European Economic Letters
ISSN 2323-5233
Vol 14, Issue 2 (2024)
http://eelet.org.uk

TO THE STUDYOF THE SECURITY OF SMART CITY APPLICATIONS BASED ON ONEM2M

[1]Veershetty Halembure
[1]Research Scholar, Department of Computer Science, Mansarovar Global University, Billkisganj, Sehore, Madhya Pradesh-466001
[2]Dr. Manisha Yadav
[2]Research Guide, Department of Computer Science, Mansarovar Global University, Billkisganj, Sehore, Madhya Pradesh-466001

ABSTRACT

The importance of the oneM2M standard in smart cities and its provisions in meeting these needs were the primary foci of this research. The IIIT-H smart city deployment employed the OM2M platform, which is an implementation of oneM2M, and experiments were carried out on it. The security of smart city applications based on oneM2M is laid forth in the guidelines. Put simply, the Internet of Things (IoT) is a network of physical and digital objects, services, and infrastructure that allows them to communicate with one another and share data in real time, regardless of physical location or user authorization. Articles may be remotely managed and detected by IoT using the current system foundation. Improved efficiency, accuracy, and financial gain are all possible outcomes of the Internet of Things, which allows for the remote detection and control of items over an already-established web network.

Keywords: IoT, Business, community, graphically, efficiency, accuracy.

INTRODUCTION

The oneM2M offers encouraging technological requirements for a safe and interoperable Internet of Things/machine-to-machine system. Possible dangers to smart city deployments based on the oneM2M standard are the subject of this chapter. In addition, we provide the eclipse OM2M setups for the oneM2M open-source implementation's baseline security. Practical trials of IIIT-H smart city deployment based on object-to-machine communication (OM2M) in India provide the basis for the suggestions made here. Passive eavesdropping, replay assaults, credential brute force, denial of service, and policy analysis are all part of the testing.

Heterogeneity, interoperability, scalability, connection, mobility, and security are some of the criteria for an Internet of Things (IoT) smart city. The oneM2M project suggests a horizontal layer of common middleware technology that addresses these needs by standardising architecture, API standards, security solutions, and interoperability for M2M/IoT technologies. Mobius, OASIS, CCSP, and eclipse OM2M are just a few of the many popular oneM2M implementations. Eclipse OM2M is a smartM2M and oneM2M implementation. It is part of the Eclipse technology project and is open-source.

The OM2M service platform is compatible with the oneM2M and ETSI M2M standards and is open-source. Service capability layers (SCL) are the backbone of OM2M, an Eclipse IoT working group component that allows for extensive functionality expansion via plugins. OM2M's RESTful APIs are available for all of the platform's services, and the design is based on the modular OSGi framework. Interoperability with older devices, reuse of administration procedures for distant devices, and support for numerous communication protocols (HTTP, HTTPS, COAP, MQTT) are all made possible by this. OM2M may be extended with custom plugins. To provide an extra degree of security, you may enable the Jetty plugin, for instance, which provides HTTPS. At the same token, the MongoDB database may be accessed by activating the home persistence MongoDB plugin.

LITERATURE REVIEW

Haitham Samih et.al (2019) Government and private sector organisations alike have launched Smart City programs in response to the mounting demand for more effective city administration. These programs use ICT to address a wide range of problems, including waste management, and to identify and implement long-term solutions. A number of scholars have sought to pin down what exactly smart cities are, what makes them tick, and what obstacles stand in the way of their construction. This brief piece also explains the connection between smart cities and the Internet of Things, which is a constantly evolving concept.

Gaurav Chaudhary et.al (2020) In a smart city, the development of a specific area with the goal of integrating the city's characteristics into smart housing and parking, efficient energy management, waste and power management, traffic control, and other community services is the basic idea. In most cases, the Internet of Things (IoT) connects a wide variety of gadgets in the smart city. This article will introduce you to the fundamental notion and broad concepts that have been adopted by smart cities throughout the globe in recent times. The Internet of Things (IoT) comprises the basis of most of the ideas used in this kind of development. When it comes to managing and developing in these areas, the Internet of Things allows for smarter and faster administration of the resources and equipment employed. In recent times, there have been several obstacles to the widespread use of the Internet of Things (IoT) in smart city operations. This article uses the Internet of Things to show the procedures and strategies for creating a smart city.

Robert R. Harmon et.al (2015) When it comes to innovative services made possible by information technology, the smart city idea is hard to beat. It paints a picture of a city where service providers collaborate with residents using digital means to build better city systems and organisations that enhance residents' quality of life. The creation of smart cities relies on the new paradigm of the Internet of Things (IoT). For value creation to occur, there must be an integrated cloud-oriented architecture comprising software, sensors, human interfaces, networks, and data analytics. The future of smart city development will be heavily reliant on Internet of Things (IoT) smart-connected items and the services they provide. In this article, we'll take a look at the idea of smart cities and provide a model for developing strategies that include Internet of Things technology into smart city initiatives.

Chai K. Toh (2020) The concept of "smart cities" is gaining traction across the world, with some governments allocating substantial funds to this cause. Smart cities have grown throughout time, starting with previous work on the digital city and progressing to many names such as omnipresent city, green city, linked city, sustainable city, eco-city, etc., thus this growth did not happen suddenly. The emergence of lightning-fast 5G wireless connection, lightning-fast GPU multi-core computers, big data, cloud computing, AI, and data analytics are all hallmarks of the modern era. The creation and implementation of "smart cities" have been aided by several of these emerging technologies. The authors of this paper clarify our meaning of "securing smart cities" and lay out a plan for implementing that plan in smart city outlines. They talk about how current security measures like firewalls, encryption, access control, and authentication can protect a smart city. We detail the potential harmful assaults on a smart city and the repercussions they may have, as well as the security of data, the internet, water supply, energy supply, the city brain, and other vital city functions. Last but not least, they go over some smart city security best practices.

Moez Krichen et.al (2020) In this ongoing project, we want to evaluate the safety features of the Internet of Things as they pertain to smart cities. To achieve this, we employ a model-based approach, which includes the following steps: first, creating a model of the system under study using the appropriate formalism; second, using the model to derive test suites; third, applying coverage criteria to choose appropriate tests; fourth, running the tests; and lastly, collecting and analysing the results to find and fix errors. The chosen formalisation stems from the extended timed automata model that incorporates inputs and outputs. We provide etioco, an extended timed input-output conformity relation, as a means of verifying conformance. We provide a cloud-based framework for test execution.

SECURITY FOR ONEM2M BASED SMART CITY NETWORK

The topic of security in IoT and M2M systems has received a lot of attention. This study concludes with work on mitigation tactics for new applications by surveying the security needs, vulnerabilities, and attack vectors of mission-critical Internet of Things (IoT) applications. Integrating blockchain with IoT to improve security is also included in the paper. Issues with the Internet of Things (IoT), potential dangers, attack scenarios, and ways to lessen their impact are the primary topics of this study, which seeks stakeholder input in order to provide solutions. Privacy and security recommended practices for both users and makers of Internet of Things devices are addressed in additional ETSI standards. In the context of Internet of Things (IoT) smart cities in particular, it offers a thorough evaluation of the risks and vulnerabilities faced by AirIoT, a smart city system for monitoring air quality. The STRIDE modelling framework is used to simulate the dangers, and then solutions are proposed along with their corresponding trade-offs. When discussing security for the Internet of Things (IoT), the oneM2M technical standards document (TS-0003) discusses many topics, including dynamic authorisation, privacy protection, application/device impersonation prevention, and access control policies (ACP). Mbed OS, the operating system stack, is secured in the article. Realising the oneM2M-specified Security Association Establishment Framework (SAEF) is an integral part of their implementation to provide secure end-to-end communications for Internet of Things (IoT) devices. The authors demonstrate secure MQTT binding on the same MbedOS, in accordance with the oneM2M technical standard. Both of these works use the standard's important security elements in their own ways. Puts forward a "Privacy Enforcement" plugin to handle the needs of managing resources and privacy.

The security study of the Internet of Things (IoT) network implemented at IIIT-H with over 200 nodes for various smart city applications is the main topic of this work. The first step is to use the STRIDE approach to simulate possible risks and attacks against oneM2M standard compliant implementations. Secondly, this network undergoes five security assessments: eavesdropping, a packet replay attack, authorisation via access control rules, OM2M credential brute force assault, and scalability requirements. Third, the report suggests options that include setting up OM2M provisions to guarantee smart city baseline security. No genuine smart city deployment based on OM2M has, as far as I am aware, undergone this kind of security examination.

Here is the breakdown of the work: In Section 2, we cover the oneM2M standard. In Section 3, we go into modelling the primary possible dangers to any system that employs the oneM2M standard. The OM2M platform and its smart city deployment via IIIT-H are detailed in Section 4. Section 6 then proposes baseline security measures for OM2M-based smart cities, after which Section 5 offers a security analysis of OM2M. Section 7 ends the study and discusses the potential next research directions.

European Economic Letters
ISSN 2323-5233
Vol 14, Issue 2 (2024)
http://eelet.org.uk

ONEM2M STANDARD

An interoperable horizontal platform for developing vertically scalable applications in the Internet of Things paradigm is the goal of the oneM2M global standard effort, which is spearheaded by eight national standardisation bodies and a number of sectors. It gives the technological specs that meet the needs of a standard M2M service layer. The common service layer is embedded in a broad range of software and hardware components, allowing for the seamless connection of various field devices to M2M application servers throughout the globe. The oneM2M standard is used to integrate all components of the Internet of Things into a solution stack.

Architecture

Common service entities (CSEs), application entities (AEs), interworking proxy entities (IPEs), and network service entities (NSEs) make up oneM2M's functional architecture.

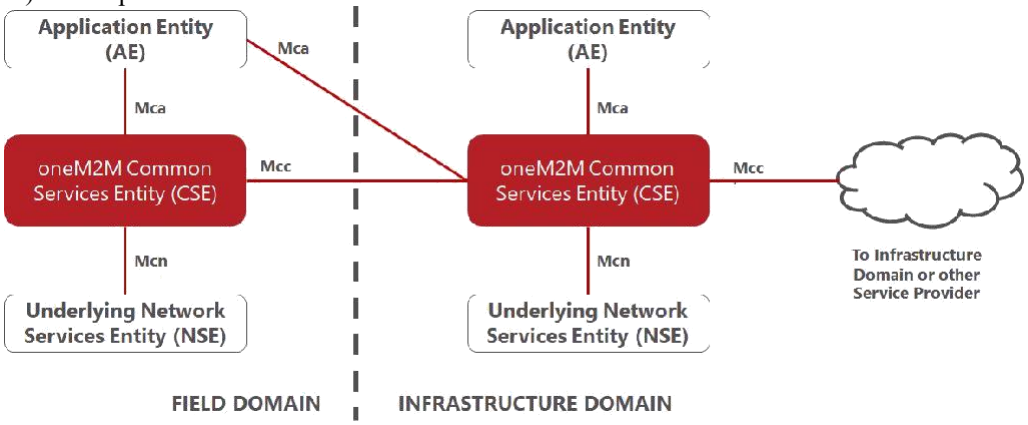

Figure 1: oneM2M architecture

Located in the sensors and interacting with the M2M service layer using RESTful (REST) APIs, application entities manage the application layer in an IoT system. The goal of IPE is to facilitate communication between the service layer and devices that aren't part of the oneM2M network. The CSFs are provided by the common service organisations. Security, data management, discovery, and registration are all part of it. The communication for services including device triggering, minor data transfer, location notification, and location enquiries is managed by network service entities. Logical entities known as nodes generally include CSEs and/or AEs in the oneM2M System. The field domain and the infrastructure domain are the two primary types of nodes. While bigger computers manage all the servers and applications, the "Field Domain" is responsible for things like sensors, actuators, and gateways. Various OneM2M reference points, including Mca for AE-CSE communication, Mcc for inter-CSE communication, and Mcn for CSE-NSE communication, are used for the entities' communication.

oneM2M for smart city: Threats and vulnerabilities

Technical report gives a high-level overview of security threats and countermeasures for oneM2M-based systems. The document briefly explains the security services of oneM2M, security requirements, threats to oneM2M systems, and suitable mitigation recommendations. This section presents Table 1 describing the potential security threats and vulnerabilities in an oneM2M based smart city-centric im-plementation. Additionally, security provisions of, and technical specifications are modeled using the STRIDE threat modeling framework. STRIDE is an acronym for Spoofing, Tampering, Fourthly, there is denial of service, disclosure of information, re-pudiation, and elevation of privilege. STRIDE is a better methodology for evaluating risks than PASTA, OWASP, and MITRE Att&ck since it focusses on products and development. The following definitions apply to each STRIDE element that can affect oneM2M systems:

Spoofing: It is a method by which cybercriminals pose as trustworthy entities in order to get access to private information, change rules, or alter communications. Spoofing compromises oneM2M's cryptographic keys, passwords, and the identifiers of CSE and AE nodes. Protecting against these dangers are the mutual authentication procedures that are part of oneM2M's safe relationship setup process.

Tampering: An attacker may compromise both authorisation and integrity in this way. An attacker may commit tampering if their illegal actions cause changes to a system, component, intended function, or data. Possible examples of such interference in a smart city configuration include physical manipulation of sensor nodes, communication channels, primitives, the oneM2M service capability layer, and the dependencies of the oneM2M system.

Repudiation: This happens when a system or program doesn't have the proper mechanisms in place to keep track of user actions, which leaves the door open for malicious editing or the false detection of new actions. You may use it to manipulate generic data under someone else's identity, much like spoofing mail messages. The data stored in log files would be considered misleading or false if this attack were to occur. A possible way to avoid non-repudiation violations

is to keep track of user actions and system operations. Messages between entities, user profiles, and primitives may all be logged by oneM2M systems.

Information disclosure: Information disclosure (also termed information leakage) occurs when a website or app makes private data accessible to those who shouldn't have it. Websites may expose a wide range of information to potential attackers, depending on the circumstances. This includes details about other users, such as their identities or financial information, sensitive business or commercial data, the website's design and technical specs, and more. In addition, protocols for communication that are not secure, including HTTP and MQTT, might expose confidential data. Secure data storage and execution of critical functions are both aided by the oneM2M-proposed secure environment.

Denial of service: An attacker's goal is to disrupt a computer system or network in order to prevent its intended users from accessing it. To do this, denial-of-service attacks flood the target with data or traffic until it crashes. In both cases, the service or resource that legitimate users were expecting is denied by the DoS attack. Buffer overflows, ICMP floods, and SYN floods are just a few examples of the many threats that may compromise data and service availability. OneM2M systems are vulnerable to denial-of-service attacks that include delivering data in an incompatible format, repeatedly querying resource trees via the REST API, or overwriting buffer limitations.

Elevation of privilege: An attacker may gain administrator, root, or more privileged access to the system by escalating their privileges. It compromises the safety of oneM2M's shared service operations and causes an authorisation violation. This danger might arise from inadequate authentication procedures, unused open ports, or misconfigurations.

Table 1: STRIDE analysis of oneM2M

STRIDE	Potential threats to oneM2M standard based implementations	Existing security provisions in oneM2M
Spoofing	AE impersonation Broken authentication Session hijacking	AE impersonation prevention Authentication mechanisms: Symmetric key-based security, Certificate-based security, Generic Bootstrapping Architecture (GBA) framework
Tampering	Corrupted service layer software Unauthorized access to oneM2M software dependencies Alteration of primitives transmitted over the Mca/Mcc/Mcc' reference points - Physical tampering of nodes	"m2m: software update" provides consistent updates to the end nodes to patch the security vulnerabilities "m2m: DeviceID" uniquely identifies a device using a URN Device certificates to authenticate the AEs or CSEs
Repudiation	Unavailability of access logs Log injection/tampering/forging	Token based authorization Role based access control "m2m:logStatus", "m2m:logTypeId" to check the log status of the event management resource
Information Disclosure	Insecure communication protocols Replay of M2M primitives between entities Device hijacking Network manipulation attacks Exposed sensitive data in AE or M2M gateways	oneM2M protocol bindings ESPrim Secure environment plug-in
Denial of	Buffer overflow Flooding of RestAPI requests	"m2m: access Control Rule" (with assigning access using m2m:ipv4,

Service		m2m: ipv6 and m2m: location Region) Mutual authentication through SAEF
Elevation of Privilege	Privileged insider attack Access mechanism violation Insecure cryptographic storage Exposed Long-Term Service-Layer Keys	"m2m:authorizationStatus" provides status of access control policies Dynamic authorization for token based temporary permissions

OM2M implementation at IIIT-H

The OM2M platform and how it fits into the smart city setup are examined in this section. The platform's security features and how well they meet security needs are investigated via a series of experiments and observations. A genuine IIIT-H dense IoT implementation in India is used for the trials.

Smart city at IIIT-H

The present OM2M-based smart city rollout. More than 200 nodes have been installed so far, spanning 66 acres in and around IIIT-H. Smart room applications (air conditioning, occupancy, air quality, energy monitoring), smart campus applications (smart street lighting), and weather and energy monitoring are all part of the smart applications addressed. Data is sent from each of these nodes to the OM2M server over several networks, including Wi-Fi, 4G, Wi-Sun, and LoraWAN. The OM2M server thereafter utilises the subscription technique to transfer the data to the data warehouse. Different applications make different use of the data housed in the data warehouse. Data is retrieved from the warehouse via the smart city dashboard, smartphone apps, Alexa interface, and home automation systems. To access the data, the average user has to sign up for the Indian Urban Data Exchange (IUDX). To get a token on IUDX, users have to sign up on their own. The token grants access to the OM2M server for the user to examine data.

SECURITY ANALYSIS OF OM2M

Here, five trials' worth of findings on OM2M's default security settings at IIIT-H are presented. These tests provide light on the OM2M protocol, data formats, authorisation and authentication settings, and general architecture.

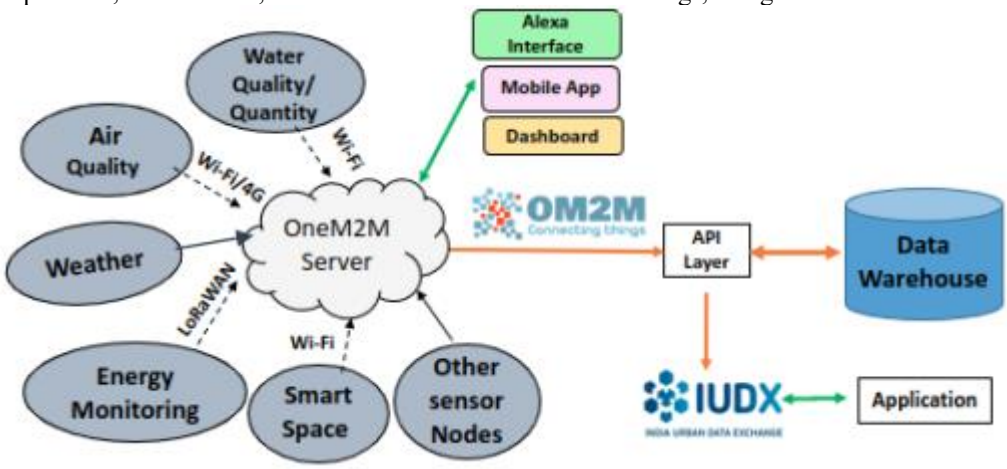

Figure 2: OM2M based smart city deployment

building design. A specialised Linux-based PC was configured as the OM2M server prior to the commencement of the tests in accordance with the procedures outlined by the Eclipse foundation.

Eavesdropping attack

Wireshark was used for eavesdropping purposes in order to intercept the OM2M server's and AE's communication. Two personal computers, one running Windows and the other Linux, were used for the experiment. Two programs were active on the Windows OS computer. To test the OM2M server, the API-building program Postman generated OM2M-specific APIs and uploaded fake sensor information. Wireshark, a program for packet analysis, was the second piece of software. The OM2M server was installed on the PC running Linux. Data was being pushed by Wireshark, which was running in promiscuous mode on the machine. After filtering for HTTP request and response packets in Wireshark,

many packets emerged. One HTTP-based packet and its contents after passively eavesdropping on the network are shown in Fig. 3.10. Notes were taken of the following:
The standard includes plugin support for many protocols, including Web-socket, COAP, HTTP, and MQTT. The default communication protocol for OM2M is HTTP. The header and payload are exposed due to the insecure default setup.
Origin of X-M2M: The authentication mechanism for managing the resource tree is this username/password combination. When configuring OM2M, be sure to include the X-M2M credentials in all API calls. These credentials will be utilised by every node in the network. The original requester, who may be a CSE or an AE, is responsible for assigning this parameter.
This header value makes it easy for an attacker to craft bespoke requests, which in turn may cripple the infrastructure.

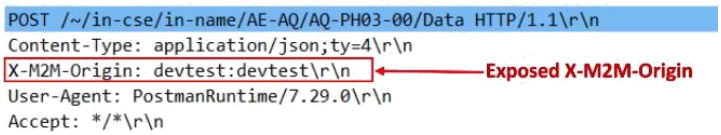

Figure 3: Confidential X-M2M-Origin visible in HTTP packet on Wireshark

Brute force attack for OM2M credentials

In order to access the entries of the resource tree, users must login to the OM2M login page using credentials assigned to them by the admin. Users and roles are authorized through ACPs in OM2M. With access to those credentials, an attacker can manipulate the entire resource tree and change the admin credentials, resulting in a DoS on OM2M administrators. Brute force, dictionary attacks and social engineering can be used to obtain the credentials. It is found that there is a possibility of launching wide-scale attacks by several thousand devices (botnets) or attempting many passwords on the single OM2M server. There is no mitigation mechanism such as blocking the attacker's IP address. Further, there is a limit on the type of special characters allowed for the passwords making the brute force attack easier. Passwords with characters such as '(', ')', '[', ']' were not allowed.

Authorization via access control policies

Two types of user profiles, admin and guest, are stored in the configuration file of the basic OM2M server. These profiles are used to access the setup. The admin profile can access the OM2M setup and do CRUD actions on all entities, whereas the guest profile can only see the resource tree. A variety of entities, including the AE, IPE, and ACP, are stored in the resource tree. Access Control Operations (ACORs) like create-1, retrieve-2, update-4, delete-8, notify-16, and discover-34 are assigned to various application entities by ACP in the OM2M configuration. As part of the experiment, a new user was established with the credentials ("devtest"). Its ACOR was set to CRUD operation 34 for the user. The OM2M platform's resource tree may now be accessed by any user with these updated credentials. Viewing the resource tree with the admin credentials is no longer possible, but all CRUD activities are still controlled by those credentials.

Scalability requirements

For smart cities to be successful, scalability is crucial. The OM2M installation sets the H2 database as the default database for storing resource tree entries. The server began recording null point errors and locking objects when the number of requests from IoT nodes exceeded a certain threshold. Consequently, the database was moved from the fast in-memory H2 database to MongoDB, an auto-scalable NOSQL database for production development, in order to address the scalability difficulties. This database

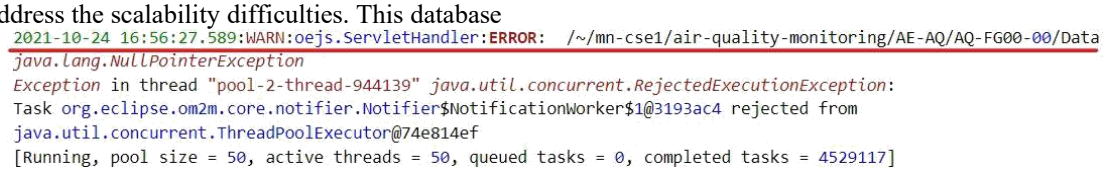

Figure 4: H2 database server crash - a scalability issue

modification prevented a denial-of-service attack by fixing the server crash problems. Viewed are 944,139 threads. Each of the more than 200 nodes in the network generates threads and stores them in the thread pool in an effort to transmit data to the server. When there are exactly as many threads as there are pools, an error occurs. Here, the write lock waits 600 seconds; after that, the thread becomes locked with all of the entities in its future container. It is the Servlet Handler Error that the server throws that resolves the Null Pointer Exception and the Rejected Execution Exception.

Packet replay attack

A replay attack occurs when a malicious actor delays or resends packets sent from a client to a server at varied intervals by intercepting (sniffing, eavesdropping) the packets. Systems that are unable to distinguish the origin of the many API requests received by the servers are vulnerable to this attack.

The MITM Proxy cybersecurity tool, which is based on Kali Linux and allows for replaying of web traffic and penetration testing, was used for this demonstration project. The program is useful for eavesdropping on HTTP and HTTPS communications and playing them back. Note that although all nodes on campus utilise HTTP connections, only nodes on the outside of campus network may connect to the server using HTTPS. An ESP8266 equipped with an SDS011 sensor transmits data perceived to an OM2M server prior to the experiment. Every five seconds, the detected data is delivered, which causes the resource tree to have many instances of content. You can see how the experiment turned out by referring to this resource tree. Here is how the experiment goes:

- The OM2M server was contacted by Postman, who generated a post request and sent a value—here 30,—at random.
- The man-in-the-middle proxy intercepts this request.
- Using Postman, the MTM proxy monitors other client-side requests and re-executes the recorded packet at a later time.

The server does not have the necessary mechanisms to verify the validity of any API call, as shown by the observation of a new content instance being created in the reference resource tree with the value specified by the postman. Client nodes must have CA certificates issued to them in order for an HTTPS connection to be established. Physical manipulation is possible due to the dispersed nature of the smart city nodes. It was noted that the OM2M server was unable to detect the client-side false CA certificate. Other well-known websites quickly detected the MITM proxy's bogus CA certificates as being fraudulent. X-M2M-RT (request timestamp) and other measures are detailed in the OneM2M security solutions paper to help against these types of attacks. It is also possible to make advantage of other options included in the standards, such as setting up sessions with a predetermined session time.

CONCLUSION

The STRIDE threat modelling framework is used to study and simulate the possible vulnerabilities and dangers associated with oneM2M implementations. By using the platform's existing default settings, experiments are conducted to examine the effect of these vulnerabilities on a real-life smart city deployment based on OM2M. Following the STRIDE framework, the necessary components for smart city baseline security are proposed.

REFERENCE

1. Samih, Haitham. (2019). Smart cities and internet of things. Journal of Information Technology Case and Application Research. 21. 1-10. 10.1080/15228053.2019.1587572.
2. Chaudhary, Gaurav & Agrawal, Harsh & Tirkey, Sneha & Nath, Vijay. (2020). Study and Design of Smart City with Internet of Things: A Review. 10.1007/978-981-15-2854-5_47.
3. Harmon, Robert & Castro-Leon, Enrique & Bhide, Sandhiprakash. (2015). Smart cities and the Internet of Things. 485-494. 10.1109/PICMET.2015.7273174.
4. Toh, Chai. (2020). Security for Smart Cities. IET Smart Cities. 2. 10.1049/iet-smc.2020.0001.
5. Krichen, Moez & Lahami, Mariam &Cheikhrouhou, Omar &Alroobaea, Roobaea&JmalMaâlej, Afef. (2020). Security Testing of Internet of Things for Smart City Applications: A Formal Approach. 10.1007/978-3-030-13705-2_26.
6. Alavi, Amir & Jiao, Pengcheng&Buttlar, William &Lajnef, Nizar. (2018). Internet of Things-Enabled Smart Cities: State-of-the-Art and Future Trends. Measurement. 129. 10.1016/j.measurement.2018.07.067.
7. Abderahman Rejeb et.al "The big picture on the internet of things and the smart city: a review of what we know and what we need to know" Internet of Things Volume 19, August 2022, 100565
8. Singh, Debabrata & Pati, Bibudhendu& Panigrahi, Chhabi & Swagatika, Shrabanee. (2020). Security Issues in IoT and their Countermeasures in Smart City Applications. 10.1007/978-981-15-1483-8_26.
9. Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. Smart Cities 2021, 4, 429-475. https://doi.org/10.3390/smartcities4020024
10. Ismagilova, E., Hughes, L., Rana, N.P. et al. Security, Privacy and Risks Within Smart Cities: Literature Review and Development of a Smart City Interaction Framework. Inf Syst Front 24, 393–414 (2022). https://doi.org/10.1007/s10796-020-10044-1
11. Talebkhah, Marieh& Sali, Aduwati& Marjani, Mohsen & Gordan, Meisam & Hashim, Shaiful &Rokhani, F.Z.. (2021). IoT and Big Data Applications in Smart Cities: Recent Advances, Challenges, and Critical Issues. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3070905.
12. Park, E.; Del Pobil, A.P.; Kwon, S.J. The Role of Internet of Things (IoT) in Smart Cities: Technology Roadmap-oriented Approaches. Sustainability 2018, 10, 1388. https://doi.org/10.3390/su10051388
13. Tragos, Elias & Fragkiadakis, Alexandros & Angelakis, Vangelis &Pöhls, Henrich. (2017). Designing Secure IoT Architectures for Smart City Applications. 10.1007/978-3-319-44924-1_5.
14. Andrea Zanella et.al "Internet of Things for Smart Cities" IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 1, FEBRUARY 2014