CRYPTOGRAPHY AND NETWORK SECURITY

LAB 8 : MAN IN THE MIDDLE ATTACK

NAME : OM SUBRATO DEY

REGISTER NO. : 21BAI1876

THEORY:

To explain an idea of the Man-in-the-Middle (MITM) attack in python we can write a basic example of how an attacker will interrupt communication between two clients. But it is important to note that practical MITM attacks involve a lot of network tweaking and normally involve scapy or mitmproxy.

For this demonstration, we'll simulate the following:For this demonstration, we'll simulate the following:

Alice: The agent which this means that Bob wants to receive a message from the sender.

Bob: The receiver who anticipate a message be sent by Alice.

Mallory: The receiver who intent to alter the Information content of the message that has been sent by the sender.

We shall develop another Python script to mimic the message exchange and present how Mallory can modify the actual message.

MITM Simulation code in python

Here is a Python script that demonstrates a simple MITM attack:Here is a Python script that demonstrates a simple MITM attack:

CODE:

```
class Party:
    def __init__(self, name):
        self.name = name

    def send_message(self, message, intermediary):
        print(f"{self.name} sends: {message}")
        return intermediary.intercept(message, self.name)

    def receive_message(self, message):
        print(f"{self.name} receives: {message}\n")

class Interceptor:
    def __init__(self):
        self.alter_message = True  # Set this to False to
disable message tampering

    def intercept(self, message, sender_name):
        print(f"*** Intercepting message from {sender_name}
***")
        if self.alter_message:
            # Attacker modifies the message
            modified_message = f"{message} [Tampered by
Attacker]"
            print(f"*** Message modified to: {modified_message}
***")
```

```
            return modified_message
        else:
            # Attacker forwards the message without tampering
            print("*** Message forwarded without tampering ***")
            return message

# Create the parties and interceptor
alice = Party("Alice")
bob = Party("Bob")
mallory = Interceptor()

# Simulate the communication
original_message = "Hello, Bob! This is Alice."
tampered_message = alice.send_message(original_message, mallory)
bob.receive_message(tampered_message)
```

OUTPUT: