IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-9427-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOMWKSHPS57453.2023.10225954

CICADA: Cloud-based Intelligent Classification and Active Defense Approach for IoT Security

[1]Roshan Lal Neupane, [2]Trevor Zobrist, [1]Kiran Neupane, [3]Shaynoah Bedford, [4]Shreyas Prabhudev,
[1]Trevontae Haughton, [5]Jianli Pan, [1]Prasad Calyam
[1]Uni. of Missouri-Columbia, [2]Southeast Missouri State Uni., [3]Uni. of the Virgin Islands, [4]Uni. of California at San Diego,
[5]George Mason Uni.; Email: {neupaner, kngbq, thgtx, calyamp}@missouri.edu, tmzobrist1s@semo.edu,
shaynoah.bedford@students.uvi.edu, sprabhudev@ucsd.edu, jpan22@gmu.edu

***Abstract*—Internet of Things (IoT) devices capture and process sensitive personally identifiable information such as e.g., camera feeds/health data from enterprises and households. These devices are becoming targets of prominent attacks such as Distributed-Denial-of-Service (DDoS) and Botnets, as well as sophisticated attacks (e.g., Zero Click) that are elusive by design. There is a need for cyber deception techniques that can automate attack impact mitigation at the scale that IoT networks demand. In this paper, we present a novel cloud-based active defense approach viz., "CICADA", to detect and counter attacks that target vulnerable IoT networks. Specifically, we propose a multi-model detection engine featuring a pipeline of machine/deep learning classifiers to label inbound packet flows. In addition, we devise an edge-based defense engine that utilizes three simulated deception environments (Honeynet, Pseudocomb, and Honeyclone) with increasing pretense capabilities to deceive the attacker and lower the attack risk. Our deception environments are based on a CFO triad (cost, fidelity, observability) for designing system architectures to handle attacks with diverse detection characteristics. We evaluate the effectiveness of these architectures on an enterprise IoT network setting with a scale of thousands of devices. Our detection results show ≈73% accuracy for the low observability attack (Zero Click) corresponding to the BleedingTooth exploit that allows for unauthenticated remote attacks on vulnerable devices. Furthermore, we evaluate the different deception environments based on their risk mitigation potential and associated costs. Our simulation results show that the Honeyclone is able to reduce risk by ≈88% when compared to a network without any defenses.**

***Index Terms*—IoT security automation, attack detection, machine learning, cyber deception, active defense**

I. Introduction

With the growing network of enterprise IoT, WiFi-enabled smart devices have the potential to greatly improve the quality of life for consumers [1]. Voice-activated speakers, smart fridges, smart lights, and other such smart devices have created a new IoT ecosystem. In 2018, eMarketer reported that 23.1% of United States enterprise networks actively used smart devices, predicting an increase to 46.5% by 2023 [2]. However, with the addition of smart devices, IoT security loopholes have arisen that can be exploited by attackers. For example, Philips recently discovered that their IoT interface for interacting with patient medical data had security flaws that exposed databases containing sensitive records [3].

Given that smart devices typically use custom protocols, there are added challenges to secure related IoT systems and prevent threats from adversaries [4]. Moreover, if the threats are evolving and dynamic, IoT systems may be incapable of effective defense due to the use of prevalent static defense measures such as e.g., passwords, encryption or firewalls. In order to secure IoT networks at large-scale and automate their defense in an evolving threat landscape, new active defense mechanisms are direly needed. These active defense mechanisms need to have the ability to effectively handle prominent (relatively easy to detect) attacks such as the DDoS and Botnets. At the same time, they also need to handle sophisticated (relatively hard to detect) attacks such as e.g., Zero Click [5], where attackers can inject code into a network packet or figure out a way to exploit a network function to take control of a device [6]. Lastly, they need to be practically usable in terms of cost and fidelity (or capabilities) depending on the risk factors involved in different attacks on a given IoT system configuration, and level of pretense desired.

In this paper, we propose a Cloud-based Intelligent Classification and Active Defense Approach viz., "CICADA" to defend IoT systems using decoy environments that implement the "defense by pretense" paradigm of active defense [7] [8]. The goal of CICADA is to offload the burden of security from the resource-constrained smart devices to cloud computing resources. Our CICADA approach involves attack classification using a multi-model detection engine featuring a pipeline of machine/deep learning classifiers to label inbound packet flows. In addition, our approach focuses on selection of a pertinent decoy environment to redirect attack traffic based on the attack observability, in order to effectively deceive the attacker and lower the attack risk on an actual production environment with legitimate user traffic.

Using a CFO triad (cost, fidelity, observability) for designing the system architectures for the deception environments, we handle attacks with low (e.g., Zero Click), medium (C&C, Portscan, Man-in-the-Middle) and high (DDoS, Botnet) observability characteristics. The 'Honeynet' decoy environment represents the lower end of the CFO triad factors, whereas the 'Pseudocomb' and 'Honeyclone' represent the middle and higher end of the CFO triad factors, respectively. The multi-model detection engine in CICADA comprises an ensemble of various ML/DL based models that cover different levels of threats in order to identify attacks based on their observability and feed relevant information to a defense engine that implements the active defense methods using the decoy environments. We assume that - with decreasing order of observability, the cost of employing effective deceptive defense methods (as given by a cost model) rises, and so does the required fidelity to lower the risk (obtained by performing a risk assessment) of threats on an enterprise IoT system.

This material is based upon work supported by the National Science Foundation under award number CNS-1950873, and the National Security Agency under award number H98230-21-1-0260. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the National Security Agency.

Authorized licensed use limited to: George Mason University. Downloaded on October 07,2024 at 20:39:28 UTC from IEEE Xplore. Restrictions apply.
978-1-6654-9427-4/23/$31.00 ©2023 IEEE

We implement and evaluate our CICADA using a realistic enterprise IoT network testbed built on the Amazon Web Services (AWS) cloud platform. For our experiment purposes, we consider: a) real-world datasets such as the Aposemat IoT-23 [9] from Stratosphere Labs, and an IoT Network Intrusion Dataset [10] collected from IEEE Dataport, and b) simulated dataset for low observability traffic, repurposed from the GHOST-IoT [1] dataset's benign Bluetooth encounter traffic. Through our experiment results, we show how CICADA enables attack detection for various attack observability levels using ensemble and binary models built using multi-layered neural networks and classifiers such as Random Forest, Extra Trees with high accuracy. Furthermore, we evaluate the different deception environments based on their risk mitigation potential and associated costs, and show how the three decoy environments (i.e., Honeynet, Pseudocomb, Honeyclone) perform relatively to handle an attack, when compared to a network without any defenses.

The remainder of the paper is organized as follows. Section II presents the related work. Section III describes our active defense methodology. Section IV discusses the performance evaluation of CICADA system implementation on an enterprise IoT network testbed. Section V concludes the paper.

II. Related Work

Intrusion Detection Systems (IDS) have emerged as a solution to the problem of smart device security within large-scale IoT-based networks [11]. Authors in [12] explain the current challenges that these IoT IDS face and how ML can be applied effectively in a detection model. Both [13] and [14] propose packet-level IDS models for IoT networks that are similar to the models we use in our work that uses multiple models, each with a certain targeted purpose.

The attack surface within large IoT-based networks is large and ever increasing as more sectors adopt smart devices with high inter-connectivity [11]. Research works such as [15], [16] highlight the presence of massive attack surfaces and the increasing amount of data being generated from modern smart devices. With the inclusion of AI-based attack methods by threat actors, numerous attacks can be generated in a very short amount of time [17]. Above prior works are mostly limited to botnet malware, such as the popular "Mirai" botnet and its variations, or other well-known attacks: DDoS, MITM, C&C.

Active defense, or the use of offensive tactics to outsmart the attacker for example, by using honeypots [18] and/or redirection/containment [19] can improve the security of a system indirectly by deflecting attackers away from a legitimate system. Active defense methods can slow down a hacker and make cyber attacks more difficult to carry out [20], by increasing the attack budget. In the case of attacker-defender games, as highlighted in [21], when the attacker does not realize deception is apart of the defender's defense strategy, security is increased without extra resource expenditure. This reasoning makes sense due to the fact that deception is only effective when the attacker is able to be deceived.

Novelty of our work lies in cloud-based intrusion detection of various attacks based on observability for modern and sophisticated attacks such as Zero Click attacks (low observability) which are not studied in depth in prior works. We also consider mapping of different deception environments given attack observability and provide pertinent decoy environment architectures to deceive attackers.

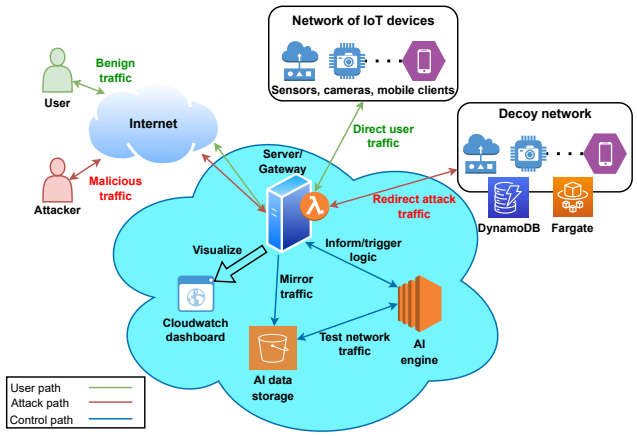

Fig. 1: Overview of CICADA residing on edge network with deception capabilities and AI engine for cyber detection and defense.

III. Active Defense Methodology

A. CICADA Architecture

The CICADA operates at the cloud for offloading security from resource-constrained smart devices as shown in the Figure 1. Any traffic (relating to both benign and attack traffic) that tries to access the underlying IoT-based network passes through a server/gateway. The related network traffic is then analyzed by our Detection Engine (DE) that is part of the collective AI Engine to detect possible cyber attacks. If the binary classifier within our DE classifies packets as benign, then the benign traffic is forwarded to a legitimate network segment. Whereas, once the traffic is classified as malign, the traffic is further analyzed in depth by the ensemble network to classify the flow as a specific threat. This aids an Active Defense Engine (ADE) to identify the suitable type of decoy environment (Honeynet, Pseudocomb or Honeyclone) to use depending on the observability of the attack type enabled through Lambda (λ) functions and load balancers.

In order to design our ADE, we consider how much realism does a decoy environment needs to possess, for it to be: (a) beneficial to the defender in terms of attack impact mitgation, and (b) also cost-effective. Traditional static security-based environments with e.g., passwords or firewalls could be a low-cost solution for mitigating simple attacks but they do not provide enough evidence to counter more sophisticated/modern attacks. Similarly, near-replica environments could provide a complex environment capable of causing a threat actor to reveal their attack details but the cost and maintenance requirements might be too high for consistent/parallel implementation. In order to provide a comparative measure between deception environments, we developed a methodology, similar to [8], to be able to map them based on the environment's individual cost, fidelity, and observability. Each of these factors are mapped out as a *CFO Triad*, and we have established three environments for different levels of observability. They are:

a) Honeynet: Environment with the lowest cost, requiring relatively less computation, networking and storage resources. This environment is designed to handle high observability attacks such as DDoS and Portscan.

b) Pseudocomb: Uses more resources than a Honeynet, but at an added cost. They tend to provide minimalistic realism of the actual operating environment to balance all three factors.

Authorized licensed use limited to: George Mason University. Downloaded on October 07,2024 at 20:39:28 UTC from IEEE Xplore. Restrictions apply.

c) Honeyclone: Honeyclone is a near-replica of the actual environment with the highest fidelity and observability at the expense of high cost. The cost of maintaining a Honeyclone is high because the data stream, computational resources, and infrastructure are designed to mirror a legitimate user traffic system in terms of realism.

B. Detection Engine

The detection engine in CICADA features a pipeline of ML/DL classifiers to label inbound packet flows.

1) Feature Selection: We separated individual text files from each data set into different subsets: easy, medium, and hard - based on the attack's observability characteristics. DDoS and Botnet (Mirai, Okiru) traffic varies largely from benign traffic and thus they make up the Easy subset. These attacks have pronounced signatures such as an excessive number of packets sent at a time or specific port communications that give away the malicious intent with a fewer number of features being observed. More covert attacks such as Man-in-the-Middle, portscanning, and command-and-control transmissions have obfuscated movements in the network, and hence they comprise the Medium subset. These attacks have features that can overlap with benign traffic in some scenarios but require more observed features to be correctly labeled as malicious. Modern, sophisticated intrusions such as Zero Click attacks, that use protocol-specific vulnerabilities to gain access can appear indistinguishable from benign traffic, and hence these attacks make up the Hard subset.

2) Model Details: The Detection Engine (DE) included in our CICADA is a multi-model method to classify incoming network traffic based on packet-level analysis. The DE consists of a neural network binary classifier that first classifies the traffic as 'benign' or 'malicious'. The binary classifier is a 8-layer (6 hidden layers) deeply connected neural network that takes the 8 input feature values through a MinMax scalar function and then feeds them through layers with 32 nodes, 64 nodes, 24 nodes, and consecutive dropout layers, and finally an output layer of 1 node. This node outputs a value with sigmoid activation that is either 0 or 1 based on the inbound network flow is either 'benign' or 'malicious'. Dropout layers were added with a 0.2 drop rate to prevent over-fitting of the model on the training set. Parameters like epochs and batch size were refined using the hyperparameter tuning methods. The binary classifier uses 320 epochs and a batch size of 64021 (equal to the full size of the training set). 'Malicious' classified flows are pipelined in our ensemble multi-class model to determine the inherent type of threat.

Our ensemble model is the latter part of CICADA's multi-model detection engine. The ensemble consists of a 8-layer (6 hidden layers) deeply connected neural network that uses 16, 64, 24 hidden layers, and an output layer of 5 (consisting 5 classes). This multi-class neural network also uses 320 epochs, a batch size of 64021, and a MinMax scalar function on the input values before classification. We also use Random Forest and Extra Trees classifiers. These machine learning algorithms were implemented with the `sklearn` python library [22]. The hyperparameters for both `sklearn` algorithms were set to the default except for the fields 'max_depth', 'max_features', 'n_estimators', 'class_weight', and 'n_jobs'. When a 'Malicious' classified flow proceeds through the ensemble model, the flow is sent to each algorithm in parallel and the prediction of each flow is gathered. If the collective algorithm's predictions are *unanimous* or *consensual*, the flow is labeled as the concurred classification. If the collective algorithm's predictions are *non-consensual*, the flow is not labeled as a specific threat. This allows for more precise classification as opposed to using only one multi-class model.

C. Active Defense Engine

Our Active Defense Engine (ADE) is designed with the goal of intelligently routing network traffic based on the analysis of the DE. If the network traffic coming-in is classified as benign, the traffic will continue to the legitimate network as intended. Once the DE classifies a packet as malicious, it will continue to the ensemble of multi-class classifiers to be analyzed and classified as a specific type of malware. There is no opportunity for a packet to go to the IoT-based network once it is classified as malicious. Once the packet is specified based on what type of attack it is, the ADE will follow a procedure to either get rid of the packet so it can not reach the network or send it to a decoy environment for further evaluation and reconnaissance. If the classified malware is a singular occurrence or an intelligence gathering attack, then the packet will simply get dropped and never make it to the actual network. If the classified malware is deemed as escalatable (i.e., attack is as a catalyst for more harmful attacks), it will be sent to one of the three decoy environments based on its observability so that more packet captures can be performed to better understand how current threat actors are infiltrating the IoT system, and to gather more data to retrain the model. In the case that the malware can not be classified by the classification model, the default response is to send the traffic to a Pseudocomb so that more information can be gathered on the attack to eventually classify it in a future retraining event.

Herein, we discuss the cost and risk analysis we performed on IoT systems that gives insight on how the usage of different decoy environments can impact architecture choice for the decoy environment.

1) AWS-based Cost Analysis: We calculate the baseline operational cost for cloud/edge services for processing of traffic and making decisions that considers the cost of deployment of detection engine, analysis service, monitoring service, VPC traffic mirroring, and storage that are given by C_d, C_a, C_m, C_t and C_s respectively. Following this, the total operational cost is given by:

$$C_{operation} = C_d + C_a + C_m + C_t + C_s \tag{1}$$

C_d is based on the Sagemaker cost for N number of data analysis notebooks per scientist, which is also given by:

$$C_d = N * s * c_i \tag{2}$$

where n is the number of notebooks, s is the number of scientists and c_i is the cost of instance type. The C_a is given by:

$$C_a = q_n * d \tag{3}$$

where q_n is number of queries, and d is the amount of data scanned per query. C_m is based on usage of the CloudWatch service for monitoring metrics such as CPU usage, memory usage, etc., and is calculated as:

$$C_m = m_n * \lambda_n + \lambda_r * \lambda_n + a_n * c_a \tag{4}$$

where, m_n is the number of metrics, λ_n is the number of AWS Lambda functions, λ_r denotes the number of requests made on those Lambda functions, a_n is the number of alarms and

Authorized licensed use limited to: George Mason University. Downloaded on October 07,2024 at 20:39:28 UTC from IEEE Xplore. Restrictions apply.

c_a is the cost per alarm. C_t defines the cost for VPC traffic mirroring, and is given by:

$$C_t = t_n * h * c_{sh} \tag{5}$$

where t_n is the traffic mirroring sessions, h is the hours per day and c_{sh} is the cost per session-hour. Lastly, C_s gives the storage requirement for the operation of CICADA.

Next, we calculate the cost associated with creating the decoy environment networks. Basing on the type of decoy environments viz., Honeynet, Pseudocomb, and Honeyclone, we identify the number of servers and Pseudocombs (because a Honeyclone could have multiple Pseudocombs) to treat as smart devices and different production servers on the enterprise IoT-based network. The cost for setting these up is given by:

$$C_{decoy} = (p_n * a_d * v_{ch} * v_{cpu}) + (p_n * a_d * m_{ch} * m_{alloc}) \tag{6}$$

Here, the p_n is the number of instances, a_d is average duration each instances are running, v_{cpu} is the number of vCPUs, m_{alloc} is the memory allocated to the servers. v_{ch} and m_{ch} is the cost per hour of vCPU and memory allocation, respectively.

The Honeyclone is a collection of different servers within an enterprise which means that it will consist of a number of interlinked Pseudocombs. The communication of different services with each other occurs through API calls and regular Internet traffic. We mirror the entire traffic of the production environment into the decoy environment of Honeyclone. The cost associated with this will be based on the number of Pseudocombs and is directly related to size of the organization. If cost of deploying a Pseudocomb for a certain service is given by P_{decoy}, the cost of deploying a Honeyclone is given by:

$$H_{decoy} = \sum_{i=1}^{n} P^i_{decoy} + C_{load} \tag{7}$$

where C_{load} is the load generated across the different services within the enterprise network and $i \in 0, 1, 2...n$ for n number of Pseudocombs.

2) CICADA Risk Assessment: We use the methodology in the NIST risk assessment guideline [23] to calculate the potential risk levels for various threats impacting an enterprise IoT-based network being secured by CICADA defense architectures. The NIST methodology populates the impact values and likelihood values for specific threats being considered. The impact values are derived from assessed potential impact resulting from a compromise of the confidentiality, integrity, or availability for any information type due to security threats. The likelihood values are a weighted factor based on a subjective analysis of the probability that - a given threat is capable of exploiting an exposed vulnerability. Following this, the overall risk values are calculated for different CICADA architectures by factoring the likelihood and impact scores, which are finally normalized into a quantitative scale of 0-10. These ranges for scales are: 9-10 indicating very high risk, 7-8 indicating high risk, 4-6 indicating moderate risk, 1-3 indicating low risk, and 0 indicating very low level of risk. We present the results of determining the risk levels for different threat events described later in Section IV-D This risk assessment guided us to catalog the different attacks to associate them with choice of the decoy environment to handle a given attack with an associated observability level along with cost considerations.

IV. Performance Evaluation

In this section, we present evaluation results in terms of detection engine's accuracy. Following this, we present results on cost analysis and risk assessment to help in choosing the pertinent decoy environment for a given attack observability.

A. Trace Datasets

To have relevance to the IoT-based enterprise networks we are studying, we utilized data sets such as Aposemat IoT-23 [9] and IoT Network Intrusion Dataset [10] that contain packet captures of malware intrusions within a simulated IoT-based network along with benign traffic from common smart devices, such as the Amazon Echo smart device, and Somfy smart door lock. These data are the PCAP captures that contain 20 captures of malware executed in the aforementioned smart devices along with 3 captures of benign smart device traffic containing 760 million packets and 325 million labeled flows over more than 500 hours.

As a dataset for Zero Click vulnerabilities that feature low observability traffic, we repurposed the GHOST-IoT [1] dataset's benign Bluetooth encounter traffic to simulate traffic expected from Bluetooth Zero Click vulnerabilities that have been observed in the past (e.g., BleedingTooth, BadKarma, BadVibes). This involved sampling entries from the set and relabeling them to be malicious while keeping benign-like feature characteristics. Further, we discarded any features that contained a majority of null (NaN) values, or were specific to the researcher's collection method, or those that did not impact the model's decision making.

B. Detection Engine Performance

The binary classifier was trained with a 60% split of the complete data set (i.e., benign traffic, easy/medium/hard subsets) using 10-fold cross validation and then tested on the remaining 40% to get cross validation results (along with standard deviation of each fold) and test subset results. The fitted binary classifier was also used to classify each subset to get individual performance results based on observability.

The ROC curve for binary NN is shown in Figure 2. We compare the neural network with a baseline model as a standard threshold of 0.5 probability for detecting a particular type of network traffic. The predicted values from the binary neural network model shows that these values are correctly recognized up to the accuracy of 90%. The predicted values show true positives upto 70% and then there are some values that show a false positive, after which the model gradually labels other packets in the true positive/false positive as each packet information is introduced. The ideal situation is present in the Multi-class Neural Network where the true positive rates are up to 1.0 for Botnet attacks. The model correctly predicts all of the packet information for Botnet attacks, gradually decreasing the prediction accuracy based on the traceability of these attacks. The threshold for Zero Click attacks settles at 76% as these attacks are highly elusive by nature. The decision tree classifiers have a higher predictability in comparison to neural network because the biases of each feature are not assigned. Consequently, we use hyperparameter tuning to improve configuration of these biases in the neural network. From these results, we can see the benefits of performing the hyperparameter tuning for setting the correct threshold for biases. Note that the ensemble of multi-class classifiers were trained and tested in the same manner but without the inclusion

Authorized licensed use limited to: George Mason University. Downloaded on October 07,2024 at 20:39:28 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Cost analysis for deployment of CICADA Architecture for varying deception environment configurations

Capability	AWS Service	Specification	Cost ($) per month		
			Honeynet	Pseudocomb	Honeyclone
Decoy IoT Network	Fargate, EC2	1 server with 1 port, 1 pseudocomb, 10 pseudocombs respectively; EC2 with 2 vCPUs and 4 GB memory allocation	72.09	1081.20	10812.00
Load Generator	VPC, API Gateway	Mirror load with 730 hours per session	109.50	1642.50	16425.00
Network Behavior Analysis	CloudWatch, VPC	Packet capture using mirroring filters and 15 analytical metrics for traffic entering, moving, and leaving server/s	N/A	1925.55	19255.50
Cross-service Load Generator	VPC, API Gateway	Mirror packets generated across different pseudocombs in an organizational network 730 hour per session	N/A	N/A	8760.00
Data Storage	DynamoDB	150 GB of storage size per pseudocomb with 30% transactional data excluding initial cost of $350	N/A	53.93	539.3
Honey Token	SpaceSiren	Limit cap of 10000 honey tokens to lure attackers per server in use [24]	N/A	5	50
			$2179.08	**$56,498.16**	**$670,735.60**

TABLE II: Detection engine performance for varying network data subsets based on malware observability

Data Subset	Binary NN	Multi-Class NN	RF	ET
Test Subset	90.16%	96.39%	98.98%	98.39%
Easy Subset	99.99%	99.95%	99.99%	99.98%
Medium Subset	84.41%	86.12%	98.89%	93.71%
Hard Subset	74.33%	53.41%	87.93%	80.13%

TABLE III: Detection engine performance for 10-fold cross validation

Metric	Binary NN	Multi-Class NN	RF	ET
Avg Accuracy	90.01%	95.44%	99.01%	98.27%
Std Deviation (σ)	0.68%	1.94%	0.08%	0.58%

of benign traffic due to the assumption that benign traffic gets filtered out by the binary classifier.

Table II shows the accuracy results for the models used in our detection engine. The Table III shows the average accuracy and the standard deviation for 10-fold cross validation of the utilized dataset. As expected, high observability (Easy) attacks that differ greatly from benign traffic were easily classified with an accuracy of 99%+ by the binary classifier along with every model within the ensemble. Medium observability (Medium) attacks, with less variance from benign traffic, lowered the accuracy of the models; by up to 15% for the binary classifier. This makes sense because the malware looks more like benign traffic in some cases and thus the models will classify more entries as either false positives or false negatives and lower the accuracy. The accuracy of medium observability malware within the ensemble still lowers but not as drastically as the binary classifier. The ensemble takes only assumed malicious traffic and hence the difference between entries is greater than, if benign traffic was also included. This phenomenon is even more extreme within the low observability (Hard) subset where the malicious traffic is designed to look identical to the benign traffic. The composition of the hard subset is 77% benign and thus this result shows that the Zero Click confuses the model into not accurately discriminating between the two. When benign traffic is not included in the ensemble models, the accuracy is better i.e., above 80% for two of the models, since the malware is being compared to every other malware the model has been trained on. Our detection engine performance evaluation shows that we are able to detect highly sophisticated and evasive threat i.e., Zero Click with an accuracy of ≈73%.

C. Defense Engine Performance

As discussed in Section III-C1, we look at the base operation cost for deployment of the Defense Engine and other related services supporting the security mechanisms. For our experimentation of a realistic enterprise IoT-based network,

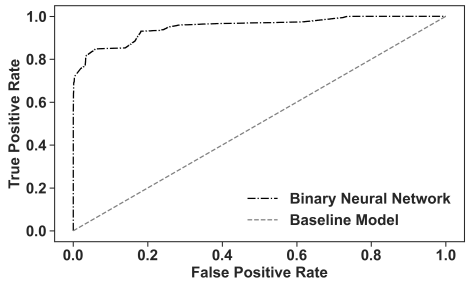

Fig. 2: ROC for Binary NN classifying benign and malicious traffic.

we have shown the costs associated for deployment of our three different deception environments in Table I. We used AWS pricing calculator for cost estimation depicted in the table for numerous associated AWS services. Decoy network simulation costs vary based on the decoy environments to be prescribed for attacks that have variable observability. For our experiment, representation of decoy environment for high observability attack such as DDoS is a port in an EC2 server. Pseudocomb represents a well-equipped server such as a mail server with related services in computation, storage, and memory. Recall, Honeyclone is a collection of Pseudocombs representing various services within a network.

We can observe that the cost required for standing up a fully-cloned decoy environment i.e., Honeyclone is much higher than the other two architectures. However, the Honeyclone choice helps in reducing the risk of sophisticated threats by about 78% when compared to the Honeynet, and 40% when compared to the Pseudocomb.

D. Risk Assessment Results

The threat risk is calculated by determining the likelihood of occurrences of various data-action based threat events and their impacts. The likelihood is a weighted risk factor based on the probability that a threat event takes place. The impact values are attained by assessing the magnitude of harm that can be expected from the consequences of compromising the confidentiality, integrity and availability of an IoT system and its data. Therefore, the overall threat risk score is calculated by factoring the likelihood and impact values scores, which are normalized into a quantitative scale of 1-10. Impacts relating to non-compliance costs, direct business costs and reputation costs are added by assuming that they range from 1 to 30. The threat risk levels are visualized in the heat map shown in Figure 3, where the red grid color represents high risk, green represents low risk and yellow color represents medium risk. The data actions are: data access control (A, B, C),

Authorized licensed use limited to: George Mason University. Downloaded on October 07,2024 at 20:39:28 UTC from IEEE Xplore. Restrictions apply.

data storage (D, E, F), data visualization (G), data transfer (H, I), data collection (J, K), and data fulfillment (L, M, N). Example threat events are: A - Modification of access role (escalation of privilege), B - Update operation on database (data tampering), D - Unauthorized users having access to the relational database to retrieve private data (information disclosure), K - Overwhelming network with requests (denial of service), L - Unlicensed users have access to critical data/system (spoofing identity).

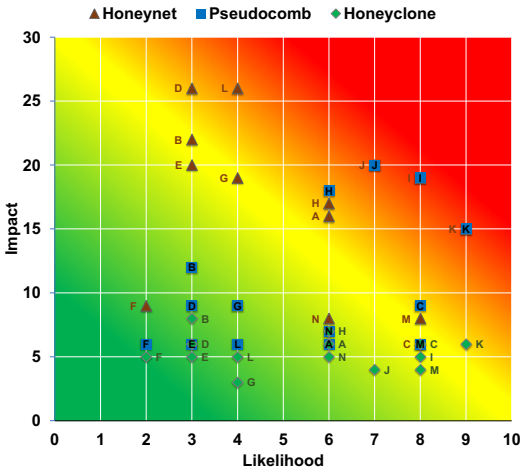

Fig. 3: Risk assessment for various deception environments within CICADA-protected enterprise IoT system.

Figure 3 shows the risk reduction for each of the decoy environment type. Honeyclone is able to significantly reduce the impact of attacks in the enterprise IoT network as opposed to the Honeynet and Pseudocomb. Pseudocomb is also able to perform well for specific service related access controls and other data actions. For readability, the data point labels are organized as left, center and right for Honeynet, Pseudocomb and Honeyclone, respectively.

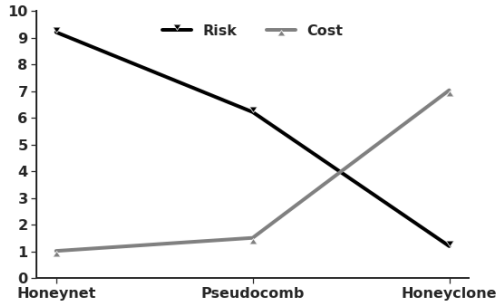

Fig. 4: Risk versus cost of deploying deception environments in CICADA.

We finally showcase the risk versus cost relation through Figure 4. The cost amount and risk values are normalized into scale of $[1, 10]$. We can see that the Honeyclone although incurs high cost, it is able to significantly reduce the risk of threat events for an enterprise IoT network by up to 88%, when compared to a network with no defense mechanisms.

V. Conclusion

In this paper, we presented a centralized intelligent classification and active defense approach viz., "CICADA" for securing smart enterprise IoT-based networks. CICADA features a detection engine with a pipeline of machine/deep learning classifiers to detect malicious network traffic packets from benign, and then classify them into specific sophisticated threats with varying observability. Our evaluation results showed that CICADA is able to detect attacks with different observability levels using ensemble of neural networks and classifiers, with up to 73% accuracy for low observability attack such as Zero Click. Lastly, we showed the cost analysis for deploying various decoy environments, along with the assessment of risks associated with the different decoy environments, with up to 88% risk reduction via Honeyclone when compared to a defenseless enterprise IoT-based network.

References

[1] M. Anagnostopoulos, G. Spathoulas, B. Viaño, and J. Augusto-Gonzalez, "Tracing your smart-home devices conversations: A real world iot traffic data-set," *Sensors*, vol. 20, no. 22, p. 6600, 2020.
[2] J. Lis, "Smart home forecast 2021," Dec 2021. [Online]. Available: https://www.emarketer.com/content/smart-home-forecast-2021
[3] "A look back at the top 12 iot exploits of 2021." [Online]. Available: https://finitestate.io/blog/top-12-iot-exploits-of-2021-p1
[4] K. Ali, S. Askar *et al.*, "Security issues and vulnerability of iot devices," *IJSB*, vol. 5, no. 3, pp. 101–115, 2021.
[5] B. Marczak, J. Scott-Railton, N. Al-Jizawi, S. Anstis, and R. Deibert, "The great ipwn: Journalists hacked with suspected nso group imessage 'zero-click' exploit," Tech. Rep., 2020.
[6] I. Beer, "An iOS zero-click radio proximity exploit odyssey," Dec 2020. [Online]. Available: https://tinyurl.com/mrxjzjcp
[7] V. Akashe, R. L. Neupane, M. L. Alarcon, S. Wang, and P. Calyam, "Network-based active defense for securing cloud-based healthcare data processing pipelines," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–9.
[8] K. Shortridge and R. Petrich, "Lamboozling attackers: A new generation of deception: Software engineering teams can exploit attackers' human nature by building deception environments." *Queue*, vol. 19, no. 5, pp. 26–59, 2021.
[9] S. Garcia, A. Parmisano, and M. J. Erquiaga, "Iot-23: A labeled dataset with malicious and benign iot network traffic," *Stratosphere Lab., Praha, Czech Republic, Tech. Rep*, 2020.
[10] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "Iot network intrusion dataset," 2019. [Online]. Available: https://dx.doi.org/10.21227/q70p-q449
[11] W. H. Hassan *et al.*, "Current research on internet of things (iot) security: A survey," *Computer networks*, vol. 148, pp. 283–294, 2019.
[12] A. Tabassum, A. Erbad, and M. Guizani, "A survey on recent approaches in intrusion detection system in iots," in *2019 IWCMC*.
[13] I. Hafeez, M. Antikainen, A. Y. Ding, and S. Tarkoma, "Iot-keeper: Detecting malicious iot network activity using online traffic analysis at the edge," *IEEE TNSM*, vol. 17, no. 1, pp. 45–59, 2020.
[14] H. Haddad Pajouh, R. Khayami, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "Ai4safe-iot: An ai-powered secure architecture for edge layer of internet of things," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16 119–16 133, 2020.
[15] M. Kuzlu, C. Fair, and O. Guler, "Role of artificial intelligence in the internet of things (iot) cybersecurity," *Discover Internet of things*, vol. 1, no. 1, pp. 1–14, 2021.
[16] F. Pereira, R. Correia, P. Pinho, S. I. Lopes, and N. B. Carvalho, "Challenges in resource-constrained iot devices: Energy and communication as critical success factors for future iot deployment," *Sensors*, vol. 20, no. 22, p. 6420, 2020.
[17] M. M. Yamin, M. Ullah, H. Ullah, and B. Katt, "Weaponized ai for cyber attacks," *Journal of Information Security and Applications*, vol. 57, p. 102722, 2021.
[18] L. Zhang and V. L. Thing, "Three decades of deception techniques in active cyber defense-retrospect and outlook," *Computers & Security*, vol. 106, p. 102288, 2021.
[19] K. Xiao, C. Zhu, J. Xie, Y. Zhou, X. Zhu, and W. Zhang, "Dynamic defense strategy against stealth malware propagation in cyber-physical systems," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1790–1798.
[20] "What is active defense and what does it mean in cybersecurity?" [Online]. Available: https://tinyurl.com/3b3pa62h
[21] T. Nguyen and H. Xu, "When can the defender effectively deceive attackers in security games?" *AAAI*, 2022.
[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[23] NIST, "SP 800-30: Guide for Conducting Risk Assessments," U.S. Department of Commerce, Tech. Rep., 2012.
[24] SpaceSiren Github Repository. [Online]. Available: https://github.com/spacesiren/spacesiren

Authorized licensed use limited to: George Mason University. Downloaded on October 07,2024 at 20:39:28 UTC from IEEE Xplore. Restrictions apply.