
IASP505

Networks
Sniffing
Coding in Python

c3

Dr. John Yoon

Network Sniffing



- Goal
  - Monitoring
  - Inspection

Coding Steps

https://docs.python.org/3/library/socket.html

- Construct a Socket object
  - Socket family: AF_INET, AF_UNIX, etc
  - Socket type: SOCK_RAW, SOCK_STREAM, SOCK_DGRAM, etc
  - Protocol: IPPROTO_*, IPPORT_*, IP_*, TCP_*
- Bind
  - .bind()
    - (HOST,0)

Python Coding

```
import socket

# the public network interface
HOST = socket.gethostbyname(socket.gethostname())

# create a raw socket and bind it to the public interface
s = socket.socket(socket.AF_INET, socket.SOCK_RAW,
socket.IPPROTO_IP)
s.bind((HOST, 0))

# Include IP headers
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

# receive all packages
s.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)

# receive a package
print(s.recvfrom(65565))

# disabled promiscuous mode
s.ioctl(socket.SIO_RCVALL, socket.RCVALL_OFF)
```

Python Struct

https://docs.python.org/3/library/struct.html

- Unpack from the buffer
  - unpack(format, buffer)

```
struct.unpack('!BBHHHBBH4s4s4s' , data[:24])
```

Reference

- Visit
  - Network Sorcery: http://www.networksorcery.com/
- As you can see from WireSHARK
  - From: https://www.wireshark.org/

```
import socket, struct

def socketObject():
    HOST = socket.gethostbyname(socket.gethostname())
    print("host: ", HOST)

    s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
    s.bind((HOST,0))

    #include IP headers
    s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

    # to receive all packets
    s.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)

    dataa = s.recvfrom(65565)

    print("whole data: ", dataa)
```

```
import socket, struct

def socketObject():
    HOST = socket.gethostbyname(socket.gethostname())
    print("host: ", HOST)

    s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
    s.bind((HOST,0))

    #include IP headers
    s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

    # to receive all packets
    s.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)

    dataa = s.recvfrom(65565)

    print("whole data: ", dataa)

    data = dataa[0]
    ipData = struct.unpack('!BBHHHBBH4s4s4s' , data[:24])
    theRest = data[24:]
    protocol = ipData[6]
    print("-- The protocol: ", protocol)
```

```
data = dataa[0]
    ipData = struct.unpack('!BBHHHBBH4s4s4s' , data[:24])
    theRest = data[24:]
    protocol = ipData[6]
    print("-- The protocol: ", protocol)

    if protocol == 6:     # tcp
      tcpData = struct.unpack('>HH4s4sHHHHHH', theRest[:24])
      print("TCP destination port: ", tcpData[1])
      print("The payload of a TCP packet: ", theRest[24:])
    elif protocol == 17:  # udp
      udpData = struct.unpack('>HHHH', theRest[:8])
      print("UDP destination port: ", udpData[1])
      print("The payload of a UDP packet: ", theRest[8:])

socketObject()
```

MERCYBER SEC