A Project Report

On

Detecting IoT Threats using Machine Learning

submitted for partial fulfillment of the requirements

for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

Tarun Kumar (1900290100171)
Shraddha Singh (1900290100147)
ShubhamBhaskar(1900290100156)
Piyush Mishra (1900290400081)

Under supervision of

Dr. Himanshi Chaudhary

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

May, 2023

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledgeand belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where dueacknowledgment has been made in the text.

Signature:

Name: Tarun Kumar, Shraddha Singh, Shubham Bhaskar, Piyush Mishra

Roll No: 1900290100171,1900290100147,1900290100156,1900290400081

Date:

CERTIFICATE

This is to certify that Project Report entitled “Project Title” which is submitted by Student name in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Date: **Supervisor Name**

(Designation)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to superisor name, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full supportand assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:

Signature:

Name: Tarun Kumar, Shraddha Singh, Shubham Bhaskar, Piyush Mishra

Roll no: 1900290100171,1900290100147,1900290100156,1900290400081

Abstract

Assessing security of application deployments in the Fog is a non-trivial task, having to deal with highly heterogeneous infrastructures containing many resource-constrained devices. In this paper, we introduce:

(i) a declarative way of specifying security capabilities of Fog infrastructures and security requirements of Fog applications, and

(ii) a (probabilistic) reasoning strategy to determine application deployments and to quantitatively assess their security level, considering the trust degree of application operators in different Cloud/Fog providers.

Understanding and dealing with the security levels of IOT applications which are to be deployed is a tedious task.

In this article we try to establish relations between security of an IOT application. We mainly deal with Cloud Edge infrastructures. We have used a probabilistic approach to establish trust relations between stakeholders and manage applications.

Keywords:
SecFog; fog computing; edge computing; testbed; mobile crowdsourcing; Internet of things; Machine Learning; ProbLog2; Trust Network.

CHAPTER 1

1.1 INTRODUCTION

Over the past few years, the Internet of Things has become one of the most important technologies of the 21st century. Low-cost computing, cloud computing, big data, analytics, and mobile technologies allow physical objects to share and collect data with minimal human intervention. In a hyper-connected world, digital systems can record, track and rectify every interaction between connected things.

Machine learning for the *Internet of Things* can be used to predict future trends, detect anomalies, and improve intelligence through image, video, and audio processing. Why use machine learning for the Internet of Things? Machine learning can help you understand hidden patterns in IoT data by analyzing vast amounts of data using complex algorithms.

SecFog is a simple declarative prototype that can be used to find multi-service application deployments to Cloud-Edge infrastructures and to assess their security level based on specific application security requirements, available infrastructure security capabilities, and considering trust degrees in different Edge and Cloud providers. SecFog constitutes a first, well-founded and explainable effort towards this direction.

Fog computing is a distributed computing infrastructure where data, processing power, storage, and applications reside somewhere between data sources and the cloud. Like edge computing, fog computing brings the benefits and power of the cloud closer to where data is created and processed. People use the terms *fog computing* and *edge computing* interchangeably. Both are about bringing intelligence and processing closer to where the data is created. This is often done for efficiency, but can also be done for security and compliance reasons.

Fog computing is a new research field, and it is very important to test the functions and performances of various applications and services before they are deployed to the production environment. However, current evaluations are more based on various simulation tools— FogNetSim , iFogSim , EdgeCloudSim , etc.—which usually leads to a large difference between the

experimental results and the actual situation. A more realistic testing environment can help users find the bottle neck, defect or limitation of their applications. We have designed piFogBed based on raspberry pies, which is a real fog computing testbed that can provide a real fog computing architecture, simulate various network scenarios for users and support users' testing of real applications.

1.2 PROJECT DESCRIPTION

We do not want our data about how we use our devices and habits related to them to be stolen and exploited by the attacker. On the other hand, we want Quality of Service from the IoT software. It is very clear that such systems need complex infrastructure. This is a change to cloud IoT. We are focusing on Cloud Edge computing. Cloud Edge computing is considered more efficient and has better support for IoT applications.

Our approach deals with the security requirements of application. These security requirements are matched with the security capabilities of different level of IoT application. Security of a system requires proper analysis of its security properties. It also works for prevention and recovery from potential attacker.

CHAPTER 2

LITERATURE REVIEW

Research on fog computing is ongoing and various tutorials and review articles have been published over the past few years. Various independent platforms and architectures for fog computing have been reported in the literature. These platforms and architectures serve a variety of IoT applications. However, the classification of fog computing frameworks is not specifically presented in existing review and overview documents. Therefore, there is a need to develop a taxonomy of fog computing environments to understand the current state of research and to identify the various research gaps that exist in the literature. This article provides a systematic literature review of existing work, considering frameworks and architectures related to fog computing. This article will help researchers continue their research on fog computing after the analysis in this review.

We have proposed a Systematic Literature Review on fog computing following guidelines proposed by Kitchenham. The main purpose of the review is to discuss important characteristics of fog computing environments and identify various issues related to architecture design, QoS metrics, implementation details, applications and communication modes.

With the rapid development of mobile internet and Internet of Things applications, the conventional centralized cloud computing is encountering severe challenges, such as high latency, low Spectral Efficiency (SE), and non-adaptive machine type of communication. Motivated to solve these challenges, a new technology is driving a trend that shifts the function of centralized cloud computing to edge devices of networks.

EdgeCloudSim is another fog simulation tool based on CloudSim. It is designed to assess the computing and network requirements of edge computing. EdgeCloudSim supports mobility by providing mobile models, network link models and edge server models. pFogSim extends EdgeCloudSim to include different networks, applications and business process models. IoTSim is also designed to simulate an edge computing environment in which IoT applications send large amounts of data to big data processing systems. As a result, it adds storage and big data processing layers

to CloudSim. EdgeCloudSim and IoTSim both inherit the same scalability and DES limitations as iFogSim.

Simulation tools simplify the evaluation process, but the differences between the simplified scene and the production environment are very large, especially for the dynamic fog computing scene; the experimental results may be infidelity. Network emulation can solve some problems of simulation to a certain extent and improve the fidelity of experimental results to a certain extent.

Compared with simulation and emulation, the fidelity of an overlay network built with real equipment in a real network is the best. However, it is high cost and difficult to build a real fog computing testbed because the fog equipment produced by different manufacturers is not compatible, and the price is high. We have built the piFogBed using raspberry pies, but it does not support the mobility of end devices.

ProbLog is a recent probabilistic extension of Prolog motivated by the mining of large biological networks. In ProbLog, facts can be labeled with probabilities. These facts are treated as mutually independent random variables that indicate whether these facts belong to a randomly sampled program.

This model places ever increasing demands on communication and computational infrastructure with inevitable adverse effect on Quality-of-Service and Experience. The concept of Edge Computing is predicated on moving some of this computational load towards the edge of the network to harness computational capabilities that are currently untapped in edge nodes, such as base stations, routers and switches. This position paper considers the challenges and opportunities that arise out of this new direction in the computing landscape.

Building secure systems is difficult for many reasons. This paper deals with two of the main challenges: (i) the lack of security expertise in development teams and (ii) the inadequacy of existing methodologies to support developers who are not security experts. The security standard ISO 14508 Common Criteria (CC) together with secure design techniques such as UMLsec can provide the security expertise, knowledge, and guidelines that are needed.

This paper describes a security requirements engineering methodology called SecReq. SecReq combines three techniques: the CC, the heuristic

requirements editor HeRA, and UMLsec. SecReq makes systematic use of the security engineering knowledge contained in the CC and UMLsec, as well as security-related heuristics in the HeRA tool. The integrated SecReq method supports early detection of security-related issues (HeRA), their systematic refinement guided by the CC, and the ability to trace security requirements into UML design models.

Trust Network provides a simple notation for expressing transitive trust relationships, and defines a method for simplifying complex trust networks so that they can be expressed in a concise form and be computationally analyzed. Trust measures are expressed as beliefs, and subjective logic is used to compute trust between arbitrary parties in the network.

We also present motivation for using probabilistic risk and provide fundamental relations for making the associated calculations. The benefits and applications of using a risk index for security assessment are discussed, and an illustration is provided for line overload security assessment in the operational context.

Edge computing is an appealing technology to compensate for stringent latency-related issues, its deployment engenders new challenges. In this article, we highlight the role of edge computing in realizing the vision of smart cities. First, we analyze the evolution of edge computing paradigms. Later, we categorize and classify the literature by devising a comprehensive and meticulous taxonomy. Finally, several indispensable open challenges along with their causes and guidelines are discussed, serving as future research directions.

Those data after analytics provide significant information that could greatly benefit IoT applications. Different from traditional applications, IoT applications, such as environmental monitoring, smart navigation, and smart healthcare come with new requirements, such as mobility, real-time response, and location awareness. However, traditional cloud computing paradigm cannot satisfy these demands due to centralized processing and being far away from local devices.

Hence, edge computing was introduced to perform data processing and storage in the edge of networks, which is closer to data sources than cloud computing, thus efficient and location-aware. Unfortunately, edge computing brings new security and privacy challenges when applied to data analytics. The literature still lacks a thorough review on the recent advances in secure

data analytics in edge computing. In this paper, we first introduce the concept and features of edge computing, and then propose a number of requirements for its secure data analytics by analyzing potential security threats in edge computing.

Despite the increasing usage of cloud computing, there are still issues unsolved due to inherent problems of cloud computing such as unreliable latency, lack of mobility support and location-awareness. Fog computing can address those problems by providing elastic resources and services to end users at the edge of network, while cloud computing is more about providing resources distributed in the core network.

This survey discusses the definition of fog computing and similar concepts, introduces representative application scenarios, and identifies various aspects of issues we may encounter when designing and implementing fog computing systems. It also highlights some opportunities and challenges, as direction of potential future work, in related techniques that need to be considered in the context of fog computing.

Chapter 3

PROPOSED METHODOLOGY

SecFog is written in the ProbLog2 language and it can be used together with existing approaches (e.g., FogTorchPi) that solve the problem of mapping IoT application services to Cloud-Edge infrastructures according to requirements other than security and trust.

Diving more inside the SecFog Ingredients and understanding it better. In Sec Fog we consider two roles:

Application Operator- It design and manage application deployments like (hardware, QoS). It also connect different s1,s2 values to calculate them.

Infrastructure operator- It manages targeted cloud-edge nodes and provisioned infrastructure capabilities to the user.

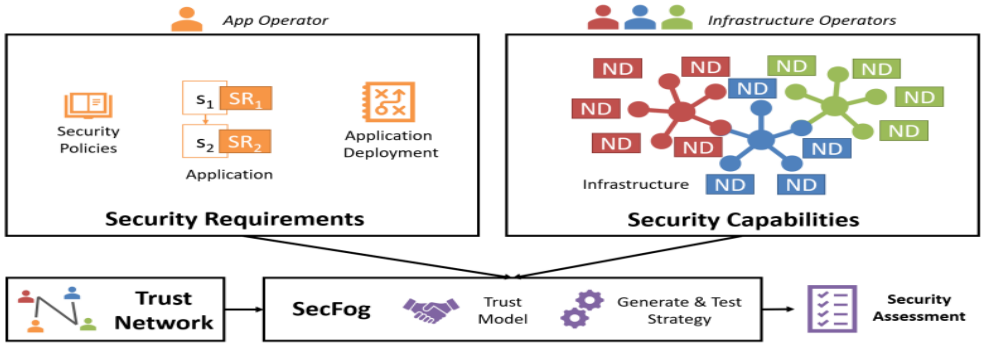


Figure 1: Bird's-eye view of SecFog.

Now by the given figure 1 we can see that there is a connection-based structure, in which Trust networks consist of transitive trust relationships between people, organisations and software agents connected through a medium for communication and interaction. These all are used by secfog and generate and test strategy for the security assessment which assess, operational and implements key securities .

The infrastructure operators have to provide a Node Descriptor for the nodes managed by it. This gives its security capabilities and estimated effectiveness

gainst attack.

The application operator deals with the application services. It specifies the security requirements for application services.

These two operators (Application and Infrastructure) are referred to as stakeholders. Our model is going to work on the trust degree between these stakeholders. By these trust degrees between different stakeholders, we make our Trust Network.

When all the security requirements, security capabilities and trust network is given to SecFog, it features:

-Trust Model

-Generate and Test Strategy

Trust Model:

Cloud edge deployments deal with trust degrades among various operators along with the adopted security capabilities.
The trust relations can be direct as well as indirect. The trust relation between two models directly depends upon the previous interaction.
When making the trust networks some of the trust failures can be missing, these values are calculated on the basis of other available values by appropriate multiplication, addition.
The truest opinion on same path multiplies and on parallel path are added.

Model Works:

Our model works on two major things-

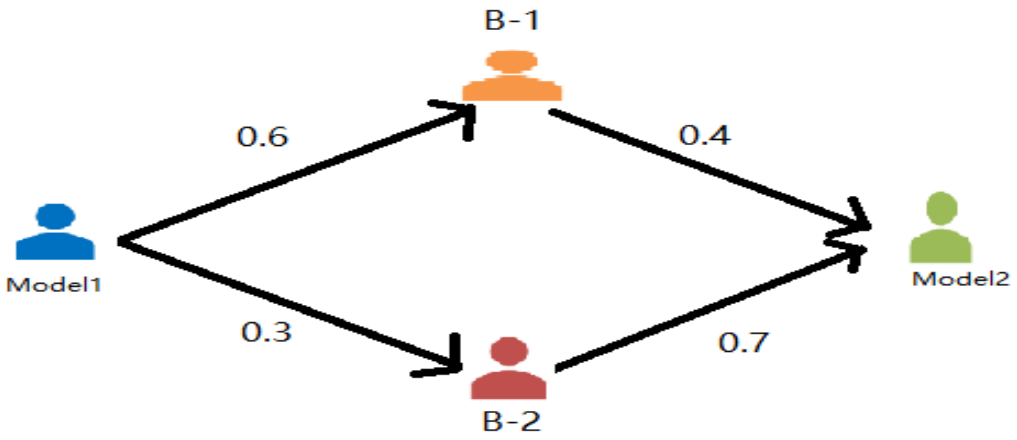
1. Security Capability
2. Security Requirements

1. The **Security Capability** have referred to the security capability of mod for each node. The node description contains all the information of security capability and also effective against attackers.
2. The **Security Requirements** referred to the security requirement of application that is specify all their security requirements.

In this model we make a trust network. This trust network is the trust relation between nodes (security capability and security requirements).

EXAMPLE:

Consider the trust network of Figure-2 and suppose to be interested in the (indirect) trust relation between model1 and model2?



A Trust Network

→ It can be simply computed with the SecFog trust model in ProbLog2 as:

```
%%% trust relations declared by model1
0.6::trusts(model1, a1).
0.3::trusts(model1, b1).
 %%% trust relations declared by a1
 0.4::trusts(a1, model2).
%%% trust relations declared by b2
0.7::trusts(b1, model2).
query (trusts2(model1, model2)).
 which returns
trusts2(model1, model2): 0.45
```

by this above it is clear that, in the proposed trust model, the contribution of trust relations deteriorates along paths and that all possible paths give their contribution to the output result. Indeed, the final result corresponds to the likelihood that it is possible to establish a trust path from model1 to model2 over the considered trust network.

Generate and Test Strategy-

Fog computing (edge computing) is a new research field and can accelerate the analysis speed and decision-making for these delay-sensitive applications [11]. It is very important to test functions and performances of various applications and services before they are deployed to the production environment, and current evaluations are more based on various simulation tools; however, the fidelity of the experimental results is a problem for most of network simulation tools.

piFogBed is a fog computing testbed built with real devices, but it does not support the testing of mobile end devices and mobile fog applications [12]. The paper proposes the piFogBedII to support the testing of mobile fog applications by modifying some components in the piFogBed, such as extending the range of end devices, adding the mobile and migration management strategy and inserting a container agent to implement the transparent transmission between end devices and containers. The evaluation results show that it is effective and the delay resulting from the migration strategy and container agent is acceptable.

Chapter 4

Results and Discussion

This article identifies IoT threat models and learning-based IoT security solutions, such as IoT authentication, access control, malware detection, and secure offloading, that are demonstrated to be effective ways to safeguard the IoT.

In this paper, we proposed a declarative methodology, SecFog, which can be used to quantitatively assess the security level of multi-service application 24 deployments to Cloud-Edge infrastructures. With a prototype implementation in ProbLog2, we have shown how SecFog helps application operators in determining secure deployments based on specific application requirements, available infrastructure capabilities, and considering trust degrees in different Edge and Cloud providers.

To the best of our knowledge, SecFog constitutes a first well-founded, efficient and explainable effort towards such direction. The well-foundedness and efficiency of SecFog are guaranteed by the state-of-the-art resolution algorithms is implemented within the ProbLog2 engine. The possibility of explaining the obtained security assessment also derives from ProbLog2 functionalities that allow the users to obtain graphical ground programs and proofs for the results of their queries. The SecFog prototype can be fruitfully used with other tools for application deployment so to identify suitable trade-offs among the estimated security level and other deployment performance indicators (e.g., QoS-assurance, resource usage, monthly cost, energy consumption), as we have shown with our prototype FogTorchΠ.

As our immediate future work, we plan to:

- prototype a GUI to provide a user-friendly view of the recommended deployment(s), by suitably highlighting how the application security requirements are satisfied,

- extend such a GUI with a visual explanation of the reasons why a give deployment is not recommended by SecFog, and

- engineer and integrate SecFog with FogTorchΠ and show their applicability to actual use cases.

We also intend to:

- enhance SecFog by embedding a more expressive and customisable trust model that can take into account confidence level towards the declared opinions, by combining it with existing strategies that have been used to quantify trust degrees (e.g., Bayesian or Dempster–Shafer theories) based on direct experience, possibly considering also the mobility of Edge nodes and IoT devices, and

- enrich the current application model of SecFog so to be able to analyse the security of (probabilistic) information flows among the constituent services, by also defining pre-defined patterns.

Acknowledgment

This work has been partly supported by the project “DECLWARE: Declarative methodologies of application design and deployment” (PRA 2018 66) funded by the University of Pisa, Italy.

Chapter 5

References

Ai, Y., Peng, M., & Zhang, K. (2018). Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks*, *4*(2), 77-86.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16).

Brogi, A., Ferrari, G. L., & Forti, S. (2018, September). Secure apps in the fog: Anything to declare?. In *European Conference on Service-Oriented and Cloud Computing* (pp. 46-61). Springer, Cham.
Forti, S., Ferrari, G. L., & Brogi, A. (2020). Secure cloud-edge deployments, with trust. *Future Generation Computer Systems*, *102*, 775-788.

Houmb, S. H., Islam, S., Knauss, E., Jürjens, J., & Schneider, K. (2010). Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. *Requirements Engineering*, *15*(1), 63-93.

Xu, Q., Zhang, J., & Togookhuu, B. (2020). Support mobile fog computing test in piFogBedII. *Sensors*, *20*(7), 1900.

Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A., & Kamal, T. (2015). A review on internet of things (IoT). *International journal of computer applications*, *113*(1), 1-7.

Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016, November). Challenges and opportunities in edge computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 20-26). IEEE.

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE communications surveys & tutorials*, *19*(4), 2322-2358.
Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Fog computing and the internet of things: A review. *big data and cognitive computing*, *2*(2), 10.

Kimmig, A., Demoen, B., De Raedt, L., Costa, V. S., & Rocha, R. (2011). On the implementation of the probabilistic logic programming language ProbLog. *Theory and Practice of Logic Programming*, *11*(2-3), 235-262.

Pan, J., & McElhannon, J. (2017). Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, *5*(1), 439-449.

Josang, A., Hayward, R., & Pope, S. (2006). Trust network analysis with subjective logic. In *Conference Proceedings of the Twenty-Ninth Australasian Computer Science Conference (ACSW 2006)* (pp. 85-94). Australian Computer Society.

McCalley, J. D., Vittal, V., & Abi-Samra, N. (1999, July). An overview of risk based security assessment. In *1999 IEEE Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No. 99CH36364)* (Vol. 1, pp. 173-178). IEEE.

Shi, W., & Dustdar, S. (2016). The promise of edge computing. *Computer*, *49*(5), 78-81.

Wang, X., Yang, L. T., Xie, X., Jin, J., & Deen, M. J. (2017). A cloud-edge computing framework for cyber-physical-social services. *IEEE Communications Magazine*, *55*(11), 80-85.

Duan, Q., Wang, S., & Ansari, N. (2020). Convergence of networking and cloud/edge computing:

Status, challenges, and opportunities. *IEEE Network*, *34*(6), 148-155.

Pan, J., & McElhannon, J. (2017). Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, *5*(1), 439-449.

Van den Broeck, G., Thon, I., Van Otterlo, M., & De Raedt, L. (2010, July). DTProbLog: A decision-theoretic probabilistic Prolog. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 24, No. 1, pp. 1217-1222).