IoT Security
A Comprehensive Life Cycle Framework

Elisa Bertino
Department of Computer Science Purdue University
West Lafayette, IN, USA
bertino@purdue.edu

***Abstract*—Securing the Internet of Things (IoT) is not straightforward because of device heterogeneity, highly dynamic and unprotected environments, and large scale. In this paper, we first outline IoT security risks and a security lifecycle for IoT systems. We then discuss various tools and methods that can be used for the different phases of such a lifecycle. For each such phase, we also discuss research directions. We conclude the paper by outlining a few challenges.**

***Index Terms*—IoT systems; anomaly detection; edge computing; autonomous devices; safety; privacy**

I. Introduction

Internet of Things (IoT) can be defined as the network of physical objects, i.e., things, augmented with electronics, software, sensors, and connectivity to enable objects to acquire data, and exchange data with data with servers, centralized systems, and/or other connected devices based on a variety of communication infrastructures [6]. Notable examples of things include heart monitoring implants, biochip transponders on farm and wild animals, automobiles with built-in sensors, and mobile devices for civil infrastructure monitoring. When IoT is augmented with sensors and actuators, IoT is able to support cyber-physical applications by which networked objects can impact the physical environment by executing “physical” actions, such as unlocking a door.

However, because of its fine-grained, continuous and pervasive data acquisition, and control and actuation capabilities, and the lack of adequate security, IoT expands the cyber attack surface [4]. Increased risks arise because of many factors. IoT systems may not have a well defined perimeters and, because of mobility, they continuously change. In some cases, it is even difficult to determine the types of devices one has in an IoT system. IoT devices are highly heterogeneous with respect to communication capabilities and protocols, and software and hardware platforms. IoT systems may include physically unprotected portions. In many domains, IoT systems are complex systems, consisting of different subsystems – possibly owned/managed by different parties. In addition, IoT systems may include objects not designed to be connected to the Internet. These objects may thus not have any security capabilities - even basic ones like the ability to digitally sign messages. Finally human interactions with all the devices is not scalable. Security breaches often arise because of mistakes in configurations, such as for example wrongly configured firewalls, and inconsistent or incomplete access control rules [3]. Therefore proper administration of security is critical. However, managing security for very large numbers of heterogeneous devices may not be always humanly possible.

Industrial IoT (IIoT) and IoT systems used in critical infrastructures have their own additional risks arising from the long lifecycle of systems, the need for continuous operations, and outdated or nonexistent policy and access management [1]. It is also important to notice that IoT systems that also include actuators may amplify the impact of cyber attacks [37] and undermine human safety.

Because of all those risks, defense techniques well established for the defense of conventional information systems and mobile environments are more difficult to deploy in IoT systems and may not always be adequate. The OWASP Internet of Things Project [25] had identified the most common IoT vulnerabilities and has shown that many such vulnerabilities arise because of the lack of adoption of well-known security techniques, such as encryption, authentication, access control, and role-based access control. A reason for the lack of adoption may certainly be security costs.

However, another reason is that existing security techniques, tools, and products may not be easily deployed to IoT devices and systems, for reasons such as the variety of hardware platforms and limited computing resources on many types of IoT devices [6]. Even well known encryption protocols prove to be very expensive when running on devices with limited computing capabilities especially when multiple encryption operations have to executed concurrently such as in the case of networked vehicles [36] and small drones [40].

In addition, as discussed by Jajali et al. [12], IoT may be deployed in settings ranging from small scales, such as well-being personal devices, to massive scales, such as connected transportation infrastructures. Such a variety makes it very difficult to came up with widely applicable security techniques and security management solutions.

In this paper we introduce a research roadmap toward addressing IoT security, based on a comprehensive security lifecycle. We report research results from a few projects and outline novel research directions. We then conclude by discussing additional challenges in the space of IoT security.

978-1-7281-6739-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CIC48465.2019.00033

II. IoT Main Application Domains

As IoT encompasses a large variety of application domains, characterized by different devices, computing platforms, communication infrastructures, and functions, we can expect security solutions to have different requirements for different domains. We outline a few significant classes of domains and identify related relevant security requirements. It is important to emphasize that in all these areas it is critical to adopt good security practices. However each area may have additional specific requirements that need novel security techniques or extensions of existing techniques. We notice, however, that most IoT systems are based on cloud architectures and/or more recently on edge computing architectures, where the cloud and/or an edge server manages the most computationally expensive part of the applications, such as machine learning-based systems. Thus security of the cloud and of edge computing systems is also relevant [11].

A. Consumer IoT

It refers to IoT devices for simple environment and ambient management applications, such as smart houses, smart appliances, and smart toys – just to name a few. Architectures in this type of applications are simpler and use low-end devices. In terms of security, data confidentiality is critical in order to assure privacy. Authentication of devices and users to devices is also critical to ensure physical security, for example, for making sure that an attacker cannot unlock the door of a house where a smart lock is used [15]. In addition, in many consumer IoT applications several users may be involved in requesting services from the same IoT device; for example, several users may have the authorization to open a door by activating a smart lock [15]. However, some of those users may be revoked this authorization, for example house guests. Therefore, protocols for selective authorization and revocation are a critical requirement.

Safety is also critical. Safety can be undermined for two main reasons. The first is related to security weaknesses that can be exploited by an attacker to carry out attacks with direct and immediate physical consequences. One interesting example related to smart toys is by Valente and Cardenas [34]. They analyzed several smart toys that communicate with children by voice commands. They found that one such toy, because of an insecure approach to manage encryption keys for communication between the toy and the toy application on the cloud, would allow an attacker to inject into the toy malicious voice commands. As such toys are designed to cognitively engage children and become a child's friends, children may likely trust the toys. Thus it would likely that, if such a toy were to tell a child to open the door of the house and walk out, the child would likely execute such action – with obvious safety risks. The second reason is that safety can be undermined by the unforeseen combined actions of different devices. For example, a device may be instructed to sprinkle water if it detects a fire starting in a room in a house. On the other hand, another device may be instructed to close the water if it detects water flooding on the house floor.

B. Medical IoT

It refers to the use of personal medical devices (such as pacemakers) and wellness devices. In addition this category includes devices which are not implanted but are used to inject drugs (such as insulin pumps, and pumps used in hospital for intravenous drug administration) and patient monitoring devices. Privacy is critical for these devices as health information is very sensitive for most individuals. Safety, that is, to make sure that the devices do not harm the patients, is critical in addition to privacy. Resiliency to attack is also critical in that one must assure that a device continues to work even under attack [31]. Therefore, making sure that the software deployed on these devices correctly works and is resilient under attack is critical. A major challenge is that, because of safety requirements, many such devices are closed and it is not possible to install additional software on them, such as software for data encryption. Therefore an important requirement is for the design of these devices to take into account privacy, security, and resiliency.

C. Industrial IoT (IIoT)

It refers to the convergence of different technologies, namely intelligent machines, industrial analytic, and sensor-driven computing. The main goal is not only to enhance operational efficiency but also to introduce novel information services [1]. Intelligent machines refer to machines, such as industrial robots, that have not only mechanical functions but are also able to communicate with other equipment and learn how to lower their own operating costs. Such machines have embedded systems and rich communication capabilities. We include IoT for critical infrastructure as part of IIoT, as they share many characteristics, such as the use of control systems, robotic devices and actuators, and intensive use of data analytics. However, whereas many IIoT systems are owned/managed by a single organization (or few organizations), such as industrial enterprises, IoT systems for critical infrastructures may easily involve subsystems and devices owned by different parties and the number of such parties may be very large and involve organizations of different types, industrial and governmental ones, as well as consumers. As a result, security management is much more complex for IoT systems for critical infrastructure.

As discussed in [1], critical requirements for IIoT include: (i) the need for continuous operations, which requires protection against attacks aiming at disrupting the operations; (ii) and assuring human health and safety. In addition IIoT systems pose specific challenges, such as the long lifecycle of systems, legacy equipment, need for regulatory compliance, that complicate the design of suitable security solutions. A specific sub-area within IIoT is represented by control systems that will be increasingly connected to sensors, devices, other systems and Internet via digital communication capabilities (we refer to them as IoT-enhanced control systems). Unlike IIoT applications mainly focusing on improving operations and on information services, control systems often have real-time requirements and must ensure the continuity of the controlled processes, which again makes the design of security solutions more complex (see [18] for a detailed discussion about the security landscape for industrial control systems). It is important to mention that the security of IoT-enhanced control systems is (will be) critical for the protection of many critical infrastructures (energy, transportation, waste management, etc.).

III. Security Lifecycle

A comprehensive approach for cyber security for IoT systems as well as conventional systems must be organized according to some notion of security lifecycle as this allows one to better organize in-depth defense and determine the security techniques and tools to be deployed at the various defense layers.

The lifecycle that we adopt (see Fig. 1) is based on the following rationale:

- Security preparation is critical in order to prevent and stopping attacks. Preparation activities are the core of cyber security. They include making sure that messages and data at rest are encrypted, and access control systems and firewalls installed. Also making sure that applications are free of vulnerabilities is critical. Because of the large number of security tools and techniques that have been developed over the years, one has many different options to choose from. It is however important to notice that security costs and risks must be taken into account in order to determine the most effective and efficient solutions to deploy.
- However even the best prepared system can still be breached. This is especially true for IoT systems which may have several vulnerable components. Therefore it is critical to continuously monitor the system of interest in order to detect attacks or anomalies that may be indicative of attacks. A real-time anomaly detection system is crucial for enabling quick responses to attacks.
- Once an anomaly or attack has been detected, it is critical to identify the type of attack, the system components affected by the attack – for example IoT devices or network links, and depending on the type of attack identifying the specific portion of the system affected and possibly the entry point of the attack or the source of attack. For example, if an attack aims at disrupting communication by interference it is important to quickly determine the location of the interference source. A real-time diagnosis of the attack or anomaly is critical for two reasons. The first is related to the fact that IoT systems are often dynamically changing systems. Therefore, one needs to gather as much information as possible before the system changes in order to perform very precise analysis of the attack or anomaly. The second is that real-time diagnoses allow one to promptly react to attacks by executing different actions and activities.
- A real-time response to attacks is a crucial step in any defense strategy. Examples of such responses include attack containment, moving critical applications to different systems, block accesses to critical resources. The specific responses however depend on the security goals of the system of interest and the application domain. For example, if the security goal of the system is to minimize data losses, one may activate additional IoT devices so that there is redundancy in data acquisition. Of course response actions are meant to stop or just contain the attack. Then, recovery actions are needed so that the system can return to a full functional state. Finally one has to fix the security vulnerabilities that led to the attack. For example, applications may need to be patched, access control systems and firewall reconfigured, and perhaps new security tools and techniques be deployed.

IV. Prepare and Prevent

There are several security "building blocks" that are critical for preparing a system against attacks. We discuss a few of them in what follows. However an important issue that needs to be addressed as part of preparation activities is where to deploy security tools and systems. We thus also discuss initial work addressing such issue for IoT systems.

A. Access Control

Access control is critical when sensitive data needs to be shared among different parties [7]. Different approaches also exist for enforcing access control policies, including cryptographic mechanisms and access control lists. In the context of IoT, access control models and systems must be designed to address different use cases [4]:

- Controlling which end-user (or application on behalf of which end-user) can access which IoT devices for which purpose (and possibly for how long and/or in which context). One example would be to allow visitors at university to operate a smart lock of lab only for the duration of their visit.
- Controlling which IoT device can invoke functions on other IoT devices or get data from other IoT devices. One example would be not to allow a device regulating plant watering in a back yard to collect data from a smart lock device if the former is an unprotected device and thus more vulnerable to attacks.
- Controlling which information an IoT device may collect from a given physical environment and under which circumstances. An example would be preventing an IoT device with a sound recorder to record sounds when one is at the premises of a customer. Access control mechanisms supporting this type of access control policies have been proposed for mobile smart phones. Under such approaches, the applications running on the mobile phone are dynamically revoked the permission to use the sound recorder based on the context. However its application to IoT devices may be challenging as IoT devices may always not be able to acquire contextual information (such as location).

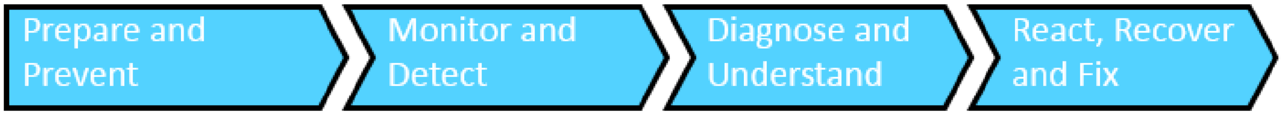


Fig. 1: IoT Security Lifecycle

- Controlling which applications running on a gateway or on the cloud or cloud users can access data from a given IoT device. We mention this purpose for completeness even though cloud security it outside the scope of the discussion. We notice that enforcing access control policies for IoT originated data may require the IoT system to provide relevant meta-data needed for determining the applicable access control policies (such the location where the data were collected by the IoT device).

The definition, implementation and deployment of access control systems for IoT entail addressing a lot of issues including the access control model (e.g. whether it should be attribute-based and context-based), the language used for specifying the authorizations, the architecture for managing and enforcing control. Given the various purposes for access control, it is easy to see that different access control enforcers may have to be used. For example, end-user permissions may be checked at gateways, whereas controlling which information an IoT device may acquire from the physical environment (through the use of its own peripherals) must be executed in the IoT device itself. The specification and administration of access control policies will however be the major challenge. Users already have problems in managing permissions on mobile phones; managing permissions in IoT systems will be undoubtedly much more complex. Approaches for the automatic administration of access control policies, perhaps based on examples or high level guidelines by end-users, will be critical.

B. Application Security

Software is always the critical element of security as many attacks exploit errors in application code. Therefore, a large number of software security techniques have been proposed, such as for ensuring memory safety and the integrity of the application execution control flow. Approaches range from compiler techniques that detect errors, such as techniques to detect buffer overflow vulnerabilities in source code to approaches that instrument or randomize binaries to make sure that even if these vulnerabilities are present in the code, they cannot be exploited by attackers. The latter are typically adopted when the application source code is not available or it cannot be modified. Other approaches do not require modifying neither the source code nor the binaries and rely on monitoring the execution of applications to verify that the execution control flow is not modified by the attacker. However, adapting such techniques to IoT systems is unfeasible without extensive redesign [21] as they have heavy requirements in storage, memory, presence of a memory management unit (MMU), and hardware memory-protection. In addition the performance overhead that existing solutions impose is not acceptable for energy-constrained devices. An approach to spatial memory safety in sensor devices has been proposed that greatly reduces overhead [21]. However, research is needed to extend other available techniques for use in IoT devices.

Securing software also requires detecting and patching logical flaws in application code. Two categories of such logical flaws are insufficient authentication, including authentication bypass [35], and insufficient authorization. Notice that these are logical flaws and not software implementation errors; therefore they cannot be detected by techniques used to detect common errors, such as buffer overflow, or by techniques used to ensure the integrity of the execution control flow. Addressing the problem of logical flaws in applications requires tools to automatically detect the presence of these flaws in code and then patch the code. Notable examples include VuRLE [17] to repair flaws in code implementing cryptographic operations in Android applications and Glaciate [16] to detect authentication vulnerabilities in Android applications. However these techniques are very much language dependent and they would need to be extended to deal with code running on IoT devices.

Understanding which specific techniques or combination techniques should be used in IoT systems for securing software is quite challenging due to the wide diversity of IoT applications and scenarios. Also the selection of one or more techniques may depend on the processes in place for managing software. While we can reasonably expect that an organization using IoT devices for critical applications may be willing to perform detailed analysis of the code, we cannot expect that such an analysis be carried out in the context of consumer IoT. In the end the selection must also be based on a risk assessment analysis. One important advantage to keep in mind, however, is that several IoT devices have very specialized functions and limited and constrained input, and therefore their behavior is predictable. So one could design techniques able to predict actions executed by the IoT devices based on input parameters and use these prediction techniques within monitoring tools to detect anomalies with respect to the predicted behavior. One such approach has been recently developed for the more complex case of database applications [8].

C. Security Provisioning

The problem of distributed security provisioning refers to the problem of the optimal allocation of security resources in a distributed system. This problem has been investigated in the area of conventional networks. The most notable approaches are based on game theory and Pareto analysis [2], [9], [10], [14]. However such problem has not been investigated much for IoT systems. The reason is that the problem is of much larger scale. In addition IoT systems may rapidly change and thus the security provision allocation may also have to change. Preliminary work on the optimal security provisioning for IoT systems has considered static IoT scenarios [29] and mobile IoT scenarios in which, however, the mobility patterns are known in advance [30]. Research is needed to develop advanced solutions for cases in which the mobility patterns cannot be anticipated. The main technical issue is to design approaches by which only portions of the optimal solutions need to be recomputed in order to be able to reallocate the security resources in real-time.

V. Monitor and Detect

Monitoring activities are critical in order to detect attacks and identify anomalies that may indicate attacks. Extensive research has thus been carried out for intrusion and anomaly detection systems (IADSes) for networks. Two popular open source IADSes are SNORT [27] and Bro [26]. Both IDSes rely on network information gathered by packet sniffers, and detect attacks using signature matching over this information. However, their techniques are not applicable to the IoT domain. For example, techniques such as host scanning or port scanning would be ineffective on most IoT systems using IPv6. Also, they only work on traditional IP-only networks, whereas IADSes for IoT need to support a wide variety of mediums and related protocols. Most traditional IADSes rely on a list of rules/patterns to detect attacks. However, while running through a large rule/pattern list is sustainable for a traditional network, small IoT networks would incur heavy overhead on the performance of the IADS. IADSes have also been developed for wireless networks [28]. However, they suffer from one or more limitations with respect to IoT: inability to adapt, applicability only to a single platform and protocol, small and specific range of detection techniques, reliance on the existence of a central control point.

The Kalis system, recently designed, addresses many of the drawbacks of previous systems [22]. Kalis is a self-adapting, knowledge-driven IDS for IoT able to detect attacks in real time across IoT systems running different communication protocols. Kalis autonomously collects knowledge about the features of the monitored network and entities, and leverages such knowledge to dynamically configure the most effective set of detection techniques. Kalis leverages a taxonomy of IoT attacks by target types, that is, the specific types of system components (e.g., such as IoT routers, IoT hubs, devices). It also uses a taxonomy of the relationships between monitored network/entity features and security incidents (Fig. 2 from [22]). By leveraging such knowledge Kalis is able not only to detect attacks and/or anomalies that can be indicative of attacks, but is also able to more accurately identify the type of attacks compared with more conventional IDSes.

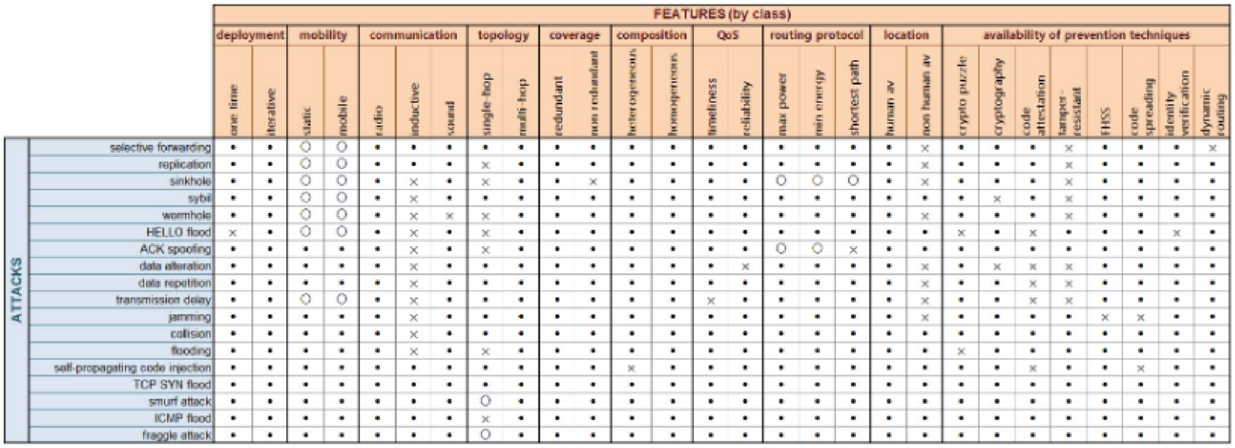
However the design of comprehensive and effective IDSes for IoT requires addressing several challenges. The first is related to the design of IDSes that monitor not only network communications, like Kalis, but also system calls executed by applications running on the IoT devices. However, deploying such an IDS directly at IoT devices may not always be feasible, especially if the IDS uses some complex machine learning techniques [32]. A possible approach leverages edge computing [24] by deploying on the IoT devices just a system call logger, implemented so to make challenging for the malware to both issue calls and tamper the log. The actual IDS engine is deployed at a small edge server, which based on the periodically received logs can execute sophisticated analyses also based on the use of machine learning. An interesting open direction would be to develop community-based IDSes for IoT systems, very much like the community developed around SNORT [27].

The second challenge is related to the automatic acquisition of knowledge about the IoT system of interest. As the experimental results about the Kalis IDS show, the availability of knowledge about the monitored system allows is critical for better security. Relevant knowledge includes knowledge about IoT device types, software running on these devices, specific communication protocols. It is critical to develop techniques by which such knowledge can be automatically acquired by IDSes, perhaps with the support of tools running at a cloud and/or at some edge servers. Initial approaches to the identification of IoT devices have been proposed [19] that, for example, identify IoT device type and vendor based on the DNS query information. However such approaches need to be validated on large scale systems. Also similar approaches need to be developed to identify the applications running on the devices, especially for more powerful devices that may host multiple applications, and network communication protocols. Ideally, we would like to be able to deploy such an "autonomous" IDS in the system of interest and let the IDS acquire by itself all relevant knowledge. Such an IDS would greatly simplify security management.

VI. Diagnose and Understand

Being able to quickly understand security-relevant events is critical for effective defenses. For example, if an IoT device is expected to periodically to collect and send data, any deviation from such expected behaviour (detected perhaps using an anomaly detection system) should be diagnosed in order to understand whether the device has been compromised or the network has been attacked by, for example, a jamming attack. For IoT systems, it is critical that such diagnoses be carried out in real-time.

There are two reasons for such requirement. The first is that IoT systems, because of mobility may rapidly change, and thus it is important to analyze the system while the anomaly is being observed. Delaying such analysis may result in loosing relevant information. The second is that a prompt diagnosis is critical for a real-time response to attacks. In many cases, damages resulting from attacks can be greatly limited if one



		FEATURES (by class)																											
		deployment		mobility		communication			topology		coverage		composition		QoS		routing protocol			location		availability of prevention techniques							
		one time	iterative	static	mobile	radio	inductive	sound	single-hop	multi-hop	redundant	non redundant	heterogeneous	homogeneous	timeliness	reliability	max power	min energy	shortest path	human av	non human av	crypto puzzle	cryptography	code attestation	tamper-resistant	FHSS	code spreading	identity verification	dynamic routing
ATTACKS	selective forwarding	●	●	○	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	×	●	●	●	×	●	●	●	×
	replication	●	●	○	○	●	●	●	×	●	●	●	●	●	●	●	●	●	●	●	×	●	●	●	×	●	●	●	●
	sinkhole	●	●	○	○	●	×	●	×	●	●	×	●	●	●	●	○	○	○	●	×	●	●	●	×	●	●	●	●
	sybil	●	●	○	○	●	×	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	×	●	×	●	●	●	●
	wormhole	●	●	○	○	●	×	×	×	●	●	●	●	●	●	●	●	●	●	●	×	●	●	●	×	●	●	●	●
	HELLO flood	×	●	○	○	●	×	●	×	●	●	●	●	●	●	●	●	●	●	●	●	×	●	×	●	●	●	×	●
	ACK spoofing	●	●	●	●	●	×	●	×	●	●	●	●	●	●	●	○	○	×	●	●	●	●	●	●	●	●	●	●
	data alteration	●	●	●	●	●	×	●	●	●	●	●	●	●	●	×	●	●	●	●	×	●	×	×	×	●	●	●	●
	data repetition	●	●	●	●	●	×	●	●	●	●	●	●	●	●	●	●	●	●	●	×	●	●	×	×	●	●	●	●
	transmission delay	●	●	○	○	●	×	●	●	●	●	●	●	●	×	●	●	●	●	●	×	●	●	×	×	●	●	●	●
	jamming	●	●	●	●	●	×	●	●	●	●	●	●	●	●	●	●	●	●	●	×	●	●	●	●	×	×	●	●
	collision	●	●	●	●	●	×	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	flooding	●	●	●	●	●	×	●	×	●	●	●	●	●	●	●	●	●	●	●	●	×	●	●	●	●	●	●	●
	self-propagating code injection	●	●	●	●	●	●	●	●	●	●	●	×	●	●	●	●	●	●	●	●	●	●	×	●	●	×	●	●
	TCP SYN flood	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	smurf attack	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	ICMP flood	●	●	●	●	●	●	●	×	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	fraggle attack	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Fig. 2: Taxonomy of relationships between IoT network/device features and attacks. Dots and crosses indicate the possibility and impossibility, respectively, of an attack in presence of a specific feature; circles indicate that the appropriate detection technique for the attack depends on the specific feature [22].

is able to quickly take response actions (see next section). The problem of real-time diagnosis of attacks has not been much investigated.

An initial approach for sensor networks has been proposed by Midi et al. [23]. Such an approach focuses on anomalies related to data acquisition and transmission across sensor networks. The approach is able to determine whether the data expected from a sensor in the network are not being transmitted because the sensor has been compromised or because the wireless network is being jammed. The approach uses some simply metrics, namely the link quality indicator (LQI) and the received signal strength indicator (RSSI), to formulate the diagnosis. In the case the diagnosis is that the data are not delivered because of a jamming attack, the approach is able to identify the possible jamming source.

Even though such an approach represents an interesting initial step, developing effective diagnosis tools for IoT systems require addressing several challenges. The first is related to learn and characterize the duration of the network interference to be able to discriminate between naturally occurring interference, such as people walking, and attack-originated interference. The second is related to approaches for detailed IoT device diagnosis to determine, in the event of that an IoT device has been compromised, which software component is affected and by which malware or attack. Such information is critical for two reasons. Knowledge about the former allows one to determine whether the IoT device can still continue to operate, even though with reduced functionality. Knowledge about the latter is critical to determine the goal of the attack and decide response actions to undertake.

VII. React, Recovery, and Fix

When protecting a system from attacks, detection and diagnosis are not sufficient. It is critical to quickly react to the attacks by taking actions that make the system able to continue its operations and at the same time to block the attacks. However, approaches, by which decisions about attack responses are taken by human security analysts and administrators, are not scalable for IoT systems, especially when such systems are large scale and complex. Approaches are needed by which IoT devices (or subsystems of IoT devices) are able to autonomously react to attacks/anomalies, perhaps also based on the information received by tools used for intrusion detection and fine-grained diagnosis, e.g. information provided by the previous steps of the security life-cycle.

An initial approach to support such autonomous attack response action has been proposed for sensor networks by Midi et al. [20]. Such an approach is based on the notion of response policy; such a policy is very much like a database trigger construct. It is defined on a security incident or anomaly and specifies actions for different security estimates, based on various conditions on the incident. Such policies are then deployed at the sensors and automatically executed by the sensors whenever an incident or anomaly is detected. The design of such an approach entails many interesting issues, such as how to deal with systems partitions that may lead to conflicting response actions by sensors in different partitions and with malicious sensors that may try to undermine the responses by other benign sensors. Approaches to such issues are presented in [20].

The development of a full-fledged response approach for IoT systems requires addressing several challenges. The responses very much depend on the relevant security application goals of the IoT system of interest. For example, in the case of the system considered by Midi et al. [20], the security goal was to minimize data losses and thus the response policies included actions to execute redundant data acquisition or transmission. More importantly it is critical to develop a methodology to design such responses, perhaps by extending/applying software design methodologies. Another challenge is that small IoT devices, or IoT devices on which one cannot install software, may not be able to host a response engine. A possible approach may perhaps be based on using edge servers.

VIII. Conclusions

IoT technology enables unprecedented opportunities for novel groundbreaking applications. However, many current and future IoT applications support critical and/or privacy sensitive functions and have safety implications. Therefore comprehensive security is a key requirement. However even deploying well known security techniques, such as encryption, may be problematic. For example, managing public keys for billions of IoT devices may not be trivial; perhaps recent approaches based on blockchain could address this problem [39] (see also discussion by Roesch [28]). In the paper we have discussed relevant techniques and related challenges. However a major challenge that needs to be addressed is how to protect against safety risks arising from security vulnerabilities. As pointed out by Wolf and Serpanos [38], the physical behavior requirements, that IoT devices may have to comply with, may offer attackers new opportunities to undermine safety. For example, the attacker can lead an IoT system into an unsafe state by simply changing the timing of key computations or introducing communication delays. We also notice that as IoT systems, because of being tightly integrated with the physical environment, can be the target of cyber-physical attacks that by manipulating the physical space can prevent IoT devices from working properly. An example of such an attack is by Son et al. [33]. They show that an attacker can incapacitate MEMS gyroscopes by generating in the environment sound noises that have frequencies resonating with the gyroscopes. Being able to test IoT systems is also challenging, especially when they heavily interact with humans and/or are expected to be deployed in highly dynamic physical settings.

However perhaps the most important challenge is security administration. In many cases, especially for consumer IoT, we cannot expect end-users to be able to configure access control lists, firewalls and so forth. We need approaches by which IoT devices and systems can automatically administer security with little or not help by end-users. In this respect a promising approach can be based on the notion of community-wiki for devices [5] to provide security configuration policies that devices can understand and enforce.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] Accenture "Driving the Unconventional Growth through the Industrial Internet of Things,"2015, downloaded from https://www.accenture.com/us-en/_acnmedia/Accenture/next-gen/reassembling-industry/pdf/Accenture-Driving-Unconventional-Growth-through-IIoT.pdf

[2] E. Altman, K. Avrachenkov, and A. Gamaev, "Jamming in Wireless Networks: the Case of Several Jammers," Proceedings of the 1st International Conference on Game Theory for Networks, GAMENETS 2009, Istanbul, Turkey, May 13-15, 2009.

[3] E. Bertino, A. A. Jabal, S. B. Calo, D. C. Verma, C. Williams, " The Challenge of Access Control Policies Quality," J. Data and Information Quality 10(2): 6:1-6:6 (2018).

[4] E. Bertino, "Security and Privacy in the IoT," Information Security and Cryptology - 13th International Conference, Inscrypt 2017, Xi'an, China, November 3-5, 2017, Revised Selected Papers. Lecture Notes in Computer Science 10726, Springer 2018.

[5] E. Bertino, G. de Mel, A. Russo, S. B. Calo, D. Verma, "Community-based Self Generation of Policies and Processes for Assets: Concepts and Research Directions," 2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017.

[6] E. Bertino, "Data Security and Privacy in the IoT,", Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016.

[7] E. Bertino, G. Ghinita, A. Kamra, "Access control for databases: concepts and systems," Found. Trends Databases 3(1-2), 1-148 (2011).

[8] L. Bossi, E. Bertino, S.R. Hussain, "A System for Profiling and Monitoring Database Access Patterns by Application Programs for Anomaly Detection," IEEE Trans. Software Eng. 43(5): 415-431 (2017).

[9] H. T.Cheng and W. Zhuang, "Pareto Optimal Resource Management for Wireless Mesh Networks with QoS Assurance: Joint Node Clustering and Subcarrier Allocation," IEEE Transactions on Wireless Communications 8(3): 1573-1583 (2009).

[10] R. Dewri, I. Ray, I. Ray, D. Whitley, " Security Provisioning in Pervasive Environments Using Multi-objective Optimization," Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Malaga, Spain, October 6-8, 2008. Proceedings. Lecture Notes in Computer Science 5283, Springer 2008.

[11] R. Di Pietro, F. Lombardi, "Security for Cloud Computing", Artech House 2015.

[12] M. S. Jalali, J. P. Kaiser, M. Siegel, and S. Madnick, "The Internet of Things Promises New Benefits and Risks: A Systematic Systematic Analysis of Adoption Dynamics of IoT Products. IEEE Security & Privacy 17(2): 39-48 (2019).

[13] J.Habibi, D. Midi, A. Mudgerikar, and E. Bertino, "Mitigating the Internet of Insecure Things," IEEE Internet of Things Journal 4(4): 968-978 (2017).

[14] Z. Han, N. Marina, M. Debbah, A. Hjorungnes, "Physical Layer Security Game: How to Date a Girl with her Boyfriend on the Same Table," 1st International Conference on Game Theory for Networks, GAMENETS 2009, Istanbul, Turkey, May 13-15, 2009.

[15] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, D. Wagner, "Smart Locks: Lessons for Securing Commodity Internet of Things Devices," Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2016, Xi'an, China, May 30 - June 3, 2016.

[16] S. Ma et al. "Two is Better than One? Using Machine Learning and Program Slicing for Flaw Detection in Password-Based Authentication Code," Proceedings of ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I. Lecture Notes in Computer Science 11735, Springer 2019.

[17] S. Ma, F. Thung, D. Lo, C. Sun, R. H. Deng, "VuRLE: Automatic Vulnerability Detection and Repair by Learning from Examples," Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II. Lecture Notes in Computer Science 10493, Springer 2017.

[18] S. McLaughin et al., "The Cybersecurity Landscape in Industrial Control Systems," Proceedings of IEEE 104(5):1039-1057 (2016).

[19] F. Le, J. Ortiz, D. Verma, D. Kandlur, "Policy-Based Identification of IoT Devices' Vendor and Type by DNS Traffic Analysis," Policy-Based Autonomic Data Governance [extended papers from the Second International Workshop on Policy-based Autonomic Data Governance,

PADG@ESORICS 2018, September 6, 2018, Barcelona, Spain]. Lecture Notes in Computer Science 11550, Springer 2019.

[20] D. Midi, S. Sultana, and E. Bertino, "A System for Response and Prevention of Security Incidents in Wireless Sensor Networks," ACM Transactions on Sensor Networks (TOSN) 13(1): 1:1-1:38 (2017).

[21] D. Midi, M. Payer, and E. Bertino, "Memory Safety for Embedded Devices with nesCheck," Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017.

[22] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, " Kalis - A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things," Proceedings of the 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017.

[23] D. Midi and E.Bertino, "Node or Link? Fine-Grained Analysis of Packet-Loss Attacks in Wireless Sensor Networks," ACM Transactions on Sensor Networks (TOSN) 12(2): 8:1-8:30 (2016).

[24] A. Mudgerikar, P. Sharma, E. Bertino, "E-Spion: A System-Level Intrusion Detection System for IoT Devices," Proceedings of the 2019 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2019, Auckland, New Zealand, July 7-12, 2019.

[25] OWASP Internet of Things (IoT) Project, https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project.

[26] V. Paxon, S. Campbell,J. Lee et al. "Bro Intrusion Dectetion Systems," Technical Report, Lawrence Berkeley National Laboratory, 2006.

[27] M. Roesch, "Snort - Lightweight Intrusion Detection for Networks," Proceedings of 13th Systems Administration Conference, LISA'99, Seattle, Washington, USA, November 7-12, 1999.

[28] The Security Ledger. Ids and the iot: Snort creator Marty Roesch on securing the Internet of Things. https://securityledger.com/2014/04/ids-and-the-iot-snort-creator-marty-roesch-on-securing-the-internet-of-things/.

[29] A. Rullo, D. Midi, E. Serra, and E. Bertino, "Pareto Optimal Security Resource Allocation for Internet of Things," ACM Transactions on Privacy and Security (TOPS) 20(4): 15:1-15:30 (2017).

[30] A. Rullo, E. Serra, E. Bertino, and J. Lobo, "Shortfall-Based Optimal Placement of Security Resources for Mobile IoT Scenarios," Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II. Lecture Notes in Computer Science 10493, Springer 2017.

[31] J. Sametinger, J. W. Rozenblit, R. L. Lysecky, P. Ott, "Security Challenges for Medical Devices,"Communications of ACM 58(4): 74-82 (2015).

[32] A. Singla, E. Bertino, "How Deep Learning is Making Information Security More Intelligent," IEEE Security & Privacy Magazine, 2019, in print.

[33] Y. Son, H. Shin, D.Kim, Y.-S. Park, J. Noh, K. Choi, J. Choi, Y. Kim, "Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors," 24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015.

[34] J. Valente, A. S. Cardenas, "Security & Privacy in Smart Toys,"Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, IoT Security and Privacy, CCS, Dallas, TX, USA, November 03, 2017.

[35] Y. Shoshitaishvili, R. Wang, C. Hauser, C. Kruegel, G. Vigna, "Firmalice-Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware," Proceedings of the 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015.

[36] A. A. Yavuz, A. Mudgerikar, A. Singla, I. Papapanagiotou, E. Bertino, "Real-Time Digital Signatures for Time-Critical Networks," IEEE Trans. Information Forensics and Security 12(11): 2627-2639 (2017).

[37] M. Wolf, D. N. Serpanos, "Safety and Security of Cyber-Physical and Internet of Things Systems [Point of View]," Proceedings of the IEEE 105(6): 983-984 (2017).

[38] M. Wolf, D. N. Serpanos, "Safety and Security of Cyber-Physical Systems and Internet of Things Systems," Proceedings of the IEEE 106(1): 9-20 (2018).

[39] J. Won, A. Singla, E. Bertino, G. Bollella, "Decentralized Public Key Infrastructure for Internet-of-Things," 2018 IEEE Military Communications Conference, MILCOM 2018, Los Angeles, CA, USA, October 29-31, 2018.

[40] J. Won, S.-H. Seo, E. Bertino, "Certificateless Cryptographic Protocols for Efficient Drone-Based Smart City Applications," IEEE Access 5: 3721-3749 (2017).