Team Cybersecurity Network Testbed System

Paige Brasfield
Logan Jacobs
Dylan Stellman
Fernando Bresso
Selim Harzallah

Sponsored by: Dr. Andrew Kalafut

Table of Contents

Overview

The Network Testbed is a comprehensive simulation platform designed to emulate diverse network conditions and cyberattack scenarios. Built upon Mininet, the system allows real-time adjustment of latency, jitter, packet loss, and bandwidth. It serves as a controlled environment for testing protocol performance in varying conditions, such as rural or bandwidth-limited networks, and supports the injection of malicious traffic (e.g., DoS).

Languages

We plan to use Python for this project. We can do light scripting for simulating various scenarios within a network topology. Mininet also has a Python API we can use to create and configure our topologies.

Libraries, SDKs, and APIs

Mininet

We are using Mininet for our machine environment. It is a Linux-based, network emulation framework that allows us to simulate realistic virtual network topologies. Mininet provides control over various link characteristics (bandwidth, jitter, latency, loss, etc.) within our virtual network for stress testing and simulating adverse conditions.

VMware

VMware lets us create virtual machines very quickly and easily on our machines, removing the need for dedicated machines to test our project. We each will use VMware to create a local Mininet environment to be able to test the work that each of us have been assigned.

Trafpy

Trafpy is a python package that allows us to generate and manage network traffic. We chose this over other libraries, like Scapy, because Trafpy focuses on creating realistic, repeatable network traffic to be tested over various topologies

Mininet Python API
One of the reasons we chose Mininet was because of its Python library. This allows us to programmatically create topologies. The Python API makes it much easier to automate and integrate with any Python-related workflows that we want to add to this project

MiniEdit
MiniEdit is the GUI version of Mininet. It allows us to drag-and-drop network components to create a virtual environment. Miniedit can export this topology to a python script for emulation

PyQt
PyQt is a GUI framework in Python based on the Qt framework in C++. It’s highly customizable and user friendly, which makes it very capable in integrating with any features we’d like to implement.

Repository Organization

Project Dashboard
We are using a project dashboard in github for our git workflow.

Backlog
This dashboard will contain a list of action items that are not yet ready to be started but are planned for.

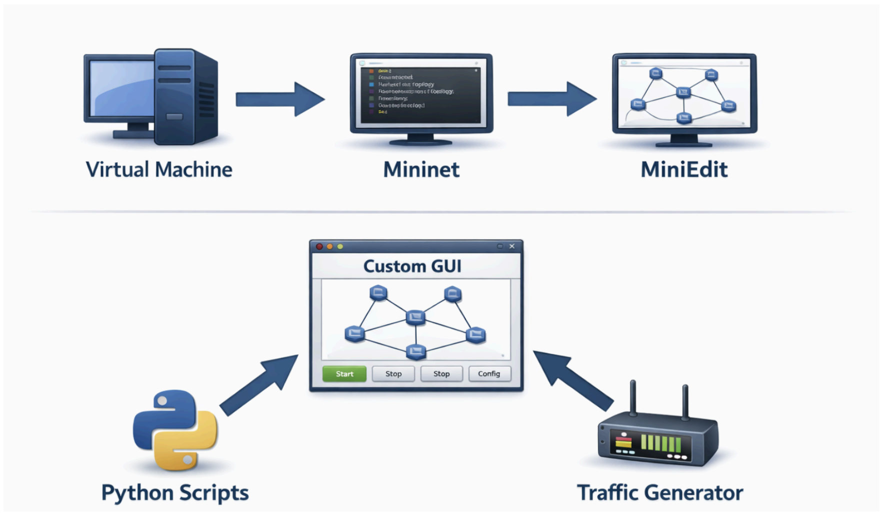
Ready
This dashboard will contain a list of action items that are ready to be worked on.

In Progress
This dashboard will contain action items that are currently being worked on as well as who is working on them. These action items will also include a due date to help manage workflow progress.

In Review
This dashboard will contain action items that are believed to be completed and are in the review process.

Done
This dashboard will contain all action items that have been completed.

System Organization



Users will require a Mininet environment and to pull the application from our repo. Once the user is in our application, they will have the ability to choose various topologies in which to test various link characteristics. On top of this, users can choose various attack methods to further stress the network. Users will be able to see telemetry on the system in the form of a .csv file once the simulation is complete for further analysis.

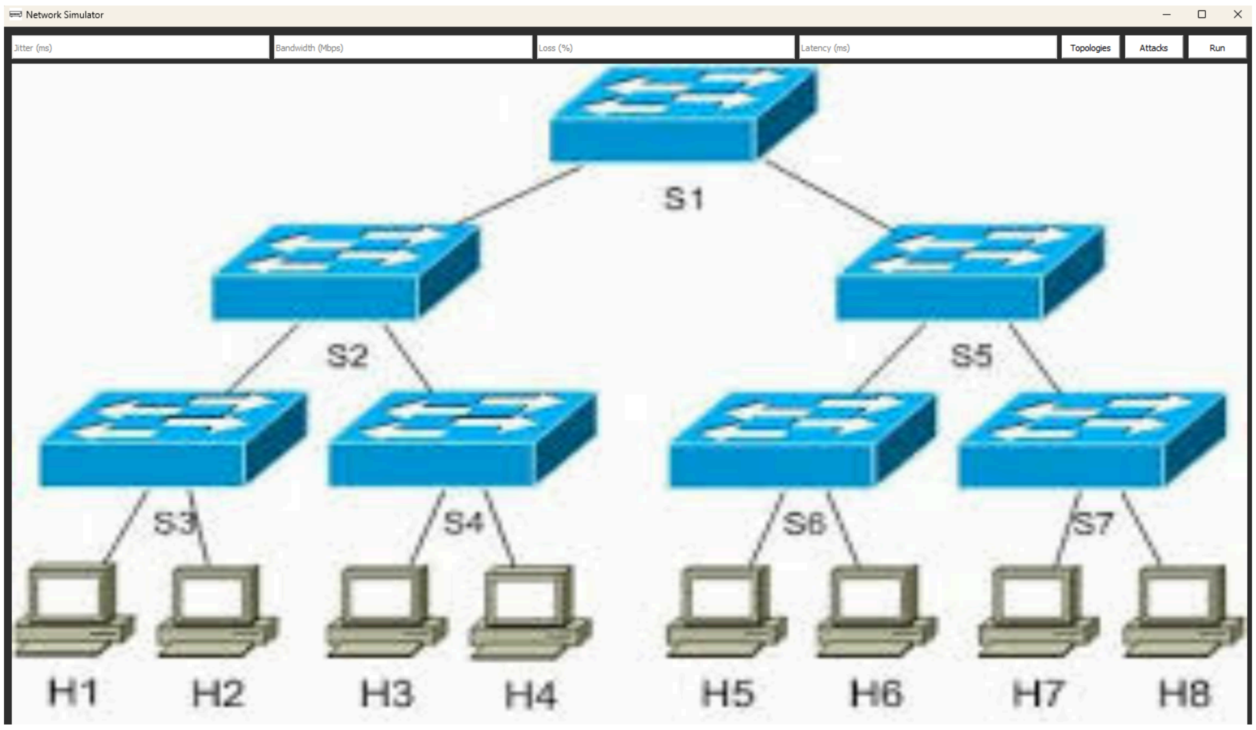
Development

Our network testbed will be created via Mininet, MiniEdit, and some custom python scripts to simulate various network stress tests and attacks. Each of us will run Mininet on VMware, working on our own parts while pulling from our Github repo to keep up to date with each other's work. MiniEdit will be used to create scripts of our topologies, as MiniEdit is quite limited in its simulation capabilities. We will be using PyQt for the UI, as it provides good compatibility and features to implement all the features we'd like for this project.

Non Trivial Requirements

Create a UI
Create network attacks
Create topologies that reflect real-world configuration

Mock Ups



Work Policies

Meetings

We will have weekly meetings on Wednesday at 7pm via zoom to discuss our agenda and goals for the week. If there is no class on Tuesday or Thursdays, we meet during class time. We have a weekly meeting on Fridays at 11am with Dr. Andrew Kalafut via zoom for weekly updates and/or questions.

Communication
We will communicate via group chat messages other than our in person meetings. Everyone is responsible for their designated areas and will be held accountable.

Division of Labor

Paige - Creating network attacks and working on User Interface abilities.

Logan - Researching topologies to be accurately created within MiniEdit

Dylan - Reporting network traffic and working on mimicking malicious attacks. Also, will work on setting rules for preventing such attacks.

Fernando - Creating network attacks and connecting the front end to the backend

Selim - Creating network test cases to ensure topologies reflect real-world results for accuracy

Testing

After creating each LAN network we will test packet transfer through it. We will then test packet transfer with each change in bandwidth and check for jitter and packet loss. After creating the different LAN networks we will test with the different attacks we created on each LAN network.

Deliverance and Milestones

Sprint 1
- Research what goes into each different networks
- Create different LAN networks
    - Rural
    - Urban
    - Suburban
- Create WAN network
- Have bandwidth limited network created

Sprint 2

- Incorporate MiniEdit ui
- Adjust MiniEdit ui based on current performance
    - Allow for bunched endpoint changes rather than individual
    - Latency
    - Packet loss
    - Bandwidth
    - Jitter

Sprint 3

- Incorporate a traffic generator into our MiniEdit ui
- Create network attacks to simulate
    - Ddos
    - Arp spoofing
    - Eavesdropping

Sprint 4

- Create a UI so users can adjust networks
    - Latency
    - Bandwidth
- Create button so user can simulate each type of attack