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
Project Sponsor Overview

- 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
Screen Mockup: Simulation Overview with Browser Display
Screen Mockup: Create Simulation Page
Screen Mockup: Simulation Overview Page
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

Diagram showing the project system architecture with components such as Flask, PyShark, HTTP, HTTPS, TCP/IP, UDP, MySQL, and various web technologies like WebUI, Chrome, Mozilla, Firefox, and cloud interfaces like AWS, Google Cloud, and Office 365.
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?