VULNERABILITY TESTING IN CELLULAR NETWORKS

Omer Rastgar

HABIB UNIVERSITY

KARACHI, PAKISTAN

2019

VULNERABILITY TESTING IN CELLULAR NETWORKS

The Capstone Design Project
presented to the academic faculty

by

Omer Rastgar

in partial fulfillment of the requirements for
BS Electrical Engineering
in the
School of Science and Engineering

Habib University

May 2019

VULNERABILITY TESTING IN CELLULAR NETWORKS

This capstone design project was advised by:

Dr. Tariq Mumtaz
Faculty of Electrical Engineering
Habib University

Approved by the Faculty of Electrical Engineering on April 19, 2023.

“If you want to find the secrets of the universe, think in terms of energy, frequency and vibration.”

Nikola Tesla

TABLE OF CONTENTS

List of Figures vii

Chapter 1: Introduction and Background 1

1.1 Introduction 1

1.2 Literature Review 1

1.3 Problem Statement/Objective 5

Chapter 2: Design Process 6

2.1 Clients/Stakeholders and their requirements 6

2.2 Society, Economic, and Ethical considerations 6

2.3 Technical Requirements 6

Chapter 3: Design Details 8

3.1 Solution Statement 8

3.2 Solution Overview 8

3.3 System Details 9

3.4 modules 14

Chapter 4: Prototyping and Testing 18

4.1 Test cases 18

4.2 Results 18

Chapter 5: Conclusion and Future Work 29

Appendix A: Manual of Prototype 34

References 35

Vita 36

LIST OF FIGURES

3.1 LTE Architecture in NS-3. 13

3.2 Communication links in an LTE Architecture 13

3.3 Block diagram of entire system 14

4.1 GUI for NS-3 19

4.2 GUI for NS-3 20

4.3 GUI for NS-3 21

4.4 GUI for NS-3 22

4.5 GUI for NS-3 23

4.6 GUI for NS-3 24

4.7 GUI for NS-3 24

4.8 data is encrypted in a file and then sent to ns3 to be transported through the network to reach the remote host. Here, if the proper key exchange doesn't take place, the key can be observed by an attacker that is present in the communication and then can use for further decryption of data once the connection has been established. 25

4.9 This is a proof of concept demonstration of Reverse Proxy which can be used for SSL Hijacking in a MITM attack. 27

4.10 Gant Chart 28

Cellular Vulnerability Testing Environment

Our aim is to design a simulation environment for testing and learning different Vulnerabilities in LTE architecture. We have used a Network simulator called NS-3. Network Simulator 3 (ns-3) is an open-source network simulation platform. It is a tool used by academics and developers to investigate network behavior and build and test new networking protocols. Ns-3 provides a versatile and configurable simulation environment that allows users to model many sorts of networks and network behaviors. It also includes a set of tools for evaluating and displaying simulation results. Ns-3 is written in C++ and Python and distributed under the terms of the GNU General Public License.

For capstone one, we had two main objectives.

Designing an LTE architecture and simulation for Man In the middle attack.

A man-in-the-middle (MITM) attack occurs when a perpetrator inserts himself into a dialogue between a user and an application, either to eavesdrop or to mimic one of the parties, giving the impression that a regular flow of information is taking place. An attack's purpose is to steal personal information such as login passwords, account information, and credit card numbers. Users of banking apps, e-commerce sites, and other websites that require signing in are typical targets. Information collected during an attack might be utilized for a variety of objectives, such as identity theft, unauthorized financial transfers, or unauthorized password changes.

The base station (also known as an eNodeB) in a cellular network is in charge of broadcasting and receiving signals to and from mobile devices within its service area. The base station runs on a certain frequency or collection of frequencies that the network operator has designated. A rogue eNodeB, on the other hand, is an unlawful base station that is set up without the network operator's authorization. Hence it is one form of MITM attack.

To simulate this we designed the network in NS3 and observed the data passing through the different nodes. Our aim was to see if our rogue eNodeB could alter the data passing through it.

Designing GUI to interact with the network to perform and discover attacks.

In order to view all the data we have an intractable GUI, where the user can drag-drop objects to the scene and see their interaction with a hand full of tools that make it easier for the user to understand the entire scene. These Tools are applications that are linked to this GUI and the data provided by the ns3 network. Once we have run the simulation the data collected at the GUI front can be viewed in tools such as Wire shark to inspect the different scenarios that are present in the scene.

KEYWORDS: Security, Penetration testing, Cellular.

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Introduction

My aim for providing this information is that every day there is an increase in threat for all networks and privacy has been a huge issue that reduces the severity of a known breach such as this. we need to realize the importance of networks that are increasingly a part of our lives and control many of the hidden processes around us. Similarly, mobile networks have a huge role in maintaining these processes which include Information technology and 2 Operational technology. Here OT may have much higher impacts as SCADA systems may be running on wireless network devices that are connected to 3G or 4G. Also in the near future IoT and Driverless cars will be using 5G technology to communicate with one another and security would be paramount. Lastly, it is important to realize that any existing solutions are temporary as every day there are new ways of exploiting devices take place. hence, our aim is not to design something impossible to hack. rather a device that increases the exploitation time to a considerable amount hence making the attack nonfeasible for most threat levels.

1.2 Literature Review

Using a Software Defined radio

In one research, they used a Software Defined Radio board, to show that it enables an attacker to launch an active interception assault and perhaps inject traffic into the network. Passive eavesdropping necessitates additional software development (on the basis of open-source projects), but the hardware required is minimal. The radio gear necessary for listening might be purchased for roughly ten euros. Although many services use upper-layer encryption (e.g., TLS), this is not universally used, and sensitive information (websites visited, source and destination IP addresses) may still be available in

plaintext on the radio interface voicemail hacking using Caller ID spoofing techniques even when upper layer encryption is present.[1] [2]

Authenticate device using preshared key in LTE

In LTE networks, preshared keys (PSKs) are a type of authentication key that is shared by the device and the network operator in advance. PSKs are typically used in networks where the authentication infrastructure is not yet in place or where the authentication process needs to be simplified. In these cases, the device and the network operator agree on a PSK in advance, and the device uses this PSK to authenticate itself to the network. This can be done manually or through a secure out-of-band channel, such as NFC or QR codes. Once the device has authenticated itself using the PSK, it can access the network and establish a secure connection. However, the use of PSKs can introduce security vulnerabilities, as the PSK must be shared and stored in advance. This can make it easier for attackers to intercept and compromise the key, potentially gaining access to the network. For this reason, network operators often prefer to use other, more secure authentication methods in LTE networks.[3]

CVE-2019-3568 for whatsapp

If a buffer overflow vulnerability is present in the PJSIP software library, an attacker could potentially use this vulnerability to gain unauthorized access to the control plane of a SIP (Session Initiation Protocol) system or application. The control plane is the part of a communication system that is responsible for managing and coordinating the various components of the system, such as routers, switches, and servers. By exploiting a buffer overflow vulnerability in PJSIP, an attacker could potentially gain access to the control plane of a SIP system, allowing them to modify or manipulate the system's configuration or behavior. This could potentially allow the attacker to intercept or modify SIP messages, redirect or block calls, or perform other malicious actions. To protect against this type of attack, it is important for SIP systems and applications to use secure and up-to-date versions of the PJSIP library, and to implement other security

measures, such as authentication and encryption, to prevent unauthorized access to the control plane. A buffer overflow vulnerability in the PJSIP software library could potentially allow an attacker to execute arbitrary code on a phone or other device that uses PJSIP. This is possible because a buffer overflow vulnerability can allow an attacker to write and execute their own code on the system, by exploiting the way that the program stores and processes data in its temporary storage areas (buffers). In the case of PJSIP, if the library is vulnerable to buffer overflow attacks, an attacker could potentially send specially crafted SIP messages or other data to the device, which could trigger a buffer overflow and allow the attacker to execute their own code on the device. This code could be used to perform various malicious actions, such as accessing sensitive information, modifying system settings, or controlling the device remotely. To protect against this type of attack, it is important for SIP systems and applications to use secure and up-to-date versions of the PJSIP library, and to implement other security measures, such as authentication and encryption, to prevent unauthorized access and code execution. [4]

Several different authentication methods [3]

1. SIM-based authentication: In this technique, the authentication keys are saved on a secure chip installed in the device known as a Subscriber Identity Module (SIM). The network provider issues the SIM card, which is linked to the specific device. When the device attempts to connect to the network, it utilizes the keys contained on the SIM to authenticate itself. Because the keys are saved on a secure chip and are linked to a specific device, this gives a high level of security.

2. USIM-based authentication: In this technique, the authentication keys are saved on a secure chip implanted in the device known as a Universal Subscriber Identity Module (USIM). The USIM is similar to a SIM in appearance, but it is intended for usage in networks that support different access technologies, such as LTE and 5G. The network operator issues the USIM, which is linked to the specific device. When the device attempts to connect to the network, it utilises the keys contained

on the USIM to authenticate itself. Because the keys are saved on a secure chip and are linked to a specific device, this gives a high level of security.

3. EAP-AKA authentication: Authentication keys are created and managed by an authentication server distinct from the network operator in this manner. To securely authenticate itself to the authentication server, the device use the Extensible Authentication Protocol-Authentication and Key Agreement (EAP-AKA) protocol. The server then produces and distributes the appropriate network keys to the device. This ensures a high level of security since the keys are produced and managed by a separate server and are not shared with the device in advance.

Methods to protect from DDOs

1. Firewalls: Firewalls are a type of security technology that can be used to filter and block incoming traffic to a network or server. By configuring the firewall to only allow legitimate traffic and to block or rate-limit suspicious traffic, it is possible to protect against DDoS attacks by limiting the amount of traffic that can reach the target network or server.

2. Traffic-shaping tools: Traffic-shaping tools are used to monitor and manage the flow of traffic on a network or server. These tools can be used to identify and block suspicious traffic, or to prioritize and allocate bandwidth to important traffic, such as real-time audio or video streams. This can help to mitigate the effects of a DDoS attack by ensuring that legitimate traffic is able to pass through and that the network or server remains available to users.

3. Network design: The design of a network can also play a role in protecting against DDoS attacks. By implementing redundant and distributed networks, it is possible to distribute traffic across multiple network paths and to prevent a single point of failure. This can help to prevent an attacker from overwhelming a single network or server, and can make it more difficult for the attacker to disrupt the network's services.

4. Security services: There are also many security services that are specifically designed to protect against DDoS attacks. These services can provide additional layers of protection, such as scrubbing centers that filter and clean incoming traffic, or intelligent routing that automatically redirects traffic away from congested or vulnerable areas of the network. These services can provide additional protection against DDoS attacks, and can help to ensure the availability and reliability of a network or server.

Designing De-authentication packets for DOS

It is possible to generate raw socket packets on an Android device. Raw sockets are not directly supported by the Android operating system, but they can be accessed using the Android NDK (Native Development Kit), which allows developers to write native code that can be compiled and run on Android devices.

To generate raw socket packets on an Android device, a developer would need to write a native code program using the Android NDK that creates a raw socket and then constructs and sends the desired packets using the raw socket. This program could then be compiled and run on an Android device, allowing the device to send raw socket packets.[5] [6]

1.3 Problem Statement/Objective

Improving security for cellular networks by simulating Vulnerabilities and analyzing then using Global System for Mobile communications Association guidelines and CIA triad to evaluate a software solution to mitigate the problem.[7]

CHAPTER 2

DESIGN PROCESS

This chapter should include the following:

2.1 Clients/Stakeholders and their requirements

1. Cellular providers: as they can use a simulation environment to train their employees

2. Phone users: As they can know more about Cellular security through an easy-to-use application.

2.2 Society, Economic, and Ethical considerations

As discussed above the impact of a Vulnerability is huge. Hence, fast and proper action needs to be taken place to mitigate the effects. This platform will enable people to understand Cellular security so that they can better protect them selves.

2.3 Technical Requirements

We could use Omnet++, NS3 or GNS3:

OMNeT++ is an open-source, component-based simulation tool for modeling and analyzing communication network performance. It is created in C++ and is intended to be modular, expandable, and scalable, allowing it to be used in a variety of simulation settings. OMNeT++ allows users to create sophisticated models of communication networks and run simulations to assess their performance under various scenarios. It also offers a set of tools for evaluating and visualizing simulation results, giving users insight into the behavior and performance of their network models. OMNeT++ is widely

used in academia and business for communication network research and development.

GNS3 is a graphical network simulator that enables users to create and configure virtual networks using a variety of devices and protocols. It is open-source software that operates on a variety of platforms like as Windows, Linux, and macOS. Network engineers and IT professionals frequently use GNS3 to build, test, and debug network settings, as well as to simulate complicated network scenarios. It enables users to build and manage virtual networks on their personal computers without the need for actual hardware or specialist equipment. GNS3 interfaces with various network simulation and emulation tools and emulators, including Cisco IOS, Juniper Junos, and Wireshark, to provide a complete and adaptable platform for network experimentation and analysis.

comparision with NS3:

OMNET++ has limited features and libraries

GNS3 contains virtual machines that provide a higher abstraction than what is required.

CHAPTER 3

DESIGN DETAILS

You are required to include the following details in this category.

3.1 Solution Statement

We have designed the LTE architecture and now in capstone two we will add solutions to the attacks that we have shown.

3.2 Solution Overview

Tools

1. Network analysis tools: These tools are used to analyze the data that is transmitted over a network, such as phone calls, text messages, and other data. They can be used to detect inconsistencies or anomalies in the data that could indicate that a device is being spoofed.

2. Phone number analysis tools: These tools are used to analyze phone numbers and other identifying information associated with a device. They can be used to determine if a phone number is valid, or if it has been altered or falsified.

3. Forensic analysis tools: These tools are used to perform detailed, technical analysis of a device, such as a phone, to determine if it has been tampered with or modified in any way. This can include analyzing the device's hardware, software, and other data to identify any changes or modifications that may indicate spoofing.

3.3 System Details

Data Pathway [8]

In an LTE network, data is typically communicated to a server using the following pathway as shown in the diagram 3.1://

1. The user's device, such as a smartphone, sends a request for data to the eNodeB (evolved NodeB), which is the wireless base station that manages and controls wireless communications in the network.

2. The eNodeB receives the request and forwards it to the Mobility Management Entity (MME), which is a network element that manages the movement of user devices within the network.

3. The MME receives the request and identifies the appropriate serving gateway (S-GW) to handle the request. The S-GW is a network element that acts as a gateway between the LTE network and other networks, such as the internet.

4. The S-GW receives the request and forwards it to the appropriate server, which is connected to the internet.

5. The server receives the request and processes it, then sends a response back to the S-GW.

6. The S-GW receives the response and forwards it to the MME.

7. The MME receives the response and forwards it to the eNodeB.

8. The eNodeB receives the response and sends it to the user's device

Sending UDP packets

UDP, or User Datagram Protocol, is a type of network protocol that is used for transmitting data over a network. In an LTE network, UDP communication works in essentially the same way as it does in other types of networks. The main difference between UDP communication in an LTE network and other types of networks is that the eNodeB,

which is the wireless base station that manages and controls wireless communications in an LTE network, may perform additional functions to ensure the efficient transmission of data. For example, the eNodeB may use a technique called packet scheduling to optimize the allocation of radio resources, such as frequency bands and time slots, to ensure that data is transmitted efficiently.

Enode B

The eNodeB provides a variety of different roles in addition to controlling wireless communications. These are some examples:

1. Radio resource management: The eNodeB handles radio resource allocation, such as frequency bands and time slots, to ensure that user devices obtain the highest signal quality and data speeds feasible.

2. Security: The eNodeB is in charge of assuring the security of data sent over the air. To prevent unwanted access to user data, it employs encryption methods.

3. Mobility management: The eNodeB is in charge of managing user device mobility inside the network. It maintains track of each device's position and guarantees that it is constantly linked to the greatest available signal.

Handover in Enode In an LTE network, the X2 and S1 interfaces are used to facilitate handovers between different base stations (eNodeBs). The X2 interface is used for handovers that occur within the same LTE network, while the S1 interface is used for handovers that occur between different LTE networks or between an LTE network and another type of wireless network.

During an X2 handover, the source eNodeB and the target eNodeB exchange information about the quality of the signal and the channel conditions for the device that is being handed over. This allows the network to determine the best time and method for the handover. Once the handover is complete, the device will continue to communicate with the target eNodeB over the X2 interface.

During an S1 handover, the source and target eNodeBs exchange similar information,

but the handover is coordinated by the core network, which manages the connections between different parts of the LTE network. This allows for seamless handovers between different networks, ensuring that the device maintains a stable connection and high-quality service.

There are many types of vulnerabilities as discuss above. we will look at MITM attack.:

Active attacks [5]

MITM attacks are a serious security concern because they can allow attackers to steal sensitive information, such as login credentials or financial data, or to perform other malicious actions. Once the attacker has intercepted the signals, they could manipulate them in various ways to perform the MITM attack. For example, the attacker could redirect the victim's traffic to a different network controlled by the attacker, allowing the attacker to monitor and potentially alter the victim's communications. The attacker could also potentially impersonate the victim's device or the LTE network to perform other malicious actions.

Spoofing a phone's ID involves falsifying the identifying information that is sent by phone to wireless carriers and other devices. This can be done in various ways, such as by using a fake SIM card or by manipulating the phone's firmware.

When a person is spoofing their phone as another phone, they would typically use techniques to falsify the identifying information that is associated with the device. This could include manipulating the IMSI number, the serial number, and other identifying data to make the device appear to be a different phone. This can be done using a variety of methods, including specialized software, hardware devices, or even physical modifications to the phone itself. The goal of this type of spoofing is to trick the network into thinking that the phone is legitimate, allowing the person to gain unauthorized access

to the network or perform other illicit activities.

Denial-of-service (DoS) attack in an LTE network is a form of cyber assault that aims to overwhelm the network and prevent it from working correctly. A DoS assault on an LTE network can be carried out in a variety of methods, including flooding the network with bogus traffic, exploiting flaws in the network infrastructure, or deploying malware to disrupt the network. A DoS attack on an LTE network aims to prevent legitimate users from accessing the network, causing communication to be disrupted and inflicting severe harm to the network and its users.

Deauthentication packets are a type of wireless network management frame, which are typically sent using a raw socket. A raw socket is a type of network socket that allows a program to send and receive data at the lowest level of the network protocol stack. This allows a program to construct and send custom packets, such as deauthentication packets, that are not provided by the operating system.

Deauthentication packets are not generic packets and are specifically designed to disconnect a device from a wireless network. These packets contain information that tells the target device to disconnect from the network and are typically sent by the network itself, rather than by individual devices.

Passive attacks

If someone is able to ”sniff” your International Mobile Subscriber Identity (IMSI) number, they could potentially use it to track your location and monitor your communications.

The IMSI is a unique number that is assigned to each SIM card and is used to identify a specific mobile device on a wireless network. It is transmitted whenever a device connects to the network, so if an attacker is able to intercept and obtain your IMSI number, they could use it to track your location and monitor your communications. In addition to tracking and surveillance, an attacker who has your IMSI number could potentially use it to impersonate your device and perform other malicious actions. This could in-

clude making unauthorized calls or accessing sensitive information on your device. Overall, having your IMSI number stolen can be a serious security concern and it is important to protect your device and communication channels to prevent this from happening.

I have performed it using SDR. you can view it using the following link: `https://youtu.be/cgddHG8059k`

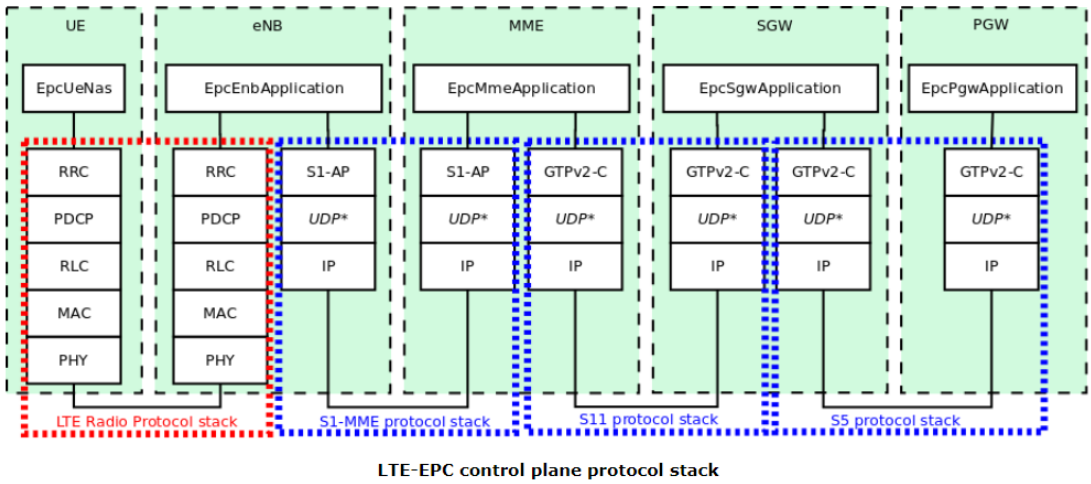

Figure 3.1: LTE Architecture in NS-3.

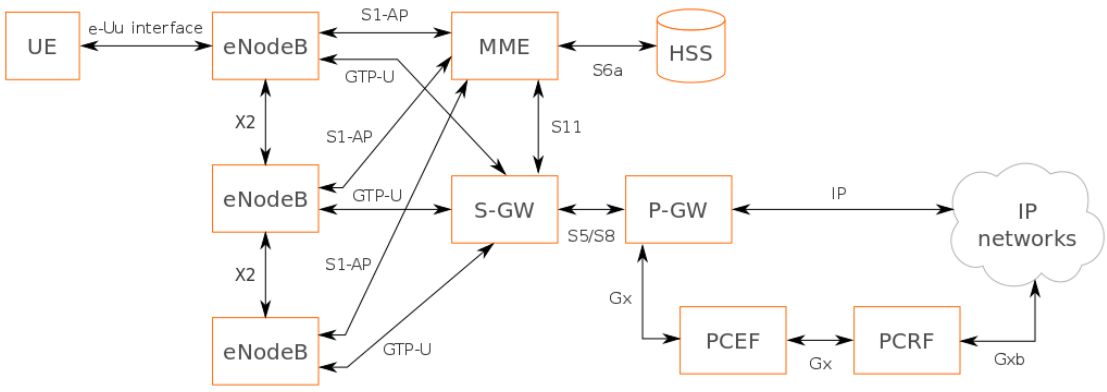

Figure 3.2: Communication links in an LTE Architecture

3.4 modules

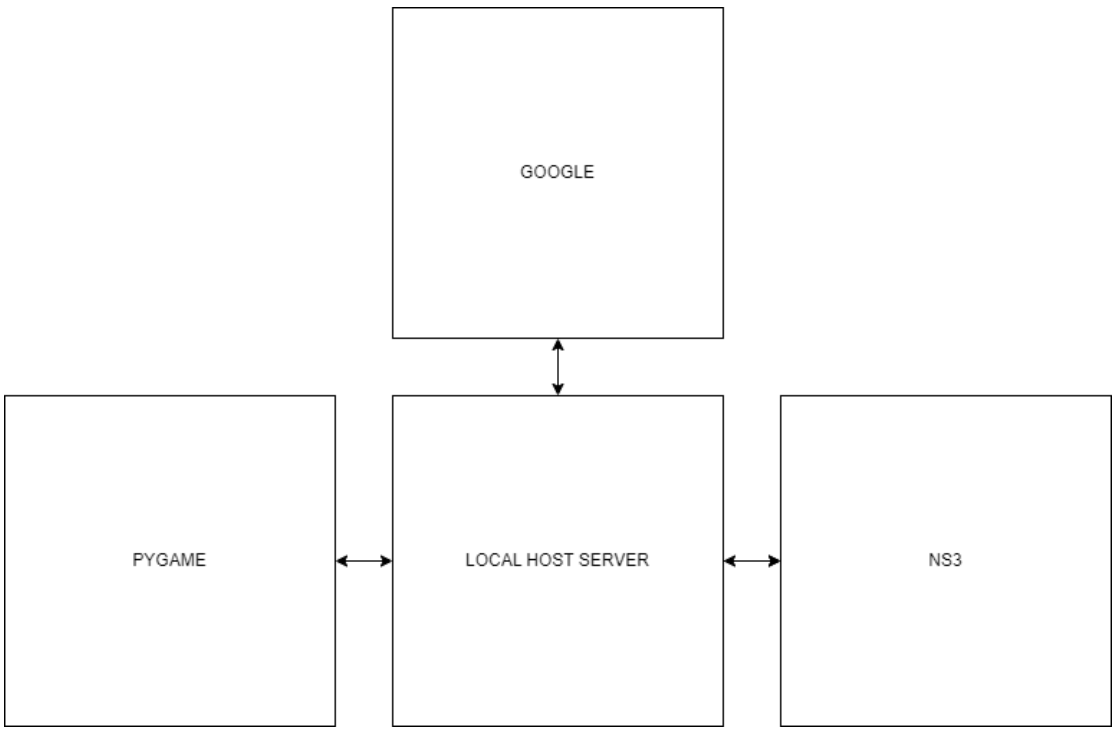

Figure 3.3: Block diagram of entire system

Setting up EPC core network

In the code below we have set up LTEhelper which will be used to design the core network.

```
lteHelper->SetEpcHelper (epcHelper);
lteHelper->SetSchedulerType ("ns3::RrFfMacScheduler");
lteHelper->SetHandoverAlgorithmType ("ns3::
    NoOpHandoverAlgorithm");
```

Connecting the internet and remote host to the lte network

```
Ptr<Node> pgw = epcHelper->GetPgwNode ();
```

```
// Create a single RemoteHost
NodeContainer remoteHostContainer;
remoteHostContainer.Create (1);
Ptr<Node> remoteHost = remoteHostContainer.Get (0);
InternetStackHelper internet;
internet.Install (remoteHostContainer);

// Create the Internet
PointToPointHelper p2ph;
p2ph.SetDeviceAttribute ("DataRate", DataRateValue (
   DataRate ("100Gb/s")));
p2ph.SetDeviceAttribute ("Mtu", UintegerValue (1500));
p2ph.SetChannelAttribute ("Delay", TimeValue (Seconds
   (0.010)));
NetDeviceContainer internetDevices = p2ph.Install (pgw,
    remoteHost);
Ipv4AddressHelper ipv4h;
ipv4h.SetBase (pchar, "255.0.0.0");
Ipv4InterfaceContainer internetIpIfaces = ipv4h.Assign
   (internetDevices);
Ipv4Address remoteHostAddr = internetIpIfaces.
   GetAddress (1);


// Routing of the Internet Host (towards the LTE
   network)
Ipv4StaticRoutingHelper ipv4RoutingHelper;
Ptr<Ipv4StaticRouting> remoteHostStaticRouting =
   ipv4RoutingHelper.GetStaticRouting (remoteHost->
```

```
    GetObject<Ipv4> ());
// interface 0 is localhost, 1 is the p2p device
```

Using static routing to connect to the network

```
  remoteHostStaticRouting->AddNetworkRouteTo (
     Ipv4Address ("9.0.0.0"), Ipv4Mask ("255.0.0.0"),
     1);
```

Initializing Enode and UE

```
NodeContainer ueNodes;
NodeContainer enbNodes;
enbNodes.Create (numberOfEnbs);
ueNodes.Create (numberOfUes);

// Install Mobility Model
Ptr<ListPositionAllocator> positionAlloc = CreateObject
   <ListPositionAllocator> ();
for (uint16_t i = 0; i < numberOfEnbs; i++)
  {
    positionAlloc->Add (Vector (distance * 2 * i -
       distance, 0, 0));
  }
for (uint16_t i = 0; i < numberOfUes; i++)
  {
    positionAlloc->Add (Vector (0, 0, 0));
  }
MobilityHelper mobility;
mobility.SetMobilityModel ("ns3::
```

```
    ConstantPositionMobilityModel");
  mobility.SetPositionAllocator (positionAlloc);
  mobility.Install (enbNodes);
  mobility.Install (ueNodes);

  // Install LTE Devices in eNB and UEs
  NetDeviceContainer enbLteDevs = lteHelper->
    InstallEnbDevice (enbNodes);
  NetDeviceContainer ueLteDevs = lteHelper->
    InstallUeDevice (ueNodes);

  // Install the IP stack on the UEs
  internet.Install (ueNodes);
  Ipv4InterfaceContainer ueIpIfaces;

  ueIpIfaces = epcHelper->AssignUeIpv4Address (
    NetDeviceContainer (ueLteDevs));

  // Attach all UEs to the first eNodeB
  for (uint16_t i = 0; i < numberOfUes; i++)
    {
      lteHelper->Attach (ueLteDevs.Get (i), enbLteDevs.
        Get (0));
    }
```

Installation Guide

All Steps related to installing and building the application can be found on Github https://github.com/or05554/NS-3-GUI-in-PYgames

CHAPTER 4

PROTOTYPING AND TESTING

4.1 Test cases

Right now we have one test case for our system. We are demonstrating man in the middle attack using a rogue enode b. Our aim is to see if the rogue Enode is able to alter data during communication of a UE with the remote server. The UE is trying to connect to Gmail.com, but the Rogue enode alters the domain to fakegmail.com. This can be observed in the results when we are using Wireshark to analyze the packets.

4.2 Results

GUI Figure 4.1 shows the GUI made using pygames. We used Pygames as it was the fastest way to design an application that could integrate with all other components. making an application in JAVA would have been more space efficient, as python is not good in memory management, hence it is a bit slow when we have multiple threads to operate. But since the network simulator configurations are in python, I feel it was a wise choice to use python instead of java.

In figuure 4.1 we can see the objects on the left. We have Enode B, a cell phone, an attacker device, and a server. We can drag and drop items to simulate them in the NS3 environment. All of the data is passed from Pygames to the NS3 when we press the play button. There is some tool present to view the data. All tools are present on the top part of the window. we have a wire tool to connect the devices together. we have a disconnect to remove the connection. We have a google button to start a webpage on the local host. this can be integrated with NS-3 so that we are sending the data to the server. We have a Wireshark tool and an architecture tool.

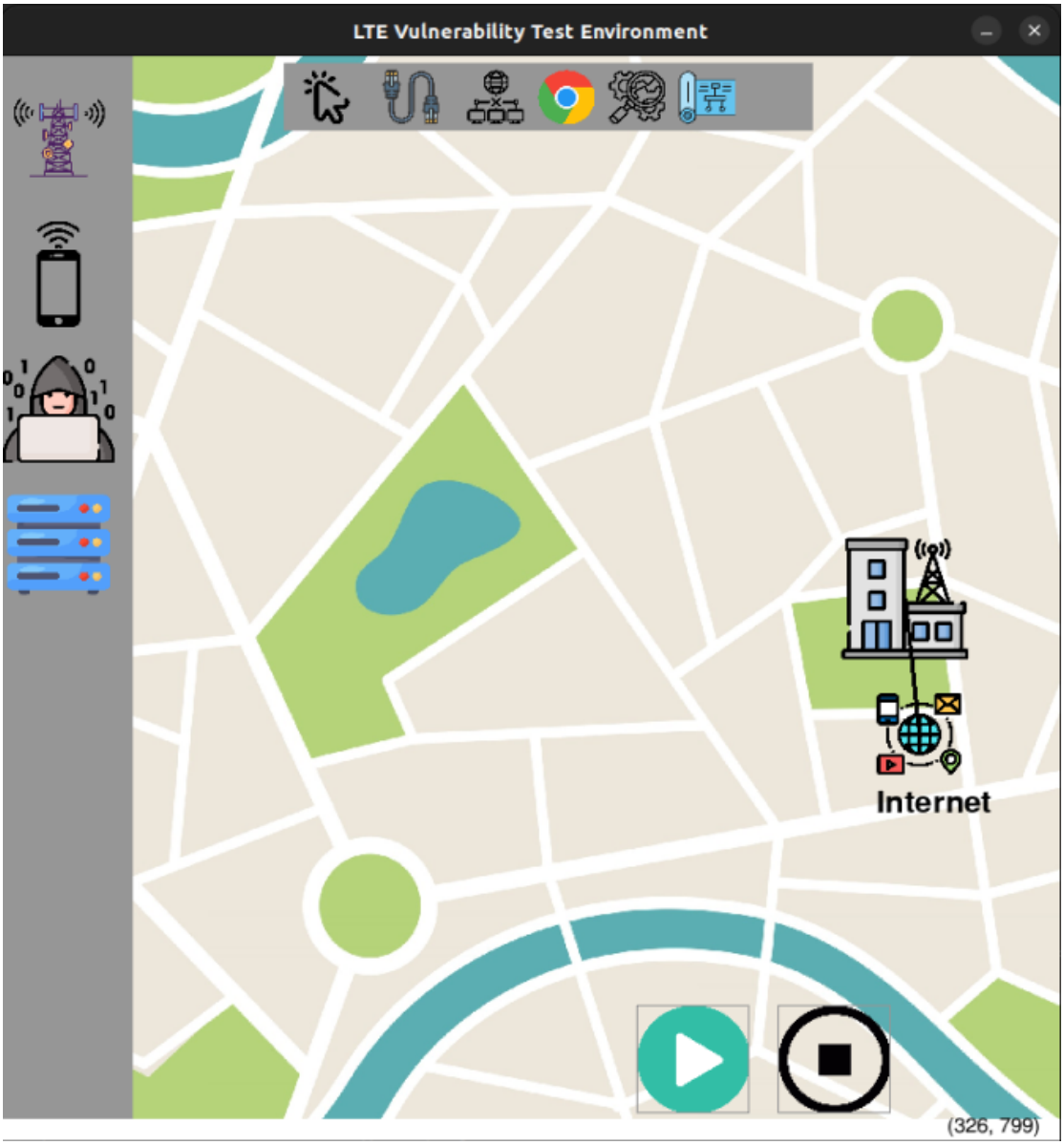

Figure 4.1: GUI for NS-3

Connections

Using the connection tool we can make a connection with the phone and the enode B. Som connections are premade, this is because they are part of the core network and cannot change. The attacker does not connect with the wire tool as it makes its own decision once deployed near a device.

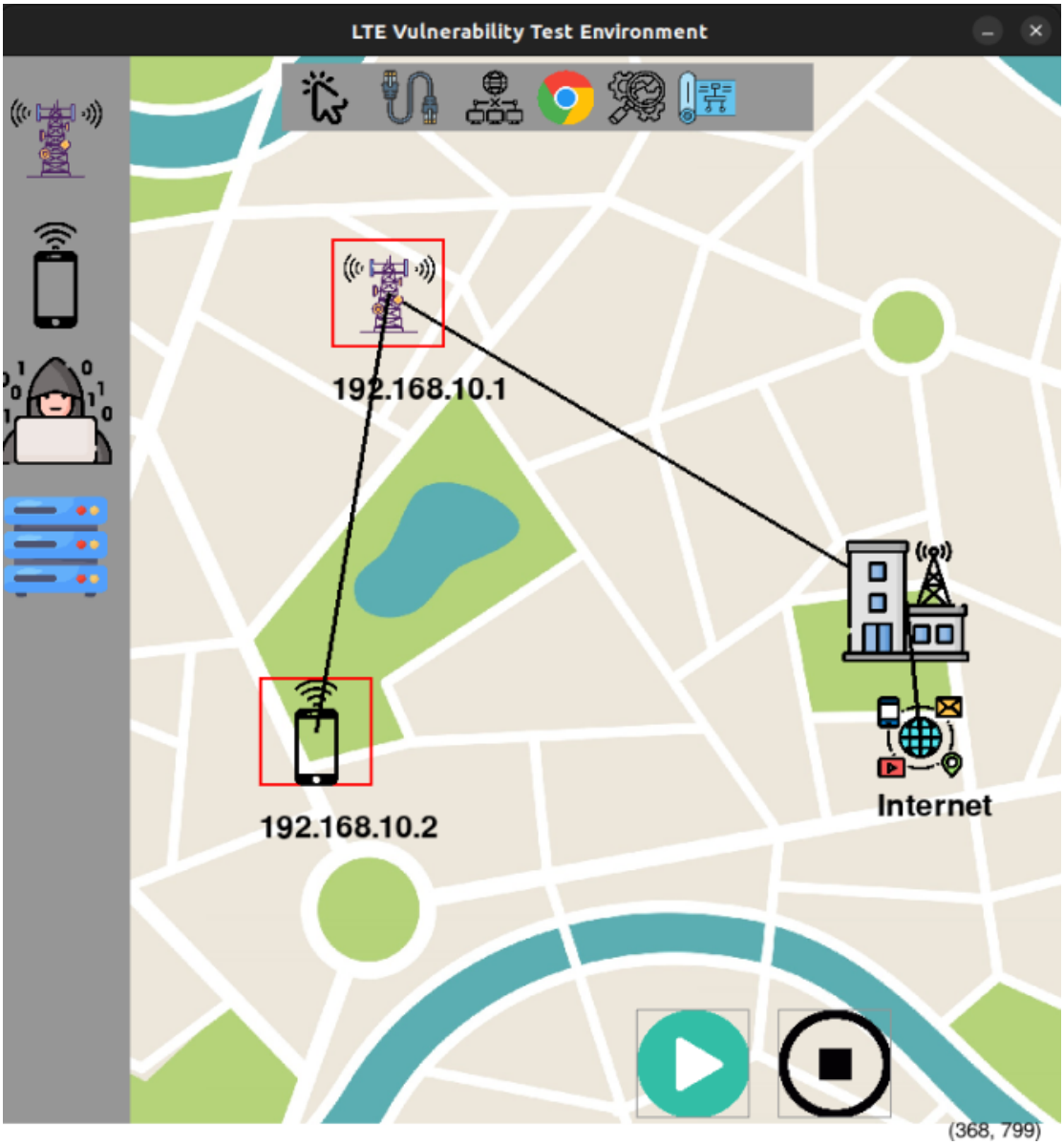

Figure 4.2: GUI for NS-3

play

Using the play feature, we can simulate the network we have created in NS-3. this will preload all the data. Now we can use the tool to view all of the information.

Figure 4.3: GUI for NS-3

wireshark TOOL

Wireshark is a free open-source application for Windows, Mac, Unix, and Linux platforms that analyses network traffic in real time. It intercepts data packets as they flow over a network interface (such as Ethernet, LAN, or SDRs) and converts them into useful information for IT experts and cybersecurity teams.

We can use wire shark to analyze our network and see if our rogue endoe is getting

the data and if it is altering the data or not.

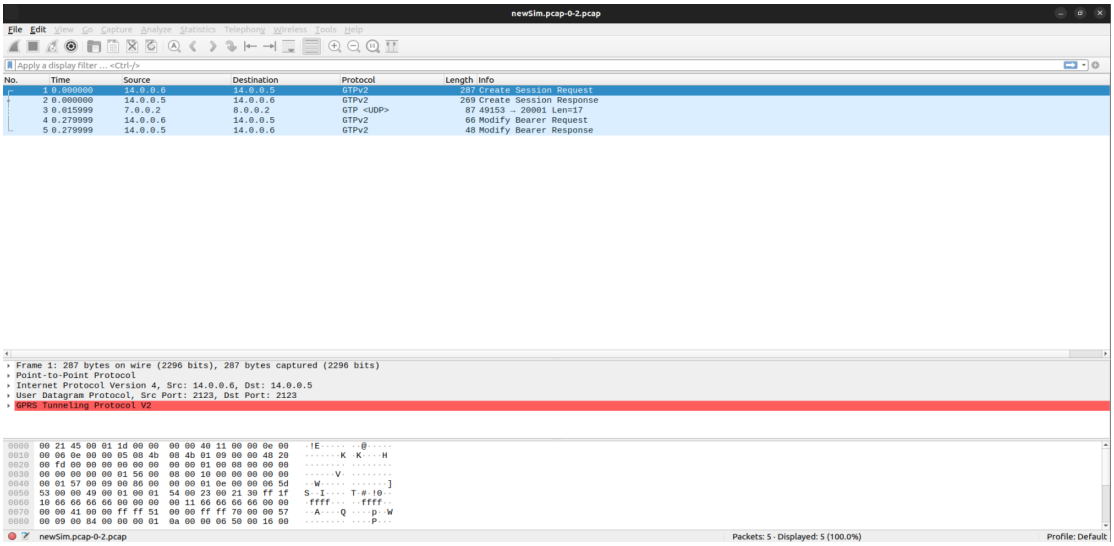

Figure 4.4: GUI for NS-3

Architecture TOOL

This can give more detail about the architecture implemented in Enode B.

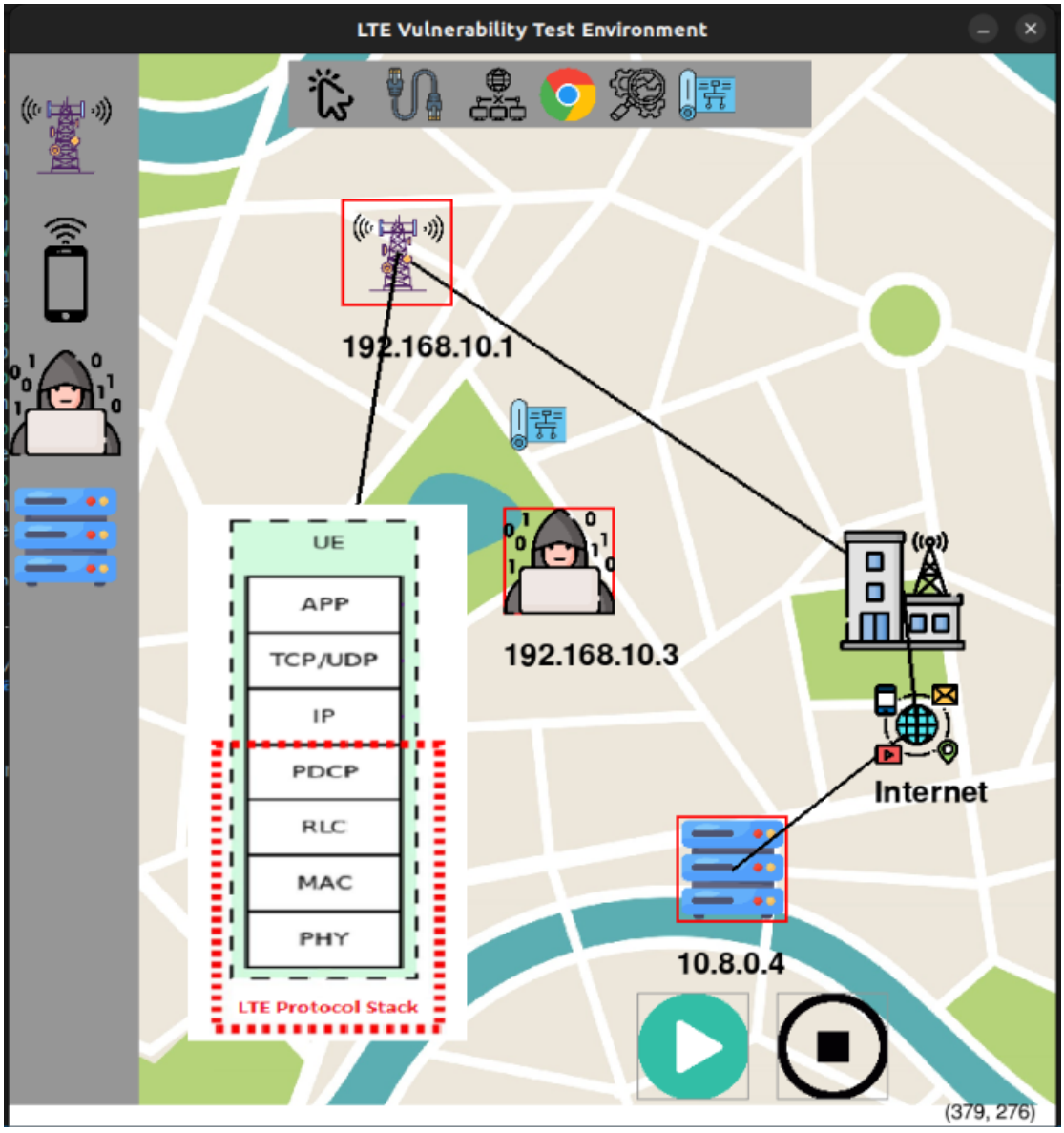

Figure 4.5: GUI for NS-3

Google TOOL

We were using a local host webpage to simulate an environment for the user to search for a website such as gmail.com and receive the page that is being requested. But in the rogue enode case we will see that it will not be the case.

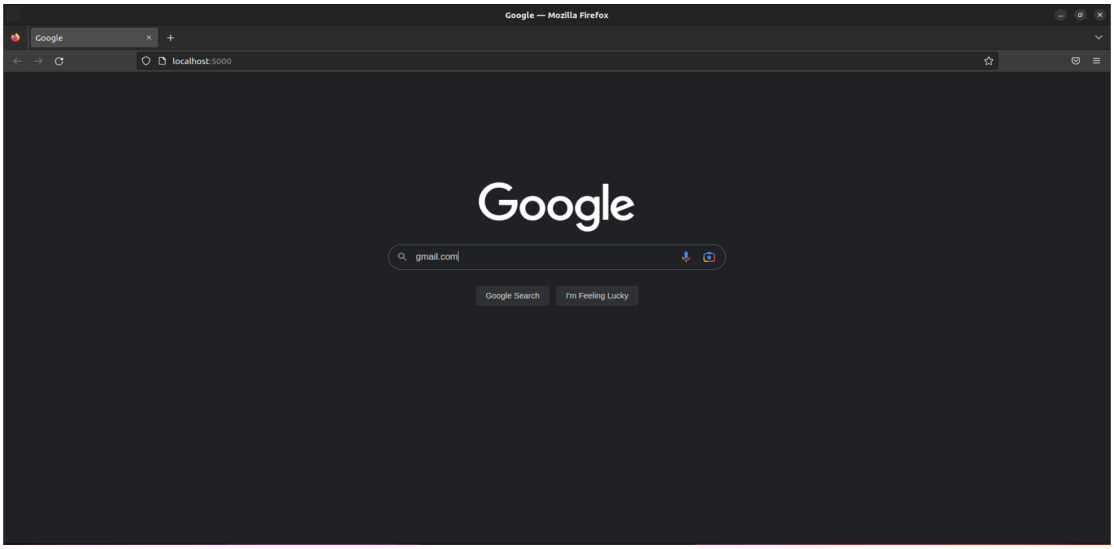

Figure 4.6: GUI for NS-3

Google -Gmail TOOL

As discussed above, as the rogue enode is active we will get the fake Gmail page.

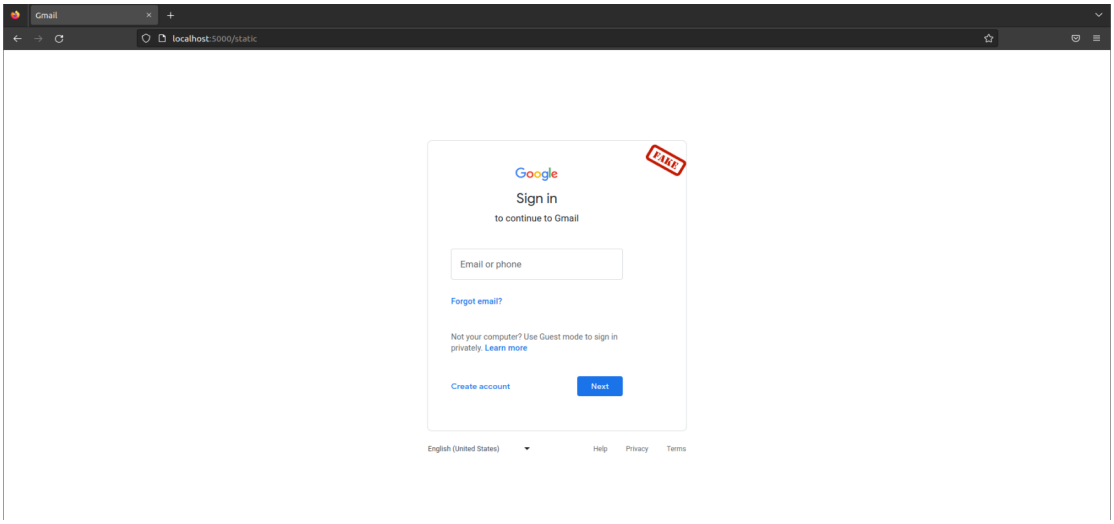

Figure 4.7: GUI for NS-3

To encrypt and decrypt data we are using OpenSSL to establish a DTLS session using the AES algorithm. Right now we just made a simple encryption program to test the functionality of integrating cryptography with NS3. The following code generates

two files using a key and salt values. The key is part of symmetric encryption, this means that there will be one key for encryption and decryption.

Figure 4.8: data is encrypted in a file and then sent to ns3 to be transported through the network to reach the remote host. Here, if the proper key exchange doesn't take place, the key can be observed by an attacker that is present in the communication and then can use for further decryption of data once the connection has been established.

Performing real attacks on similar networks to view the plausibility of a given attack.

In this example shown in figure 4.9, we have used a mobile phone with Charles's software to create a reverse proxy. The reverse proxy establishes two separate connections

with the client and the server. When the client wishes to establish a connection it sends a TLS handshake to the proxy. This proxy is configured beforehand so all the data goes through the computer that has Charles installed in it. If this was a man in the middle we will have a similar process to connect the client. The TLS handshake is between the computer and the phone. This is established with valid certificates as HTTPS is required for all modern browsers. Once the TLS handshake is established, the reverse proxy will then establish a TLS handshake with the required server, in this example we have daraz.com. Once The request from the client starts, the proxy decrypts the data and then encrypt it with the other session keys, and then sends it to the server. This way the proxy can see all the data. We can observe the figure above which has the phone opened to the Daraz website and connecting to the user account while on the right we see the Charles software that is viewing the data.

To evaluate the KPI we have used CIA triad

The CIA triad is a model for understanding the three fundamental security principles of information security: confidentiality, integrity, and availability

Confidentiality is the principle of ensuring that sensitive or private information is not disclosed to unauthorized individuals or systems. This can be achieved through various means, such as encryption, access controls, and secure communication channels.

Integrity is the principle of maintaining the accuracy, completeness, and consistency of information. This means that information cannot be modified or corrupted in any way without being detected, and that it can be trusted to reflect the true state of affairs.

Availability is the principle of ensuring that authorized users have access to information when they need it. This means that the information must be stored in a reliable and secure manner, and that it must be protected from disruptions or attacks that could prevent access to it.

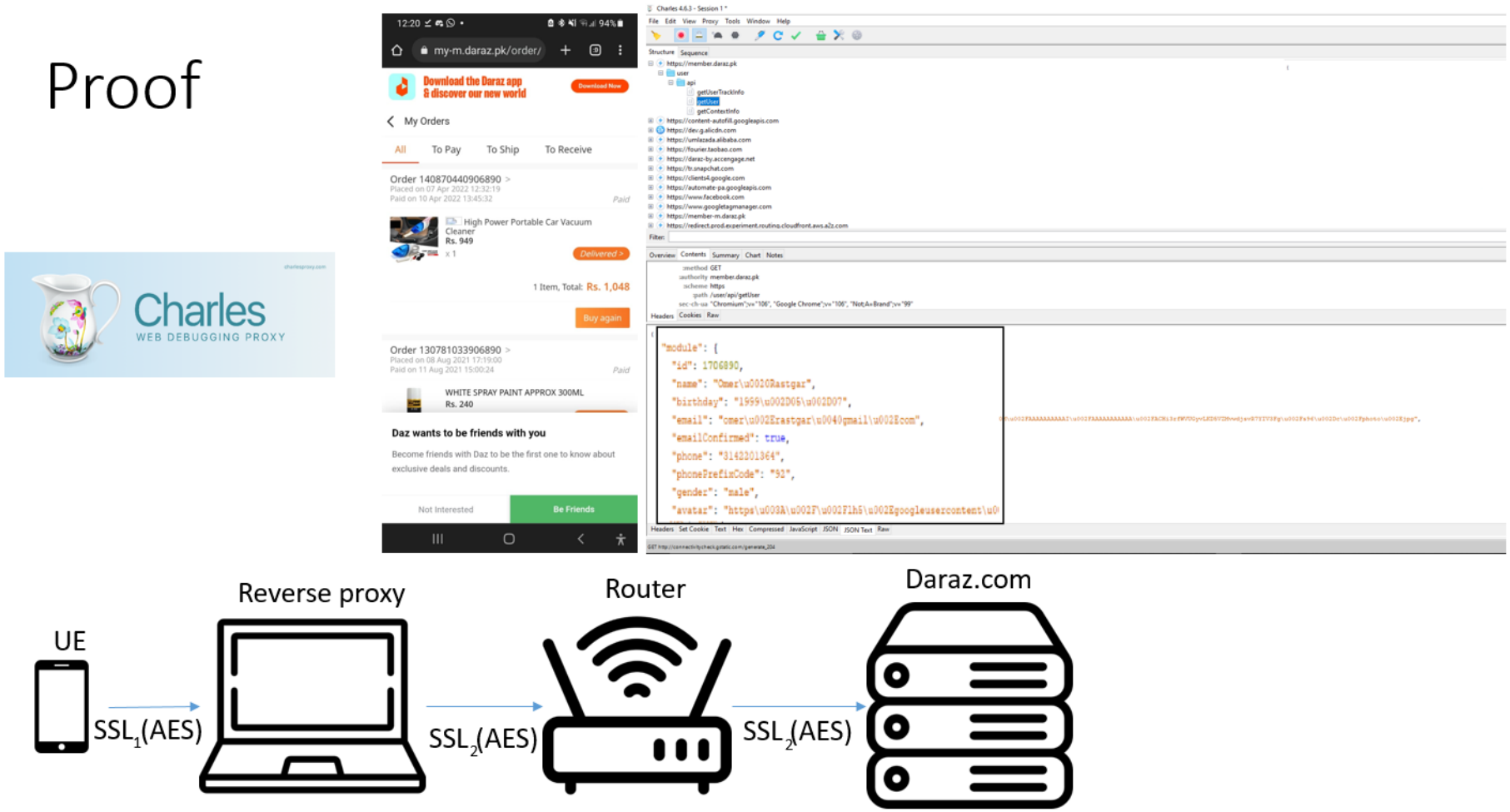

Figure 4.9: This is a proof of concept demonstration of Reverse Proxy which can be used for SSL Hijacking in a MITM attack.

attacks mentioned above cause all of these components to be effected hence we require a solution to simulate and protect users.

Gant chart for Capstone 1

		Capstone Fall 2022										
Task Name	**Team Members**	**Week 5**	**Week 6**	**Week 7**	**Week 8**	**Week 9**	**Week 10**	**Week 11**	**Week 12**	**Week 13**	**Week 14**	**Week 15**
Finalizing project proposal	Omer Rastagar											
Getting fimiliar with Cellular architecture	Omer Rastagar											
Literature review	Omer Rastagar											
Designing cellular architecture in NS3	Omer Rastagar											
Encryption using openSSL	Omer Rastagar											
Python game GUI	Omer Rastagar											
Building code for NS3	Omer Rastagar											
Rogue BTS archtecture	Omer Rastagar											
Mid progress Report	Omer Rastagar											
App development	Omer Rastagar											
Final Progress Report	Omer Rastagar											

Figure 4.10: Gant Chart

CHAPTER 5

CONCLUSION AND FUTURE WORK

Completion of the rouge Enode B.

Right now we have an Enode B that is connected to a UE. When the rogue enode B is enabled, a handover takes place that connects the UE to the other node B. The flow of packets is direct to the EPC core rather than first going to the original node b and then going to EPC core. It has been verified with Wireshark that the packet information is correct for Enode handover but it is not correct for a rogue Enode B.

Our first task will be to build a Enode function that has the same frequency as the legitimate Enode b. hence will have the same PCI number. Then we will be broadcasting at a higher power than the legitimate Enode b and we can display it using a graph of LTE RSSI, RSRP and RSRQ. We can show the interaction of packets in Wireshark where we will have the same ip for the UE as it will be a private address. The rouge Enode will have the same ip as the legitimate Enode and it will be broadcasting UE IP to connect with the legitimate Enode b. All of these nodes will be designed to reduce any interference from matching IP addresses.

Improve GUI + interaction with NS-3.

In many cases the data flow from the GUI and ns3 was not sufficient as the ip address where different in the py game and the ns3 model was instantiating some on its own. The remove feature is not designed and the simulation fails for more than two similar nodes such as two UE or Two Enode B.

Our task will be to keep the IP address constant for those NS3 nodes which we cannot change and provide the ip for ns3 node from pygames if they are editable. One of the nodes we can change is the PWG and remote host IP address so we can have them

as editable nodes.

For remove feature we have to use object oriented programming to handle all memory allocations and remove all the data associated with the node in order for it to work properly.

Two get more than one UE we will need to configure NS-3 to be dynamic and have the ability to connect automatically with multiple UE, there can be about 295 UE connected to one Enode. The mobility module manages the distance from the enode B and the power being transmitted.

Testing other vulnerabilities on LTE network.
Once we have a complete functioning environment that is dynamic. We can add different scenarios that can be loaded to try different windows of pygame to show implementation such as a DOS attack or impersonation attack.

For DOS attacks we can use the system we built with the rogue enode. We can force you of a network using a rogue enode.

Our aim would be to completely design a DOS attack, if we are able to complete this before our final evaluation we can work on any other vulnerability.

The main work will be in designing a separate interaction diagram in pygames to illustrate the packet transfer and in ns3 we will modify our rogue enode B to send Deauthentication packets.

(Tentative) demonstrating a solution only in ns3 by using the exiting architecture If we have completed the tasks explained above, we can describe a way to save the UE from a rogue Enode. This can be done by monitoring the power in UE and flagging

any irregular increase in power if the distance is not changing. All of the modules have been made for this implementation the only requirement will be to create a logic for detecting this and displaying it on PYgames.

Big picture In the final evaluation for capstone two, we will have a simulation environment for LTE. This tool can be used for education as it can demonstrate two major LTE attacks. It can be open-sourced for other people to work and build on top of the code. NS3 is well known for its simulation capabilities but until now there have been very few tools for representing nodes in GUI. This can be a starting point for a generic system used for other research purposes. This is achieved by the fact that all of the modules that I have made in PYgames are separate objects that can be called for any dynamic application.

Deliverables for (EE492)

1. Mid evaluation: Completing the work left in capstone one and improving the GUI.
2. Final evaluation: Design another simulation using existing architecture to simulate another vulnerability.

Appendices

- Include operational details or a manual of your prototype as an appendix. Most of your prototypes will be disassembled for parts, so this manual should include sufficient details to recreate prototype
- Include all drawings: Circuit schematics, PCB layouts, CAD designs
- Include code (if necessary), software scripts or hardware drivers, etc.
- HW component details , include all details like data sheet, vendor name & cost.
- Any other data related to your project which can be useful for future use.

APPENDIX A

MANUAL OF PROTOTYPE

Include the above distributed in various chapters.

REFERENCES

[1] W. b. admin and W. by, *Tag: Lte*.

[2] F. SANDU, S. CSEREY, and E. MILE-CIOBANU, "Simulation of lte signaling," *Advances in Electrical and Computer Engineering*, vol. 10, no. 2, 108–114, 2010.

[3] Y. E. El Idrissi, N. Zahid, and M. Jedra, "An efficient authentication protocol for 5g heterogeneous networks," *Ubiquitous Networking*, 496–508, 2017.

[4] *Cve-2019-3568*.

[5] D. Forsberg, H. Gunther, W.-D. Moeller, and V. Niemi, *LTE Security*. John Wiley amp; Sons, 2013.

[6] *Black hat europe 2016 eu-16-holtmanns-detach-me-not*.

[7] *Security guidelines for uicc profiles version 1.0 12 june 2020 - gsma*.

[8] C. Cox, *An introduction to LTE LTE, LTE-Advanced, SAE, VoLTE and 4G mobile communications*. Wiley, 2014.

VITA

The length of the vita is preferably one page. It may include the place of birth and should be written in third person. This vita is similar to the author biography found on book jackets.