Software Defined Membrane: Policy-Driven Edge and Internet of Things Security

Massimo Villari and Maria Fazio
University of Messina

Schahram Dustdar
TU Wien

Omer Rana
Cardiff University

Lydia Chen
IBM Zurich Research Lab

Rajiv Ranjan
Chinese University of Geoscience

The Internet of Things (IoT) is the latest evolution of computing technology, incorporating potentially billions of devices (such as cameras, sensors, RFIDs, smart phones, and wearables). It is not owned or coordinated by any central authority, but is a heterogeneous mix of devices, components, lightweight OS's, technologies, and protocols, from different organizations and by individuals deploying and using them for their own purposes. There are currently 6.4 billion IoT devices in use around the world (according to Gartner). Their number, capabilities, and scope of use keep growing and changing rapidly. Gartner also forecasts that the number of IoT devices will reach 20.8 billion by 2020, and that IoT service spending will reach $1,534 billion, and hardware spending $1,477 billion by this period. Similarly, the volume of generated data and computing/storage requirements of IoT applications will continue to increase. However, security and data privacy remain major challenges in the use of such device in a complex environment. We illustrate that software-defined membrane can agilely integrate security policies that enables resilient and dependable migration of microservices/data among edge and cloud resources.IoT technologies are introducing many billions of Internet Connected 'Devices' or 'Things' where programmability remains a major feature. Vendors are increasingly adding additional capability into their devices without fully realizing the potential security implications that such features introduce. Edge devices (sensors, actuators, mobile phones, surveillance cameras, routers, gateways, and switches) ubiguitously monitor the cyber and physical worlds. Similarly, IoT devices provide unprecedented ability to collect data, but also necessitate timely processing of the data collected. This requires intelligent approaches to reduce the network latency as well as the cost of processing. Designing security measures for IoT is particularly challenging due to the heterogeneity (types, data formats, firmware, etc.) of devices, leading to potentially a range of attack vectors that are not relevant for other types of computing infrastructure.[1] Some refer to the increasing take-up of such devices as leading to an "untrusted internet". Recently, various reports have emerged of insecure IoT and edge device deployments inadvertently exposing personal or corporate data. However, the introduction of additional capability at the network edge creates both security challenges and opportunities. One issue that has been highlighted in the recent past is whether microservices deployed on edge devices are more secure than those deployed on remote cloud-based data centers. For example, a malicious attacker able to replace a microservice could compromise the subsequent processing of sensor data and any decisions that are reached on such data. On the other hand, new opportunities with edge computing include the ability to aggregate and anonymize data close to sensors to thwart an attack in the remote data center, or in the network link connecting the edge and the cloud.

The Osmotic Computing Paradigm and MicroELements

To foster integration of edge computing and cloud computing datacenters, in a previous Blue Skies column we proposed the "Osmotic Computing" paradigm (hereby referred to as OSMOSIS) that focuses on techniques and mechanisms to expand IoT capabilities exploiting edge and cloud resources by identifying, designing, and implementing a new computing model.[2] The benefits of integrating different computing paradigms in this way, such as edge and cloud computing, have already been acknowledged by both academic and industry-based initiatives, including Cisco, Amazon Web Services (e.g., Snowball Edge and Greengrass), and the Open Fog Consortium. Overall, IoT applications and services deployed in an OSMOSIS system can be viewed as a graph of MicroELements (MELs), where a MEL can be composed of two types of software components: 1) MicroServices, that implement specific functionalities and which can be deployed and migrated across different virtualized infrastructures, and 2) MicroData, that represents a piece of information flowing from and to IoT sensor and actuator devices, and which may occur in a variety of domain-specific data formats.[3] OSMOSIS goes beyond simple elastic management of deployed MELs since deployment and migration strategies are related to requirements of both infrastructure (e.g. load balancing, reliability, availability) and applications (e.g. sensing/actuation capabilities, context awareness, proximity, quality of service (QoS), security, privacy), can change over time, and have specific secuity requirements.

In contrast to existing academic and industry-based initiatives in IoT and edge computing, OSMOSIS focuses specifically on enabling the dynamic management of composed MELs across IoT devices, edge devices and micro-datacenters, and cloud data centers. To overcome IoT resource heterogeneity, the MEL abstraction enables us to support a virtual environment that can be adapted based on the

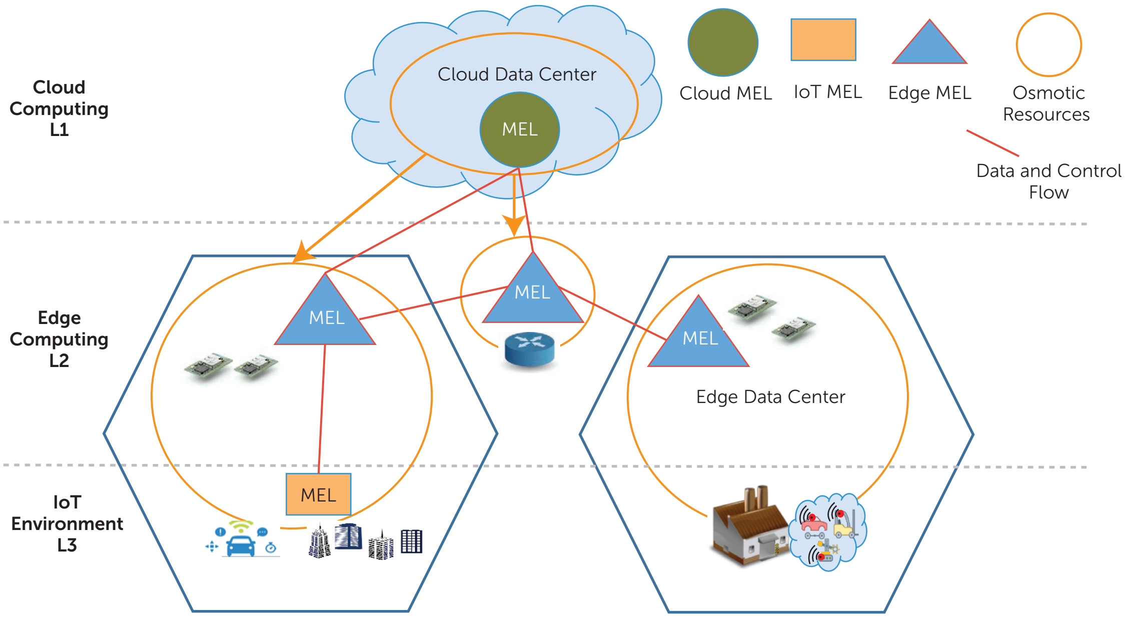

FIGURE 1. Components in OSMOSIS and the MEL abstraction.

underlying infrastructure and/or IoT devices. Based on particular performance triggers (e.g. latency, response time, battery usage, etc) and data flow policies (e.g. security and privacy aspects) MELs can migrate from IoT or edge devices to one or more cloud data center (and vice versa). For example,

- Mapping a MEL that analyses sensitive sensor data to the nearby IoT gateway in the Edge micro-Data Centre may reduce security risks and data transfer network latency but it could lead to slower data processing due to inferior hardware features of gateway as compared to Cloud Data Centre hosted servers.
- On the other hand, encrypting and anonymizing sensitive data on IoT devices may thwart a security attack while data is being transferred to the MEL hosted at a nearby IoT gateway and/or remote data centre for processing. However, this could lead to the draining of battery power for the resource-constrained IoT sensor and/or gateway.

Figure 1 shows MELs at different layers. MELs at layer L2 are deployed in multiple embedded devices (e.g. IoT gateways such as Raspberry Pi) at the network edge. Gateway nodes undertake operations (average, min, max, filtering, aggregation, etc.) on data streaming from L3. Often, they capture data with predefined frequency (e.g. dictated by the rate of change of the observed phenomenon), depending on the device capacity to record/collect data and also based on specific system requirements that must be satisfied. Conversely, they might process distributed queries or commands received from either layer L3 or layer L1, returning a response or command to the same or a different layer. It should be clear that this is a simplified abstraction for purposes of exposition: some geographical, organizational, architectural, or jurisdictional areas might have only two layers, whereas others might have multiple intermediate layers. Layer L1 might not be a single cloud data center, but multiple regions and availability zones, multiple clouds, a hybrid cloud, a multicloud, multiple clouds with an intermediate market, broker, or exchange or combinations of any or all of these.[4]

At layer L1, more complex computational and storage capability is made available to MELs, enabling more specialized (for example quantum, HPC, AI, etc.) and/or complex, generally long running simulations and/or analytics to be carried out on the data. Each infrastructure type (IoT, edge, and cloud) also has its own objective function, influencing the types of operations carried out. For instance, edge (L2) generally consist of resource constrained devices (limited battery power, network range, etc.); operations must be performed in the context of these constraints. Hence, storage and compute capacity at the Edge must be shared across multiple concurrent data flows (possibly from L3), so analysis is limited by the number of flows and time constraints in carrying out the filtering/pre-analysis. Operations in Cloud (L1) are based on pre-agreed targets between a client and a data center provider, e.g. throughput, response time, cost. Understanding how an application hosted on a Cloud at L1 can interact and coordinate with IoT (L3) and Edge (L2) is a key research challenge, particularly for real time, streaming applications. Driven by QoS, sensing, actuating (L3), and security/privacy requirements the MEL can be distributed across Cloud, Edge, and IoT. Distributing analysis of data across these different infrastructures can improve overall IoT application performance and reduce core network load. OSMOSIS suggests that a MEL is not confined to a specific type of infrastructure (or location); the same MEL can be realized across various different resources types with varying levels of complexity.

At layer L3, IoT devices communicate based on standardized protocols such as the Constrained Application Protocol (CoAP), supported through Erbium REST interface and specialist operating systems such as Contiki and RiotOS, and network functions as Open Virtual Network (OVN). A network virtualization abstraction, such an OVN (an open source framework originally launched by the Open vSwitch team at Nicira (now part of VMware)) can be usefully adopted to support MEL interaction over a network and device-centric, data-centric, and network-centric security.

OSMOSIS enables management of deployed MELs in response to QoS, security/privacy requirements, and run-time perturbations (see Figure 1). This approach decouples IoT application deployment from infrastructure management, and makes it possible for MELs to migrate from Cloud to the Edge and/or IoT devices and vice versa. For instance, consider a situation where several IoT devices are collecting large volumes of data at L3. Consider also that, given the network stability and capacity, the amount of data produced and its subsequent transmission to a Cloud data center (L1) are unsustainable from a networking point of view. Analizying this data in the Cloud would become unfeasible, resulting in the current system not being able to continue functioning. Using the OSMOSIS approach, upon detecting such a bottleneck, the OSMOSIS approach shifts processing of some of the data to the Edge (L2).

Case for OSMOSIS Security Membrane

Borrowing from the chemistry analogy (which is the motivation behind OSMOSIS), a membrane regulates the flow of molecules across the membrane in solutions with different solute concentrations. In OSMOSIS, the "membrane" concept strictly relates to the control of MEL flows in the whole system, hence allowing us to manage security and privacy issues. A membrane enables grouping and filtering of MELs based on their properties and use. Membranes among different sub-systems allow MELs to migrate, subject to constraints identified in the Membrane, guaranteeing isolation of one system from another. This isolation can be seen as a security mechanism, limiting the impact of particular violations on sub-parts of the overall system being managed.

The Software Defined Membrane (SDMem) in our OSMOSIS architecture acts as a filter to limit how MELs can be migrated, and under which context. We propose realizing security and privacy mechanisms for supporting the seamless flow of MELs in OSMOSIS, contextualized as the realization of a "Security SDMem". Security in OSMOSIS involves two aspects: i) design and specification of MEL; ii) migration and management of MEL, subject to security and privacy policies.

The Attacker Security Model

Attacker Security Model involves developing potential attack scenarios to protect against particular types of security violations and vulnerabilities, thereby leading to consideration of mitigation strategies to prevent/minimize disruption due to such attacks. Hardware-based secure cryptographic approaches can also be made use of for Edge (L2)

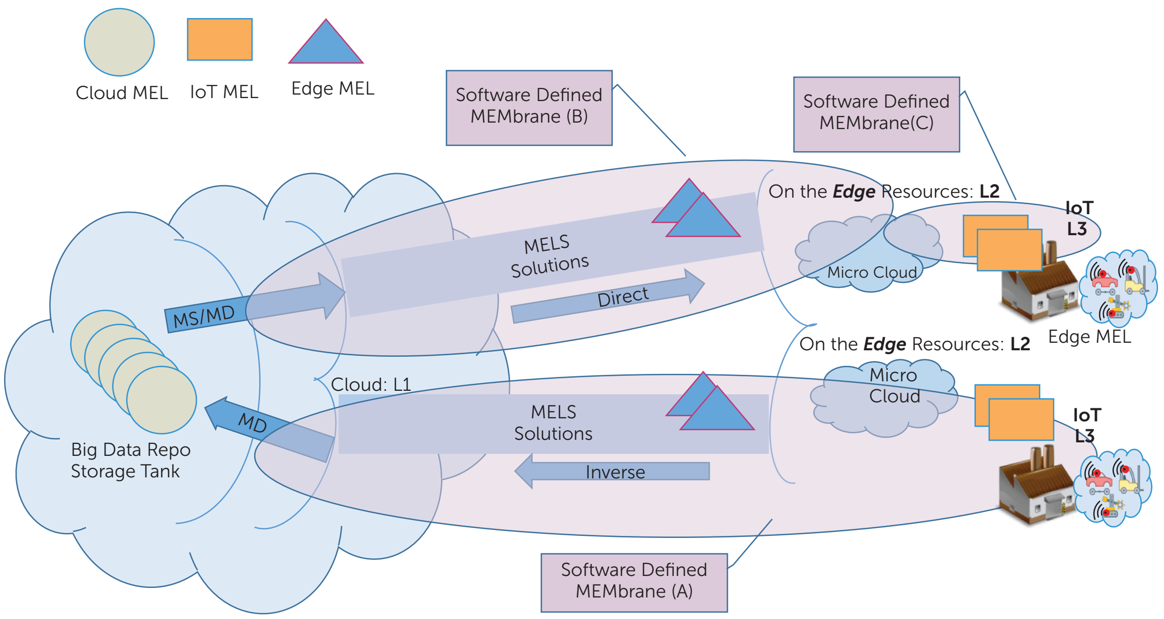
Cloud MEL
IoT MEL
Edge MEL
Software Defined MEMbrane (B)
Software Defined MEMbrane(C)
On the *Edge* Resources: L2
IoT L3
MELS Solutions
Micro Cloud
MS/MD
Direct
Edge MEL
On the *Edge* Resources: L2
Micro Cloud
Cloud: L1
MD
MELS Solutions
IoT L3
Big Data Repo Storage Tank
Inverse
Software Defined MEMbrane (A)

FIGURE 2. Software-defined Membrane (SDMem) in OSMOSIS.

and IoT (L3) devices to support such mitigation strategies, where these are available. However, such approaches could incur additional performance overheads if implemented. The attack scenarios we consider can be either device or network-centric, and may target the scarcity of IoT resources (e.g. CPU, memory, battery) or the difficulty of updating their firmware with security patches, which typically leads to inadequate security. In the wider context, it is also necessary to consider attacks due to human error or intervention. We consider the following types of attacks:

- Device-centric attacks (requiring physical access to the device): reverse engineer firmware to recover passwords and unencrypted sensitive data. Creating a “side channel” to access sensitive data held on the resource, without knowledge of device owner;
- Network-centric attacks (possible with just remote access): i) rogue devices in the local network: sniffing, Man-in-the-Middle attack, SSL-strip for TLS interception, delete/replace/inject data in the connection between the IoT device and cloud, aiming to disrupt the data transfer protocol; ii) remote attack: unsanitized data injection into the device aiming to penetrate the device, fuzzing attack; iii) availability attack: denial of service attack, sleep deprivation attack; iv) attack towards Cloud/IoT: spoofing the identity of one to the other; v) Combinations of the above attacks in attack trees. The overall objective is to consider issues around: breaking Confidentiality, Integrity, Availability and Authentication. We summarize the attack types from [2,5,6,7] by their model, risks, attacks, and targets in Figure 1.

Software Defined Membrane (SDMem) for Security Management

The Attacker Security Model discussed above can be used to specify how security policies can be used to support SDMem. In Figure 2, three SDMems (A, B, and C) are defined, corresponding to different application contexts over shared resources and MELs. The SDMem is responsible for creating isolated networks among MELs, based on *communication tunnels*. The C SDMem creates an isolated context where a Gateway (on the Edge) can communicate with all the deployed MELs.

Communications use the same Internet I/O protocols (for example CoAP) and the same security

TABLE 1. Attack Types.[1,7–9]

Attacker Security Model	Risks	Attacks	Targets
Device	Data Leakage	Data Sniffing and Man-in-the-Middle	Transmitted data
Device	Spoofing/Identity Modification. This could also be Selective Forwarding of data	Sybil Attack	Insertion of inaccurate data; masquerading as another device
Device	Energy/Resource Bleeding	Barrage and Deprivation of Sleep State	System operation and battery
Device & Network	Service Disruption	Denial of Service	System availability and reachability
Network	Network Function Disruption	TCP SYN flood and/or ACK spoofing	Data transmission
Network	Smurf attack	Internet Control Message Protocol and Broadcast	Data transmission

capabilities (such as Advanced Encryption Standard (AES)) encryption on hardware in order to encrypt MELS inside the C SDMem.

If the Attacker Security Model highlights a possible vulnerability (for example in a Smart Factory IoT 4.0), each SDMem with specific MELs will rearrange the virtual system in order to satisfy security requirements. The B SDMem should take into consideration any extra needs, e.g. avoiding a Distributed Denial of Service (DDoS) attack (availability: A of CIA). Here, IoT and Cloud resources are responsible for creating the B Membrane to deal with the DDoS attack. Example new membranes might be created to resetup new communication tunnels. From Figure 2, various confidentiality approaches can be used. For Membrane C, it is possible to use simple Encryption and Hashing Algorithms (such as ECC, AES, SHA-256, and so on) whereas in Membrane B the use of Homomorphic Algorithms may be used. We therefore identify *Encryption modules* to support this. In this way, OSMOSIS allows providers to easily customize a proper security level.

Future efforts to be considered in context of SDMem should focus on the following perspectives: first, MEL design has to follow the security patterns that can guarantee a high degree of resiliency while MELs are executed and/or migrated across heterogeneous infrastructure (IoT, Edge, Cloud), along with the needs to secure APIs. Second, new networking abstractions, such as Network Function Virtualization (NFV) and Service Function Chaining can be exploited to develop end-to-end secure communication tunnels and novel packet-level analysis techniques. Third, developing a secure API for remote orchestration of heterogeneous edge and IoT devices exploiting Software Defined Networking and NFV capabilities would provide a useful research challenge. While doing so, the research efforts also need to cater for relative lack of resources (CPU, memory, battery) at IoT and edge layers and difficulty in updating firmware with security patches—typically leading to inadequate security.

Related Work on IOT Security

IoT security is becoming increasingly important due to the deployment of a huge number of IoT devices (sensors and actuators) across various sectors, including Industry 4.0, Smart Cities, Healthcare, Homes, Buildings, and Smart Cars. The survey describes many existing IoT protocols and presents open research issues, specifically in the area of security.[5] The authors highlight key security requirements as privacy, anonymity, liability and trust, which will be fundamental for social acceptance of future IoT applications employing Internet integrated sensing devices. Internet-originated attacks such as Denial of Service (DoS) are a challenge to solve, and availability and resilience are also relevant requirements to consider. Mechanisms will also be required to implement protection against threats to the normal functioning of IoT communication protocols, an example of which may be fragmentation attacks that the 6LoWPAN adaptation layer suffers. Finally, security mechanisms designed to protect communications using the main IoT protocols must provide appropriate assurances in terms of confidentiality, integrity, authentication, and non-repudiation of the information flows. Indeed, security of IoT communications may be addressed in the context of the communication protocol itself. In this domain, other work includes securing IoT communications, as well as the Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT that uses the homomorphic Paillier encryption, Chinese Remainder Theorem, and one-way hash chain techniques.[6]

Conclusion

We describe the integration of security strategy as a programmatic "Security Software Defined Membrane", to enable optimal migration of microservices and their data (referred to as MELs), between Edge and Cloud resources. The SDMem is configured based on a security policy informed by an "attacker model" – this involves a user or security analyst specifying potential attack types and resources that will be impacted. In this work, we focus on security concerns rooted in devices and the network between them. The approach advocated here can be adapted based on changes in the type of Edge and/or IoT device being considered.

Acknowledgement

We would like thank Joe Weinman for this valuable comments and edits.

References

1. D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to Networking Cloud and Edge Datacenters in the Internet of Things," *IEEE Cloud Computing*, vol. 3, no. 3, May-June 2016, pp. 64–71.
2. M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic Computing: A New Paradigm for Edge/Cloud Integration," *IEEE Cloud Computing*, vol. 3, no. 6, Nov. –Dec. 2016, pp. 76–83; doi:10.1109/MCC.2016.124.
3. M. Fazio, A. Celesti, R. Ranjan, C. Liu, L. Chen, and M. Villari, "Open Issues in Scheduling Microservices in the Cloud," *IEEE Cloud Computing*, vol. 3, no. 5, Sept.–Oct. 2016, pp. 81–88; doi:10.1109/MCC.2016.112.
4. J. Granjal, E. Monteiro, and J. Saa Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, 2015, pp. 1294–1312.
5. R. Lu, K. Heung, A. Lashkari, and A. Ghorbani, "A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IOT," *IEEE Access*, vol. 5, 2017, pp. 3302–3312.
6. M. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," *Int'l J. Computer Applications*, vol. 111, no. 7, 2015.
7. D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis—A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things," *Proc. 37th IEEE Conf. Distributed Computing Systems* (ICDCS), June 5–8, 2017.
8. E. Anthi, A. Javed, O. Rana, and G. Theodorakopoulos, "Secure Data Sharing and Analysis in Cloud-Based Energy Management Systems," 2*nd* EAI Conf. Cloud, Networking and IoT (CN4IoT), May 2017.

MASSIMO VILLARI *is an associate professor of computer science at the University of Messina. His research interests include cloud computing, distributed systems, wireless network, security systems, big data analytics and Internet of Things. He has a PhD in computer engineering from the University of Messina. He is a member of IEEE and IARIA boards. Contact him at mvillari@unime.it.*

MARIA FAZIO *is an assistant researcher in computer science at the University of Messina (Italy). Her main research interests include distributed systems and wireless communications, especially with regard to the design and development of Cloud solutions for IoT services and applications. She has a PhD in advanced technologies for information engineering from* *the University of Messina. Contact her at mfazio@unime.it.*

SHAHRAM DUSTDARD *is a full professor of computer science heading the Distributed Systems Group at TU Wien, Austria. His work focuses on Internet technologies. He is an IEEE Fellow, a member of the Academy Europeana, and an ACM Distinguished Scientist. Contact him at dustdar@dsg.tuwien.ac.at.*

OMER RANA *is a full professor of performance engineering in the School of Computer Science and Informatics at Cardiff University, where he also leads the Internet of Things (IoT) laboratory. His research interests include performance modelling, simulation, and scalable algorithms for cloud computing, IoT, and edge analytics. Contact him at o.f.rana@cs.cadiff.ac.uk.*

LYDIA CHEN *is a researcher staff member at the IBM Zurich Research Lab, Zurich, Switzerland. Her research interests include modeling, optimizing performance and dependability for big data applications and highly virtualized datacenters. She received a PhD in operations research from the Pennsylvania State University. Contact her at vic@zurich.ibm.com.*

RAJIV RANJAN *is reader in the School of Computing Science at Newcastle University, UK; Chair Professor in the School of Computer, Chinese University of Geoscience, Wuhan, China; and a visiting scientist at Data61, CSIRO, Australia. His research include grid computing, peer-to-peer networks, cloud computing, Internet of Thinks, and big data analytics. Ranjan has a PhD in computer science and software engineering from the University of Melbourne (2009). Contact him at raj.ranjan@ncl.ac.uk.*