90S NET

Contact Info

- Group Members
  - Chad Lape: lapech@mail.uc.edu
  - Joshua Hale: haleju@mail.uc.edu
- Advisor
  - Will Hawkins: hawkinwh@ucmail.uc.edu

Goals

This senior design project, called "90s Net" aims to recreate the network and systems of the University of Cincinnati West main campus in the 1990s in a completely virtual environment. Users should be able to connect to this network and interact with it as if it were a true network. Admins should be able to control and view aspects of the network using common network tools. An interface should be able to create network views and run automation's triggered by user actions all from an externa l perspective. In total, this network should be an effective training ground for basic cyber exercises in a (hopefully) non-derived and familiar environment for students.

Intellectual Merits

- Realistic and scalable network sandbox used for cyber exercises
- Monitor Docker containers' file systems
- Automated adversaries making changes to systems and network in a competition setting
- Method to automatically 'score' player's activity on systems in a competition setting
- Graphical user interface for system and network management

Broader Impacts

- Increase education of next generation of cyber security professionals by providing them a live fire environment to practice in.
- Provide an open-source method for hosting engaging live fire cyber competitions.
- Competing in a live fire environment encourages a culture of innovation and problem solving.

Design Diagram #1

- Competitors and Admins connect over network connection to the 90s net project.
  - Admins connect over dedicated isolated network
  - Competitors connect over proxy connection into network

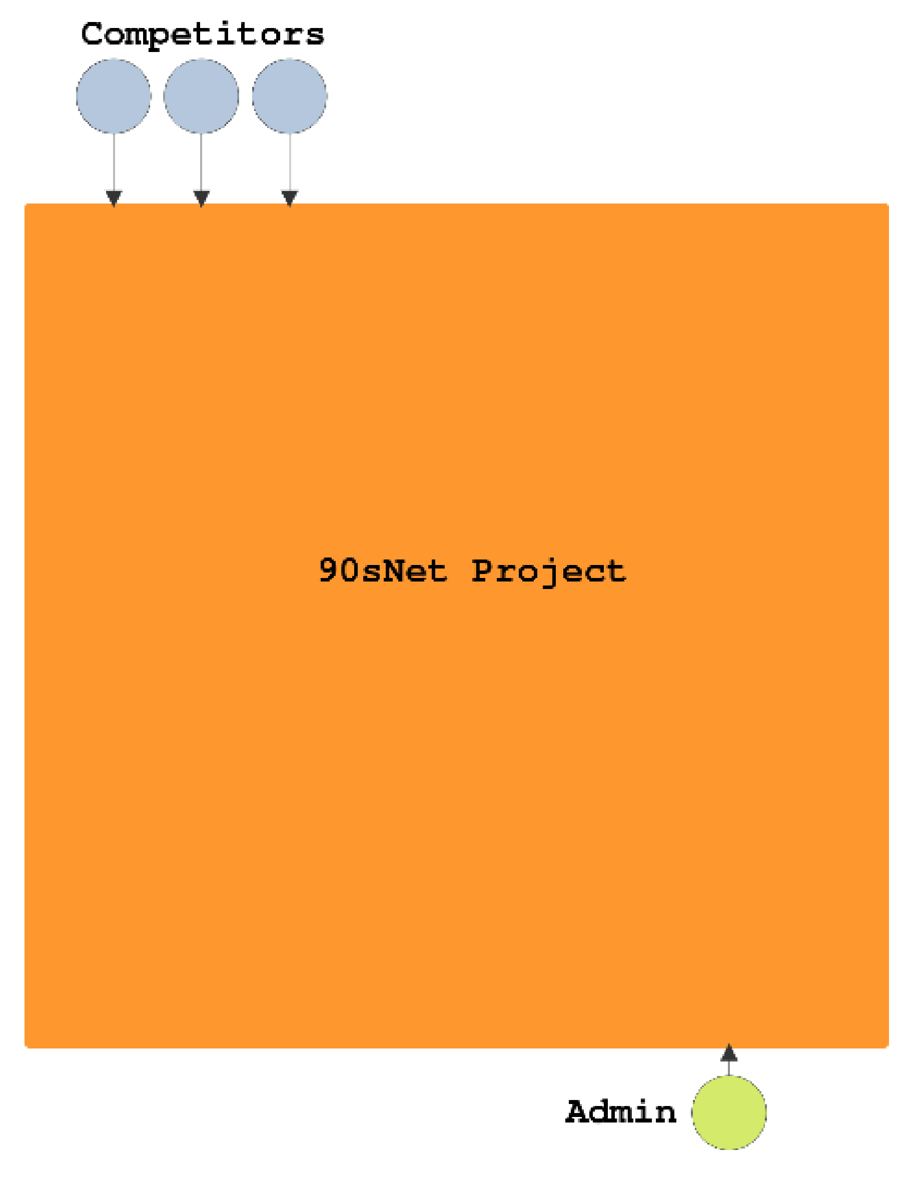
← Network connection

■ Computer System

● User

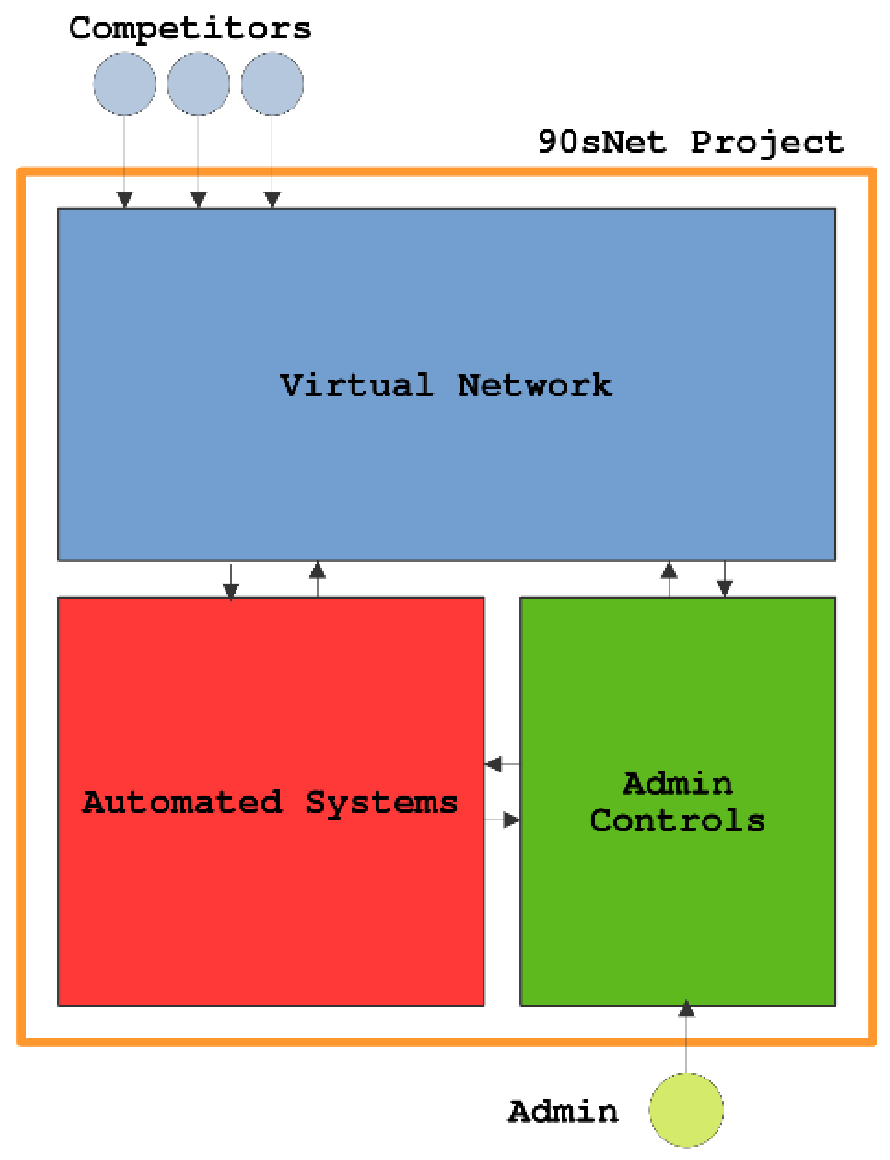
Name: 90sNet
Goal: Create a virtual network representing that of UC's main campus network from the 1990s to be used for cyber security exercises

Josh Hale & Chad Lape



Design Diagram #2

- Competitors connect to virtual network
- Admins use Admin controls over isolated network
- Automated Systems connected to admin controls via isolated network.
- Automated Systems inspect virtual network via existing in parallel to docker host



← Network connection

🟦 Computer System

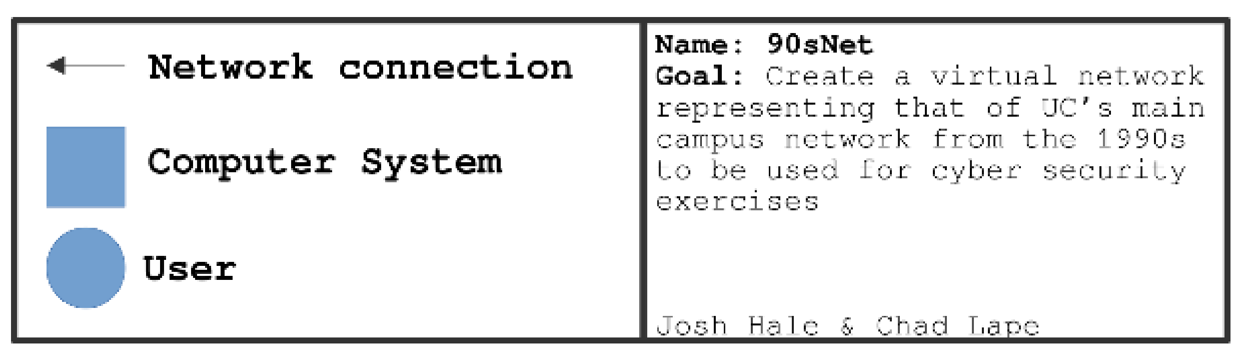
● User

Name: 90sNet
Goal: Create a virtual network representing that of UC's main campus network from the 1990s to be used for cyber security exercises

Josh Hale & Chad Lape

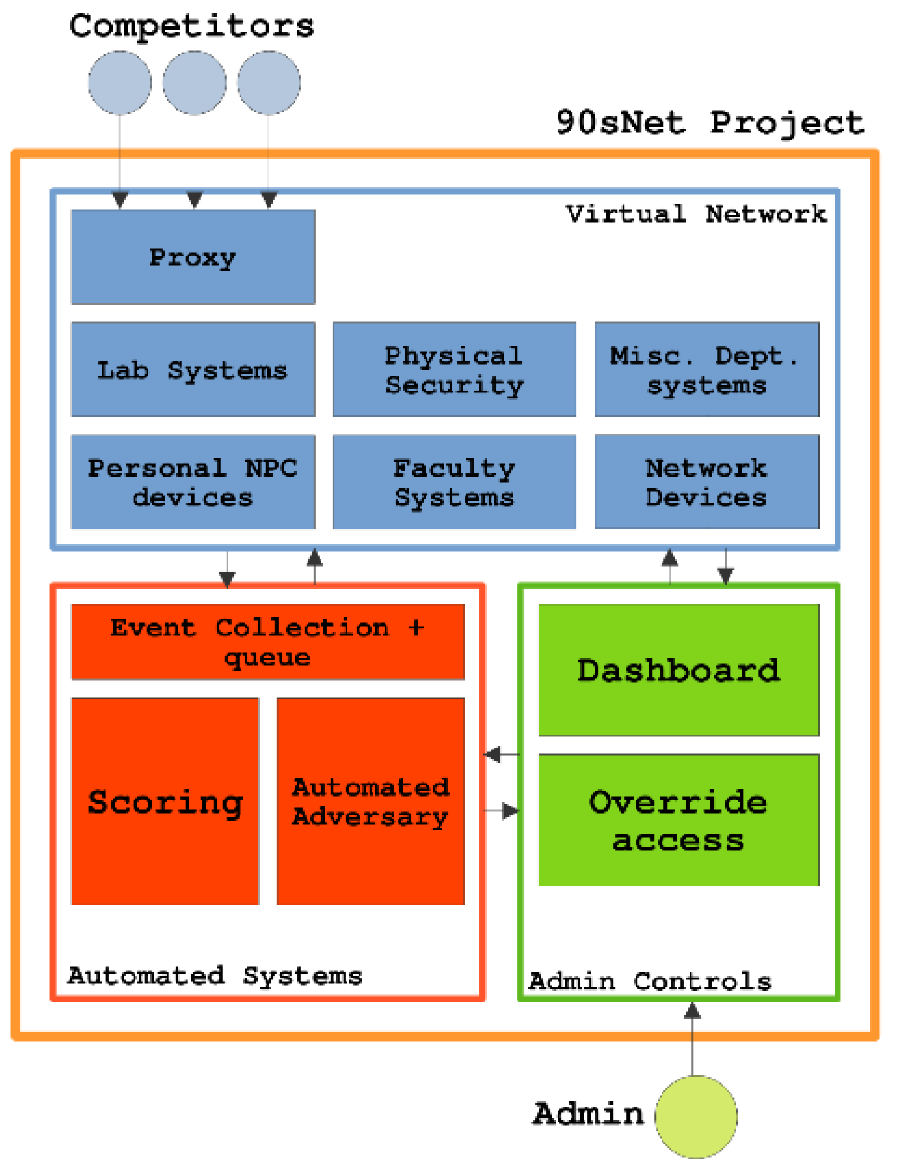
Design Diagram #3

- Virtualized systems on the network are interconnected via docker virutal network
- Event collection system watches virtual network traffic and systems for changes
- Scoring interprets events and adds scores to players accordingly
- Automated adversaries interact with competitors on the network



Name: 90sNet
Goal: Create a virtual network representing that of UC's main campus network from the 1990s to be used for cyber security exercises

Josh Hale & Chad Lape



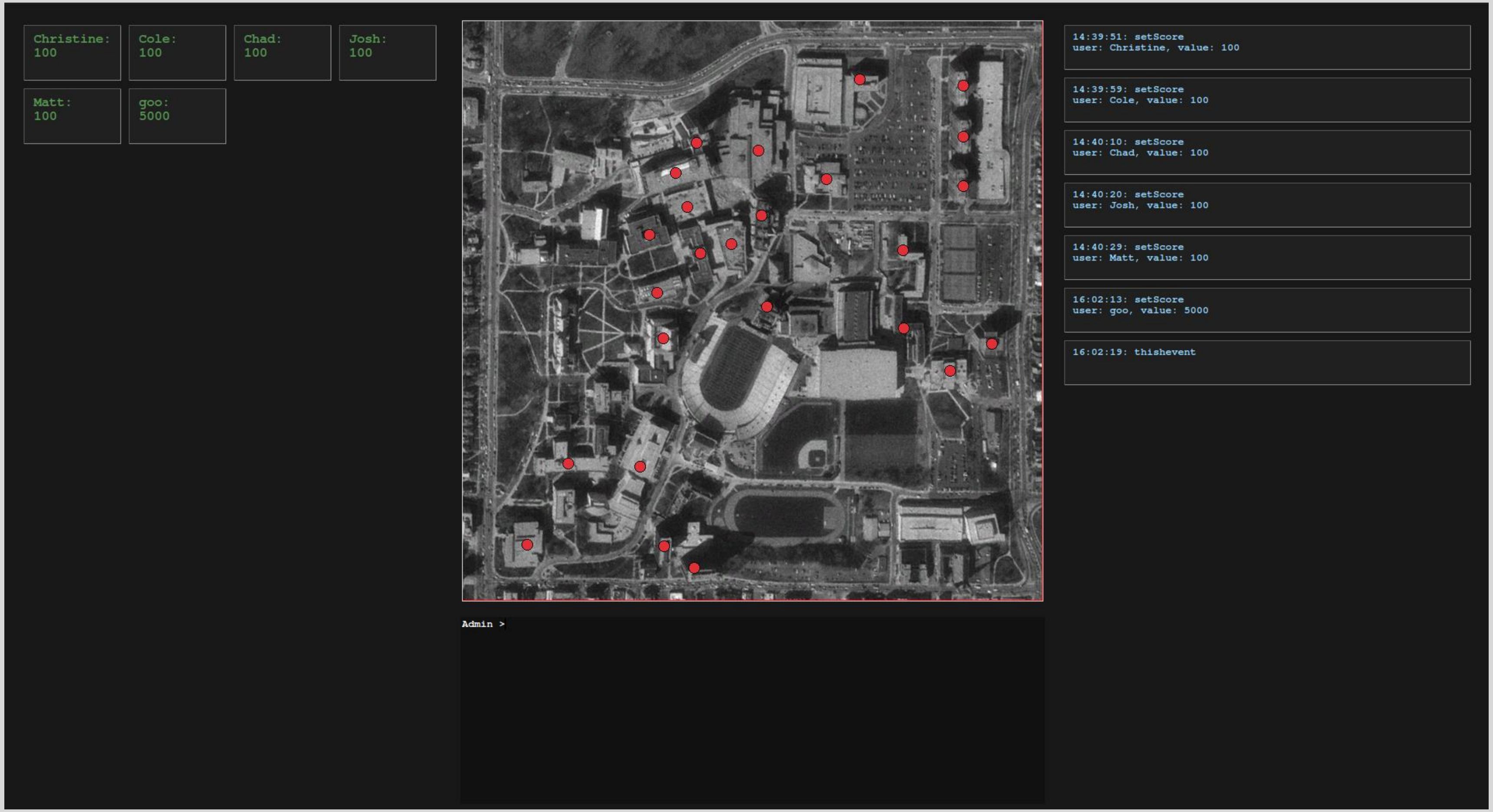
Technologies

- Network design and recreation
  - Recreating a campus network from the 90s using period accurate methods and historical knowledge
- Docker
  - Virtualizing entire network within a series of containers allowing for a light weight and scalable system
- Network automation
  - Automatically monitoring containers for states and conditions
  - If states or conditions are met, events are generated and inserted into a database
  - Database is then queried to see if compounded conditions are met to activate scores or automated adversaries
- GUI
  - Platform agnostic front end in python, allowing for ease of use and set up

Milestones

- 10/31: Lock Down Network Design
- 10/31: Small scale network demonstration
- 11/30: Network Interaction Demonstration
- 1/30: Network Device Template Design
- 2/28: Admin Infrastructure Implemented
- 2/28: Event Scanning Implemented
- 3/31: Event Handler Implemented
- 4/15: Full Network Demonstration

Results – Front End Development



```
Christine:    Cole:    Chad:    Josh:
100           100      100      100

Matt:         goo:
100           5000
```

```
Admin >
```

```
14:39:51: setScore
user: Christine, value: 100

14:39:59: setScore
user: Cole, value: 100

14:40:10: setScore
user: Chad, value: 100

14:40:20: setScore
user: Josh, value: 100

14:40:29: setScore
user: Matt, value: 100

16:02:13: setScore
user: goo, value: 5000

16:02:19: thishevent
```

Results – Front End Development

- The front end, or Admin Control system has been completed and automatically tested
- Features:
  - Scoreboard display (left)
  - Event stream (right) - shows all events on network detected by the monitor. Used for progress tracking and diagnosis
  - Terminal (bottom) - admin terminal access to control aspects of network and automated systems
  - Map (middle) - Live display showing geographic layout of systems with colored nodes. Nodes will flash colors based on event occurring at that location

Results – Back End Development

- Isolated network server
  - Monitors and generates events based on network and system activity
  - Database that stores events
  - Serves events to frontend over isolated network
- Docker Event Monitor
  - Have early success with checking for artifacts in a docker container which an attacker leaves
  - Multiprocessing to be able to handle the many small jobs required to check different machines
- Flask Data Server
  - Handling persistent data related to users, machines, and events
  - Can be queried by each of the components of 90sNet

Results – To Do

- Automated Adversaries
- Finalize network design and built in challenges
- Administrative work
  - Posters
  - Reports
  - Etc.

Challenges

- Finding the best container solution that would fit out projects needs (Chad Lape)
  - Started with Nomad
    - Decent solution with potentially to use in the cloud
    - Came to the issue of networking together the containers so a user could go between them as though on a normal network.
    - This became a challenge which had begun to expand the scope of the issue.
  - Docker Compose
    - Potential for cloud solutions in the future
    - Works for now on our systems
    - Have prior experience using it for deploying CTF Challenges
    - Can use container networks to simulate connections between these machines
- Finding resources on historical university systems (Josh Hale)
  - After much research, very little information was able to be found on historical UC systems.
    - Not many people employed from before 2000
    - No documentation (that we could access) from before 2000
  - Expanded research scope
    - General network design in the 90s
    - Physical UC systems and buildings
  - New Plan of attack: interpret general systems and methods from time period and apply to what we know about UC infrastructure at the time.