Progress Update

SALADS
Students Assembling a Location Access Detection System

Team Roles

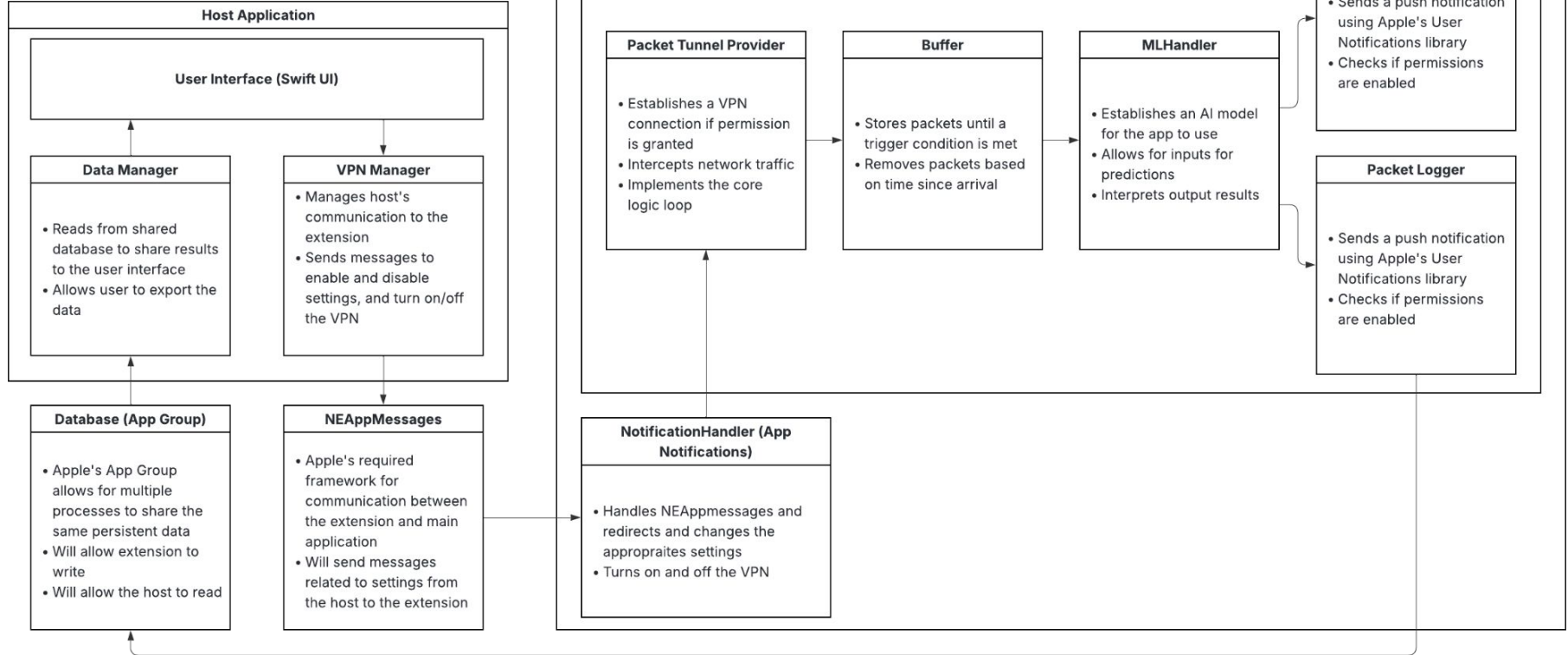Ethan Youmans - Project Manager, Lead Mobile App Developer

Luke Sutor - Lead AI Model Developer
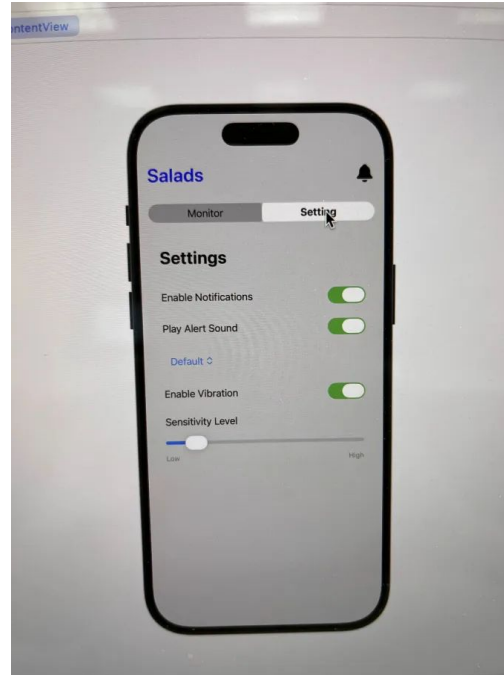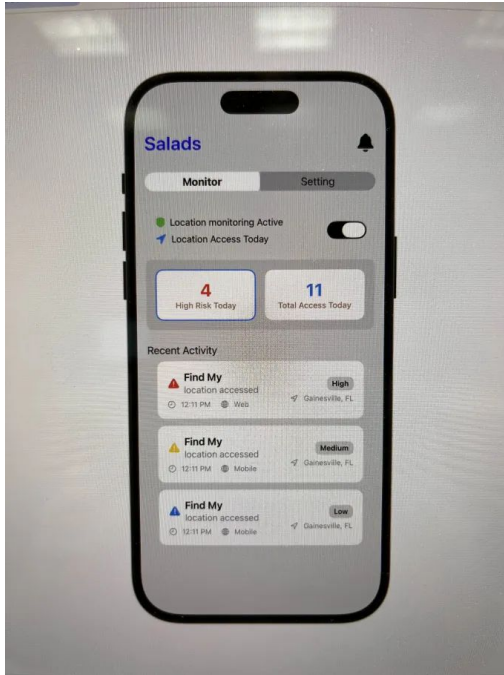
Bhaskar Gnanasakthi - Data Developer

Yao Wen Liu - Frontend Developer

Aiden Parsons – Developer

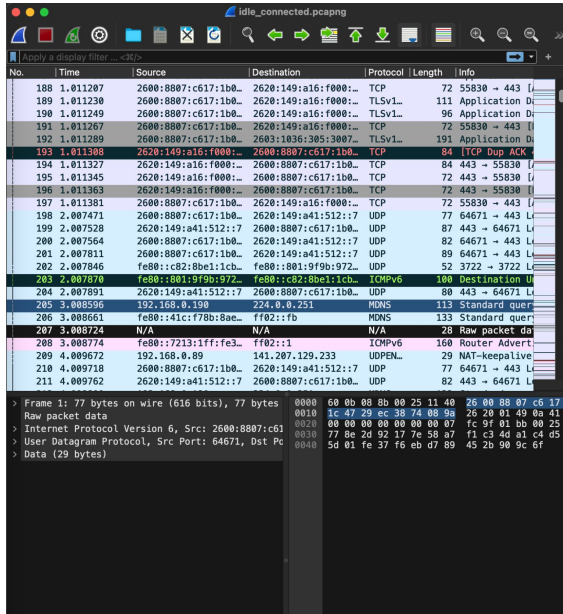# iOS Application Component Architecture

Host Application

User Interface (Swift UI)

Data Manager

- Reads from shared database to share results to the user interface
- Allows user to export the data

VPN Manager

- Manages host's communication to the extension
- Sends messages to enable and disable settings, and turn on/off the VPN

Database (App Group)

- Apple's App Group allows for multiple processes to share the same persistent data
- Will allow extension to write
- Will allow the host to read

NEAppMessages

- Apple's required framework for communication between the extension and main application
- Will send messages related to settings from the host to the extension

Extension

Pipe & Filter Pipeline

Packet Tunnel Provider

- Establishes a VPN connection if permission is granted
- Intercepts network traffic
- Implements the core logic loop

Buffer

- Stores packets until a trigger condition is met
- Removes packets based on time since arrival

MLHandler

- Establishes an AI model for the app to use
- Allows for inputs for predictions
- Interprets output results

NotificationHandler (Push Notifications)

- Sends a push notification using Apple's User Notifications library
- Checks if permissions are enabled

Packet Logger

- Sends a push notification using Apple's User Notifications library
- Checks if permissions are enabled

NotificationHandler (App Notifications)

- Handles NEAppmessages and redirects and changes the appropraites settings
- Turns on and off the VPN

User Interface Diagrams

Wireshark Monitoring
Data Collection

Experiment Design

		Requesting location	
	2	Once every 1 minute	1
	2	Requesting location Once every 2 minutes	1
	2	Requesting location Twice every 1 minute	1
	2	No requests Both phones idle	1

idle_connected.pcapng

No.	Time	Source	Destination	Protocol	Length	Info
188	1.011207	2600:8807:c617:1b0…	2620:149:a16:f000:…	TCP	72	55830 → 443 [
189	1.011230	2600:8807:c617:1b0…	2620:149:a16:f000:…	TLSv1…	111	Application D
190	1.011249	2600:8807:c617:1b0…	2620:149:a16:f000:…	TLSv1…	96	Application D
191	1.011267	2600:8807:c617:1b0…	2620:149:a16:f000:…	TCP	72	55830 → 443 [
192	1.011289	2600:8807:c617:1b0…	2603:1036:305:3007…	TLSv1…	191	Application D
193	1.011308	2620:149:a16:f000:…	2600:8807:c617:1b0…	TCP	84	[TCP Dup ACK
194	1.011327	2620:149:a16:f000:…	2600:8807:c617:1b0…	TCP	84	443 → 55830
195	1.011345	2620:149:a16:f000:…	2600:8807:c617:1b0…	TCP	72	443 → 55830 [
196	1.011363	2620:149:a16:f000:…	2600:8807:c617:1b0…	TCP	72	443 → 55830 [
197	1.011381	2620:149:a16:f000:…	2600:8807:c617:1b0…	TCP	72	55830 → 443 L
198	2.007471	2620:149:a41:512::7	2600:8807:c617:1b0…	UDP	77	64671 → 443 L
199	2.007528	2620:149:a41:512::7	2600:8807:c617:1b0…	UDP	87	443 → 64671 L
200	2.007564	2600:8807:c617:1b0…	2620:149:a41:512::7	UDP	77	64671 → 443 L
201	2.007811	2600:8807:c617:1b0…	2620:149:a41:512::7	UDP	89	64671 → 443 L
202	2.007846	2600:8807:c617:1b0…	fe80::801:9f9b:972…	UDP	52	3722 → 3722 L
203	2.007870	fe80::801:9f9b:972…	fe80::c82:8be1:1cb…	ICMPv6	100	Destination U
204	2.007891	2620:149:a41:512::7	2600:8807:c617:1b0…	UDP	80	443 → 64671 L
205	3.008596	192.168.0.190	224.0.0.251	MDNS	113	Standard quer
206	3.008661	fe80::41c:f78b:8ae…	ff02::fb	MDNS	133	Standard quer
207	3.008724	N/A	N/A	N/A	28	Raw packet da
208	3.008774	fe80::7213:1ff:fe3…	ff02::1	ICMPv6	160	Router Advert
209	4.009672	192.168.0.89	141.207.129.233	UDPEN…	29	NAT-keepalive
210	4.009718	2600:8807:c617:1b0…	2620:149:a41:512::7	UDP	77	64671 → 443 L
211	4.009762	2620:149:a41:512::7	2600:8807:c617:1b0…	UDP	82	443 → 64671 L

> Frame 1: 77 bytes on wire (616 bits), 77 bytes
  Raw packet data
> Internet Protocol Version 6, Src: 2600:8807:c61
> User Datagram Protocol, Src Port: 64671, Dst Po
> Data (29 bytes)
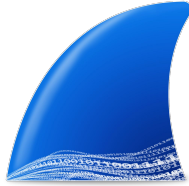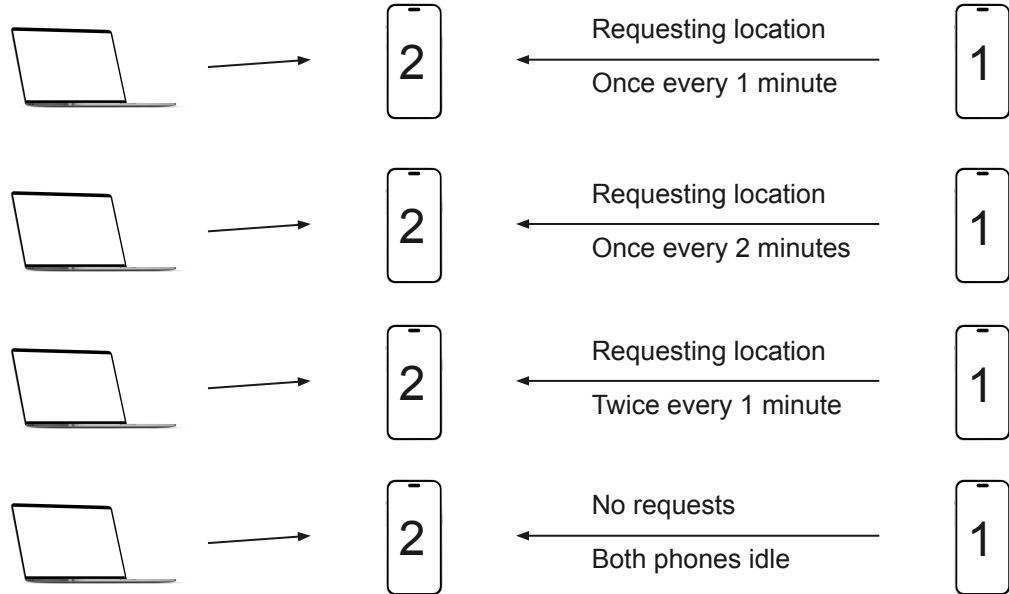
Machine Learning Model

We are using a Multiple Instance Learning Temporal Convolutional Network.

- We will be predicting if the user's location was accessed based on a group of packets opposed to individual packets.
  - Greater model efficiency for on-device inference
  - Different sized groups of packets to handle unpredictable traffic
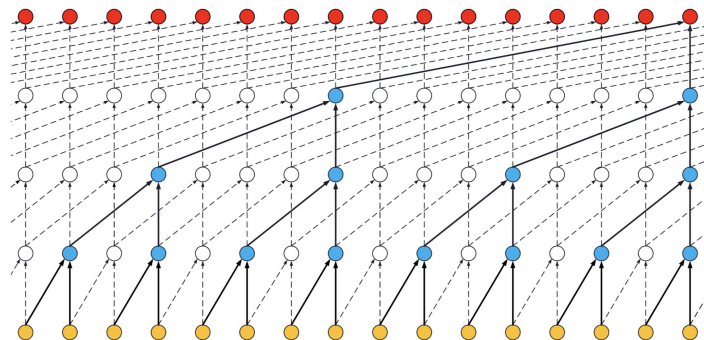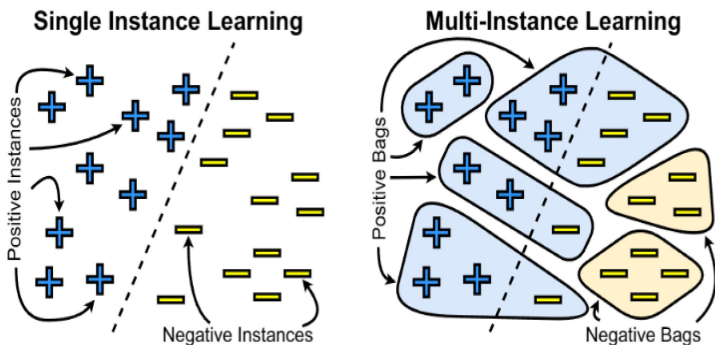
Machine Learning Input and Output

Our model takes the following features from each packet as input:

length	l4_tcp	l4_udp	l4_icmp
direction_out	src_port	dst_port	tcp_syn
tcp_ack	tcp_fin	tcp_rst	flow_hash
iadelta			

And it outputs a probability from 0 to 1 for whether or not the group of packets contains a location access

Completed Milestones

- First functional ML model
- Baseline training script and data loader
- Minimum UI
- VPN Packet Sniffer (50%)
- App architecture

Next Steps

- Complete packet pipeline
  - Ensure privacy and accuracy
- Develop Front End to a more finalized state
- Fine tune our data collection methods
- Run machine learning experiments and hyperparameter optimization
- Get the machine learning model running on an iPhone