International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 4 Volume No.8 July - August – 2024
DOI:10.46647/ijetms.2024.v08i04.006 ISSN: 2581-4621

FOUREYE: DEFENSIVE DECEPTION BASED ON HYPERGAME THEORY AGAINST APTS

[1]Ms. J Sumedha, [2]M. Vaishnavi, [3]G. Sai Sowmya, [4]N. Saakshi

[1]Assistant Professor, Dept. of IT, Stanley College of Engineering and Technology for Women, India
[2]Student, Dept. of IT, Stanley College of Engineering and Technology for Women, India
[3]Student, Dept. of IT, Stanley College of Engineering and Technology for Women, India
[4]Student, Dept. of IT, Stanley College of Engineering and Technology for Women, India

Abstract--Defensive deception techniques have emerged as a promising proactive defense mechanism to mislead an attacker and thereby achieve attack failure. However, most game-theoretic defensive deception approaches have assumed that players maintain consistent views under uncertainty. They do not consider players' possible, subjective beliefs formed due to asymmetric information given to them. In this work, we formulate a hyper game between an attacker and a defender where they can interpret the same game differently and accordingly choose their best strategy based on their respective beliefs. This gives a chance for defensive deception strategies to manipulate an attacker's belief, which is the key to the attacker's decision-making. We consider advanced persistent threat (APT) attacks, which perform multiple attacks in the stages of the cyber kill chain where both the attacker and the defender aim to select optimal strategies based on their beliefs.

Key Words: **Defensive deception, APT, Hyper game.**

I. Introduction

Hyper game theory is a variant of game theory that provides a form of analysis considering each player's subjective belief, misbelief, and perceived uncertainty and accordingly their effect on decision-making in choosing the best strategy. Hyper game theory model players, such as attackers and defenders in cyber security to deal with advanced persistent threat (APT) attacks. First of all, it is not trivial to derive The key purpose of a defensive deception technique is to mislead an attacker's view and make it choose a suboptimal or poor action for the attack failure. When both the attacker and defender are constrained in their resources, strategic interactions can be the key to beat an opponent. In his sense, non-game-theoretic defense approaches have inherent limitations due to lack of efficient and effective strategic tactics. Forms of deception techniques have been discussed based on certain classifications, such as hiding the truth vs. providing false information or passive vs.

A. Problem Statement

Advanced persistent threats (APTs) represent a formidable challenge to cyber security, often outmaneuvering traditional defense mechanisms due to their sophisticated and adaptive nature. Current defense strategies lack the strategic depth to effectively mislead and thwart these attackers. The primary problem is to develop and implement hyper game theory-based defensive deception techniques that can dynamically adapt to the uncertainty and strategic interactions between attackers and defenders, thereby improving the overall security and performance of cyber systems.

B. Aim and Scope

The scope of this project encompasses the development and evaluation of advanced defensive deception techniques using hyper game theory to counter advanced persistent threats (APTs) in cyber security. The project aims to model the strategic interactions between attackers and defenders under uncertainty, incorporating dynamic and subjective beliefs.

II. Existing System

Garg and Grosu proposed a game-theoretic deception framework in honey nets with imperfect information to find optimal actions of an attacker and a defender and investigated the mixed strategy equilibrium. Carroll and Grosu used deception in attacker-defender interactions in a signaling game based on perfect Bayesian equilibria and hybrid equilibria. They considered defensive deception techniques, such as honeypots, camouflaged systems, or The system modeledan attacker's and a defender's uncertainty towards its opponent (i.e., the defender and the attacker, respectively) based on how long each player has monitored the opponent and its chosen strategy. To the best of our knowledge, prior research on hyper game theory uses a predefined constant probability to represent a player's uncertainty. In this work, we estimated the player's uncertainty based on the dynamic, strategic interactions between an attacker and a defender. The system conducted comparative performance analysis with or without a defender using defensive deception (DD) strategies and with or without perfect knowledge available towards actions taken by the opponent. We measured the effectiveness and efficiency of DD techniques in terms of a system's security and performance, such as perceived uncertainty, hyper game expected utility, action cost, mean time to security failure (MTTSF or system lifetime), and improved false positive rate (FPR) of an intrusion detection by the DD strategies taken by the defender.

A.Drawbacks

The system can't track attack which can be performed to exploit unknown vulnerabilities of software, which are not patched yet. The system can't track Fake identity attack which can be performed when packets are transmitted without authentication or internal nodes spoofing the ID of a source node.

III. Proposed System

The system modeled an attack-defence game under uncertainty based on hyper game theory where an attacker and a defender have different views of the situation and are uncertain about strategies taken by their opponent. The system reduced a player's action space by using a sub game determined based on a set of strategies available where each sub game is formulated based on each stage of the cyber kill chain (CKC) based on a player's belief under uncertainty. The system considered multiple defence strategies, including
defensive deception techniques whose performance can be significantly affected by an attacker's belief and perceived uncertainty, which impacts its choice of strategy.

The system modeled an attacker's and a defender's uncertainty towards its opponent (i.e., the defender and the attacker, respectively) based on how long each player has monitored the opponent and its chosen strategy. To the best of our knowledge, prior research on hyper game theory uses a predefined constant probability to represent a player's uncertainty. In this work, we estimated the player's uncertainty based on the dynamic, strategic interactions between an attacker and a defender. The system conducted comparative performance analysis with or without a defender using defensive deception (DD) strategies and with or without perfect knowledge available towards actions taken by the opponent. We measured the effectiveness and efficiency of DD techniques in terms of a system's security and performance, such as perceived uncertainty, hyper game expected utility, action cost, mean time to security failure (MTTSF or system lifetime), and improved false positive rate (FPR) of an intrusion detection by the DD strategies taken by the defender.

B.Advantages

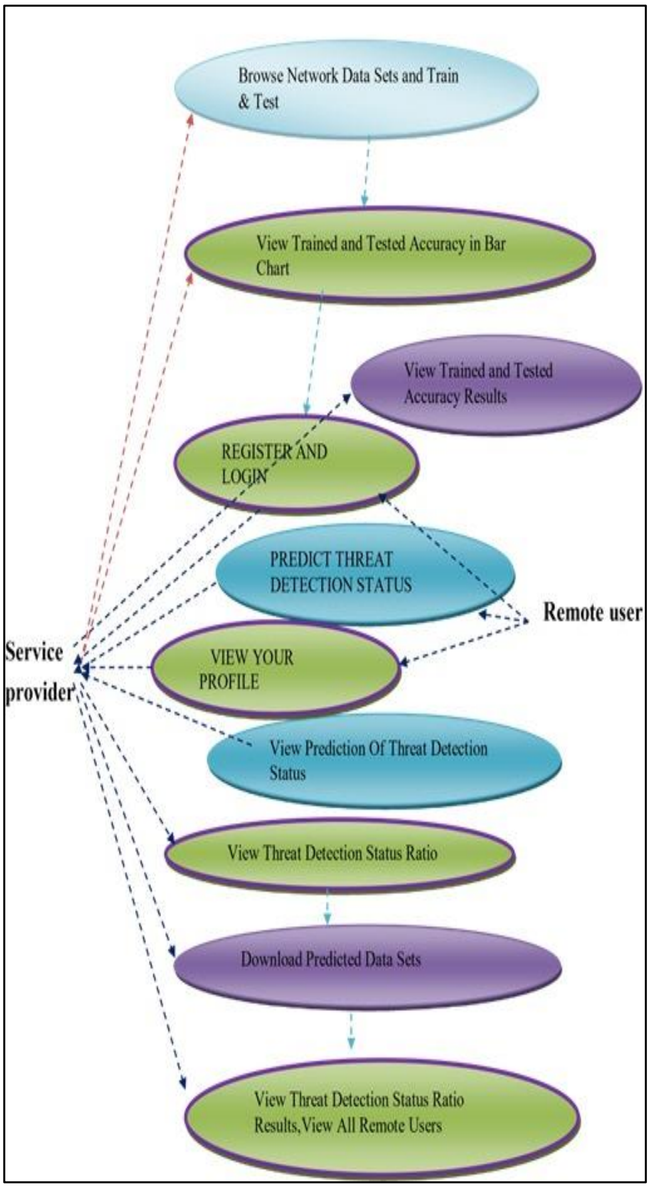
1. APT Attack Procedure to Achieve Data Exfiltration in which the system define an APT attacker's goal in that the attacker has reached and compromised a target node and successfully exfiltrated its confidential data.
2. The system proposed many ML Classifiers to test and train the different types of attacks and can be predicted by using same classifiers.

IV. System Architecture

1. SERVICE PROVIDER: In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Browse Network Data Sets and Train & Test, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Threat Detection Status, View Threat Detection Status Ratio, Download Predicted Data Sets, View Threat Detection Status Ratio Results, View All Remote Users.

2. VIEW AND AUTHORIZE USERS: In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name email, address and admin authorizes the users.

REMOTE USER: In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT THREAT DETECTION STATUS, VIEW YOUR PROFILE


Browse Network Data Sets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
REGISTER AND LOGIN
PREDICT THREAT DETECTION STATUS
Remote user
Service provider
VIEW YOUR PROFILE
View Prediction Of Threat Detection Status
View Threat Detection Status Ratio
Download Predicted Data Sets
View Threat Detection Status Ratio Results,View All Remote Users

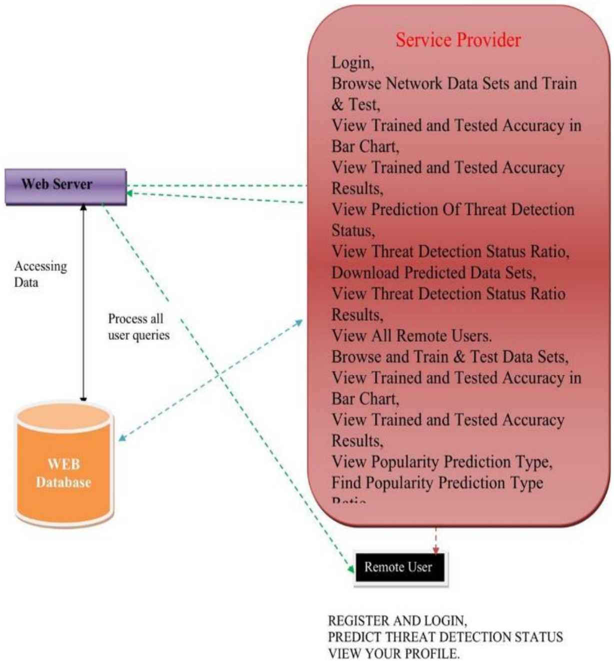

Service Provider
Login,
Browse Network Data Sets and Train & Test,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Threat Detection Status,
View Threat Detection Status Ratio,
Download Predicted Data Sets,
View Threat Detection Status Ratio Results,
View All Remote Users.
Browse and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Popularity Prediction Type,
Find Popularity Prediction Type Ratio
Web Server
Accessing Data
Process all user queries
WEB Database
Remote User
REGISTER AND LOGIN,
PREDICT THREAT DETECTION STATUS
VIEW YOUR PROFILE.

Fig: Architecture Diagram

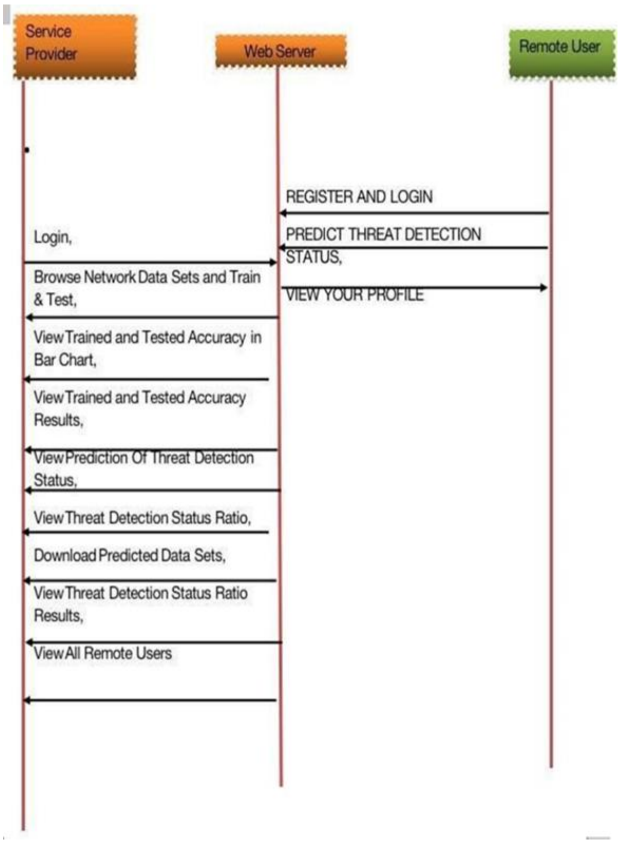

Service Provider
Web Server
Remote User
REGISTER AND LOGIN
PREDICT THREAT DETECTION STATUS,
VIEW YOUR PROFILE
Login,
Browse Network Data Sets and Train & Test,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results, 55,
View Prediction Of Threat Detection Status,
View Threat Detection Status Ratio,
Download Predicted Data Sets,
View Threat Detection Status Ratio Results,
View All Remote Users

Fig: Sequence Diagram

V.Feature Extraction

Feature extraction is a crucial step in threat detection, involving identifying and selecting relevant data points or characteristics from raw data. A structured approach involves understanding the threat landscape, data collection, preprocessing the data, identifying features, enhancing the raw features, selecting the most relevant features, validating features, building detection models, and continuously improving. To extract features, one must understand the threat landscape, gather data from various sources, such as network traffic logs, system logs, security event logs, application logs, user behavior analytics (UBA) data, and endpoint detection and response (EDR) data. Preprocessing the data involves cleaning and normalizing it, identifying features.

Feature selection involves using techniques such as correlation analysis, mutual information, Principal Component Analysis (PCA), and feature importance from models. Validating features involves testing them with historical threat data and analyzing their performance using cross-validation techniques. Building detection models involves using supervised learning techniques, unsupervised learning, or hybrid approaches. Continuous improvement involves incorporating new threat intelligence and feedback from detection performance to fine-tune features and models.

VI. Algorithms

Decision Tree Classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision-making knowledge from the supplied data. Decision tree can be generated from training sets. A decision tree classifier is a valuable tool in threat detection projects due to its simplicity and interpretability. The process begins with data collection and preparation, where data related to potential threats is gathered from various sources like network and system logs. Relevant features are then extracted, such as IP addresses and access patterns, and the data is labeled as "threat" or "non-threat." During feature selection, key features that effectively distinguish between threats and non-threats are identified.

The decision tree classifier is trained on the training data to create rules that separate threats from non-threats, followed by pruning to avoid overfitting. Model evaluation involves testing the model on the testing set and using metrics like accuracy, precision, recall, and the F1-score to assess performance, along with a confusion matrix to visualize the results.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias). While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique. Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra.

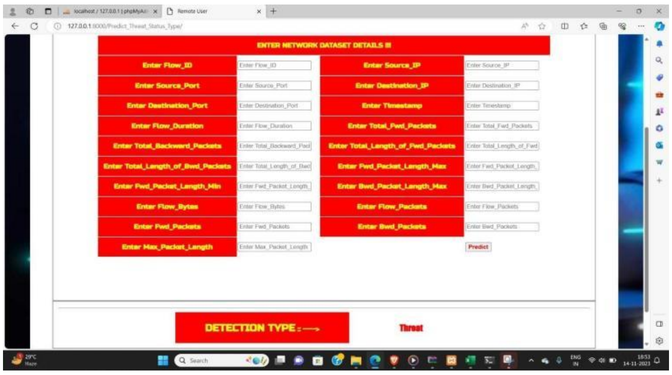
SVM

Support Vector Machines (SVM) are effective for threat detection due to their capability to handle high-dimensional data and create complex decision boundaries. The process begins with data collection and preparation, where data from sources like network logs, system logs, and user activity records are gathered and labeled as "threat" or "non-threat" based on historical data or expert knowledge. Relevant features such as login times, IP addresses, and access patterns are extracted. Feature selection involves choosing the most indicative features through statistical techniques and domain expertise.

In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed(iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. SVM is a discriminant technique because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (Gas) or perceptron's, both of which are widely used for classification in machine learning. For perceptron's, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptron's is only to minimize error during training, which will translate into several hyperplane's meeting this requirement.

VII. Result

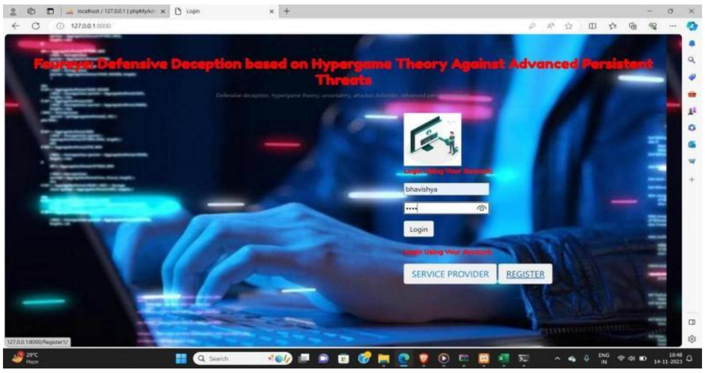
The threat detection project aims to develop a system for identifying and responding to potential threats such as malware, intrusions, and unauthorized access. The system will collect data from various sources like sensors, network logs, and user activity logs, which will then be normalized and processed. Detection algorithms, including anomaly detection and machine learning models, will analyze this data to identify threats. An alerting system will notify administrators of any detected threats, while response mechanisms, either automated or manual, will be established to handle them. The project will involve integrating data sources, developing detection algorithms, setting up necessary infrastructure, and designing user interfaces for effective monitoring and management.


localhost / 127.0.0.1 | phpMyAdm
Remote User
127.0.0.1:8000/Predict_Threat_Status_Type/
Predict
Search

ENTER NETWORK DATASET DETAILS !!!

Enter Flow_ID	Enter Flow_ID	Enter Source_IP	Enter Source_IP
Enter Source_Port	Enter Source_Port	Enter Destination_IP	Enter Destination_IP
Enter Destination_Port	Enter Destination_Port	Enter Timestamp	Enter Timestamp
Enter Flow_Duration	Enter Flow_Duration	Enter Total_Fwd_Packets	Enter Total_Fwd_Packets
Enter Total_Backward_Packets	Enter Total_Backward_Pacl	Enter Total_Length_of_Fwd_Packets	Enter Total_Length_of_Fwd
Enter Total_Length_of_Bwd_Packets	Enter Total_Length_of_Bwd	Enter Fwd_Packet_Length_Max	Enter Fwd_Packet_Length_
Enter Fwd_Packet_Length_Min	Enter Fwd_Packet_Length_	Enter Bwd_Packet_Length_Max	Enter Bwd_Packet_Length_
Enter Flow_Bytes	Enter Flow_Bytes	Enter Flow_Packets	Enter Flow_Packets
Enter Fwd_Packets	Enter Fwd_Packets	Enter Bwd_Packets	Enter Bwd_Packets
Enter Max_Packet_Length	Enter Max_Packet_Length		Predict

DETECTION TYPE :---> Threat



Foureye: Defensive Deception based on Hypergame Theory Against Advanced Persistent Threats

Defensive deception, hypergame theory, uncertainty, attacker,defender, advanced per

Login Using Your Account

bhavishya

Login Using Your Account

VIII. Conclusion

An attacker's and defender's perceived uncertainty can be reduced when defensive deception (DD) is used. This is because the attacker perceives more knowledge about the system as it performs attacks as an inside attacker. On the other hand, the defender's uncertainty can be reduced by collecting more attack intelligence by using DD while allowing the attacker to be in the system.

Attack cost and defence cost are two critical factors in determining HEUs (hyper game expected utilities). Therefore, high DHEU (defender's HEU) is not necessarily related to high system performance in MTTSF (mean time to security failure) or TPR (true positive rate) which can also be a key indicator of system security. Therefore, using DD under imperfect information (IPI) yields the best performance in MTTSF (i.e., the longest system lifetime) while it gives the minimum DHEU among all schemes.DD can effectively increase TPR of the NIDS in the system based on the attack intelligence collected through the DD strategies.

IX. References

[1] "Common vulnerability scoring system (CVSS)."[Online].
Available: https://www.first.org/cvss/ Y. M. Aljefri, M. A. Bashar, L. Fang, and k. W. Hipel,"First-level hyper game for investigating misperception in conflicts," IEEE Trans. Systems, Man, and Cybernetics: Systems, vol. 48, no. 12, pp. 2158–2175, 2017.

[2] H. Almeshekah and H. Spafford, "Cyber security deception," in Cyber Deception. Springer, 2016, pp. 25–52.

[3] J. W. Caddell, "Deception 101-primer on deception," DTIC Document, Tech. Rep.,2004

[4] T. E. Carroll and D. Grosu, "A game theoretic investigation of deception in network security," Security and Communication Networks, vol. 4, no. 10, pp. 1162– 1172, 2011.

[5] N. M. Fraser and K. W. Hipel, Conflict Analysis: Models and Resolutions. North-Holland, 1984.

[6] GmbH. MindNode. [Online]. Available: https: //mindnode.com/

[7] J. Han, J. Pei, and M. Kamber, Data Mining: Concepts and Techniques. Elsevier, 2011

[8] H. Almeshekah and H. Spafford, "Cyber security deception," in Cyber Deception. Springer, 2016, pp. 25–52.

[9] U. Brandes, "A faster algorithm for betweenness centrality," Jour. mathematical sociology, vol. 25, no. 2, pp. 163–177, 2001.

[10] B. Gharesifard and J. Cort´es, "Evolution of the perception about the opponent inhypergames," in Proc. 49th IEEE Conf. Decision and Control (CDC), Dec. 2010, pp. 1076–1081.