Firmware Attack Detection Using Logistic Regression (FAD-LR)

[1]A.Punidha
Dept of Computer Science and Engineering
Coimbatore Institute of Technology
Coimbatore, India
punitulip@gmail.com

[2]Dr. E.Arul
Dept of Information Technology
Coimbatore Institute of Technology
Coimbatore, India
arulcitit@gmail.com

[3]E.Yuvarani
Department Master of Computer Application,
SNS College of Technology,
Tamilnadu, India

Abstract*— The smart devices, commonly referred to as IoT devices, are experiencing a significant surge in demand and are becoming increasingly integrated into our daily lives. Cyber felons perceive monetary potential, thereby intensifying and setting apart their assaults. One of the risks faced by IoT device enthusiasts is that threats can arise unexpectedly, and seemingly harmless methods can turn into powerful tools for illegal activities. Possible paraphrased text: - Crypto currency could be subject to hostile withdrawals, DDoS attacks, or botnet schemes that expose computers to harm. - Perils for crypto currency users may include malicious withdrawals, DDoS assaults, or botnets that exploit vulnerabilities in computer systems. - Risks to cryptocurrencies could entail malevolent withdrawals, DDoS offensives, or botnet activities that expose devices to compromise. - Threats to digital coins might involve harmful withdrawals, DDoS attacks, or botnet campaigns that compromise the security of computers. - Challenges facing virtual currency could involve malicious withdrawals, DDoS strikes, or botnet activities that compromise the confidentiality of computing devices. Once the IoT system belonging to the victim is infiltrated, the malwares seize command of the device and engage in malevolent actions. In this paper, the LR classification technique is suggested to cluster IoT app service calls to kernel API calls that are related to network. By utilizing LR, the pool network connected unidentified executable API calls that executed malicious activities specifically targeting IoT devices. After setting up the IoT kernel's network of API calls, a LR algorithm was utilized to identify the closest association to the harmful behavior. The study involved evaluating 1621 malware specimens, derived from diverse sources and representing all malware groups, yielding an optimistic precision rating of 99.39% and a false positive rate of 0.1%.

Keywords*— Internet - of - Things, Firmware, API calls, Logistic Regressionm, Machine Learning, Backdoors, Malware.

I. INTRODUCTION

Malware pertains to detrimental software intentionally established to infiltrate your computer or any other electronic device for the aim of inducing harm. According to specialists, IoT technology is at risk of being targeted by malware attacks. Their lack of security measures causes them to stay connected to the internet continuously. Enumerated are various possible hazards capable of causing harm to your IoT gadget[1].

Firmware is the term used to refer to the software which manages the functions of an IoT gadget. It functions as a middleman connecting the hardware and external surroundings, typically identified as either embedded or OS-based firmware. Customized firmware, which is the software that runs on IoT devices, is often utilized because of the limited resources available on these devices[12]. In many cases, it may be cost-effective for device manufacturers to hire software developers with extensive familiarity of the hardware so that they can produce embedded software, also known as firmware, that seamlessly integrates with the hardware.

Develop a thorough understanding of all the equipment present in your network. Introduce effective measures to inspect and assess network logs in order to comprehend the communication activities and operations of all devices that are connected to your network. Make changes to all default passwords. Although it may not always be feasible to execute password management measures, a crucial initial step for installing company-owned IoT devices involves enhancing passwords and incorporating two-factor authentication.

Keep up-to-date by regularly revising and rectifying any missing sections or faults. Operations on a daily basis across numerous industries, including manufacturing and healthcare, heavily rely on the utilization of IoT devices. It is important to stay updated on any recently discovered security vulnerabilities and consistently update your device's security with the latest patches. Create independent networks for Internet of Things devices. To hinder any sideways progression, it is recommended that autonomous networks be equipped with IoT devices, isolated and controlled with limitations on ingress and egress network traffic. To reduce external network connectivity, restrict communication solely to essential IPs and ASNs and block unnecessary port entry[10].

It is anticipated that embedded software engineers will fulfill two roles simultaneously. In addition to creating the firmware, they also design the applications necessary for hardware communication, such as drivers, and the software that allows for user interaction, such as the configuration interface[9]. Just as computers advanced from using firmware in ROM to having an operating system like MS-DOS to improve their capabilities, IoT devices are undergoing development and growth[2]. Typically, an IoT device functions through an OS that affords a degree of segregation between the hardware and additional software running on the device.

Introducing the IoT operating system enables a distinct allocation of responsibilities through the implementation of a boundary between the device's application software and the

A. K. Visvam Devadoss et al. (eds.), *Proceedings of the 6th International Conference on Intelligent Computing (ICIC-6 2023)*, Advances in Computer Science Research 107,
https://doi.org/10.2991/978-94-6463-250-7_8

hardware underneath. Highly skilled software engineers who possess deep understanding of hardware can now devote their time to developing device drivers, while software programmers who do not require extensive knowledge of hardware can concentrate on creating smart software for the device [13].

A popular choice for many device manufacturers is Busybox, which is essentially a simplified version of the Unix operating system. It includes commonly used tools, takes up minimal storage, and provides a range of Unix features all within a single program. IoT devices rely on wireless communication as a standard feature, enabling them to operate seamlessly in any area of your living or work space. The communication demands of the device differ depending on its intended purpose.

Prerequisites for protecting IoT devices from malevolent software. While it's impossible to completely avoid malware attacks, effective techniques exist to identify and hinder them, effectively safeguarding your computer systems from harmful programs. Employ secure techniques to authenticate your identity. Deploy multi-factor authentication by introducing additional elements beyond just passwords, like PINs or security questions. Malware typically possesses equivalent access rights as the logged-in user, which is a prevalent occurrence[3]. Typically, accounts that are not designated as administrators have limited access to the computer or network system's most susceptible areas. To guarantee safety measures, it is recommended to use the administrator account only when necessary and consistently follow the principle of minimum privileges[7]. Provide users with only the crucial functionalities, services, and information required for completing their tasks[11].

II. RELATED WORK

Kalutharage et.al., Machine Learning has become increasingly popular in creating intrusion detection systems. These systems have demonstrated remarkable abilities in identifying attacks[4]. The challenge of developing IoT ML-powered IDS (Intrusion Detection Systems) lies in the obscurity of their decision-making approach and the lack of attack data to facilitate their training, posing a significant barrier. Employing methods for detecting anomalies and examining anticipated results in connection with feature contribution or executing analysis on impacts based on features can augment the trustworthiness of those holding a vested interest. In this paper, a novel strategy for monitoring IoT security is presented. This entails combining deep auto encoder models and Explainable Artificial Intelligence (XAI) to assess the precision and reliability of detecting attacks generated by machine learning-based IDSs. We propose a strategy to improve the ML-based decision-making process of IoT security monitoring, with the goal of reducing uncertainty related to predictive results. It reduces the uncertainty in the process by providing intricate information on the factors that influence the forecast and the extent of their influence. Our methodology makes use of SHAP (SHaply Adaptive values exPlanations) to establish a connection between suitable credit distribution and specific explanations at a local level, resulting in measurable data that sheds light on the rationale behind every forecast. The utilization of the USB-IDS benchmark dataset yielded a success rate of 84% in recognizing harmless operation and a perfect rate of identification of likely breaches during the evaluation process. Our experiment has demonstrated that the integration of XAI into the autoencoder model is effective in removing the need for malicious data during training and ensuring accurate identification of anomalies with a high level of certainty in the event of an attack.

S. Gaba et.al., Detecting IoT malware is a major challenge due to the absence of a safety plan and the distinct features of IoT devices, which have different processor designs[5]. Especially when detecting malicious software that has the potential to harm various kinds of Internet of Things devices. The primary focus area of the community security department is presently identifying malicious software in the realm of the Internet of Things. Several tests rely on either dynamic or static analysis for identifying IoT malware, however, static methods prove to be superior in handling the difficulty of various technologies. This piece provides a detailed examination of identifying stationary malicious software in the context of the Internet of Things. At the outset, we provide an explanation, advancement, and probable hazards related to malware in IoT[10]. Afterwards, they condense, analyze, and inspect recent suggestions for detecting malware in IoT. At last, we have effectively identified the exact techniques for forthcoming evaluations.

N. Nimalasingam et.al., Incorporating IoT devices has become a critical necessity in modern times, spanning from handheld devices to automated industrial machinery and even automobiles equipped with IoT technology[6]. The IoT realm observes an average of over 100 device connections in one second, leading to an immense amount of data being produced and incorporated into the environment. The value of information is noteworthy, and occasionally immeasurable, making it a prime target for cyber attackers, leading to an increase in instances of cybercrime. It is critically important to promptly detect malicious software in the Internet of Things (IoT)[9]. By employing Machine Learning algorithms, researchers have achieved a successful result in identifying Internet of Things (IoT) malware through the examination of their network traffic traits in a forensic manner. The strategy consists of carefully selecting the most unique features and combining them with the binary characteristics of the various types of malware. Diverse features of network traffic were utilized to analyze a large dataset containing various network traffic collections. As a result, through the process of feature extraction for each type of malware, the proposed model achieved an almost perfect detection accuracy rate of 100% during the experimental phase of the study. To improve this model, incorporating the level of fog integration in the IoT layer could enhance machine learning's ability to detect harmful packet transmissions to the primary network.

III. THEORETICAL BACKGROUND

DELINEATION OF FIRMWARE ATTACK DETECTION USING LOGISTIC REGRESSION (FAD-LR)

The process of logistic regression is associated with categorizing information. The objective is to forecast a result IoT executable file API that can be classified into two distinct groups by utilizing a variety of uncorrelated elements. There are only two potential outcomes within a binary scenario, with one being the occurrence of an event malicious (1) and the other being its non-occurrence benign (0). Independent variables are those that are capable of

influencing the dependent variable, specifically those that potentially affect the end result[8]. To categorize an executable as potentially dangerous or benign. In order to apply linear regression to tackle this problem, it is essential to establish a threshold that can assist in the process of classification IoT API calls. In situations where the projected numerical result for a malicious category is 0.3 and the threshold is set at 0.6, the data point could be inaccurately categorized as harmless, resulting in serious consequences during real-time IoT devices or application running scenarios. Thus, it can be deduced that linear regression is not an appropriate approach to solve classification issues. The boundless nature of linear regression calls for the utilization of logistic regression. The values assigned to them are restricted to a specific range of 0 to 1 and cannot surpass that boundary.

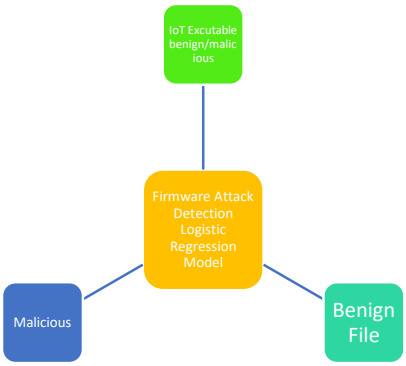

Fig 1 : Delineation of Firmware Attack Detection Logistic Regression Model

The mathematical model known as Logistic Regression (LR) can be formulated as follows,

$$\log\left(\frac{\mathrm{pe(E)}}{1-\mathrm{pe(E)}}\right) = \beta 0 + \beta 1 \mathrm{E}$$

E represents the various scale values of IoT API calls applied in runtime.

The mathematical concept referred to as odds, symbolized by p(e)/(1-p(e)), on the left side of the equation, is equivalent to the logit or log-odds function. Odds, which are commonly referred to as the likelihood of failure because of the missing information of IoT API calls, serve as a measure for determining the probability of success. Therefore, Logistic Regression transforms a mixture of various IoT device API inputs to logarithmic form, resulting in a 1 output. The function that follows is the opposite of the previously mentioned one.

$$\mathrm{p(E)} = \frac{e^{\beta 0+\beta 1 \mathrm{E}}}{1+e^{\beta 0+\beta 1 \mathrm{E}}}$$

Consistently, the result is a likelihood estimation that lies between the values of zero and one. By employing the Sigmoid function, it becomes feasible to convert expected values into likelihood estimates and better analysis of IoT API calls. The sigmoid procedure converts a benign IoT executable to numerical quantity into a number that falls within the range of 0 to 1. In the domain of artificial intelligence, the utilization of sigmoid is necessary to transform forecasts into likelihoods.

The sigmoid function in mathematics may be expressed as follows:

$$\mathrm{f(E)} = \frac{1}{1+e^{-(e)}}$$

The utilization of cost functions in machine learning is intended to estimate the deficiency displayed by models while they are functioning. Simply put, In IoT executable analysis the cost function acts as a measure to assess how accurate a model's estimation is in deciphering the correlation between input factor E and output factor O. Usually, this phenomenon is articulated as a difference or discrepancy between the estimated and actual numerical values. The aim of a LR model based on machine learning is to identify the weights, parameters, or configuration that can result in a significant reduction of the cost function during analysis of IoT API calls [15].

Convexity in a function ensures that there are no points of intersection between any two points on the curve, whereas a non-convex function must have a minimum of one point of intersection. When considering cost functions, a convex form guarantees a minimum value that is applicable globally, whereas a non-convex form only ensures minimum values within specific regions [9].

The estimated probability is the result produced by the hypothesis. This is employed to deduce the degree of certainty that the predicted outcome matches the real outcome when a particular IoT API call input E is provided. Take into account the following instance. The probability estimation is the outcome of the supposition. This is employed to establish the level of trustworthiness in the relationship between the predicted and actual values, given a particular input E. In an analysis the attack involves portable executables, if the e1 value is considered; it appears to result in a calculated probability of 0.7. As per the declaration, there is a 70% chance that IoT executable could result in harm.

$$\mathrm{h_\Theta(x)} = \mathrm{P(Y = 1 \mid X; theta\ )}$$

Probability that $\mathrm{Y = 1}$ given X which is parameterized by 'theta'.

$$\mathrm{E(e = 1 \mid E; theta\ ) + E(O = 0 \mid E; theta\ ) = 1}$$
$$\mathrm{E(e = 0 \mid E; theta\ ) = 1 - P(O = 1 \mid E; theta\ )}$$

The application of data is achieved through the use of the linear regression model, followed by the application of a logistic function that can predict the categorical dependent variable.

$$\mathrm{Cost}(c\Theta(e), O(\mathrm{actual})) = \begin{cases} -\log(c \odot (e)) \text{ if } e = 1 \\ -\log(1 - c \odot (x)) \text{ if } e = 0 \end{cases}$$

$$\mathrm{Cost}(c_\Theta(e), o) = -o\log(c_\Theta(o)) - (1-o)\log(1 - h_\Theta(e))$$

If $y = 1, (1-y)$ term will become zero, therefore $-\log\left(h_\ominus(x)\right)$ alone will be present

If $y = 0, (y)$ term will become zero, therefore $-\log(1 - h_\Theta(x))$ alone will be present

The root cause of this unfavorable conduct can be attributed to our focus on minimizing the loss function while concurrently augmenting the likelihood during the training process[14]. If it is assumed that the samples are from an identical and independent distribution, reducing the cost would lead to an increase in the maximum probability.

IV. EXPERIMENTAL RESULTS AND COMPARISON

IoT malevolent program collection [18,19,20] is utilized to dissect and is fundamentally designed to execute web connections. All malware tests are assessed in IoT [21], with an accentuation on the obscure executable that will attempt evil activity to reach the web. The backdoor virus's IoT firmware [16] made utilize of a number of API calls. Too, they found that firmware has less of an accentuation on commerce organize endpoints and more of a center on shopper products such savvy domestic contraptions, lighting installations, indoor regulators, domestic security frameworks, and cameras. Utilizing Windows API calls, the LR algorithm[17] was utilized to pre-process each test. With a essential center on common IoT framework API calls, LRLearning has boosted inner keenness to related unsafe organize association API demands that are related to malware test API calls. The model's method, which uses LR and groups similar information together, helps make new discoveries. Advance particular net-work association API calls have been found interior an inner malware API call connected to the Internet of Things. Fig 2 and 3 demonstrate the feature selection of the different API calls in the IoT executable dataset and comparisons with several machine learning models that used the suggested FAD-LR-IoT. Fig 4 presents the confusion matrices of the proposed model.

```
In [35]: features = []
         index = numpy.argsort(extratrees.feature_importances_)[::-1][:nbfeatures]

In [36]: for f in range(nbfeatures):
             print("%d. feature %s (%f)" % (f + 1, dataset.columns[2+index[f]], extratrees.feature_importances_[index[f]]))
             features.append(dataset.columns[2+f])

1. feature DllCharacteristics (0.139943)
2. feature Machine (0.105156)
3. feature Characteristics (0.091041)
4. feature VersionInformationSize (0.068964)
5. feature SectionsMaxEntropy (0.066953)
6. feature MajorSubsystemVersion (0.059901)
7. feature Subsystem (0.059210)
8. feature ImageBase (0.049987)
9. feature ResourcesMaxEntropy (0.048410)
10. feature ResourcesMinEntropy (0.043082)
11. feature SizeOfOptionalHeader (0.041048)
12. feature MajorOperatingSystemVersion (0.031233)
13. feature SectionsMinEntropy (0.024692)
```

Fig 2. Internal FAD-LR Internet Network Call Groups are created.

```
In [52]: results = {}
         for algo in model:
             clf = model[algo]
             clf.fit(X_train,y_train)
             score = clf.score(X_test,y_test)
             print ("%s : %s " %(algo, score))
             results[algo] = score

DecisionTree : 0.9901847156827257
LogisticRegression : 0.9939152480985151
Adaboost : 0.9848967765360427
GradientBoosting : 0.9881564659488953
GNB : 0.7007605939876856
LinearRegression : 0.5269415368861684
RandomForest : 0.7007243752263672
```

Fig 3. FAR-LR-IoT recommended malware approaches are predominant to existing malware strategies.

```
In [69]: clf = model[winner]
         res = clf.predict(X_new)
         mt = confusion_matrix(y, res)
         print("True positive rate : %f %%" % ((mt[0][0] / float(sum(mt[0])))*100))
         print('True negative rate : %f %%' % ( (mt[1][1] / float(sum(mt[1])))*100))
         print("False positive rate : %f %%" % ((mt[0][1] / float(sum(mt[0])))*100))
         print('False negative rate : %f %%' % ( (mt[1][0] / float(sum(mt[1])))*100))
         print(mt)

True positive rate : 99.894545 %
True negative rate : 99.796723 %
False positive rate : 0.105455 %
False negative rate : 0.203277 %
[[96622   102]
 [   84 41239]]
```

Fig 4. FAR-LR-IoT model Confusion matrics.

V. CONCLUSION AND FUTURE WORK

A significant number of malicious software utilizes APIs of IoT systems to carry out partially assessed harmful actions on the intended device. By making use of the entire bandwidth available on the IoT mobile network, malicious software spreads unsolicited messages and simultaneously gathers personal information from users, sending it to a server controlled by a hacker. The proposed study employs the LR Learning method for clustering malicious executables and firmware API calls that execute network activities in the realm of the Internet of Things. By utilizing an extensive Logistic Regression Learning method, additional resemblances to the hazardous actions of any executable were sought out. The legitimate success rates false positive for the firmware attacks on various IoT devices are 99.39% and 0.1%, in that order. There will be an increase in the utilization of additional IoT APIs in the near future for conducting harmful network actions.

REFERENCES

[1] J Steven Perry," Anatomy of an IoT malware attack - How to prevent your IoT devices from joining the zombie bot horde", https://developer.ibm.com/articles/iot-anatomy-iot-malware-attack/ (accessed August 7 , 2019)

[2] Aneri Barevadia," Malware is a growing threat to IoT devices- find out how to protect your device!", https://www.einfochips.com/blog/malware-is-a-growing-threat-to-iot-devices-find-out-how-to-protect-your-device/ (accessed September 9 , 2022)

[3] Help Net Security," IoT malware attacks rose 700% during the pandemic", https://www.helpnetsecurity.com/2021/07/20/iot-malware-attacks-rose/ (accessed July 20, 2021)

[4] Kalutharage, C.S., Liu, X., Chrysoulas, C. (2022). Explainable AI and Deep Autoencoders Based Security Framework for IoT Network Attack Certainty (Extended Abstract). In: Li, W., Furnell, S., Meng, W. (eds) Attacks and Defenses for the Internet-of-Things. ADIoT

2022. Lecture Notes in Computer Science, vol 13745. Springer, Cham. https://doi.org/10.1007/978-3-031-21311-3_8

[5] S. Gaba, S. Nagpal, A. Aggarwal, R. Kumar and S. Kumar, "An Analysis of Internet of Things (IoT) Malwares and detection based on Static and Dynamic Techniques," 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC), Solan, Himachal Pradesh, India, 2022, pp. 24-29, doi: 10.1109/PDGC56933.2022.10053115.

[6] N. Nimalasingam, J. Senanayake and C. Rajapakse, "Detection of IoT Malware Based on Forensic Analysis of Network Traffic Features," 2022 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka, 2022, pp. 122-130, doi: 10.1109/SCSE56529.2022.9905212.

[7] Hamdija Sinanovi´c, Sasa Mrdovic," Analysis of Mirai Malicious Software", 2017,25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM),DOI: 10.23919/SOFTCOM.2017.8115504.

[8] Saishruthi Swaminathan , https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc [Accessed Mar 15, 2018]

[9] Premanand S, "Building an End-to-End Logistic Regression Model",https://www.analyticsvidhya.com/blog/2021/10/building-an-end-to-end-logistic-regression-model/ [Accessed August 26th, 2022]

[10] Ben Dickson,"The IoT ransomware threat is more serious than you think", https://www.iotsecurityfoundation.org/the-iot-ransomware-threat-is-more-serious-than-you-think/

[11] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, "Iot security: Ongoing challenges and research opportunities," in 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, pp. 230–234, Nov 2014.

[12] Anna-senpai, "Mirai-source-code/mirai/bot/scanner.c:// set up passwords." https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/bot/scanner.c#L124, 2016. [Accessed 20.5.2017.].

[13] Robert Moskovitch, Nir Nissim, Yuval Elovici," Malicious Code Detection and Acquisition Using Active Learning",2007 IEEE Intelligence and Security Informatics, DOI: 10.1109/ISI.2007.379505.

[14] Anamika Thanda, "What is Logistic Regression? A Beginner's Guide",https://careerfoundry.com/en/blog/data-analytics/what-is-logistic-regression/ [Accessed December 19, 2022]

[15] Aruna Jain, Akash Kumar Singh," Integrated Malware analysis using machine learning", 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), DOI: 10.1109/TEL-NET.2017.8343554.

[16] Tian, R., Batten, L. and Versteeg, S. (2008) "Function Length as a Tool for Malware Classification". Proceedings of the 3rd International Conference on Malicious and Unwanted Software, Fairfax, pp.57-64, 7-8 October 2008,

[17] Available:http://blog.echen.me/2011/04/27/choosing-a-machine-learningclassifier/

[18] Malware data set. [Online] [Accessed: Nov 29,2019].Available: www.kernelmode.info

[19] Malware data set. [Online]Accessed: October, 2016.Available: http://dasmalwerk.eu/

[20] Malware data set. [Online]Accessed: October, 2016.Available: http://virusshare.com/

[21] Joanna rutwoska, Redpill technique for Vm detection [online].Accessed: October 2016.Available:https://blog.invisiblethings.org/