WELCOME TO CLASS 7!

BLACK HAT PYTHON3

RALEIGH ISSA

GITHUB REPO

https://github.com/tiarno/bhp3_class

SUMMARY FROM LAST CLASS

- scapy graphics + world map
- arp poisoning + dns spoofing
- building images from packet capture streams
- identify faces in those captured images

EXTRACT CONTENT FROM PCAP FILE

`recapper.py`

PYTHON CODING

- named tuple
- dictionary creation
- regular expression/raw strings
- string slicing/indexing
- interating dict keys
- byte/string conversion

REGEXP

- https://pythex.org

HTTP HEADERS

- https://developer.mozilla.org/en-US/docs/Glossary/MIME_type

CLIENTS AND SERVERS

- TCP Client: foundation for sending and receiving
- TCP Server: foundation for creating, listening, sending, receiving
- UDP Client / UDP Server
- TCP Proxy: machine-in-the-middle, leveraging the TCP client/server code

TCP CLIENT

- first, a quick http server: python -m http.server --bind=9999

TCP CLIENT

```
import socket

def connect(host, port):
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((host, port))
    client.send('GET /')
    return client.recv(4096)

if __name__ == '__main__':
    host = '192.168.1.69'
    port = 9999
    print(connect(host, port))
```

- server `tcp_server.py`
- client `tcp_client.py`

NETCAT

- `netcat.py`

HEXDUMP

```
          1         2
012345678901234567890123456789
 Black Hat Python, Raleigh ISSA
:Black Hat :
s = 'Black Hat '
hexa = ' '.join(['%04X' % ord(x) for x in s])
0042 006C 0061 0063 006B 0020 0048 0061 0074 0020
text = ''.join([x if 32 <= ord(x) < 127 else '.' for x in s])
'Black Hat '
```

YOUR JOB

- Finish out your `recapper.py`
- Test it with the newly added `pcap.pcap` file
- Start building out your version of `netcat.py`

READING

- NetCat `ncat`, `nc`: the real thing and how it works
  https://www.binarytides.com/netcat-tutorial-for-begi
  http://forensicswiki.org/wiki/Netcat
  http://forensicswiki.org/wiki/Tcpdump#Using_Tcpdu

FEEDBACK PLEASE!

- tim@reachtim.com
- discord: https://discord.gg/WR23qUj