MICHIGAN STATE
UNIVERSITY

Project Plan Presentation
Hybrid Cyberattack Simulator

The Capstone Experience

Team Vectra AI

Henry Barton
Alisha Brenholt
Nathan Motzny
Campbell Robertson
Andrew Talbott

Department of Computer Science and Engineering
Michigan State University

Spring 2024

From Students…
…to Professionals

Project Sponsor Overview

VECTRA®

- Leader in Cybersecurity
- Focus on network security
- Utilizes AI to detect, investigate, and report attacks
- Modular security to provide customized coverage

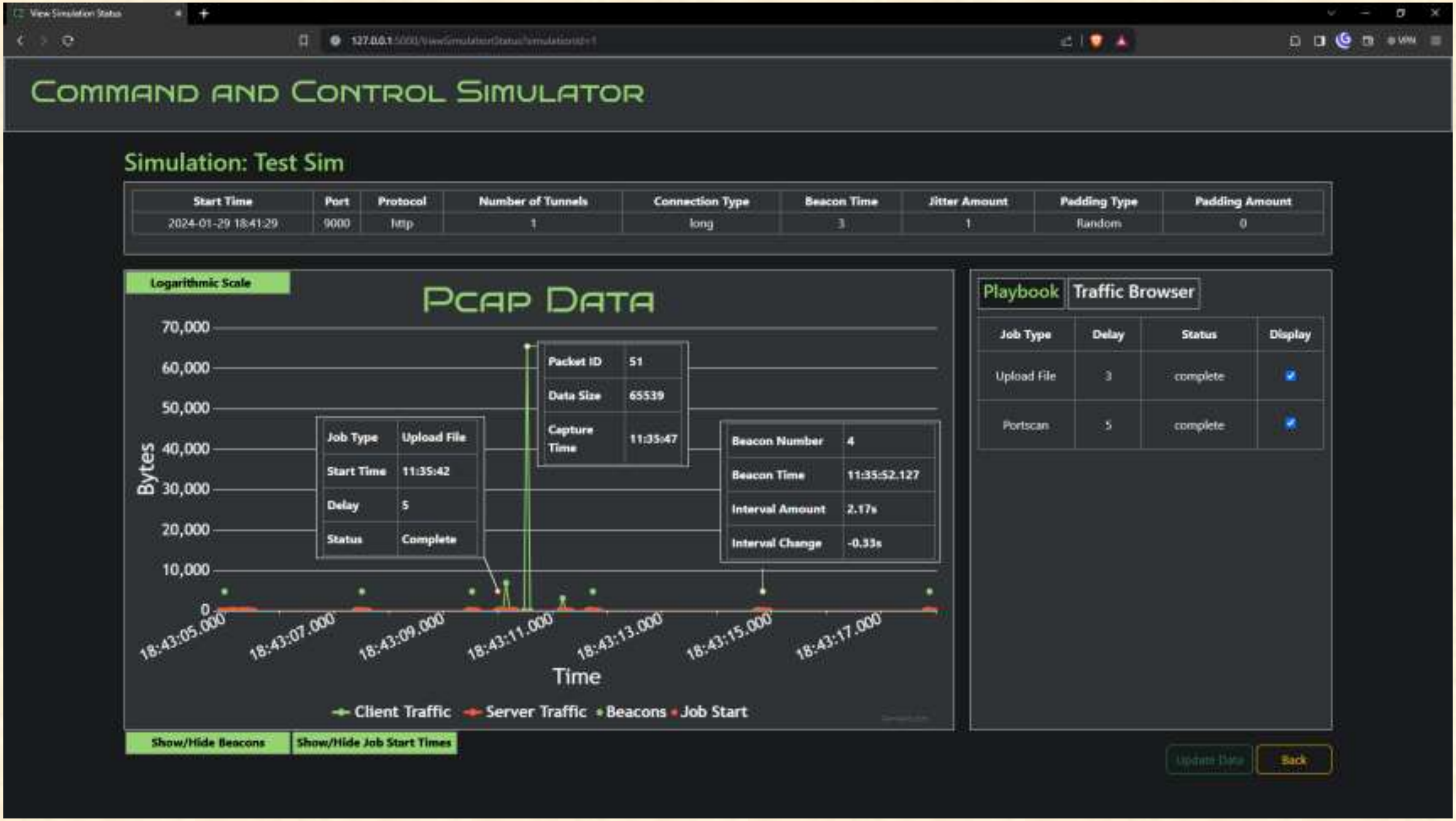
Project Functional Specifications

- Vectra's AI models need relevant training data to maintain accuracy
- Several more network protocols and hybrid attack capabilities are being added to the Command & Control (C2) Simulator
- Vectra will be able to prepare for diversified attacks using many different protocols and attack vectors

Project Design Specifications

- Hybrid Attack Simulation
  - MAAD-AF
  - DeRF
- Advanced C2
  - Webshells
  - Beaconless servers
- UI Enhancements

Screen Mockup: Simulation Overview with Playbook Display



Command and Control Simulator

Simulation: Test Sim

Start Time	Port	Protocol	Number of Tunnels	Connection Type	Beacon Time	Jitter Amount	Padding Type	Padding Amount
2024-01-29 18:41:29	9000	http	1	long	3	1	Random	0

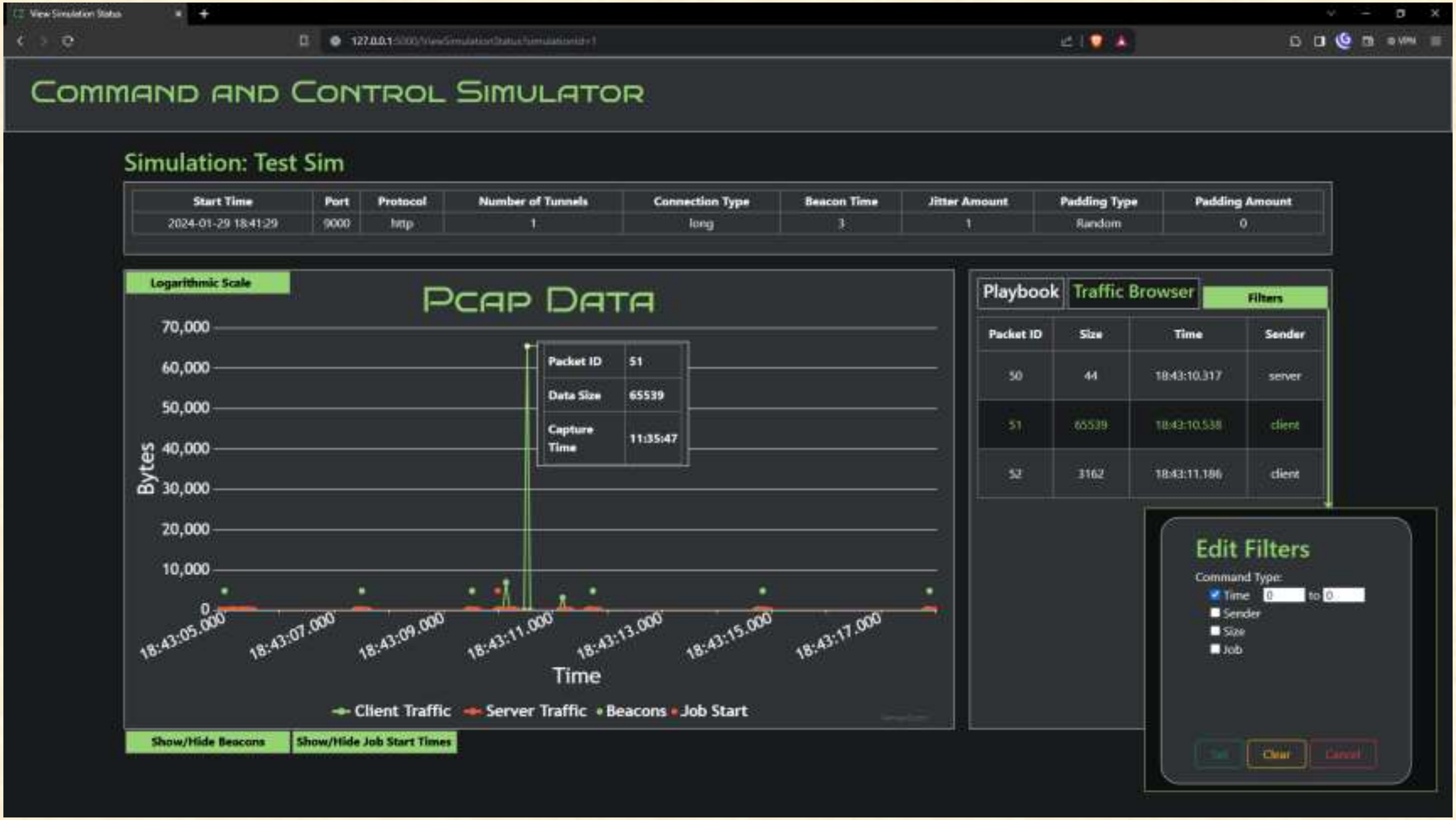
Job Type	Upload File
Start Time	11:35:42
Delay	5
Status	Complete

Packet ID	51
Data Size	65539
Capture Time	11:35:47

Beacon Number	4
Beacon Time	11:35:52.127
Interval Amount	2.17s
Interval Change	-0.33s

Job Type	Delay	Status	Display
Upload File	3	complete	☑
Portscan	5	complete	☑

Screen Mockup: Simulation Overview with Browser Display



Command and Control Simulator

Simulation: Test Sim

Start Time	Port	Protocol	Number of Tunnels	Connection Type	Beacon Time	Jitter Amount	Padding Type	Padding Amount
2024-01-29 18:41:29	9000	http	1	long	3	1	Random	0

Packet ID	51
Data Size	65539
Capture Time	11:35:47

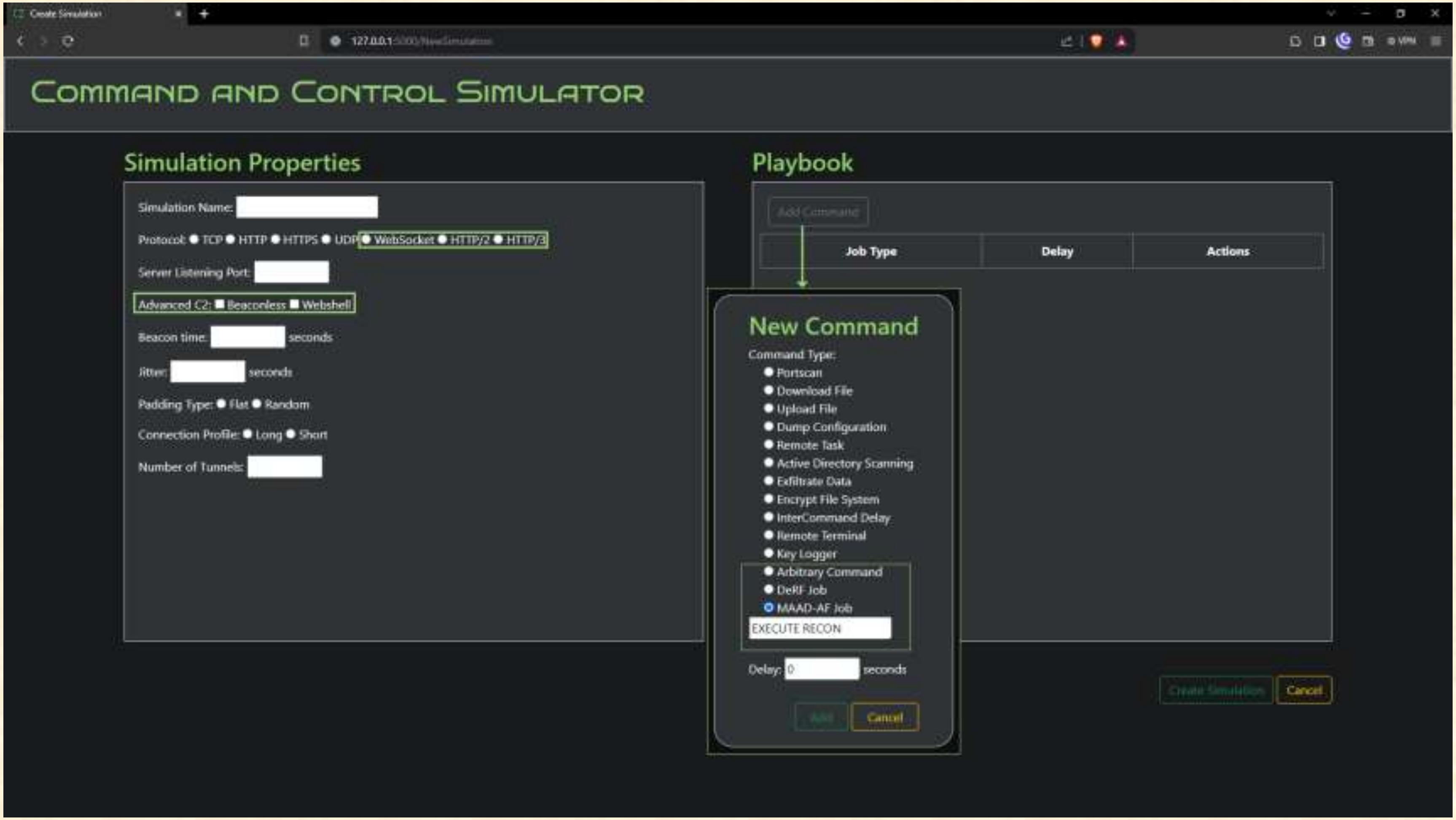
Packet ID	Size	Time	Sender
50	44	18:43:10.317	server
51	65539	18:43:10.538	client
52	3162	18:43:11.186	client

Edit Filters

Command Type:

- ☑ Time 0 to 0
- □ Sender
- □ Size
- □ Job

Screen Mockup: Create Simulation Page



Simulation Properties

Simulation Name:

Protocol: ○ TCP ○ HTTP ○ HTTPS ○ UDP ○ WebSocket ○ HTTP/2 ○ HTTP/3

Server Listening Port:

Advanced C2: □ Beaconless □ Webshell

Beacon time: seconds

Jitter: seconds

Padding Type: ○ Flat ○ Random

Connection Profile: ○ Long ○ Short

Number of Tunnels:

Playbook

Job Type	Delay	Actions

New Command

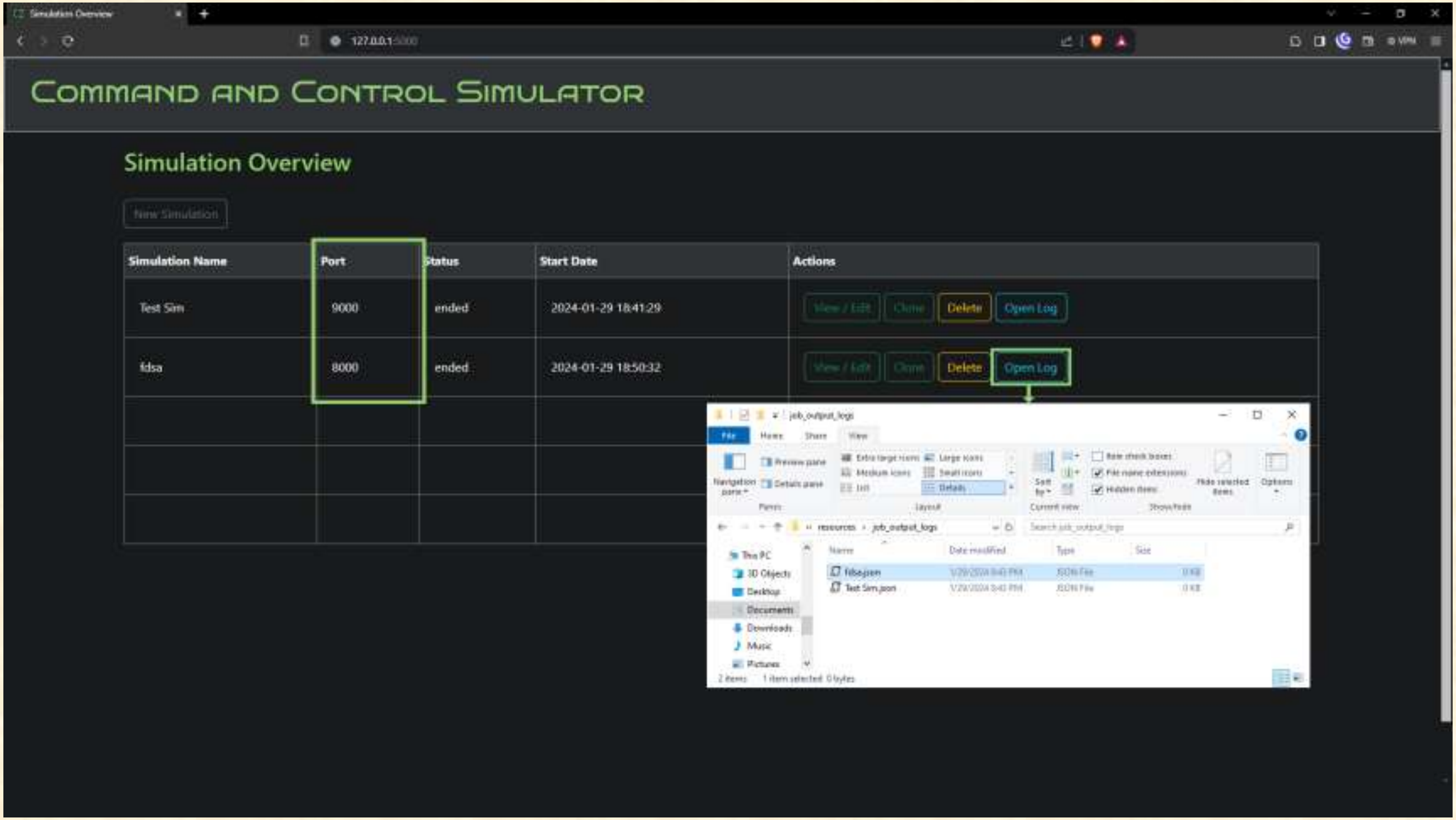
Command Type:

- ○ Portscan
- ○ Download File
- ○ Upload File
- ○ Dump Configuration
- ○ Remote Task
- ○ Active Directory Scanning
- ○ Exfiltrate Data
- ○ Encrypt File System
- ○ InterCommand Delay
- ○ Remote Terminal
- ○ Key Logger
- ○ Arbitrary Command
- ○ DeRF Job
- ● MAAD-AF Job

EXECUTE RECON

Delay: 0 seconds

Screen Mockup: Simulation Overview Page



Simulation Overview

Simulation Name	Port	Status	Start Date	Actions
Test Sim	9000	ended	2024-01-29 18:41:29	View / Edit Clone Delete Open Log
fdsa	8000	ended	2024-01-29 18:50:32	View / Edit Clone Delete Open Log

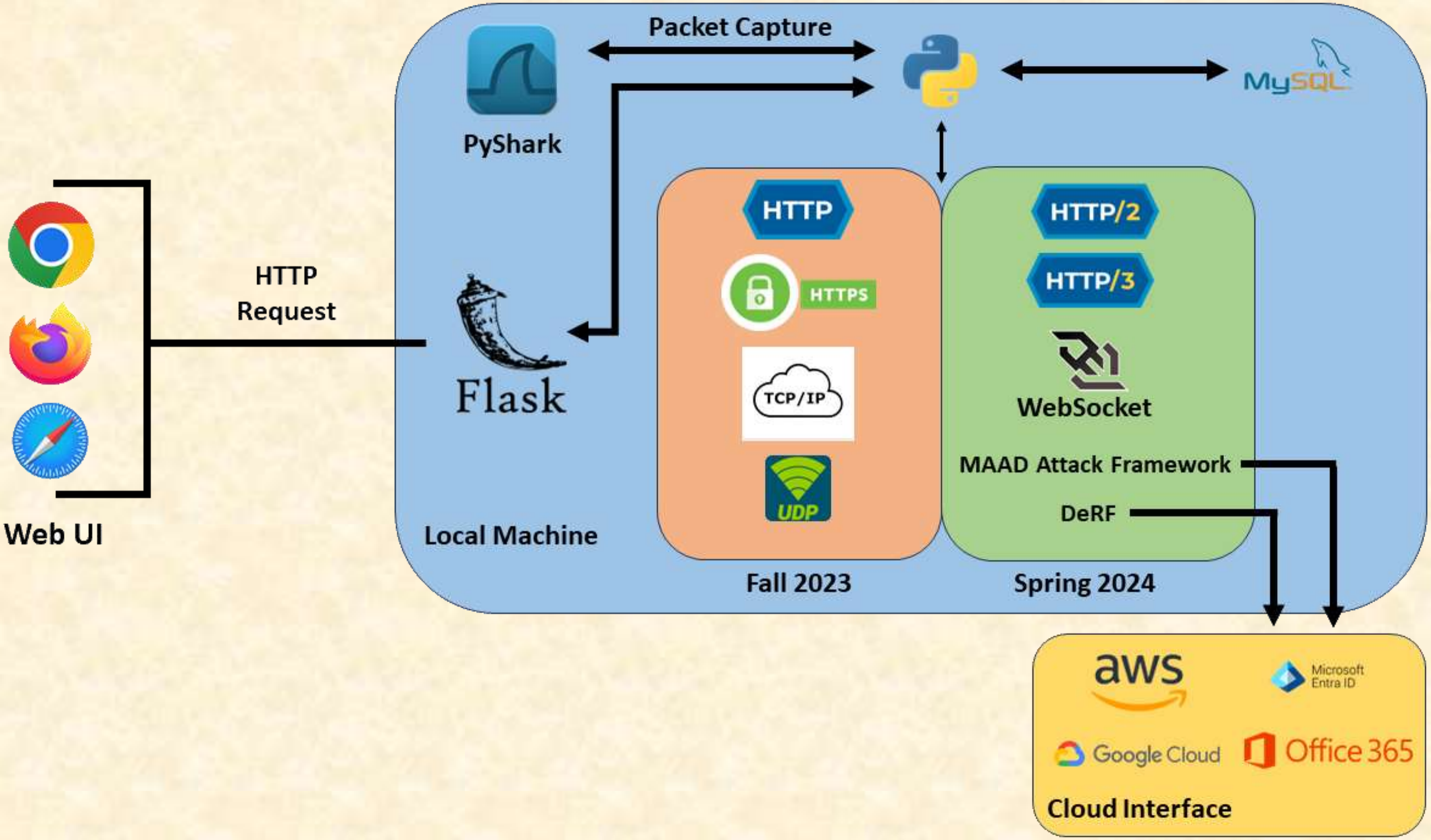
Name	Date modified	Type	Size
fdsa.json	1/29/2024 6:40 PM	JSON File	0 KB
Test Sim.json	1/29/2024 6:40 PM	JSON File	0 KB

2 items 1 item selected 0 bytes

Project Technical Specifications

- Implementing WebSocket, HTTP/2 and HTTP/3
- Also adding MAAD-AF and DeRF frameworks for Hybrid Attacks
- PyShark library to capture network packets, and MySQL to store them
- All of this is built on Python3

Project System Architecture



Project System Components

- Hardware Platforms
  - AWS Server
  - Google Cloud Server
  - Two Lab PCs
- Software Platforms / Technologies
  - Python
  - MySQL
  - Flask
  - VSCode

Project Risks

- Compatibility
  - Make sure all third-party apps work together
  - Using active libraries and using version control
- Generating realistic data
  - Generate realistic enough data for AI models to train on
  - Analyzing real world attacks and mimicking their outputs
- Performance Issues
  - Make large amounts of data in reasonable amounts of time
  - Spending time optimizing code; looking at distributed computing
- Portability
  - The program needs to be able to run on multiple OS without issue
  - Using cross-platform libraries and allowing API calls to server to abstract user operating system

Questions?

? ? ? ?

? ?

? ? ?