R. Roman, R. Rios, J. A. Onieva, and J. Lopez, "Immune System for the Internet of Things using Edge Technologies", *IEEE Internet of Things Journal*, vol. 6, pp. 4774-4781, 2019.
http://doi.org/10.1109/JIOT.2018.2867613
NICS Lab. Publications: https://www.nics.uma.es/publications

Immune System for the Internet of Things using Edge Technologies

Rodrigo Roman, Ruben Rios, Jose A. Onieva, and Javier Lopez, *Senior Member, IEEE*

***Abstract*—The Internet of Things (IoT) and Edge Computing are starting to go hand in hand. By providing cloud services close to end-users, edge paradigms enhance the functionality of IoT deployments, and facilitate the creation of novel services such as augmented systems. Furthermore, the very nature of these paradigms also enables the creation of a proactive defense architecture, an immune system, which allows authorized immune cells (e.g., virtual machines) to traverse edge nodes and analyze the security and consistency of the underlying IoT infrastructure. In this article, we analyze the requirements for the development of an immune system for the IoT, and propose a security architecture that satisfies these requirements. We also describe how such a system can be instantiated in Edge Computing infrastructures using existing technologies. Finally, we explore the potential application of immune systems to other scenarios and purposes.**

***Index Terms*—Security, Internet of Things, Edge Computing, Immune Systems.**

I. Introduction

THE human immune system (HIS) [1] consists of a group of cells, proteins and organs that protect the body from external pathogens such as bacteria, viruses and fungi. The main tasks of the HIS are (i) recognizing and neutralizing external harmful substances and pathogens, and (ii) recognizing body cells that have mutated due to illnesses and fighting against them. To realize these tasks, the HIS has two parts that work together and complement each other to protect the body from different types of hazards.

There is an innate immune system and an adaptive or acquired immune system. The innate immune system provides general defense mechanisms against common pathogens. On the contrary, the adaptive immune system generates highly specific protection mechanisms against unusual threats. After the body has been exposed to an unfamiliar threat, it generates a special type of protein, called antibody, which can be attached to immune cells to help them recognize that particular pathogen in the future.

The immune system counts on different kinds of immune cells, which are generally referred to as leukocytes or white blood cells. The macrophages are immune cells programmed to recognize invaders and serve as body sentinels. When these cells encounter an intruder, they release small proteins, called cytokines, to call for reinforcements. The most common reinforcement cells are the neutrophils. These are short-lived immune cells whose main duty is to kill the invaders detected by macrophages. Another important HIS player is the natural killer cell, which defends against cells infected by viruses and tumor cells.

Most living organisms have an immune system and the Internet of Things (IoT), despite being a computational system, introduces some novel features that make it resemble a living organism. First, IoT networks consist of a multitude of IoT devices – cells – distributed over a wide geographical area, which can be static or mobile. Second, IoT devices are capable of perceiving the environment with their built-in sensors and interact with it by means of different types of actuators. This changes the context surrounding the device, which in turn introduces new changes at all levels of the IoT deployment. Third, typical IoT deployments lack sufficient computing and processing power to defend from and adapt to all kinds of anomalies. Finally, just like an organ is vital for the correct operation of the human body, malfunctioning IoT deployments might pose a serious threat to the operation of an entire system. Maintaining IoT security is a serious challenge, mainly due to the complexity in the integration of security mechanisms [2] and the existence of vulnerable and misconfigured IoT systems [3].

Cloud computing technologies have been considered crucial for the deployment of IoT platforms [4], [5] mainly due to their ability to complement the intrinsic hardware constraints of the majority of IoT devices. Continuing our simile, it could be said that the Cloud becomes the brain that interacts and controls the rest of the body. However, it has been shown that the centralized nature of the Cloud imposes some important limitations regarding latency, bandwidth utilization, mobility support, and so on. These limitations have motivated the emergence of Edge Computing paradigms [6], which rather than relying on a single cloud server at the backbone of the network, distributes computing resources in a three-tier architecture, bringing computational resources closer to end devices (see Fig. 1). As a result, the elements of the IoT infrastructure can be distributed from the cloud to the edge, creating a scalable hierarchical infrastructure which is not constrained by the aforementioned limitations.

Note that the inner tier of the architecture may consist of several layers of edge devices itself and may be hierarchically organized. Each of these edge devices can be regarded as a mini-cloud server, which may be deployed in a cellular tower, a dedicated in-house computer, a single-board computer, and so on.

However, the protection of edge computing infrastructures

The authors are with the Network, Information and Computer Security (NICS) Lab, Computer Science Department, University of Malaga, 29071 Spain, e-mail: (see https://www.nics.uma.es/contact).

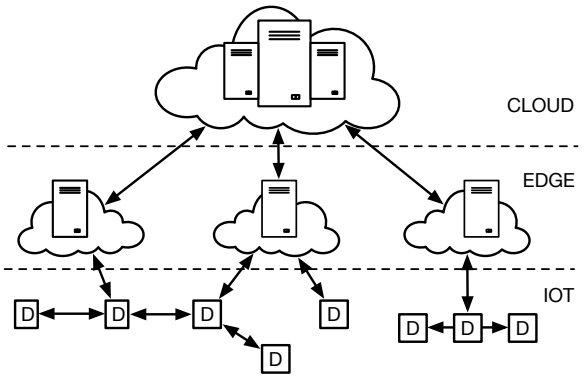

Fig. 1. Edge Computing Architecture

and their applications, including the IoT, presents its own challenges [7], [8]: the vulnerability of Edge Computing as a new computer paradigm; the operation of edge nodes and the deployed virtual machines in physically exposed environments; the need for interoperability between the edge nodes and IoT devices with very different capabilities; the lack of resources in IoT deployments and other related paradigms like sensor networks [9], etc. These issues introduce the need to provide additional mechanisms for the protection of IoT infrastructures as a whole.

Given the similarities between living organisms and IoT deployments, together with the security challenges faced by the edge infrastructure itself, in this paper we propose the design of a virtual immune system for protecting the devices in the IoT. In particular, we define the architectural components and interfaces of virtual immune systems and cells, which include mechanisms for not only recognizing the presence of invaders and alert any relevant systems, but also for proactively analyzing the state of IoT entities – including the existence of vulnerabilities and misconfigured systems. One of the main benefits of our proposal is that it leverages edge technologies to relieve constrained IoT deployments from this workload.

The rest of this paper is organized as follows. First, we review related works in Section II. Next, in Section III we elicit the requirements that should satisfy a virtual immune system for the IoT. The following two sections are devoted to the definition of the architecture and its instantiation based on current standardization efforts. Further applications of the proposed immune system are discussed in Section VI. Finally, Sections VII and VIII present an analysis of the elicited requirements and the conclusions of our proposal.

II. Related Work

Various researchers have used the human immune system (HIS) as a model to design and implement artificial immune systems (AIS) as intrusion detection systems (IDS) [10], some of them even in the context of cloud computing [11]. These implementations mainly follow three paths: (i) methods inspired by the HIS and employ conventional algorithms, (ii) the negative selection paradigm, and (iii) methods based on the danger theory. The first category of methods uses established techniques (e.g., fuzzy matching) but orchestrated in different stages to mimic HIS responses. The second category of methods uses negative selection algorithms that generally define self patterns, generate non-self detectors and monitor the occurrence of anomalies using these detectors (see for instance [12], [13]). The third category of methods relies and puts the focus on the immune system triggering effect. The AIS is triggered when a small amount of damage is observed at an early attacking phase. Note that most of these solutions are limited to the detection of ongoing attacks against a network, and do not provide a holistic solution that i) could be applied to edge computing scenarios, and ii) could integrate other proactive detection mechanisms.

In terms of the creation of an AIS-inspired architecture for edge computing scenarios, to the best of our knowledge, there exists only one solution [14] that introduces a cell-based Fog infrastructure for intrusion detection. However, unlike our approach, their solution does not comply with the *on-demand*, *flexibility*, and *adaptability* properties that a virtual immune system should have (cf. section III). Their immune cells must be deployed at all times at the sensors layer, which imposes a heavy burden upon the IoT network. Moreover, their architecture is focused on the deployment and execution of a single AIS-based intrusion detection mechanism. It does not provide explicit support for the deployment of other mechanisms, for the interaction with external sources of information, and for the integration of IDS mechanisms with the monitored IoT systems. Besides, it should be noted that it is possible to integrate the specific AIS-based IDS defined in [14] in our approach.

Finally, beyond the existence of cloud-specific IDS solutions [15], there are several works whose goal is to provide security services to edge-based IoT deployments. For instance, Rongxing et al. [16] provide a lightweight privacy-preserving data aggregation scheme that filters out false data at the network edge. Wang et al. [17] design a label-based access control to protect IoT data caching from sabotage in edge nodes and to further ensure their reliability. Also, the authors in [18] propose a general and simplified architecture instantiation for securing IoT devices without specific solutions but using general approaches to security. Their main contribution is identifying the edge components in which these security elements (e.g., an IDS like Snort) are deployed.

III. System Requirements

The realization of an immune system for the IoT based on edge technologies poses a number of challenges. These challenges respond to the fulfillment of some requirements that are expected to be addressed by the system. Such requirements match those exiting in the HIS.

A *virtual immune system* (VIS) for the IoT must support the creation and delivery of different kinds of virtual immune cells. Virtual immune cells (VIC) are the basic component of our immune system and they can be implemented as virtual machines, linux containers, or other forms of virtualized entities. Each type of VIC is specialized in performing a particular task, such as monitoring the infrastructure (just like macrophages), the execution of audits, and so on. Thus, our

virtual immune system must be *flexible* and provide virtual immune cells for different tasks.

As in the human immune system, VICs must be able to move and interact with other elements of the environment. The edge infrastructure should be able to deploy, execute, suspend and migrate virtual immune cells to and between edge devices. Therefore, VIC *mobility* is a requirement for our system. As such, our virtual immune cells need to be *lightweight* enough to facilitate a rapid migration. This is also relevant since providing them with plenty of resources is not the primary function of the edge nodes where they are executed.

The previous requirements are also in line with the ability to provide immune services *on-demand* at the right time and location. Instead of filling up the body with a huge number of immune cells, the HIS creates them when necessary and deliver them to the area of potential infections. This same property is deemed useful to our VIS as the deployment of virtual immune cells through all the edge infrastructure might be costly, not only from the point of view of resources but also in economical terms.

We also consider the need to adapt to the presence of new threats just like the acquired immune system does. For example, if a new vulnerability that affects our IoT devices is discovered, our immune system must be able to internalize this information (i.e., create virtual antibodies) and launch the necessary actions to prevent security breaches. Note that information about vulnerabilities may be self-detected (e.g. using negative selection algorithms) or received from external sources, resembling vaccines. Therefore, the VIS should be sufficiently *adaptable* to enable the detection and neutralization of novel threats as well as to extend this capacity to other IoT deployments.

Finally, the infrastructure must be able to discern actual threats from virtual immune cells, since their actions may be in conflict with the security policy established by the edge infrastructure. In the case that the infrastructure does not provide the required supervision services, the immune system could take on the responsibility for monitoring the correct operation of virtual cells performing a task similar to that of natural killer cells in the HIS. This means that virtual immune cells should perform no other actions than the ones that have been agreed upon. In any case, there must exist a *Security Operations Level Agreement* (SOLA) that must be enforced either by the edge infrastructure or the virtual immune system itself.

IV. System Architecture

This section describes the architectural components and the way they interact with each other and with the environment to satisfy the requirements presented in Section III.

As shown in Fig. 2, our virtual immune system consists of two functional parts: the VIS Kernel, deployed in the Cloud and comprised of various components, and the virtual immune cells (VIC), executed in the edge. An important component of the VIS Kernel is the *VIS orchestrator*, which is in charge of making decisions on the configuration and deployment of VICs within the edge infrastructure. Additionally, the VIS

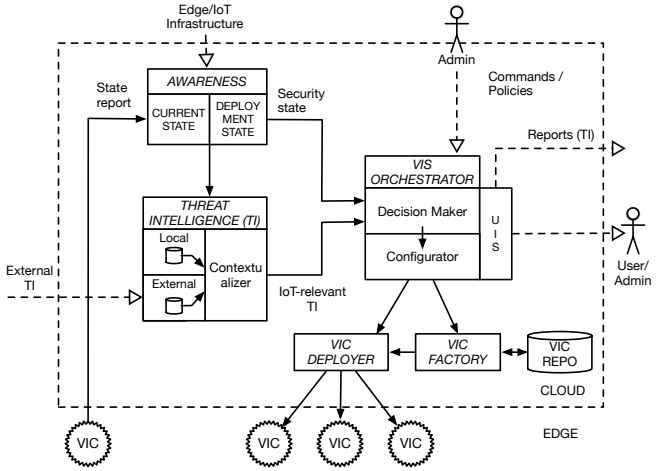

Fig. 2. VIS Architecture

orchestrator generates reports intended to human actors or other systems thanks to the user interface subcomponent (UIS) module embedded in this component. To fulfill these tasks, the VIS orchestrator receives information from various sources, both internal and external.

One of the inputs to the VIS orchestrator comes from system administrators in the form of commands or policies. The VIS orchestrator is assumed to make decisions autonomously once deployed, but it should also be able to receive commands eventually from authorized entities. For example, a system administrator may want to deploy a given number of cells of a particular type in a critical location, say next to a nuclear plant, to improve the security in that area.

The *Threat Intelligence* (TI) component allows the VIS to adapt to new threats in a similar way as the acquired HIS does. This component provides the VIS orchestrator with threat intelligence information that is relevant to IoT devices in the deployment. This information, which can be regarded as virtual antibodies, is obtained by digesting external threat intelligence feeds distributed by non-profit organizations, industry groups, vendors or even other VIS platforms [19]. These feeds will be delivered using standard formats, such as the Structured Threat Intelligence eXpression (STIX). Also, threat intelligence information can be received from the Awareness component. The *contextualizer* module is in charge of discerning which TI information is relevant to the IoT deployment based on the inputs from the Awareness component.

The *Awareness* component collects information from two sources, namely the Edge/IoT infrastructure and the VICs deployed by the virtual immune system. The data obtained from the edge and IoT infrastructures give insight into the state of the IoT deployment itself, and facilitates the interaction between the immune system and the monitored IoT entities. This is what we depict in Fig. 2 as deployment state, which includes details about the location and addresses of IoT devices, the interfaces and protocols used by such devices and their gateways, the hardware and software platforms in use, the credentials that are required to interact with the IoT infrastructure, etcetera. To this end, the Awareness component needs

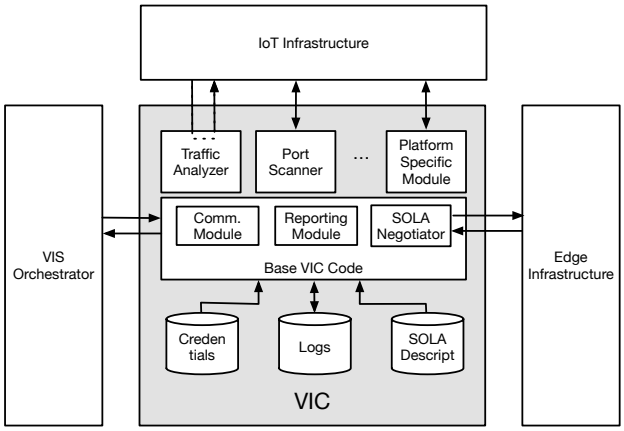

Fig. 3. VIC Architecture

to define communication interfaces to interact with the IoT infrastructure manager. This component includes also current state information obtained from changes in the infrastructure or most importantly from the virtual immune cells themselves.

The *virtual immune cells* (VICs) can be regarded as the leukocytes of our VIS. They are the only components deployed in the edge infrastructure and prior to their distribution, the VIS orchestrator determines the type of cells that are necessary and where they should be deployed. This information is passed on to the VIC Factory and VIC Deployer components, respectively. The *VIC Factory* is instructed by the VIS orchestrator on which services to incorporate into the VICs to be deployed. To fulfill this task, the VIC Factory relies on a VIC repository containing off-the-shelf security services which can be further configured with the indications of the configurator module. In this way, we may create many types of virtual cells, including macrophages-like VICs that include intrusion detection services, and neutrophils-like VICs with vulnerability scanners and configuration testers. The *VIC Deployer* takes already configured VICs and deploys them into the edge infrastructure according to the indications of the VIS orchestrator.

As shown in Fig. 3, VICs consist of a base code and additional services, which are added depending on the desired functionality. The base code is intended to provide the mechanisms for interacting with the VIS Kernel and the edge infrastructure. VICs must be able to report status data and other information to the kernel components, as well as receive commands or updates from them. Like any deployed virtual object, this exchange of information (cytokines) will be done through secure channels. As for the interaction with the edge infrastructure, it is basically done by means of the SOLA negotiator module. This module is in charge of presenting the set of operations (together with the necessary credentials) to be performed by the VIC. After reviewing both the credentials and the SOLA descriptors, the edge infrastructure may or may not allow the VIC to operate. This helps to prevent malicious entities from performing undesirable actions on the IoT deployment or the edge infrastructure. This functionality is similar to that of natural killer cells in the HIS.

VICs also interface with the IoT infrastructure. The modules configured by the VIC Factory component are responsible for this interaction. Some of these modules will establish direct connections with IoT devices, while other modules will instead interact with the IoT gateway (cf. section V-B for a more detailed discussion on this subject). Direct communication is usually desirable when probing for vulnerabilities in particular IoT platforms. Conversely, taking advantage of the gateway makes sense in operations like network traffic analysis and intrusion detection.

Finally, note that the information gathered by virtual immune cells is susceptible to be logged for further processing even though status reports are periodically sent to the VIS Kernel. Also, as a security measure, the scope of the security credentials provided to the IoT Infrastructure should be limited to the operations that the virtual immune cell will perform.

V. System Instantiation

A. Deployment and Interaction with the Edge

In order to instantiate our architecture, we need to focus on both the current and the short future of edge-based IoT deployments being devised by standardization bodies and consortia, and the specific modules defined in Section IV and how they can be implemented with existing technologies. In this subsection we put the focus on identifying the general mapping existing between our architecture and the interfaces provided by emerging edge-related IoT architectures.

For the deployment of our architecture in edge infrastructures we need to take into account that the deployment model of the virtual immune system is closely related to the model used by existing edge-based IoT platforms [20]. In such platforms, a cloud orchestrator deploys and manages various virtual IoT gateways, which are deployed close to the IoT objects they interact with. In the case of our architecture, the VIS kernel deploys and manages multiple virtual immune cells, which in turn will be deployed close to the virtual IoT gateways and IoT objects they monitor.

Therefore, our immune system architecture will utilize the very same edge interfaces that are used to deploy and manage edge-based IoT platforms. According to the Multi-Access Edge Computing (MEC) reference architecture [21], users can interact with the Operations Support System through the Customer Facing Service portal in order to get virtual appliances running in a desired location of the network. As for Fog Computing, its reference architecture [22] mentions the existence of edge interfaces that allow cloud platforms to distribute services into the fog. Although the exact nature of such distribution mechanisms is yet to be fully described, it has not only been explored how to transmit virtual services through the use of lightweight virtualization technologies (e.g containers, unikernels) [23], but also certain specifications have already defined potential mechanisms that could be used to migrate virtual appliances [24]. Regarding the actual virtualization of computational and network resources, it will be enabled by various building block technologies such as Software-Defined Networking (SDN) and Network Function

Virtualization (NFV), although other solutions are also being considered [25].

There are other issues that must be considered at this level. For example, the VIS must be able to set up the VICs within the system it is monitoring. This requires instantiating the cells in the very same virtual network where the monitored systems are deployed. While the enabling technologies that make this task possible are known (e.g. SDN), at present the exact edge interfaces that should facilitate this integration are not defined. Nevertheless, existing specifications, such as the specification that describes the deployment of MEC in 5G-based NFV environments, already define the interfaces and components (such as the VNF Forwarding Graph, or VNFFG) that will be used internally by the architecture to interconnect different virtual machines with each other [26]. Note that these networking capabilities and services provided by certain edge approaches (e.g. SDN and dynamic slicing) can also be used to implement more advanced response mechanisms. For example, traffic from malicious entities can be reconfigured (e.g. reducing its QoS), or even routed to other subsystems like honeypots for further testing.

Lastly, there is the issue of informing edge nodes about the purpose of VICs through the use of Security Operations Level Agreements (SOLA). Without the SOLA, an edge node will not be able to differentiate a virtual immune cell from a malicious entity trying to probe the system. Although current edge infrastructures do not consider the existence of such type of agreements, it might be possible to provide support for SOLAs by using and/or extending already existing components and interfaces. For example, the MEC reference architecture [27] specifies that all virtual machines will provide application descriptors to describe the capabilities of the application, plus other application requirements and rules. Certain descriptors, like *ServiceDependency*, might be used to indicate the location of the services that the cell will test, alongside with a set of requested permissions (e.g. type of test, time of testing, frequency of testing). Then, the MEC platform can make use of the monitoring mechanisms of the virtualization infrastructure to oversee the behaviour of the virtual immune cell and check whether it is complying with the permissions indicated in its SOLA or not. In the case that actions are performed outside the limits established by the SOLA, the infrastructure could take actions against the cell.

B. Architectural Components Implementation

In this subsection we identify different technologies, tools, configurations and choices that allow us to carry out the implementation of the different modules identified in Section IV.

When implementing our architecture, we will need to integrate different technologies to provide the described functionality. One of the necessary features is the ability to place and deploy VICs in specific locations. The VIC Deployer needs for the existence of secure geolocalized virtual services provision within the infrastructure. Other virtual machine migration routing mechanisms might be in place, but we consider this property more suited for covering affected and neighbouring areas. Ideally, this functionality has to be inherently secure so that compromised edge nodes are excluded as a destination. A proof-of-concept solution that addresses these issues in cloud environments is described in [28].

The VIC Factory can fulfill its functionality with different strategies. Starting from a base virtual unit, the automatic configuration of virtual immune cells can be realized using an XML-based approach. Either i) the base unit, which can follow OVF (Open Virtualization Format), integrates a startup script that after the first execution downloads the software needed to perform its duties (e.g. pentesting, configuration audits, IDS, etc.) or ii) all the possible and distinct VICs are pre-created.

The first implementation choice allows a VIC to *mutate* into a different kind of cell and is therefore more *flexible*. It also results in lighter virtual images but further network communication and storage resources are needed when is to be executed. Moreover, mutation is not a property exempt from risk since allowing this flexibility requires to protect new components (e.g. software repositories). Static VICs allows for less flexibility but lacks the risk of VIC mutation if proper mechanisms to avoid changes are put in place in the cell itself.

In the same way we foresee different approaches to manage the configuration of IoT deployments. Static information could be stored in XML format and be either retrieved from edge nodes acting as IoT gateways, the cloud or received through JSON messages from specific VICs in charge of configuration and topology testing. That is, it could be received from the IoT infrastructure or VIC State arrows depicted in Fig. 2. In case of information retrieval, the Awareness component needs to present valid credentials to be authorized to request these data. These credentials can be securely obtained from the IoT infrastructure manager upon the deployment of the virtual immune system.

Finally, various tools can be used by VICs to perform their duties. There are multiple intrusion detection mechanisms specifically designed for IoT environments, which can be executed within the VICs [29]. Besides, it is also possible to integrate solutions that delegate more costly operations (e.g. machine learning and/or data mining [30]) to the cloud. Other tools, like vulnerability testing mechanisms and configuration analysis systems, can be used to detect a potential problem before it is exploited. Regarding vulnerability testing, not only more traditional pen-testing tools specifically designed for IoT environments [31] are available, but other researchers are also experimenting with more advanced vulnerability testing mechanisms based on fuzzing techniques [32]. On the other hand, the research on solutions that analyze the validity and security consistency of the configuration of an IoT system is extremely limited. Still, various researchers have studied how to model the behaviour of IoT devices, including the representation of their behaviour through automatic learning [33], formal models [34] or profile specifications [35]. Such models can become the foundation for analyzing configurations in the near future.

We have to note, however, that the capabilities of platform-specific modules in VICs are heavily dependent on the features of the monitored platforms. As aforementioned, the Awareness and VIC Factory components can configure a VIC with

the information it needs, such as credentials, protocols, and platform restrictions (e.g. end device limitations, including energy and communications). VICs can also be deployed in promiscuous mode in the same network as the monitored platform thanks to the services provided by the edge (e.g., VNFFG). This way, all the interfaces (e.g., REST interfaces) from the IoT platform are available for testing, and the traffic sent directly from IoT devices to edge IoT gateways can be captured and analyzed. Yet there will be situations, such as when IoT traffic is aggregated in a local router outside the edge infrastructure, in which the capturing capabilities of our modules become limited. In those cases, it is necessary to interact more closely with the components of the IoT platform. For example, in the FIWARE middleware [36], a VIC module could register itself as an IoT Device Manager or IoT Gateway in order to have access to the native IoT device interfaces.

VI. Further Applications

In this section, we show that the characteristics of the proposed architecture make it a suitable candidate for protecting scenarios other than the IoT, as well as for solving other issues in edge computing platforms.

a) Virtual immune systems in other scenarios: The concept of a VIS can be applied to other edge scenarios, such as optimized local content distribution, augmented reality solutions, edge-enhanced vehicular networks, and others [37]. In all these scenarios, a set of virtual appliances are deployed at the edge of the network to provide various services (e.g., cache of multimedia streams), and such appliances could be queried by specialized virtual immune cells deployed by virtual immune systems.

The adaptation of the VIS described in this article to these scenarios involves various aspects that require further study. Among them, it is important to consider the influence of the specific requirements of the scenario. Although the elements and components of the architecture have been designed according to requirements that apply to most, if not all, edge scenarios (cf. sections I and III), certain scenarios have very specific needs that will require of the integration of novel components. For example, in the case of vehicular networks, one of those requirements are the stringent timing requirements in the management of beacon and safety messages between cars. It is then mandatory for the immune system to respect the operation of these critical processes. Therefore, in vehicular scenarios, the architecture should incorporate a set of policy components, which may allow or deny the deployment of virtual immune cells based on both the state of the monitored component and the overall status of the system.

b) Virtual immune systems and the edge infrastructure: The concept of VIS can also go beyond the protection of user-deployed applications, being used by edge infrastructures themselves to analyze their own public interfaces and configurations. This could facilitate the creation of automated edge security management systems, which provide a real-time status report of the security of the elements of the edge infrastructure. Such systems could be also used as a foundation for the integration of more proactive security solutions (e.g. network slicing reconfiguration, replacement of components).

As shown in Section V, many of the building blocks that are required for the creation of edge-controlled virtual immune cells are already in place. In fact, one of the basic use cases of MEC [37] and 3GPP 5G [38] involves the management of virtual appliances that can be deployed by edge operators or infrastructure providers. This way, it may be possible to analyze the security of infrastructure edge services (e.g. bandwidth manager) located in edge nodes.

Still, the amount of challenges that must be overcome in order to create such edge VISs are numerous. Some of those challenges are related to the challenges of our IoT virtual immune system, like the definition and negotiation of the Security Operations Level Agreement. Other challenges are related to the actual deployment of the system components: due to the need to monitor an entire edge infrastructure instead of a single centralized entity it is necessary to consider the development of a federated/distributed architecture of VIS Kernels, which collaborate in the management of all elements. Finally, some challenges are related to the security of the virtual immune system itself, and the need to control its behaviour: the more privileges a virtual immune cell has, the higher the risk for a potential malicious actor disrupting the infrastructure.

c) Accountability and service assurance at the edge: In this article, we have described how virtual immune cells are able to analyze the security and correctness of any element or service deployed by third parties at the edge. However, due to the flexibility of the cells, it is also possible to incorporate mechanisms to analyze if the services deployed at the edge comply with certain quality of service metrics (e.g., response time, service availability). Moreover, virtual cells can also retrieve and validate information about the edge infrastructure itself, including the network bandwidth available to the cell, the location where the virtual cell is deployed, etc.

By embedding this functionality into virtual cells, it is possible to create the foundation of a distributed audit system. This element is, in fact, one of the components of Cloud Accountability [39]. Although Cloud Accountability is mainly focused on verifying whether cloud providers comply with the requirements of data protection regulations, it also takes into account the need to verify the claims made not only by third-party services deployed at the edge but also by edge providers themselves [40]. Therefore, an evolved version of VIS could be used as one of the first components of Edge Accountability systems.

VII. Analysis

In the following list, we analyze how the different functionality provided by our VIS architecture satisfies the requirements identified in Section III.

- *Adaptability*. The continuous feeding of the system with internal (current state) and Edge/IoT infrastructure configuration and anomaly information, dumped into the Awareness and Threat Intelligence components, allows it to adapt to raising attacks, deployment reconfiguration and computation and networking loads. The adaptation can therefore occur in different dimensions: how (what

type of VICs are involved), when (the VIS might delay temporarily VICs placement if edge resources are scarce at the moment), where (direct deployment to suspicious locations), etc.

- *On-demand*. Since the VIC Deployer can place VICs in specific IoT deployments using the information provided by the Awareness component, there is no need to have an homogeneous distribution of VICs throughout the edge infrastructure. It is straightforward to observe that the virtual immune system can act on-demand when the adaptability property is fulfilled. Nevertheless, the Decision Maker needs to be configured to operate in such a manner.
- *Flexibility*. The VIS needs to respond to different types of threats. Also, when possible, proactively. Note that this requires different types of VICs. These are either preloaded with different functionality (test modules, non-self detectors, etc.) or mutated from pre-estasblished base virtual units. The latter allows for an additional degree of flexibility but introduces additional risks, as already mentioned in Section V-B.
- *Lightweight*. VIC migration needs this property to be fulfilled in order to make our system effective. Using base VIC units, the migration of virtual machines is considered to be optimal, but in contrast, that requires later network communication to download the required modules. On the contrary, pre-created VICs suffer from longer migration times but are light, specialized virtual units that are ready to work as soon as they get to their destination. Additionally, due to the existence of different types of VICs to perform different functions (see Section IV), the footprint is lighter than in case of a virtual appliance trying to implement a complete testing and analysis system.
- *Mobility*. In order to fulfill this property we rely on the interfaces provided by the edge infrastructure (MEC and OpenFog are suitable candidates) as explained in Section V-A.

In Section III we have not considered the *survivability* of our Immune System as a requirement. There are several reasons for this. As aforementioned, VIC Kernel components are assumed to reside in the Cloud. And existing commercial Cloud solutions already implement various services that help to maintain the availability of the system, such as resources replication and fault tolerance. Besides, even though already deployed VIC cells might lose connection with these central element, they will still be able to carry on with their mission: they are factorized and deployed to be as autonomous as possible, and can store reports for later communication when the central elements are recovered. If needed, also migration could be carried out autonomously.

VIII. Conclusions

The IoT is an inherently insecure paradigm that demands for novel security mechanisms and architectures to lower the risk of attacks. In this paper we have devised a virtual immune system that leverages edge computing technologies to satisfy this need. The proposed system makes decisions on the number and type of virtual immune cells to be deployed in the edge infrastructure based on the input from different sources. Virtual immune cells are distributed on demand, and they are capable of negotiating with the edge infrastructure their ability to perform the tasks they have been created for based on the credentials they posses.

We have shown that the proposed architecture is practical and it can be implemented by taking advantage of the functionality provided by current edge computing architectures. Moreover, we have depicted implementation choices for the components of the architecture, and provided additional case applications that could benefit from our virtual immune system.

Still, we consider that a more formal study of the impact of the infrastructure core point of failure (e.g., how the functionality can be recovered when only one of the central components fails) is needed, and indeed it is the next step we plan to analyze before starting the development of a proof of concept.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the SMOG (TIN2016-79095-C2-1-R) project. R. Rios is funded by the 'Captación de Talento para la Investigación' fellowship from the University of Malaga.

References

[1] L. M. Sompayrac, *How the Immune System Works*, 5th ed. Wiley-Blackwell, 2015.
[2] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.
[3] M. Hypponen and L. Nyman, "The Internet of (Vulnerable) Things: On Hypponen's Law, Security Engineering, and IoT Legislation," *Technology Innovation Management Review*, vol. 7, pp. 5–11, 2017.
[4] P. P. Ray, "A survey of IoT cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1, pp. 35–46, 2016.
[5] B. Zhang, N. Mor, J. Kolb, D. S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E. A. Lee, and J. Kubiatowicz, "The Cloud is Not Enough: Saving IoT from the Cloud," *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud'15)*, 2015.
[6] R. Milito, "Fog in Support of Emerging IoT Applications," Fog Computing Conference and Expo, 2014.
[7] B. A. Martin, F. Michaud, D. Banks, A. Mosenia, R. Zolfonoon, S. Irwan, S. Schrecker, and J. K. Zao, "OpenFog security requirements and approaches," in *Fog World Congress (FWC'17)*, 2017, pp. 1–6.
[8] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.
[9] A. J. Perez, S. Zeadally, and N. Jabeur, "Security and privacy in ubiquitous sensor networks," *Journal of Information Processing Systems*, vol. 14, no. 2, pp. 286–308, 2018.
[10] J. Kim, P. J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection – a review," *Natural Computing*, vol. 6, no. 4, pp. 413–466, 2007.
[11] R. Zhang and X. Xiao, "Study of danger-theory-based intrusion detection technology in virtual machines of cloud computing environment," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 239–251, 2018.
[12] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *IEEE Computer Society Symposium on Research in Security and Privacy*, 1994, pp. 202–212.
[13] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principles of a computer immune system," in *ACM SIGSAC New Security Paradigms Workshop*, 1998, pp. 75–82.

[14] F. Hosseinpour, P. V. Amoli, J. Plosila, and T. Hämäläinen, "An Intrusion Detection System for Fog Computing and IoT based Logistic Systems using a Smart Data Approach," *JDCTA International Journal of Digital Content Technology and its Applications*, vol. 10, pp. 34–46, 2016.
[15] N. Keegan, S.-Y. Ji, A. Chaudhary, C. Concolato, B. Yu, and D. H. Jeong, "A survey of cloud-based network intrusion detection analysis," *Human-centric Computing and Information Sciences*, vol. 6, no. 1, p. 19, 2016.
[16] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.
[17] Q. Wang, D. Chen, N. Zhang, Z. Qin, and Z. Qin, "LACS: A Lightweight Label-Based Access Control Scheme in IoT-Based 5G Caching Context," *IEEE Access*, vol. 5, pp. 4018–4027, 2017.
[18] C. Aggarwal and K. Srivastava, "Securing IoT devices using SDN and edge computing," in *2nd International Conference on Next Generation Computing Technologies (NGCT'16)*, 2016, pp. 877–882.
[19] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & Security*, vol. 72, pp. 212–233, 2018.
[20] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.
[21] "ETSI GS MEC 003 v1.1.1 - Mobile Edge Computing (MEC); Framework and Reference Architecture," ETSI, Tech. Rep., 2016.
[22] "OpenFog Reference Architecture for Fog Computing," Openfog Consortium, Tech. Rep. February, 2017.
[23] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT Edge Computing with Lightweight Virtualization," *IEEE Network*, vol. 32, no. 1, pp. 102–111, 2018.
[24] "ETSI GR MEC 018 v1.1.1 - Mobile Edge Computing (MEC); End to End Mobility Aspects," ETSI, Tech. Rep., 2017.
[25] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
[26] "ETSI GR MEC 017 v1.1.1 - Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment," ETSI, Tech. Rep., 2018.
[27] "ETSI GS MEC 010-2 v1.1.1 - Mobile Edge Computing (MEC); Mobile Edge Management; Part 2: Application lifecycle, rules and requirements management," ETSI, Tech. Rep., 2017.
[28] M. J. Bartock, K. Scarfone, and L. Feldman, "Implementing trusted geolocation services in the cloud," NIST, Tech. Rep., 2016.
[29] B. B. Zarpelo, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.
[30] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, p. 3, 2018.
[31] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, "Systematically evaluating security and privacy for consumer IoT devices," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, 2017, pp. 1–6.
[32] J. Chen, W. Diao, Q. Zhao, C. Zuo, Z. Lin, X. Wang, W. C. Lau, M. Sun, R. Yang, and K. Zhang, "IOTFUZZER: Discovering Memory Corruptions in IoT Through App-based Fuzzing," in *Network and Distributed Systems Security Symposium (NDSS'18)*, 2018, pp. 1–15.
[33] A. Hamza, D. Ranathunga, H. H. Gharakheili, M. Roughan, and V. Sivaraman, "Clear as MUD: Generating, Validating and Applying IoT Behaviorial Profiles (Technical Report)," p. 10, 2018. [Online]. Available: https://arxiv.org/abs/1804.04358
[34] M. Mohsin, Z. Anwar, F. Zaman, and E. Al-Shaer, "IoTChecker: A data-driven framework for security analytics of Internet of Things configurations," *Computers & Security*, vol. 70, pp. 199–223, 2017.
[35] E. Lear, R. Droms, and D. Romascanu, "Manufacturer Usage Description Specification (draft-ietf-opsawg-mud-24)," p. 61, 2018. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-opsawg-mud/
[36] FIWARE, "FIWARE Architecture Specification," 2018. [Online]. Available: https://forge.fiware.org/
[37] "ETSI GS MEC 002 v1.1.1 - Mobile Edge Computing (MEC); Technical Requirements," ETSI, Tech. Rep., 2016.
[38] 5GPPP Architecture Working Group, "View on 5G Architecture," p. 140, 2017. [Online]. Available: https://5g-ppp.eu
[39] S. Pearson, V. Tountopoulos, D. Catteddu, M. Sudholt, R. Molva, C. Reich, S. Fischer-Hubner, C. Millard, V. Lotz, M. G. Jaatun, R. Leenes, C. Rong, and J. Lopez, "Accountability for cloud and other future Internet services," in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, 2012, pp. 629–632.
[40] C. Fernandez-Gago, V. Tountopoulos, S. Fischer-Hübner, R. Alnemr, D. Nuñez, J. Angulo, T. Pulls, and T. Koulouris, "Tools for cloud accountability: A4cloud tutorial," in *9th IFIP Summer School on Privacy and Identity Management for the Future Internet in the Age of Globalisation*, vol. 457. Springer IFIP AICT, 2015, pp. 219–236.

Rodrigo Roman is a postdoctoral researcher at the University of Malaga, Spain. His main topic of research is the protection of IoT architectures and its building block technologies in various contexts, such as Industry 4.0 and digital homes. In addition, Dr. Roman is currently researching the security challenges of all Edge infrastructures, such as Fog and Multi-Access Edge Computing. He has participated in several Spanish and European research projects, and published more than 40 articles in various international journals and conferences.

Ruben Rios is a postdoctoral researcher at the University of Malaga, Spain. His main research activities are centered on the design and development of solutions for the protection of digital privacy and anonymity with a focus on scenarios with resource-constrained devices. He is also interested in the security of Edge Computing platforms and services. Dr. Rios was awarded the FPU fellowship from the Spanish Ministry of Education and received the prize to the most outstanding Ph.D. thesis from the University of Malaga.

Jose A. Onieva received his PhD degree from the University of Malaga (2006) and since 2011 he has been working as Assistant Professor in the Computer Science Department at the University of Malaga. He has been actively involved in ICT European and national funded information security related projects. He has published in several international journals and conferences in the field of Information Security. He is author of the book "Secure Multi-Party Non-Repudiation Protocols and Applications" published by Springer. Currently he is involved in the research of core security services for edge computing, covert channels and digital evidence. He is Regional Director of the Future Technology Research Association and Editor of the Journal of Convergence.

Javier Lopez is Full Professor at the University of Malaga and Head of the Network, Information and Computer Security Laboratory (NICS Lab). His research activities focus on network & information security and Critical Information Infrastructures. He is currently Editor-in-Chief of the International Journal of Information Security, and member of the editorial boards of the journals Computers & Security, IET Information Security, IEEE Wireless Communication, Journal of Computer Security, and IEEE Internet of Things Journal, amongst others. Prof. Lopez is the Spanish representative at IFIP Technical Committee 11 Security and Protection in Information Processing Systems, and has been former Chair of ERCIM WG on Security and Trust Management (2009-2012) and Chair of IFIP Trust Management WG (2006-09). He is Senior Member of IEEE and ACM