



January 25, 2024:

Question: Communication of Simulated Banking Transactions

You are tasked with replicating a communication sequence between an ATM and a bank server using Python and Scapy. The communication begins with authentication using a basic username and password, followed by a series of banking transactions. Plaintext passwords can be used for simplicity purposes.

1. Authentication:

- The bank server receives an authentication request packet from the ATM that includes a username and password.
- The bank server returns an authentication success message.

2. Banking Transactions:

- After successful authentication, the ATM performs a series of banking transactions.
- Complete the following scenarios' packet implementations:
 - Request an ATM balance inquiry.
 - The bank's server responds to balance inquiries.
 - Withdrawal Request from an ATM.
 - The bank server responds to withdrawals.

3. Pcap Export:

Here is some code to get you started

```
'''  
  
https://scapy.readthedocs.io/en/latest/build\_dissect.html#simple-example  
1.) filename=customProtocolCreation.py  
2.) To execute this file, enter command in bash terminal  
    python customProtocolCreation.py  
'''# Example code to create custom protocol StudentProtocol and bind it on
```

```
from scapy.all import *
class StudentProtocol(Packet):
    name = "StudentProtocol"
    fields_desc = [
        StrField("name", ""),

    ]

bind_layers(TCP, StudentProtocol)
# Create two packets
packet1 = IP(dst="127.0.0.1") / TCP(dport=1234) / StudentProtocol(name="Ali")
packet2 = IP(dst="127.0.0.1") / TCP(dport=5678) / StudentProtocol(name="Bob")

# Combine the packets into a list
packets = [packet1, packet2]

# Save the packets to a pcap file
wrpcap("custom_packets.pcap", packets)

# To open "custom_packets.pcap" in wireshark
# On bash terminal run command
wireshark custom_packets.pcap
```