About Security Solutions in Fog Computing

Eugen Petac
Faculty of Mathematics and Computer Science
"Ovidius" University of Constanța, Romania
epetac@univ-ovidius.ro
Andreea-Oana Petac
Faculty of Mathematics and Computer Science
"Ovidius" University of Constanța, Romania
andreea.petac@gmail.com

Abstract

The key for improving a system's performance, its security and reliability is to have the data processed locally in remote data centers. Fog computing extends cloud computing through its services to devices and users at the edge of the network. Through this paper it is explored the fog computing environment. Security issues in this area are also described. Fog computing provides the improved quality of services to the user by complementing shortages of cloud in IoT (Internet of Things) environment. Our proposal, named Adaptive Fog Computing Node Security Profile (AFCNSP), which is based security Linux solutions, will get an improved security of fog node with rich feature sets.

Key words: Fog Computing, IoT, Fog Computing Security
J.E.L. classification: L8, M1, M3

1. Introduction

Fog computing is a modern computing paradigm, representing distributed computing services, applications, access to pieces of information and various storage data, the user not needing to know the physical configurations for the systems that provide these services. This new technology is based on the tendency of cutting out the costs of the delivery services and increasing the dexterity of the deployment of the services. Utilizing this distributed computing concept, the services can be hosted at end devices (e.g. access points), creating an automated response that drives the value.

The term fog computing was first introduced by Cisco Systems and it refers to a model aimed at broadening cloud computing to the edge of an enterprise's network. Fog computing emerged as a response to the development of new technologies based on IoT (Internet of Things). A fog node is a structure of physical objects or "things" that are equipped with electronics, software, sensors and Internet connections, through which data is collected and distributed. The main requirements highlighted by Cisco White Paper (Cisco, 2015) for fog computing are: latency minimization, network bandwidth conservation, reliable operation, addressing of security concerns, data collection and securing across various wide geographic areas.

The routers, switches and IP based video cameras are some of the devices that are close to the client, having the intention to provide data services, processing, storage and applications within fog computing. A real challenge for fog computing is also the extraordinary development of IoT. In the IDC study, Worldwide Internet of Things Forecast, 2015-2020 (IDC, 2015) the following are expected by 2020: the global Internet of Things market will grow to $1.7 trillion in 2020 from $655.8 billion in 2014; over 29.5 billion of embedded and intelligent systems will be functional; over 25 million of IoT applications will be available for those that are interested in; the amount of data that is handled by IoT will reach 50 trillion GB.

There are many advantages of using fog computing. As the encrypted data moves toward the network core, security is greatly improved. Then, as it approaches the enterprise, the data is checked, passing through firewalls and other security points. Another benefit it represents the fact that using edge computing consumes less amount of bandwidth. It also provides high levels of scalability, reliability and fault tolerance.

Along the all benefits that fog computing has to offer, there are some security issues which can especially affect the reliability, data safety, data theft and the efficiency of this modern technology. One of the considerations taken when talking about the fog are the security aspects, for example the data security that can present different problems.

In this paper we first present a state of the art concerning the fog computing technology in section 2. We then explore its security issues faced by consumers. Section 4 encompasses some conclusion regarding the benefits and downsides of this paradigm, along with some tips for the issues presented to be tackled.

2. Related Work

BETaaS (2012) suggested the term of "local Cloud" instead of using the original Cloud. This "local Cloud" refers to the devices that provide connectivity of smart things to the Internet, for example phones, routers etc. Using this method, applications are able to respond in a systematic way and also to require simple and repetitive interactions.

Maharjan, et al. (2013) introduced the Demand Response Management problem in a network of multiple utility companies and consumers in which every unity's objective is to maximize its own advantage. Korzhyk, Conitzer and Parr (2011) proposed a Stackelberg game (Mazalov, 2014) between utility companies and end users to maximize the revenue of each utility company and the payoff of each user. A distributed algorithm was developed that merges to an equilibrium having had only local information available for both the utility companies and the end users. However, the most important downside is that it exists a significant communication overhead between users and utility companies.

Fadlullah, et al. (2014) studied the way energy consumption can be optimized by considering the interaction between both parties. The energy price is represented through a function of total energy consumption. The target function optimizes the difference between the value and the cost of energy. The interaction between the power company and its consumers is patterned by using a two-step centralized game named the Game-Theoretic Energy Schedule (GTES) method.

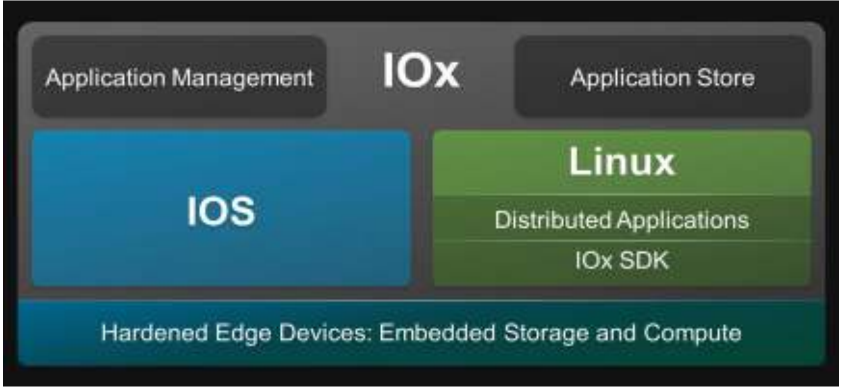
Zhou, et al. (2010) proposed an adaptive traffic light control for maximization of the traffic for both single and multiple lines. Moreover, Li and Shimamoto (2012) suggested a three-tier structure for traffic light control. Firstly, an electronic toll collection system (ETC system) collects the road traffic flow data and calculates the advised speed. Secondly, the radio antennas are installed near the traffic lights. The last step consists in obtaining the road traffic flow information by having a wireless communication between the antennas and the electronic toll collection devices.

Hong, et al. (2013) presented applications of fog computing in the area of mobile devices. He suggested a spatio-temporal event processing system that utilizes a prediction-based continuous query handling. The created system forecasts future query regions for moving consumers and the processing of the events is done early in order for the consumer to have the live situational information when he will reach the future location.

Nishio, et al. (2013) proposed a mathematical framework for diverse resource sharing based on the concept of service-oriented utility functions. In Mobile Cloud Computing, the mobile devices share their heterogeneous resources and support services. Madsen, et al. (2013) combines Smart Grid, Cloud computing and sensors towards Fog computing.

The Cisco IOx (Figure no. 1.) allows devices (routers, switches, wireless access points, video surveillance cameras, and Cisco Unified Computing System (UCS) servers (Gai, Salli and Andersson, 2011)) from a Cisco fog node to be programmable. The devices do not come from the factory with a standard behavior, the user having the possibility to construct it later. The purpose of IOx works is to host applications in a Guest Operating System (GOS) running in a hypervisor directly on the Connected Grid Router (CGR). Python (PSF, 2016) or compiled code at the network edge can be run. By default, the CGR comes packaged with Yocto Linux (Linux Foundation, 2016), but it's possible to replace that image with another operating system. All Cisco fog nodes have converged compute, networking, and storage, which simplifies management and reduces power and space requirements. The user can develop and enhance IoT applications in the cloud, and then deploy them to run in the cloud and in the fog, the same application being able to run on different kinds of fog nodes without doing any modifications.

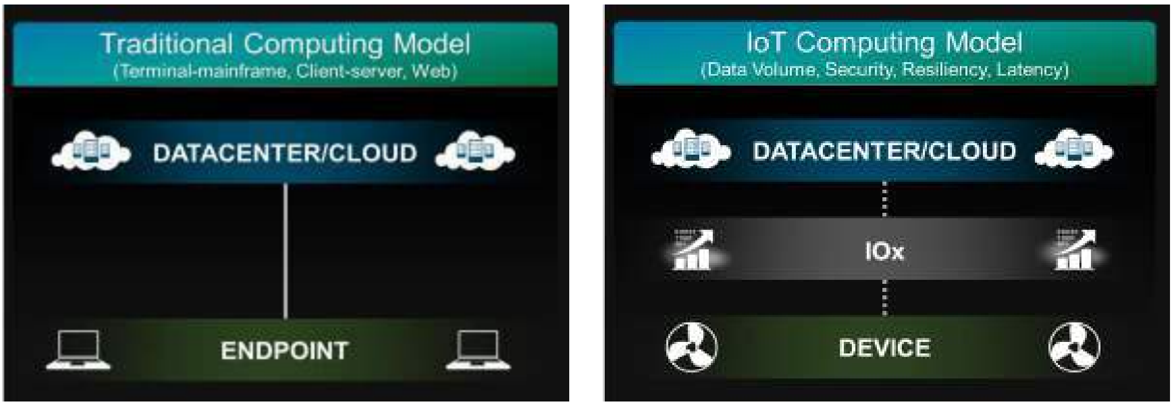
Figure no. 1. Cisco IOx platform



Source: (De La Mora, 2014, 25)

A seamless application enablement framework and compute platform across various devices operating at the network edge are provided by the IOx, having the ability to host applications and services, connecting them in a secure and reliable way to applications in the cloud. The term Application enablement covers all life cycle aspects of applications including development, distribution, deployment, hosting, monitoring and management. Traditional Computing Model versus IoT Computing Model based IOx platform are presented in figure no. 2.

Figure no. 2. Traditional vs IoT Computing Models


Traditional Computing Model
(Terminal-mainframe, Client-server, Web)
DATACENTER/CLOUD
ENDPOINT
IoT Computing Model
(Data Volume, Security, Resiliency, Latency)
DATACENTER/CLOUD
IOx
DEVICE

Source: (De La Mora, 2014, 25)

3. Security Issues in Fog Computing

IoT devices, such as fog nodes and back-end clouds, form what is known to be the Fog. Security represents one of the biggest issues when it is talked about fog computing. This technology involves various providers such as: the fog provider (the entity that deals with providing the infrastructure to the users), the services provider (the entity that uses the infrastructure in order to deliver applications/services to end users) and the consumers of the services provided (the entity that uses the services stored in infrastructure). Each element has its own security management system and everyone has their own requirements and capabilities. As a result, all the parts can lose control over data, especially the providers of fog – who are not always aware of the content and the security requirements stored on their infrastructure – and the consumers – who cannot control the security of their own data.

The client must be able to protect data as it travels between fog nodes and the cloud by using

Cisco cybersecurity solutions (Cisco, 2016). This must be provided before, during, and after attacks. For example, detection of anomalous activity using Cisco NetFlow, Cisco TrustSec, and Cisco Identity Services Engine (ISE). Prevention of breaches using Cisco Advanced Malware Protection is also very important. Responding to anomalous activity by automatically enforcing security policy may be also done. With Cisco Intrusion Prevention System (IPS), the security policy can take into account the target of the threat. In IT environments, the response to a threat might be to quarantine or shut a system. In an Operational Technology (OT) environment, the response to the same threat might be to alert system operators who have the knowledge to decide on the best action.

The security of stored data represents one of the most significant issues within the fog computing security area. In this environment, all the data is stored withing a third party, which makes data security to become the main security concern. This way, the traditional security solutions cannot be implemented directly.

Moreover, cryptographic methods can be used. By using these methods to ensure the security within Fog, the consumers do not have any control over the data stored in the Fog. As a result, there is no explicit knowledge of the data stored.

Fog computing is known to be managed through data centers working in a cooperative manner, distributed protocols having a key role in ensuring the security system on Fog storage. Also, when attacks happen, nobody can be identified. Moreover, it cannot be detected which file was hacked.

Most security incidents regarding Fog computing are posed by the traditional hostile attacks. Man-in-the-middle attacks will become increasingly common in the Fog area, in which gateways which are operating as fog devices may be compromised or replaced by mock devices. Depending on the scenario, it may be difficult to protect the communication between the fog nodes and the IoT devices using encryption method. A downside to this is represented by the well-known fact that encryption and decryption methods consume large amount of battery on mobile device. (Stojmenovic and Sheng, 2014)

Intrusion detection systems come as a support to analyze and monitor the access control policy, as well as the log file in order to detect intrusion behavior. This system can be run to detect DoS attacks (Petac, Alzoubaidi and Duma, 2013) and port scanning. Because of the limited resources IoT devices have, it is more difficult to detect rootkits than the other attacks. These can cause several issues regarding the information extraction by having higher privileges than the embedded hypervisor.

Another problem is represented by malicious detection technique within the environment. This can be used whenever some fog nodes are compromised, the technique detecting malicious code. It is a facility that is combined with signature-based detection technique and behavior-based detection technique (Wu and Irwin, 2015). The first technique being more useful because of its lower costs.

We propose an Adaptive Fog Computing Node Security Profile (AFCNSP) based on security Linux solutions. For fog nodes it is very important to decide what security protections will be necessary. A fog node application cannot access the network, the services and the data of other application without authorization. Simplified Mandatory Access Control Kernel – SMACK (Schaufler, 2016), Discretionary Access Control – DAC (Schneider, 2016), Cynara (Tizen, 2016) and netfilter (Gheorghe, 2006) are Linux kernel security modules that protects data and process interactions from malicious manipulation by using a set of custom mandatory access control rules. The functional requirements that appear in AFCNSP include the following security services: Authentication, Authorization, and Accounting (AAA); Mandatory, Discretionary, Role and Rule Based Access Control; Symmetric and Asymmetric Encryption; Data Integrity; Audit. AFCNSP provides the required security services and assurances in order to process administrative, private, and sensitive pieces of information. The moment when something or someone compromises a fog node, its system recovery and manageability tools can proceed to clean up the system. These tools can also prevent future attacks and identify the portions of the system that are no longer trustworthy. AFCNSP is based on the following security solutions with the purpose to protect fog node: Virtual Private Network (VPN) connections, Secure Socket Layer/ Transport Layer Security (SSL/TLS) certificates, Secure Shell (SSH) authentication, firewalls, Service auditing, File Auditing, Intrusion Prevention Systems (IPS), Intrusion Detection Systems (IDS).

4. Conclusion

The fog computing architecture is one of the most and newest promising models for service providers, fog providers and users. But for this model to be used in a proper way, the existing security issues must be solved. Based on the issues presented above, it can be summarized that some of the problems of fog security are as follows:

- Some of the security problems come from technologies used such as virtualization and service-oriented architectures;
- Security management is difficult to control, given the number of requirements;
- The Fog should have a structure ensuring access to data in the cloud just after passing through security levels previously.

Some solutions that are recommended to be taken into consideration are the inherence in Fog architecture - when the delivery mechanism and user interfaces should provide flexible interfaces security, the support and integration with other security methods for different layers to ensure better security and a continuous adaptation to environmental changes and to the needs of providers. With this purpose, an Adaptive Fog Computing Node Security Profile (AFCNSP) based on security Linux solutions is proposed. Embedded developers using AFCNSP will get improved security of the fog node with richer feature sets.

5. References

1. BETaaS, 2012. *Building the environment for the things as a service.* [online] Available at: <http://www. http://www.betaas.eu/description.html> [Accessed 25 April 2016].
2. Cisco, 2012. *Cisco IOx: Making Fog Real for IoTBuilding the environment for the things as a service.* [online] Available at < http://blogs.cisco.com/digital/cisco-iox-making-fog-real-for-iot > [Accessed 25 April 2016].
3. Cisco, 2015. *Fog Computing and the Internet of Things*: Extend the Cloud to Where the Things Are. [online] Available at: <http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf> [Accessed 20 April 2016].
4. Cisco, 2016. *Security Solutions*. [online] Available at: < http://www.cisco.com/c/en/us/products/security /solution-listing.html> [Accessed 20 April 2016].
5. Fadlullah, Z., Quan, D., Kato, N. and Stojmenovic, I., 2014. GTES: an optimized game-theoretic demand-side management scheme for smart grid. *IEEE Systems Journal,* 8(2), pp.588–597.
6. Gai, S., Salli, T. and Andersson, R., 2011. *Cisco Unified Computing System (UCS): A Complete Reference Guide to the Cisco Data Center Virtualization Server Architecture*. 2nd ed. Indianapolis: Cisco Press.
7. Gheorghe, L., 2006. *Designing and Implementing Linux Firewalls and QoS using netfilter, iproute2, NAT and l7-filter*. [online] Available at: <https://www.packtpub.com/sites/default/files/SampleChapter-Designing-and-Implementing-Linux-Firewall-and-QOS.pdf> [Accessed 25 April 2016].
8. Hong, K., Lillethun, D., Ramachandran, U., Ottenwälder, B. and Koldehofe, B., 2013. Mobile fog: a programming model for largescale applications on the internet of things. In: ACM (Association for Computing Machinery), *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing (MCC'13)*, Hong Kong, People's Republic of China, 12 August 2013. New York: ACM.
9. IDC, 2015. *Explosive Internet of Things Spending to Reach $1.7 Trillion in 2020, According to IDC (IDC # 256397).* [online] Available at: < http://www.idc.com/getdoc.jsp?containerId=prUS25658015> [Accessed 25 April 2016].
10. Korzhyk, D., Conitzer, V. and Parr, R., 2011. Solving Stackelberg games with uncertain observability. In: IFAAMAS (International Foundation for Autonomous Agents and Multiagent Systems), Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems - volume 3 (AAMAS '11), Taipei, Taiwan, Republic of China , 02-06 May 2011. Richland: International Foundation for Autonomous Agents and Multiagent Systems.
11. Li, C. and Shimamoto, S., 2012. An open traffic light control model for reducing vehicles CO2 emissions based on etc vehicles. *IEEE Transactions on Vehicular Technology,* 61(1), pp.97–110.
12. Linux Foundation, 2016. *Yocto Project*. [online] Available at: < https://www.yoctoproject.org/> [Accessed 20 April 2016].
13. Madsen, H., Albeanu, G., Burtschy, B. and Popentiu-Vladicescu, Fl., 2013. Reliability in the utility computing era: towards reliable fog computing. In: IEEE (Institute of Electrical and Electronics

Engineers), *Proceedings of the 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Bucharest, Romania, 07-09 July 2013. New York: IEEE.

14. Maharjan, S., Zhu, Q., Zhang, Y., Gjessing, S. and Basar, T., 2013. Dependable demand response management in the smart grid: a stackelberg game approach. *IEEE Transactions on Smart Grid,* 4(1), pp.120–132.
15. Mazalov, V., 2014. Mathematical Game Theory and Applications. New York: John Wiley & Sons.
16. Nishio, T., Shinkuma, R., Takahashi, T. and Mandayam, N.B., 2013. Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud. In: ACM (Association for Computing Machinery), *Proceedings of the First International Workshop on Mobile Cloud Computing and Networking* (*MobileCloud'13)*, Bangalore, India, 29 July-1 August 2013. New York: ACM.
17. Petac, E., Alzoubaidi, A.,R. and Duma, P., 2013. Some experimental results about security solutions against DDoS attacks. In: IEEE (Institute of Electrical and Electronics Engineers), *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS 2013)*. Iasi, Romania, 11-12 July 2013. New York: IEEE.
18. Python Software Foundation (PSF), 2016. Python. [online] Available at: <https://www.python.org/> [Accessed 20 April 2016].
19. Schaufler, C., 2011. *Smack Project*. [online] Available at: < http://schaufler-ca.com/home> [Accessed 20 April 2016].
20. Schneider, B., F., 2012. *Discretionary Access Control*. [online] Available at: < https://www.cs.Cornell.edu / fbs/publications/chptr.DAC.pdf> [Accessed 25 April 2016].
21. Stojmenovic, I. and Sheng, W., 2014. The Fog Computing Paradigm: Scenarios and Security, Issues, In: IEEE (Institute of Electrical and Electronics Engineers), *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS'14)*, Warsaw, Poland, 7-10 September 2014. New York: IEEE.
22. Tizen, 2016. *Security: Cynara*. [online] Available at: < https://wiki.tizen.org/wiki/Security:Cynara> [Accessed 25 April 2016].
23. Wu, C.,H. and Irwin, J., D., 2015. *Introduction to Computer Networks and Cybersecurity.* Boca Raton: CRC Press.
24. Zhou, B., Cao, J., Zeng, X. and Wu, H., 2010. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In: IEEE (Institute of Electrical and Electronics Engineers), *Proceedings of the 72nd Vehicular Technology Conference(VTC 2010-Fall)*, Ottawa, Canada, 6-9 September 2010. New York: IEEE.