Name – Akash Nair Subject – Cyber and Information security

Practical -1

Aim: The aim of this practical exercise is to implement the principles of the CIA triad (Confidentiality, Integrity, and Availability) in a networked environment. By designing and configuring security measures, cryptographic techniques, and access controls, this practical aims to demonstrate the effective safeguarding of sensitive data, ensuring its confidentiality, integrity, and availability throughout the communication process.

Code:

server.py:

import socket

import pyaes

import hashlib

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

server\_address = ('localhost', 8080)

server\_socket.bind(server\_address)

print("\n\tServer is listening for incoming messages...")

plain, client\_address = server\_socket.recvfrom(1024)

plain = plain.decode()

data, client\_address = server\_socket.recvfrom(1024)

# print(f"Received message 1 from {client\_address}: {data}\n")

aes = pyaes.AESModeOfOperationCTR(b'DESCRYPTDESCRYPT')

decrypted\_data = aes.decrypt(data).decode()

print(f"\t{server\_address}\n")

if decrypted\_data == plain:

print("\tConfidentiality verified: Decrypted message matches original plaintext.\n")

else:

print("\tConfidentiality compromised: Decrypted message does not match original plaintext.\n")

# data, client\_address = server\_socket.recvfrom(1024)

# print(data)

received\_md5\_hash, client\_address = server\_socket.recvfrom(1024)

received\_md5\_hash = received\_md5\_hash.decode()

calculated\_md5\_hash = hashlib.md5(decrypted\_data.encode()).hexdigest()

# print(f"Received message 2 from {client\_address}: {data.decode()}")

if received\_md5\_hash == calculated\_md5\_hash:

# print(f"Hash is: {calculated\_md5\_hash}")

print("\tIntegrity verified: Hash value matches.\n")

else:

print("\tIntegrity compromised: Hash values do not match.\n")

if decrypted\_data:

print("\tAvailability verified: Message received properly.\n")

print(f"\tMessage from client: {decrypted\_data}\n")

else:

print("\tAvailability compromised: Message not received properly.\n")

server\_socket.close()

#udp = sock\_dgram

client.py:

import socket

import pyaes

import hashlib

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

server\_address = ('localhost', 8080)

aes = pyaes.AESModeOfOperationCTR(b'DESCRYPTDESCRYPT')

plaintext = input("Enter message to send: ")

ciphertext = aes.encrypt(plaintext.encode())

md5\_hash = hashlib.md5(plaintext.encode()).hexdigest()

# print(md5\_hash)

client\_socket.sendto(plaintext.encode(), server\_address)

client\_socket.sendto(ciphertext, server\_address)

client\_socket.sendto(md5\_hash.encode(), server\_address)

client\_socket.close()

# UDP = sock\_dgram

A screenshot of a computer

Description automatically generated