CySec Game: A Framework and Tool for Cyber Risk Assessment and Security Investment Optimization in Critical Infrastructures

Burhan Hyder, Harrison Majerus, Hayden Sellars, Jonathan Greazel, Joseph Strobel, Nicholas Battani, Stefan Peng, and Manimaran Govindarasu

2022 Resilience Week (RWS) | 978-1-6654-8819-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/RWS55399.2022.9984040

***Abstract*—Cyber physical system (CPS) Critical infrastructures (CIs) like the power and energy systems are increasingly becoming vulnerable to cyber attacks. Mitigating cyber risks in CIs is one of the key objectives of the design and maintenance of these systems. These CPS CIs commonly use legacy devices for remote monitoring and control where complete upgrades are uneconomical and infeasible. Therefore, risk assessment plays an important role in systematically enumerating and selectively securing vulnerable or high-risk assets through optimal investments in the cybersecurity of the CPS CIs. In this paper, we propose a CPS CI security framework and software tool, CySec Game, to be used by the CI industry and academic researchers to assess cyber risks and to optimally allocate cybersecurity investments to mitigate the risks. This framework uses attack tree, attack-defense tree, and game theory algorithms to identify high-risk targets and suggest optimal investments to mitigate the identified risks. We evaluate the efficacy of the framework using the tool by implementing a smart grid case study that shows accurate analysis and feasible implementation of the framework and the tool in this CPS CI environment.**

***Index Terms*—CPS, Critical Infrastructure, Cybersecurity, Game Theory, Attack-Defense Tree, Risk Assessment**

I. Introduction

The President's National Infrastructure Advisory Council (NIAC) report [1] highlights the need to secure critical infrastructures like the smart grid against the growing number of cyber incidents in these infrastructures that can lead to human and economic loss. The NIAC report, amongst its recommendations for protecting critical infrastructures against cyber threats, highlights the need for optimally aligning resources for cyber defense of these infrastructures. Given the dynamically changing cyber threat landscape, it has become necessary for the stakeholders of the CIs to develop and implement novel methods for securing these critical assets against such potential attacks. Due to the intricate and complex design of the CPS CIs and the increasing sophistication of targeted cyber attacks, risk assessment and optimization of security investments in these infrastructures are as important as it is challenging to prevent successful cyber intrusions [2], [3].

While there is a plethora of research for cyber risk assessment and investment optimization [4], [5] in CPS CIs like smart grids, the existing research mostly lacks providing solutions that the CI stakeholders can readily adopt into the current CPS CI environments. A step towards bridging the gap between academic research and industry practices is the development of frameworks and tools that can assist the stakeholders of these infrastructures in better assessing the cyber risk and optimally allocating security resources. Previous work in the development of such tools has focused on using attack-defense trees [6], Petri nets-based models [7], reinforcement learning, and Bayesian models [8]. PRISM [9] is a framework for organizations to assess cybersecurity risk. Authors in [10] do a survey of various cyber risk assessment tools like Nessus, EyeRetina, OpenVAS, etc. in the cybersecurity domain comparing their properties, metrics, and strategies. Moreover, several commercial and open-source products are available to perform cyber-physical security assessment of industrial control systems. For example, the CALDERA framework [11] allows users to run simulated breach-and-execution scenarios. CyberX [12] is designed to passively scan an IOT/ICS network to identify vulnerabilities specific to IOT/ICS systems.

A majority of these tools have either innovative theoretical analysis, but lack in the practical implementation [6]–[9] or are well implemented for practical use but lack sound mathematical frameworks [11], [12].

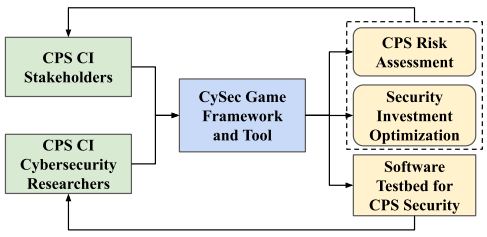

Fig. 1. CySec Game Framework and Tool: Use Cases

In this paper, we propose a CPS CI security framework, CySec Game, for quantitative risk assessment and cybersecurity investment optimization in CPS CIs like the smart grid and an open-source web-based software tool based on the framework. The framework consists of (1) a user interface implemented as a web-based front-end application that allows users to input the security structure of their system in the form of attack-defense trees, and (2) an analytical engine implementing three algorithms for risk assessment and investment optimization, namely, attack tree, attack-defense tree, and game theory algorithms. Fig. 1 shows a possible use-case of the CySec Game framework and tool which can be used by (1) CPS CI stakeholders to perform quantitative cyber risk assessment and investment optimization within their infrastructure, and (2) CPS CI cybersecurity researchers to test and evaluate their algorithms for CPS CI cybersecurity with the CySec Game platform acting as a software testbed for performing attack and defense studies.

II. CySec Game Framework: Design and Methodology

Fig. 2 shows the conceptual flow of information of the proposed CySec Game framework. The information flow essentially follows the following four steps:

1) **Step-1:** Information that the user needs prior to using the framework
2) **Step-2a/2b/2c:** List of inputs that the user has to provide
3) **Step-3a/3b/3c:** The back-end processes that the framework uses to calculate the outputs using the user inputs
4) **Step-4a/4b/4c:** The outputs that the framework provides the user with

The information that the user needs to prepare before using the framework includes (1) defining the architecture of the CPS CI for which risk assessment needs to be done, (2) identifying all the potential access points that the attacker can use to intrude into the system, (3) identifying the potential defense measures that can protect each of the identified access points, and (4) defining the impact of a successful attack on the system. The framework provides the user with three algorithms for risk assessment and security investment optimization, namely, Attack Trees (AT), Attack-Defense Trees (ADTs), and, Game-Theory (GT). Each algorithm has different options for inputs to be provided by the user. Based on the algorithm and the inputs, the analytical engine provides quantitative risk assessment (risk indices) for all the attack paths that the attacker can take to achieve its objectives. Additionally, using ADT and GT algorithms, the framework also provides security investment strategies for mitigating the risk. A comparison of the three techniques used in this work is discussed in one of our previous works [5].

A. User Interface: Front-end

The user interface allows the user to choose between the three different analytical engine algorithms and visualize the created tree diagram. The different inputs for each of the algorithms are listed below:

1) Attack Tree:

- Risk threshold is given for tree (acceptable level of risk).
- Each attack node (leaf node) has an associated attack probability.
- Root node (of the tree) has an attack impact value.

2) Attack-Defense Tree:

- Risk threshold is given for tree (acceptable level of risk).
- Each attack node has a defense mechanism with a cost associated with it.
- Each attack node has two probabilities: one denoting "pre-defense" probability of attack, the other denoting "post-defense" probability of attack. $[P^{attack}_{post-defense} \leq P^{attack}_{pre-defense}]$
- There is a fixed budget provided for security investment.

3) Game theory:

- Each attack node has an associated attack probability and attack cost.
- Each attack node has a corresponding defense node with a defense cost.
- Each attack node has an associated probability of attack.
- There is a fixed budget provided for security investment.

B. Analytical Engine: Back-end

To better understand the functioning of the analytical engine algorithms, an example of an attack-defense tree is shown in Fig. 3. The figure shows four access points (leaf nodes) for the attacker to breach the system $L_1,\ L_2,\ L_3$, and L_4. Each leaf node is associated with a defense node $D_1,\ D_2,\ D_3$, and D_4, respectively. The attacker target (root node) R can be reached through three attack paths: ***Attack Path-1***: L_1 and L_2; ***Attack Path-2***: L_3; and ***Attack Path-3***: L_4

Note that both L_1 and L_2 nodes need to be breached to reach R as these are conjointed by an *AND* node, while either one of the L_3 or L_4 nodes can be breached to reach R as these are conjointed by an *OR* node.

Furthermore, each leaf node (j) is associated with a cost, C^j_{aL}, that the attacker incurs to breach that access point. The leaf nodes also have an associated probability of attack prior to and post-defense, P^{pre}_j and P^{post}_j, respectively. Each defense node (k) is associated with a cost, C^k_{dL}, that the defender incurs to secure that node against an attack. The root node has an impact cost, C_I, associated with it which represents the additional cost incurred by the defender if the attacker achieves its target.

In order to secure all the attack paths, the defender needs to invest in one of the defense nodes that precede the *AND* nodes and all the defense nodes that precede the *OR* nodes.

The algorithms used in the analysis engine are described in the following sub-sections.

*1) **Attack tree engine**:* The attack tree algorithm does not use defensive measures as shown in the example in Fig. 3 to mitigate risk and the associated post-defense attack probability, and defense costs. This engine performs a recursive depth-first search of the input network tree to find all possible attack scenarios. The core logic is given below:

1) If the node is a leaf node, the only possible attack scenario is the set containing the leaf node
2) If the node is an AND node, the set of possible attack scenarios is the Cartesian product of the child nodes' attack scenario sets
3) If the node is an OR node, the set of possible attack scenarios is given by the union of the child nodes' attack scenario sets

Risk index (R) for attack scenario j is calculated for each attack path as given in (1).

$$R_j = C_I * \prod_{i=1}^{m} P^i_j \tag{1}$$

where m is the number of leaf nodes in the attack scenario j.

The engine outputs a list of all the attack paths (or scenarios) in decreasing order of risk index.

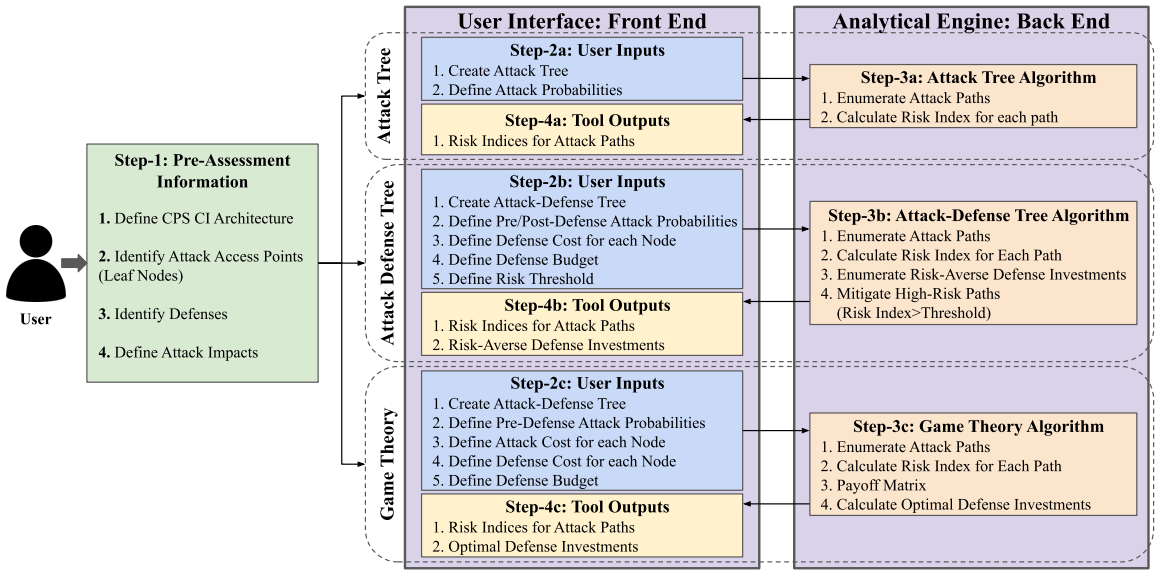

Fig. 2. CySec Game Framework: Functional Flowchart

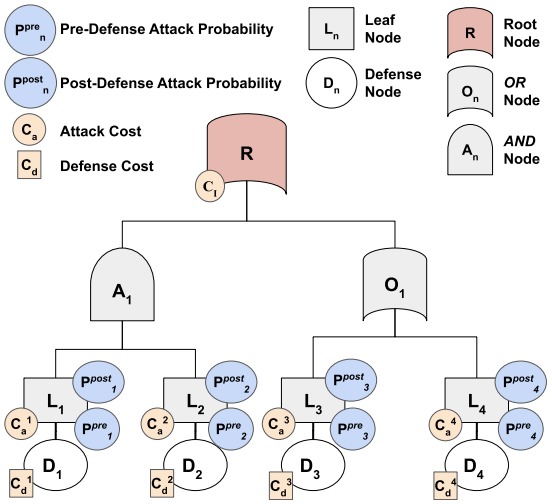

Fig. 3. Attack-Defense Tree Example

*2) **Attack defense tree engine**:* This engine computes the set of successful attack scenarios in a similar manner to the attack-tree engine. The addition is that the algorithm also determines risk-averse defense strategies given the following:

1) Defense cost associated with each defense scenario
2) Change of attack probability from pre- to post-defense
3) Defense Budget (C_{budget})
4) Risk Threshold ($R_{threshold}$)

The algorithm loops through successful attack scenarios and if scenario j has $R_j \geq R_{threshold}$, it looks through all defenses that could apply to that scenario and defends an attack with the highest return on investment (i.e., attack probability change per unit cost incurred) if $C_d^j \leq C_{budget}$. The algorithm continuously evaluates defenses against attacks until none of the defenses are within the current budget or if the scenario's risk percentage drops below $R_{threshold}$. The output includes the successful attack scenarios along with their pre-defense and post-defense R_j, the defenses that should be deployed against attacks to bring risk within $R_{threshold}$ given C_{budget}, and the cost of defense, C_d^j.

*3) **Game theory engine**:* Like the attack tree engine, the game theory engine also performs a recursive depth-first search of the input network tree, but it computes defense scenarios in addition to attack scenarios. Here, a "defense scenario" is a combination of defense nodes that prevents a compromise of the root node. The tree recursion is called on the root node of the diagram and works as follows:

1) If the node is a leaf node
   a) The only set of attack scenarios is the set containing the leaf node.
   b) The only set of defense scenarios is the set containing the corresponding defense node.
2) If the node is an AND node
   a) The set of possible attack scenarios is the Cartesian product of the child nodes' attack scenario sets.
   b) The set of defense scenarios is the union of the child nodes' defense scenario sets.
3) If the node is an OR node
   a) The set of possible attack scenarios is given by the union of the child nodes' attack scenario sets.

After computing the set of attack and defense scenarios, the payoff matrix for the system is calculated. The payoff of the attacker, U_a^{jk}, for strategy j when the defender chooses strategy k is given by (2).

$$U_a^{jk} = C_d^k + C_I - C_a^j \tag{2}$$

where C_d^k is the cost of defense for defense strategy k and C_a^j is the cost of attack for attack strategy j. C_d^k and C_a^j are calculated as shown in (3).

$$\begin{aligned} C_d^k &= \sum_{i=1}^{v} C_{dL}^i \\ C_a^j &= \sum_{i=1}^{w} P_i * C_{aL}^i \end{aligned} \tag{3}$$

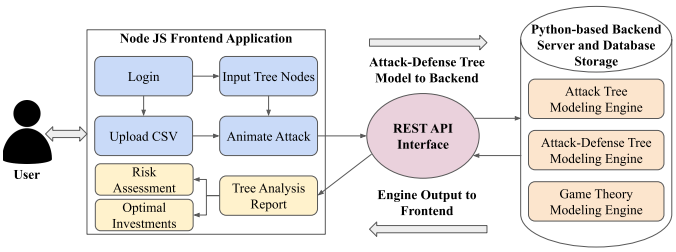

Fig. 4. CySec Game Tool: Architectural Overview and Interfaces

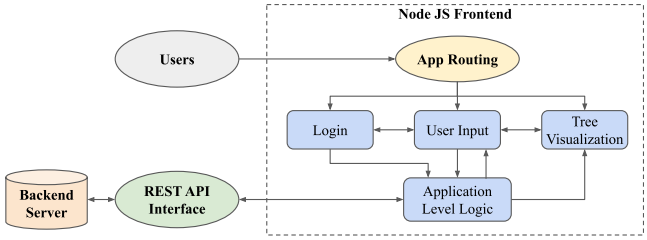

Fig. 5. CySec Game Tool: Front-end Application Architecture

where v and w are the number of elements in strategy sets k and j, respectively, and P_i is the probability of attack of the leaf node i.

For a zero-sum game, the defender's payoff is the negative of the attacker's payoff. The payoff for the defender, U_d^{jk}, when the defender chooses strategy k and the attacker chooses strategy j is given by: $U_d^{jk} = -U_a^{jk}$.

From the payoff matrix, Nash Equilibrium is used to generate the optimal defense strategy for the defender to choose from the defense scenario set from the payoff matrix. The algorithm outputs the recommended optimal investment strategy given the C_{budget} along with the cost of defense.

Moreover, calculations and assumptions for obtaining attack probabilities in the three algorithms used in this work have been discussed in our previous works [3], [5].

III. CySec Game Tool: Design and Architecture

The CySec Game Tool implements the CySec Game framework and the details about the front-end application, the back-end server, and the interfacing of different components of the tool are provided in the following subsections.

A. Overall Architecture and Interfacing

Fig. 4 shows the overall architecture and the various interfaces of the CySec Game Tool. The main interfaces of the tool are:

1) Between the User and the front-end application
2) Between the front-end application and the back end server
3) Between the back-end server and the analysis engines
4) Between the back-end server and the database

Upon opening the web-application, the user is prompted to login. From there the user is able to view saved diagrams and to create a new model or upload a file to import a model using CSV files. From the imported model or using a newly created model in the front-end web application, the user is able to interactively edit/create their model inside the tool. After finalizing their model, the application converts the model into JSON format so the back-end can process it which is interfaced using a REST API module. In the back-end server, based on the algorithm chosen by the user, the model is analyzed for risk and/or potential recommendations for optimal investments are developed. The user then sees a summary of risk indices with the recommended optimal investments based on the risk assessment in the front-end application. The user also has the option to save and export their results and model.

B. Front-end Application

Fig. 5 shows the front-end application logical architecture. The front-end is a web-based application that can be accessed using a web browser. The user logs into the application and chooses to create a new diagram or upload an existing diagram.

C. Back-end Server

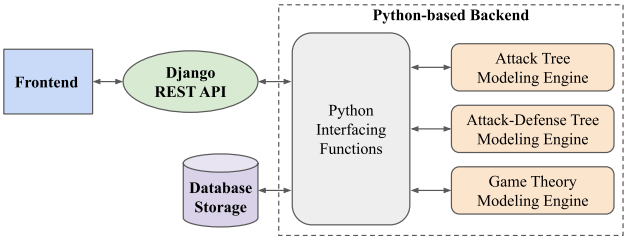

Fig. 6. CySec Game Tool: Backend Architecture

Fig. 6 shows the back-end server architecture along with the database storage interface. The figure shows the three algorithms (or analysis engine) interacting with the front-end and the database storage using the *Python interfacing functions*. These functions enable the transfer of data back-and-forth between the back-end, and the front-end and the database. Each analysis engine normalizes input probabilities such that the probability of attack nodes sum to 1. After normalization, each engine computes the relevant attack and defense scenarios, performs further calculations, and returns a result to the front end.

D. Database

The database storage is interfaced with the back-end server as shown in Fig. 6. The database is used to store and retrieve user created models/trees for future use and reference. The functions performed by the database storage system are:

1) Construct tables to store model and parameter data.
2) Create required queries to load model and parameter data.

E. Implementation Software

The front-end application is developed in Angular using GoJS platform for graphing. The GoJS platform enables the use of the platform for visualization and the TreeLayout library which prevents the user from submitting false trees. The back-end server uses the Python platform in a Django framework to implement the algorithms and request and return values to and from a web-based client.

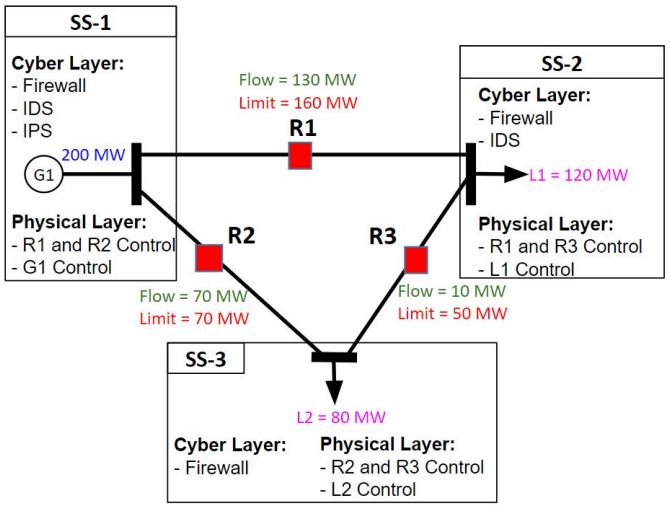

Fig. 7. Cyber-Physical Model of a 3-bus Smart Grid System

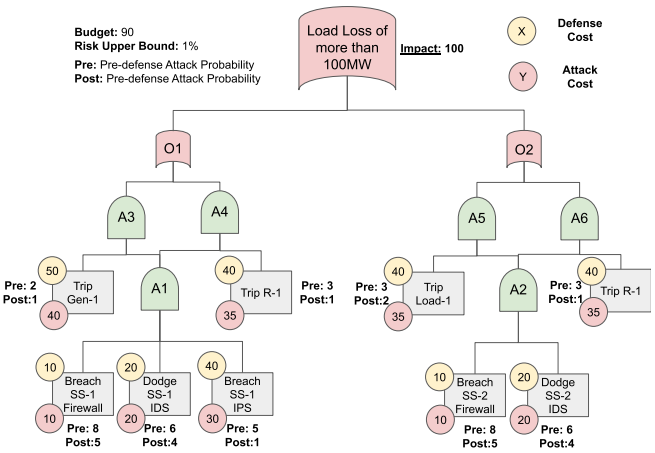

Fig. 8. Attack-Defense Tree for the 3-bus Smart Grid System

IV. Evaluation: A Smart Grid Case Study

A. Cyber-Physical Smart Grid Model

Fig. 7 shows a schematic of a synthetic cyber-physical smart grid model of a 3-bus power system. Each bus represents a cyber-physical substation, namely, $SS-1$, $SS-2$, and $SS-3$. Each substation consists of cyber-layer components and physical-layer components. The cyber-layer components include *Firewall*, *Intrusion Detection System (IDS)*, and/or *Intrusion Prevention System (IPS)*. The physical-layer components include, *generation*, *relay*, and/or *load control*.

B. Attack Defense Tree Engine Evaluation

With the assumption that the attacker aims to cause a load loss of $100MW$ or more, the ADT for this case study is shown in Fig. 8. The ADT also shows the cost of attack and defense for each leaf node as well as the impact cost for the root node. The probability of attack pre- and post-defense for each leaf node is also indicated in the ADT. The probabilities of attack for each substation are normalized by the tool to add up to 1. Instead of showing defense nodes in the ADT, only the defense cost for securing the leaf nodes are indicated in Fig 8. Suggesting possible technologies for defense against the given attacks is out of the scope of this work. The acceptable risk threshold percentage and defense budget are also indicated in the figure.

Fig. 9 shows a snapshot of the front-end application of the CySec Game Tool with the aforementioned case study example implemented. The attack defense tree is built in the tool while providing various inputs for each node in the tree as outlined in Section III-B. The output from the tool on the right pane correctly identifies 4 attack scenarios for this case study. The output also shows risk-averse defensive investments recommended by the back-end engine that mitigate the risk of the system (mitigation of high-risk attack scenarios) below the threshold level, given the defense budget.

C. Attack Tree and Game Theory Engine Evaluation

Fig. 10 shows the output of the CySec Game Tool for this case study using the Attack Tree and Game Theory engines. The left-most pane shows the output for attack tree engine where all four possible attack scenarios are listed with the associated risk indices and probabilities of successful attack. The right two panes show the output for the game theory engine with recommended optimal defense investments given the budget of the defender along with other analysis such as various attack scenarios, the payoff matrix, and the Nash Equilibria for the given attack-defense tree. The inputs for the attack tree and game theory engines are very similar to the attack defense tree algorithm except that the defensive nodes are not included in attack tree engine input and the post-defense attack probabilities are not considered in both the engines.

Additionally, we have made the tool available online for use for the general public [13].

V. Conclusion and Future Work

In this paper, we proposed a framework and software tool, CySec Game, for the cyber risk assessment and optimal allocation of cybersecurity investments to mitigate risks in CPS CI environments. The proposed framework uses attack tree, attack-defense tree, and game theory algorithms to identify high-risk targets and suggest optimal investments to mitigate the identified risks. The tool is developed and implemented using open-source software libraries with a web-based interface that makes it easier to use and implement without any major external software dependencies. The framework is evaluated using the tool implementing a 3-bus cyber-physical smart grid case study and the results show accurate analysis of the attack-defense scenarios, and feasible implementation of the proposed framework and tool in a smart grid environment. For future work, we plan to improve the back-end algorithms in order to minimize the number of inputs that the user has to provide such as including attacker and defender models that can estimate the probability and cost of attacks.

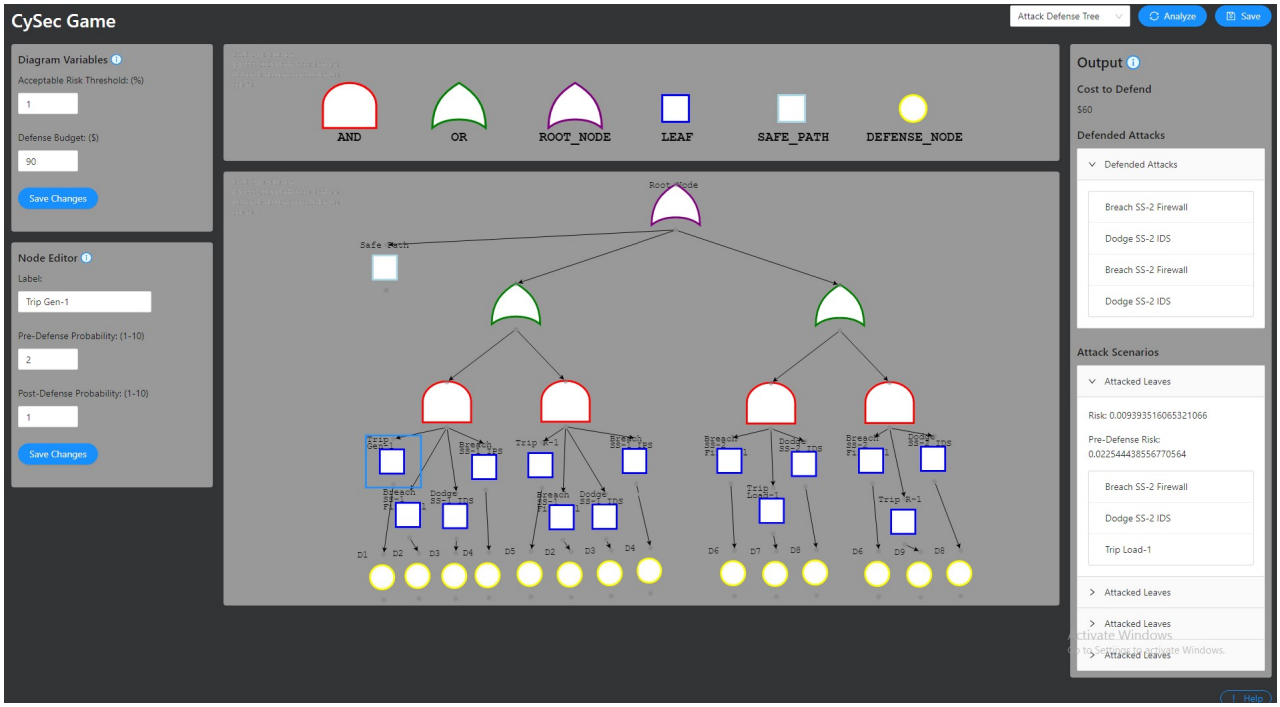

Fig. 9. CySec Game Front-end Application Snapshot for the Case Study

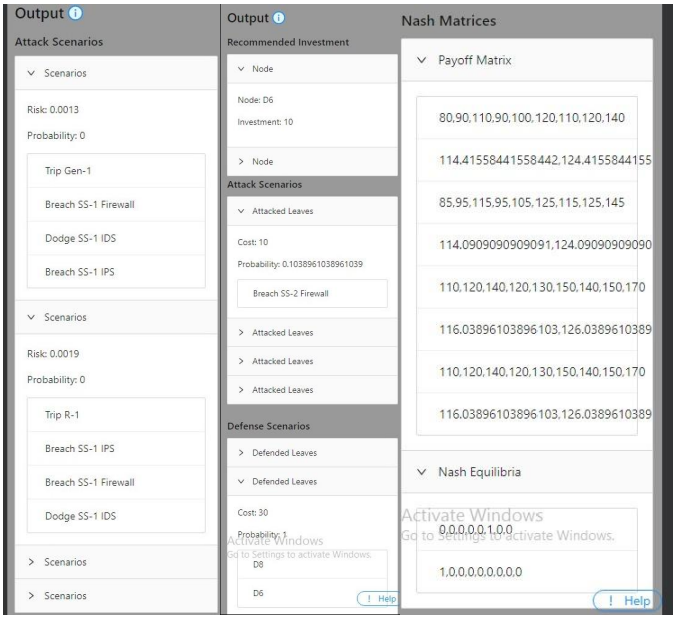

Fig. 10. Attack Tree and Game Theory Engine Outputs: CySec Game Tool

ACKNOWLEDGMENT

This work is funded in part by the NSF CPS grant ECCS 1739969. The authors would also like to acknowledge the contribution of Kush Khanna to develop the requirements for the tool presented in this paper.

REFERENCES

[1] The President's National Infrastructure Advisory Council (NIAC), "Securing Cyber Assets: Addressing Urgent Cyber Threats to Critical Infrastructure," 2017. [Online]. Available: https://www.cisa.gov/

[2] Electric Power Research Institute Palo Alto CA, "Cyber security strategy guidance for the electric sector," 2012. [Online]. Available: https://www.epri.com/research/products/1025672

[3] B. Hyder and M. Govindarasu, "Optimization of cybersecurity investment strategies in the smart grid using game-theory," in *IEEE PES ISGT*, 2020.

[4] R. Leszczyna, "A review of standards with cybersecurity requirements for smart grid," *Computers Security*, 2018.

[5] B. Hyder and M. Govindarasu, "A novel methodology for cybersecurity investment optimization in smart grids using attack-defense trees and game theory," in *IEEE PES ISGT*, 2022.

[6] B. Kordy, P. Kordy, S. Mauw, and P. Schweitzer, "Adtool: Security analysis with attack–defense trees," in *Quantitative Evaluation of Systems*, K. Joshi, M. Siegle, M. Stoelinga, and P. R. D'Argenio, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 173–176.

[7] S. Pudar, G. Manimaran, and C.-C. Liu, "Penet: A practical method and tool for integrated modeling of security attacks and countermeasures," *Computers & Security*, vol. 28, no. 8, pp. 754–771, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404809000522

[8] Y. Wadhawan and C. Neuman, "Rl-bags: A tool for smart grid risk assessment," in *2018 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*, 2018, pp. 7–14.

[9] R. Goel, A. Kumar, and J. Haddow, "Prism: a strategic decision framework for cybersecurity risk assessment," *Information & Computer Security*, vol. 28, no. 4, pp. 591–625, Jan 2020. [Online]. Available: https://doi.org/10.1108/ICS-11-2018-0131

[10] G. Roldán-Molina, M. Almache-Cueva, C. Silva-Rabadão, I. Yevseyeva, and V. Basto-Fernandes, "A comparison of cybersecurity risk analysis tools," *Procedia Computer Science*, vol. 121, pp. 568–575, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917322755

[11] The MITRE Corporation, "CALDERA," 2020. [Online]. Available: https://www.mitre.org/research/technology-transfer/technology-licensing/caldera

[12] CyberX Labs, "CyberX: IoT & ICS Security," 2020. [Online]. Available: https://cyberx-labs.com/

[13] , "CySec Game Tool," 2022. [Online]. Available: https://sdmay21-50.sd.ece.iastate.edu/