Topics we are interested in;

- DDoS Mitigator (1st choice)
- Canaries (2nd Choice)

DDoS Mitigation:

Summary

Denial of Service (DoS) Attacks happen when a victim (host or network) is flooded by an abundant amount of malicious traffic until the victim cannot process any legitimate traffic or just crashes.

This project would include setting up a testing environment with a target and attacking machines. The target machine would include logging and metrics as a means to measure success of mitigation techniques. The target will have various mitigation techniques necessary for a complete analysis. Attackers will send malicious requests to the target machine to perform the DDoS attack. Attacker machines will support various forms of DDoS attacks.

Deliverables:

1. Set up Target Machine with sufficient logging of incoming requests and resource loads
2. Set up a Single Attacking machine to perform DoS attacks. Support 2-3 different DoS attack types
3. Extend Single Attacking Machine into multiple replicated machines to perform DDoS attacks. (Better set up for IP based mitigation techniques)
4. Include Consumer machines or valid requests to measure success rates of valid requests during attacks.
5. Include application layer (TCP buffers) and Network layer mitigation techniques (IP Filtering, CDNS…)
6. Enhance logging and metric tracking on target machine(s). Add attacker logging if necessary. Centralize below.
7. Either on Target machine or separate machine host dashboard monitoring to see logs and metrics of attacks and mitigations. Optionally allow controlling of the environment from the same dashboard.

Testing:

We will need either a cloud or physical setup with multiple machines in order to replicate real world attacks.

Testing and analysis will be dependent on the target machine logging service. Initially this will be small in scale (Somewhat manual analysis of logs). A stretch goal would be to make a dashboard to easily visualize logs and loads between attacks.

Barriers:

The main barriers will be setting up the initial environment during deliverables 1-3. This will be heavy on network and host configuration. After setting up this environment it might also be difficult to adapt quickly/programmatically to support multiple setups.

Canaries as Early-Warning Security Mechanisms:

Description:

Canaries are security mechanisms used to provide early detection of cyberattacks. They function as intentionally placed decoy assets—such as files, credentials, or network services—that have no legitimate purpose and should never be accessed during normal operations. Because of this, any interaction with a canary is a strong indicator of unauthorized or malicious activity within a system. This project examines how canary technologies are designed and deployed, the types of attacks they are capable of detecting, and how effective they are in improving an organization's overall security posture through early warning and threat awareness.

List of Potential Deliverables:

- Canary Implementation eg. Fake files, credentials, or services and Simple alerting when accessed.
- Testing Environment; controlled network or VM setup Simulated attack activity
- Collected Data such as logs of canary triggers, detection timing and frequency
- Documentation such as explanation of each canary, setup instructions, and evaluation/analysis of effectiveness

How are we going to be testing this project:

The project will be tested in a controlled environment such as a virtual machine or isolated file system. Canary assets (fake files, credentials, or services) will be deployed and monitored for access. Simulated attack behaviors—such as searching for sensitive files or attempting to use fake credentials—will be performed. Canary triggers, alert timing, and access logs will be recorded and analyzed to evaluate detection accuracy, response speed, and overall effectiveness.

Potential barriers to success:

- **False positives:** Legitimate users or system processes may accidentally trigger a canary, reducing alert reliability.

- **Limited realism:** If canaries are not convincing, an attacker may ignore or avoid them.

- **Scope limitations:** A small or simplified test environment may not fully represent real-world networks.

- **Detection-only capability:** Canaries detect attacks but do not prevent or stop them.

- **Configuration errors:** Improper setup or monitoring could result in missed alerts.

- **Attacker awareness:** Skilled attackers who recognize canaries may deliberately avoid them.