## CyHelics

### sdmay24-28

**2.1 Task Decomposition**

 Our project will have a structured sequence of events. Beginning with the setup of virtual machines within a dockerized environment. We then establish connections between Helics, Pandapower, and OpenDSS using the virtual machines we have already created. Followed by crafting an electric grid diagram for visualization. Going deeper into electrical grid design we will create smaller sections of the transmission and distribution grid. Once they are set up we will then simulate these with an emphasis on how they interact with each other. The security aspects of our project are addressed through a Kali purple box in company with an implementation of Security Onion for defense. Finally we will be implementing a front end website to create an enhanced user experience.

**Task Dependency Graph:**



**2.2 Project Management/Tracking Procedures**

 Our team decided upon an Agile Project Management approach. This gives us the flexibility to accommodate multiple adjustment periods into our project. The adjustment periods ensure a smooth transition for our team to become acclimated with the proposed software. Agile development allows for small incremental parts of our project to be developed and tested. As goals and tasks become more advanced the agile management structure allows us to break these tasks down into smaller attainable goals. To streamline our Agile management style we will be using Gitlab for version control and centralized project management.

**2.3 Milestones, Metrics, and Evaluation Criteria**

 Our team has identified milestones that we deem fit for evaluation criteria. First, in the preliminary grid phase, we simulate pre-existing transmission, distribution, and load models, ensuring they simulate seamlessly through the Helics software. Moving forward into the grid design phase, we dive into designing multiple transmission models, ensuring proper power flow, and conducting analyses encompassing power flow, fault simulations, harmonics, and unbalanced power flow. We also will make distribution models, capable of accommodating dynamic loads, employing both linear and nonlinear models. A dynamic load profile is created, emulating real-world scenarios, including electric vehicle usage and residential neighborhood power consumption, with fluctuations based on actual data. In the simulation setup phase, each simulation is dockerized individually. The attack modules phase entails setting up a Kali box for executing targeted attacks against the simulated grid and designing a user-friendly frontend interface to streamline attack execution. Lastly, in the defense phase, we deploy Security Onion to detect and counteract grid attacks, implementing automated defense mechanisms, centralizing logs in a database, and establishing rules to swiftly identify and address potential issues. These milestones collectively shape our project's progression, ensuring its success and security at each stage.

**2.4 Project Timeline/Schedule**

|  | **1/22-29** | **2/5-12** | **2/19-26** | **3/4-11** | **3/18-25** | **4/1-8** | **4/15-22** | **4/29-5/6** | **5/13-20** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Analyze pre existing transmission and distribution models** | Deliver by 1/29 |  |  |  |  |  |  |  |  |
| **Analyze the power flow and design of models** |  | Deliver by 2/12 |  |  |  |  |  |  |  |
| **Co-Simulation between transmission and distribution models** |  | Deliver by 2/12 |  |  |  |  |  |  |  |
| **Working simulation** |  |  |  | Deliver by 3/11 |  |  |  |  |  |
| **Set up VM and Dockerized Environments** | Deliver by 1/29 |  |  |  |  |  |  |  |  |
| **Create and run attacks** |  |  |  |  |  |  | Deliver by 4/22 |  |  |
| **Frontend for attack modules** |  |  |  |  |  |  |  |  | Deliver by 5/20 |
| **Integrating security onion** |  |  |  |  |  | Deliver by 4/8 |  |  |  |
| **Using security onion to detect the attacks** |  |  |  |  |  |  |  |  | Deliver by 5/20 |

**2.5 Risks and Risk Management/Mitigation**

As our entire project will be done within a dockerized VM environment, there will not be significant risks associated with this project.

Risks:

* Losing our VMs and our progress - 3
	+ Mitigation: Create regular backups and VM snapshots
* Using too many resources, causing the department to be mad at us - 0.3
	+ Mitigation: Being aware of our resources and making sure nothing runs out of hand

**2.6 Personnel Effort Requirements**

| Task | Justin Hrs | Kaya Hrs | Matt Hrs | Tommy Hrs | Tyler Hrs | Zach Hrs | Total Hrs |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Set up and connect Helics with pandapower and OpenDSS | 5 | 5 | 2 | 2 | 5 | 5 | 24 |
| Analyze the power flow and design of models | 0 | 0 | 5 | 5 | 0 | 0 | 10 |
| Create smaller sections of the transmission and distribution grid | 5 | 5 | 30 | 30 | 5 | 5 | 80 |
| Co-Simulation between transmission and distribution models | 2 | 2 | 2 | 2 | 2 | 2 | 12 |
| Create a working simulation | 20 | 20 | 10 | 10 | 20 | 20 | 100 |
| Create a Kali/red team box that can perform attacks against the power grid | 2 | 2 | 0 | 0 | 2 | 2 | 8 |
| Set up security onion to defend against the attacks against the power grid | 0 | 0 | 0 | 0 | 20 | 0 | 20 |
| Frontend for attack modules | 5 | 5 | 0 | 0 | 5 | 5 | 20 |

**2.7 Other Resource Requirements**

 Our team has identified a few other resource requirements for our project to function effectively. At a base we will need access to the university's High-Performance Compute Cluster serving as our backbone for testing complex powergrid designs. Additionally, our team will need a multitude of virtual machines. We have identified a few types of machines we will need. The first is a Kali Linux purple pentesting box for heading the cyber attack section of our project. We will need a few Ubuntu and Windows 10 machines for hosting HELICS, Panda Power, & Open DSS. Finally we will need a larger virtual machine to host Security Onion for network monitoring.