

# CS484 - Computer Networks

## Project #3 - Covert ICMP Channel

A CIA agent has successfully infiltrated FAPSI (the Russian equivalent of the NSA) and needs a covert means to pass information from their internal network to the case officer at a remote location. The network is highly secure, employing layered IPS, IDS, AI-driven firewalls, and real-time traffic monitoring. As a highly motivated Cyber Officer, you have been tasked with solving this problem.

In this programming assignment, you will design a network client that hides secret messages in the data section of an ICMP Ping Request, as well as a network server that captures the Ping requests and prints the secret messages. The packets **MUST** appear to be normal ICMP protocol traffic, otherwise, the CIA agent will be discovered. You will write your code in Python 3. You will work with a partner to complete this assignment. Only one deliverable is necessary per team.

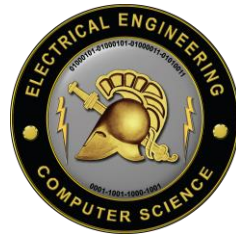
**100 points; Due Lesson 30 (11DEC2018) at the beginning of class.**

### Requirements

1. The client must create a properly-formed ICMP Ping Request packet with a text message in the data section and send it over the network to a remote server.
2. The server will receive the ICMP Ping Request packet and print the message in human-readable text.
3. The packets you send must appear in Wireshark to be valid, properly formatted, ICMP Ping Request packets.
4. All ICMP packets must have a proper, correct checksum included in the packet header.
5. You cannot "hardcode" the secret message. It must come from either standard-in or an external file.
6. The secret message in the data section of the packet must be obfuscated in some fashion (i.e. I shouldn't be able to see the contents of the message when examining the packets in Wireshark). This will make IDS detection using deep-packet inspection more difficult.
7. The only networking module that may be used is Python socket. Any non-network Python module may be used to complete this assignment.
8. On lesson 40, you will demonstrate your program by sending me a secret message from your client.

### Bonus Features (5 points per feature)

1. Two-Way Transmission: Implement the client and server such that both can send and receive covert data (will require multithreading).
2. Data Encryption: Use AES Encryption with a pre-shared key to properly encrypt the secret messages. There are many tutorials online that explain how to do this.
3. Image Transfer: Send a picture from the client to the server.
4. TCP Connection: Establish a TCP connection over the ICMP packets (yes, it can be done!) and send a large file from client to server.



## Submissions

- Printed copy of client and server code, with signed eAcknowledgment coversheet. One printed submission per team is sufficient, but both members must sign the eAck.
- Email a .zip file with your client and server code to your instructor, named as follows:
  - Project3\_cadetname1\_caetname2.zip
    - client.py
    - server.py

## Resources

Python sockets module. Class materials. Your brain. Grit.

## Pro Tips

- Start with the working TCP Client/Server Code from Lesson 5.
- Ensure that you are using Python 3.
- You will have to run your program as admin in order to create raw sockets.
- Python socket documentation: <https://docs.python.org/3/library/socket.html>
- Start with a simple client and server program that sends data from client to server, then build up from there.
- Get your programs running with localhost before you try to send traffic across the network.
- Test your additions as you go.
- If you need help, ask.