Chapter 4
An Overall Perspective on Establishing End-to-End Security in Enterprise IoT (E-IoT)

Vidya Rao, K. V. Prema, and Shreyas Suresh Rao

4.1 Introduction

IoT is a vast network of networks consisting of physical and virtual interconnected entities. These entities have unique addressing schemes and interact with each other to provide certain customized or generic services. In 2012, the International Telecommunication Union (ITU-T) recommended a standard definition of IoT as "a global infrastructure for the information society, enabling advanced services by interconnecting virtual and physical things based on existing and evolving interoperable information and communication technologies [1]". Technically speaking, IoT has its applications in diverse areas like healthcare, surveillance, transport, security, manufacturing, environmental monitoring, and food processing, and it is integrated with technologies like autonomic networking, decision making, machine-to-machine communication, cloud computing, big data analytics, confidentiality protection, and security [2].

Enterprise Internet of Things (E-IoT) is the next level of sensor technology that connects every physical object to form a vast network of embedded computing devices. These devices are generally made up of tiny components. They have constrained processing capabilities, low memory, and limited power resources. This emerging technology has reduced manual intervention and has increased business efficacy.

Gartner Press released an article in August 2019 showcasing that by 2020, there would be about 5.8 billion IoT endpoints, as compared to 4.8 billion endpoints during 2019. That means there is almost a 21% increase in the addition of new endpoints.

V. Rao (✉) · K. V. Prema
Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India
e-mail: prema.kv@manipal.edu

S. S. Rao
Sahyadri College of Engineering and Management, Mangalore, Karnataka, India

A. Haldorai et al. (eds.), *Business Intelligence for Enterprise Internet of Things*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-44407-5_4

These endpoints are categorized under various use cases like utilities, government buildings, automation, physical security, healthcare providers, manufacturing and natural resources, information and transportation, retail, and wholesale. Among these use cases, utilities have taken a major share of 17% with 1.33 billion endpoints with applications like electric smart grid, smart metering, and smart electricity supply. Apart from this, physical security application surveillances, intruder systems, as well as CCTVs have taken about 0.70 billion endpoints.

These endpoints have generated a total revenue of about $389 billion in countries like North America (NA), Greater China (GC), and Western Europe (WE). Statistics of Gartner's study have shown that about 75% of the revenue would be generated by electronic endpoints in the world, i.e., about $120 billion revenue from NA and $91 billion and $82 billion revenue from GC and WE, respectively, by 2020. It is expected that the two main use cases that shall take a good share in the electronic revenue are connected to consumer cars and networkable printing and photocopying with $71 billion and $38 billion revenue, respectively. Then comes the government indoor and outdoor surveillances that add on to the revenue as the government is considering civilian security as its top priority.

These endpoints are enabled with various sensors like cameras, proximity sensors, temperature sensors, air quality sensors, flow sensors, and many more sensors that are unprotected. The reason is that the agent-based technologies do not protect them from various attacks like distributed denial of service (DDoS), ransomware attack, stealing of sensitive intellectual properties, cryptojacking attacks, etc. [3]. This is a major cause of concern as the data produced by these devices consists of user health information, bank details, passwords, location information, and many more. Hence these devices are subjected to security threats due to (a) malicious or compromised node in the network, (b) defective manufacturing, and (c) presence of an external adversary. There may also be threats to security initiated by nature. These natural threats include earthquakes, floods, fire, and hurricanes that cause severe damage to the computer systems. As it is hard to safeguard against natural calamities, it is advisable to reduce the damage by collecting backup of data through a contingency plan. Similarly, there could be human threats that can be classified under information-level attacks, adversary location attacks, access-level attacks, and host-based attacks. To enable the security of the devices, it is essential to select the hardware components that have the following properties: default authentication capabilities, end-to-end traffic encryption, secure boot loading process, enforcement of digital signatures during firmware update, and transparent transactions.

Also, it has been identified that there are almost 1.1 billion data points created every week, with 2.5 billion GB of data being generated across the world. Likewise, about 500 GB of data is generated by offshore oil rigs and 100 GB of data from oil refineries per week. Also about 10,000 GB of data is generated by jet engines every 30 min. Overall, it is said that about 90% of the world's data has been generated in the last 2 years. Thus, when such a huge amount of data points and data are available on the public network like the Internet, they are susceptible to various attacks. Hence, it is essential to identify the possible security safeguards at the earliest.

With the growth of connected devices under IoT, there is an increase in the potential vulnerability on security, privacy, and governance. Though IoT can make people's life convenient, it might fail to ensure security and privacy of the user data leading to a number of undesirable consequences. For example, in 2015, IoT baby monitors were hacked through which the hackers were able to monitor the live feeds of the baby, change the camera settings, and authorize other users remotely to view and control the baby monitor [4]. During 2017, intruders could over-write the part of Ukraine's power grid that caused the first cyber attack [2]. Even the Internet-connected cars and wearable devices can also become a threat to the user's security and privacy.

In [5] Atmali et al. have analyzed the impact of the above attacks on IoT applications like power management, smart car, and the smart healthcare system. Through their study, they have projected that there is a need for security and privacy considerations at the level of (a) actuators, (b) sensors, (c) RFID tags, and (d) the Internet/network. Attack on actuators in power management applications can lead to financial loss due to excessive power consumption. Similarly, in smart cars, these compromised actuators may control the broken system costing a driver's life. Also, in the healthcare system, these compromised actuators can inject the wrong dosage of medicine to a patient who is remotely monitored by the doctor. Likewise, a compromised sensor can fake the data that may lead to the wrong diagnosis of a patient. At the same time, these compromised nodes can reveal the personal information of the patient or the data related to a user's home through power management system.

Section 4.2 of this chapter explains the various security threats and attacks, followed by elements of security in Sect. 4.3, some of the lightweight existing solutions in Sect. 4.4, threat modeling tools in Sect. 4.5, Kali Linux-based ethical hacking in Sect. 4.5.4, major IoT security practices of E-IoT in Sect. 4.6, and lastly, conclusion in Sect. 4.7.

4.2 Security Threats and Attacks

Devices within the IoT communicate personalized data of many users. This data consists of user health information, bank details, passwords, location information, and many more. These devices are subjected to security threats like (a) malicious node in the network, (b) defective manufacturer, and (c) external adversary [5]. These threats lead to security attacks that can be initiated either by nature or human. The natural threats may include earthquake, floods, fire, and hurricane that cause severe damage to the computer system. Although it is hard to safeguard against natural calamities, it is advisable to reduce the damage by collecting backup of data through contingency plan. Accordingly, the security attacks caused by the humans affect the node privacy [6, 7]. Such attacks can be classified as follows [5, 8]:

1. *Information-level attacks:* All IoT devices are enabled with sensors that record the data from the physical environment and communicate the information over

the Internet. As the Internet is an open domain, attackers can easily tamper the information under following categories [7–9]:

(a) Denial of service (DoS): DoS is an attack over the network component that makes it unavailable for an intended user.
(b) Masquerade: An intruder behaves as an intended user and tries to talk with the network component.
(c) Modification of message: An intruder can alter or delete or fake a message sent by a legitimate user.
(d) Man-in-the-middle (MITM): MITM is a kind of attack wherein a malicious user takes control of the communication channel between two or more endpoints.
(e) Message replay attack: It is a security breach in which the message is stored by malicious node without the knowledge of intended users, and the malicious node transmits an altered message that is forwarded to the receiver.

2. *Adversary location attack:* An intruder can be present in any part of the IoT ecosystem. He can either be within or outside the IoT environment [9]:

(a) Internal attack: An attack caused by the components within the IoT border. It is also called as insider attack where the intruder tries to inject malicious code toward the IoT components.
(b) External attack: An attack caused by an advisory that is located outside the IoT environment in a remote place.

3. *Access-level attack:* Access-level attacks are broadly classified into active and passive attacks [10]. In the passive attack, an attacker can read the packet that is transmitted, but he/she cannot alter the packets like eavesdropping and traffic analysis. On contradictory, in active attack, the attacker sees the data and then alters the content of the data and transmits the altered data back to the network.
4. *Host-based attack:* Many devices in an IoT environment are made up of different manufacturers [10]. These devices are subjected to user compromise attack, software compromise attack, and hardware compromise attack. This is because the manufacturer can hold the devices' information which can be misused by him. Hence the production of such poorly secured goods results in compromising the user privacy. At the same time, any manufacturer can attack his competitors through their devices.

4.2.1 IoT Four-Layered Architecture and Associated Attacks

P. P. Ray [10] has surveyed various domain-based architectures that vary from RFID to healthcare to security to cloud services. But in general, a four-layered design of IoT is considered for different research as in Fig. 4.1. Mainly it comprises of perception layer, network layer, transport layer, and application layer. Each layer has its own properties and protocols. Primarily, the perception layer forms the physical

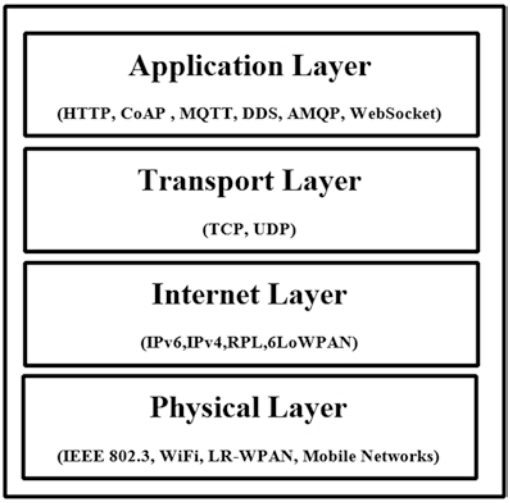

Fig. 4.1 Generic four-layered IoT architecture

layer of the IoT ecosystem. It deals with sensors, devices, machines, actuators, and movements of unprocessed raw data. In this layer, the data transmission medium used is copper wire, coaxial cable, or radio wave. They have protocols like IEEE 802.3, Wi-Fi, LR-WPAN, 2G, 3G, 4G, and LTE networks [11].

Next is the Internet layer, which is also called the network layer. The main job of this layer is to provide host identification and packet routing. IETF has proposed many routing protocols that are suitable for low-powered device networks. Some of the protocols are IPv6, IPv4, RPL, 6LoWPAN, multipath RPL (MRPL) [12], energy-efficient probabilistic routing protocol (EEPR) [13], congestion avoidance multipath routing protocol (CA-RPL) [14], movement-aided energy balance (MABE) [15], least path interface beaconing protocol (LIBP) [16], and cognitive machine-to-machine RPL (CoRPL) [17].

Then comes the transport layer which is considered for end-to-end message transfer. The transmission can be either connection-oriented or connectionless with protocols like transmission control protocol (TCP) and user datagram protocol (UDP), respectively. This layer involves various processes like segmentation and reassembly of packets, congestion control, error control, and flow control.

Lastly, the application layer interfaces with all the lower layers by establishing a secure connection between the devices and servers. It uses standard port 80 and port 22 for most of HTTP and SSH protocols, respectively. Some of the protocols standardized by IETF are constrained application protocol (CoAP), message queuing telemetry transport protocol (MQTTP), extensible message and presence protocol (XMPP), data distribution services (DSS), and advanced message queuing protocol (AMQP) [11].

Likewise there are various attacks based on layers of IoT as shown in Fig. 4.2 [3, 18]. Sensing/perception layer is generally made up of sensors, RFIDs, NFCs, ZigBee, Bluetooth, and other intelligent hardware devices. These devices are exposed to more external attacks like node compromise attack, fake node injection,

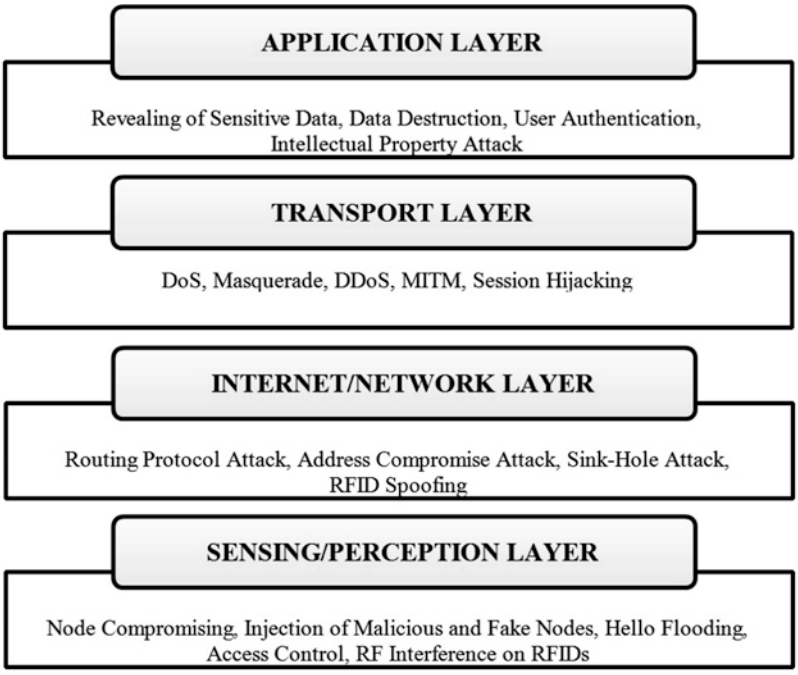

Fig. 4.2 Attacks based on architecture

access control, and RF interference on RFIDs. The second layer is the Internet layer and is subjected to attacks like address compromise attacks, routing information attack, RFID spoofing, and sinkhole attack. The next layer is the transport layer that experiences attacks like denial of service (DoS), masquerade, distributed DoS (DDoS), man-in-the-middle (MITM) attack, and session hijacking. And finally, the application layer experiences attacks like phishing attack, viruses, worms, malicious scripting, revealing of sensitive data, user authentication attacks, software vulnerability, and stealing of intellectual property.

4.2.2 Attacks Based on Phases of IoT

IoT can also be defined as an interconnection of "factual and virtual" objects placed across the globe that are attracting the attention of both "makers and hackers." IoT can be divided into five different phases as mentioned in [19] by Jeyenthi as shown in Fig. 4.3. The first phase is termed as the data collecting phase: primary interface between physical environment and sensors. There can be either static objects like body sensors or RFIDs or dynamic objects like sensors and chips on vehicles. The second phase is the storage phase: as many IoT devices are having low self-storage capability, IoT provides a server or cloud-based storage. Next is the intelligent pro-

PHASES	ATTACKS
Data Perception	Data Leakage, Data Authentication, Data Loss
Storage	Data Availability, Modification of Message, DOS, Attack on Integrity, Data Fabrication
Processing	Attack on Authentication
Transmission	Channel Security Attack, Session Hijacking, Routing Protocol Attack, Flooding
End-to-end Delivery	Man Induced or Machine Failure, Maker or Hacker

Fig. 4.3 Phases of IoT and their possible attacks

cessing phase: it is where the analysis of stored data and later appropriate services are provided to the users. IoT devices can be queried and controlled remotely using the results obtained after processing of data. The fourth phase is data transmission: it deals with processing of data communication between all of the above phases. Last is the delivery phase: it is where the activity of delivering the processed data to all the objects in time without being altered or hacked is performed.

Among the five phases, the data perception phase is subjected to more attacks like data leakage, data authentication, and data loss as the devices are easily available to users and hackers. Similarly, in storage phase, we can see attack on availability, modification of message, denial of service (DoS) attack, attack on integrity, and data fabrication. Attacks on authentication are seen at the processing phase, and channel security attack, session hijacking, routing protocol attack, and flooding are seen at the transmission phase. Lastly, at the delivery phase, man- and machine-made attacks are found as shown in Fig. 4.3.

4.3 Elements of Security

To ensure the IoT security, there are four elements of security [18]. They are device authentication, secure connections, secure data storage, and lastly, secure code execution. The device authentication grants the access privilege of the devices to the legitimate users. Secure connection enables the protection of the data that is travelling across the network (data in motion). Secure storage provides protection for data in rest using various lightweight encryption schemes. And lastly, the secure code execution serves the intended host machines to use the data and process it in a secure manner as in Fig. 4.4.

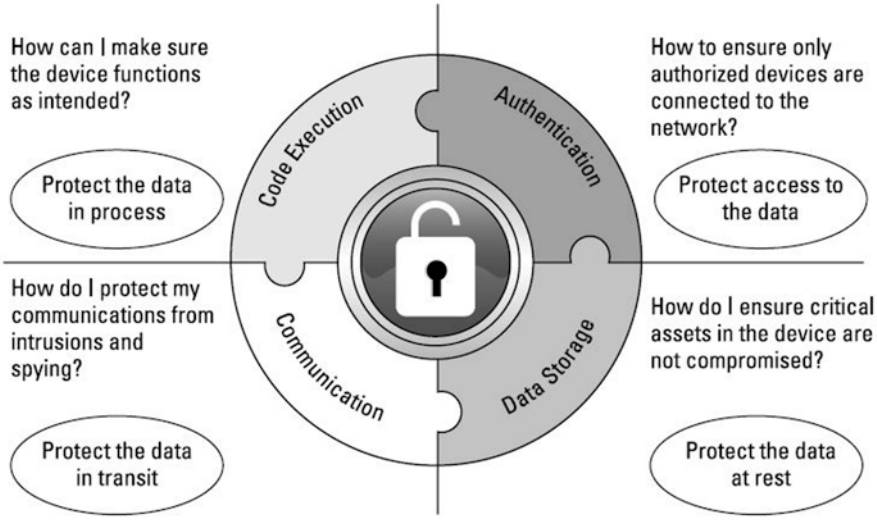

Fig. 4.4 Elements of IoT security [18]

As these poorly secured IoT devices can serve as means of entry point for cyber attackers by allowing various malicious individuals to re-program a device and cause malfunctioning, it becomes essential to provide security and privacy at the devices level. In order to develop a safer IoT solution, it is required to consider three major security requirements: (i) confidentiality, (ii) integrity, and (iii) authentication [20].

- Confidentiality means keeping information secret from the unauthorized user. For example, when transmitting certain sensitive data like location of military camp to the base station, it must be forwarded in secrecy to avoid intruders to understand the information that is being transmitted.
- Data integrity ensures that the messages transmitted are reached at the destination unaltered. Data integrity certifies the user that it has never been altered or corrupted by protecting the data over a communication channel.
- Authentication is a process of determining whether the data is transmitted by legitimate users or not. The user needs to identify the peer nodes that they need to communicate.

4.4 Lightweight Secure Measures for IOT

Elliptic curve cryptography (ECC) was introduced in early 1985 by Neal Koblitz and Victor Miller [21]. They stated that the hardness of ECC security depends on the discrete logarithmic problem defined on the elliptic curve. Later, Gura et al. [22] experimented ECC and RSA on an 8-bit CPU to compare their performance and found that the use of ECC for a lower-bit processor provides the same level of secu-

rity as that of RSA. Later during 2013, Wenger [23] developed an ECC-based access control scheme over a prime field on 16-bit MSP430 micro-controller whereby the results confirmed the feasibility of ECC for resource-constrained devices.

Basically, ECCs are often implemented by using a static public elliptic curve that is shared among all the users in the network. In [24] the recommended elliptic curve domain parameters are provided for the Weierstrass curve equation $y2 = x3 + ax + b$ that is accepted by various researchers [25]. Liu et al. [26] have proposed software and hardware architecture for resource-constrained embedded devices. Their work has shown the feasibility of ECC on the embedded system. But the use of a fixed elliptic curve can be challenged on intensive cryptanalysis. Wang et al. [27] made a study on using a fixed prime field to build a crypto-system for applications developed for different processors varying from 8 bits to 256 bits.

A lightweight multi-message and multi-receiver heterogeneous-based signcryption is proposed by Rahaman et al. [28]. They have used the hybrid elliptic curve to generate signatures. The work is evaluated for various attacks like replay attack, forward secrecy attack, and unforgeability using the AVISPA simulator tool. For the heterogeneous environment, the attackers are inclined to impersonate legitimate users. To solve such an issue, Jingwei Liu et al. [29] have proposed a novel authentication scheme. They have provided a lightweight anonymous authentication and key agreement scheme as proposed. Their scheme could toggle between the public key infrastructure (PKI) and certificates analysis. Their method showed resistance against replay and DoS attacks.

The combination of cloud-based services with IoT has raised the issue of limitation regarding low latency and high mobility. To address such issues, Haldorai et al. [30] have proposed the authentication and key agreement scheme for fog-based IoT for the healthcare application. By using bilinear key agreement protocol, they have proposed a protocol that showed resistance against MITM, replay attack, known-session key attack, and intractability.

Recently, based on card shuffling logic, a data confidentiality algorithm is designed using ECC, proposed by Khan [31]. The use of random card shuffling has shown double encryption and increased the security of the algorithm. The algorithm can encrypt or decrypt any type of ASCII values. As the algorithm uses ECC, it is suitable for resource-constrained devices. Li et al. [32] proposed a lightweight mutual authentication protocol using public-key encryption schemes for smart city applications. Their simulated work has shown a balance among ciphertext size, usability, and efficiency. The generation of online and offline signatures created overhead on the device storage. Diro et al. [33] have used ECC to provide lightweight encryption for fog-based IoT applications. They have shown better efficiency regarding runtime, throughput, and ciphertext expansion. But they could only handle a smaller data size.

An OTP-based end-to-end authentication scheme was proposed by Shivraj et al. [34]. Their scheme used Lamport's OTP scheme with ECC-based authentication algorithm. Even though the scheme performed better than existing OTP-based signature schemes, they could not justify the implementation on a real-time scenario. A security framework for IoT and cloud computing is proposed by Daisy Premila

et al. [20]. They used ECC-based message encryption and multi-factor authentication to ensure confidentiality, integrity, privacy, and authentication. They have concluded that the use of ECC-based security measures is better than RSA to eliminate the ambiguity and enhance security. But the research to collaborate IoT and cloud computing needs to depend on infrastructure.

During 2018, to address the usage of the static curve in ECC, Jia Wang et al. [35] proposed a dynamic elliptic curve-based Internet of Vehicles (IoV) network. Their work showed good computational efficiency and security for a smaller key size. But storing the elliptic curves as a plain text in embedded systems would lead to security concern. To address the data integrity issue of Java card-based application, Gayoso et al. [36] initiated the use of ECC-based encryption algorithm called an elliptic curve integrated encryption scheme (ECIES) and concluded that ECIES-based encryption is the best among encryption schemes for resource-constrained devices.

4.5 Threat Modeling for IOT Security

Threat modeling (TM), whose lifecycle is depicted in Fig. 4.6, is a process of identifying the potential threats, enumerating and prioritizing the threats, and providing countermeasures to mitigate the threats. TM can be applied to any platform of a working process like software, application, networks, IoT devices, or business processes. Shostack [37] has summarized the reasons to incorporate a threat model in SDL which are (i) to find the bugs at the earliest, (ii) understand the security requirements, and (iii) engineer and deliver a better product. Basically, TM includes components like *target-of-evaluation* (*ToE*) (a design or model of what type of platform needs to be analyzed), a list of *assumptions* that can be threats on ToE, a list of *potential threats* on ToE, *possible countermeasures* toward the identified threats, and *verification of success* (VoS) that validates the threat model.

Before modeling a threat, there are four questions that need to be answered, which are as follows:

1. What are we building? A detailed data flow diagram (DFD) is designed by specifying various roles and responsibilities of each participant.
2. What can go wrong? The various possible threats are analyzed using methods available in STRIDE, PASTA, STRIKE, or VAST.
3. What are we going to do about that? Potential mitigation strategies against the threats are framed.
4. Did we do a good job? Once the mitigation is applied, the system is validated for the stability and security against the threats.

4.5.1 Microsoft Security Development Lifecycle

Microsoft SDL was introduced during 2008 to ensure security and privacy considerations throughout all the phases of the development process. This helped developers to build highly secure software, address security compliance requirements, and reduce development cost. The core of Microsoft SDL is threat modeling. Threat modeling helps in shaping the application design and meeting the security objectives of the company by reducing the risk severity. The five major steps of threat modeling involve (Fig. 4.5) the following:

1. Defining security requirements: To understand the ecosystem of the device, i.e., analysis of the ToE by framing various use cases. In this process the external and internal assets are identified.
2. Creating an application diagram: Here a detailed data flow diagram of the proposed ToE is framed with appropriate trust boundaries and security requirements for each participant.
3. Identifying the threats: Microsoft TMT follows STRIDE-based threat modeling where the threats are identified. Potential adversaries are identified under four categories called remote software attacker, network attacker, malicious insider attacker, and advance hardware attacker.
4. Mitigating the threats: For the threat identified, relevant countermeasures are established.
5. Validating that threats have been mitigated: Finally, the verification of the threat model against the mitigation is performed to check the stability of the proposed system.

4.5.2 STRIDE Framework Methodology

It is important to develop a secure design for any software application or system. Failing to do so may cost about 30 times higher than estimated cost [38]. Hence threat modeling plays a vital role in the software development lifecycle. Among various threat modeling methods like STRIDE, PASTA, VAST, and STRIKE,

Fig. 4.5 Microsoft Security Development Lifecycle (SDL) using TMT

Table 4.1 STRIDE threat model with associated security properties

Threats	Security property	Definition
Spoofing	Authentication	Unauthorized access Using another user's identity
Tampering	Integrity	Malicious modification Unauthorized information changes
Repudiation	Non-repudiation	Denying to perform action
Information disclosure	Confidentiality	Unprivileged user gains access and compromises the system
Denial of service	Availability	Denying services to valid users Threats to system availability and reliability
Elevation of privilege	Authorization	Exposure of information to individuals not supposed to access

STRIDE has taken a major share among the industrial development processes [39, 40]. STRIDE is developed by Microsoft as a part of their Security Development Lifecycle. STRIDE is an acronym for spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege [41]. The security properties and attack types associated with STRIDE are summarized in Table 4.1 [38].

4.5.3 Overview of Threat Modeling Tool (TMT)

Microsoft TMT is used to provide assistance in analyzing the design of a system or an application in order to check for security risks and provide solution for the threat found. Figure 4.4 displays the initial page of TMT when launched. This page has two partitions; the top part is used to create the threat model of the user's choice using the templates provided by the Microsoft, while the bottom part helps the user to customize his own template on the default Microsoft Security Development Lifecycle (SDL) template as in Fig. 4.6.

4.5.4 Kali Linux-Based Ethical Hacking

Kali Linux was developed by Mati Aharoni and Devon Kearns of Offensive Security and was mainly suitable for digital forensic and penetration testing under ethical hacking [42]. Kali Linux has approximately 300 hacking tools that are broadly categorized under information gathering, vulnerability analysis, wireless attacks, web application, exploitation tools, forensic tools, sniffing and spoofing tools, password attacks, maintaining access, reverse engineering, and hardware hacking tools. Among these, the most commonly used tools are Metasploit framework, dsniff, tcpdump, Nmap, Wireshark, Aircrack-ng, Armitage, Burp Suite, BeEF, and so on [42].

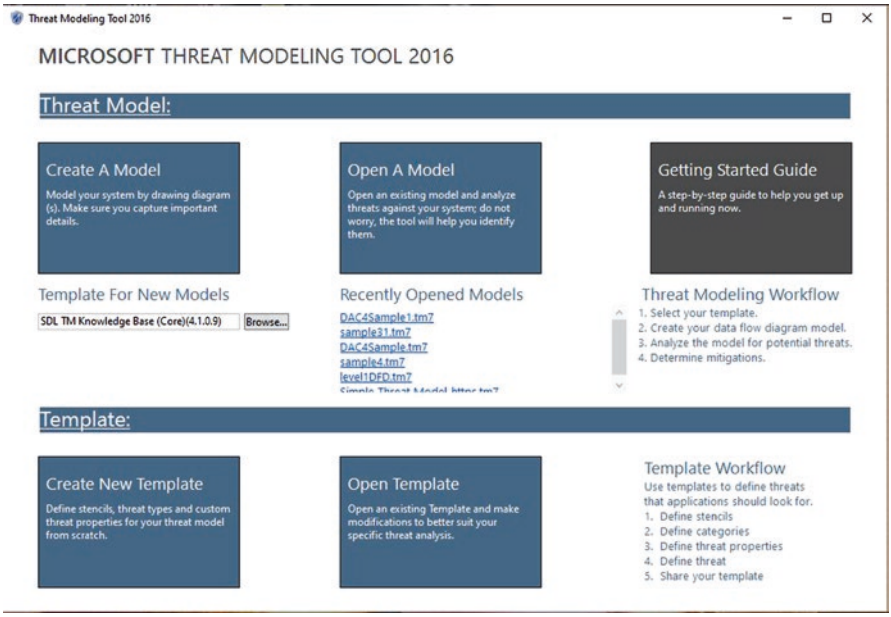

Fig. 4.6 Microsoft TMT initial screen

Featuring the rapid growth of smart cities, Barghuthi et al. [43] have made a study of how the increase in the population of smart cities shall add to an increase in the security breach and damage to businesses by 2050. Thus, they have proposed Kali Linux-based vulnerability assessment and penetration testing solution using low-cost Raspberry Pi 3 devices. Through their results, it has been concluded that Raspberry Pi 3 can be used as a machine to check the vulnerability similar to any traditional PC or laptop-based Kali Linux machine.

To replace the expensive and resource-intensive devices used for industrial vulnerability and assessment tests, Hu et al. [44] proposed an automated vulnerability assessment using OpenVAS and Raspberry Pi 3 device. They have detailed methods for analyzing the vulnerability assessment of distributed architecture. They made the study on variables like CPU temperature, CPU usage, and CPU memory of the device at the time of vulnerability assessment.

Visoottiviseth et al. [45] developed a GUI-based penetration testing tool called PENTOS used for IoT devices. PENTOS runs on Kali Linux and is specifically designed for the ethical hacking of wireless communication like Wi-Fi and Bluetooth. PENTOS enables the analysis of password attack, web attack, and wireless attack that ensure to gain access privilege of the various algorithms. They also have explained the Open Web Application Security Project (OWASP) specified ten vulnerabilities of IoT applications.

Finally, they have given the recommendations for the secure deployment of the IoT environment. Denis et al. [46] performed various penetration tests using tools available on Kali Linux. They were able to set up a private network and generate

attack reports and visualize the reports using Kali Linux tools. The attacks they performed were hacking phones, MITM attack, smartphone penetration testing, spying, hacking phones' Bluetooth, and hacking WPA-protected access, and then they hacked the remote PC using IP and open ports.

Liang et al. [47] experimented on different methods of doing DoS attack using Raspberry Pi-based Kali Linux. They have provided an attack framework and compared various DoS attacks on their framework. They have used Hping3 with random IP, SYN flood with spoofed IP, and TCP connection flood tools. The comparison was made under the parameters like CPU utilization, memory utility, time for the success of an attack, and packet loss rate. Ryan Murray [48] has proposed a forward-looking approach for a secure eHealth solution called HealthShare. It could share data among various organizations that were hosting the patient's data over the cloud. Detailed steps as to conduction of MITM and DoS attack using tools like Ettercap, Pexpect, manual SET, threads using the timer and Nmap timer, and Scapy have also been provided.

4.6 Major E-IOT Security Practices

As E-IoT is deployed on a larger scale with heterogeneous business applications, the cybersecurity space has obtained an intense research spectrum. Some of the important security practices that should be followed by enterprise IoT are explained below.

(a) *Understand your endpoints*: Every endpoint of the business network is assembled by various manufacturers using different open-source operating systems. These devices are potential entry points for cybercriminals. Thereby, it is essential to deploy devices in a tamper-proof environment using secure hardware and software resources.
(b) *Track and manage the endpoints*: Business enterprise poses the responsibility of constant check on the devices that are deployed under their network and should be updated with frequent firmware and security patches. As it is infeasible to monitor each device physically, Earl Perkins of Gartner Solutions has recommended "rolling out an asset discovery, tracking, and management strategy" to be implemented before the IoT project begins.
(c) *Change the default passwords and other credentials*: The manufacturers set their devices with a common default password, which has to be updated by the enterprise officials frequently. This is because, most of the time, hackers are well aware of default passwords and sneak into your network by brute force attacks.
(d) *Execute risk-driven strategies*: IoT projects need to be analyzed for risk possibility using various threat modeling tools. Such tools help to identify the risks in the network and guide the network administrator to take corrective actions. Also, performing regular pen-testing at the hardware and software levels shall ensure the attack resistivity of the network.

(e) *Consideration of the latest encryption protocols*: Business enterprises should encrypt the data passing from and to their network using the updated and latest encryption schemes. If in case a single device is accessed by multiple users, then the focus should be on user authentication, identity-level control, and providing data integrity.

4.7 Summary

IoT is a rapidly growing network that has its major contribution in making the business enterprise smarter. E-IoT could connect to a diverse domain of applications and devices across the globe thus leading to various levels of attacks and threats. Various levels of hardware and software issues are studied with possible lightweight solutions. A generalized layer of security architecture is discussed, followed by a brief description on threat modeling tool. In addition, Kali Linux-based pen-testing on a real-time E-IoT is also studied. Finally, the major E-IoT practices are generalized that help future researchers to concentrate on the specific issues in E-IoT.

References

1. Telecommunication Standardization Sector of ITU. (2012). *Series Y. Global information infrastructure, internet protocol aspects and next-generation networks* (pp. 1–6). Geneva: ITU.
2. Vermesan, O., & Friess, P. (2014). *Internet of things-from research and innovation to market deployment* (Vol. 29). Aalborg: River Publishers.
3. Ahemd, M. M., Shah, M. A., & Wahid, A. (2017). IoT security: A layered approach for attacks and defenses. In *2017 international conference on Communication Technologies (ComTech)*, IEEE, pp. 104–110.
4. Khan, R., Maynard, P., McLaughlin, K., Laverty, D., & Sezer, S. (2016). Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. In *ICS-CSR*.
5. Atamli, A. W., & Martin, A. (2014). Threat-based security analysis for the internet of things. In *2014 international workshop on Secure Internet of Things (SIoT)*, IEEE, pp. 35–43.
6. Stankovic, J. A. (2014). Research directions for the internet-of-things. *IEEE Internet of Things Journal, 1*(1), 3–9.
7. AbdAllah, E. G., Hassanein, H. S., & Zulkernine, M. (2015). A survey of security attacks in information-centric networking. *IEEE Communication Surveys and Tutorials, 17*(3), 1441–1454.
8. Abomhara, M., & Kien, G. M. (2015). Cyber security and the internet-of-things: Vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security, 4*, 65–88.
9. Nawir, M., Amir, A., Yaakob, N., & Lynn, O. B. (2016). Internet of things (IOT): Taxonomy of security attacks. In *3rd International Conference* on *Electronic Design (ICED)*, IEEE, pp. 321–326.
10. Ray, P. P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences, 30*(3), 291–319.

11. Arulmurugan, R., Sabarmathi, K. R., & Anandakumar, H. (2017). Classification of sentence level sentiment analysis using cloud machine learning techniques. *Cluster Computing, 22*(S1), 1199–1209.
12. Quynh, T. N., LeManh, N., & Nguyen, K. N. (2015). Multipath RPL protocols for greenhouse environment monitoring system based on internet of things. In *2015 12th international conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE.
13. Park, S. -H., Cho, S., & Lee, J. -R. (2014). Energy-efficient probabilistic routing algorithm for internet of things. *Journal of Applied Mathematics*, Hindawi Publishing Corporation, Vol. 2014, Article ID 213106, 7 pages.
14. Haldorai, A., & Ramu, A. (2019). *Cognitive social mining applications in data analytics and forensics* (Advances in social networking and online communities). Hershey: IGI Global. https://doi.org/10.4018/978-1-5225-7522-1.
15. Haoru, S., Wang, Z., & An, S. (2013). MAEB: Routing protocol for IoT healthcare. *Advances in Internet of Things, 3*, 8–15. Scientific Research.
16. Ngqakaza, L., & Bagula, A. (2014, May 26–28). Least Path Interference Beaconing Protocol (LIBP): A frugal routing protocol for the internet-of-things. In *12th international conference proceedings, Wired/Wireless Internet Communications- WWIC2014*, Paris, France, pp. 148–161.
17. Aijaz, A., & Aghvami, A. H. (2015). Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective. *IEEE Internet of Things Journal, 2*(2), 103–112.
18. Miller, L., & Johnson, C. A. (Eds.). (2016). *IoT security for dummies*. Chichester: Wiley.
19. Anandakumar, H., & Umamaheswari, K. (2017). Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers. *Cluster Computing, 20*(2), 1505–1515.
20. Bai, T. D. P., Rabara, S. A., & Jerald, A. V. (2015). Elliptic curve cryptography based security framework for Internet of Things and cloud computing. In *Conference on recent advances on computer engineering by WSEAS*, pp. 65–73.
21. Miller, V. S. (1985). Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques* (pp. 417–426). New York: Springer.
22. Gura, N., Patel, A., Wander, A., Eberle, H., & Shantz, S. C. (2004). Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *International workshop on cryptographic hardware and embedded systems* (pp. 119–132). Berlin/Heidelberg: Springer.
23. Wenger, E. (2013). Hardware architectures for MSP430-based wireless sensor nodes performing elliptic curve cryptography. In *International conference on applied cryptography and network security* (pp. 290–306). Berlin: Springer.
24. SEC, S. (2000). *SEC 2: Recommended elliptic curve domain parameters*. Standards for Efficient Cryptography Group, Certicom Corp.
25. Silverman, J. H. (2009). *The arithmetic of elliptic curves* (Vol. 106). New York: Springer.
26. Liu, A., & Ning, P. (2008). TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks*, IEEE Computer Society, pp. 245–256.
27. Wang, J., & Cheng, L. M. (2017). Dynamic scalable ECC scheme and its application to encryption workflow design. In *Proceedings of the international conference on Security and Management (SAM)*, pp. 261–262.
28. Rahman, A. U., Ullah, I., Naeem, M., Anwar, R., Noor-ul-Amin, Khattak, H., & Ullah, S. (2018). A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve. *International Journal of Advanced Computer Science and Applications, 9*(5), 160–167.
29. Liu, J., Ren, A., Zhang, L., Sun, R., Du, X., & Guizani, M. A. (2019). *Novel secure authentication scheme for heterogeneous internet of thing*. arXiv preprint arXiv:1902.03562.

30. Haldorai, A., Ramu, A., & Murugan, S. (2019). Smart sensor networking and green technologies in urban areas. In *Computing and communication systems in urban development* (pp. 205–224). Cham: Springer. https://doi.org/10.1007/978-3-030-26013-2_10.
31. Khan, M. A. (2018). Multidisciplinary Journal of European University of Bangladesh. *Cell, 1713*(006814), 01914–098494.
32. Li, N., Liu, D., & Nepal, S. (2017). Lightweight mutual authentication for IoT and its applications. *IEEE Transactions on Sustainable Computing, 2*(4), 359–370.
33. Diro, A. A., Chilamkurti, N., & Nam, Y. (2018). Analysis of lightweight encryption scheme for fog-to-things communication. *IEEE Access, 6*, 26,820–26,830.
34. Shivraj, V., Rajan, M., Singh, M., & Balamuralidhar, P. (2015). One time password authentication scheme based on elliptic curves for Internet-of-Things (IoT). In *5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, IEEE, pp. 1–6.
35. Wang, J., Li, J., Wang, H., Zhang, L. Y., Cheng, L. M., & Lin, Q. (2018). Dynamic scalable elliptic curve cryptographic scheme and its application to in-vehicle security. *IEEE Internet of Things Journal, 6*(4), 5892–5901.
36. Gayoso Martínez, V., Hernández Álvarez, F., Hernández Encinas, L., & Sánchez, C. (2011). Analysis of ECIES and other cryptosystems based on elliptic curves. *Journal of Information Assurance and Security, 6*(4), 285–293.
37. Shostack, A. (2014). *Threat modeling: Designing for security*. Indianapolis: Wiley.
38. Verheyden, L. (2018). *Effectiveness of threat modelling tools*. Master thesis.
39. Bodeau, D., McCollum, C., & Fox, D. (2018). *Cyber threat modeling: Survey, assessment, and representative framework*. HSSEDI, The Mitre Corporation.
40. Meghanathan, N., Boumerdassi, S., Chaki, N., & Nagamalai, D. (2010, July 23–25). Recent trends in network security and applications. In *Third international conference, CNSA-2010, Chennai, India, 2010 proceedings*, Vol. 89. Springer.
41. Khan, R., McLaughlin, K., Laverty, D., & Sezer, S. (2017). Stride-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies conference Europe (ISGT-Europe)*, IEEE, pp. 1–6.
42. K. Linux. (2016). *Kali linux tools listing*. https://www.kali.org
43. Al Barghuthi, N. B., Saleh, M., Alsuwaidi, S., & Alhammadi, S. (2017). Evaluation of portable penetration testing on smart cities applications using raspberry pi III. In *2017 fourth HCT Information Technology Trends (ITT)*, IEEE, pp. 67–72.
44. Hu, Y., Sulek, D., Carella, A., Cox, J., Frame, A., & Cipriano, K. (2016). Employing miniaturized computers for distributed vulnerability assessment. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, IEEE, pp. 57–61.
45. Visoottiviseth, V., Akarasiriwong, P., Chaiyasart, S., & Chotivatunyu, S. (2017). PENTOS: penetration testing tool for internet of thing devices. In *TENCON 2017–2017. IEEE region 10 conference*, IEEE, pp. 2279–2284.
46. Denis, M., Zena, C., & Hayajneh, T. (2016). Penetration testing: Concepts, attack methods, and defense strategies. In *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT),* IEEE, pp. 1–6.
47. Liang, L., Zheng, K., Sheng, Q., & Huang, X. (2016). A denial of service attack method for an IOT system. In *2016 8th international conference on Information Technology in Medicine and Education (ITME)*, IEEE, pp. 360–364.
48. Murray, R. (2017). *A raspberry pi attacking guide*. Birmingham: Packt Publishing.