Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES)

Tasks 3 and 4 Report

Submitted by:
Lawrence Berkeley National Laboratory (LBNL)

Technical Point of Contact:
Dr. Daniel Arnold (dbarnold@lbl.gov)

Date of submission:
July, 27, 2023

Submitted to:
DOE CESER Risk Management and Tools (RMT) program, care of the National Energy Technology Laboratory (NETL)

NETL Project Manager:
Joel Lindsay

Project Website:
https://secpriv.lbl.gov/project/ceds-spades/

Table of Contents

List of Acronyms

- BESS - Battery Energy Storage System
- CEDS - Cybersecurity for Energy Delivery Systems
- CESER - Cybersecurity Energy Security and Emergency Response
- CIGAR - Cybersecurity via Inverter Grid Automatic Reconfiguration
- DER - Distributed Energy Resource
- DOE - Department of Energy
- HIL - Hardware In the Loop
- IEEE - Institute of Electrical and Electronics Engineers
- KPI - Key Performance Indicators
- MCTS - Monte Carlo Tree Search
- NRECA - National Rural Electric Cooperative Association
- OMF - Open Modeling Framework
- PV - Photovoltaic
- SPADES - Supervisory Parameter Adjustment for Distribution Energy Storage
- RMT - Risk Management and Tools

Introduction

Increasing adoption of Distributed Energy Resources (DER), specifically rooftop photovoltaic (PV) generation systems and Battery Energy Storage Systems (BESS), is challenging many conventionally-held models and practices regarding the operation of the electric power system. While the presence of PV and BESS devices gives individuals and communities the ability to self-generate a portion of their load and participate in providing services to the grid, they also make proper management of the power system more difficult as many DER are not utility-owned/operated. With the recent changes in regulations allowing DER to gain entry into wholesale markets [1], these challenges will undoubtedly increase as more DER asset owners and aggregators seek to take advantage of new revenue streams. Given the superlinear nature of expected growth in the storage market specifically (see Figure 1), it is likely that addressing these challenges will become increasingly important in the near term.

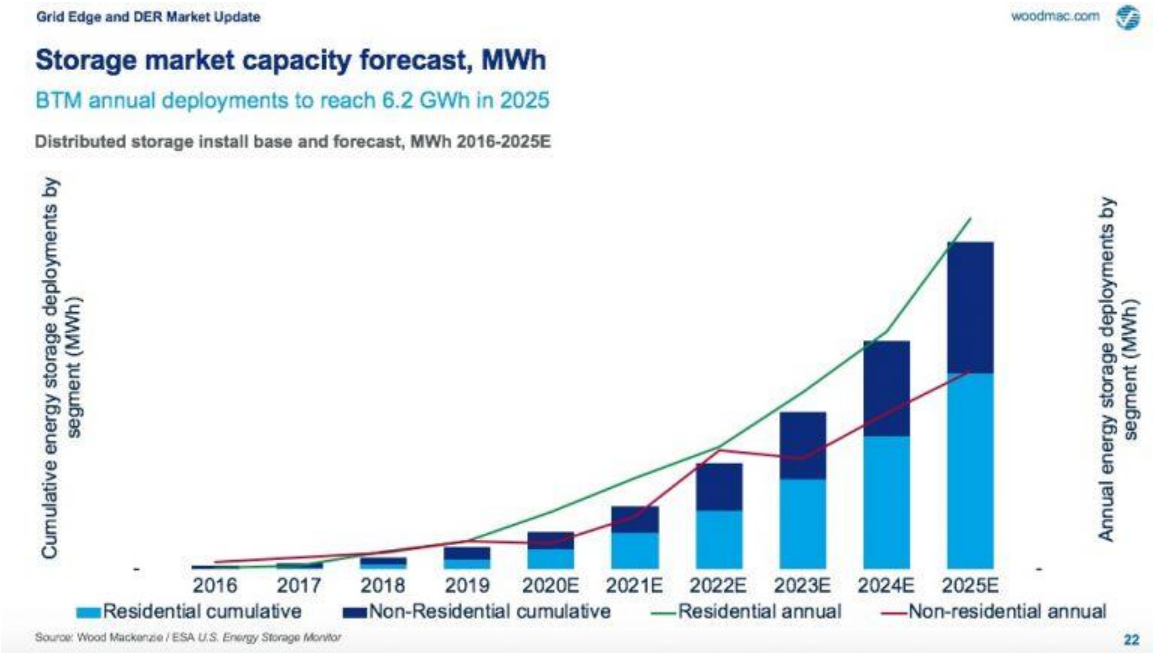

Figure 1 - Project growth of electric storage market[1]

Technical specifications governing the testing, interconnection, and behavior of DER are necessary to ensure the safe and reliable operation of the power system as more renewable generation sources are brought online. Of particular interest is the functionality included in the relevant standards that outlines DER behavior in response to changes in local grid conditions sensed at the point of interconnection [2-4]. These autonomous control functions essentially enlist DER to help correct undesirable frequency, voltages, and power factors and (in theory) provide a mechanism to allow DER to mitigate power quality issues that they themselves can introduce in grids with high penetrations of renewables.

[1] Source: United States Distributed Energy Resources Outlook: DER Installations and Forecasts, 2016-2025E (Wood Mackenzie)

While the standardization of DER control functionality through IEEE 1547-2018 provides mechanisms for DER to regulate their power injections in response to local grid conditions, the remote update capability of DER constitutes a significant expansion of the cyber attack surface [2]. Via remotely updating the settings of standardized control functions in aggregations of DER, malicious entities could substantially alter grid conditions with relatively small adjustments to parameters of individual devices. Small changes to individual DER may seem innocuous at the device level, making attacks difficult to discern. However, if enough DER with standardized control behavior adjust their control parameters and/or setpoints, even just a little, they could destabilize the feedback interconnection between DER aggregations and the grid [3] or create deleterious power quality issues [4]. Both outcomes could cause damage to sensitive devices or device disconnection from the power grid, further disrupting the provisioning of electricity. Storage devices, in particular, further extend the cyber attack surface compared to photovoltaic systems as they not only adopt the same control functionality outlined in IEEE 1547, but are capable of acting as a load or generation source (somewhat) independently of available sunshine.

On the other hand, the same standardized control functionality and remote update capabilities of DER devices can be leveraged to mitigate the effect of cyber attacks on both DER and the electric grid in real time. Previous work on the CEDS/RMT project CIGAR [5] demonstrated the use of advanced control approaches to control *non-compromised* (i.e., devices which have not had their settings adjusted as part of a cyber attack) DER to ameliorate the effects of attacks on a portion of DER in a given grid.

While CIGAR focused exclusively on photovoltaic systems, the Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES) extends the types of DER considered to include BESS devices. As DER systems do not exist in isolation (a grid will most likely not solely consist of rooftop photovoltaic systems, for example), characterizing the complex interaction across a heterogeneous device set is increasingly important to properly understand the effects of cyber attacks on DER populations as well as determine mitigation strategies. Indeed, as the attack surface becomes increasingly complex those seeking to develop strategies to defend the system are simultaneously given more opportunities to implement effective countermeasures.

This document outlines the progress made and results during the third and final year of the SPADES project. SPADES sought to extend work conducted in project CIGAR developing reinforcement learning and adaptive control approaches to update the settings of photovoltaic generation systems to mitigate the effect of cyber attacks on PV systems in real time. SPADES extends CIGAR via the addition of energy storage devices (e.g., electric batteries) in both the threat models and the remediation strategies. While the first two years of SPADES focused on power system modeling and algorithm design, year 3 focused on integration of the developed mitigation algorithms into the National Rural Electric Cooperative Association (NRECA) Open Modeling Framework (OMF) and red team testing of the SPADES algorithm (conducted by Siemens Corporate Technologies). The following sections detail work performed during the period between Jan. 1, 2022 - July 31, 2023 where the SPADES team focused on completing Tasks 3 and 4 of the Field Work Proposal.

Task 3 focused on Hardware-In-the-Loop (HIL) experiments at the LBNL FlexGRID facility and validation of the performance of the developed control algorithms by the Red Team. The goal of the Red Team was to disrupt the operation of battery storage devices and other DER in simulation and to degrade the performance of the reinforcement learning controller in preventing a cyberattack that disrupts DER ability to provide grid services. As part of this task, Siemens developed an attack optimization engine to determine cyber attacks which would be the most effective for a given distribution network topology and DER deployment.

Task 4 focused on integrating the defensive reinforcement learning-based agent into the NRECA Open Modeling Framework (OMF) simulation tool. This capability will allow utility users to upload their network models, choose a desired ESS mode of operation, and conduct simulations to evaluate the effectiveness of the defensive agent to defend against user-selected cyberattacks.

Both Tasks 3 and 4 were completed prior to Dec. 31, 2022, but, due to remaining subcontractor budget, the project was granted a no-cost extension until July 31, 2023. During this period, the following activities were undertaken:

- NRECA extended the Open Modeling Framework interface to visualize key performance indicators developed by the red team. If enabled by the user, this feature will allow the visualization of metrics used by the red team optimization module to gauge the effectiveness of cyber attacks on DER.

- Siemens experimented with parallelization of the Monte Carlo Tree Search (MCTS) attack optimization engine using LBNL's Lawrencium supercluster. While the no-cost extension period ended before the parallelization of the MCTS optimization engine was completed, this effort laid the groundwork for enhanced efficiency and performance of the red team optimization engine which could be completed in later efforts by the Siemens team.

- LBNL developed a set of python notebook tutorials illustrating the use of the PyCIGAR software developed in SPADES. The tutorials showcase the use of the algorithms in mitigating attacks on use cases identified in Subtask 1.4 and illustrate how to adjust the cyber attack scenario, reinforcement learning training parameters, and the electric grid topology, and will serve as training materials for parties wishing to gain a deeper understanding of the software developed in SPADES. These tutorials will be made available on github following the public release of the PyCIGAR software developed in this project (scheduled for August 2023).

The following sections detail work undertaken in Tasks 3 and 4 as well as the no cost extension period. First, a discussion of work undertaken in Task 3 (Red Team experiments) is presented, followed by a discussion of work undertaken in Task 4 (OMF Integration). These discussions will include all work occurring during year 3 of the project, as well as the no cost extension period.

Task 3 - Hardware in the Loop (HIL) Experiments and Red Team Activities

HIL Experiments and Simulations

Use Case Overview

Hardware in the loop experiments designed to showcase the effectiveness of the reinforcement learning control algorithms developed in the SPADES project (in Task 2) and utilizing real time data obtained from LBNL's FlexGrid facility were conducted in the fall of 2022. These experiments were modeled on a real use case provided by NRECA. A description of the use case is now provided.

Okracoke Island, North Carolina, is a popular tourist destination which is connected to the mainland via a single line. The island has on-site generation options including a diesel generator, battery storage system, and some photovoltaic systems. In periods where the connection to the mainland grid is severed, the generation sources on the island are responsible for meeting electricity demand. Additional details of the electricity generation on the island are as follows.

- Peak demand: 5 MVA (summer); 1 MVA (winter)
- PV Penetration: 1% (up to 15% used for demonstration purposes)
    - 15 kW PV on top of the diesel generator housing at substation
    - Very little residential solar
- Diesel generation of 3MW.
- Battery Energy Storage: 0.5 MW / 1 MWh
    - Serves the load through the time it takes to start diesel generator (about 10 minutes)

Figure 2 - Properties on Okracoke Island

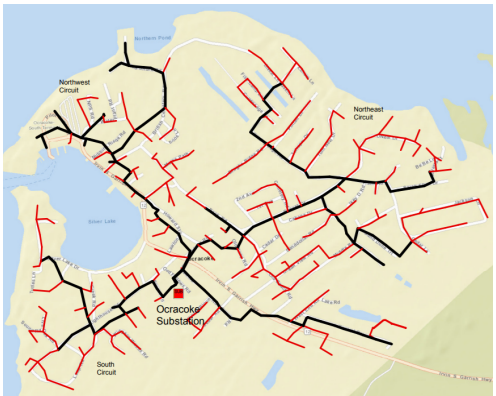

Figure 3 - Okracoke Substation circuit map.

In this scenario, a microgrid consisting of utility-scale PV, battery storage system and diesel generation works alongside residential PV and battery storage system to support a specific section of the full circuit. This scenario models an outage during the early stages of hurricane Irene, which made landfall on North Carolina's Outer Banks around 8am on August 27, 2011. The experiment takes place between 15:00 and 19:00 on August 26, 2011, when strong winds were beginning to fell trees. At 16:00, a vulnerable section of line is damaged and the downline system goes into backup/islanded mode. The battery is used to serve the load while the diesel generator starts up.

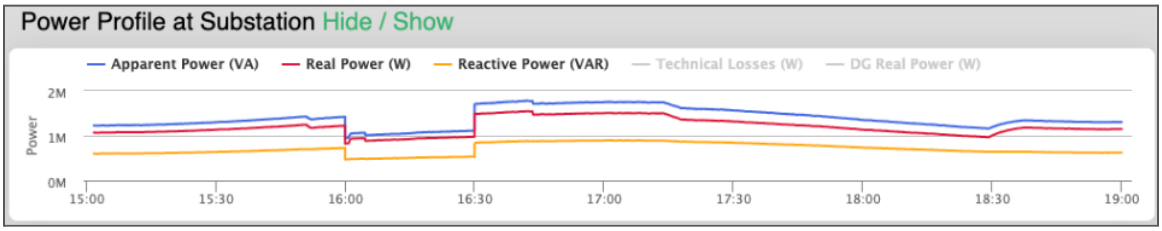

Figure 4 - Substation power for the Okracoke Island use case with hybrid PV and BESS. Substation power is lost at 16:00 and the battery discharges until the diesel generator can take load at 16:30. The battery then supplements as needed to maintain service.

HIL Description

Hardware in the loop experiments occurred during the fall of 2022 and utilized components from LBNL's FlexGRID facility [6]. FlexGRID enables real-time analysis of demand, renewables, inverters, and storage in a hardware and software co-simulation environment. The facility allows researchers to develop technologies and controls that span both the supply and demand sides of the power system. The facility is highly instrumented and submetered and features an array of controllable/programmable devices including:

- Three 14.6kW (total) photovoltaic systems (rooftop mounted)
- Three sets of Tesla PowerWall batteries (19kWh total)
- Three 7.6kVA Solar Edge inverters
- Micro synchrophasor measurement units
- Ametek MX-30 regenerative power supply
- Opal-RT grid simulator
- Bidirectional CHAdeMO EV charger

Figure 5 - Rooftop photovoltaic systems located at LBNL's FlexGRID facility.

Figure 6 - Tesla PowerWall battery storage devices located at LBNL's FlexGRID facility.

Experiment Description and Results

In the scenario considered in the experiments, distributed storage assets are being utilized to support the load in the lost section of Okracoke island, but a cyber attack has deactivated all of the distributed battery systems on a specific phase of the island, creating a large voltage imbalance. Voltage imbalances can cause outsized current imbalances in three phase motors which can, in turn, cause excessive heating and winding burnout. In these conditions, motors should be derated or disconnected from the system. The algorithms designed by the SPADES team will seek to minimize the level of voltage imbalance in the network, thereby preventing damage to/disconnection of motors and other sensitive devices.

Hardware in the loop experiments involved the use of the rooftop photovoltaic systems and the Tesla PowerWall batteries from FlexGRID. Solar production data obtained from real time measurements of the rooftop photovoltaic systems were used as inputs in an OpenDSS simulation of the IEEE 37 node test feeder, which is a suitable representation of the distribution system circuit on Okracoke. As the goal of the SPADES project is to develop algorithms to mitigate attacks *without compromising the provisioning of existing services being provided by battery storage systems*, the SPADES team designed an algorithm that uses excess capacity of battery storage devices to inject/consume *reactive* power to mitigate large voltage imbalances. A detailed overview of the algorithm design and simulation results for this attack vector can be found in associated publications [7,8].

Real time measurements of Tesla PowerWall active power injections were used to determine excess battery inverter capacity, which was then used as an upper limit on the amount of reactive power which could be sourced/sunk by each battery. This reactive power upper limit was incorporated into simulated battery systems in OpenDSS. The SPADES team chose to simulate battery reactive power injection capabilities rather than command reactive power injections in the Tesla PowerWall units as the PowerWalls included at FlexGRID do not support this feature. However, this capability is expected to be available in future battery systems which are compliant with IEEE 1547. With this reactive power upper limit obtained from the PowerWalls now input into the OpenDSS simulation, the algorithms developed in Task 2 of the SPADES project (having been trained in an offline environment) were used to adjust the reactive power injection of simulated battery storage devices to mitigate a cyber attack designed to create large voltage imbalances.

The results of a single experiment are shown in Figures 7 and 8, which depict the three phase voltage profile of a node which experienced the worst case voltage imbalance in this experiment. The left hand subplots show the voltage profile and the right hand subplot shows the per unit reactive power injection for the storage device at this node (reactive power injected per phase). Figure 7 depicts the voltage profile and reactive power injection without supervisory control determining new reactive power setpoints (i.e., without the SPADES algorithm). Figure 8 depicts the three phase voltage and reactive power injection with the SPADES algorithm determining new reactive power setpoints. The reduction of voltage imbalance between Figs. 7 and 8 is quite clear, as is the difference in the amount of reactive power injected. Interestingly, in observing the reactive power injection in Figure 8, which enabled the reduction in voltage

imbalance, the reactive power is consumed on one phase and injected on another. This non-uniform response highlights the benefit of using reinforcement learning and neural networks as these approaches can produce more complex and nonlinear control actions to achieve their objectives.

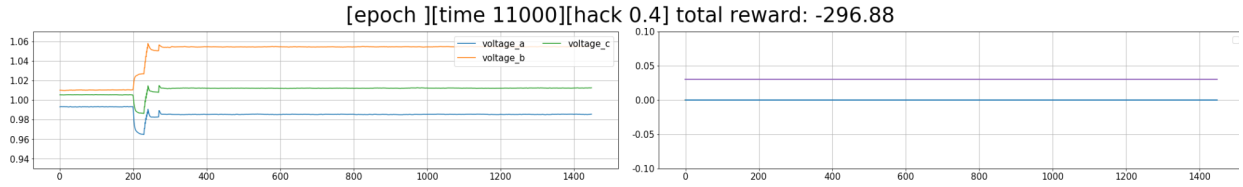

Figure 7 - Plots showing the three phase voltages (left-hand subplot) and the battery reactive power injections (right-hand subplot) without additional reactive power injection provided by the SPADES algorithms. The voltage imbalance is clearly seen in the left hand subplot.

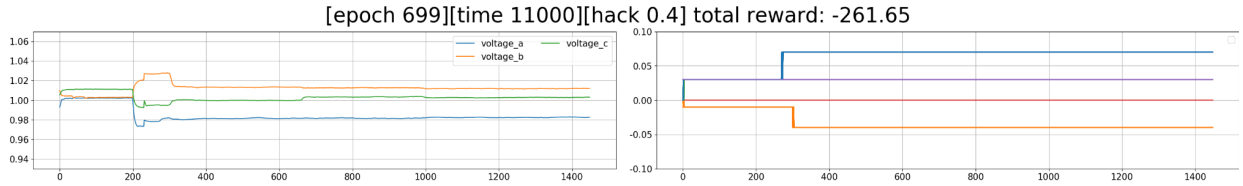

Figure 8 - Plots showing the three phase voltages (left-hand subplot) and the battery reactive power injections (right-hand subplot) with additional reactive power injection provided by the SPADES algorithms. The level of reduction of voltage imbalance is clearly visible compared to Figure 7.

Another experiment was conducted where distributed batteries are used to perform a peak shave for Okrakoke. Results of this experiment are shown in Figs. 9-13. Similar to the previous experiment, in this case an attacker shuts down all of the distributed storage assets contributing to the peak shaving effort on a certain phase in the network, causing a large voltage imbalance. The following figures demonstrate the effectiveness of the SPADES algorithm in controlling reactive power injections using excess inverter capacities of storage devices to mitigate the effect of the voltage imbalance attack.

Figures 9 and 10 show the substation active power profile along with the upper limit of active power which is to be provided by the substation with and without SPADES control. Figures 11 and 12 show the average and worst case voltage imbalance with and without SPADES control. Figure 12 shows the reactive power output of different batteries during the experiment. Only the battery reactive power output with SPADES control is shown, as otherwise these values are always 0.

As previously discussed, the SPADES controllers are designed to defeat cyber attacks without compromising the level of service being provided by the battery storage systems. As is shown in Figs. 9 and 10, the peak shave provided by the batteries is not degraded through the use of reactive power injections from the SPADES algorithm (i.e. Figs. 9 and 10 are virtually identical).

The level of voltage imbalance reduction is easily visible in comparing the worst case and average imbalance levels seen in Figs. 11-12, where SPADES control reduced voltage imbalances by approximately a percentage point (both worst case and average imbalance). The reactive power injections from individual batteries is shown in Figure 13, which shows a negative injection for some devices, indicating that these batteries are consuming reactive power.

In both experiments, SPADES algorithms were able to ameliorate the cyber attack designed to create voltage imbalances without compromising the service being provided to the grid by the battery storage devices.

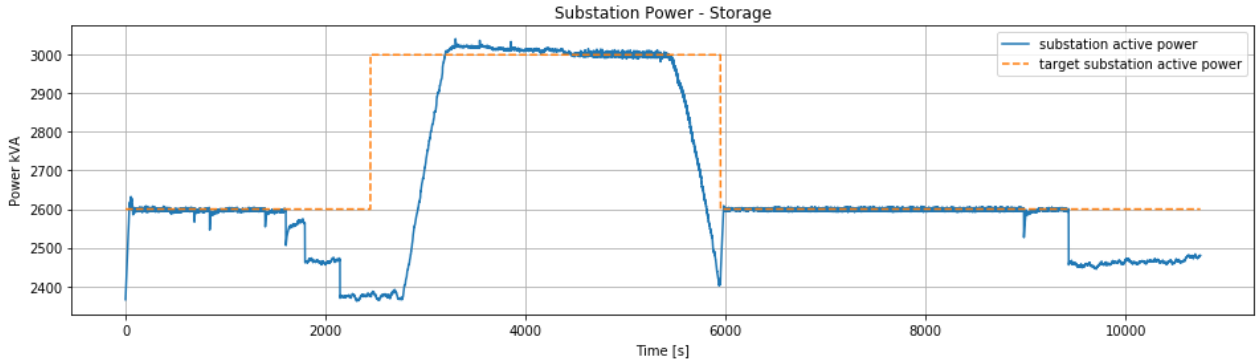

Figure 9 - Active power at the substation node after distributed batteries are used to makeup for a generation shortfall. The yellow dotted lines represent the upper allowable limit of power to be provided by the substation. This case is without any SPADES control.

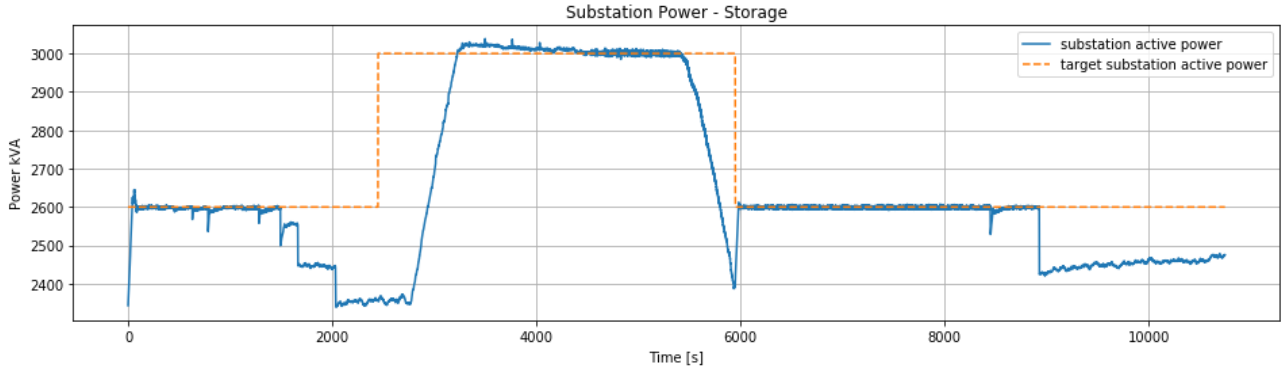

Figure 10 - Active power at the substation node after distributed batteries are used to makeup for a generation shortfall. The yellow dotted lines represent the upper allowable limit of power to be provided by the substation. This case is with SPADES control.

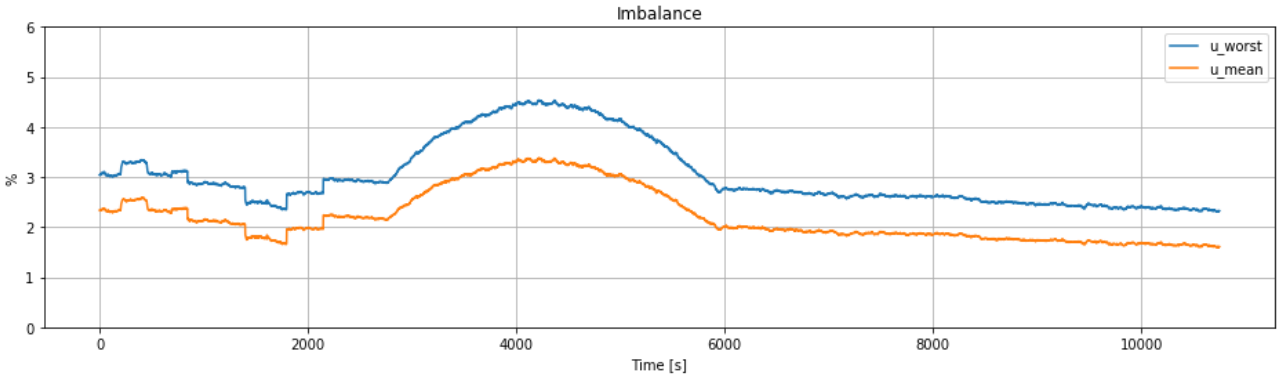

Figure 11 - Worst case and average voltage imbalance in the simulation experiments without SPADES control.

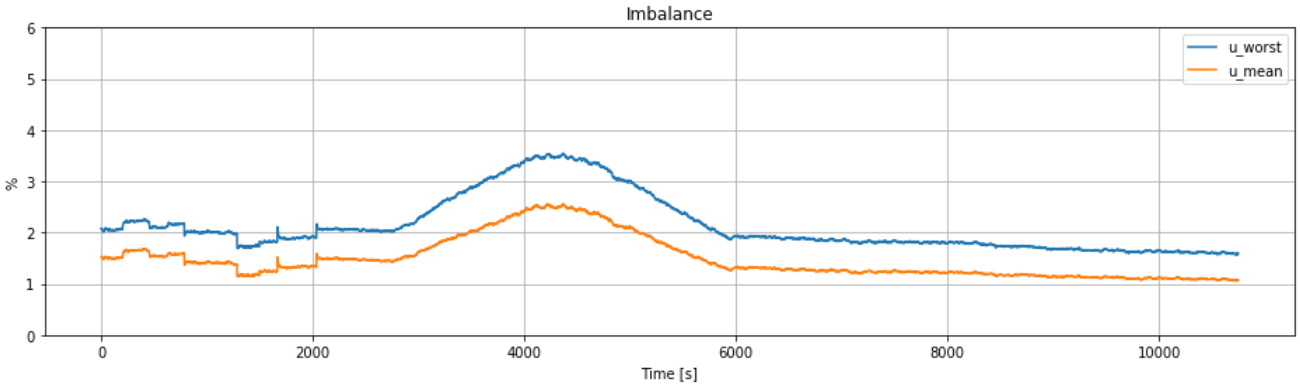

Figure 12 - Worst case and average voltage imbalance in the simulation experiments with SPADES control.

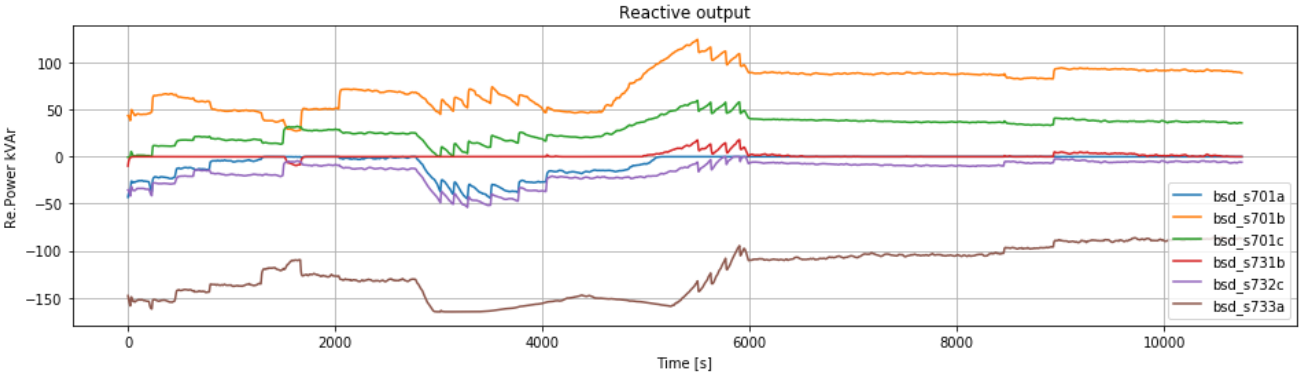

Figure 13 - Battery reactive power injections with SPADES control. Note that some power injections are negative, indicating that reactive power is being consumed.

Red Team Experiments

The Red Team, led by Siemens, was tasked with developing approaches to circumvent or degrade the performance of the SPADES algorithms. To that end the Siemens Red Team created a software addition to the SPADES software framework to introduce cyber attacks against smart inverters, energy storage systems, and legacy distribution system regulation and protection equipment. Siemens utilized a Monte Carlo Tree Search-based optimization approach to create attacks to maximize several Key Performance Indicators (KPIs). Some of the KPIs which the Monte Carlo Tree Search algorithm sought to maximize were:

1. Power Delivery Disruption: this is a function of the number of loads disconnected and the associated duration
2. Voltage Oscillations: created through destabilizing the feedback interconnection between DER smart inverter voltage regulation functions and the power grid
3. Voltage Imbalances: created through shutting off loads on certain phases of the feeder or adjusting DER smart inverter voltage regulation control parameters

The Siemens red team experiments were able to determine cyber attack scenarios (network topologies, DER deployments, etc.) for which the SPADES algorithms were ineffective. **The major conclusion of the Red Team experiments is that the SPADES algorithms suffer from degraded levels of performance when applied to cyber attack scenarios outside of their training set. Thus, it would behoove researchers studying this technology to look for methods to enhance the training set to ensure the appropriate levels of robustness to changing attack scenarios, or investigate other means to switch control policies as the environment is adjusted.** An example of an effective Red Team scenario is shown in Figure 14, where the Red Team optimization algorithm periodically adjusted the attack scenario parameters to defeat the SPADES algorithm.

Detailed description of the Red Team attack architecture, the Monte Carlo Tree Search algorithm, and efforts to speed up computational burdens of the Monte Carlo Tree Search algorithm are provided as an appendix to this report.

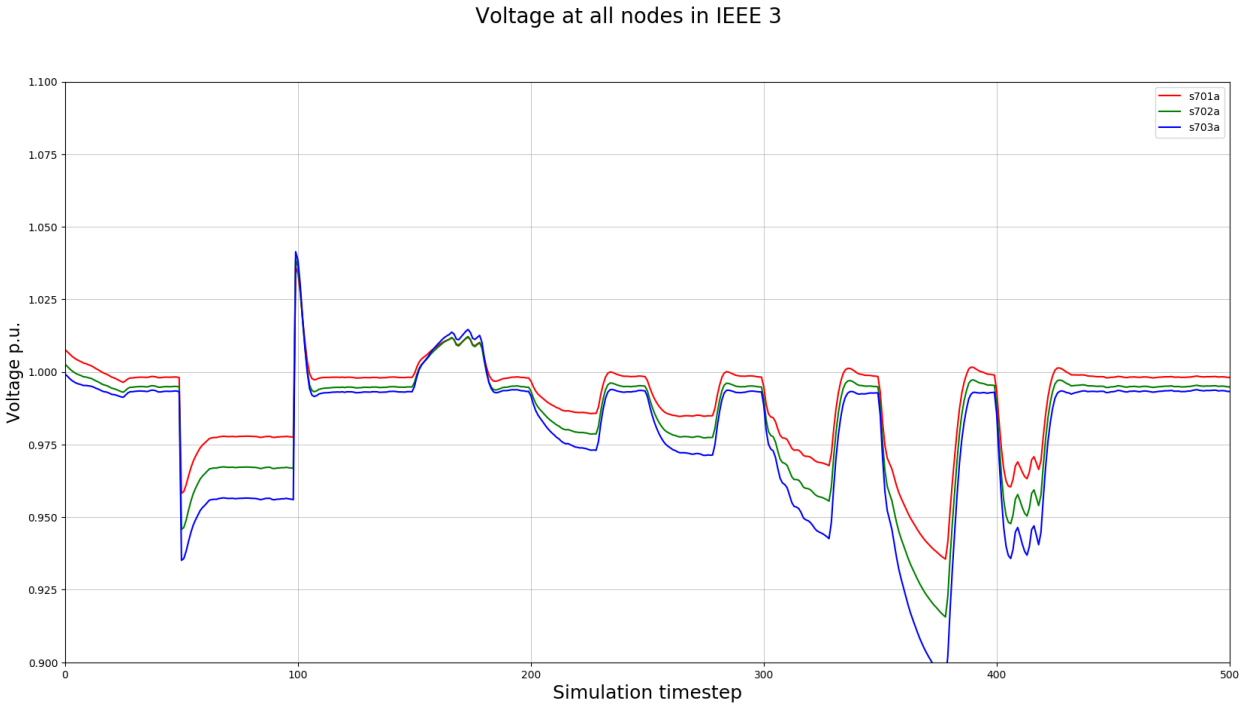

Figure 14 - Example of Red Team attack which continually adjusted the settings of DER devices in an experiment. The SPADES controllers were ineffective in mitigating the increasing levels of voltage imbalance in this experiment.

Open Modeling Framework Integration

This section details the user interface developed by NRECA to set up and visualize experiments of the SPADES algorithm in the Open Modeling Framework [2](OMF). OMF users can interact with the *cyberInverters* module to upload network models, configure the details of the cyber attack, conduct experiments, and view and download experimental results.

Model Input Interface

The Model Input interface is shown in Fig. 15, where the user can specify different parameters of the simulation. As is shown in the model, the user is responsible for determining the start time and length of the simulation as well as specifying the network model and various configuration files that specify the details of the experiment. A description of the input files is provided below:

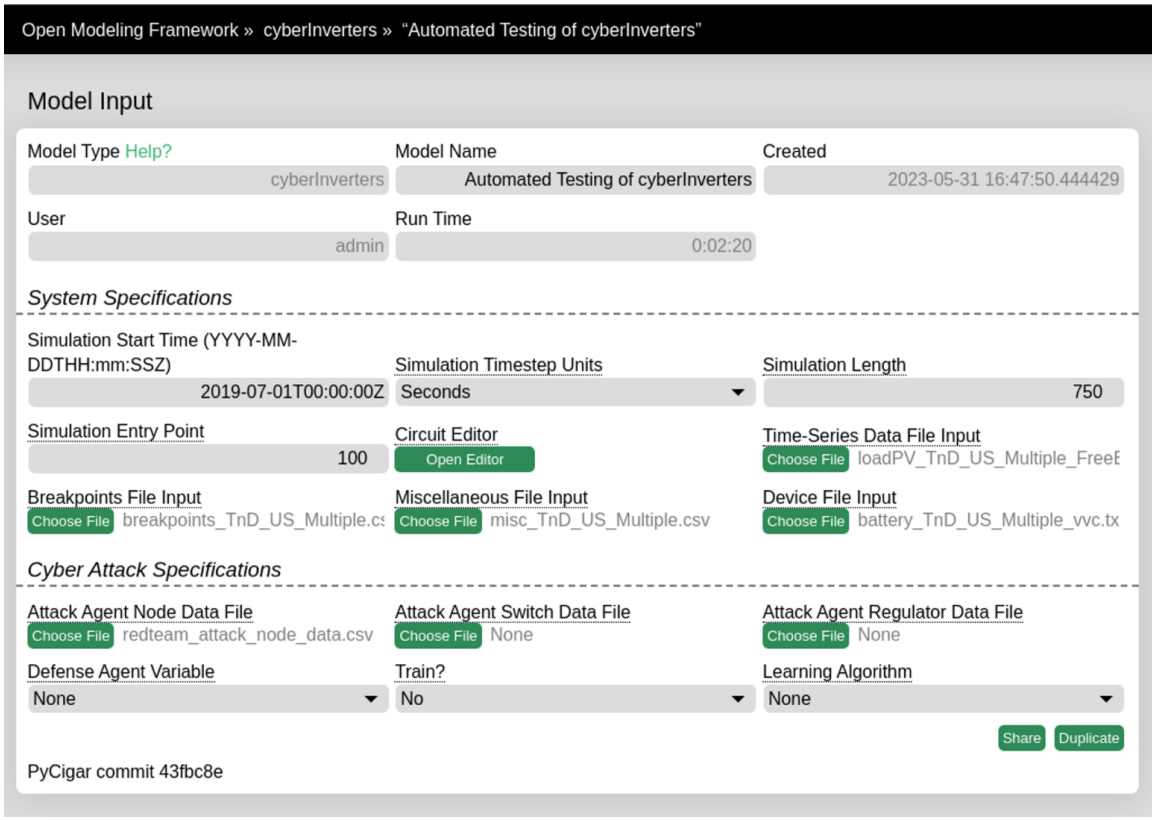

Figure 15 - cyberInverters Model Input user interface

Attack Specification Files:

- *Attack Agent Node Data File* - a .csv file containing data that pertains to an attack to individual nodes on the circuit

[2] https://www.omf.coop/

- *Attack Agent Switch Data File* - a .csv file containing data that pertains to an attack to individual switches on the circuit
- *Attack Agent Regulator Data File* - a .csv file containing data that pertains to an attack to individual regulators and/or capacitors on the circuit

Note that template data files can be downloaded from the OMF interface which can then be configured by the user. The Model Input screen also allows users to specify the type of learning algorithm used for model training (if more than one algorithm is supported). The user can select the learning algorithm from the drop down menu that allows choosing from a list of predefined algorithms that will specify how the defense agent is trained (user must also choose to train defense agent). The Learning Algorithm dropdown menu is shown in Fig. 16.

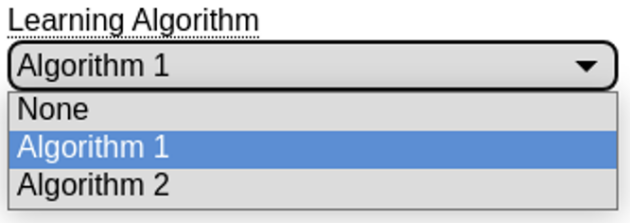

Figure 16 - Dropdown menu to choose learning algorithm in cyberInverters user interface.

Visualization

This section discusses how the results of experiments are visualized in the OMF. An example model input is shown in Fig. 17. After providing the necessary configuration files and selecting the learning algorithm (in the screen below, this drop down menu is referred to as Defensive Agent Variable", but this has been changed in the final version of the software) the user can either choose to train a new agent or run the model with an existing agent.

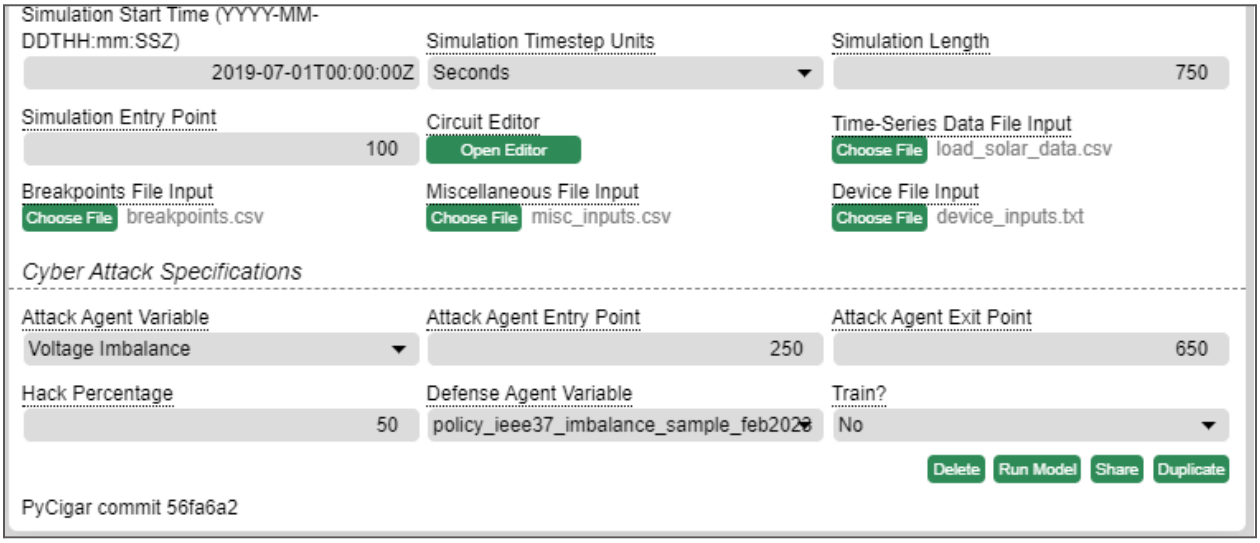

Figure 17 - Relevant model inputs for run 3 (voltage imbalance attack/LBL's defense)

It should be noted that at the time of the writing of this report, agent training can be time consuming depending on the size of the distribution network and the parameters of the learning algorithm.

Example results are now shown in the figures below.

Figure 18 displays the nodal voltage and active and reactive power outputs of inverters associated with battery storage systems at 3 nodes in the example system. The top two plots show the outputs associated with storage devices/inverters that have had their settings changed as part of a cyber attack (these are identified as “adversarial_inverters”). An inverter whose settings are being adjusted by the SPADES algorithm is shown in the bottom sub plot. The raw time series data can be downloaded as well.

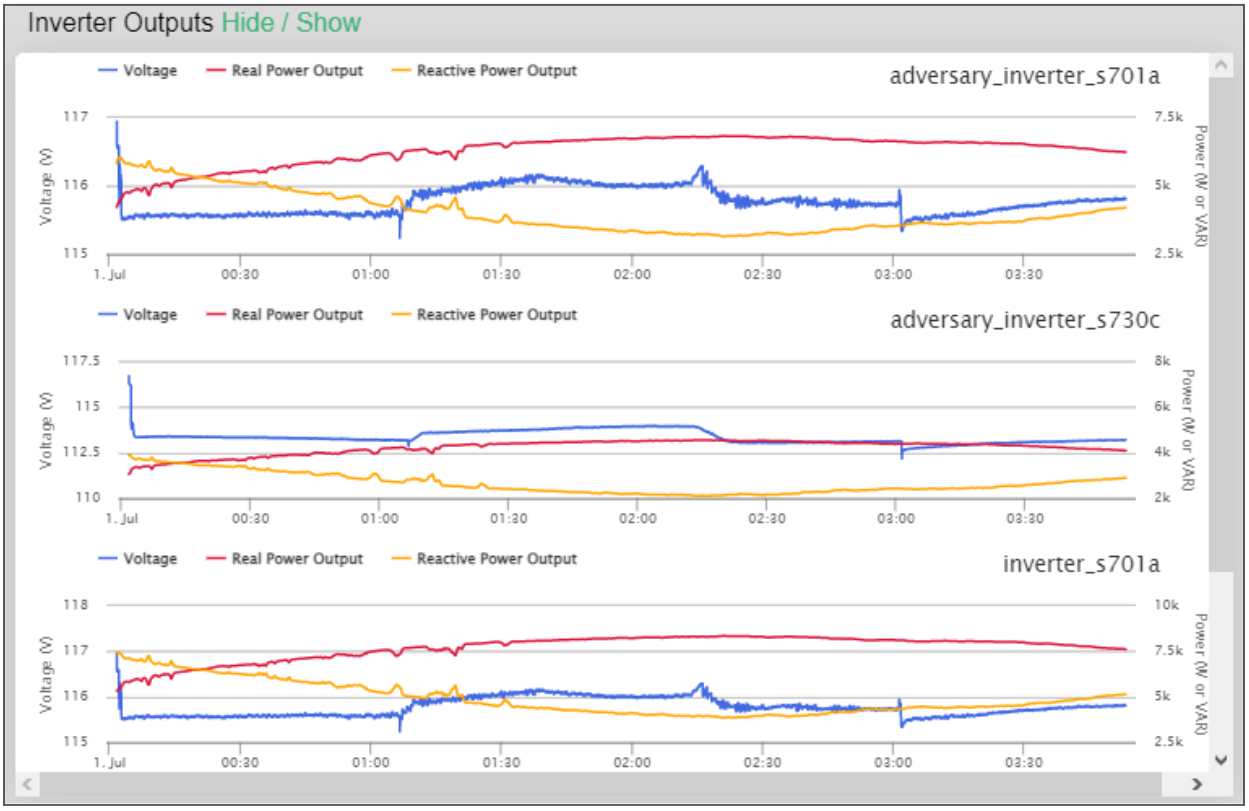

Figure 18 - Inverter outputs for a voltage imbalance attack using the defensive agent trained by LBNL.

In the no cost extension period, NRECA added visualization capabilities for the Key Performance Indicators used in the Red Team experiments. An example showing visualization of voltage imbalances is shown in Fig. 19

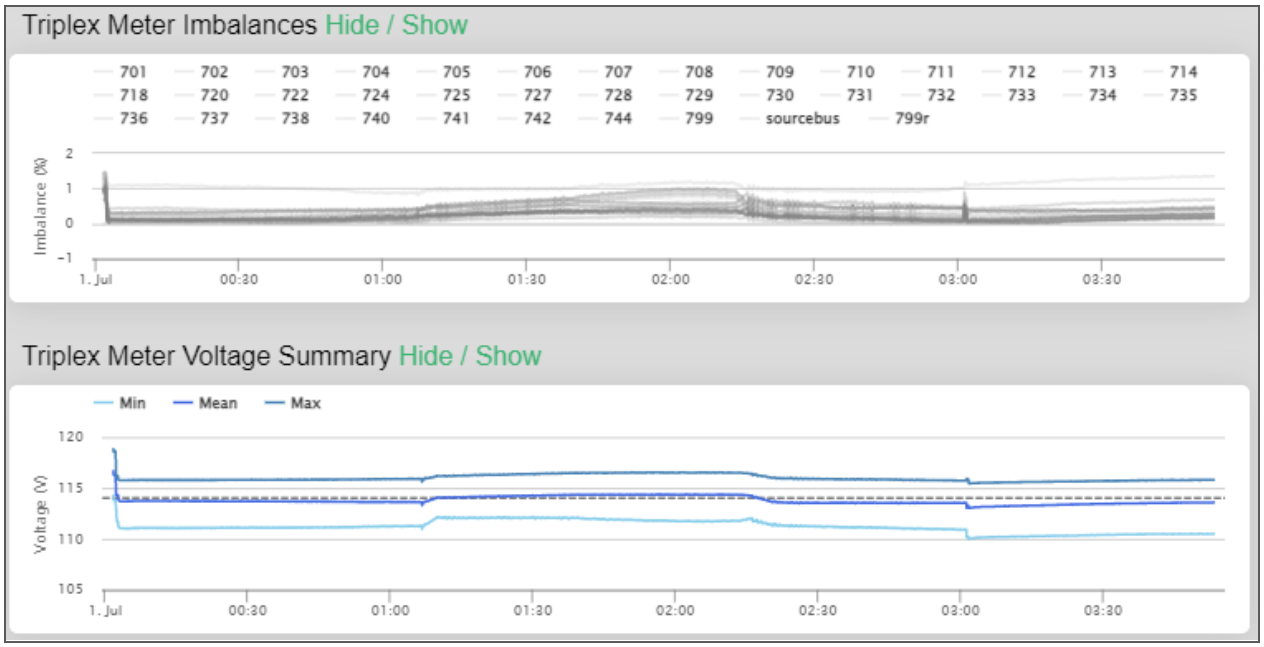

Figure 19 - Triplex meter imbalances for a voltage imbalance attack using LBNL's defensive agent.

The KPIs which are presently displayed in the OMF are:

- Voltage Oscillation over time - A summary value that quantifies the oscillation magnitude for each second
- Voltage Imbalance over time - A summary value that quantifies the imbalance magnitude for each second

OMF Integration Conclusions/Future Work

Integration of the SPADES defensive algorithms into the OMF has revealed several ways in which the user experience can be improved. At present, the OMF currently struggles to support the training of defense files due to limited computational resources. As of June 2023, The most practical way for OMF users to train defensive agents is to provide the specification to LBNL who would train the agents using HPC resources and then return the agents to the OMF. With the SPADES project ending, this will hinder OMF users from efficiently training their own defenses on a given circuit configuration.

In the future, several options could be adopted to improve the user experience:

- Option 1) Use fewer resources but extend the training time
    - Add user-facing messaging that asks them to be patient, with a rough estimated time of completion or a % progress bar. This still doesn't fit the OMF deployment environment, e.g., if multiple users try to train a defense.
- Option 2) Acquire computational resources from the cloud on-demand

- **This is highly recommended**: OMF deployment environments lack GPUs, sufficient computational resources.
- GPUs are much less expensive on-demand than provisioning the OMF deployment servers with GPUs that could meet maximum demand.
- Due to time limits, this feature has not been implemented.

Additionally, the NRECA team has identified additional outstanding questions and possible solutions that should be considered in any follow-on work:

How do we authorize OMF users to access computational resources needed to train the defensive agents?

Options considered:

1. Authorize any defense training by default for all users
    a. This option does not adequately bound compute cost adequately.
2. By request / approval only
    a. Benefit: maximum control over costs.
    b. But: this option discourages experimentation, and requires human intervention for experimentation.
3. Contingent upon payment
    a. Benefit: shifts training costs to users.
    b. But: high complexity for small anticipated cost savings, while constraining impromptu experimentation.
4. (Low) rate limited by default, more by authorization
    a. best anticipated cost/convenience/maintenance compromise.

How do we install and run the defense training environment?

Options considered:

1. Single cloud provider
    a. Simple and probably optimal
2. Two or more pre-configured cloud providers
    a. Potential runtime savings
    b. Likely too much trouble
3. Abstraction layer via infrastructure automation, e.g., Terraform
    a. Maximal potential agility, cost-savings
    b. Stack complexity and development time likely not worth the effort
4. Standalone application provided to users to train agents using their own computational resources
    a. Need to design solutions that requires minimal configuration on part of user (automatic detection and utilization of available computational resources)

Conclusions and Recommendations

The major conclusion of the SPADES project is that the use of adaptive control and reinforcement learning are effective approaches to act as supervisory controllers for DER (photovoltaic and battery storage devices) to ameliorate cyber attacks on aggregations of DER in distribution networks. When portions of DER in a given network have had their control parameters adjusted maliciously, SPADES supervisory controllers can adjust the settings of *non-compromised* DER to mitigate the attack in real time. Interested readers can review previous reports and presentations of the SPADES project to understand the algorithms and attack vectors in detail[3].

Interestingly, the SPADES team also discovered that the effectiveness of the algorithms in mitigating attacks is heavily dependent on distribution system network topology, the locations and capacities of DER which have been attacked, and the locations and capacities of DER which are used for defense. **When the attack is more evenly distributed across DER in a given network, the algorithms developed in SPADES are effective where up to approximately 40% of DER (by installed inverter capacity) are compromised.** When the attack affects larger percentages of DER capacity, there simply is not enough DER capacity available to completely mitigate the attack. **However, even though attacks cannot be entirely mitigated, the SPADES algorithms have consistently shown that attacks on DER smart inverter control settings can be ameliorated by controlling whatever DER is available.** Thus, the algorithms produced in this project could enhance the cyber resiliency of a network even if installed in a single device. The positive effects in mitigating attacks will grow with increasing numbers of devices using the SPADES algorithms.

The major result of the red team experiments showed that the performance of the SPADES algorithms was degraded when tested in networks different from the network used to generate the reinforcement learning algorithm training data. Two conclusions can be drawn from this result. The first is that the SPADES solution should be effective if agents can be trained on a network-specific basis. This has implications for entities wanting to implement the SPADES solution as this may require computational resources not traditionally found within utilities (certainly for electric cooperatives). Secondly, the reinforcement learning training architecture adopted by the SPADES team does not result in policies that are sufficiently generalizable to different network topologies. **Further effort could be expanded to improve how reinforcement learning agents are trained for DER cybersecurity so that a single agent is effective for differing network structures.**

In addition to the development of algorithms for DER cybersecurity, the research and development conducted in the SPADES project has made substantial contributions to academic literature, workforce development, and technology transfer. Through the 3 years and 7 months of the project, five conference papers and eight journal papers have been published in high impact venues (such as IEEE Transactions on Power Systems). A list of publications produced by the project is available on the SPADES website. The project has supported three graduate

[3] https://secpriv.lbl.gov/project/ceds-spades/

students who have gone on to roles in EPRI, Span Inc., and the Midwest Independent System Operator, as well as one postdoctoral researcher. The project has been presented to the NERC Security Integration and Technology Enablement Subcommittee (SITES), the Department of Defense, the Department of Homeland Security, Siemens, Northrop Grumman, and several universities. Additionally, a patent on technology developed in SPADES has been filed by DOE and is presently under review [9].

The project team wishes to thank the CESER Risk Management and Tools program and the National Energy Technology Laboratory management team for their sponsorship and management of the RMT program and the SPADES project.

References

1. “FERC Order No. 2222: Fact Sheet,” Available: https://www.ferc.gov/media/ferc-order-no-2222-fact-sheet, Federal Energy Regulatory Commission, accessed: Sept. 2022. [Online].

2. IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces, Institute of Electrical and Electronics Engineers, IEEE 1547-2018, April 2018.

3. Rule 21 Interconnection, Available: https://www.cpuc.ca.gov/Rule21/, California Public Utilities Commission, Std.

4. B. Seal, “Common Functions for Smart Inverters, 4th Ed.” Electric Power Research Institute, Tech. Rep. 3002008217, 2017.

5. “Cybersecurity via Inverter-Grid Automatic Reconfiguration (CIGAR)”, Available: https://secpriv.lbl.gov/project/ceds-cigar/, Lawrence Berkeley National Lab., accessed: June 2023.

6. FLEXGRID, Available: https://flexlab.lbl.gov/flexgrid, Energy Technologies Area, Lawrence Berkeley National Laboratory, 2023.

7. C. Roberts, S. Ngo, A. Milesi, A. Scaglione, S. Peisert, and D. Arnold, “Deep Reinforcement Learning for Mitigating Cyber-Physical DER Voltage Unbalance Attacks,” Proceedings of the 2021 American Control Conference (ACC), virtual, May 2021, pp. 2861-2867.

8. Arnold, D., Ngo, S., Roberts, C., Chen, Y., Scaglione, A., and Peisert, S., “Adam-based Augmented Random Search for Control Policies for Distributed Energy Resource Cyber Attack Mitigation,” Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA., June 2022, pp. 4559-4566.

9. Arnold, D. (2022). System and Method for Control of Distributed Energy Resources for Distribution Grid Voltage Stability (U.S. Patent Application Serial No. 17956310)

Appendix - Red Team Reports

The following section contains reports written by the Red Team (Siemens) detailing the red team testing approach and experimental results. As the red team did not interact with the rest of the SPADES project team during the project, these reports were independently written by Siemens and are included here in their entirety.

REPORT Subtask 3.2 – Deliverable 3.2.1

Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES)

DOE CESER

CEDS Program

SUBMITTED BY

Siemens Corporation Technology

755 College Rd East, Princeton NJ

Submitted: Dec. 13th, 2021

Technical Point of Contact

Dr. Bruno Leao - bruno.leao@siemens.com

Project Manager

Ramamani Ramaraj - ramamani.ramaraj@siemens.com

SUBMITTED TO

Lawrence Berkeley National Laboratory

Contents

Introduction

This report describes the work developed by Siemens Technology (ST) and corresponding results related to Subtask 3.2 as part of the Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES) project. ST plays the role of red team in the project. Following the preliminary analysis of potential attacks developed as part of Subtask 3.1, the scope in Subtask 3.2 comprised the development of small proofs of concept with the goal of validating the proposed attacks, evaluating the requirements for producing these attacks in PyCIGAR and supporting the definition of proper approaches for the associated implementation. Proofs of concept have to a great extent been based on OpenDSS simulations using IEEE grid models. PyCIGAR implementations were also developed based on small manipulations of existing functionality for a prototype implementation of the new behaviors required for producing the attacks and their integration into the original framework. Those prototype developments were discussed with LBNL in order to define the changes required in PyCIGAR for final implementation of required functionalities.

Besides the proofs of concept, the work in Subtask 3.2 also included initial implementation of means for representing computer network information associated to power systems in PyCIGAR. This information will be employed for definition and evaluation of attack costs which will in turn be used together with pre-defined attack budgets to limit the action of the adversary in order to make the attacks more realistic. Detailed definitions of this approach can be found in Subtask 3.1 report.

Attack Proofs of Concept

This section describes the proofs of concept developed using OpenDSS and prototype implementations in PyCIGAR to evaluate the proposed attacks and guide the final implementation of functionality required for their execution.

Focus was on system level attacks which adversely affect the power system behavior in the short (order of seconds) to medium term (order of minutes). These attacks can be a result of a single compromised component or a coordinated attack on multiple components that leads to network instability and/or power delivery disruption. KPIs for definition of successful attacks are described in detail in Subtask 3.1 report. Based on discussions with NRECA and LBNL an additional Tier 1 KPI was included, corresponding to substation power factor. Attacks that result in the substation power factor being out of the acceptable range, usually between 0.95 and 1.05, are considered successful as most distribution utilities will pay penalties to the transmission operator if they exceed these limits. Many of the attack types described below, such as the ones associated to connecting/disconnecting loads, DERs and capbanks, can affect the substation power factor. Below is an updated list of KPIs. They are currently defined qualitatively only:

- Tier 1
    - Power delivery disruption, including
        - DER disconnection based on IEEE 1547 standard
        - Disconnection due to overloading of lines/transformers
    - Instability (oscillation)
    - Voltage imbalance
    - Substation power factor
- Tier 2
    - Equipment useful life degradation
    - Power quality degradation (poor power factor or over/undervoltage conditions)

A single attack may affect multiple KPIs. For instance, a successful attack could cause voltage oscillations and imbalances simultaneously while also disrupting power delivery. All attacks considered here were discussed with project partners including LBNL and NRECA and verified as being reasonable and viable considering the project needs and real-world operation.

PyCIGAR Integration

For the PyCIGAR integration, prototype implementations were created and several design decisions were discussed with LBNL to provide a granular controllable environment for deploying attacks to the power system simulations during the execution of tests. Discussions and implementations were performed considering a separation of concerns between the core PyCIGAR functionality and the additional functionality required for deployment of attacks, referred here as “Red Team Addon”, and also the easy integration path for this Red Team Addon.

Separation of Concern

The PyCIGAR Framework is still evolving and improving. At the same time the Attack Addon prototyped by Siemens and discussed with LBNL will also evolve over time. Final integration and development of related functionalities will be developed by LBNL.

In order to not interfere with the development of core functionality, the Red Team Addon was implemented as independently as possible, but still integrated with PyCIGAR. Controller and Devices implementing hacked functionality were developed which inherit the control functionality of the parent class they hack and are therefore capable of mimicking the same behavior as the corresponding non-hacked classes. The handover of parameters to the simulation utilizes the existing infrastructure and modifies or adds to the existing parameters. Those classes are stored in a separate location and are accessible for use into PyCIGAR models using the usual means for importing and device registration.

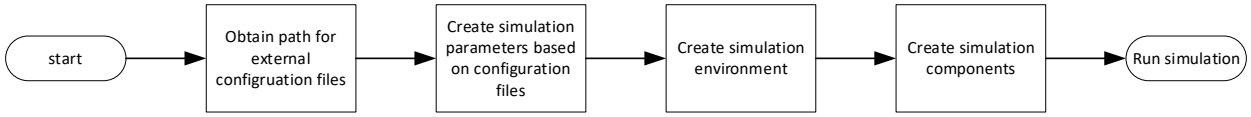

Figure 1: Workflow PyCIGAR

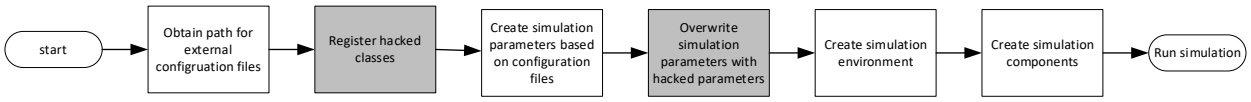

Figure 2: Workflow Attack Addon

Comparing the PyCIGAR workflow for running a simulation (Figure 1) with the workflow using the Attack Addon (Figure 2) two additional steps are included. The registration of the hacked classes uses the same mechanism used by PyCIGAR, for instance:

```
register_devcon('hacked_controller_pv', HackedControllerPV)
```

After the classes are registered, simulation parameters associated to the specific attack type are overwritten.

Granular controllable environment

The Red Team Addon enhances the current PyCIGAR functionality for attacks on the power grid. The addon enables the execution of multiple attacks defined independently over time, which is referred to here as horizontal scaling, and also provides means for executing various types of attacks affecting the same device/controller, which is referred to here as vertical scaling. These functionalities provide flexibility for the red team to attack different devices of the power grid represented in PyCIGAR during different times in the simulation.

Horizontal Scaling

Every configured attack in the Red Team Addon is associated to independent start and end times. During this period in the simulation the corresponding device will have its functionality affected. Each device can have more than one attack per simulation at different start and end times as long as there are no overlaps among their time periods. Figure 3 shows the differences in definition of a device where an attack takes place. Originally, attacks defined in PyCIGAR are configured by defining when the default controllers' settings will be used and when the adversary controllers' settings will be used, all at once. Through the new structure attacks can be individually designed.

```
# Hacked controller with Attack Addon
{'name': 's701a', 'devices': [{'name': 'inverter_s701a', 'device': 'hacked_pv_inverter_device',
'controller': 'rl_controller', 'custom_controller_configs': {'default_control_setting': [0.974, 1.004, 1.034, 1.034, 1.064]},
'adversary_controller': 'hacked_controller_pv', 'adversary_custom_controller_configs': {'default_control_setting': [1.01, 1.01, 1.01, 1.01, 1.02],
'attack_1': {'hack_start': 40, 'percentage_hack': 0, 'hack_end': 50, 'attack_type': 'voltageunbalance_attack'},
'attack_2': {'hack_start': 60, 'percentage_hack': 0,  'hack_end': 70, 'attack_type': 'voltageunbalance_attack'}},
'hack': [0, 1, None]}]}

#Hacked Controller without Attack Addon
{'name': 's701a', 'devices': [{'name': 'inverter_s701a', 'device': 'pv_device',
'controller': 'rl_controller', 'custom_controller_configs':{'default_control_setting': [0.974, 1.004, 1.034, 1.034, 1.064]},
'adversary_controller': 'hacked_controller_pv', 'adversary_custom_controller_configs': {'default_control_setting': [1.01, 1.01, 1.01, 1.01, 1.02],
'hack': [0, 1, None]}}]}
```

Figure 3: Horizontal scaling provided by Red Team Addon

Vertical Scaling

Hacked devices in Red Team Addon can provide the possibility to execute different kinds of attack. For each attack an attack type needs to be specified. The specified attack will be executed at the specified time and according to the defined duration. For each attack there are means for providing additional parameters to adjust the behavior according to the specific situation. The implementation of the various hacked behaviors are part of the hacked controller and respective hacked device classes. Figure 4 presents an example of part of the definition of a hacked component where different types of attacks can be triggered.

```
if attack_type == "voltageunbalance_attack":
    unbalanced_attack(control_settings, device_id[-1])

if attack_type == "volt_var_attack":
    volt_var_attack(control_settings, env, node_id)
```

Figure 4 - Vertical scaling provided by Red Team Addon

Easy integration

To integrate the functionality of the hacked components, classes are developed that hold the functionality of the hacked devices and controllers. The classes corresponding to the hacked components inherit the functionality of the original classes from the PyCIGAR framework. The use of the hacked device classes follows the same principles as the regular ones: classes must be registered in PyCIGAR before the simulation, and they implement the correspondent alternative behavior executed during the simulation. Figure 5 presents an example of part of definition of a hacked PV device inheriting from the original PV device class.

```
class HackedPVDevice (PVDevice):
    def __init__(self, device_id, additional_params, is_disable_log=False):
        super(HackedPVDevice,self).__init__(device_id, additional_params, is_disable_log)

    def update (self, k):
        super(HackedPVDevice, self).update(k)
```

Figure 5: Example for implementation of HackedPVDevice

Developed classes and their integration

Table 1 presents hacked component classes developed as part of the work developed in Subtask 3.2.

Table 1 - Hacked versions of devices and comtrollers implemented in Subtask 3.2.

pyCIGAR class	Addon class
PVDevice	HackedPVDevice
BaseController	HackedControllerPV, HackedControllerLoad, HackedControllerBattery
SwitchController	HackedSwitchController

PyCIGAR workflow as depicted in Figure 6 is not changed in a significant way. During the attacks, hacked controllers provide commands to the hacked devices, where they are applied to the simulation.

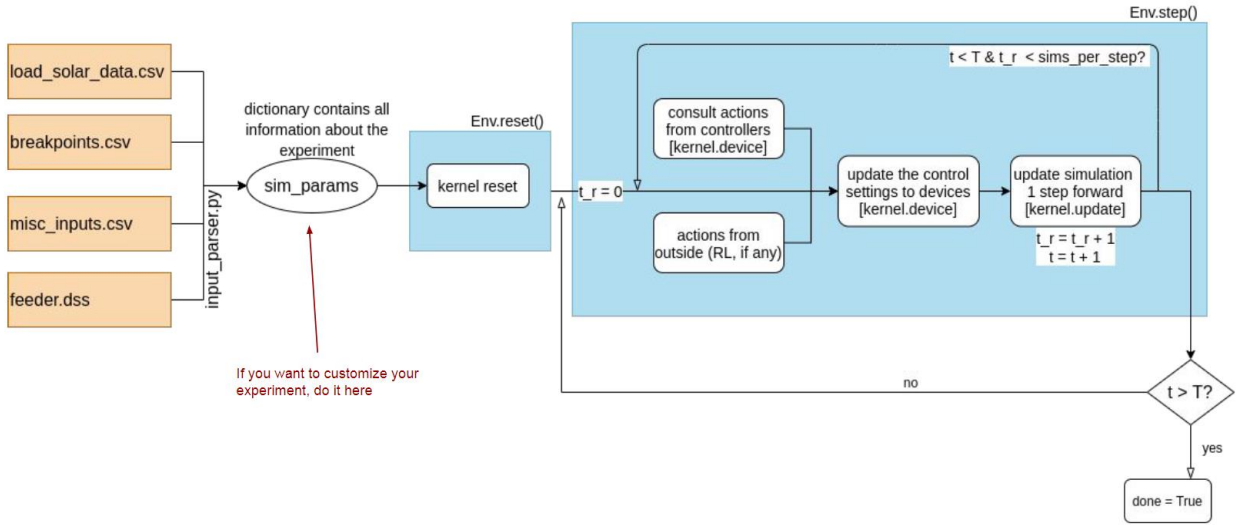

Figure 6: PyCIGAR workflow (provided by LBNL - Sy-Toan Ngo)

The only change in the diagram of Figure 6 as a consequence of the Red Team Addon is the use of an additional input file defining the information and parameters for each attack. The attack parameters are parsed based on a wrapper defined around the original input parser defined in PyCIGAR. Figure 7 presents a high-level view of how the wrapper is structured. In the figure, *network_settings* refer to the representation of the computer network discussed in section Attack Budget and *attack_settings* correspond to the definition of the attacks as described above.

```
def redteam_addon_parser(computer_network_paths,redteam_attack_path, pyCIGARconfig ):
    sim_params = input_parser(pyCIGARconfig)
    if computer_network_paths:
        parse(network_settings)
    if redteam_attack_path:
        parse(attack_settings)
    return sim_params
```

Figure 7: High level view of the Red Team Addon parser which is a wrapper to the original PyCIGAR parser.

Topology Reconfiguration

For this attack scenario we consider that the attacker can take control of Normally Open (NO) and Normally closed (NC) and/or sectionalizing switches in the power network. In essence, the attacker can close any open switches and open any closed switches. Given the radial structure of most distribution networks an attacker can easily create islands. However, as outlined in Subtask 3.1 report we do not support islanded operations in this project and consider such attacks to be out of scope. Any attacks that reconfigure the topology to other radial structures (without creating islands) is still considered in scope and will be part of our potential attack vectors. Topology reconfiguration attacks aim to i) increase the overall length of the feeder to cause under/over-voltage conditions that may lead to disconnect of DERs (if limits are violated according to the IEEE 1547 standard) and loads; ii) overload certain lines and/or transformers to trigger protective devices and cause power delivery disruption (PDD). Repeated operation or switching of grid topologies can also lead to oscillations. Note that topology reconfiguration can be intra-feeder (within the same feeder) or inter-feeder (between two or more feeders). To demonstrate the effect of topology reconfiguration attacks we augmented the OpenDSS model corresponding to the IEEE 37-bus network with additional switches as presented in Figure 8.

An over/under voltage type PDD attack consists of the following steps:

- Step 1: Open NC line or sectionalizing switch
- Step 2: Close a NO switch to obtain a radial topology (intra- or inter-feeder)
- Step 3: Repeat 1-2 to cause voltage oscillations (optional)

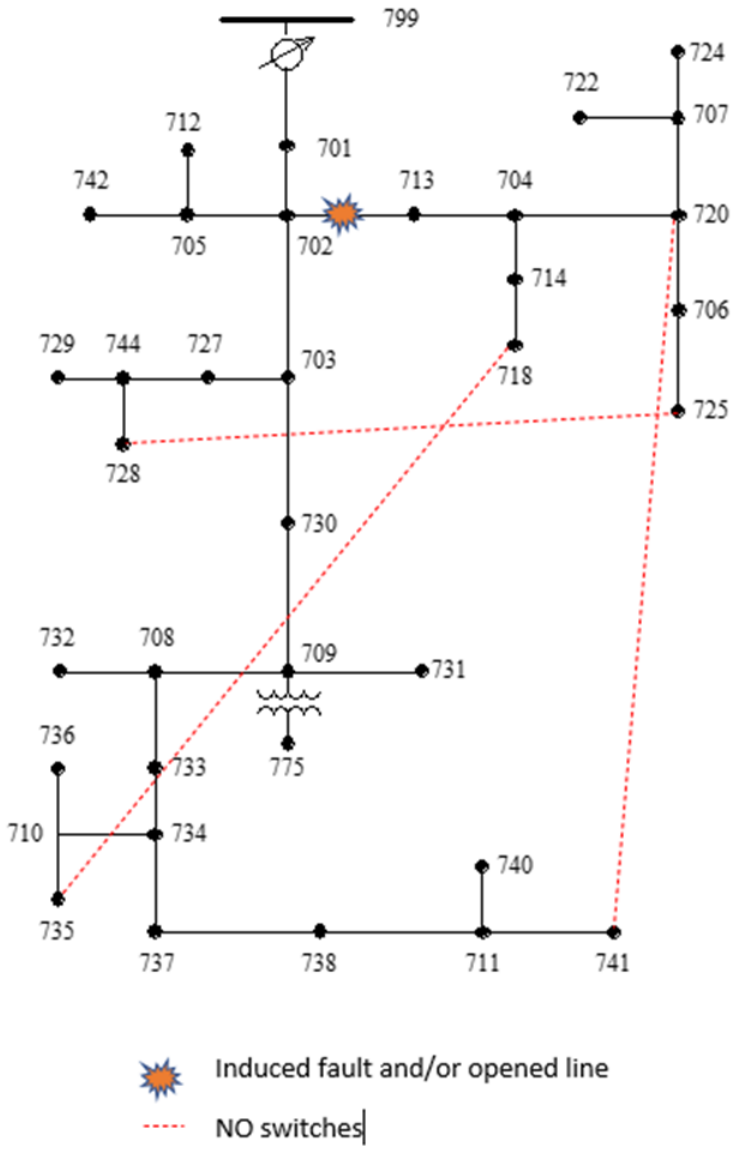

Figure 8: Modified IEEE 37-bus network. Red dashed lines denote the NO switches that were added to the base network

Following steps 1-2, we opened the line connecting buses 702-713 (equivalent to opening a NC switch) and then closed NO switch connecting buses 741-720 on the modified IEEE 37 bus network. All tests were performed directly in OpenDSS. As seen in Figure 9, the feeder is longer after reconfiguration and experiences lower voltages towards the end of the feeder. Although the test here comprises only intra-feeder reconfiguration, we also plan to consider inter-feeder reconfigurations in the future. Inter-feeder reconfigurations are more likely to cause overloading on lines and transformers due to the additional load that needs to be picked up. To mimic such a scenario in the intra-feeder reconfiguration setup we

uniformly scale the loads to cause overloading on a line and the substation transformer. Table 2 presents additional information.

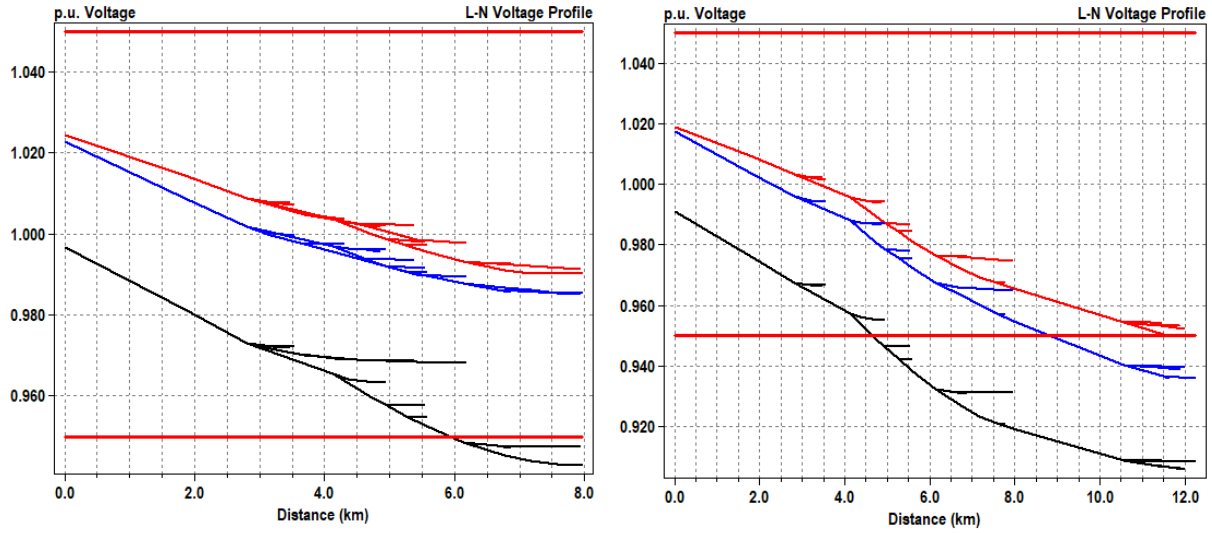

Figure 9: Comparison of voltage profiles before (left) and after (right) topology reconfiguration attack.

Table 2 - Overloading on the substation transformer connected to bus 799 and the line 799-701. The first column shows the elements that are overloaded, the second and third columns show respectively the amount by which the current (amps) and the power (kVA) exceed the nominal ratings of the affected elements.

Element	AmpsOver	kVAOver
Transformer.SUBXF	9.73	132740.4
Line.L35	42.42	336.01

For implementing the attack into pyCIGAR a new device class was developed. The pyCIGAR Framework was enhanced with a switch device class consisting of two dependent switches so that the opening of a switch is combined with the closing of another one for topology change. For testing purposes, the IEEE3 network was used. Figure 10 presents the corresponding circuit diagram.

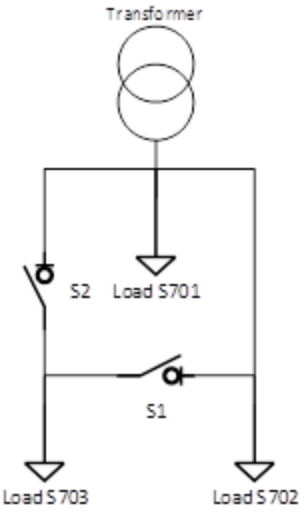

Figure 10 - diagram of the setup used to test topology reconfiguration attacks in PyCIGAR

Two switches were introduced as presented in the figure. Switch 1 connects the busses 702 and 703. Switch 2 connects the busses 701 and 703. Through their dependability opening Switch 1 results in the closure of Switch 2 and therefore a topology change. The results of the changes in the topology performed through PyCIGAR are shown in Figure 11.The left picture shows the simulation without attack. The right picture shows the execution of the attack twice at different times within the simulation. The voltage on node S703 increases due to the direct connection to S701 after switching S1 and S2. To execute this attack with the Red Team Addon a HackedSwitchController was introduced which provides means for operating the two dependent switches to create a new topology. The class provides the possibility to execute one or several topology changes during a simulation. For final implementation it was agreed with LBNL that the reconfiguration possibilities will be pre-defined.

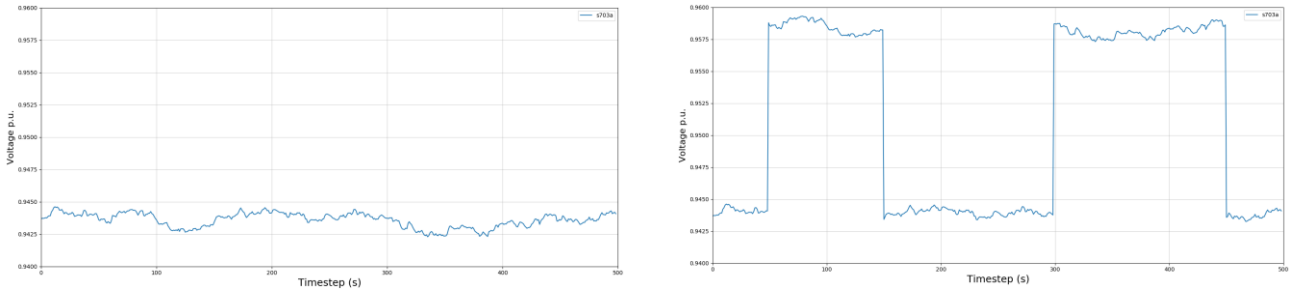

Figure 11: Voltage results comparing normal operation and topology switching in IEEE 3 network

Load/DER Disconnect Attack

For this attack scenario we consider that the attacker can quickly connect/disconnect DERs or loads in the network by opening switches. The intent of these attacks is to do phase targeted connects/disconnects that directly impact the voltage imbalance KPI. In particular, selective load shedding/increase on a single phase can worsen phase imbalance. Moreover, repeated such actions could also cause oscillations. We demonstrate such an attack on the base IEEE 37-bus network through simulations in OpenDSS. The base case (see left panel of Figure 12) already has severe imbalance between phases “c” (black line) and phases

"a" (blue line) and "b" (red line). To worsen phase imbalance between phases "a" and "b", we disconnect the following single-phase loads connected to phase a: (S701a, S714a, S738a). As evidenced from the right panel in Figure 12, the phase imbalance is indeed worsened between phases "a" and "b". A similar affect could be achieved by bringing on additional single-phase load or DERs. Figure 13 presents the results of combination of load disconnect type attack with the topology reconfiguration attack. Such a combo attack degrades the voltage, while worsening imbalance at the same time.

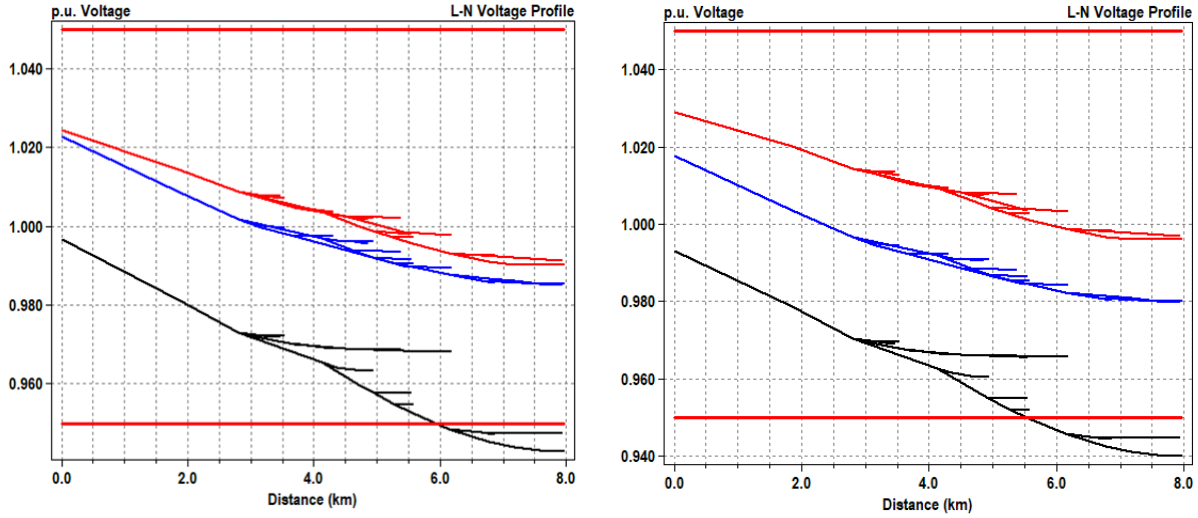

Figure 12: Comparison of voltage profiles before (left) and after (right) phase imbalance attack through disconnection of loads on phase a (blue line).

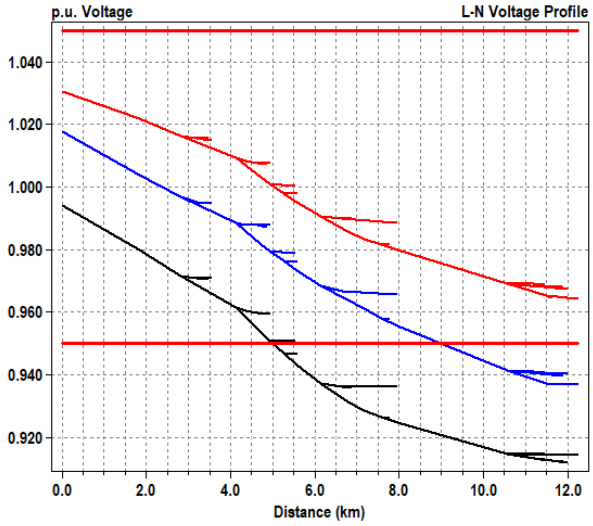

Figure 13: Feeder voltage profiles after topology reconfiguration + phase imbalance attack

For the integration of this attack to PyCIGAR additional classes were introduced through the Red Team Addon. The HackedLoadDevice and HackedControllerLoad represent together a hacked load and the means for manipulating such load. The attack is parameterized by a value defining the scale of change of

the load value. Figure 14 shows the test results of execution of this attack using PyCIGAR. On the left picture the simulation was executed with no attack. On the right picture the load was scaled down. Two attacks are performed where in the first one the load was scaled down with a factor 0.9 and in the second one a factor of 0 is used, which corresponds to a load shed. The voltage increases accordingly on node S701 in the IEEE 3 network. For each load that can be target of an attack, corresponding device and controller must be included in the simulation.

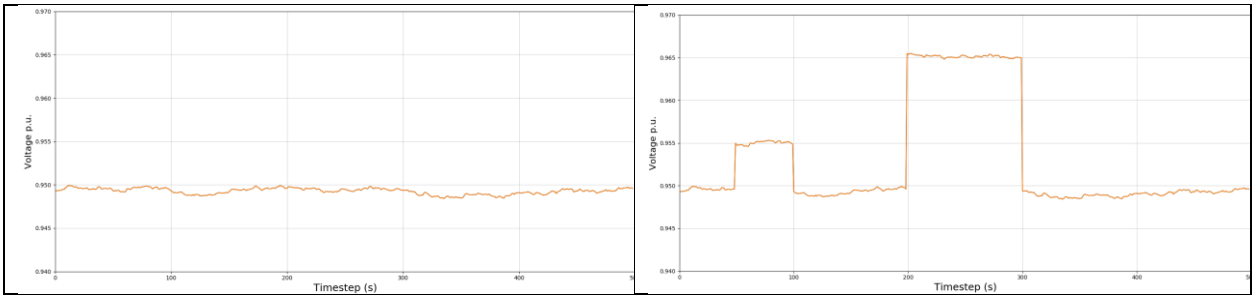

Figure 14: Load scaling attack on IEEE 3 network. Left picture: No attack. Right picture: Load scaling with factor 0.9 and with factor 0

Regulator Attack

For this attack scenario we consider that the attacker can either: i) manually change taps on substation transformer load tap changers (LTC's) and/or line regulators; ii) change settings for the operation of LTC's or line regulators. Here we demonstrate the effects from the latter through OpenDSS simulations on IEEE feeder models. Such type of attack is intended to target the PDD KPI through creation of over/undervoltage conditions. Similarly to other cases, cycling changes could also be employed to cause oscillations.

As a first test we disable the substation transformer LTC on the IEEE 37-bus feeder. As seen in Figure 15, the overall voltage profile (across all three phases) degrades significantly once the LTC is disabled.

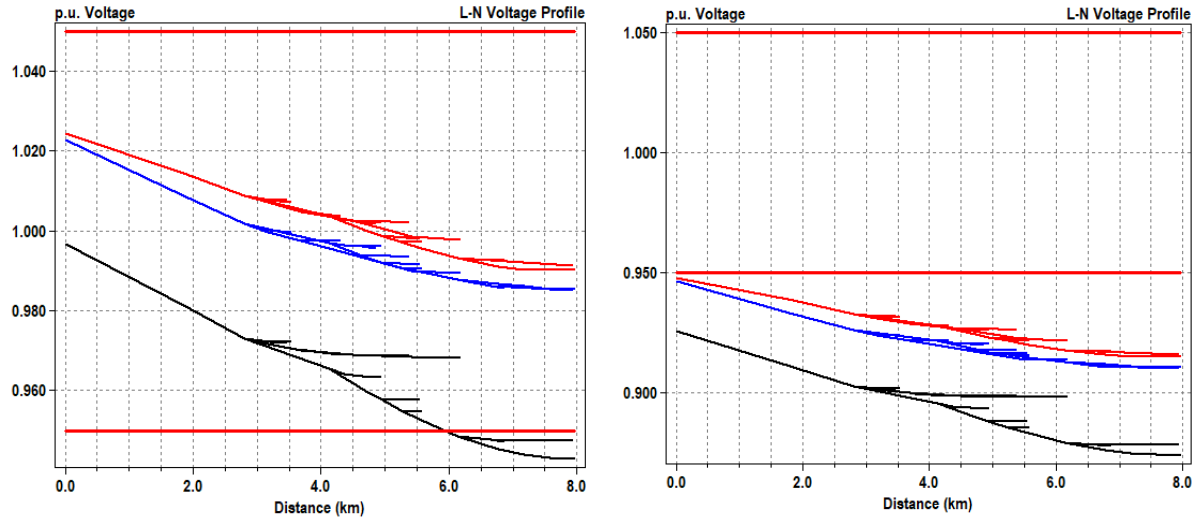

Figure 15: Comparison of Voltage profiles with default settings of regulator at substation transformer (left) and with regulator disabled (right).

The second attack consists of a coordinated effort to change delay settings of the LTC and line regulators on the IEEE 123-bus feeder. The default settings usually allow for a longer time delay on the LTC and shorter time delays on the line regulators. Here we reverse the delays such that the substation LTC acts after the line regulators. This reverse delay raises voltages as seen in Figure 16 and can lead to severe over-voltage conditions.

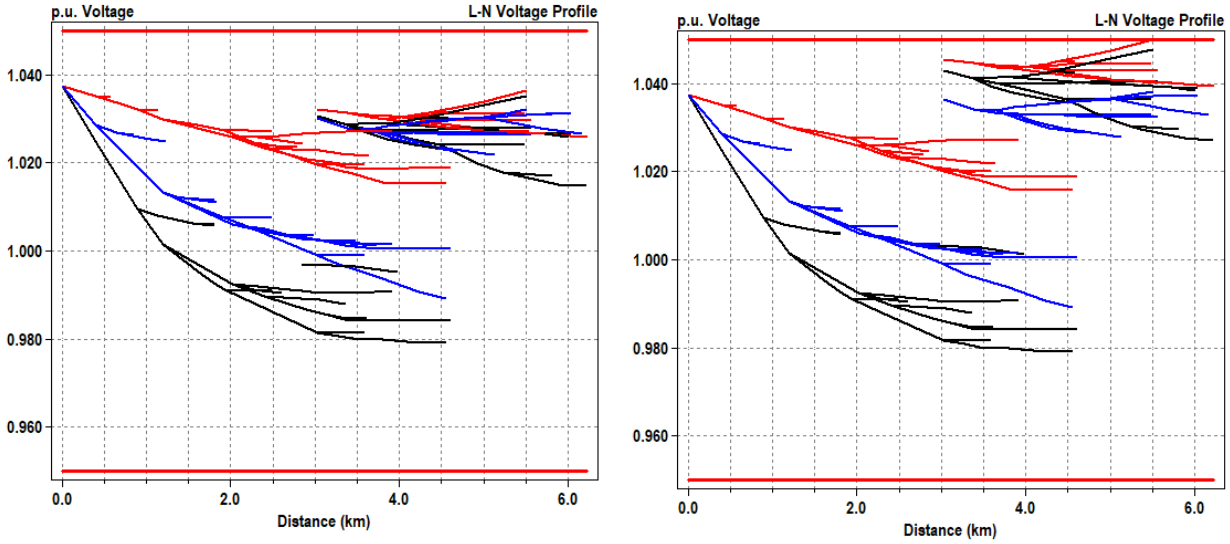

Figure 16: Comparison of voltage profiles with default settings of substation transformer regulator and other line regulators (left) and with time delay settings reversed (right). This test was performed on the IEEE 123 bus network.

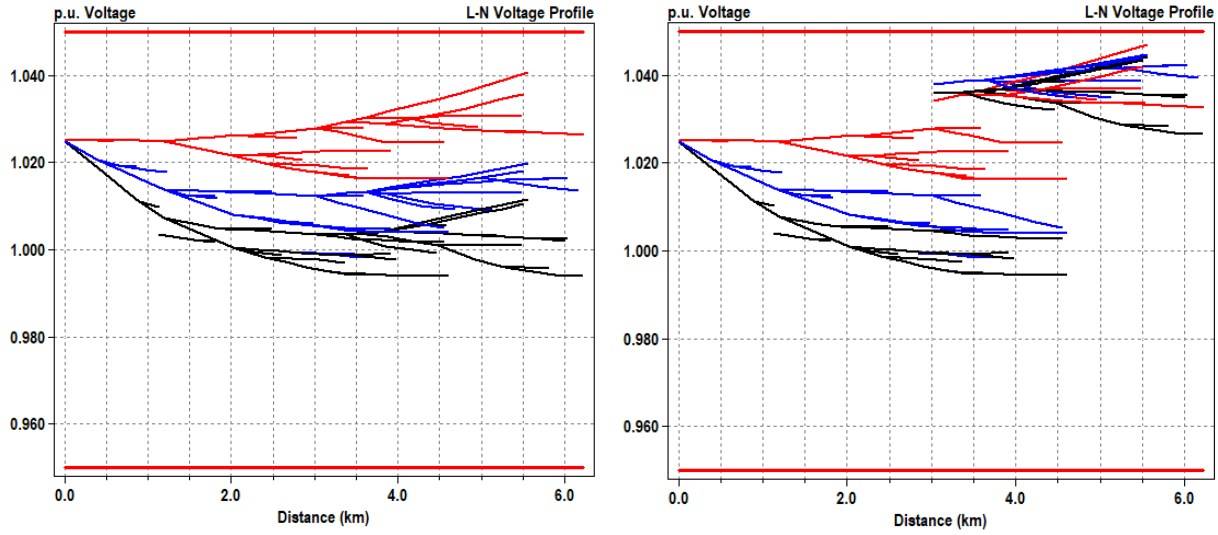

Figure 17: Comparison of voltage profiles with default settings of substation transformer regulator and other line regulators for reverse power flow (left) and with regulator setting disabled for reverse power flow (right).

The third attack is performed on a modified IEEE 123-bus network with high penetration of DERs. The DERs are modeled to create reverse power flow conditions in the feeder. The default settings for line-

regulators bring the taps back to the neutral position on detecting reverse power flow. As shown in Figure 17, disabling the reverse power flow setting can lead to over voltage conditions.

The fourth attack changes the regulator settings to operate on reverse power flow but does not force the regulator taps to the neutral position. Essentially, the regulator works like it would under normal operating conditions. This interestingly leads to severe over voltage on two phases and undervoltage on one phase as presented in Figure 18 (left panel). Such an attack could affect both the PDD KPI and the voltage imbalance KPI. The last attack combines the former (fourth) attack with the second attack to worsen voltage imbalance and undervoltage conditions. Corresponding results are presented in the right panel of Figure 18.

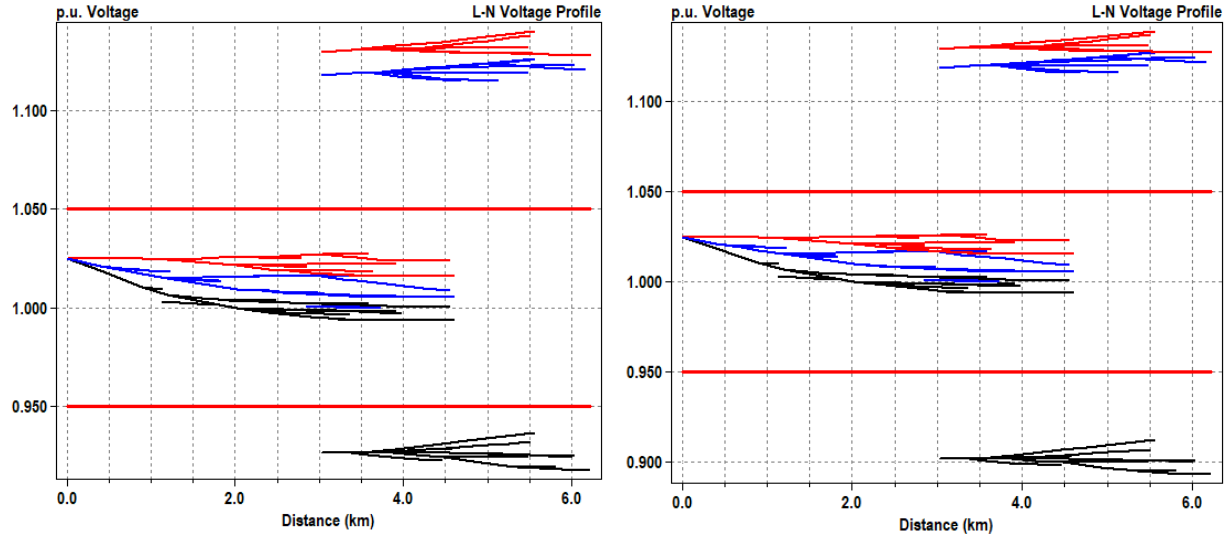

Figure 18: Comparison of voltage profiles with regulator settings modified to operate to a non-neutral tap position under reverse power flow (left) and combination of previous attack with time delay settings reversed (right).

Integration of the regulator attack in PyCIGAR has not yet been implemented, but it should follow the same methodology employed for the other attacks. The existing pyCIGAR regulator class will be used as a base to create attacked regulator classes. Change in regulator behavior corresponding to the attack will be implemented by encapsulating the corresponding OpenDSS API command employed in the OpenDSS tests described above.

Capbank Attack

This attack is similar to the load/DER connect/disconnect type attack. Here instead of loads or DERs, we explicitly target capacitor banks or other VAR compensation devices. One could disconnect individual phases of such devices to worsen voltage imbalance. Alternately, an attacker could also connect/disconnect three-phase devices to degrade voltage. Figure 19 shows the degraded voltage after disconnecting all capacitor banks on the IEEE 123 bus feeder. Such an attack could be used in conjunction with any of the previous attacks to cause severe voltage violations.

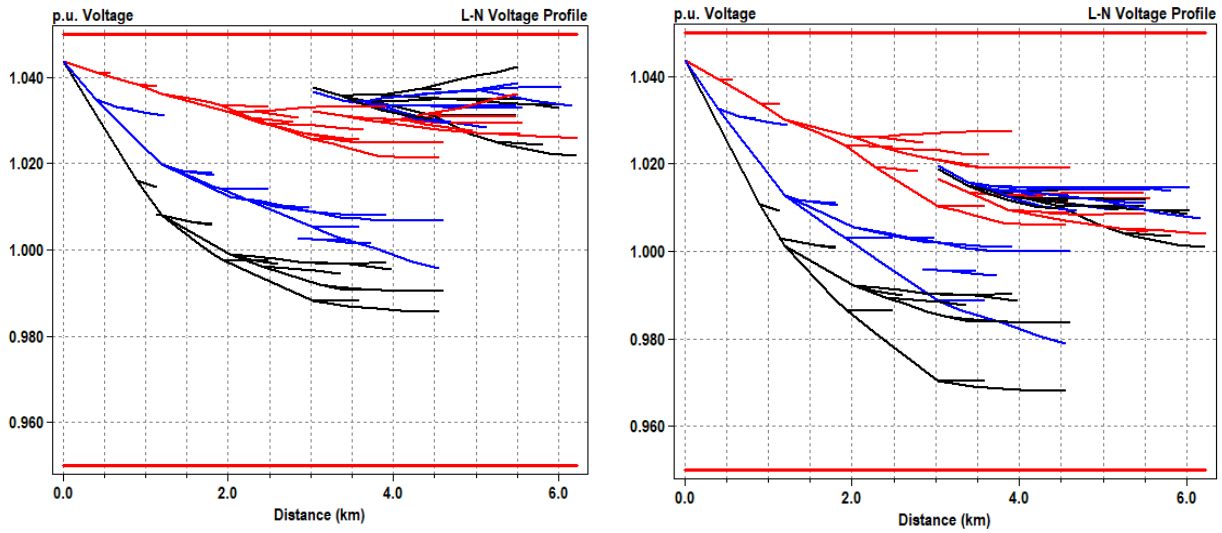

Figure 19: Comparison of voltage profiles with default settings of capbanks (left) and with capbanks disabled (right). This test was performed on the IEEE 123 bus network.

This attack was not yet integrated in PyCIGAR but its integration should follow the same methodology employed for the other attacks. PyCIGAR currently has no implementation of capacitor banks. Additional discussions must be performed for definition of how to better encapsulate the behavior required for performing the attack.

Energy Storage Attack

This attack is also similar to the load/DER disconnect type attack. However, instead of explicitly connecting/disconnecting a device here we manipulate active and reactive power setpoints sent to the storage device. These setpoints could be 0, in which case this would be equivalent to disconnecting a storage device. Such an attack is targeted towards use cases where an energy storage device may be needed at a later point in the day. For instance, a storage device may be needed at the substation transformer for peak shaving to avoid overloading conditions. Subtask 3.1 report presents an analysis of possible related attacks. If the battery is simply disconnected it may be brought back online for the peak shaving operation. However, if the battery is provided malicious setpoints to drain it completely and/or spoof the measurements for the centralized controller, then it may not be available for peak shaving. This would then lead to overloading affect the PDD KPI.

Currently, additional energy storage controllers are being defined in PyCIGAR by other team members for implementation of operating scenarios which are relevant for the project. Future plans of the red team include the implementation of more elaborate attacks aiming at those specific operating scenarios. A preliminary analysis of those scenarios and corresponding attacks was performed in Subtask 3.1.

The integration of this energy storage setpoint manipulation attack in PyCIGAR is under development. The base of the hacked classes has already been implemented in the Red Team Addon. Additional discussion will be required for extending the attacks to specific operating scenarios. In PyCIGAR the battery controllers have different capabilities depending on whether they are distributed or centralized. Although

the basic exploitation scenarios apply to both of these cases, different implementations may be needed for those cases depending on the specific operating scenarios.

Attack Budget

The proposed approach for defining budgets for the attacker and associating costs to the attacks is explained in detail in Subtask 3.1 report. The purpose of this approach is to limit the reach of the adversary in a realistic way, while providing flexibility for definition of a large variety of attacks which may include combinations of multiple methods and targets. Budgets may be used for controlling how large an attack will be, and the red team may employ domain expertise or optimization methods to choose among the best attack options, i.e. attacks which will achieve greatest impact on defined KPIs, respecting such constraint.

Real world attacks usually comprise multiple steps. In order to analyze or define such steps, besides the information of the effort/resources required to attack a certain target, it is also very important to understand how the computational devices are interconnected in computer networks. Currently, PyCIGAR does not support a way to define computer network and associated properties for individual devices. Therefore, subtask 3.2 also included the definition and development of means for integrating this information to PyCIGAR power system models.

The NetJSON format was chosen for this task, as it consists of a powerful and flexible yet lightweight way of encoding computer network information:

"*NetJSON is a data interchange format based on JavaScript Object Notation (JSON) designed to describe the basic building blocks of layer2 and layer3 networking. It defines several types of JSON objects and the manner in which they are combined to represent a network: configuration of devices, monitoring data, network topology and routing information.*"[1].

Therefore, NetJSON can be used to create a simple overlay of computer network information over the existing power system, which comprises the computational devices which are part of the power system, such as controllers, and additional computer network specific devices that do not directly play a role in the power system operation, such as routers or firewalls. In terms of implementation, additional input files are defined to contain:

- information of computer network specific devices.
- relevant properties associated to all computational devices in the network (including the power system ones).
- information about how all computational devices are interconnected.

Considering the properties of power system devices, the current implementation comprises the definition of default values for such properties within the corresponding PyCIGAR class definition so that those they can be transparently applied to all devices of the same type. However, the default values can be overwritten to be made specific to each component by means of the input files.

Processing of input information to create the NetJSON representation was integrated into the parser used to create simulation configurations for PyCIGAR. NetJSON specification allows flexibility for associating

[1] https://netjson.org/docs/what.html

any number or type of properties to each device. Those properties will be used to store information about the possible attacks and associated costs and relevant computer network information, such as firewall rules, for instance.

Figure 20 presents sample input files defining computer network devices (left) and interconnections between them (right).

```
device,property_dict
controlcenter,{"type":"control_center"}
network_switch1,{"type":"switch"}
network_ids1,{"type":"ids"}
network_firewall1,{"type":"firewall"}
network_switch2,{"type":"switch"}
network_firewall2,{"type":"firewall"}
network_switch3,{"type":"switch"}
network_firewall3,{"type":"firewall"}
ntp_clock1,{"type":"ntp_clock"}
hackme_wifi,{"type":"wifi_router"}
```

```
device_a,device_b,property_dict
inverter_s701a,network_switch1,{"type":"wired"}
network_switch1,network_ids1,{"type":"wired"}
network_ids1,network_firewall1,{"type":"wired"}
network_firewall1,controlcenter,{"type":"wired"}
inverter_s702a,network_switch2,{"type":"wired"}
network_switch2,network_firewall2,{"type":"wired"}
network_firewall2,controlcenter,{"type":"wired"}
inverter_s703a,network_switch3,{"type":"wired"}
network_switch3,network_firewall3,{"type":"wired"}
network_firewall3,controlcenter,{"type":"wired"}
ntp_clock1,network_switch1,{"type":"wired"}
hackme_wifi,network_switch1,{"type":"wireless"}
```

Figure 20 - Sample input files for definition of computer network information. In real applications, device properties will include additional information such as attack costs.

Figure 21 presents a simplified sample output in NetJSON format.

```
{
  "type": "NetworkGraph",
  "label": "Devices",
  "protocol": "static",
  "version": null,
  "metric": null,
  "nodes": [
    {
      "id": "inverter_s701a",
      "properties": {
        "type": "pv_device"
      }
    },
    {
      "id": "inverter_s702a",
      "properties": {
        "type": "pv_device"
      }
    },
    {
      "id": "inverter_s703a",
      "properties": {
        "type": "pv_device"
      }
    },
    {
      "id": "controlcenter",
      "properties": {
        "type": "control_center"
      }
    }, … ],
  "links": [
    {
      "source": "inverter_s701a",
      "target": "network_switch1",
      "properties": {
        "type": "wired"
      }
    },
    {
      "source": "network_switch1",
      "target": "network_ids1",
      "properties": {
        "type": "wired"
      }
    },
    {
      "source": "network_ids1",
      "target": "network_firewall1",
      "properties": {
        "type": "wired"
      }
    },
    {
      "source": "network_firewall1",
      "target": "controlcenter",
      "properties": {
        "type": "wired"
      }
    }, … ]}
```


Figure 21 – sample excerpt of NetJSON representation of the computer network.

Once the computer network information is represented in NetJSON format, it is possible to perform various types of analysis and visualization. Figure 22 presents a graph depicting a sample NetJSON generated using the developed implementation. Nodes and edges correspond respectively to computational devices and their interconnections.

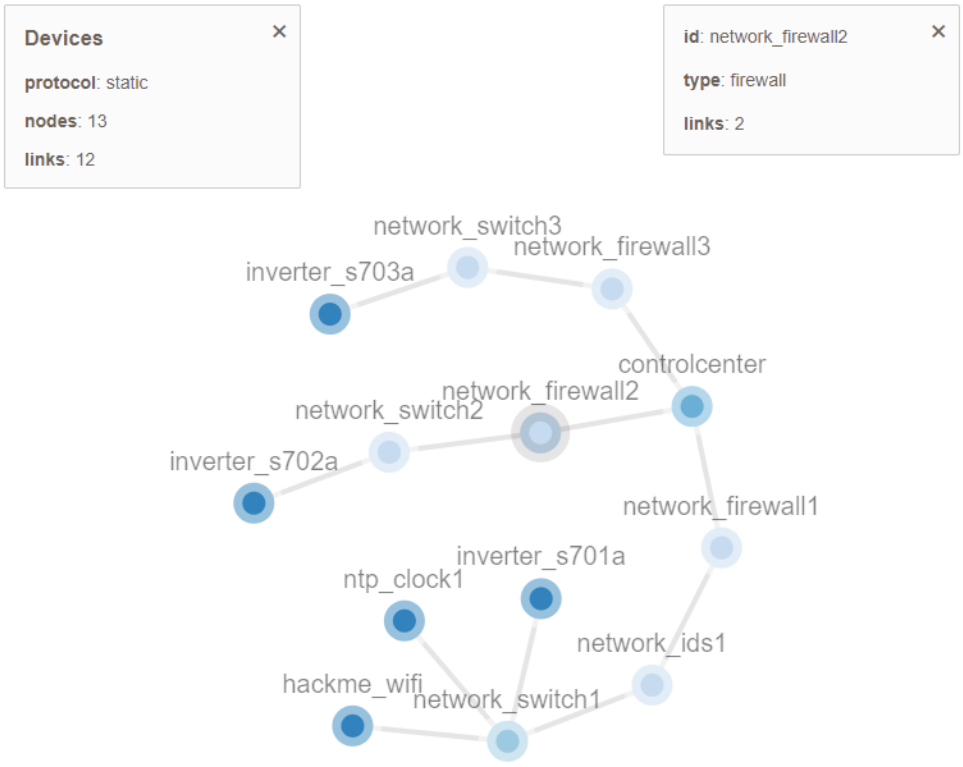

Figure 22 – Graph visualization of sample NetJSON generated using the developed implementation.

Conclusion and Future Work

This report presented the work developed by Siemens Technology (red team) corresponding to Subtask 3.2 in the SPADES project. Preliminary implementations of attacks have been created in OpenDSS and some of them also included a prototype implementation in PyCIGAR. Those implementations and tests performed based on them have been employed for obtaining a better understanding and validation of the consequences of proposed attacks and also for evaluating the best approaches for implementing the associated functionality in PyCIGAR.

The work developed in Subtask 3.2 also included the development of means for representing an overlay of computer network information on top of PyCIGAR power system models. This representation will be employed for evaluating and defining attack vectors subject to constraints imposed by attack budgets

available to the attackers and attack costs corresponding to each action taken during an attack. Consideration of the computer network information and corresponding attack budgets/costs is expected to result in the definition of more realistic attack vectors compared to the same definitions based on the power system alone, contributing to the real-world applicability of the results achieved in the project.

Next steps planned for the red team in the project, besides supporting implementation of attacks in PyCIGAR, comprise the beginning of tests based on the final implementations of attacks in PyCIGAR, including the optimization of such attacks and the evaluation of attack costs. Concerning the last item, the red team will work with LBNL and NRECA for definition of proper computer network information associated to the power systems under analysis.

REPORT Subtasks 3.3, 3.4, 3.5, 3.6

Deliverables 3.3.1, 3.4.1, 3.5.1, 3.6.1

Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES)

DOE CESER

CEDS Program

SUBMITTED BY

Siemens Corporation Technology

755 College Rd East, Princeton NJ

Submitted: Nov. 17th, 2022

Technical Point of Contact

Dr. Bruno Leao - bruno.leao@siemens.com

Project Manager

Ramamani Ramaraj - ramamani.ramaraj@siemens.com

SUBMITTED TO

Lawrence Berkeley National Laboratory

Contents

Introduction

This report describes the developments performed and results obtained by the Red Team (Siemens Technology) during the period ranging from January/2022 to November/2022 as part of the Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES) project. During this period, the focus of the Red Team was in developing, implementing, and testing an optimization solution for designing cyberattacks to disrupt key performance indicators (KPIs) that quantify how well a power distribution grid is operating. Such design of cyberattacks takes into consideration a variety of power system devices which can be targets of attacks and the characteristics of the computer network connecting those devices. The power system devices and corresponding attacks as well as the computer network characteristics are considered to the extent previously defined as part of the project and documented in previous Red Team reports [1][2]. The ultimate goal of this effort is to employ the developed optimization tool for designing cyberattacks for testing the AI-based countermeasures developed by other members of the project team. This provides a context for testing that is considerably broader and more realistic compared to previous tests of the solution in terms of the actions taken by the adversary to disrupt the power system operation.

In order to achieve the optimization solution, five different building blocks had to be developed and integrated:

- **Power System and KPIs**: definition of a power system topology and corresponding implementation in OpenDSS/PyCIGAR; definition of KPIs and means to quantify them based on outcomes of the model simulations in PyCIGAR.
- **Attack implementation in PyCIGAR**: modeling of all adversary actions which may affect the power system in PyCIGAR.
- **Computer network topology**: definition of the computer network topology which connects the devices from the power system of interest and representation of such computer network in a format that can be properly employed for the optimization task.
- **Cyberattacks and costs**: definition of possible actions an adversary can take when he has control over each of the devices or communication links in the computer network; each of those actions must be associated to a "cost" value which quantifies the effort associated with performing it.
- **Attack optimization methodology**: definition of the method for combining the above information and obtaining a sequence of steps an adversary may take to maximize the disruption of a certain power system KPI given a pre-defined attack "budget".

The development and results associated to each of those building blocks and their integration are presented in detail in the following sections.

Power System and KPIs

This section describes the power system that was custom built in OpenDSS and used for testing various attacks on system components (switches, DERs, controllable loads, regulators, capacitor banks, and energy storage devices) through PyCIGAR. In addition, we discuss the identified KPIs that were used to quantify the success of attacks on the network.

Our primary focus was on creating a realistic network that allows us to explore system level attacks which adversely affect the system behavior in the short (order of seconds) to medium term (order of minutes). These attacks can be a result of a single compromised component or a coordinated attack on multiple

components that leads to undesirable network behavior. After discussing with NRECA and LBNL we chose to modify the standard IEEE 123 bus network for our case studies; see Figure 1 - Modified IEEE 123 multi-feeder topology. Feeder 1 is shown above and Feeder 2 is setup to mirror Feeder 1. The color-coded boxes show the location of the inter-feeder ties that are normally open.

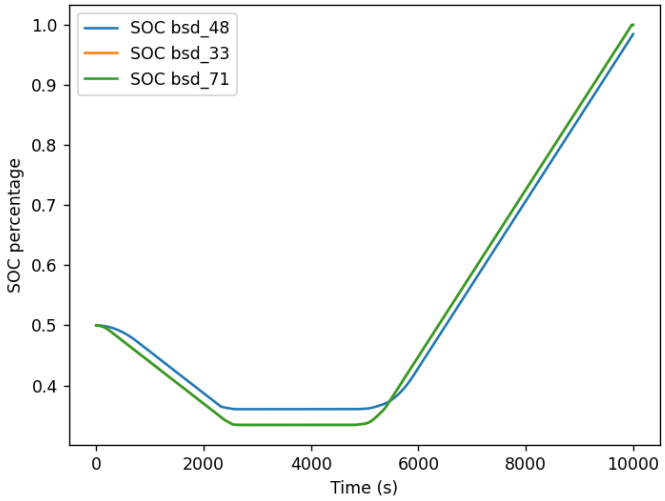

Figure 2 - State of Charge (SOC) for BESS at buses 33, 48, and 71 over the entire time series of load and PV data. Each battery is equipped with a centralized valley filling and peak shaving controller that kicks in when specified thresholds are exceeded. The BESS neither charges nor discharges during periods where the substation active power is between the valley filling and peak shaving thresholds or if minimum/maximum allowed SOC is reached. All batteries are initialized here with a 50% SOC.

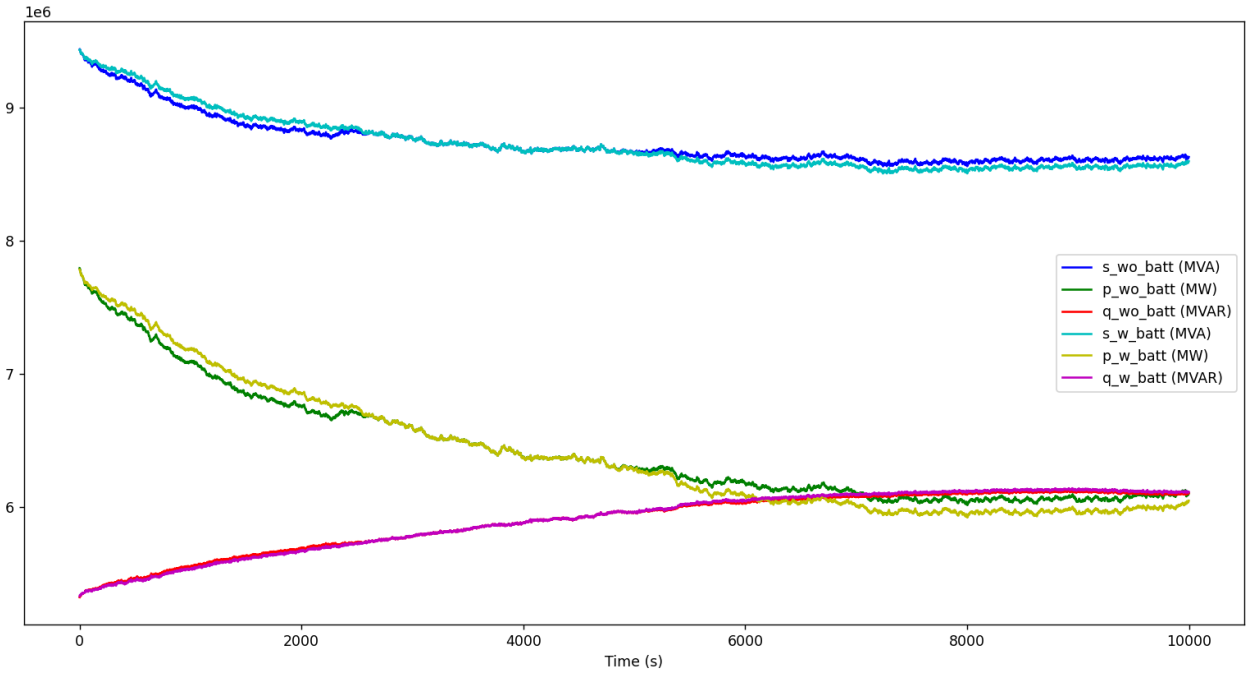

Figure 1. The IEEE 123 bus feeder contains four voltage regulators, shunt capacitor banks, and multiple switches. The main modifications were as follows:

- First to capture a realistic distribution substation we included one additional feeder at the substation. The additional feeder was setup to mirror the standard IEEE 123 bus feeder[1]; see Figure 1
- Second, we added normally open ties between the two feeders – one between buses 152 in feeder 1 and 2, one between buses 300 in feeder 1 and 2, and one between buses 135 in feeder 1 and 2; see Figure 1
- Third we added PV (time-series) on almost every node to simulate a feeder with high penetration of renewables. The time-series covered second based data for a 4-hour window from approximately 9:00 a.m. – 1:00 p.m. For testing, the most interesting times of the day are in the morning 9:00 a.m. – 10:00 a.m. when loading is high and between 12:00 p.m.-1:00 p.m. when PV output is high.
- Fourth we added three large battery energy storage systems (BESS) that were randomly dispersed in the network on buses 33, 48 and 71 in feeders 1 and 2. The control logic was setup to perform peak shaving and valley filling functions from a centralized controller that was monitoring the substation active power in-feed. The valley filling function allows the BESS to charge when the net active power at the substation is below a specified threshold. Similarly, the peak shaving function was setup to discharge the BESS when the net active power at the substation crossed above a specified threshold. We performed extensive simulations to set the right values for valley filling and peak shaving based on the pre-defined load and PV profiles. In particular, the thresholds were set as 2600 kW for valley filling and 3000 kW for peak shaving. Note that in between the two threshold values [2600-3000] kW, the BESS is neither charging nor discharging; see Figure 2 and Figure 3.
- Fifth we scaled the PV and load, respectively by 1.375 and 1 to simulate medium loading conditions. These scaling factors were chosen to allow inter-feeder topology reconfiguration over the entire time-series PV and load data without overloading the substation transformer, i.e., if the topology were to be reconfigured to bring Feeder 2 onto Feeder 1, then the substation would be heavily loaded, yet it would not cause overloading or severe undervoltage issues; see Figure 3.
- Lastly, we enabled time control mode in OpenDSS and validated against PyCIGAR so that we can capture realistic local control actions from components such as regulators and capacitor banks.

[1] https://cmte.ieee.org/pes-testfeeders/resources/

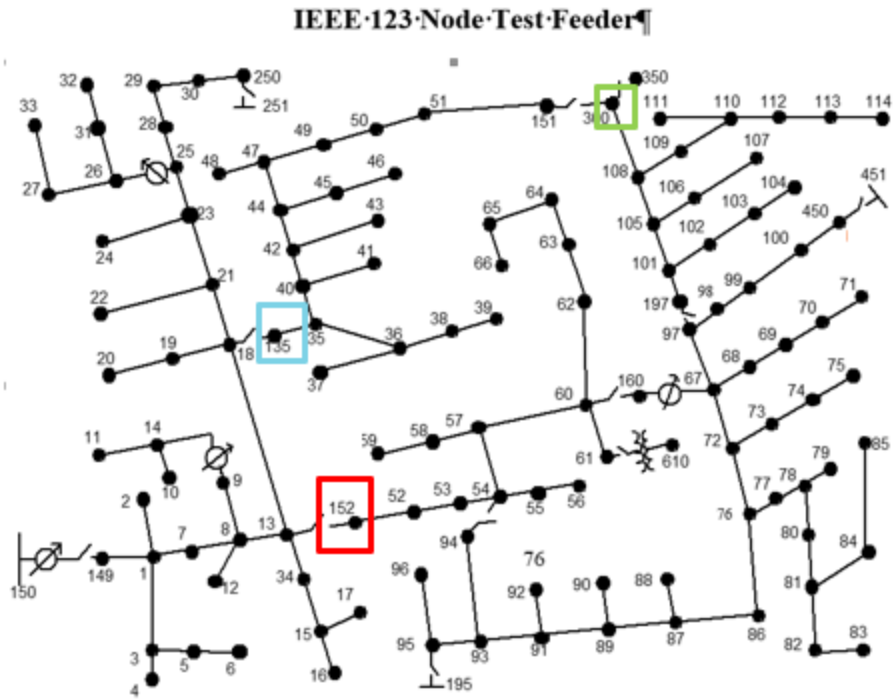

Figure 1 - Modified IEEE 123 multi-feeder topology. Feeder 1 is shown above and Feeder 2 is setup to mirror Feeder 1. The color-coded boxes show the location of the inter-feeder ties that are normally open.

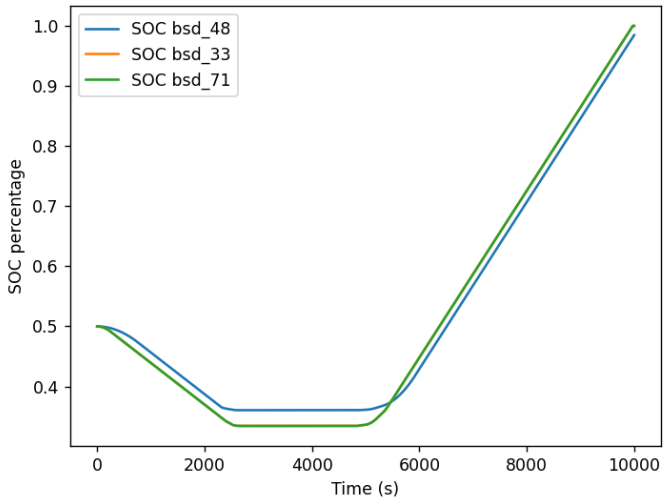

Figure 2 - State of Charge (SOC) for BESS at buses 33, 48, and 71 over the entire time series of load and PV data. Each battery is equipped with a centralized valley filling and peak shaving controller that kicks in when specified thresholds are exceeded. The BESS neither charges nor discharges during periods where the substation active power is between the valley filling and peak shaving thresholds or if minimum/maximum allowed SOC is reached. All batteries are initialized here with a 50% SOC.

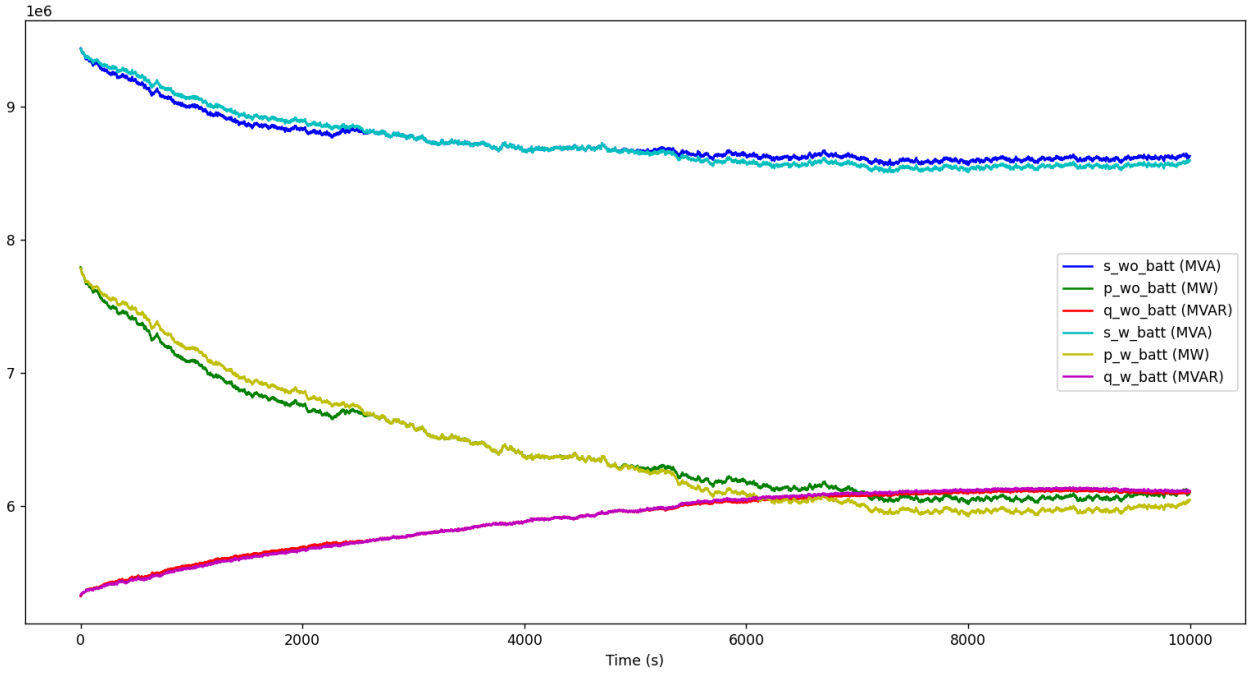

Figure 3 - The apparent power (s_wo_batt), active power (p_wo_batt), and reactive power (q_wo_batt) at the substation (feeder 1 + feeder 2) without BESS and the apparent power (s_w_batt), active power (p_w_batt), and reactive power (q_w_batt) at the substation (feeder 1 + feeder 2) with BESS. The effect of the BESS charging and discharging cycles can be clearly seen on the net power in-feed at the substation; see also Figure 2

KPIs for definition of successful attacks are described in detail in previous reports [1] and have since been updated based on discussions with NRECA and LBNL. Although some of these were only qualitatively defined before, we now have quantitative definitions for the key Tier 1 KPIs below:

- Tier 1
  - Voltage imbalance (VI)
    - $VI=\max[\max[(\text{abs}(Va\text{-}Vm), \text{abs}(Vb\text{-}Vm), \text{abs}(Vc\text{-}Vm)]/Vm]$
    - Here Va, Vb, and Vc are the line-to-neutral voltage magnitudes for each phase on a given bus and $Vm = (Va+Vb+Vc)/3$. The inner max function is the maximum across all phases on a given bus, whereas the outer max function maximum across all buses in a network.
  - Substation power factor (SPF)
    - $SPF=\cos(\arctan(q_{network}/p_{network}))$, where $q_{network}$ and $p_{network}$ are respectively the net reactive and active power at the substation.
  - Combined VI+SPF KPI which is the initial choice for the optimization is defined as $(1\text{-}VI+SPF)/2$. They were combined so that the higher the KPI the higher the impact of the attack. Therefore this can be directly used as a goal function for attack optimization.

- Tier 2
  - Instability (oscillation)
  - Power delivery disruption (PDD), including
    - DER disconnection based on IEEE 1547 standard (PDD DER)
    - Disconnection due to overloading of lines/transformers

- Tier 3
    - Equipment useful life degradation
    - Power quality degradation (poor power factor or over/undervoltage conditions)

The Voltage Imbalance and Substation power factor KPIs were normalized to be on scale of [0,1]. Attacks that minimize these KPIs, i.e., give values closer to 0 are considered better than ones that give values closer to 1. The combined VI+SPF KPI is similarly normalized on a [0,1] scale. The SPF KPI was simply the absolute value of the measured PF at the substation. The VI KPI was setup to capture the worst-case deviation from the average across all phases at a given bus. See Figure 4 for tracking of the tier 1 KPI's over the first 10000 seconds of load and PV data.

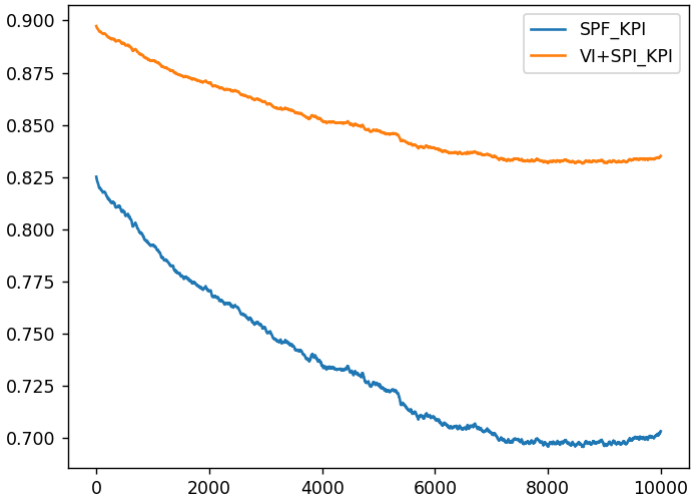

Figure 4 - SPF and the combine VI+SPF KPI over the first 10000 seconds of load and PV data

It was decided not to pursue the Instability (oscillation) KPI since it would require a dynamic simulation of the power system under study or require that the oscillation is constantly forced based on repeated connection/disconnection of system components or controllers. Dynamic simulation mode is weak in OpenDSS and all other KPIs can be easily captured using a quasi-steady-state simulation. Similarly, the PDD and PDD DER KPIs were not explicitly considered due to the need to analyze and implement protection. Adding the relevant components that can be appropriately coordinated would require a detailed protection study. However, we are considering adding overcurrent protection on inter-feeder tie switches so that it is possible to perform topology reconfiguration attacks that cause overloading on lines and transformers. To simulate this use case, we will scale the PV and load profiles by a factor of 2.75 and 2, respectively. Note that the scaling is done only to simulate a use case where an attacker can take control of multiple distribution tie switches to stack multiple feeders onto a substation feeder. A substation feeder may be designed to bring on one/two feeders without causing any overloading on lines or transformers. Otherwise, bringing on multiple feeders can lead to severe overloading that can then lead to severe under/over voltages issues causing protection to disconnect components or entire sections of the feeders. Based on NRECA feedback, Tier 3 KPIs were not explicitly considered.

PyCIGAR Attack Implementation

This section describes how the PyCIGAR Framework was enhanced to incorporate additional attacks, how the interaction between PyCIGAR and those attacks is implemented and what effects the attacks can have on an example network.

Attack Addon PyCIGAR Integration

PyCIGAR represents devices of the power network through a hierarchical approach. On the top level is the device itself that provides the power set points to OpenDSS. Devices are connected to at least one controller, which defines the controls (e.g.: operation mode, maximum ramps, and others) that are used to determine setpoints on device level.

During the period covered by this report, PyCIGAR was extended to include the following device classes:

- PV Inverter Device
- Switch Device
- Regulator Device
- Battery Storage Device
- Capacity Device

For each device there is a device class implementation as well as a controller class implementation. PyCIGAR devices may be controlled in one of two schemes. In the first control scheme, referred hereafter as local control, each device is associated to two controllers. One controls the device when it is not being attacked and the other one controls the device otherwise. In the second control scheme, referred hereafter as distributed control, one controller controls multiple devices and derives the required controls across all devices at the same time. Table 1 provides an overview of which device utilizes which control scheme.

Table 1 - Control schemes for devices in PyCIGAR

Device	Control scheme
PV Inverter Device	Local control
Battery Storage Device	Distributed control
Switch Device	
Capacity Device	
Regulator Device	

Depending on the control scheme the attack is performed differently. For the local control scheme the connection between controller and device is a 1-1 relation, therefore modification on the particular controller don't influence other devices. For the distributed control scheme the controller cannot be modified based on a single device because this would impact the other devices associated with it. For this control scheme the controller will be disassociated with the device which is then associated with an alternative controller, referred hereafter as hacked controller. This hacked controller contains the implementation of all attacks which can affect a certain device.

In order for attacks to be used within PyCIGAR while not interfering with PyCIGAR implementation it was decided that the Attack Addon will be reusing existing in PyCIGAR implemented controller and devices. The implemented classes of the AttackAddon inherit the functionality from the PyCIGAR classes and

extend them with aspects needed for the attack. The inheritance tree is depicted in Figure 5Figure 5. Through inheritance, AttackAddon classes can operate both as normal controller /device as well as controller/ devices that are hacked. Additionally, the AttackAddon stays independent from the existing PyCIGAR implementation. The classes implemented in by the red team are referred to as AttackAddon.

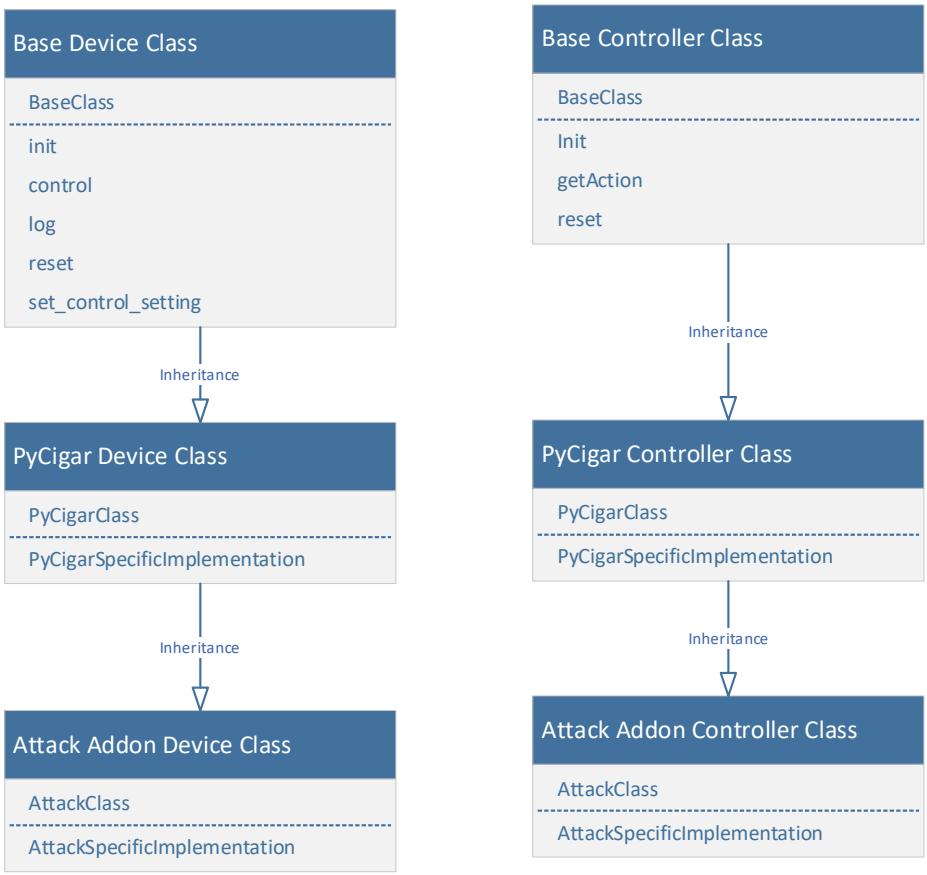

Figure 5 - Inheritance Tree of PyCIGAR AttackAddon

The creation and utilization of the readteam_addon_parser is discussed in detail in the previous report [2]. During the parsing process, in case of the utilization of the AttackAddon, the corresponding PyCIGAR device/controller classes are replaced by Attack Addon device/controller classes and every AttackAddon object will receive the information of whether or not the device will be attacked during the scenario simulation so that this information can be used to define its behavior.

Attack Types

As shown in Table 2, different attacks are implemented for different devices. The various attacks are discussed in detail below including their description, parameters, and testing.

Attacks are defined based on a set of parameters. The utilization of attack parameters as well as their impact on the attack were first introduced in the previous report [2].

Testing of the attacks is performed utilizing a PyCIGAR model based on the IEEE 3 bus network. The model was adapted to include each device under test so that the impact of the different attacks could be assessed.

Table 2 - Device attack overview

Device	Attack	Description
PV Inverter Device	Connect/Disconnect	Disconnects the PV Device for the specified time
	VoltageBreakPoints	Modifies the Volt/Var and Volt/Watt behavior
	Unbalanced	Creates an unbalanced output across the 3 phases
Battery Storage Device	OperationMode	Modifies the operation mode of the battery (e.g.: charge instead of discharge)
	PowerInjection	Forces to discharge the battery with maximum power
	PowerConsumption	Forces to charge the battery with maximum power
	Battery Settings	Modifies the control parameters of the battery (e.g.: reduce the max ramp rate)
Switch Device	Open/Close (Topology)	Changes the topology of the grid by operating two switches that create a new one
Capacitor Device	Curtailment	Reduces the capacity of the capacitor bank
Regulator device	Change Settings	Modifies the regulator settings to create unintentional regulator behavior
	RegulatorDeactivate	Prohibits the regulator from executing controls and fixes is to a specific tap
	RegulatorProhibitControl	Prohibits the regulator from executing controls
	ChangeTaps	Changes the regulators tap

Attack Test Scenario

The IEEE3 network was used for testing the developed attacks. The configuration of the system is depicted in Figure 6. The components contain different load profiles autogenerated from PyCIGAR with Table 3 showing the initial conditions employed for simulations.

Table 3 - Test scenario configuration

Scenario component	sizing
Transformer	2500 kVA
Loads 701 / 702	150 kW
PV 701 / 702	100 kW
Battery 701 / 702	10 kWh
Loads 703	400 kW
PV 703	300 kW
Battery 703	100 kWh

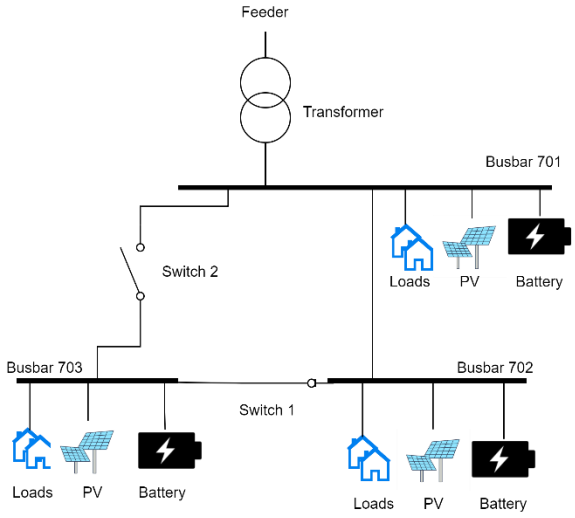

Figure 6 - IEEE-3 AttackAddon test network

PV Inverter Device

The PV Inverter control scheme is based on voltage break points. The control measures the voltage at the busbar and compares it to preconfigured voltages. Based on this comparison it is decided how the inverter operates. This control scheme (Figure 7) Figure 7models a representation of Volt/Var as well as Volt/Watt behavior as it is used to control Inverter and it previous reports and as shown in Figure 8.

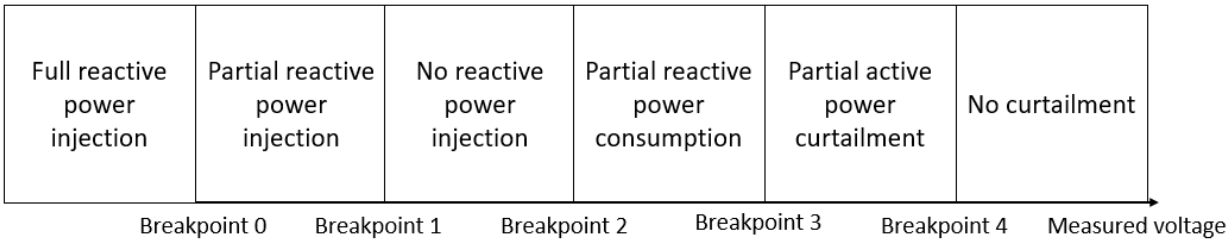

Figure 7 - Voltage Break Points control scheme

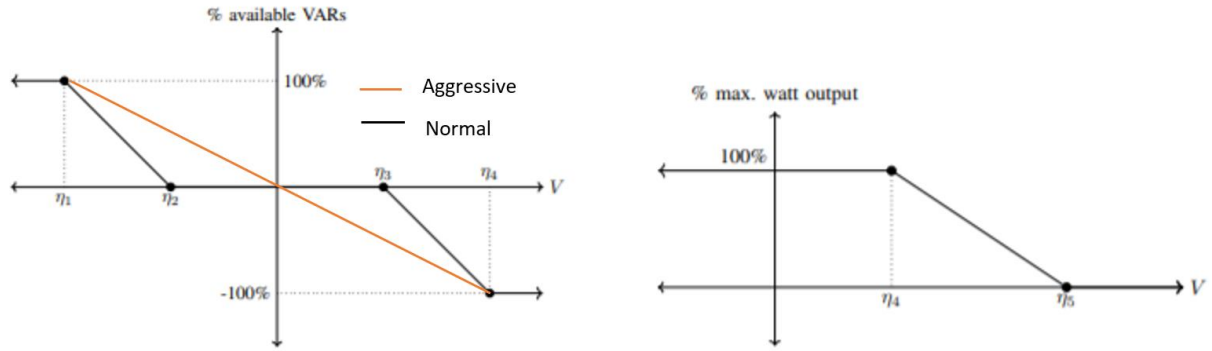

Figure 8 - Volt/Var and Volt Watt behavior characteristics [1]

The VoltageBreakPoint attack modifies the configured breakpoints and therefore shifts the complete control scheme.

The PV Inverter Connect/Disconnect attack represents the disconnection or reconnection of PV devices during operation. During the disconnection of the PV device, the device will not participate in any control scheme neither will it provide any power to the grid.

Additional to the discussed PV Inverter attack an unbalance attack is implemented to create an unbalanced power injection across the 3-phases.

Tests and results

For the PV attacks the impact of the different attacks are provided in Figure 9. The power output at the substation is measured and the impact of the attacks are compared by running the same simulation first without attack (reference scenario) and afterwards with the different attacks at different timeslots (attack scenario)

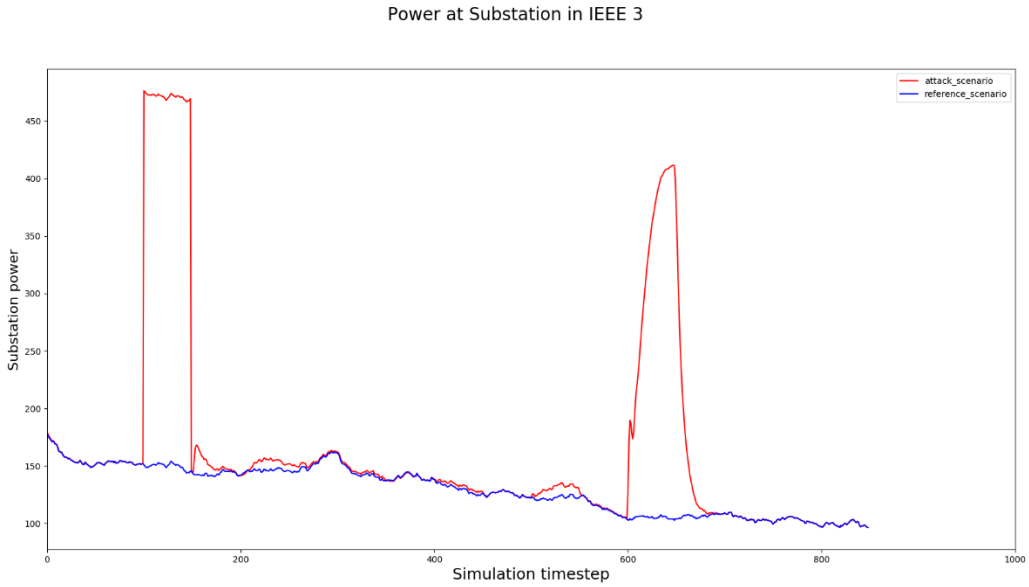

Figure 9 - Test results for Connect/Disconnect attack and PV VoltageBreakPoint attacks

For the test run the following attacks were executed:

Attack	Timestep	Component
Connect.Disconnect	100 - 150	PV 703
VoltageBreakPoint: MaxReactive Power	200 - 250	
VoltageBreakPoint: partial Reactive	300 - 350	
VoltageBreakPoint no Reactive	400 – 450	
VoltageBreakPoint: Reactive Usage	500 – 550	
VoltageBreakPoint NoActive	600 - 650	

Battery Storage Device

The battery storage device has various parameters that can be modified with resulting impact in its operation. Attacks on parameters that can be modified are captured in the Battery Settings attack. Another attack type consists of changing the operating mode of the battery. These attacks are captured in the Operation Mode attack. Table 4 presents the parameters associated to each attack. MaxCharge and MaxDischarge power temper with the battery settings by reducing the maximum rating to 20% of the original rating. Max Ramp rate allows the shorten the time till a battery is charging or discharging with full power.

Table 4 - Battery setting and Operation Mode attacks

Attack	Attacked parameters
Battery Settings	MaxCharge Power, MaxDischarge Power, Max Ramp rate
Operation Mode	Mode of operation (charge, discharge, standby)

Additional to the Battery Settings and Operation Mode attacks, Power Injection and Power Consumption was also implemented. These attacks correspond to an operating condition which is not a normal state of the system, therefore it is more than a change in operating mode. They result respectively in discharging or charging of the corresponding batteries at maximum possible rate. Those can exceed the regular battery operation limits.

Tests and results

Per default batteries in PyCIGAR operate in a min max cycle, meaning they charge and discharge depending on their state of charge (SoC). For the reference simulation the batteries are all operating in charging mode. Through the simulation an attack is executed on battery connected to Bus 701 of the system that switches the battery from standby to charging to discharging operation. Figure 10 shows the impact of the operation change on the voltage. The yellow lines correspond to voltages from the reference scenario. In the initial time steps of Figure 10 the voltage of the reference scenario is lower than in the attack simulation. The battery is operating in charging mode for the reference scenario whereas it is operating in standby mode for the attack scenario. After the attack scenario switches the battery to charging mode the voltages are almost identical. At timestep 200 the battery is switched to discharging mode which results in an increased voltage across all busses within the network. Figure 11

shows the power provided to the system. When the battery is in standby mode or discharging the load on the system is lower than with a charging battery.

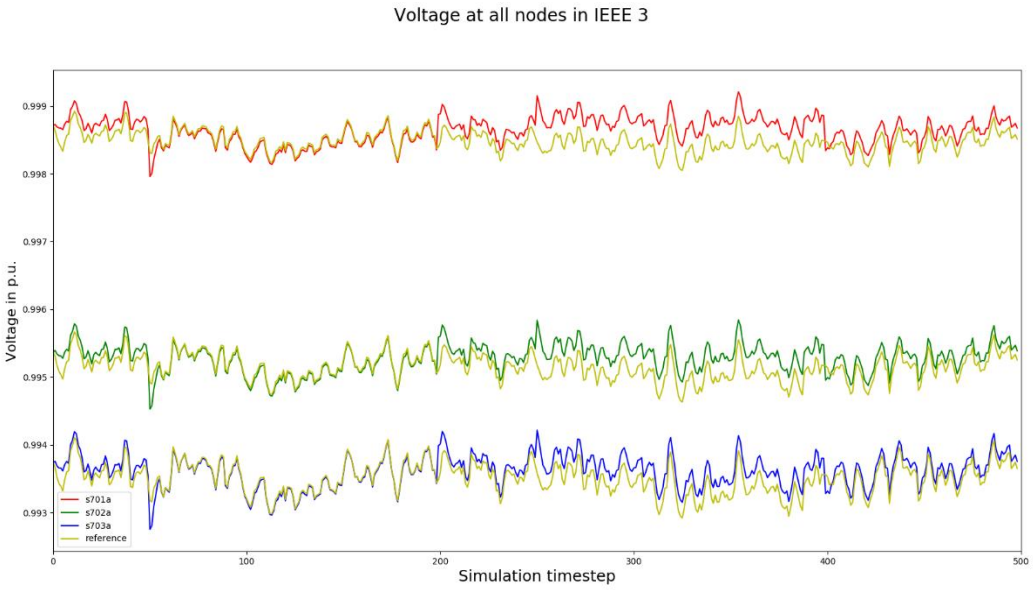

Figure 10 - Voltage at all nodes for Battery attack

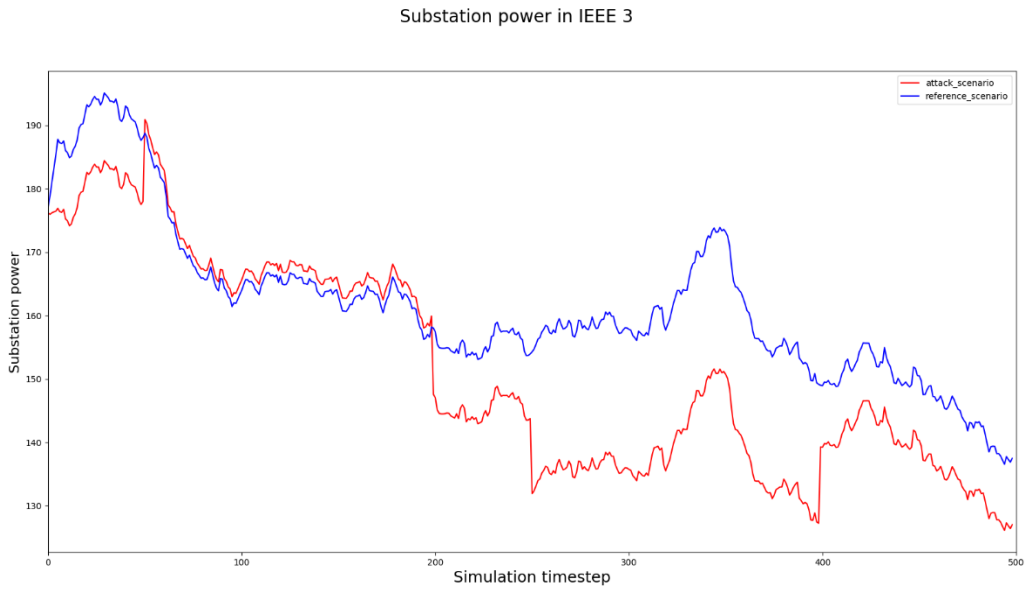

Figure 11 - Power at Substation for Battery attacks

Attacks on the battery settings depend on the scenario set up and sizing of the battery. The impact of battery setting attacks is limited in the scenario employed for testing. The system jumps to the maximum charging or discharging rate within 3 timesteps and a limitation of the maximum output power or power consumption would rather limit the impact on the grid.

Switch Device

To change the topology of a running simulation, switches were introduced. Switch devices can represent both breakers and switches in power grid. Based on discussions with LBNL, islanding of parts of the grid is not within the scope of the attacks we would like to consider during testing. To prevent this behavior, switches always need to be paired together so that operating one switch automatically results in an operation of the paired switches to ensure a valid topology. The result of this concept is that through switching attacks, topologies within a scenario change and therefore the power quality across the network. As the control scheme for this attack is a distributed control scheme, a new controller will be generated for the attack and associated with the correspondent devices that will be attacked.

Tests and results

For testing the attack on the provided IEEE3 network the Switch 1 and 2 in Figure 6 are operated at the same simulation step to change the topology of the network. Figure 12 shows that for the time of the attack the voltage is increased. Through the attack bus 703 is directly connected to bus 701 whereas bus 702 does not have a connection to bus 703 anymore. Through this reconfiguration the voltages in the network are shifted and, in this case, they increase at both locations.

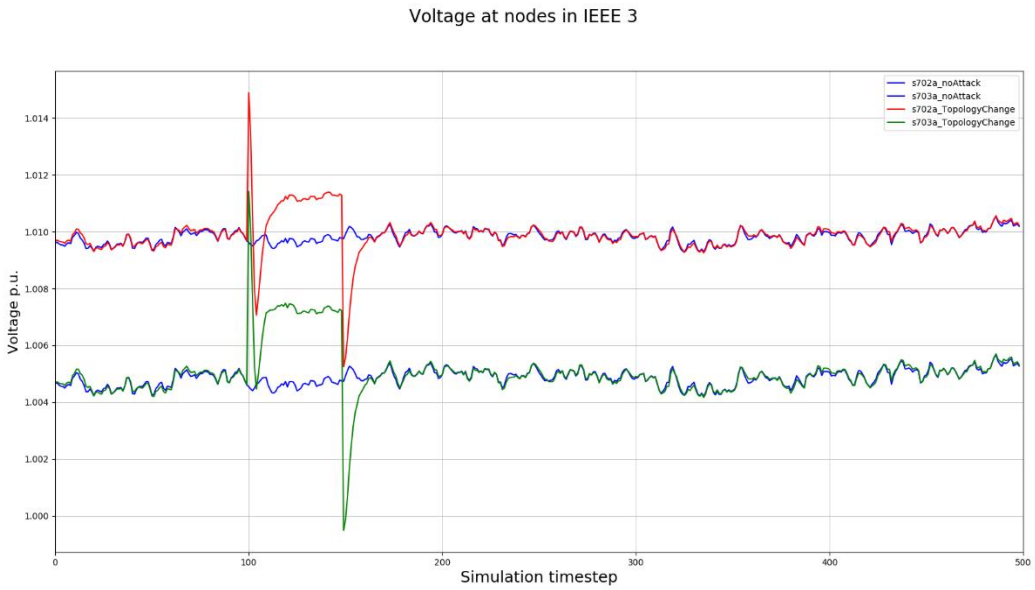

Figure 12 - Switch of the topology of the grid

Capacitor Device

Capacitors are deployed to improve the voltage in the grid. They can be controlled to connect more or less capacitance for this purpose. Attacks to the capacitor device focus on this functionality therefore impacting grid voltage.

Tests and results

Capacitor attacks are highly dependent on the sizing of the capacitor. Originally no capacitors are deployed for the IEEE 3 network. Therefore, the IEEE 123 network was used to test the functionality. The attack limits the capacitor reactive power output to 1% of its original value. Figure 13 shows that impact of the limitation of the capacitor at node 88 on the grid. The capacitor was originally providing 50kVAr

and after the attack provides only 0.5 kVAr to the grid. Therefore, the voltage at node 88 drops. The attack can lead to under voltages at nodes that are attacked or to imbalances in the system if the capacitor is not connected to all three phases.

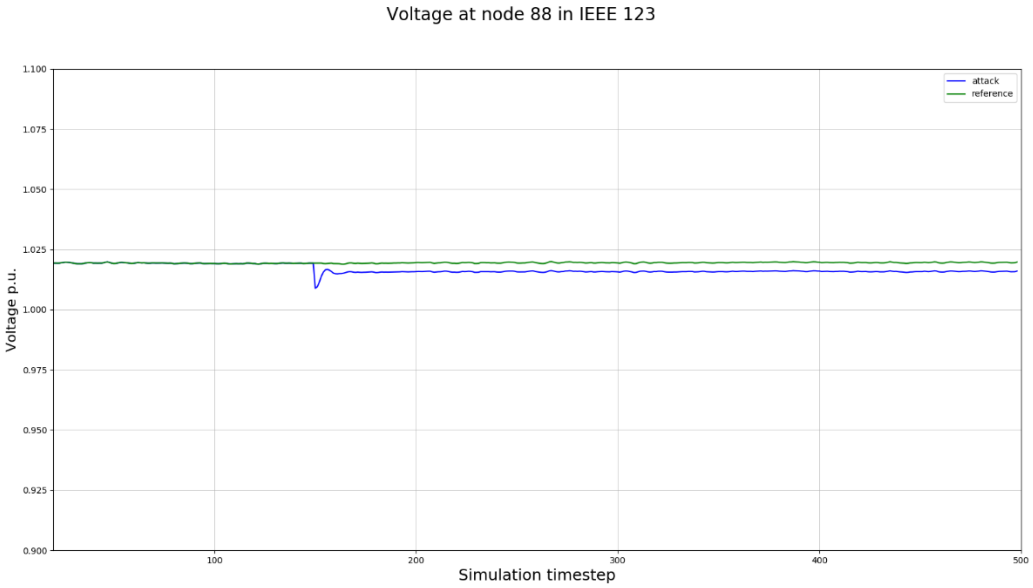

Figure 13 - Capacitor attack in IEEE 123 Node network

Regulator Device

Power transformer and line transformer devices are all represented by the regulator device class. Regulator devices' duty is to provide voltage quality assurance for the grid. Attacks on regulators are used to change the settings and therefore the operation characteristics of the regulator. Table 5 provides an overview of all implemented attacks which affect regulator devices.

Table 5 - Regulator attacks

Regulator attacks	Description
ChangeTaps: MaxTapNumber	Changes the tap to the highest available tap number of the regulator
ChangeTaps: MinTapNumber	Changes the tap to the lowest available tap number of the regulator
ChangeTaps: ZeroTapNumber	Changes the tap to the neutral tap
RegulatorDisconnect	Changes the tap to the neutral tap and does not allow it to be changed in the future
RegulatorProhibitControl	Leaves the tap as is and does not allow it to be changed in the future
ChangeSettings: TapDelay	Changes the delay time after which tap changes occur
ChangeSettings: IsReversible	Enables/Disables reverse power flow control of the regulator

Tests and results

Attacks to change the taps have been tested. Figure 14 shows the results of the simulation. In the first the tap is set to the minimum which resulted in a voltage decrease across the complete network. In the initial stage of the attack an oscillation is introduced due to the change of the tap. From timestep 200 to 300 the tap is changed to neutral. This attack corresponds to the reference scenario where the tap is in neutral for the complete simulation time. At timestep 300 the tap is set to the maximum. This results in another oscillation and an increased voltage across the simulation network. The attacks demonstrate that regulator attacks can have an impact of the voltage across the complete power grid.

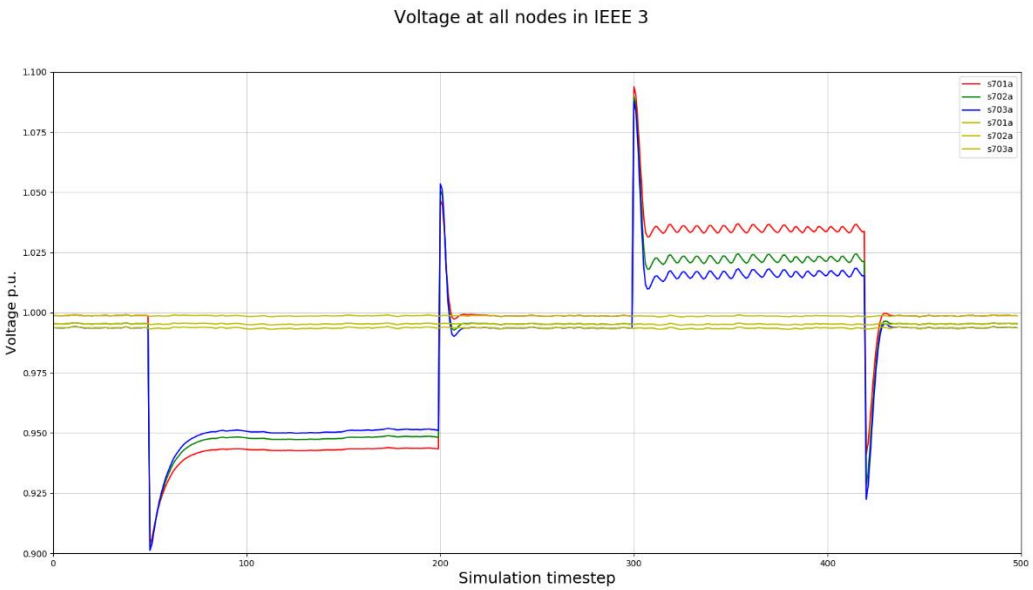

Figure 14 - Regulator attack to change Taps from Min to Zero to Max

The attacks RegulatorDisconnect, RegulatorProhibitControl and ChangeSettings do not have any impact on the simulation scenario. RegulatorDisconnect changes the tap to neutral position which is the operation position of the tap in the simulation. RegulatorProhibitControl prohibits the control of the taps from the moment of the attack and does not have an impact since the taps are not changed in the simulation. ChangeSettings are attacks that depends on the set up and control scenario of the simulation.

Effect of Attacks in Simulation

The various attacks shall be used to create scenarios where they disrupt power grid operation and AI-based countermeasures developed by other project team members are employed with the goal of keeping operation as close as possible to normal. To specify which attacks shall be employed at each scenario, additional information is specified as described in the sections Power System and KPIs, **Error! Reference source not found.**, and Computer Network Topology. Table 6 shows which KPIs are impacted by each attack (refer to KPI definitions in section Power System and KPIs). It also presents whether each one of them can be executed by means of network-based attacks or only through direct access to the device, which is taken into consideration during attack optimization. This information was defined based on discussions with the utility partner in the project (NRECA).

Table 6 - Connection between attacks, KPIs and cyberattack

Attack	Remote accessible	KPI
PV: Connect/Disconnect	Yes	PDD DER
PV: VoltageBreakPoints	Yes	PF, PDD (DER), VI, Instability
PV: Unbalanced	Yes	VI
Battery: OperationMode	Yes	PF, PDD (DER), Instability
Battery: PowerInjection	No	PF, PDD (DER), Instability, VI
Battery: PowerConsumption	No	PF, PDD (DER), Instability, VI
Battery: Battery Settings	Yes	PDD (DER), VI
Switch: Open/Close (Topology)	Yes	PDD (DER), VI, PF, Instability
Capacity: Curtailment	Yes	Instability, PDD (DER)
Regulator: ChangeTaps	Yes	PDD (DER), VI, Instability, VI (if single phase connected)
Regulator: Change Settings	Yes	PDD (DER), Instability, VI (if single phase connected)
Regulator: Regulator deactivate	Yes	PDD (DER), Instability, VI, (if single phase connected)
Regulator: RegulatorProhibitControl	Yes	PDD (DER), Instability, VI

Attack Optimization Methodology and Testing

The starting point of the attack optimization methodology is the hierarchy of possible adversary actions previously defined by the Red Team [1]:

1. Exposure: Actions associated with getting access to a certain device. This category comprises both entry points for the attacks and actions such as lateral movement where an attacker gets access to a device from another one. For entry point cases this may comprise factors such as physical security. Once an exposure action is performed on a certain device, the attacker has access to perform exploits as described below.
2. Exploitability: Actions related to exploiting the system to be able to alter its behavior. Exploits in this context can give the adversary access to other connected devices or to end effect actions as described below.
3. End Effect (previously referred to as “Attack”): Those are actions which can be performed by the adversary which have impact on the power grid operation. Such actions have a corresponding counterpart in the power grid simulation model employed for optimization which enables the evaluation of the impact of the attack.

This hierarchy was also extended to consider network-based attacks. In this case, the hierarchy of possible actions is simplified to comprise only categories #1 and #3, assuming that once the adversary has access to a network communication link he can already perform end effect actions through mechanisms such as man-in-the-middle, false data injection and flooding. Another concept which is relevant to the consideration of network-based attacks is the differentiation between physical links and logical links. Figure 15 presents an illustration of four devices (control center, router, regulator1 and inverter3) and their interconnections in terms of physical links (solid lines) and logical links (dashed arrows). Physical links as the name states, correspond to the physical network connections. Logical links

connect devices which exchange information that is relevant for the power systems operation. In the figure, both *regulator1* and *inverter3* exchange information with the *control center*, therefore they are connected by means of logical links. These communications are performed through *router*, so there are physical links which correspond to those connections. *Router* is not a producer or consumer of information in this context, therefore it is not included in logical links. Each physical link may be associated to multiple logical links and each logical link may be associated to multiple physical links. Attack costs are based on the physical link, e.g. the effort required to get physical access to a certain connection will depend on the physical link characteristics, such as location. Attacks on the other hand are defined based on logical links, e.g. a network-based attack affecting the communication between the *regulator1* and *control center* (logical link) could happen either in the physical link between *regulator1* and *router* or between *control center* and *router* and the effect of the attack should be the same in both cases.

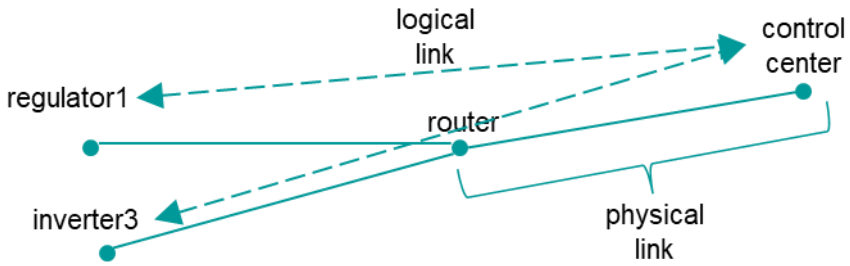

Figure 15 - Illustration of physical and logical network links

Another important building block is the cyberattack information layer, corresponding to the use of NetJSON format to represent computer network information associated to the power grid as previously defined by the Red Team [2]. This includes computer network devices and their interconnection and also information about all possible action an attacker can take associated to each device and network link in each category as presented above. The main information associated to each action is its cost which corresponds to a numerical value that quantifies the effort associated to that action. For each analysis an attack budget is defined as the limit of cumulative effort an adversary can employ during an attack. This is employed as a constraint during optimization.

The diagram in Figure 16 presents a high-level view of the optimization process. An AI-based sequential decision-making optimization solver based on the Monte Carlo Tree Search (MCTS) method interacts with the cyberattack information layer to identify potential actions the adversary can take and runs the power grid simulation in PyCIGAR to identify what the consequences of the adversary actions are in terms of impact to selected key performance indicators (KPIs). A certain KPI or set of KPIs must be selected a priori so that the optimization is performed in order to maximize their disruption. Multiple analyses can be performed considering different KPIs. Another relevant input for the optimization is an attack budget corresponding to the total effort which can be performed during the attack. The attack ends when the budget ends. Multiple budget values can also be employed in order to consider a variety of attacker profiles.

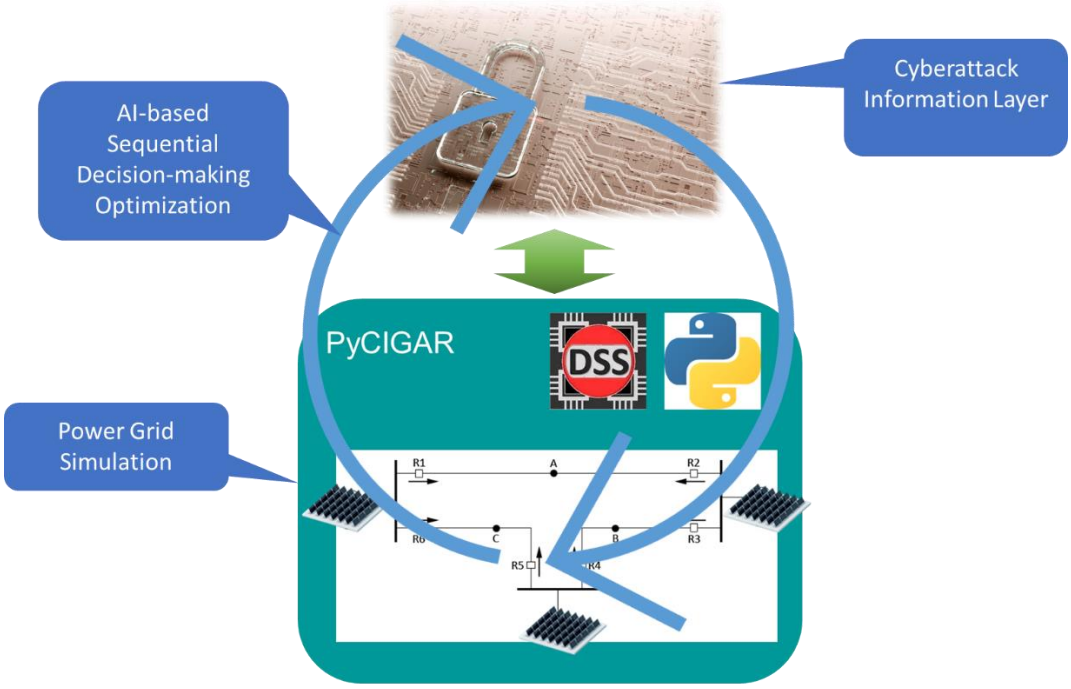

Figure 16 - Attack Optimization

During the optimization process, every adversary action affecting a device produces a state change. The diagram of Figure 17 presents how such state transitions are performed considering a fictitious "Device X". Every transition marked with a dollar sign ($) indicates that the corresponding cost (effort) associated with the action is subtracted from the attack budget. Dashed arrows correspond to alternative paths ("Device X" accessed as the entry point and not from another device) or options which may not be available for all devices (End Effect actions are only available for devices which can affect the power system operation). The diagram of Figure 17 only shows the transitions and states associated to devices, but analogous transitions and operations are also performed considering the network links.

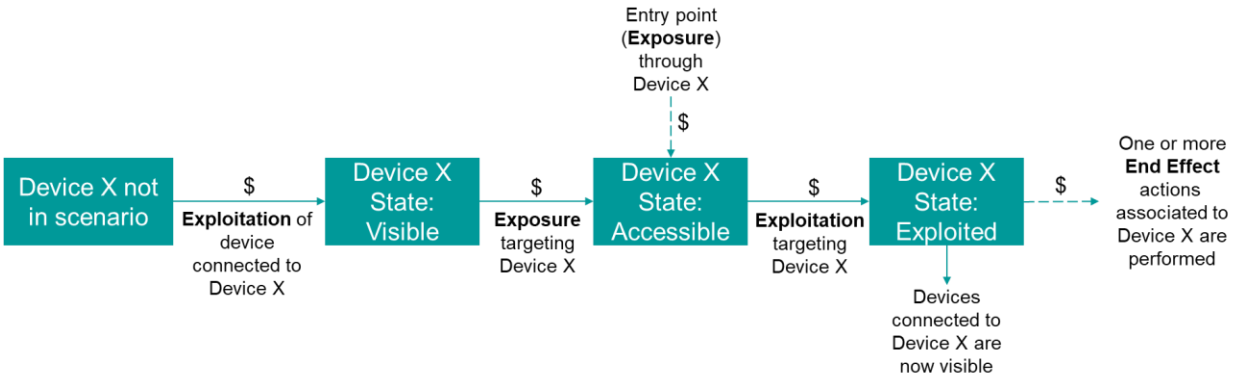

Figure 17 - State transition performed during optimization

The result of the optimization is a sequence of actions performed by the attacker and the resulting impact on the KPI.

Attack optimization tests have been performed using the same IEEE 3 bus network employed in other tests described in this report. The corresponding PyCIGAR model was employed for simulations. A fictitious network topology was designed for such tests and encoded in NetJSON as previously described. Figure 18 presents the visualization of such topology, which include 3 power system devices which correspond to the inverters (s701a, s702a and s703a) which control PV generation at each of the three buses.

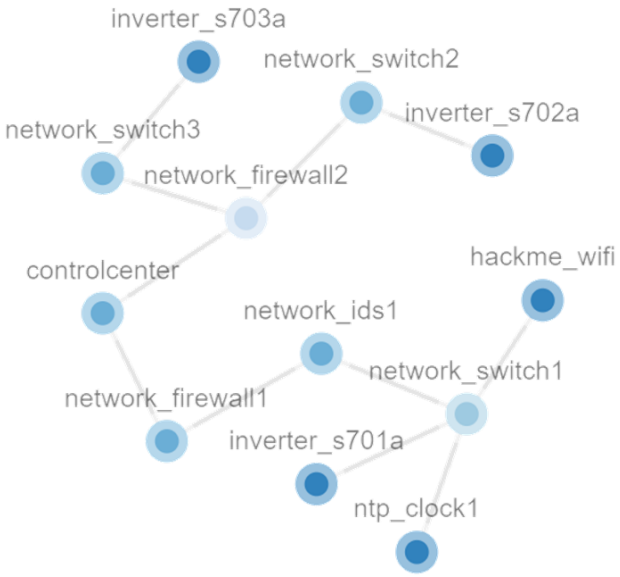

Figure 18 - Network topology employed for attack optimization testing based on IEEE 3 bus network.

NetJSON was populated not only with the devices and connections but also with possible adversary actions corresponding to the three hierarchical levels presented above. Figure 19 presents and excerpt of the NetJSON definition corresponding to inverter s701a. It can be noticed that information of possible adversary actions is included for each of the three hierarchical levels (Exposure, Exploitability, End Effect). Each possible action is associated with a cost that quantifies the effort required by the adversary to perform it. This quantification does not need to have a specific meaning in absolute terms, it must only be consistent in a relative sense, when comparing costs associated with different tasks, i.e., tasks which require more effort must have higher cost.

```
"id": "inverter_s701a",
"properties": {
  "type": "pv_device",
  "attack_cost": {
    "exposure": {
      "entry_point": {
        "cost": 100,
        "duration": 3
      },
      "from_connected": {
        "cost": 10,
        "duration": 1
      }
    },
    "exploitability": {
      "exploit_a": {
        "cost": 200,
        "duration": 3,
        "visible_attacks": [
          "change_setpoint"
        ],
        "visible_devices": []
      },
      "exploit_b": {
        "cost": 300,
        "duration": 4,
        "visible_attacks": [
          "oscillation"
        ]
      }
    },
    "end_effect": {
      "change_setpoint": {
        "cost": 10,
        "duration": 1
      },
      "oscillation": {
        "cost": 20,
        "duration": 1
      }
```


Figure 19 - Excerpt of the NetJSON representation of the computer network. The part presented corresponds to one of the inverters. It can be noticed information about possible actions in each of the three hierarchical levels and corresponding cost (effort)

Attack optimization was successfully tested based on the described inputs and simulation model. KPI employed was based on voltage imbalance. Table 7 presents the results of the optimization as a sequence of actions. Column “Device/Physical Link” presents the device or physical link which is the target of the attack. Physical link connecting “Device A” to “Device B” is presented as <Device A>__<Device B>. Column “Level” presents the hierarchy level of the action (Exposure, Exploitability or End Effect). Column “Action” presents the actual action taken. A network-based attack of type “Type” affecting the logical link connecting “Device X” to “Device Y” is presented as <Device X>__<Device Y>__<Type>. The last column presents the time required for computing the MCTS solution for the corresponding step.

Table 7 – Attack optimization result consisting of a sequence of adversary actions.

#	Device/Physical Link	Level	Action	Computation Time
1	inverter_s703a	exposure	entry_point	6185.385823965073s
2	inverter_s703a	exploitability	exploit_a	4807.120194196701s
3	inverter_s703a	end effect	pv_disconnect	3436.57035112381s
4	inverter_s703a	exploitability	exploit_b	1952.0697228908539s
5	network_switch3	exposure	from_connected	1891.1593968868256s
6	network_switch3	exploitability	exploit_b	2506.256336927414s
7	inverter_s703a__network_switch3	end effect	inverter_s703a__controlcenter__false_data_injection	2375.2099990844727s
8	inverter_s703a	end effect	volt_var_attack	1407.188632965088s
9	network_firewall2	exposure	from_connected	514.6099021434784s
10	network_firewall2	exploitability	exploit_a	367.4212591648102s
11	network_switch2__network_firewall2	end effect	inverter_s702a__controlcenter__false_data_injection	279.3138041496277s
12	network_switch2	exposure	from_connected	98.53733587265015s
13	network_switch2	exploitability	exploit_a	95.5657970905304s
14	inverter_s702a	exposure	from_connected	76.0338180065155s
15	inverter_s702a	exploitability	exploit_a	70.43082690238953s
16	inverter_s702a	end effect	pv_disconnect	56.886260986328125s

Referring to Figure 18, it can be noticed that the adversary takes a path from inverter s703a to s702a hoping to each device in between them. Other notable aspects of the results are as follows:

- It can be noticed that the resulting attack path was chosen so that two inverters could be affected in a way that did not require reaching the control center. This is compatible with the costs definition as higher costs have been associated to the control center compared to other devices.
- Computation time is in general reduced from one step to the next as the remaining attack budget is reduced after each step and therefore the number of action choices for the adversary is also reduced in most cases.

It is also interesting to evaluate the results from one specific MCTS step. Figure 20 presents the tree resulting from MCTS solution of the initial step (entry point) and expanded according to the final solution. Blue circles correspond to states and green labels correspond to actions. Although the clutter precludes identification of many of the labels, it can be noticed that each action is associated with *Q* and *N* values which correspond respectively to the Upper Confidence for Trees (UCT) and the number of times that node was visited. Those are the main metrics associated to the selection of nodes for each MCTS iteration [3].

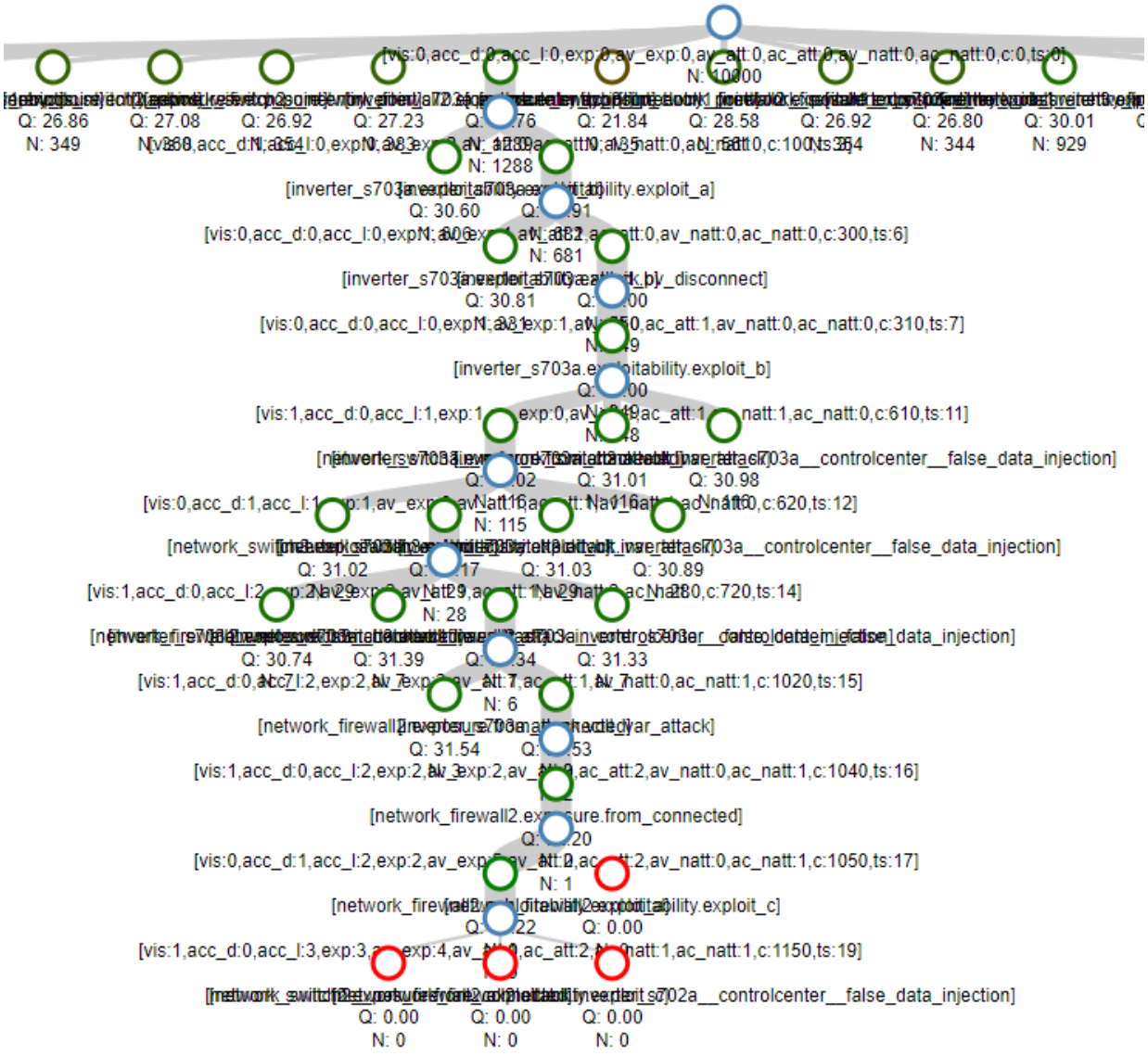

Figure 20 - Tree resulting from MCTS solution to entry point action (step #1). The tree was expanded following the final solution from the optimization reaching up to step #11. Part of the entry point options is not shown in the figure.

Computer Network Topology

In this section, we describe the communication network topology defined for the modified IEEE 123 multi-feeder described in section Power System and KPIs. We also describe the system architecture, components, design models and design considerations.

Network Architecture

A mesh topology is adopted to design a computer network for the IEEE 123 feeder system. A graph view of the defined computer network is presented in Figure 21. We factored in the following components while designing the network:

- The size of the network – number of controllable loads, location of the power equipment, area of coverage, etc.

- The amount of data being transferred and exchanged
- Security of the data
- Connectivity requirements

To achieve high availability, network components such as Routers and Edge devices (Network Switches) are placed at critical locations. The mesh network designed was inspired from Cisco's Distribution Automation Feeder Automation Design Guide [5]. The document provides a comprehensive explanation of the entire end-to-end Smart Grid Field Area Network (FAN) solution along with case studies.

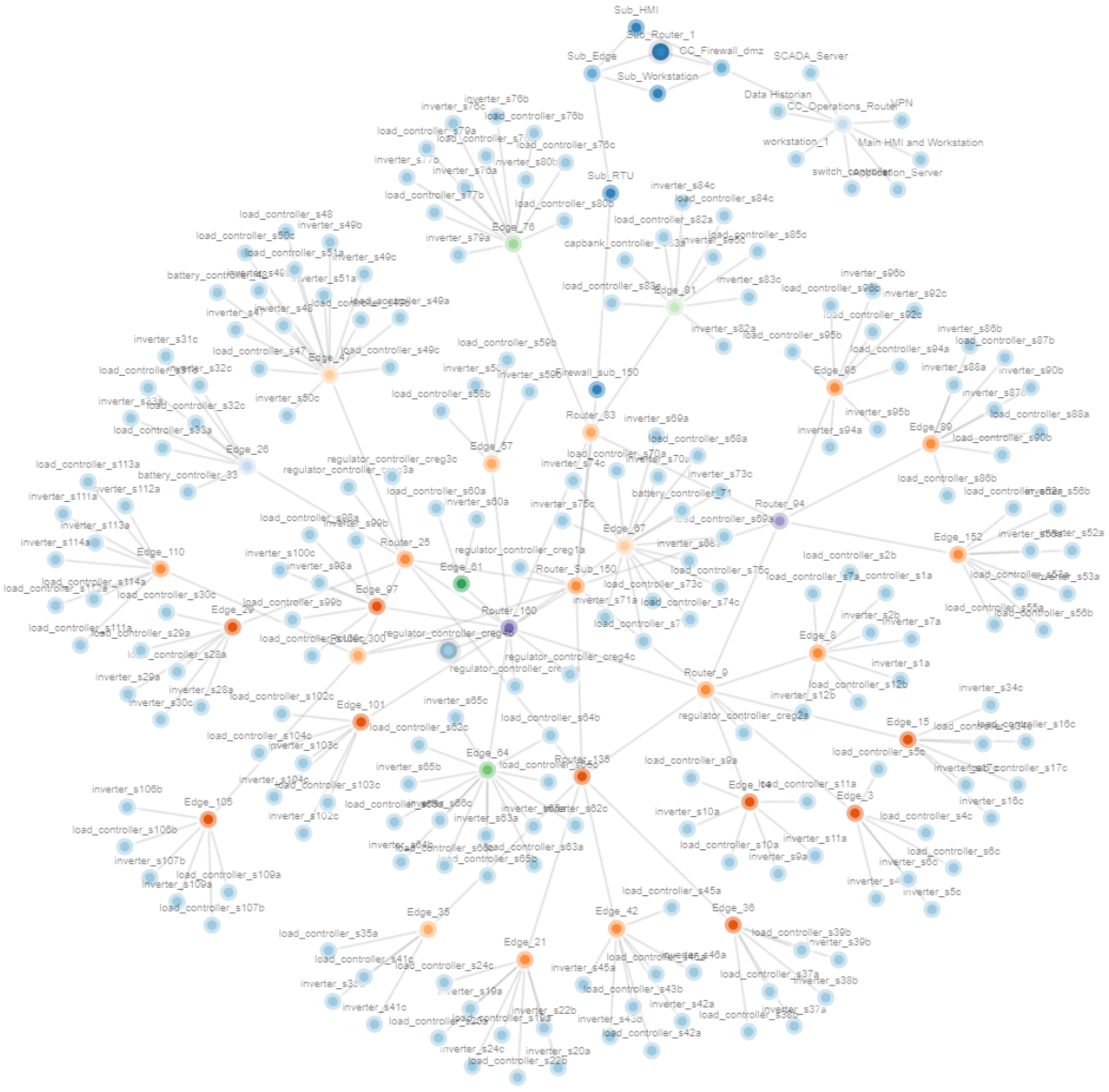

Figure 21 - Graph view of designed computer network topology associated to the modified IEEE 123 model

After defining the network topology, the network was designed in an interactive tool called Architecture Generation (Arcgen), which is described below. The sequence of steps from definition of architecture in Arcgen to final NetJSON output is shown in Figure 22 and described below.

1. First, we design the network architecture in the Arcgen tool
2. The architecture is then exported as a JSON file and parsed.
3. The missing links and devices are added to the file.
4. The possible adversary actions and corresponding costs are identified and mapped to the devices. The action categories are those described in section Attack Optimization Methodology and Testing. Additional information about how the attack costs have been defined is provided in section Cyberattacks and Costs.

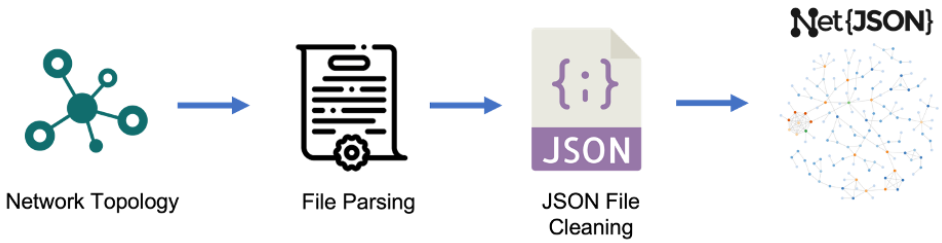

Figure 22 - Sequence of steps to construct a NetJSON file

Tools used to build a Network Topology:

Following are the tools that were used in this work to construct the network topology and associate costs to the devices and their links.

Arcgen

To design a network topology, we leveraged a publicly available tool designed by Pacific Northwest National Laboratory (PNNL), called Architecture Generation (Arcgen) as shown in Figure 23. It is an interactive asset management tool that provides illustrative OT architectures based on Purdue reference model [9]. The tool can be used to conceptualize and plan OT networks. It is an intuitive drag and drop framework that supports various network components and devices used in OT Networks such as Routers, Firewalls, Sensors, Controllers, etc. Once the components are placed and connected in the appropriate levels, the design can be exported to a JSON file. This JSON file is used as the basis for creation of the NetJSON.

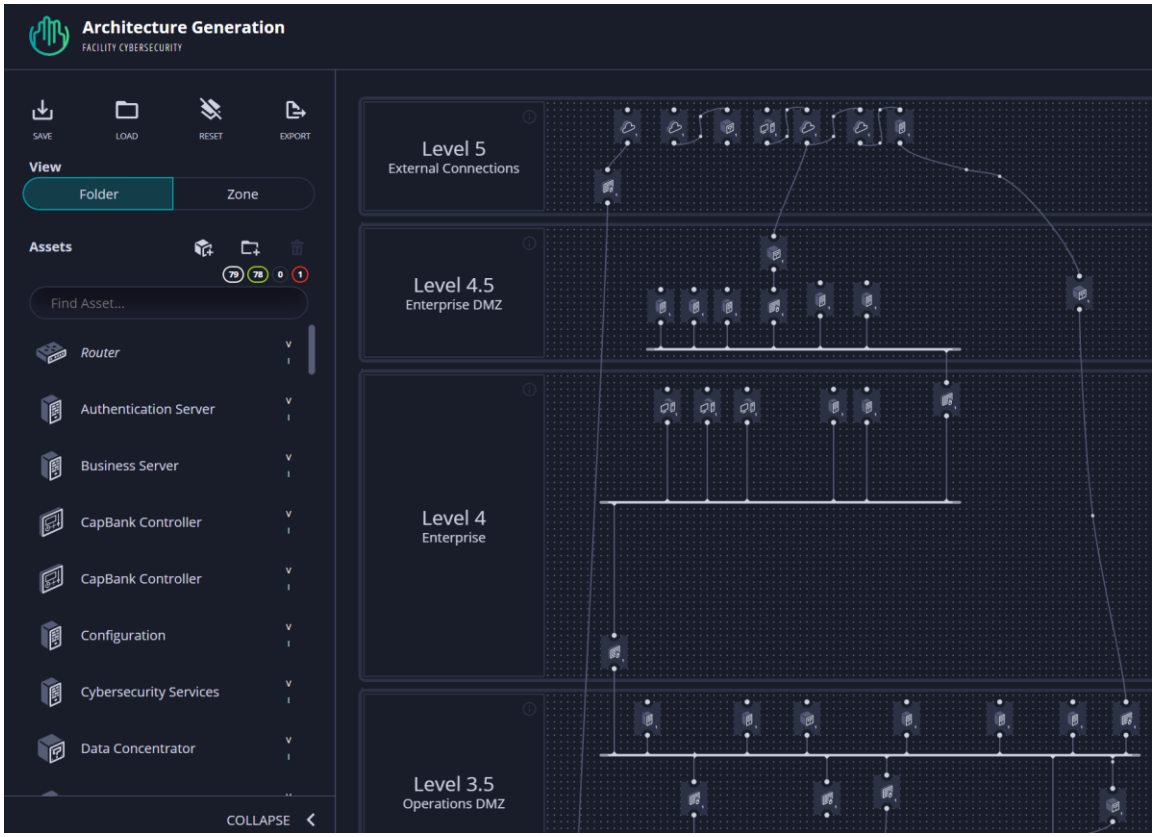

Figure 23 - Arcgen: Architecture Generation tool from PNNL that presents a user interface to design the network. The network architecture is imported as a JSON file.

File Parser

In order to ensure a proper NetJSON file is created, holding appropriate properties and adhering to the standard, the output JSON file of the Arcgen tool needs to be parsed. During this parsing, the missing links and devices are identified and added to the file. Every device and link is associated to attack cost properties in exposure and exploitability levels and also end effect if applicable. This is based on templates containing the information associated to each device type. The parser also identifies all logical links corresponding to each physical link, based on a definition of which devices exchange relevant information with each other. The functionality of Python package networkx [4] is employed to traverse the graph to determine such logical links. First, all the source nodes are identified and then the graph is traversed to identify the sink nodes and the paths connecting sources to sinks. The parser then associates each physical link to the corresponding logical links and to the possible attacker actions and properties. The result is a NetJSON with all required information.

Cyberattacks and Costs

In order to define realistic values for costs (effort) associated with specific cyberattacks, it was prudent to conduct offensive assessments on sample devices that resemble those of interest for the project and evaluate what effort-based costs are. Initial assessments allowed researchers to review the device and all aspects which were in scope for potential testing. This included potential hardware attacks, and over the wire or air attacks.

Attack methodology followed these standards in the cybersecurity industry [6]:

- Reconnaissance
- Initial Access
- Internal Reconnaissance
- Privilege Escalation / Lateral Movement
- Maintaining Access

Concerning the Power Grid devices selected for testing, their corresponding models are not mentioned here for confidentiality purposes, but they are of the following types:

- Smart Meter
- Solar Array w/AC Inverter
- Capacitor Bank
- Industrial Edge Device

The following Attacker Profiles were considered:

- Novice
- Amateur
- Pen-Tester
- Hacktivist
- State-Sponsored

Offensive Actions Evaluated [7]:

- MiTM Attacks
- Firmware Attacks
- Data Collection / Reconnaissance
- Tampering

Using the definitions listed above we were able to assess the devices from a perspective of our own experience and understanding of Power Systems and Cybersecurity. During most cybersecurity assessments, devices that utilize a microprocessor and links objects in memory to the physical world allow us as users to interact with them. From here as attackers, we can test the logic, and most importantly understand how the device was intended to work, this gives us an idea of how we are able to make the device work in a way it was not intended.

An example of the research performed is identified in the Automatic Metering Infrastructure (AMI). We aimed to understand the effort required to reverse engineer a Smart Meter that is equipped with a Communications Module to the point where we understood the functionality of the device and were able to start interacting with the firmware. Our intention here was to understand the level of effort required to create malware that can attempt to utilize the network created by the module.

Once we started to tamper with the device and inspect the PCB, we quickly noticed that UART pins were commonly named on the PCB. We attempted to connect to the UART interface but quickly found out the interface had been disabled. Since the microprocessor is a proprietary device, there is no published datasheet or schematic. We concluded that considerable effort would be required to reverse engineer this device further, ranking it accordingly in our data.

However, we did not stop there, the meter is equipped with 200A remote disconnect device, that has a two-way transponder. This device utilizes a commercial microcontroller that has ~256B of Flash Memory and 128B of ram. With the intent to dump the contents of the microcontroller we used a Development Kit to interface with the microcontroller and dump the contents of its flash memory. Using the datasheet, we were able to understand the microcontrollers memory organization and dump the firmware. While we were able to reverse engineer and develop malware for this device, we had yet to determine a viable delivery mechanism. Since we are not looking to develop actualities, we can assume that delivery could be over-the-air updates or Rogue RF style attacks [8].

Another example of analysis targeted a small commercial distributed energy resource (DER) system, consisting of solar panels, batteries, and controllers/inverters. As attackers we wanted to evaluate if it was possible to interfere with the equipment to disrupt generation.

A potential attack approach became apparent when our research team noticed that the controller connected to a module that transmits MODBUS over the air in cleartext. MODBUS is a communication protocol commonly employed in this type of application. When such communication is restricted to isolated ports there is very little impact or ability to abuse the protocol. However, when this data is transmitted over the air to an application that can be used to manage the device, such link can be targeted by an attacker.

This is exactly what we did, using a custom MODBUS MiTM proxy we intercepted the MODBUS TCP packets and modified the Function Code's Coil Status, as illustrated in Figure 24. We were then able to manipulate the data received by the remote application to indicate that the generation was OFF when in fact it was ON, resulting in discharge of the batteries.

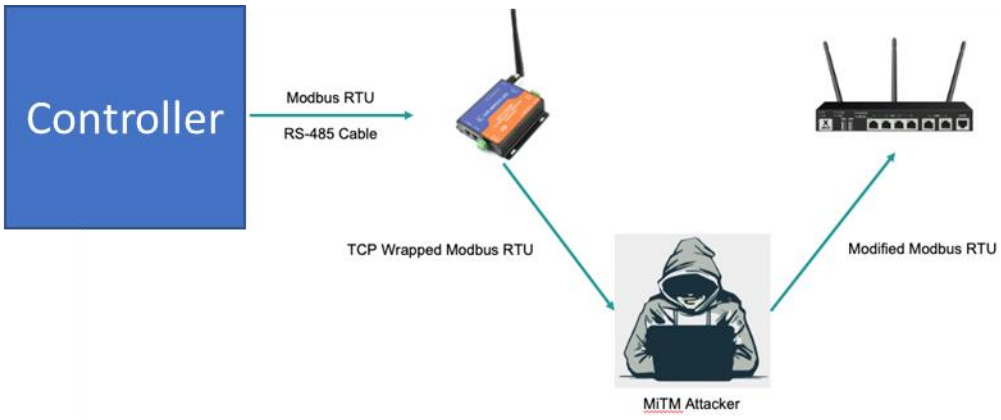

Figure 24 – diagram representing the attack performed targeting the DER system.

The information gathered from the studies such as those described above has been used as the basis for definition of the adversary actions and corresponding costs employed for the attack optimization.

Conclusion and Future Work

This report presented the development of the five building blocks required for optimizing attacks so that they can be used to test RL-based cyberattack countermeasures. Those building blocks are:

- Power System and KPIs
- Attack implementation in PyCIGAR
- Computer network topology
- Cyberattacks and costs
- Attack optimization methodology

Next steps consist of employing these building blocks to perform actual attack optimization targeting the modified IEEE 123 multi-feeder network.

References

[1] Siemens Technology, SPADES Red Team Report - Task 3.1. Submitted December/2020.
[2] Siemens Technology, SPADES Red Team Report - Task 3.2. Submitted December/2021.
[3] Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S. & Colton, S., (2012). A survey of Monte Carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1), pp.1-43.
[4] https://networkx.org/
[5] "Distribution Automation Feeder Automation Design Guide - Distribution Automation Feeder Automation Design Guide [Solutions]." Cisco. Cisco, September 24, 2020. https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/Distributed-Automation/Feeder-Automation/DG/DA-FA-DG/DA-FA-DG.html.
[6] "Cyber Kill Chain®." Lockheed Martin, June 29, 2022. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.
[7] "ICS ATT&CK Matrix." Matrix | MITRE ATT&CK®. Accessed November 17, 2022. https://attack.mitre.org/versions/v12/matrices/ics/.
[8] Alladi, Tejasvi, Vinay Chamola, Biplab Sikdar, and Kim-Kwang Raymond Choo. "Consumer IoT: Security vulnerability case studies and solutions." IEEE Consumer Electronics Magazine 9, no. 2 (2020): 17-25.
[9] Williams, T. J. (1992) The Purdue enterprise reference architecture: a technical guide for CIM planning and implementation. Research Triangle Park, NC: Instrument Society of America.

Addendum to REPORT Subtasks 3.3, 3.4, 3.5, 3.6

Supervisory Parameter Adjustment for Distribution Energy Storage (SPADES)

DOE CESER

CEDS Program

SUBMITTED BY

Siemens Corporation Technology

755 College Rd East, Princeton NJ

Submitted: June 20th, 2023

Technical Point of Contact

Dr. Bruno Leao - bruno.leao@siemens.com

Project Manager

Ramamani Ramaraj - ramamani.ramaraj@siemens.com

SUBMITTED TO

Lawrence Berkeley National Laboratory

Contents

Introduction

This addendum is a complement to the final project report submitted by Siemens Technology [1] corresponding to the Red Team efforts related to Subtasks 3.3, 3.4, 3.5, 3.6. This document describes the tasks performed by Siemens Technology as the Red Team for the SPADES project during the period between submission of the report (November/2022) and end of its no-cost extension period (June/2023). Siemens Technology work focused on two topics during this period:

- Extended development and testing of the proposed attack optimization framework described in the report.
- Support to NRECA for integration of attack simulation and power system key performance indicators (KPIs) to the Open Modeling Framework (OMF).

More details of the related work are described in the following section.

Tasks and Results

Extension of the Attack Optimization Framework

Work related to the attack optimization framework performed during the period relevant to this report comprised:

1. Mathematical formulation of the modeling and optimization problem.
2. Improved methodology for attack costs definition.
3. Attack optimization tests based on the modified IEEE 123 network described in the report, including extensive exploration of the solution space based on supercomputing for benchmarking purposes.
4. Preparation and submission of a paper describing the proposed attack optimization framework and related experiments.

Items 2-3 above are presented in more detail below. The paper mentioned in item 4 is available as a pre-print [2]. The mathematical formulation described in item 1 can be found in the paper.

Improved Attack Costs Definition Methodology

As presented in the report, each possible adversary action is classified in one of three categories:

- Exposure
- Exploitability
- End Effect

The improved methodology involves categories *Exposure* and *Exploitability*, as explained below.

Exposure

The *Exposure* category corresponds to the efforts required for accessing the target system either externally - from the internet or creating a physical security breach - or internally - by means of lateral movement. High costs for actions in this category could correspond to devices or systems that are hard to access and are well protected. Low costs on the other hand could result, for instance, from devices or systems that are directly connected to the internet with little or no protection.

Exposure costs are associated with factors such as network topology, access controls, and security posture. A subset of MITRE ATT&CK[1] tactics, techniques and procedures (TTPs) can be mapped to this category. Some examples of this mapping include:

- Network Scanning: An adversary may use network scanning to identify exposed systems or services that can be targeted for exploitation.
- Phishing: An adversary may use phishing to trick users into providing credentials or clicking on malicious links, allowing the adversary to gain access to the network.

Determining the exposure of an asset is a subjective evaluation that involves considering various factors. To compute costs associated to actions in the *Exposure* category, the following factors must be considered:

- Network Accessibility (NA): This factor represents the effort required to gain network access to the target system. It considers factors such as network security controls, encryption, authentication mechanisms, and network segmentation. A device directly connected to the internet and an isolated system would correspond respectively to high and low NA scores.
- Authentication Effort (AE): This factor represents the effort required to bypass or overcome authentication mechanisms to access the target system. It includes aspects such as the strength of passwords, multi-factor authentication, account lockouts, and other authentication-related defenses.
- Privilege Escalation (PE): This factor represents the effort required to elevate privileges or gain higher levels of access within the target system. It relates to aspects such as privilege separation, access controls, least privilege principles, and the complexity of privilege escalation techniques.
- Protection Level (PL): This represents the level of protection applied to the system. This may include firewall settings, antivirus software, or other cybersecurity measures.

Effort scores are associated with the abovementioned factors to determine their relative significance in the exposure assessment. These scores are defined based on expert judgment by considering factors such as the potential impact of the vulnerability, likelihood of exploitation, criticality of the asset, and the system's operational requirements. The higher the weight, the more influential the factor determines the associated action cost. The final cost is obtained by the sum of the factor scores as presented in the equation below, where φ represents attack cost:

$$\varphi_{exposure} = \varphi_{NA} + \varphi_{AE} + \varphi_{PE} + \varphi_{PL}$$

Each factor score has a range from 1 to 10, indicating their relative significance in determining the overall cost.

Exploitability

Actions in this category correspond to technical vulnerabilities that can be exploited by the adversary. Such actions may be associated, for instance, with software vulnerabilities and misconfigurations. The corresponding costs reflect the complexity associated to exploiting the system to be able to execute the attack. High costs associated to an action in *Exploitability* category could correspond to a need for

[1] https://www.mitre.org/focus-areas/cybersecurity/mitre-attack

specialized knowledge or even the need for a new zero-day exploit. Low costs on the other hand may be associated with known exploits that are easy to use. A subset of MITRE ATT&CK TTPs can also be mapped to this category. This includes the following:

- Exploit Public-Facing Application: An adversary may exploit vulnerabilities in public-facing applications, such as web applications or email clients, to gain access to the network.
- Remote Command Execution: This technique involves an adversary remotely executing commands on a targeted industrial control system. By exploiting vulnerabilities or leveraging authorized remote access, the adversary gains control over the system and executes malicious commands to manipulate or disrupt its operation.
- Rogue Master: Adversaries have the capability to establish a rogue master that takes advantage of control server functionalities to communicate with outstations. This rogue master can be utilized to send control messages that appear legitimate to other control system devices, resulting in unintended impacts on processes.

To compute the costs for tasks in the *Exploitability* category, we employ the Common Vulnerability Scoring System (CVSS)[2]. The Base metric group represents the inherent qualities of a vulnerability that remain consistent regardless of time or user environments. This group comprises two sets of metrics: (i) the Exploitability metrics and (ii) the Impact metrics. The Exploitability metrics capture the ease and technical methods involved in exploiting a vulnerability. These metrics represent the attributes of the vulnerable component itself, formally known as the vulnerable entity. Exploitability metrics comprise four components – Attack Vector, Attack Complexity, Privileges Required, and User Interaction. These metrics align with our needs for calculating the attack budget. The attack budget is then defined as follows:

$$\varphi_{exploitability} = 100 \cdot e^{4-ES}$$

Where *ES* corresponds to the exploitability score from CVSS which ranges from 0.1 to 4 (valid for CVSS version 3.1 and later). This is the cost to launch an attack, considering resources like time, personnel, and technology, without any specific vulnerability in mind.

Attack Optimization Tests

Attack optimization tests were performed based on the developments described in the report:

- The power system simulation model consists of a modified IEEE 123 network implemented in OpenDSS/PyCIGAR. This includes all types of devices and corresponding attacks described in the report.
- The VI+SPF KPI was employed, corresponding to a combination of voltage imbalance and substation power factor.
- All required cybersecurity information including computer network topology and information on each device was created based on the NetJSON format. This includes information about all possible adversary actions associated to each device or network link with their corresponding category and cost.

[2] https://www.first.org/cvss/

- Monte Carlo Tree Search (MCTS) is employed for optimization based on an implementation developed using the Julia programming language.

As a basis for evaluation of the quality of results, a large number of attack scenario samples were generated and evaluated for computation of the corresponding reward using a node from the Lawrencium super computing cluster comprising an Intel Xeon Gold 5218 processor with 32 cores and 1584GB of RAM. The same logic and constraints applied in the optimization were employed for generation of these samples. The attack budget considered is also the same as the experiments. All cores of the computing node were employed to generate the samples and memory sharing mechanisms were applied to minimize the repetition of samples in separate processes. A total of 504 wall-clock hours were employed for such processing, producing 504217 attack scenario samples and the corresponding reward values. An empirical cumulative distribution function (ECDF) based on the reward samples is used to evaluate the performance of the attack optimization experiment results, indicating the percentage of samples that produce rewards that are lower than the one under analysis. Such percentage will be referred to as p_{CDF} hereafter.

Optimization experiments have been performed on a computer running Ubuntu 18.04 operating system, comprising Intel Xeon Silver 4210 processor and 187GB of RAM. Table 1 presents the results. All tests were performed considering the same attack budget and MCTS configurations. The durations presented in the table were normalized for a fair comparison as simulations performed later benefit from stored results produced by previous tests. It can be noticed that all results correspond to high p_{CDF}. Most results are within 0.951 and 0.984 with one exception in terms of lower value (0.931 in scenario #6) and another corresponding to a high value (>0.999 in scenario #4). It can also be noticed from the results that increasing the number of iterations has an expected impact in duration, but it does not seem to impact p_{CDF}. The possible explanation is that all values tested for the number of iterations are the same order of magnitude and it would require order of magnitude increases in this number to obtain systematic improvements. Such order of magnitude increase would be feasible by applying some approaches described below in future work.

Table 1 - Attack optimization experiments and results

OPTIMIZATION EXPERIMENTS AND RESULTS

Scenario	Iterations	Duration [h]	p_{CDF}
1	2000	27	0.984
2	2000	24	0.962
3	2000	21	>0.999
4	5000	60	0.951
5	5000	69	0.984
6	5000	60	0.931
7	7500	103	0.981
8	7500	64	0.962
9	7500	103	0.984

As an illustration of the adversary steps associated to those attack scenarios, below is a description of the steps yielded in scenario #3 which corresponds to the best p_{CDF} in the table:

- Adversary uses as entry point (*Exposure* category) a network link connecting two routers which contain communication from multiple controllers but is exposed in the field and not behind a firewall. This can be considered a good trade off in terms of attack cost and impact.
- Once the link is accessible, the adversary performs false data injection attacks (*End Effect* category) affecting four controllers corresponding to three controllable loads and one PV inverter.

Support to NRECA in Integration of PyCIGAR Functionalities to OMF

As described in the report, Siemens Technology has implemented additional functionality into PyCIGAR including a variety of power system devices and associated attacks, means for configuring those attacks as part of an attack scenario, and definition of new KPIs.

NRECA is integrating PyCIGAR as part of their OMF. We have supported them in including the functionality developed by Siemens Technology by means of explanations and discussions about the implemented code and generation of sample data representing attack scenarios that could be used for testing.

Conclusion and Future Work

This Addendum to the final Red Team report corresponding to subtasks 3.3, 3.4, 3.5 and 3.6 presented the tasks developed between the submission of the report and the end of the project considering its no-cost extension until June/2023. Tasks developed included: (i) enhancements of the developed attack optimization methodology making the definition of attack costs more systematic and connected to well accepted cybersecurity practices; (ii) experiments to test the methodology based on the modified IEEE 123 model and corresponding computer network and cyberattack definitions described in the report; (iii) support to NRECA in the integration of novel functionality to OMF.

Results obtained from the attack optimization experiments presented promising results, which achieved the 93rd percentile in all cases and the 95th percentile in all but one case. Over 500000 attack scenario samples generated and run through the simulation for reward calculation using the Lawrencium supercomputer cluster have been employed as baseline for this assessment.

One of the key aspects that limit the use of the proposed methodology is computational performance as it relies on the simulation for calculation of KPIs which are used as rewards for optimization. Considering the MCTS solution, one clear possibility to explore consists of parallelizing the calculations based on methods such as the one proposed by Liu et al. [3]. This kind of enhancement can be very desirable as it may provide benefits which are application agnostic. Heuristics can also be designed to speedup MCTS computations.

An alternative possibility for reducing computation costs is the use of surrogate models to replace the original simulation, however this approach is in general very dependent on the application under consideration. It must be noticed that the possibilities described above are not mutually exclusive, hence they can also be combined for additional benefit.

Another opportunity for future work is the integration between the attack optimization and training of the automatic mitigation, e.g. using the attack optimization to generate training data related to the most relevant cases to make sure those are considered during training of machine learning models.

Finally, it is also valuable to work further on testing and improving the methodology itself. The pursuit of systematic means for the definition of costs and budgets so that they are less dependent on subjective evaluation by subject matter experts is a relevant topic to consider. The incorporation of extensively adopted cybersecurity resources such as MITRE ATT&CK and CVSS in such definitions, which is already part of the methodology, is one step in this direction. However further improvements in this aspect would facilitate additional applications of the methodology.

References

[1] Siemens Technology, SPADES Red Team Report - Subtasks 3.3, 3.4, 3.5, 3.6. Submitted November/2022.

[2] B. P. Leao, J. Vempati, S. Bhela, T. Ahlgrim, and D. Arnold. “Augmented Digital Twin for Identification of Most Critical Cyberattacks in Industrial Systems,” available at: https://doi.org/10.48550/arXiv.2306.04821, 2023.

[3] A. Liu, J. Chen, M. Yu, Y. Zhai, X. Zhou, and J. Liu, “Watch the unobserved: A simple approach to parallelizing monte carlo tree search,” in International Conference on Learning Representations, 2020. [Online]. Available: https://openreview.net/forum?id=BJlQtJSKDB