Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06441-5

S.I. : IMPROVING HEALTHCARE OUTCOMES USING MULTIMEDIA BIG DATA ANALYTICS

A threat intelligence framework for protecting smart satellite-based healthcare networks

Muna Al-Hawawreh[1] · Nour Moustafa[1] · Jill Slay[2]

Received: 28 April 2021 / Accepted: 17 August 2021

Abstract

Human-to-machine (H2M) communication is an important evolution in the industrial internet of health things (IIoHT), where many H2M interfaces are remotely interacting with industrial and medical assets. Lightweight protocols, such as constrained application protocol (CoAP), have been widely utilised in transferring sensing data of medical devices to end-users in smart satellite-based healthcare IIoT networks (SmartSat-IIoHT). However, such protocols are extensively deployed without appropriate security configurations, making attackers' mission easier for abusing these protocols to launch advanced cyber threats. This paper, therefore, presents a new threat intelligence framework to examine and model CoAP protocol's attacks in these systems. We present a ransom denial of service (RDoS) as a new threat that would exploit this protocol's vulnerabilities. We propose many RDoS attack's techniques to understand the attack indicators and analyse their behaviour on systems. Moreover, we present a real-time discovery of attacks' network behaviours using deep learning. The experiment results demonstrate that this proposed discovery model obtains a better performance in revealing RDoS than other conventional machine learning algorithms and accomplishing high fidelity of protecting SmartSat-IIoHT networks.

Keywords Smart satellite · IIoHT · DDoS · Ransom · Machine learning

1 Introduction

The Industrial Internet of Thing (IIoT) is becoming a critical part of healthcare systems and the medical world (i.e. IIoHT) that scale from a closed cyber-physical control loop (i.e. sensor, controller and actuator) to massive cross-platform deployments of connected industrial and medical systems, edge and cloud technologies connecting in real time [1]. The emerging technology of IIoHT would be connected using Smart Satellites (SmartSat) and new communication protocols that enable flexible and rapid connectivity between on-premise devices, humans, physical and medical assets [2, 3]. Humans usually interact with and remotely control connected physical and medical assets. They can use their mobile applications to apply the recommended decisions directly to the connected devices, read telemetry data from sensors and send commands to actuators [4]. This is an example of human-to-machine (H2M) messaging communication, where humans are involved in the cyber-physical control loop to assert the legitimate behaviour of these devices at the application layer [5].

CoAP is one of the most common messaging protocols that is broadly deployed as an H2M messaging channel in SmartSat-IIoHT networks [6]. As a case study of smart healthcare systems, doctors can read data of patients heart rates on their smartphones from an implanted chip [7]. The

✉ Muna Al-Hawawreh
m.al-hawawreh@student.adfa.edu.au

Nour Moustafa
nour.moustafa@unsw.edu.au

Jill Slay
Jill.Slay@unisa.edu.au

[1] School of Engineering and Information Technology, UNSW Canberra at ADFA, Campbell, Australia

[2] University of South Australia, Adelaide, Australia

Published online: 08 September 2021

human operators can read the temperature and humidity in COVID-19 vaccines using mobile applications. The interaction between mobile human-machine interface devices (e.g. smartphones and tablets) and critical physical assets and medical devices has increased attack surfaces [8]. The critical concern comes from the emerging messaging protocols such as CoAP, which are deployed without appropriate configurations and security mechanisms [9, 10]. Considering that humans are the weakest link in the security chain, attackers can easily inject malware into human's (e.g. doctor, nurse, or operator) mobile devices to exploit H2M messaging protocols (e.g. CoAP) and launch advanced threats against endpoints [3, 8]. For example, CoAP has been recently exploited for performing DDoS attacks, and it is expected to be the most abused protocols for such threats [11].

The integration of the recent information technology (IT) (e.g. cloud, edge, mobile, satellites and current connectivity protocols) with operational technology (OT) (medical and physical assets) is an extremely difficult task to entirely secure SmarSat-IIoHT networks. It is hard to protect thousands of new communications and endpoints, each of which with its features and firmware to update, against every potential threat [3, 12]. This raises the need for implementing a more effective and proactive security approach such as threat intelligence (TI) [13, 14]. TI refers to any evidence or event-based knowledge about potential attacks that highlight the risk landscape [15]. It can be described as a big picture of the attacker's intention and capability to target a specific asset that enables the organization (e.g. healthcare) to prepare this threat and defend against it [16]. Understanding the potential threats and the nature of attacks that would abuse these IIoHT protocols (e.g. CoAP) and collecting data fundamentals associated with these threats can enhance the security team's situational awareness and develop efficient risk assessment models.

Current TI solutions for SmartSat-IIoHT highly depend on the intelligence related to traditional IT system attacks and their indicators (e.g. malicious URL and blacklist IP) and OT malware behaviour (e.g. Triton) against physical assets [14, 17]. Unfortunately, this sort of TI is not sufficient to provide holistic security. There is a need to keep up with TI related to new and specific SmartSat-IIoHT attacks, which can be associated with emerging protocols (e.g. CoAP) and deployed devices [18]. The research in this area is still in its early stages, and much effort is required. This is highly needed for the CoAP protocol as it is extensively used in operating critical healthcare applications without any consideration for its security. Therefore, some research focus on the CoAP protocol is required to protect the healthcare systems against any potential attacks.

This paper proposes a new framework that describes the RDoS-CoAP threat intelligence modelling. It elucidates how the RDoS attack can exploit the CoAP protocol's weakness to affect the critical physical process in SmartSat-IIoHT systems, discover its behaviour and indicators, make the decision, and perform the appropriate actions to prevent or mitigate this attack. Our paper is the first to introduce a RDoS where attackers exploit the CoAP protocol to send multiple requests to the available resource for affecting the server endpoint and threatening to organize a huge DDoS attack in case of ransom is not paid. Considering the critical of the actions performed by physical and medical assets (i.e. sensors), volumetric RDoS-CoAP attacks could have more devastating consequences to the whole system and human safety, making such a system much easier profitable for attackers. The key contributions of this paper thus as follows:

- We design a system architecture that illustrates the implementation and integration of Smart Satellite and IIoHT systems.
- We propose a new framework that describes the RDoS-CoAP threat intelligence modelling. In this framework, we introduce a Ransom Denial of Service (RDoS) threat to exploits network vulnerabilities of COAP protocol. It is worth noting that such an attack has not been presented yet.
- We propose several attacker tactics, techniques and procedures for performing RDoS-CoAP attacks.
- We discuss how the proposed and extracted TI can be used to make decisions and actions to prioritize and enrich defence mechanisms. We also highlight key challenges with their implementation in SmartSat-IIoHT networks.
- We propose an online discovery model using long short-term memory (LSTM) to reveal such attacks and protect the SmartSat-IIoHT networks.

The rest of the paper is organized as follows. Section 2 describes the background and related work. Section 3 describes our proposed framework and the testbed architecture. Our proposed intelligence-driven threat discovery model is introduced in Sect. 4. Section 5 presents experimental results and discussions of TI and discover models. Lastly, the conclusion and future work are presented in Sect. 6.

2 Background and related work

2.1 Smart satellites-based healthcare systems

Satellites play today a critical role as an alternative for the cellular network in connecting remote IIoT devices in any sector. They are a solution for the fragmented regulations and enhancing global connectivity in low-power wide-area (LPWA) networks for supporting long-range, low-bit rate and low-power of different IIoT devices [13, 19]. They have unique traits in linking IIoT devices and assets, providing truly all-embracing coverage to reach items with or without limited access to terrestrial or cellular networks. This sort of communication is highly reliable where the proper satellite constellation provides more than 99% availability (much higher than the cellular network), and a consistent service across the coverage area [19]. In SmartSat-IIoHT systems, as described in Fig. 1, satellite is integrating with deployed devices and systems to facilitate earth's long-range digital communication in different aspects of healthcare.

A satellite can support hospitalization, surgery, pre-hospitalization care, nursing, telemedicine and remote health monitoring. It receives signals from IIoHT sensors deployed in rural areas, mountains, peaks, oceans and other places where communication cannot reach easily, amplifies and enhance these signals and then sends them back to the earth. As healthcare systems operate critical services, they require dependable, high-throughput and low latency connections. These requirements can be achieved by using and deploying appropriate satellites [19]. For example, satellites that operate close to the earth, such as Low Earth Orbit (LEO) and Highly Elliptical(HEO), are the best candidates as they are light and provide low path loss and latency [20]. They can be integrated with wide body area network (WBAN), which operates inside the body and within a limited range, to collect the data and send it to the cloud or remote healthcare providers. High throughput satellites (HTS) are also integrated with healthcare systems to increase the speed and capacity of LEO and HEO satellite constellations as they utilize spot beams and high-frequency bands and reuse [19].

The adoption of satellite in the healthcare industry has recently been accelerated by the increasing prevalence of chronic and contagious diseases. Particularly, the abrupt emergence of the CoVID-19 pandemic threw a significant burden on the organisation and provided health and social services, especially in isolated regions with weak internet connectivity. This virus's unregulated spread often affected the mental and physical of the elderly and vulnerable people because of the virus and the imposition of physical distances. SmartSat is presented as a solution to deal with pandemic and mitigate the impact of COVID on mental and physical issues to the people in rural area or regions without cellular or internet connectivity [21]. It can support the immunisation programming by assisting with health resource mapping, population estimation, immunisation services microplanning, modelling regional accessibility to health services, disease tracking, campaign monitoring and modelling vaccine coverage [19, 21].

For example, in SmartSat-IIoHT systems, a tele-health service is used in remote sites to connect health experts to others via satellite communications, or to patients anywhere in the world (e.g. rural area) [13]. Moreover, maps can be compiled with satellite Earth observations (EO) and global satellite positions to produce geospatial data. Typically, these satellite data are incorporated into the geographical information system to visualise these data or to conduct advanced spatial assessments using other land characteristics (such as highways, buildings and

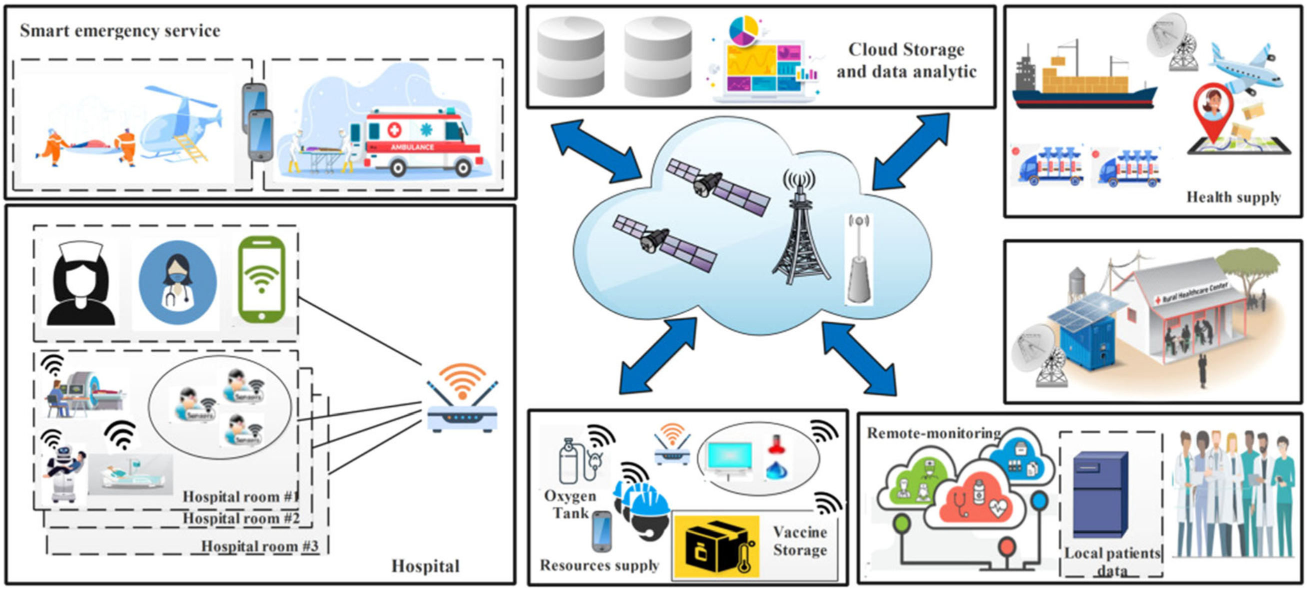


Fig. 1 Smart satellite IIoHT systems

landmarks), and populations (such as how many people there live in a given area) in smart city [21]. SmartSat mapping can help locate hospitals and other healthcare facilities or locate a relief workers' camp [22]. In case of emergency and if the patient's location is far away from hospitals, the treatment can be provided remotely using tele-medicine. The near doctors can provide the appropriate care based on given advice from remote specialists through high-quality video calls.

Integration SmartSat, and IIoT technologies with healthcare systems and deploying new devices and connectivity protocols create major cyber security concerns. This is because most deployed medical devices are designed as stand-alone without communications and networking or any considerations for cyber security requirements. Other devices have been deployed with poor and default passwords, or their security is vendor task-specific such as X-rays scanners. These devices are nowadays integrating with mobile applications, wireless, new technologies (i.e. edge, fog and cloud computing), and new connectivity protocols (e.g. CoAP and MQTT) [18, 23]. Furthermore, new services are introduced and stored in the cloud, such as E-prescription services and health electronic records (HER). Implementing these services and technologies without any risk assessment and management plan has expanded the threat landscape and paved new ways for targeting these SmartSat-IIoHT systems by attacks such as RDoS.

2.2 An overview of threat intelligence

Threat intelligence (TI) can be defined as any information or knowledge obtained about malicious attacks through the collection, transformation, observation, analysis and interpretation [24]. It also can be described as the process of analyzing information about indicators of potentially malicious attacks, which allow organizations to take the appropriate action for safeguarding their systems and network [14]. Also, TI can be described as information related to the adversary's or attacker's intention, capabilities and opportunities. According to studies [25, 26], the threat consists of capabilities multiplied by intention and opportunities; if anyone is zero, the threat is zero. This TI should be relevant, actionable and valuable. The relevant means it should have sufficient information about potential targets (opportunities), the attacker's intention (purpose) and attacker capabilities, tactics and technique; actionable means information should be sufficient to take the serious action and decision to prevent or detect attacks. Valuable means it should help in securing the business outcome [13]. The information related to the attacker's intention and capabilities is the most essential TI and business priority as they can be implemented and shared among organizations.

TI is commonly multi-purpose, and it can be used in many practical ways before, during and after the attacks. By integrating TI, which is a collection of correlated data points about potential threats, with intrusion detection systems (IDSs) and firewalls, new and known threats can be easily detected before attacks happen. This means that TI can be used as a data or information source for IDSs by providing them with attack patterns. During attacks, intelligence-driven attack detection can be used to speed up the detection time and response process. This also helps prioritise relevant IOCs and focus more on severe security alerts. After attacks, TI can be used in forensic, attack investigation and reporting after attacks which help the cyber security incidents response team to provide the required actions [24, 25].

2.3 An overview of CoAP protocol

CoAP is a client-server application protocol similar to the HyperText Transfer Protocol (HTTP) but on the top of the UDP protocol. Given its flexibility and lightness, it is being adopted to provide all kinds of communications such as H2M, machine-to-machine (M2M) and machine-to-human (M2H) [4]. CoAP exchanges request/response messages (as shown in Fig. 2). Each message consists of a fixed size (bytes) headers including information such as version number (two bits), message type (i.e. Confirmable (CON), Non-confirmable (Non-CON), Acknowledgement (ACK) and Reset (RST)), token length (4 bit), method code (8 bits) that is unique for request/response message (i.e. GET, POST, DELETE, OBSERVE and PUT) and message ID (16-bit) for detecting duplicate messages and matching ACK/RST to CON/NON-messages. These contents are also followed by a variable-length token value (0 to 8 bytes) to correlate request/response message, a sequence of zero or more options in Type-Length-Value (TLV) format, optional Uniform Resources Identifier (URI) and payload.

To access CoAP resources, a specific URI ("$/resource\_name''$) is used to connect with a server on default UDP port 5683 [8, 10]. To address the lack of data transmission reliability over UDP, CoAP implements lightweight reliability features including message ID to detect any message duplication and stop-wait mechanism with a back-off exponential retransmission time [5]. CoAP also supports block-wise for dividing a large payload into blocks and sending them separately, which tackles the risk of amplification-DDoS attacks. Although CoAP protocol supports multiple security mechanisms, including Datagram Transport Layer Security (DTLS) at the transport layer and Internet Protocol Security (IPSec) at the network layer, it does suffer from several internal and external attacks such as IP spoofing, URI parsing attacks, DoS/DDoS and cross-protocol attack [8].

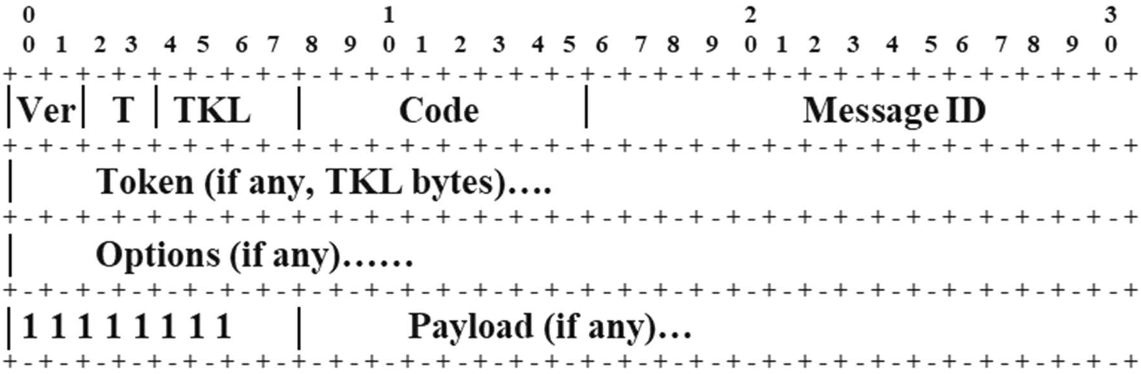


Fig. 2 CoAP message packet

Most deployed SmartSat-IIoHT messaging protocols such as CoAP are poorly designed, configured and implemented. This allows attackers to execute a malicious command, re-programming firmware, inject malware in the endpoint or expose sensitive information [3]. Among the many cases that recently attracted the researchers' attention, growing CoAP-based devices as one of the most prevalent types of cyber weapons in a DDoS-botnet [27]. Given the criticality of SmartSat-IIoHT devices, such malicious attacks generate a profound impact on the system's resiliency and reliability and human safety. Therefore, generating TI for such protocol is becoming of the utmost importance proactively discovering attacks and taking the appropriate actions before it might happen. Thus, exploring the protocol's vulnerabilities, providing potential attack scenarios, and extracting malicious attack indicators and patterns will give proper visibility in such emerging security issues to reduce the potential risk of disrupting critical healthcare system operations.

2.4 Related work

In this section, we provide reviews for state-of-the-art research related to CoAP security. Examples of the existing literature that can be categorized under the banner of threat intelligence related to CoAP protocol include the study of [28] where the authors presented a comprehensive study for IIoT lightweight protocols (e.g. MQTT, AMQTT and CoAP) and their security issues and vulnerabilities. They found and detected many security issues in each of the protocols and; thus, they provided a framework to measure the risk of these vulnerabilities. These studies' purposes were to provide intelligence related to new IIoT devices and protocols that can be used to protect them from potential threats. Our work also aims to provide intelligence about potential threats that can exploit CoAP protocol to affect the endpoints.

Studies of [29–32] provided intelligence specifically for CoAP protocol; these studies introduced various DoS/DDoS CoAP attack scenarios such as malformed CoAP requests, a non-intended message for CoAP from other protocols such as TCP and ICMP, and invalid CoAP messages. They also introduced insights about detecting these attacks based on the count of active connections on a specific host or port, malicious and suspicious IP and the payload size. Vieira et al. [33] tested two attacks, including port scanning and host discovery against the COAP server. In related work, Canuto et al. [34] studied CON and NON-CON CoAP/CoAPS flow messages. Their experiments found that DDoS attacks can be detected based on the number of traffic flows where the number of received traffic flows from the external network is more than the number of sending flows from the internal IoT network. [30] extracted indicators from IEEE 802.15.4, 6LoWPAN, IPv6 and COAP protocols parameters for detecting invalid CoAP request DoS attacks. Similarly, Granjal et al. [35] investigated DDoS against CoAP and other protocols in the 6LoWPAN network.

In summary, existing cyber security studies on CoAP protocols focus on how the underlying layers' DoS/DDoS attacks (i.e. network and physical layer) affect the server. Other studies also concentrate on CoAP attacks, including sending malformed requests, unsupported resources (i.e. URI), and fake acknowledgement along with their countermeasures which highly relied on signatures and rules models with a predefined threshold. However, our work focuses on RDoS attack over CoAP that utilises the supported requests for sending multiple "GET" requests embedded with a ransom note for the available resource. We propose several attacker tactics, techniques and procedures. We analyse RDoS intention (i.e. motive) and their impact on server resource and logs, network traffic and physical asset, and discover and reveal attack behaviours using LSTM and based on extracted intelligence CoAP-RDoS attack. It is noteworthy that our paper is the first and novel TI framework to explore the vulnerability of CoAP through performing RDoS attack and hunting its indicators.

3 Proposed framework

3.1 Proposed RDoS threat intelligence modelling framework

In this section, we propose and design a new framework that describes the RDoS threat intelligence modelling. It elucidates how the RDoS attack exploits CoAP protocol's weakness to affect the critical physical process in SmartSat-IIoHT systems, how to discover its behaviour and indicators, making the decision and performing the appropriate actions to prevent or mitigate this attack, as shown in Fig. 3. It is worth noting that, this paper is the first to provide and design such a TI framework specifically for CoAP-RDoS attacks in SmartSat-IIoHT systems.

- **Threat observation** It includes different types of information about potential threat and vulnerabilities. It consists of RDoS attacks modelling to provide how attacks can exploit CoAP protocol in the H2M scenario to affect the physical process and the endpoint. It also includes the potential tactics, techniques and procedures that attackers can use to achieve their goal (obtaining financial profit). Furthermore, it describes behavioural and extracted indicators and their impact on the targeted system. This module is explained using RDoS attack scenarios, behaviour and impact in Sect. 4.
- **Discovering** It correlates the provided information or extracted intelligence to identify the RDoS-CoAP attacks. In our proposed framework, we utilise two techniques for discovering: In the first technique, we use the hunting technique to chase attacks and correlate its indicators. This will be provided through the paper during attack experiments. In the second one, we use deep learning to perform discovery task based on network-based intelligence. We choose to use the DL technique to provide an efficient way to discover the hidden pattern of the provided information and generalise it to new and unseen data [13].This module is explained as intelligence-driven RDoS discovery in Sect. 6.
- **Decision making** In this module, the extracted or the provided results from the previous model are pondered to select a cluster of actions to address the RDoS-CoAP attack (i.e. security recommendations). This can include defining which security approaches are the best to handle this attack. Section 5 explains and discusses the extracted intelligence and how it can be used to take the appropriate security decision.
- **Taking action** This module is used to perform the appropriate response or action to prevent and mitigate the RDoS-CoAP attack. This can include several steps such as updating the firewall to block the traffic, re-routing traffic through a router to different devices, and among others of possible containment solutions. This module is also used in two ways. First, it used to provide action and prevent the attacks before it happens as discussed in Sect. 5. The second way is to describe the action in case of post-discovering, as explained in Sect. 6.

The output of making a decision and taking action modules can reflect on the provided threat observations and used to add-on create, update, or delete any information or intelligence related to the attack. Our proposed can be deployed in the cloud segment of SmartSat-IIoHT systems to utilise high resources in storing and analysing collected data and

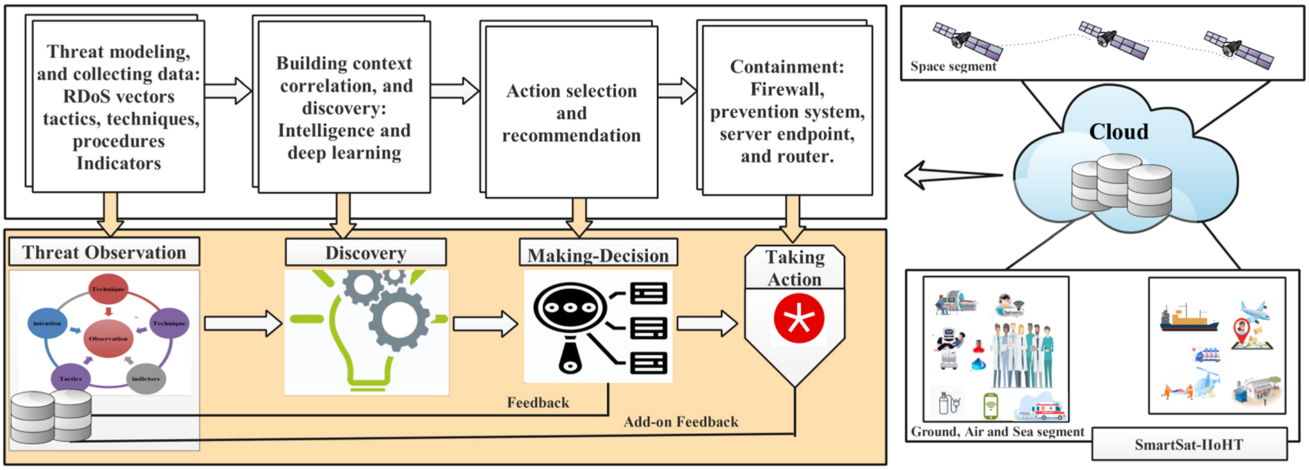

Threat modeling, and collecting data: RDoS vectors tactics, techniques, procedures Indicators
Building context correlation, and discovery: Intelligence and deep learning
Action selection and recommendation
Containment: Firewall, prevention system, server endpoint, and router.
Threat Observation
Technique
intention
Technique
Observation
Tactics
indictors
Discovery
Making-Decision
Taking Action
Feedback
Add-on Feedback
Space segment
Cloud
Ground, Air and Sea segment
SmartSat-IIoHT

Fig. 3 Proposed TI framework

sharing extracted intelligence among different systems components and segments. The extracted intelligence related to CoAP-RDoS in a local H2M scenario can be distributed and shared with other network segments. For example, this extracted CoAP-RDoS intelligence can be used to protect the CoAP protocol in satellite communications.

3.2 Proposed SmartSat-IIoHT testbed

We examine RDoS-CoAP using a case study of a SmartSat-IIoHT implementation where IIoHT protocols are used to connect between physical assets (e.g. sensors) and mobile application as H2M communication or among machines as M2M [4, 36]. The key issue with such implementation (i.e. SmartSat-IIoHT) is their deployed medical devices that were designed as stand-alone without communications, networking and any considerations for security requirements [37], along with other devices that have been deployed with poor and default passwords or their security is vendor task-specific. The integration of these devices with satellite, mobile applications, wireless and IT systems and using new emerged technologies and connectivity protocols without any risk assessment and management plan [38] make these systems unprepared for IIoHT protocols-specific attacks such RDoS-CoAP.

Our testbed architecture, as illustrated in Fig. 4, focuses on providing a simple prototype for the system, with a particular focus on CoAP implementation. This architecture considers the employment of edge server mediating the communications between physical devices (i.e. sensors and actuators), cloud and connected caregiver mobile applications, the usage of MQTT for sending data to the cloud broker over wired Ethernet, and the usage of CoAP over IEEE 802.11 wireless network (i.e Wi-Fi) for reading sensors and controlling actuators via operators mobile application. We use the MPL3115A2 sensor for sensing pressure and temperature measurements and light emitting diode (LED) device for actuating. These devices create a medical closed cyber-physical control loop.

We utilize “CoAPClient” application from App Store as mobile CoAP clients to access the resources of the txThings CoAP server [39], which is a Python implementation of CoAP based on Twisted-asynchronous I/O framework, that runs at Raspberry pi 3 B+ (i.e. edge server). We choose txThings as it is one of the most common open-source software, supports most of RFC standards, and it proved that it is one of the most robust servers against failures and error according to [40] study. We also use

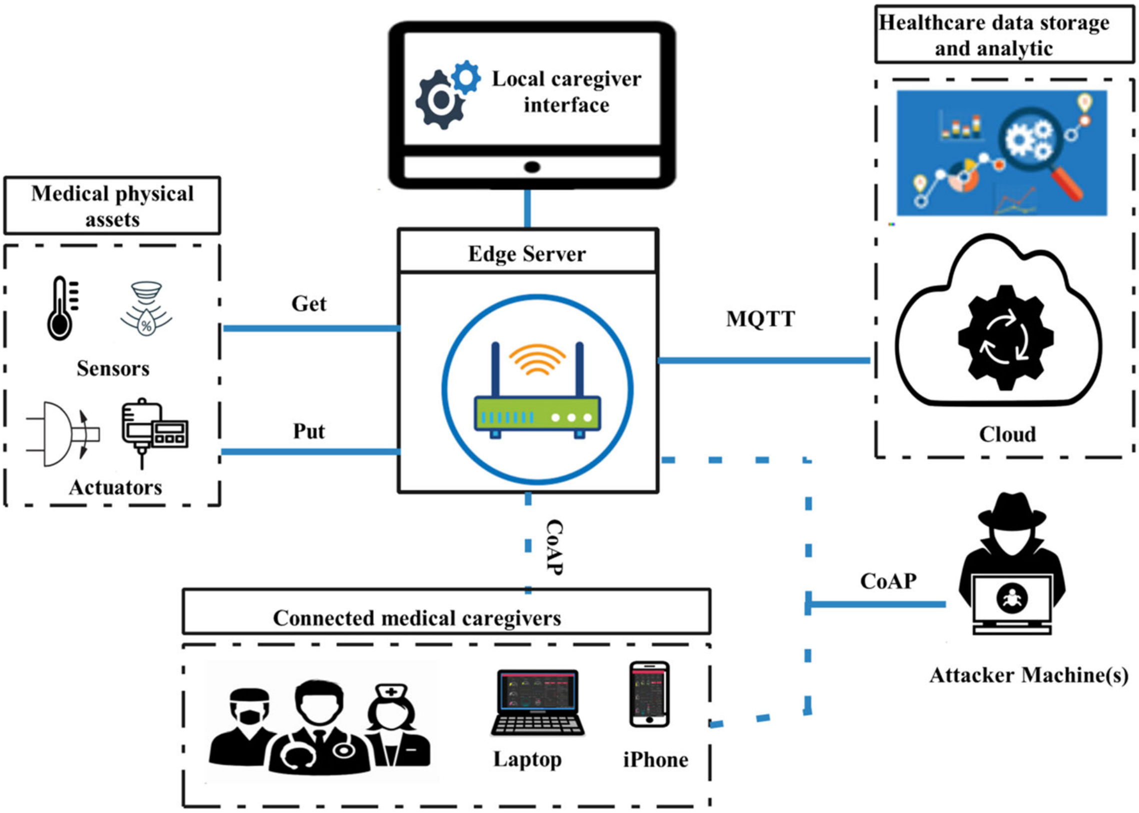


Fig. 4 Testbed architecture

txThings Python script as clients run at laptop devices to generate more CoAP traffic at the Wi-Fi network.

Here, caregiver (i.e. doctors or nurses) read the sensor values (i.e. temperature and pressure) by sending "GET" request to the available resource "/act", and they change/update the status of the actuator (i.e. infusion pump) by sending "PUT" request to the same resource. Note that, we use different CoAP platforms such as txThing clients and CoAPClient App to reflect the heterogeneous nature of SmartSat-IIoHT systems. The edge server polls data from the sensor acts on it and then sends it in a JSON format to CoAP clients, and vice versa for controlling the actuator. The edge server also provides a local caregiver interface and acts as an access point, and a publisher for telemetry data to the cloud broker (i.e. mosquito broker). Besides, we utilize multiple virtual machines running Kali Linux for performing our attack. In this current implementation of the testbed, we developed our Python codes for our CoAP resource, edge server tasks, MQTT messaging and RDoS attack.

3.3 RDoS attack modeling and scenarios

To describe and perform the RDoS-CoAP attack, we utilize the MITRE ATT&CK framework [41]. The ATT&CK framework describes in a structured way the actions (tactics, techniques and procedures) that may be taken by attackers when operating with an enterprise network. Based on this framework, we can present the RDoS-CoAP attackers' key impact tactic as extorting victims and obtaining profit by affecting system safety, availability, reliability and resiliency. The key RDoS attackers' technique can be here represented as a volumetric CoAP application-layer DDoS attack embedded with a ransom note, and the detailed implementation of this attack technique to achieve the impact tactic is represented as a procedure. The stages of such attacks can be described in Fig. 5.

As an initial step, attackers need to know what resources are available and supported by CoAP servers. They, therefore, may conduct several tactics, including initial access, credential access and execution, to fulfill this step. These tactics can include various techniques and procedures. For example, attackers can utilize the SHODAN engine to explore information about the online edge CoAP server. They can be in the range of Wi-Fi networks and perform critical reinstallation attacks. They then start dumping the resources list by sending a "GET" request with "/.well- known/core" as URI-path. Another technique can include utilizing the spoofed requests described in the study of [8], where attackers get access to legitimate user devices and communications. Mobile applications on the caregiver's devices can also be used to harvest information and launch the attack. While to achieve the final objective and impact, we propose several techniques and procedures to perform the RDoS-CoAP attack. The main focus is only on sending malicious "GET" requests to the available resource "/act".

3.3.1 Constant low rate RDoS technique

Consider a technique of an RDoS attack that may be derived from a constant low rate DoS attack. An attacker sends malicious traffic at a constant low rate for a short time. This type of attack in the long term can flood the server and prevent the service from additional requests from legitimate traffic. However, for the RDoS attacker's purpose, it is only for sending ransom notes and providing evidence that the attack is real. As shown in Fig. 6, the RDoS-CoAP attack is only constant low byte rate traffic with an embedded ransom note.

The attacker procedure is sending many confirmable "GET" requests with resource name "/act" (using only one device) for one minute and half a time (we assume 90 seconds, the same time that is used by the recent CoAP DDoS incident). Each CoAP request has an embedded ransom note as a payload asking for 1500 XMR or Monero (roughly $76048.88) and threatens to follow this attack with a volumetric DDoS attack in case of ransom is not paid by the deadline. Attackers send the ransom note in a message payload as they are aware that security analysts can see this note when they inspect the packets to defend their organizations.

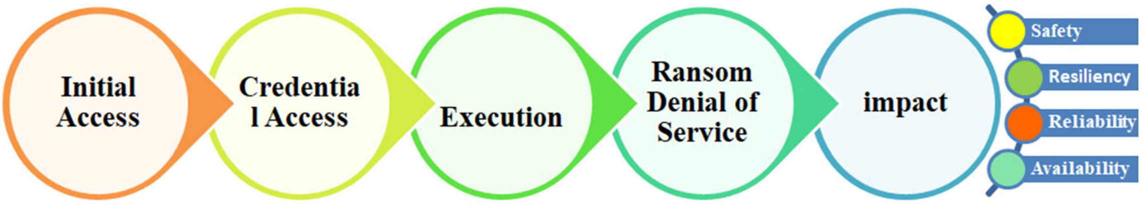


Fig. 5 RDoS attack stages

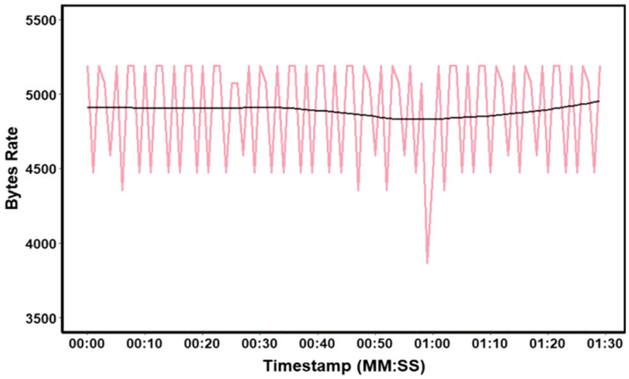


Fig. 6 Constant low rate RDoS attack byte rate

3.3.2 Burst or hit-and-run RDoS technique

Consider a technique of an attack that may be derived from burst or hit-and-run DoS attacks that utilize repeated short bursts of high volume traffic at random or deterministic intervals. This attack is common with TCP protocol as it exploits the TCP slow-timescale dynamic of the retransmission timer, the attackers send fast and high rate bursts having round trip time *(RTT)* burst length *(L)* and repeating periodically at retransmission time out *(RTO)* timescales *(T)* for bringing the server down. A successful burst TCP attack will have a large enough traffic rate for prompting packets loss, duration of *RTT (L=RTT)*, and period of *RTO (T=RTO)* [42]. The CoAP protocol operates over UDP, and it does not implement this TCP feature, but it supports retransmission mechanisms with exponential back-off for CON messages.

The CON request message requires ACK from the server and retransmits it many times before considering the request timeout state. Here, we adopt the same procedure of burst TCP for performing burst RDOS-CoAP attack. As the attacker only attempts to threaten and force the victim to pay a ransom rather than causing real service damage, we assume that the attacker sends burst high byte rate traffic as described in Fig. 7 only for few seconds *(L>RTT)* and repeating each deterministic period (T=RTO). The attacker sends periodic confirmable "GET" requests combined with a ransom note for approximately L=4 seconds. It repeats sending malicious requests each T=93 seconds (the default value for txThings request timeout) for roughly 3.5 min. This volumetric application layer DoS attack can disrupt the service if it is performed for a longer time.

3.3.3 RDDoS technique

Consider a technique of attack inspired by Memcached DDoS attacks [43] where attackers send out huge traffic directly from distributed devices to obtain a quick profit. We call this attack RDDoS as it represents Ransom Distributed Denial of Service (RDDoS). Here, the attacker procedure is sending the bulk of confirmable "GET" requests embedded with a ransom note to the available resource "/act" from multiple devices. This volumetric application layer attack can be harmful but for a short period to inflict pain to the victims and force them to pay the ransom to stop or avoid any potential longer and more harmful attack. The bytes rate of RDDoS attack for 90 seconds is depicted in Fig. 8.

4 Proposed intelligence-driven threat discovery model

Our proposed model is shown in Fig. 9 mainly consists of a real-time discovery engine represented by an online LSTM algorithm and the network-threat intelligence (i.e. IoC) that security analysts provide. Based on our observations in previous Sections, examples of such IoC are an unexpected increase in the number of network packets, the number of bytes, multiple network flows with small-time duration and others. This intelligence can be applied mathematically on the incoming network flows to generate the final network traffic information fitted to the discovery engine. The final

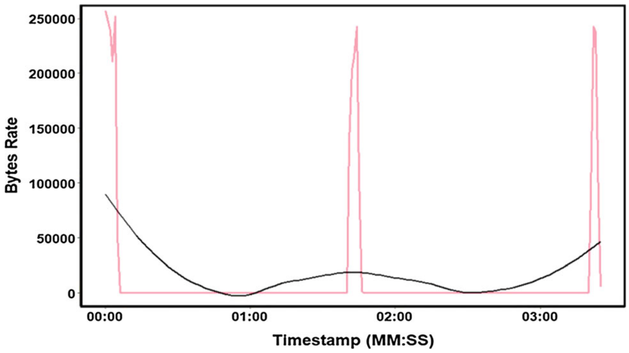


Fig. 7 Burst RDoS attack byte rate

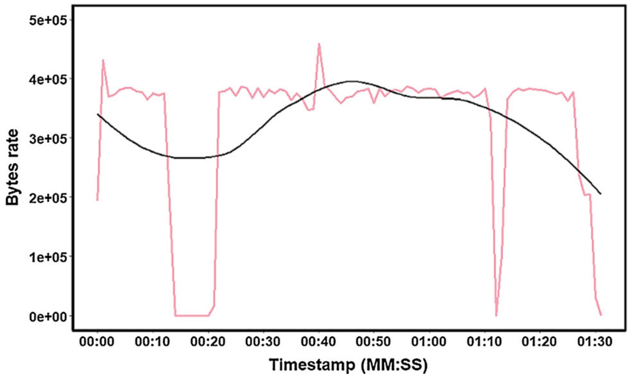


Fig. 8 RDDoS attack byte rate

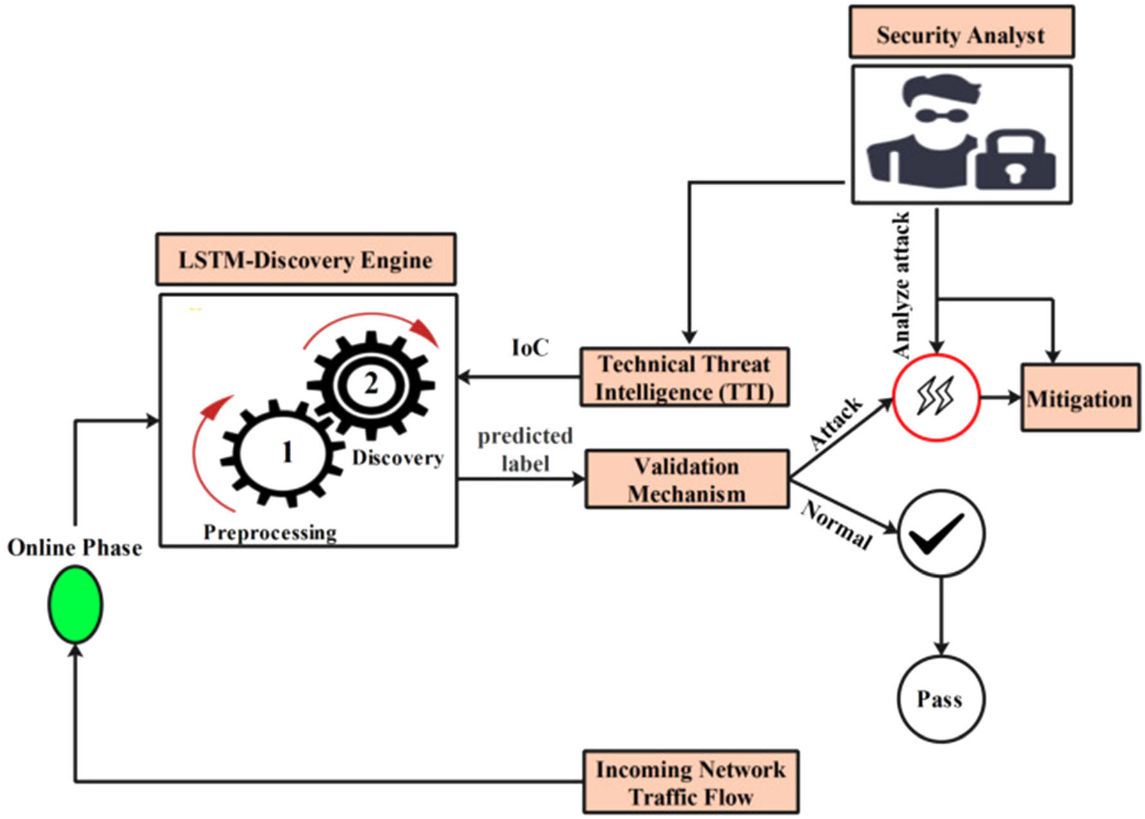


Fig. 9 Proposed real-time intelligence-driven RDoS attack detection model

network information could be data points related to network statistics where the combination of them can create RDoS attack patterns. Later, the discovery engine based on online deep learning can play a critical role in identifying the hidden patterns of RDoS attacks automatically.

The pre-processing engine is also used with the discovery engine to scale the provided information to have a zero mean and unit variance, which prevents the discovery engine bias in the learning process. The validation mechanism is used to evaluate the discovery performance if it is correctly or incorrectly identified the provided IIoT network traffic information. In this proposed model, the received observation (i.e. feature vectors) is categorized as normal or RDoS attack, allowing the security analyst in case of the identified attack and generated alarm to analyze RDoS attack, take the appropriate action or mitigation mechanism (e.g. redirecting the RDoS traffic to null route) and update the provided intelligence (if needed). At the update phase, the discovery engine-based deep learning is updated to learn from new input.

4.1 LSTM-based online discovery engine

Online discovery engine is a deep learning-based engine that searches potential threats against physical assets of SmartSat-IIoHT systems through analyzing the relevant collected data and intelligence. The process of data analysis is performed automatically using DL techniques to discover many elements of IoC for RDoS threat. The deep learning technique is used due to its high capability to identify complex network traffic patterns and deal with heterogeneous, unstructured and large volumes of data. It can extract the most relevant features in input data and without human intervention. DL-based models also have high scalability, and their performance is improved with increasing the training data size, as has been proved in many pre-existing studies [1, 13].

A long short term memory network (LSTM) is utilized in our work as the automatic and online RDoS-CoAP discovery engine. It was introduced by Hochreiter and Schmidhuber [44], and it is a type of recurrent neural network (RNN) that is specialized in reflecting the past learning into the current learning through penalizing weights through a chain of networks. Unlike other types of RNN, LSTM has a cell state and three gates. The cell state acts as conveyor belts that transfer relative information straight down the entire sequence chain. The information is got added or removed using three gates that are composed of a sigmoid neural net layer and point-wise multiplication operation. In this way, the gates decide to keep the most relevant information and forget the reminded one during the learning process. Given that, an LSTM-based discovery engine can keep the most relevant network data points related to RDoS-CoAP attacks.

Mathematically, LSTM takes a sequence of network data points (i.e. intelligence) as inputs, $(X = (X_1, X_2, \ldots..X_t))$ is carried over the timesteps $(t = (1, 2, 3, \ldots m)$ where (m) is the number of network data-points, and it gives output as a number defines the type traffic normal or RDoS-CoAP attack. As a first step, each LSTM block decides what information should be discarded and removed from cell state using "forget gate layer". It takes a network data-point (X_t), and the previous hidden state information (h_{t-1}) as inputs, pass them through sigmoid function (σ) to get output between 0 and 1. A "0" represents completely forget this information, while a "1" represents completely keep this information. The forget gate layer function is described in equation.1, where (w_f) , and (b_f) are the parameters (i.e. weight and bias) of forget gate layer (f).

$$Gate_{forget} = (f_t) = \sigma(w_f.[h_{t-1}, X_t] + b_f) \tag{1}$$

In the next step, the LSTM block decides which information should be stored in the cell state and this process goes through two parts. First, the input gate layer (i) decides which values will be updated using its parameters. i.e. (w_i) , and (b_i) and both inputs, i.e. (X_t) and (h_{t-1}). It also uses the sigmoid function (σ) to transform the values to be between "1" and "0" (important and not important consequently). Next, it passes the current input to the function (α) to create new candidates values between "-1" and "1" that could be added to the cell state (C_t) and help in regulating the network. Then, the output of input gate layer (i_t) is multiplied with the function output $(C_t^{'})$, and then, it is combined with forget gate layer output (f_t) and the old cell value (C_{t-1}) to update the old cell state with new one. The full process is described in Equations 3 and 4.

$$Gate_{input} = (i_t) = \sigma(w_i.[h_{t-1}, X_t] + b_i) \tag{2}$$

$$Cell_{candidate} = (C_t^{'}) = \alpha(w_c.[h_{t-1}, X_t] + b_c) \tag{3}$$

$$Cell_{current} = (C_t) = (f_t * C_{t-1} + i_t * C_t^{'}) \tag{4}$$

To decide the next hidden state which contains information on previous inputs, the sigmoid function (σ) is used to decide which parts of the cell state will be the output of the LSTM block. Then, the output of this step, as defined in Eq. 5, is passed to the tanh function (α). The output (O_t) is multiplied with the output of the function for cell state $\alpha(C\_t)$. Thus, the next hidden state will carry the desired information. This new cell state and the new hidden state are then passed to the next LSTM block (i.e. the next timestep). The same process from Eqs. 1–6 will be repeated for each timestep.

$$Gate_{output} = (O_t) = \sigma(w_o.[h_{t-1}, X_t] + b_o)) \tag{5}$$

$$New\_Hidden_{state} = (O_t * \alpha(C_t)) \tag{6}$$

LSTM repeats the same mathematical processes in the training phase in multiple timesteps based on the number of inputs. In addition, the output from the LSTM layers is passed to the output layer (with a sigmoid function) to determine the appropriate decision $(\hat{y})$ regarding the sequence of input (X). This is achieved by reducing the loss function value between the actual output (y) and predicted output $(\hat{y})$ for n observations using the following Equation.

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i}^{n}(\hat{y}^i - y^i)^2 \tag{7}$$

5 Experiments results and discussion

5.1 CoAP server resources and logs

We examine RDoS-CoAP behaviour in terms of edge server's CPU usage since application-layer DoS attacks usually affect the victim's CPU resources. We collect the total CPU time usage (i.e. CPU load) during a normal situation (no attack), burst RDoS, constant low rate RDoS and RDDoS attacks. Figure 10 shows that the total CPU usage during constant low rate RDoS, burst and RDDoS attacks is higher than the normal one. During constant low rate RDoS attack (at approximately 06:30-08:30), CPU usage goes significantly above 75%. While during the burst RDoS attack, the results show an increase in CPU usage over three times (06:30, 08:00 and 09:30), and the same behaviour can be observed during RDDoS attacks.

Another indicator of abnormal activity can be extracted from CoAP application logs. A sample of CoAP application log records is provided in Fig. 11. We found many records showing the CoAP server waiting for the next block-wise request timed out. CoAP performs a block-wise technique where a large payload is divided and sent in separate messages. After receiving the first block and based on the header, the client must send a new request for obtaining the second block.

5.2 Analysis of network traffic

Figure 12 shows the packet rate (number of packets/second) for RDoS, burst RDoS, RDDoS attack traffic (where their traffic is mixed with normal in real world), and the

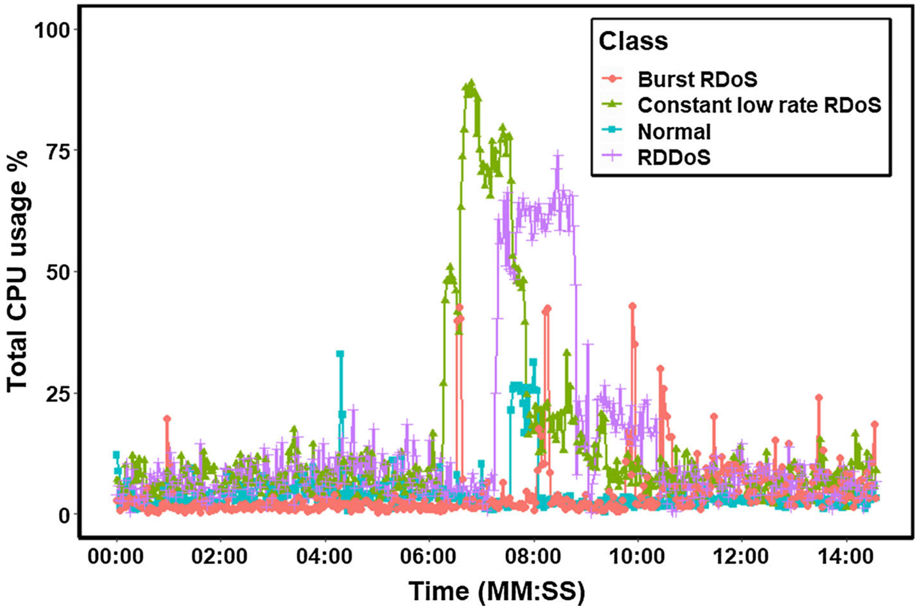
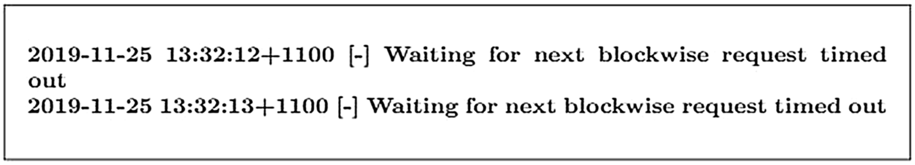


Fig. 10 Total CPU usage (load) for normal state and attacks

Fig. 11 Application logs



```
2019-11-25 13:32:12+1100 [-] Waiting for next blockwise request timed
out
2019-11-25 13:32:13+1100 [-] Waiting for next blockwise request timed out
```

pure normal traffic. Through a comparative analysis of Fig. 12, it can be noticed that the constant low rate RDoS attack traffic is almost hidden in the normal traffic, and the differences among them in terms of packet rate are quite small. While the burst attack appears in the shape of high peaks with roughly the same (L) length and interval (T), it can be seen that the burst RDoS attack has approximately deterministic characteristics even if it is mixed with normal traffic. This can be noticed in terms of packet rate and the timing relationship among multiple peaks. The RDDoS shows in the shape of high and relatively continuous waves for a while, and it can be seen that the packet rate is higher for a longer time. There is an unexpected increase in the network traffic in packets, which can be an indicator of RDoS attacks.

Figure 13 illustrates the distribution of received byte flows of the constant low rate RDoS, burst RDoS, and RDDoS attacks. It directly compares them with the received byte flows distribution during the normal situation. If the data points, in this a quantile-quantile (Q-Q) plot, are close to the diagonal line, it means that both data samples have the same distribution. This can be observed at the beginning of the line in three boxes where the distribution of attacks bytes is solely identical to the normal bytes because the attack traffic is mixed with normal. These bytes thus may not belong to attack bytes. The constant low rate RDoS distribution, as shown in the left-hand side of the figure, is approximately identical to the normal except for few outlier points at the middle, which represent the actual constant low rate RDoS bytes. In contrast, the distribution of burst and RDDoS attack bytes is spread out. These outlier points may represent the actual bytes of burst RDoS and RDDoS attacks. CoAP attack requests hold a ransom note; therefore, their number of bytes in each flow is higher than the normal ones.

We also inspect "GET" request packets based on the CoAP message for mat (see Sect. 2.3). The legitimate request packet of txThings Python script and CoapClient App is represented in Figs. 14 and 15, respectively, while the attacker's request packet is illustrated in Fig. 16. Here, we focus on the token length and option parts of the "GET" CoAP packet as they are the most distinguishable features. A randomized token number is used to match the request and the response by the server, and it is generated when the transport layer does not protect the CoAP message. Each client uses a randomized token number to connect to the Internet, but it is not mandatory. This can be noticed from CoapClient App (Fig. 15), which has zero bytes token length, while the txThings Python client (Fig. 14) has 4 bytes token with value 000021db. The attacker CoAP request (see Fig. 16) also has zero bytes to ken length. Arguably, there is no difference between

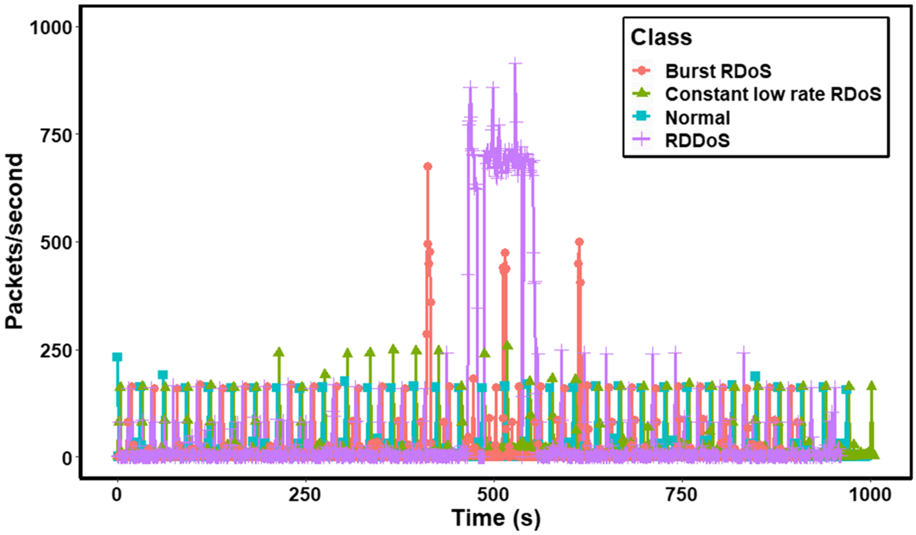


Fig. 12 Normal and mixed RDoS Packet rates

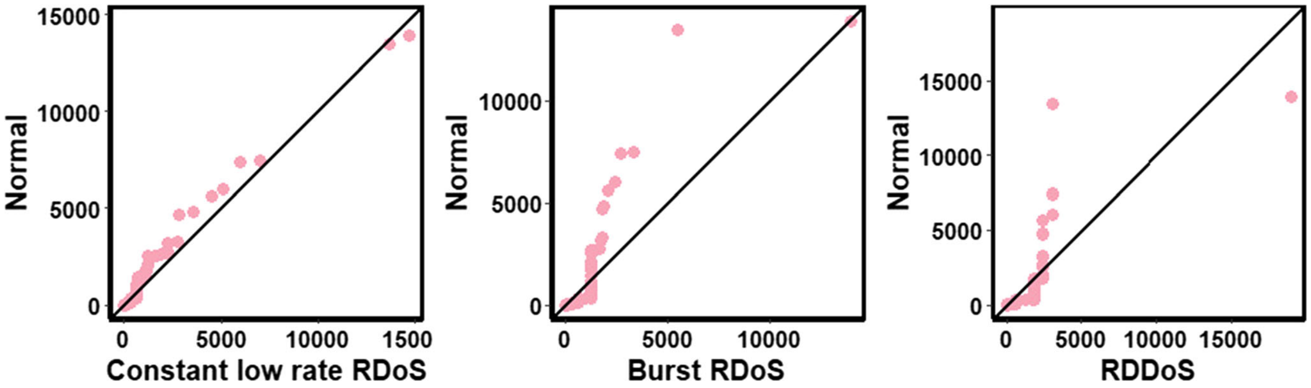


Fig. 13 Q–Q plots of RDoS attack compared with normal traffic

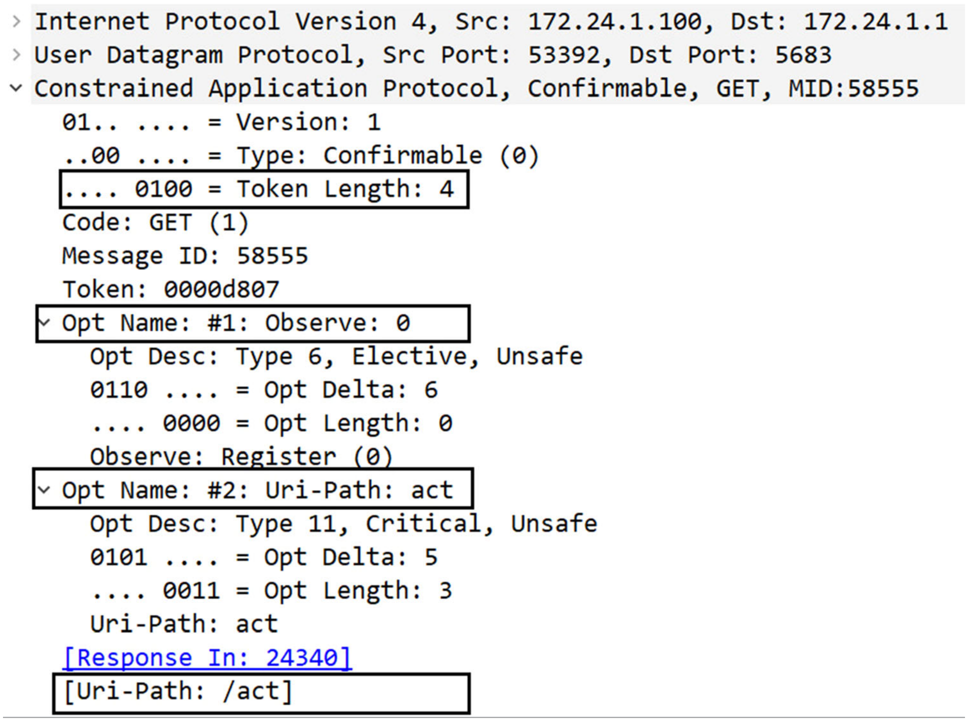
malicious and legitimate CoAP "GET" requests regarding token numbers.

CoAP also defines many options that can be combined in messages in which each has an options number, name, length, and value. For example, the Observe option is defined by RFC 7641 [45] and is issued as a GET message to register the client to the list of observers of act resources. TxThings Python client only uses this option here. The Uri-Path option is used to specify the path to the resources (i.e., /act). It can be observed from Figs. 14, 15, 16 that Uri-Path is included in all CoAP requests, while the payload option is only included in the attacks request as it holds the ransom note (see Fig. 16). The most interesting in this analysis that we found the CoAP server (txThings) accepted and decoded the GET message and responded without any error. Although CoAP standards define GET as a safe method and only for resource retrieval (here for reading sensor measurements), the attacker may exploit this optional part of the CoAP request for malicious action. It is worth noting that we obtained the same result with Aiocoap (one of the most recent and common servers) [46].

As CoAP performs a block-wise technique (see Fig. 17), we found that while analyzing network traffic, the attackers only receive the first block and do not send a request for the second block. This explains why the large number of next block-wise waiting time out records in the application log. This can be considered as a unique RDoS indicator, as it depends on the CoAP implementation. If the size of resource representation is less than the predefined block size (txThings default value is 64 bytes), the message will send as one block.

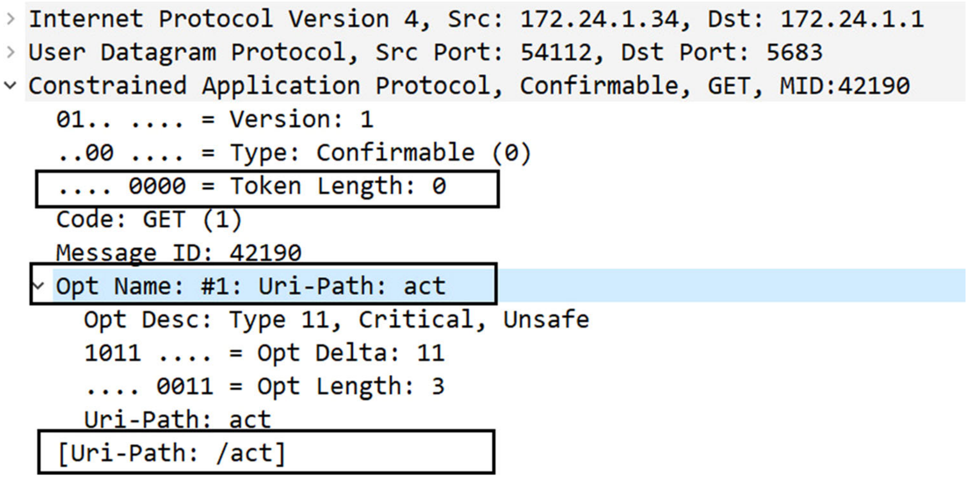
5.3 Physical asset behaviour and attack's impact

As the CoAP server polls data from the connected sensor, we consider this collected data to extract intelligence and study how the physical asset, i.e., connected sensor, behaves under RDoS attacks. Flooding the server with GET requests can affect the connected sensor response



```
> Internet Protocol Version 4, Src: 172.24.1.100, Dst: 172.24.1.1
> User Datagram Protocol, Src Port: 53392, Dst Port: 5683
v Constrained Application Protocol, Confirmable, GET, MID:58555
    01.. .... = Version: 1
    ..00 .... = Type: Confirmable (0)
    .... 0100 = Token Length: 4
    Code: GET (1)
    Message ID: 58555
    Token: 0000d807
  v Opt Name: #1: Observe: 0
      Opt Desc: Type 6, Elective, Unsafe
      0110 .... = Opt Delta: 6
      .... 0000 = Opt Length: 0
      Observe: Register (0)
  v Opt Name: #2: Uri-Path: act
      Opt Desc: Type 11, Critical, Unsafe
      0101 .... = Opt Delta: 5
      .... 0011 = Opt Length: 3
      Uri-Path: act
    [Response In: 24340]
    [Uri-Path: /act]
```

Fig. 14 Legitimate CoAP-txThings Python client

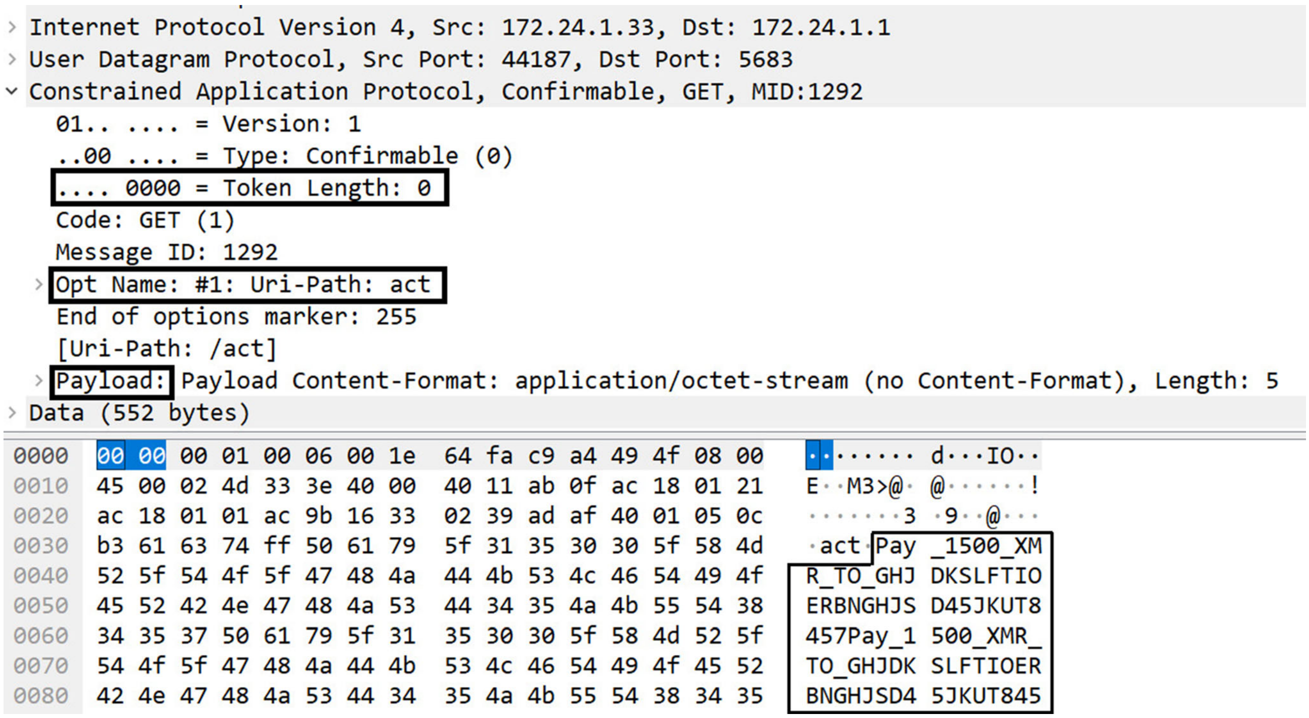


```
> Internet Protocol Version 4, Src: 172.24.1.34, Dst: 172.24.1.1
> User Datagram Protocol, Src Port: 54112, Dst Port: 5683
v Constrained Application Protocol, Confirmable, GET, MID:42190
    01.. .... = Version: 1
    ..00 .... = Type: Confirmable (0)
    .... 0000 = Token Length: 0
    Code: GET (1)
    Message ID: 42190
  v Opt Name: #1: Uri-Path: act
      Opt Desc: Type 11, Critical, Unsafe
      1011 .... = Opt Delta: 11
      .... 0011 = Opt Length: 3
      Uri-Path: act
    [Uri-Path: /act]
```

Fig. 15 Legitimate CoAPClient App

process. Fig. 18 shows the time difference (i.e., response time) between the read command by the edge server and the sensor response messages for the burst RDoS, constant low RDoS, and RDDoS attacks. As it can be observed, in the case of a normal situation (before and after the attack occurrence), the time difference or response time is roughly within the range of 0.001 and 0.00145 seconds. When the burst RDoS attack is in progress (13:27–13:29), the edge server received the measurements with a delay where the response time goes over 0.00145 seconds. The same results can be observed for constant low rate RDoS (10:15–10:17) and RDDoS (9:56–9:59). When the sensor is operating in a non-standard way (delay in response is greater than the predefined tolerance threshold), this indicates an attack in progress.

To provide a holistic attack analysis, we consider the response time to evaluate the systems availability, reliability, and resiliency under RDoS-CoAP. We define the availability as the degree of ability of the sensor to function at attack time, reliability as the degree of ability of the sensor to work regarding its specifications during attack time, and resiliency as the degree of ability of the sensor to complete its work during an attack and recover its performance after the attack. As shown in Fig. 18, there is no interruption on the sensors response (i.e., sensor availability) during three attacks, and it continuously responds

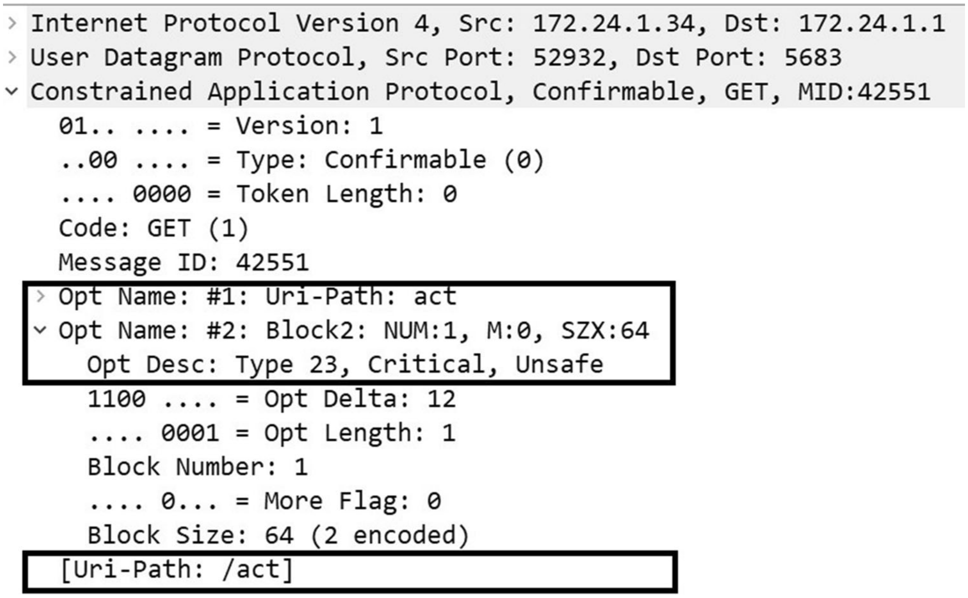


```
> Internet Protocol Version 4, Src: 172.24.1.33, Dst: 172.24.1.1
> User Datagram Protocol, Src Port: 44187, Dst Port: 5683
v Constrained Application Protocol, Confirmable, GET, MID:1292
    01.. .... = Version: 1
    ..00 .... = Type: Confirmable (0)
    .... 0000 = Token Length: 0
    Code: GET (1)
    Message ID: 1292
  > Opt Name: #1: Uri-Path: act
    End of options marker: 255
    [Uri-Path: /act]
  > Payload: Payload Content-Format: application/octet-stream (no Content-Format), Length: 5
> Data (552 bytes)

0000  00 00 00 01 00 06 00 1e  64 fa c9 a4 49 4f 08 00   ········ d···IO··
0010  45 00 02 4d 33 3e 40 00  40 11 ab 0f ac 18 01 21   E··M3>@· @······!
0020  ac 18 01 01 ac 9b 16 33  02 39 ad af 40 01 05 0c   ·······3 ·9··@···
0030  b3 61 63 74 ff 50 61 79  5f 31 35 30 30 5f 58 4d   ·act·Pay _1500_XM
0040  52 5f 54 4f 5f 47 48 4a  44 4b 53 4c 46 54 49 4f   R_TO_GHJ DKSLFTIO
0050  45 52 42 4e 47 48 4a 53  44 34 35 4a 4b 55 54 38   ERBNGHJS D45JKUT8
0060  34 35 37 50 61 79 5f 31  35 30 30 5f 58 4d 52 5f   457Pay_1 500_XMR_
0070  54 4f 5f 47 48 4a 44 4b  53 4c 46 54 49 4f 45 52   TO_GHJDK SLFTIOER
0080  42 4e 47 48 4a 53 44 34  35 4a 4b 55 54 38 34 35   BNGHJSD4 5JKUT845
```

Fig. 16 Malicious CoAP request packet

Fig. 17 Second block request



```
> Internet Protocol Version 4, Src: 172.24.1.34, Dst: 172.24.1.1
> User Datagram Protocol, Src Port: 52932, Dst Port: 5683
v Constrained Application Protocol, Confirmable, GET, MID:42551
    01.. .... = Version: 1
    ..00 .... = Type: Confirmable (0)
    .... 0000 = Token Length: 0
    Code: GET (1)
    Message ID: 42551
  > Opt Name: #1: Uri-Path: act
  v Opt Name: #2: Block2: NUM:1, M:0, SZX:64
      Opt Desc: Type 23, Critical, Unsafe
      1100 .... = Opt Delta: 12
      .... 0001 = Opt Length: 1
      Block Number: 1
      .... 0... = More Flag: 0
      Block Size: 64 (2 encoded)
    [Uri-Path: /act]
```

to the edge server. As there is a noticeable delay in the sensor response during attacks, RDDoS indicates that the sensor is not working accurately within the expected time (normal situation time). This violates the sensors specification, which poses an effect on the systems reliability. However, it can be noticed that the system is resilient against three attacks as the sensor continues in its work during attacks and the response time after attack periods (13:30, 10:18, and 10:00) recovers to the normal situation. Nevertheless, a simple delay at any element of the closed cyber-physical control loop may impact the overall safety of SmartSat-IIoHT systems, particularly if this delay exceeds the fault-tolerance threshold (safety system). The larger the attack, the larger damage that can happen, and the high likelihood of the ransom being paid.

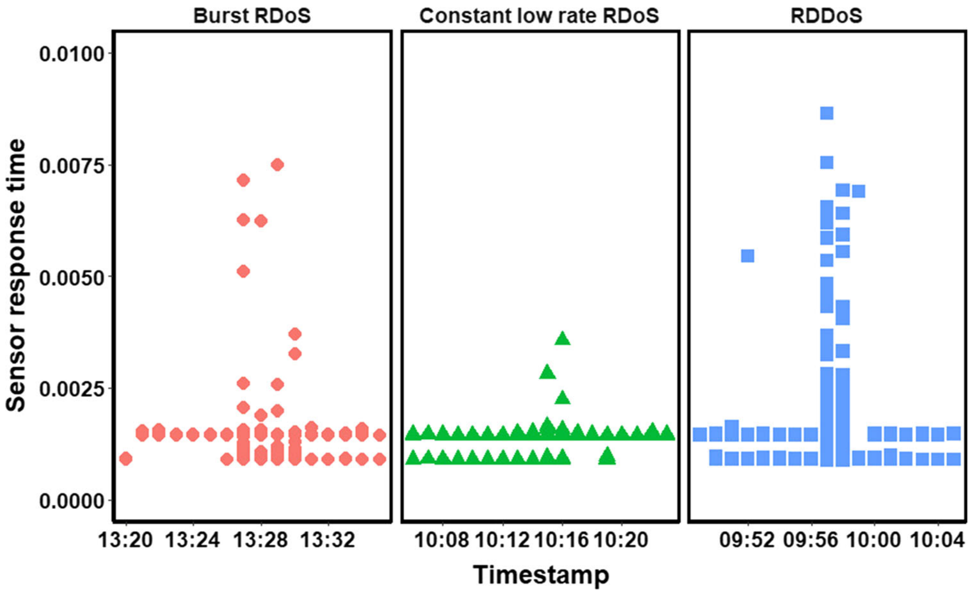


Fig. 18 Second block request packet

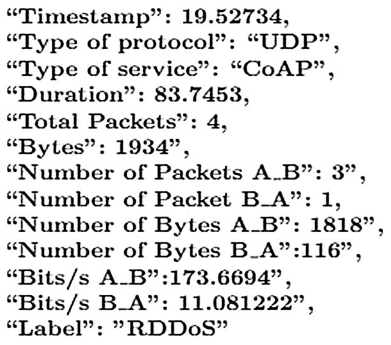

“Timestamp”: 19.52734,
“Type of protocol”: “UDP”,
“Type of service”: “CoAP”,
“Duration”: 83.7453,
“Total Packets”: 4,
“Bytes”: 1934”,
“Number of Packets A_B”: 3”,
“Number of Packet B_A”: 1,
“Number of Bytes A_B”: 1818”,
“Number of Bytes B_A”:116”,
“Bits/s A_B”:173.6694”,
“Bits/s B_A”: 11.081222”,
“Label”: ”RDDoS”

Fig. 19 Example of attack feature vector

6 Decisions and incident response

Once the attacker tactics, techniques and procedures are identified, those can be proactively used to secure SmartSat-IIoHT systems. Our proposed RDoS-CoAP tactics, techniques and procedures can help the SmartSat-IIoHT security team (i.e. IT and OT individuals) improve their contextual awareness about their systems. This can help them understand how their systems are likely targeted by RDoS-CoAP, what appropriate prevention and detection techniques are needed to implement. For example, when we consider a spoofed request (legitimate operator mobile device) as the primary vector for RDoS-CoAP attacks against the edge server, the security team can use this intelligence to prioritize this threat in their threat-centric

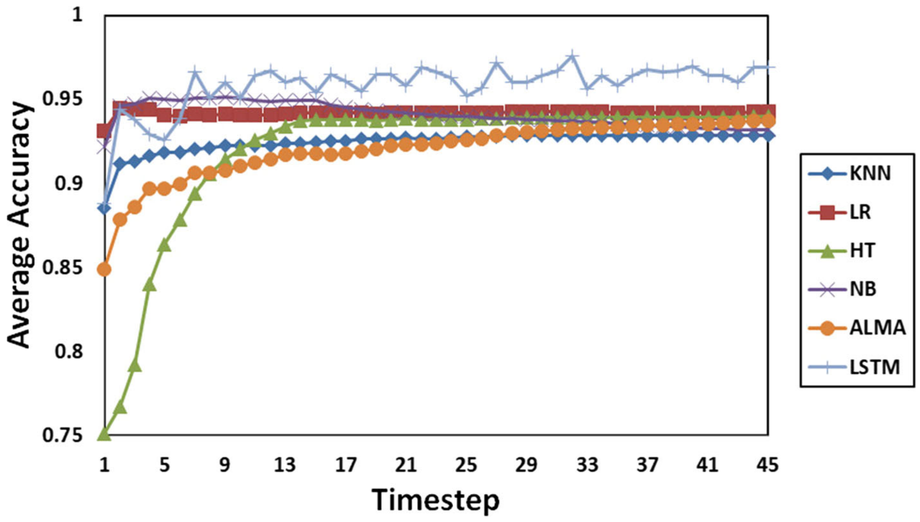


Fig. 20 Average accuracy over timestep

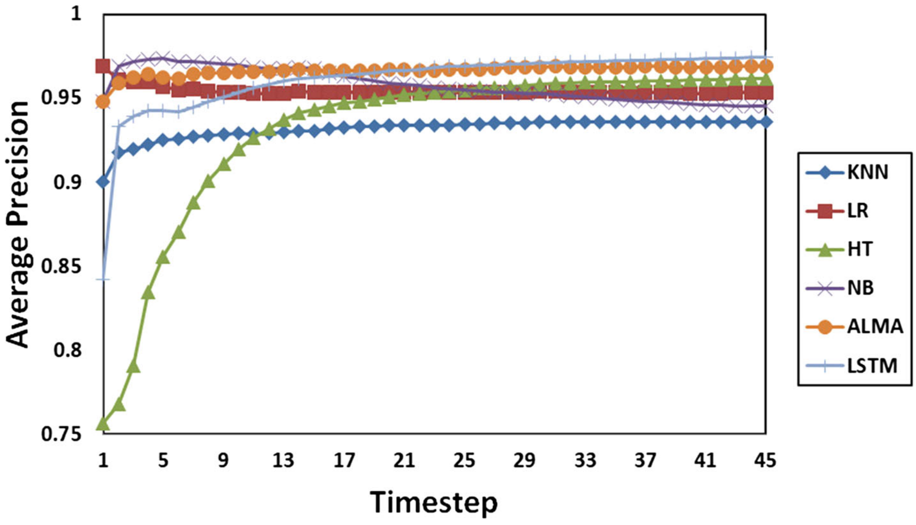


Fig. 21 Average precision over timestep

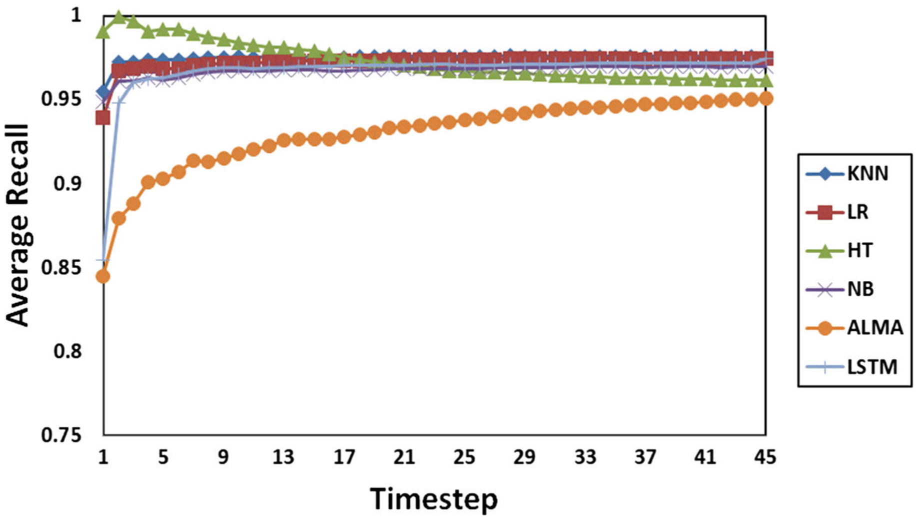


Fig. 22 Average recall over timestep

security approach. In its turn can help in performing an accurate risk assessment.

Given the provided intelligence related to techniques and procedures of implementing RDoS-CoAP attacks based on sending confirmable "GET" requests containing payload (ransom note) to the available resource, the security team can also prioritize specific defence mechanisms. For instance, abusing the optional payload of the "GET" CoAP request (tested against two common CoAP servers, see Sect. 4) indicates the lack of these optional parts' security. We can, therefore, state that Object Security for Constrained RESTful Environment (OSCORE) [47] has the highest priority in security implementation as it provides an end-to-end security protocol. It has been recently standardized by Internet Engineering Task Force (IETF) to protect message content format, payload, the request method and resources' identifier. OSCORE can prevent RDoS attacks by verifying the existence of an optional payload (legitimate "GET" request should not have a payload) and discarding the request that has it (i.e. RDoS-CoAP). It can easily be applied in SmartSat-IIoHT systems between edge servers and CoAP clients, and it may be complemented with DTLS and IPsec to strengthen the security approach.

We can also use the extracted intelligence as feeds to enrich security mechanisms to provide rapid detection and response against RDoS-CoAP attacks. The security team can use the Security Information and Event Management (SIEM) security tool to generate an alert if there is any match with any RDoS-CoAP indicators. For examples, a significant variation is seen from application logs analysis, i.e. an existing large number of waiting for the next block-wise request time out records, a sudden increase in CPU usage from host resource monitoring, the unexpected increase in the total network traffic (packets or bytes), "GET" request with payload (ransom note), and frequency

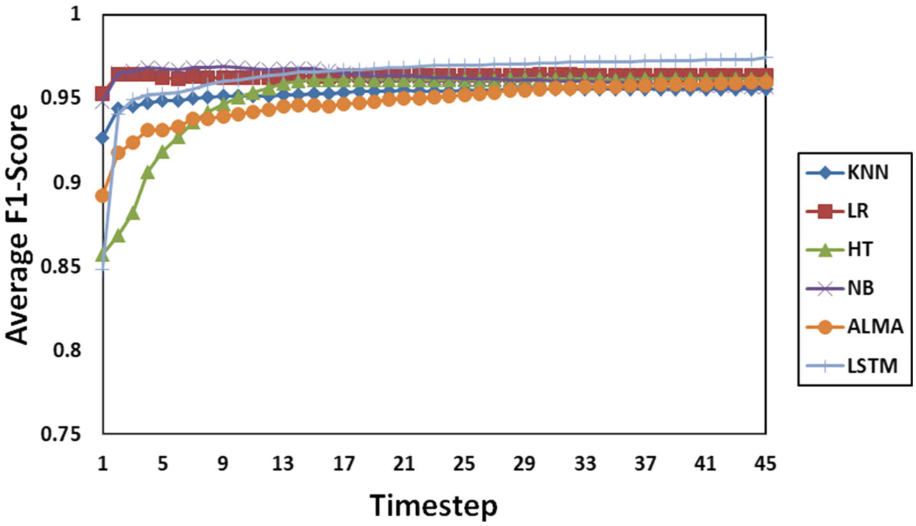


Fig. 23 Average f1-score over timestep

Table 1 Final performance metrics values

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
HT	93.99	96.18	96.17	96.17
LR	94.27	95.37	97.43	96.39
ALMA	93.75	96.91	95.08	95.99
KNN	92.87	93.63	**97.55**	95.55
NB	93.19	94.52	96.95	95.72
LSTM	**96.90**	**97.43**	97.18	**97.31**

Bold indicates the highest values

of CoAP requests in a short time from network traffic analysis or an increase in the sensor response time from data historian tags. Another ideal intelligence feed for SIEM can be achieved by looking at the set of all of the indicators mentioned above and correlating them to provide a larger RDoS-CoAP pattern. For instance, we can combine IT and OT events, such as a large number of CoAP “GET” requests with content containing payload within a short time and an increase in sensor response time.

We also noticed critical challenges with obtaining intelligence and information about RDoS-CoAP behaviour in SmartSat-IIoHT systems through our analysis. Chasing attack and threat indicators take a long time and need a deep knowledge of protocol specifications. Furthermore, the lack of standardization and consistency among IIoHT connectivity protocols’ contents (i.e. multi-platform protocol) creates also a challenge in extracting meaningful indicators. For example, as provided in Sect. 4, legitimate caregivers can connect via various application platforms, which offer different CoAP request’s content, particularly for optional parts (e.g. token length). Each system also has its CoAP protocol and server software configuration and requirements. For instance, a block-wise mechanism is an optional CoAP implementation, and it depends on the message size value configuration. Therefore, the significant number of records waiting for the next block-wise request time out records in application logs may be used as RDoS indicator for specific implementation and cannot be used as general intelligence. Furthermore, the SmartSat-IIoHT systems are fragmented, and their network traffic is a large dynamic volume and heterogonous. These challenges raise the importance of coming up with new solutions to get the benefits of TI efficiently and protect SmartSat-IIoHT systems.

New security solutions should consider SmartSat-IIoHT systems requirements such as scalability and interoperability [6, 33, 48]. In this regard, we propose a new security approach, that is, an intelligence-driven RDoS discovery model. This model can utilize network indicators irrespective of the CoAP protocol configuration and platform used (e.g. txThings, Aiocoap and CoAPApp mobile application), deal with the continuous evolution of attackers tactics, techniques and procedures and enable an automatic analysis for the provided intelligence and learning its significance. It can be integrated with any IIoT network irrespective of CoAP protocol configuration and platforms. It can also continuously learn the new provided intelligence and use it in detecting attacks, and it can thus scale and handle the growing size of TI. Moreover, the proposed model can rapidly detect and respond to attack early to prevent their spreading for the complete SmartSat-IIoHT system.

6.1 Discovery engine-based LSTM experiment results

In our model, LSTM has three hidden layers where each of which has 256, 128 and 24 neurons, respectively. RMSprop was used as an optimizer function to provide fast learning, and the mean square error was used as a loss function as LSTM predicts the label of new input data. The activation function is "Tanh", and the recurrent activation is "Sigmoid", while the batch size and epoch are equal to "1" as we used LSTM in the incremental mode. The final dataset has 10,314, 876, 4246 and 32,703 observations for normal, constant low rate RDoS, burst RDoS and RDDoS attacks, respectively. Each observation shown in Fig. 19 consists of a transport protocol name, the total number of transferred packets, the total number of received packets and the total number of received bytes. In addition to the total number of transferred bytes, total flow duration, type of service, received byte rate, transferred byte rate, and "Label" to identify the label of each observation. As it can be seen, all these features can be extracted from any SmartSat-IIoHT system. Furthermore, the SmartSat-IIoHT network and attack traffic have an unbalanced distribution/or number of observations, representing the realistic case of real-world data. We use the following metrics, i.e. accuracy, precision, recall (i.e. detection rate), and F-score, to evaluate the model's performance [49].

The results as shown in Figs. 20, 21, 22 and 23 describe our proposed model based on online LSTM performance and compared with the most common online machine learning algorithms such as logistic regression(LR) , Hoeffding Tree (HT) [50], Naive Base (NB), K-nearest neighbour(KNN) [51] and Approximate Large Margin Algorithm (ALMA) [52]. It can be seen that our proposed model can obtain a significant improvement in its performance over timesteps, where each timestep represents 1000 observations. Figure 20 shows that the average accuracy of algorithms increases with increasing the numbers of training observations. It can be observed that all algorithms at the first timestep predicted the label for the first 1000 observations with an average accuracy of more than 75%. However, our model's accuracy increases as long as it is trained using more data and reaches more than 95%. As shown in Figs. 21, 22and 23, our proposed model obtains the best performance in terms of precision and F1-score. Its performance continuously increases over timesteps to more than 95%. However, in terms of recall or detection/discovery rate, LSTM achieved approximately a similar performance to KNN and LR.

The same result can also be noted from Table 1, where the final performance metrics are calculated for the entire dataset. All algorithms achieved a significant performance in distinguishing between normal and RDoS attacks, while our model achieved the best performance in terms of final accuracy, precision and F1-Score. The LSTM obtained 96.90 %, 97.43% and 97.31%, respectively. However, in terms of final recall, the KNN achieved a better performance than LSTM, which obtained 97.55%. Overall, LSTM achieved considerable performance as it continuously updates the discovery engine with new incoming observations and improves its learning capabilities with the increasing size of training data. This proves the scalability of our model's deployments. It dynamically adjusts itself to the new incoming observation, trains the network using only incoming observation without storing data in the memory and improves its performance over time with the increasing number of training observations. These capabilities make our model is better than other online machine learning algorithms and an appropriate solution to SmartSat-IIoHT networks as such networks have a large volume of data that continuously increases over time.

Our model has many advantages that allow it to effectively detect and identify the new tactics and techniques of attackers, particularly RDoS attackers, and take the appropriate response. Firstly, our model learns the real-time dynamic behaviour of the SmartSat-IIoHT network traffic and continuously learns the new provided intelligence (i.e. IoC) and attacks' hidden patterns. Secondly, it can discover the knowledge and utilise the provided intelligence without fitting them in memory. The summary of this information can only be stored, which reduces resource utilisation. This makes our model highly scalable and memory-efficient and an appropriate solution for SmartSat-IIoHT systems due to their devices' resource-constrained nature and large traffic volume. Our model refines its performance over time in contrast to the static or off-line learning that progressively decreases its performance.

Our model mainly relies on deep learning techniques in developing a discovery engine. This technique provides our model with high capabilities to handle the large volume of SmarSat-IIoHT network traffic and the evolving attackers' tactics and techniques. Our model was built using LSTM, which can model long-term patterns from input data and the dependency among features. This makes it better than other deep learning techniques. Also, it has high robustness and generalisation capabilities as it can adapt and adjust itself to the dynamic and changeable traffic over time. Typically, LSTM requires high memory due to the presence of many memory cells. However, this problem has somehow been resolved with online learning as it updates its network based on one observation each time. Another challenge with our discovery engine-based LSTM is choosing the appropriate network depth before starting the online training process. Selecting a simple or complex model may lead to a restricted learning process or slow convergence. In our experiments, we chose a network with three hidden layers based on many trial-error experiments and the performance metrics during training. Our model started learning and converging after many observations and took longer than other online algorithms in the learning process. Still, it quickly converged and achieved a better performance than different algorithms during the first 1000 observations. However, a common way to resolve this limitation is by designing an adaptive LSTM that automatically changes its depth from shallow to deep. We will address this limitation in our future work.

7 Conclusion and future work

This paper has proposed and designed a threat intelligence modelling framework that can be implemented in SmartSat-IIoHT networks. It consists of multiple modules where these modules describe how the attackers can exploit CoAP protocol to perform RDoS attacks. Several attacker tactics, techniques and procedures for conducting volumetric RDOS attacks against the CoAP edge server are proposed. RDoS-CoAP behaviour, indicators and impact on server resources, network traffic and physical asset are also investigated. Moreover, how the proposed and extracted threat intelligence can prioritize and enrich defence mechanisms is discussed. We found that the lack of standardization in the SmartSat-IIoHT connectivity platforms can lead to provide specific intelligence through our experiments. This means indicators and intelligence are appropriate only for a particular platform and configuration. Thus, this raises the need for a generic security solution to protect the SmartSat-IIoHT system. In this regard, we proposed an intelligence-driven discovery model that depends on online LSTM. Our proposed model used network-based intelligence to discover RDoS attacks. The experiment results proved its efficiency in discovering RDoS attacks compared with other online machine learning algorithms.

In future work, we plan to investigate more complex RDoS techniques and procedures and their impact on the entire closed control loop (i.e. sensor, controller and actuator). We also plan to extend the threat intelligence modelling for other lightweight protocols such as MQTT and AMQT in healthcare systems. In addition, we plan to design an adaptive discovery engine-based LSTM model that chooses the appropriate LSTM network depth during the learning process. Another future direction is to investigate security events and produce threat intelligence for satellite communications.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. HaddadPajouh H, Khayami R, Dehghantanha A, Choo KKR, Parizi RM (2020) AI4SAFE-IoT: An AI-powered secure architecture for edge layer of Internet of things. Neural Comput Appl 32(20):16119
2. Patan R, Ghantasala GP, Sekaran R, Gupta D, Ramachandran M (2020) Smart healthcare and quality of service in IoT using grey filter convolutional based cyber physical system. Sustainable Cities Soc 59:102141
3. Tange K, De Donno M, Fafoutis X, Dragoni N (2020) A systematic survey of industrial internet of things security: requirements and fog computing opportunities. IEEE Commun Surv Tutorials 22(4):2489
4. AL-Hawawreh M, Sitnikova E (2020) Developing a security testbed for industrial internet of things. IEEE IOT J
5. Iglesias-Urkia M, Orive A, Urbieta A (2017) Analysis of CoAP implementations for industrial Internet of Things: a survey. Procedia Comput Sci 109:188
6. Mishra S, Paul A (2020) A critical analysis of attack detection schemes in IoT and open challenges. In: 2020 IEEE international conference on computing, power and communication technologies (GUCON) (IEEE, 2020), pp 57–62
7. Washiro T (2016) Electric RFID communication via human body. In: 2016 IEEE international conference on RFID technology and applications (RFID-TA) (IEEE, 2016), pp 129–132
8. Roselin AG, Nanda P, Nepal S, He X, Wright J (2019) Exploiting the remote server access support of CoAP protocol. IEEE Internet Things J 6(6):9338
9. Al-Hawawreh M, den Hartog F, Sitnikova E (2019) Targeted ransomware: a new cyber threat to edge system of brownfield industrial Internet of Things. IEEE Internet Things J 6(4):7137
10. Khalil K, Elgazzar K, Abdelgawad A, Bayoumi M (2020) A security approach for CoAP-based internet of things resource discovery. In: 2020 IEEE 6th world forum on internet of things (WF-IoT) (IEEE), pp 1–6
11. Asert. Coap attacks in the wild (2019). https://www.netscout.com/blog/asert/coap-attacks-wild
12. Zhou W, Jia Y, Peng A, Zhang Y, Liu P (2018) The effect of iot new features on security and privacy: new threats, existing solutions, and challenges yet to be solved. IEEE Internet Things J 6(2):1606
13. Al-Hawawreh M, Moustafa N, Garg S, Hossain MS (2020) deep learning-enabled threat intelligence scheme in the internet of things networks. In: IEEE transactions on network science and engineering
14. Bou-Harb E, Neshenko N (2020) Generating and sharing IoT-centric cyber threat intelligence. In: Cyber Threat Intelligence for the Internet of Things (Springer), pp 77–84
15. Montasari R, Carroll F, Macdonald S, Jahankhani H, Hosseinian-Far A, Daneshkhah A (2021) Application of artificial intelligence and machine learning in producing actionable cyber threat

intelligence. In: Digital forensic investigation of internet of things (IoT) devices (Springer), pp 47–64

16. Crest. what is cyber threat intelligence and how is it used? (2019). https://www.crest-approved.org/wp-content/uploads/CREST-Cyber-Threat-Intelligence.pdf
17. Zhang H, Yi Y, Wang J, Cao N, Duan Q (2019) Network attack prediction method based on threat intelligence for IoT. Multimedia Tools Appl 78(21):30257
18. Alladi T, Chamola V et al (2020) HARCI: a two-way authentication protocol for three entity healthcare IoT networks. IEEE J Sel Areas Commun
19. Routray SK, Hussein HM (2019) Satellite based IoT networks for emerging applications. arXiv preprint arXiv:1904.00520
20. Pradhan B, Bhattacharyya S, Pal K (2021) IoT-based applications in healthcare devices. J Healthcare Eng
21. Molling PE, Holst TT, Anderson BG, Fitzgerald K, Eddy M, Weber BD, Schwan B, Heiderscheit CJ, Jagim AR (2020) Drive-through satellite testing: an efficient precautionary method of screening patients for SARS-CoV-2 in a rural healthcare setting. J Primary Care Commun Health 11:2150132720947963
22. Williams JS (2003) Manufacturers move to help hospitals comply with joint commission requirements on clinical alarms. Biomed Instrum Technol 37(6):385
23. Hassija V, Chamola V, Bajpai BC, Zeadally S, et al (2020) Security issues in implantable medical devices: fact or fiction? Sustainable Cities and Society p. 102552
24. Tounsi W, Rais H (2018) A survey on technical threat intelligence in the age of sophisticated cyber attacks. Comput Secur 72:212
25. Brown R, Lee RM (2019) The evolution of cyber threat intelligence (CTI): 2019 SANS CTI survey. SANS Institute, Singapore
26. Yeboah-Ofori A, Islam S (2019) Cyber security threat modeling for supply chain organizational environments. Future internet 11(3):63
27. Díaz JEM (2020) Internet of things and distributed denial of service as risk factors in information security. In Bioethics (IntechOpen)
28. Figueroa-Lorenzo S, Añorga J, Arrizabalaga S (2020) A survey of IIoT protocols: a measure of vulnerability risk analysis based on cvss. ACM Comput Surv (CSUR) 53(2):1
29. Bhatt P, Morais A (2018) HADS: hybrid anomaly detection system for iot environments. In: 2018 international conference on internet of things, embedded systems and communications (IINTEC) (IEEE, 2018), pp 191–196
30. Granjal J, Pedroso A (2018) Intrusion detection and prevention with internet-integrated CoAP sensing applications. In: IoTBDS , pp 164–172
31. Kajwadkar VK Jain A (2018) novel algorithm for DoS and DDoS attack detection in internet of things. In: 2018 conference on information and communication technology (CICT) (IEEE, 2018), pp 1–4
32. Tiloca M, Hoglund R, Al Atiiq S (2018) Sardos: self-adaptive reaction against denial of service in the internet of things. In: 2018 fifth international conference on internet of things: systems, management and security (IEEE, 2018), pp 54–61
33. Vieira L, Santos L, Gonçalves R, Rabadão C (2019) Identifying attack signatures for the internet of things: an IP flow based approach. In: 2019 14th Iberian conference on information systems and technologies (CISTI) (IEEE, 2019), pp 1–7
34. Canuto L, Santos L, Vieira L, Gonçalves R, Rabadâo C (2019) CoAP flow signatures for the internet of things. In: 2019 14th Iberian conference on information systems and technologies (CISTI) (IEEE, 2019), pp 1–6
35. Bediya AK, Kumar R (2020) Real time DDoS intrusion detection and monitoring framework in 6LoWPAN for internet of things. In: 2020 IEEE international conference on computing, power and communication technologies (GUCON) (IEEE, 2020), pp 824–828
36. Yaqoob I, Salah K, Jayaraman R, Al-Hammadi Y (2021) Blockchain for healthcare data management: opportunities, challenges, and future recommendations. Neural Comput Appl pp. 1–16
37. Fernandez Maimo L, Huertas Celdran A, Perales Gomez AL, Garcia Clemente FJ, Weimer J, Lee I (2019) Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. Sensors 19(5):1114
38. Hathaliya JJ, Tanwar S (2020) An exhaustive survey on security and privacy issues in Healthcare 4.0. Comput Commun 153: 311
39. Wasilak M (2018) txthings. https://pypi.org/project/txThings
40. Liljedahl F (2019) Exploring the possibilities of robustness testing of coap implementations using evolutionary fuzzing
41. Alexander O, Belisle M, Steele J (2020) Mitre att&ck$\mathring{R}$ for industrial control systems: design and philosophy
42. Wu CC, Cheng RS, Hsu CW, Wu LW (2019) Lightweight, low-rate denial-of-service attack prevention and control program for IoT devices. J Internet Technol 20(3):877
43. Dahiya A, Gupta BB (2020) A QoS ensuring two-layered multi-attribute auction mechanism to mitigate DDoS attack. Mobile Netw Appl, pp 1–16
44. Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J (2016) LSTM: A search space odyssey. IEEE Trans Neural Netw Learn Syst 28(10):2222
45. Hartke K (2015) Observing resources in the constrained application protocol (CoAP), IETF RFC 7641
46. Maciej Wasilak CA The python coap library. https://aiocoap.readthedocs.io/en/latest/
47. Selander G, Mattsson J, Palombini F, Seitz L (2019) Object security for constrained restful environments (oscore). Work in Progress
48. Marques G, Pitarma R, Garcia NM, Pombo N (2019) Internet of things architectures, technologies, applications, challenges, and future directions for enhanced living environments and healthcare systems: a review. Electronics 8(10):1081
49. Jaber AN, Zolkipli MF, Shakir HA, Jassim MR (2017) Host based intrusion detection and prevention model against DDoS attack in cloud computing. In: International conference on P2P. Parallel, grid, cloud and internet computing (Springer), pp 241–252
50. Mirkhan M, Haeri MA, Meybodi MR (2019) Analytical split value calculation for numerical attributes in hoeffding trees with misclassification-based impurity. Ann Data Sci, pp 1–21
51. Al-Hawawreh MS (2017) SYN flood attack detection in cloud environment based on TCP/IP header statistical features. In: 2017 8th international conference on information technology (ICIT) (IEEE, 2017), pp 236–243
52. Qian C, Cai X, Zhu J, Xu Y, Tang Z, Li C (2019) Learning large margin support correlation filter for visual tracking. J Electron Imag 28(3):033024

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.