Noname manuscript No.
(will be inserted by the editor)

A decentralized honeypot for IoT Protocols based on Android devices

Irini Lygerou · Shreyas Srinivasa · Emmanouil Vasilomanolakis · George Stergiopoulos · Dimitris Gritzalis

Received: date / Accepted: date

Abstract The exponential growth of internet connected devices in this past year, has led to a significant increase in IoT targeted attacks. The lack of proper integration of security in IoT development life-cycle along with a plethora of different protocols (e.g. Zigbee, LoRa, MQTT, etc.) have greatly impacted the resilience of such devices against cyber-attacks, a fact also exacerbated by the size and physical hardware structure of these devices. Thus, it is imperative to develop effective and efficient countermeasures that can also be applied post-production to help build resilience in modern IoT systems. Honeypots are prime example of this notion. Being designed to act as vulnerable computer components or systems, they provide useful intelligence regarding potential attackers. Nevertheless, honeypots have seen little use in protection IoT systems and their underlying protocols, especially in cases where honeypots can leverage the decentralized nature of IoT. In this research, we enhance the HosTaGe honeypot to build an IoT protocol honeypot that runs over mobile devices. The purpose of this paper is to introduce a honeypot specifically for IoT communication protocols over public networks that is easy-to-use and utilizes Android-devices. The protocol honeypot utilizes the cellular network to establish decentralized, simulated infrastructures of IoT systems over different types of IoT network protocols. We test four IoT network implementations, one for each of the newly implemented MQTT, CoAP and AMQP protocols. Additionally, we upgrade existing Telnet and SSH protocols used in IoT systems to work over the simulated mobile honeypot. We use the virtualized honeypot networks to capture log, and analyze real-world public attacks on these protocols from the internet and provide an interface for interaction with the implemented honeypot.

Keywords Honeypots · IoT · Android

Irini Lygerou, Dimitris Gritzalis
Department of Informatics
Athens University of Economics and Business
Athens, Greece
E-mail: irene irini.lyg@gmail.com, dgrit@aueb.gr

Shreyas Srinivsa, Emmanouil Vasilomanolakis
Aalborg University, Denmark
E-mail: shsr@es.aau.dk, emv@es.aau.dk

George Stergiopoulos
Department of Info. and Comm. Systems Engineering
University of the Aegean, Samos, Greece
E-mail: g.stergiopoulos@aegean.gr

1 Introduction

Internet of Things (IoT) devices primarily aim to provide utility rather than complex networking and access control. These low-power, specific-use devices have been introduced into critical systems and processes, amplifying both opportunities and challenges across different industries including agriculture, energy, health, and transport [43, 24]. The amount of IoT devices is constantly growing at a global scale, along with associated cyber threats. In 2020 alone, millions of attacks have targeted such devices [42]. This obviously increases the attack surface for malicious actors and the need for more security measures [1], but also provides a novel opportunity for security practitioners: If a lightweight honeypot could be supported by IoT devices alongside their actual software, the sheer amount of such embedded systems could establish a major decentralized honeypot for detection of cyber-threats.

A honeypot's sole purpose is to be discovered and attacked by potential adversaries. Users interacting with such systems are automatically considered suspicious

and their activity is logged and monitored as part of security intelligence. Honeypots seem to be a general-use method to provide early detection of possible cyber attacks. In Kaspersky's 2019 report, deployed Telnet honeypots detected over 105M attacks in the first half of the year. This is a 775% increase from the 12M recorded attacks in the first half of the previous year [17].

These systems have been known for their utility in traditional IT and industrial systems [12,30,50]. Regarding IoT devices, however, their use is rather limited. Even worse and to our knowledge, there are currently no honeypots able to build and simulate multi-system IoT protocols over large geographic regions with an easy-to-install and stable framework to capture attacks.

Existing approaches in literature provide either complicated solutions to install honeypots over long-distance, or do not exist at all as regards multi-LAN IoT networks over specific IoT protocols such as MQTT, Lora or ZigBee. IoT protocol simulation implementations are either trivial simulations of which lack the necessary real-world functionality, or opt for proprietary-use tools aimed at product design [16] and application testing [11] [9] and not on security. In this work, we provide a way to fully simulate multiple IoT Protocols for security implementations on an open-ended, decentralized system of interconnected mobile phones over typical 4G/LTE connections. Specifically we build a honeypot over MQTT, CoAP and AMQP protocols and embed high-level, deceiving interactions within the deployed protocol simulations without restricting the honeypot implementation into a single LAN or limited network section. The unique trait of this honeypot is its ability to fully simulate popular IoT protocols while maintaining its decentralization and mobility, utilizing everyday cellular and wireless infrastructure. This enables scalability and interoperability since any mobile device can act as a honeypot agent within a predefined VLAN. To achieve this, we use an existing honeypot (HosTaGe) and adapt it to turn mobile phones into virtual IoT devices running over modern IoT protocols.

This implementation leverages API simulations to contextualize interactions of real IoT networks over the aforementioned IoT protocols. We utilize existing, off-the-shelf mobile phones that mimic IoT devices parallel to their everyday use and evaluate results from a public honeypot implementation. Experiments emulate four different IoT systems. Live mobile devices on the Android OS environment realize various IoT protocol honeypots, such as the MQTT, AMQP, CoAP, along with typical protocols such as SSH and TELNET. For example, the MQTT honeypot realizes (1) an MQTT Broker, (2) an MQTT Temperature Sensor, (3) an ESP8266 Smoke Sensor and an (4) an Arduino system.

To aid in interoperability, the honeypot app does not run directly in the Android OS. Instead, we utilize an additional layer along with multiple modules that can co-exist in the same mobile environment interchangeably. This allows for the creation of different virtual IoT honeypots, that can be swapped on-the-fly inside the same device.

To test the efficiency and effectiveness of each IoT protocol honeypot, we deploy multiple mobile devices as honeypots and capture real-world attacks aiming internet enabled IoT devices. Our honeypot logs, stores and processes all captured information in a UI or on other formats (e.g., XML, JSON). We perform a statistical analysis to extract some meaningful results, such as common attack IoT patterns and current attackers' origin per type of attack. Our implementation supports both cellular and WIFI networks. The main aim and contribution of this paper is a virtual mobile honeypot with IoT emulated systems on an Android OS, with interchangeable IoT devices.

The remainder of this paper is structured as follows. Section 2 discusses Related Work. Section 3 presents an analytic view of our Honeypot design. In Section 4 we introduce our methodology and results. Finally, Section 5 concludes the paper.

2 Related Work

Common IoT system architectures are usually divided into three layers (see Figure 1) [36]: the Object (Perception) Layer, Network Layer and Application Layer [11]. There are dedicated protocols for each layer of the architecture. The Object Layer is responsible for the information gathering from physical devices such as sensors and actuators. Perception Layer protocols include Zigbee, EPCglobal, IEEE 802.15.4, Z-Wave and BLE. The next layer represents the Network Layer which connects to the IoT network and advances information across. The routing and addressing of protocols provide a reliable connection in the network and allows the identification of each device. The Application Layer provides services to end users. It includes various protocols, such as MQTT, AMQP, CoAP, XMPP and DDS.

Additionally, honeypots, are classified based on physical characteristics, their interaction, deployment, and level of simulation. A physical honeypot is a real machine on a network with an assigned IP address. A virtual honeypot is a simulated machine that emulates a vulnerable system. Based on their interaction, honeypots can be either *high* (usually a real system [38]),

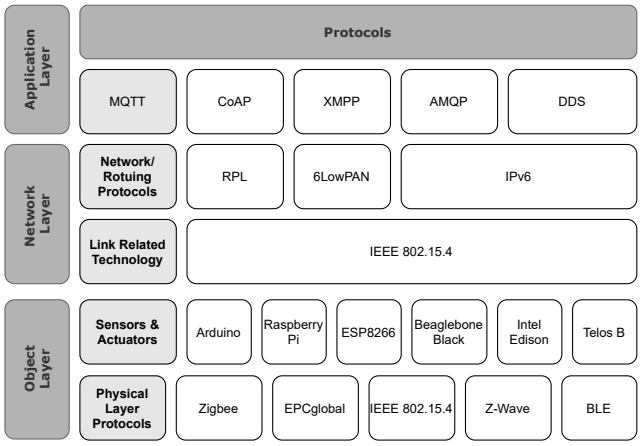

Fig. 1: IoT protocols per Operating Layers Table

medium/low (offering respective interaction capabilities usually constrained on the network layer), or *hybrid* [40].

2.1 Relevant Honeypots and implementations

HoneyThing [19] is a honeypot for Internet of TR-069 things. It is designed to act as modem/router that has RomPager embedded web server and supports TR-069 (CWMP) protocol. Cowrie [17] is a medium to high interaction honeypot that simulates the Telnet and the SSH protocol and logs brute force attacks and the shell interaction in a *JSON* format. Cowrie provides an emulated file system in python in a high interaction mode (proxy) with SSH and Telnet functionality. Dionaea [18] is a low interaction honeypot that simulates multiple protocols including HTTP, MYSQL, SIP, UPnP SMB, MSSQL, FTP, and MQTT. This honeypot logs all the record attacks in *JSON* format or in an SQLite database the main objective of this honeypot is to capture malware and keep a copy of it for analysis.

Kako's [3] default configuration will run several service simulations to capture attacking information from all incoming requests, the full body, the process attempts including the payload. It supports the functionality for Telnet, HTTP and HTTPS servers. IoTPOT [33] is a low interaction honeypot to emulate Telnet services of various IoT devices. It is integrated with IoTBOX a high interaction backend virtual Linux sandbox environment for common embedded IoT Telnet systems utilizing different types of CPU architectures. It establishes a Telnet connection with a backend IoTBOX and forwards the command. IoTPOT revealed different types for Telnet targeting attacks and botnet networks.

IoTCandyJar [31] is a novel intelligent interaction honeypot. The honeypot optimizes itself to send the appropriate request for the attacker. ThingPot [49] is a medium/hybrid interaction honeypot that includes XMPP and MQTT protocols emulating IoT devices with a REST API. As a Proof-of-Concept, a Philips Hue system, containing two smart bulbs being connected to one Phue bridge, was simulated and collected numerous attacks.

The Multi-purpose IoT honeypot was inspired from the IoTPOT and was classified as high interaction. The honeypot had two main components, being the frontend and the backend. The frontend included 4 proxies for each protocol simulation: SSH, Telnet, HTTP, CWMP. The backend was based on existing custom modified docker containers for the honeypot implementation [29]. U-PoT is a novel honeypot framework for UPnP based IoT devices. It automatically creates an emulated UPnP device based on a description file. It emulates a smart switch that uses UPnP protocol for communication [23].

Multi-faceted [44] is an IoT honeypot ecosystem with three distinct services: First, the honeypot server farms with different honeypots deployed on premise and cloud instances. Second, a vetting system including Shodan and Census, and third an analysis infrastructure is used to collect and analyze the captured data using fingerprinting techniques. Existing honeypots were used to set up the Ecosystem, along with a custom low-interaction honeypot for IoT cameras called HoneyCamera. IoTCMal is a hybrid honeypot [47] consisting of both high and low components. The low component includes the Telnet and SSH services, while the high component uses IoT device services with common vulnerabilities.

Authors in [48] presented a medium-interaction honeypot with corresponding vulnerability attack surfaces . The simulation module is developed to obtain the trust of attackers. In general, we present three honeypots to simulate different vulnerable IoT-related targets with fidelity maintaining strategies. The honeypot's primary aim is to to efficiently distinguish botnet attacks from legitimate traffic and perform attack pattern analysis. Contrary to the presented work, this honeypot does not focus on IoT protocols, nor on the ad-hoc modeling of simulated honeypots over mobile devices. Instead, authors focus on diversifying botnet detection through three types of honeypots: (i) Protocol-based, which mostly included common IT protocols, i.e. Open network video interface forum (ONVIF), Network time protocol (NTP), SSH/Telnet; (ii) Vulnerability-based, modeling attacks based on CVE-2013-6117 (DHDVR) and the Netis backdoor of 53413 port; and (iii) Application-based where authors modeled a honeypot for the Domain name system (DNS), OpenVPN and relevant apps. This honeypot follows a similar modular logic for de-

tecting attacks as the presented work, with multiple protocols and surfaces but still has a different focus as regards the IoT protocol focus and the modular, ad-hoc modeling of the presented work.

Authors in [7] present an overview of the the MQTT and CoAP IoT protocols from a Smart Healthcare perspective. Authors utilize multiple characteristics and appropriate simulator tools to test, analyze, and validate preexisting concepts concerning these protocols. Also, they perform relevant research based on interdependent criteria and different simulation tools to have an in-depth understanding of their pros and cons. Although some of the presented tools can be effectively used to simulate MQTT and COAP protocols, still this work aims to compare and analyze two specific protocols in terms of performance. Relevant tools either focus on specific protocols or have a simulation-first approach that aims to follow up on real-world implementation test. Instead, our approach has a modular design and an ad-hoc, plug-n-play setup along with different attributes to support security analysis of attacks. Specifically, we support a wide range of IoT protocols simultaneously and focus on the ease-of-use along with the ability to gather data specifically for security insights rather than performance.

2.2 ICS/OT Honeypots

Numerous publications and gray literature exists concerning honeypots on ICS and industrial applications. These application often engage in IoT protocols alongside industrial protocols like Modbus, DNP-3 etc.. To this end, some key research regarding ICS/OT honeypots is presented in this section.

Trend Micro Research presented a production honeypot that mimicked a real system, including a human-machine interface (HMI) and other components of an ICS [25]. They ran four PLCs from three different brands: one Siemens S7-1200, two AllenBradley MicroLogix 1100 units, and one Omron CP1 combined with OSINT knowledge to detect and block certain attacks in ICS.

In [8], authors presented MimePot, a Model-based Honeypot for Industrial Control Networks. MimePot is able to simulate physical industrial processes and uses Software Defined Networks (SDNs) to create honeynets and traps for attackes. Authors demonstrated how MimePot captures data integrity attacks against a water distribution system in a simulated environment.

Work in [28] presents the use of a high-interaction honeypot that simulates a grid named GridPot. Using GridPot, authors identified and characterized threats to electric-power grids by analysis of traffic in a simulated grid. They deployed a network-traffic packet capture utility to analyze honeypot-activity logs and demonstrate the effectiveness of our honeypot at collecting intelligence for threat analysis. Simulated network protocols supported included HTTP , MODBUS, S7COMM, SNMP and IEC 61850.

Lastly, authors of [34] combined ICS and IoT protocols to build an approach that utilizes network activity from both types of systems, in an effort to analyze modern ICS connected to the Internet, in the context of their digitalization as a part of the Internet of Things (IoT) domain. Authors implemented an interactive, proof-of concept ICS honeypot, based on Conpot, able to emulate a physical ICS device, by replicating realistic traffic from the real device and study traffic aimed to be used in the context of both ICS and IoT systems. Authors tested their approach using a real-life demonstration scenario involving a hydro power plant.

2.3 Summary and orientation

Existing mobile honeypots include various smart devices such as tablets, smart phones or even laptops [26] but, to our knowledge, none. In our paper, we focus on smart phones with an Android Operating system. Smart phones nowadays have high computational power and are almost always connected to the internet. They are not commonly recommended for high interaction honeypots, due to their limited resources and battery life [4]. In a virtual honeypot deployment, one physical device can host several virtual machines or emulated systems that act as honeypots. It is designed to resemble an authentic network.

This work extends previous design and implementation of hybrid-interaction IoT/OT honeypots [40]. In [40], authors presented RIoTPot, which was implemented as a preliminary design of a modular IoT honeypot with low-interaction and high-interaction modules. RIoTPot supported an early version of Telnet, SSH, CoAP, Modbus and MQTT protocols, with an emphasis on low-interaction, scalable VM implementations but lacked the robust, real-world simulation of the work presented in this paper, had no mobile plug-n-play support and did not introduce any novelty in designing IoT protocol simulations over ad-hoc networks.

3 The IoT Honeypot Model

3.1 Design

The presented work builds on the HosTaGe honeypot, originally released in 2014 [46]. The original HosTaGe

implementation supported multiple systems simulations including industrial systems. Examples of these include Windows HMIs, Linux server machines, Web Servers, Modbus masters etc. In this work, we opt for a low interaction honeypot to minimize dependencies and power usage. High interaction honeypots are more complex and more difficult to deploy on limited resources. The proposed design takes into consideration the limited resources of mobile phones and facilitates a smooth migration or scalability. Furthermore, its layered approach allows us to install multiple services and emulate different systems in the same mobile app. There is no need for the systems to be fully implemented.

For the purposes of this research, we had to update the framework to support the features necessary for emulating IoT protocols. The updated architecture improves HosTaGe with added IoT functionality. New Features were added, consisting of the new greenDAO ORM, IoT protocol simulation, HPFeeds Integration, JSON and Pcap Export, 4g compatibility, un-rooted phones capturing.

GreenDAO is an object-oriented database for Android with higher performance benchmarks than most of the existing alternatives [22]. In our app, greenDAO was uti-lized for accelerating the performance of queries inside the app and helping with the pagination of the records eliminating performance issues in record retrieval.

HPFeeds is a lightweight publish-subscribe protocol that transfers different for-mats of data in authenticated users. Different feeds are separated in channels with a unique username and password. With respect to the HosTaGe Architecture, the protocol is used to transfer attack records in *JSON* format. An HPFeeds broker is installed on a server machine, including the MongoDB database, in which the published records are saved. An HPFeeds subscriber listens to published events and then redirects them in a local or remote database. Multiple subscribers can follow the events in the same channel and store them, as long as they know the secret (password) and ident (username) of the channel. The advantage of HPFeeds is portrayed on the ability of multiple devices (clients) to publish the attack records and store them in the same central database. The diagram below depicts the HPFeeds Architecture.

The advantage of the updated "HosTaGe" app is its interchangeable nature. It contains multiple systems, and therefore different type of honeypots. The goal of these improvements is to add the IoT honeypot type with new emulated IoT systems.

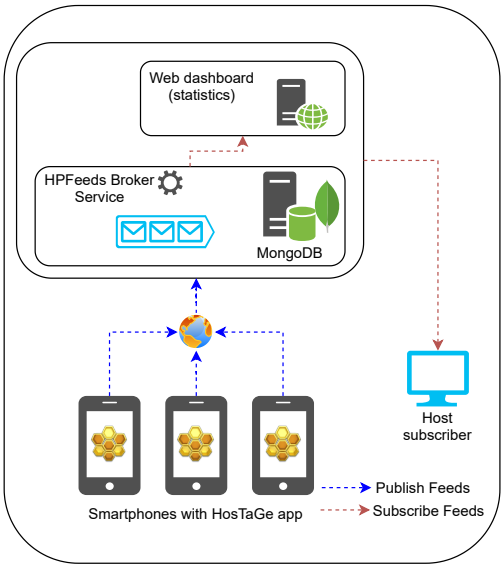

Fig. 2: HosTaGe HPFeeds Architecture

3.2 Protocols Simulation

Two IoT protocols already existed in the previous implementation, TELNET and SSH, but they were not sufficient to support for the new IoT systems simulation. The protocols selected for our new design are the MQTT, CoAP and AMQP. The reasons for the selection of the previous protocols are their light implementation which make them suitable for a low-interaction honeypot and their popularity in IoT devices.

3.2.1 MQTT

This is an OASIS standard messaging protocol for the Internet of Things. It is designed as a lightweight publish-subscribe messaging transport that is ideal for connecting remote devices with limited resources. Most common attack for this protocol is the Denial of Service [32].

3.2.2 CoAP

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and networks on the Internet of Things. The protocol is designed for machine-to-machine (M2M) applications where the resources are scarce [31]. The CoAP protocol is vulnerable to IP spoofing which makes it susceptible to DDoS attacks [13].

3.2.3 AMQP

AMQP stands for Advanced Message Queuing Protocol and it is an open standard application layer protocol

[15]. Despite using TLS/SSL encryption sophisticated attacks could decipher the protocol's communication. It is also vulnerable in DoS attacks due to insufficient rate-limiting protection.

3.2.4 SSH

The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another [41]. SSH is not an IoT protocol per se, but many IoT devices use it for secure transfers. SSH though is vulnerable to MitM (Man-in-the-Middle-attacks) and other vulnerabilities. ESET IoT Research team has found numerous security vulnerabilities in three different home hubs, one of them creating a working exploit opened an SSH backdoor [21].

3.2.5 Telnet

This is a protocol that provides a command line interface for bidirectional remote communication. Telnet protocol has an easy implementation but it's not a trusted protocol and is recommended to be used over SSH. But users do not follow this recommendation and as result this year over 500,000 Credentials for Telnet Exposed IoT Devices [42].

3.3 Cellular Network Connectivity

The majority of IoT devices are being served by radio technologies designed for short-range connectivity with limited Quality of Service and security requirements for a vast spectrum of applications. Cellular technologies 3GPP technologies like GSM, WCDMA, LTE and 5G NR provide high quality mobile data services [20].

With respect to the HosTaGe design, the added 3GPP compatibility creates new vector surfaces for IoT attack discovery. The 3G and 4G intranet is dynamic with non-trustworthy neighbours. This attack vector openness allows the attacker to scan the network and exploit detected vulnerabilities [35]. In the presented honeypot, 3G/4G connections are detected automatically, when the user turns on the virtualization app and all available systems are simulated in the underlying network. This does not require any adjustments of the main app's functionality. The honeypot utilizes the cellular networks as an additional "vector" for mobile networks, and possibly attracting different types of attacks when enabled. The only difference is how the network is dis-played on the corresponding network interfaces. Since the SSID (service set identifier) is not applicable in cellular networks, the type of cellular network is displayed instead.

3.4 Implementation

Improvements to the previous HosTaGe implementation [46] refer to updates, deprecated code removals, and new feature addition, along with a re-launch on the Google Play Store. Existing unrelated features such as multistage, synchronization Tracing, Bro signature generation and old deprecated libraries, were removed. In the updated "HosTaGe" mobile honeypot version, multiple systems exist in the same An-droid Device. The multi-layered approach supports its ability to simulate different systems interchangeably.

The Honeypot works with both Wi-Fi and Cellular networks. Mobile networks operators are fully capable of providing the types of services needed for such honeypots [5], since HosTaGe's IoT implementation mostly works above OSI Layer 4 and only publishes an IP to the internet infrastructure. The implementation is simple since the previous Wi-Fi implementation became abstract and encapsulated both Cellular and Wi-Fi. The app can identify the type of the network the moment the user is connected. The implementation supports a total of 15 existing protocols simulations, with three new IoT protocols added (AMQP, CoAP, MQTT) for IoT and two were renewed to support IoT services (Telnet and SSH). Similarly, and regarding the app's functionality, 15 different Profiles (modes) were included to support virtualization of each protocol. Each profile may implement simulation of a single or two different protocols simultaneously, depending on the IoT device that is simulated. We also included a "Paranoid" profile that simulates all protocols at once. Specifically, for IoT, four profiles were implemented:

3.4.1 Connection Broker

An Connection Broker is a server that receives all messages from the clients and then routes the messages to the corresponding destination clients. It is useful to have a "vanilla" MQTT broker without specific topics to find more malicious clients that target any kind of IoT system.

3.4.2 Temperature Sensor

Temperature Sensor is a common IoT device and if not properly configured can cause several vulnerabilities. For the emulated system we will use the MQTT protocol without encryption. The sensor continuously publishes temperature and humidity readings and has two topics for subscribing /Humidity and /Temperature. These topics are continuously publishing random-generated values (in a specific range to appear realistic)

that represent humidity percentages and temperatures in Celsius.

3.4.3 ESP8266 Smoke Sensor

The ESP8266 is a well-known low-cost Wi-Fi microchip. Most of the known smoke detectors are made within the chip. This system simulates the smoke sensor using CoAP protocol to publish temperature and abnormalities in the environment that can detect a fire. The values published are randomly generated. Most companies have an installed smoke detector, which is connected with the rest of the network. They are usually not maintained properly and rarely get firmware updates which make them susceptible to attackers.

3.4.4 Arduino system

Arduino is an open-source hardware and software device which is widely used in IoT. The available simulated protocols are HTTP, Telnet, CoAP, MQTT and AMQP. When someone tries to access the http port, the GET request returns a web page that shows a regular Arduino monitor page that many Arduino sys-tems have.

Regarding the logging functionality of the Honeypot, network packets from de-tected attacks are logged and can be exported to *JSON* and plain-text format. Addition-ally, for rooted mobile devices, captured packets can also be saved in PCAP format for use with network protocols analyzers like Wireshark. Moreover, they are published with HPFeeds in a MongoDB database.

Specifically for rooted phones, an *iptables* API was introduced for this purpose by installing the needed libraries and executes every *iptable*'s command separately. This overcomes Android's limitation of not being able to bind ports smaller than 1024. A lot of protocols have default port numbers smaller to 1024. Android has a security policy that prevents any service binding to these ports. We have overcome this in rooted phones with *iptables*, but in non-rooted phones we had to follow a different path. The solution was straightforward and consisted of allocating all protocols with default ports, to random ports with a number higher than 1024. For the publication and distribution of attacks in remote databases such as MongoDB, the HPFeeds protocol is integrated in the app. HPFeeds is a lightweight authenticated publish-subscribe protocol. It is used to transfer different payloads. In the app it is used for transferring the packets with *JSON* payloads in a channel. In our app HPFeeds is utilized for publishing the attack recorded from multiple devices that have the app installed in a central database, as more detailed described in section

A unique trait of this virtual honeypot is that one simulated system can be enabled at a time, though having a unique IP Address. A user can change a system at any time and add additional protocols in that system

3.5 Implementation Libraries

The target Android version of our app is the latest (SDK 29) Android 10, with supported migration for (SDK 30) Android X. The minimum SDK version is (SDK 24) Android Nougat 7.0. They are two modes for both rooted and non-rooted device. In rooted devices, the services can bind in ports smaller than 1024, which is not the case for un-rooted devices. A main difference between the rooted and non-rooted devices, can be found under the pcap capturing, that can only work in rooted devices due to raw sockets limitations. The libraries used for the IoT protocols development are:

- **AMQP:** Qpid-broker-core for version 1,0.8., 0.9 (modified version for the app, as the original project does not support android), RabbitMQ (client).
- **CoAP:** Armbed/CoAP-core-5.1.0 (Pull Request for IP other than localhost is now merged in the original repository).
- **MQTT**: Io.moquette:moquette-broker, HiveMQ (client).

Helping libraries for the attack capturing:

- **Linux iptables** (for rooted phones): *iptables* is a firewall utility compatible with Linux based operating systems. Most Android-based phones have *iptables* by default, although some manufacturers disable it and have a custom module for the firewall. Therefore, we developed an API that installs the binary along with the HosTaGe app.
- **tcpdump** (for rooted phones): *tcpdump* is a command line network packet capturing utility and analyzer. With this library, the packets can be captured in raw format and produce a pcap file. The same API used for *iptables* was also utilized to install the *tcpdump* binary when the app first boots. *tcpdump* captures traffic for all ports in a *pcap* format.

3.6 Lab Setup

As mentioned earlier, this app has different profiles, each one emulating a different system. In the Paranoid Profile all systems are simulated at once and thus we can collect different type of attacks. Also, IoT profiles were used in order to have more "targeted" attacks for the IoT protocol. The attacks are stored in a MongoDB

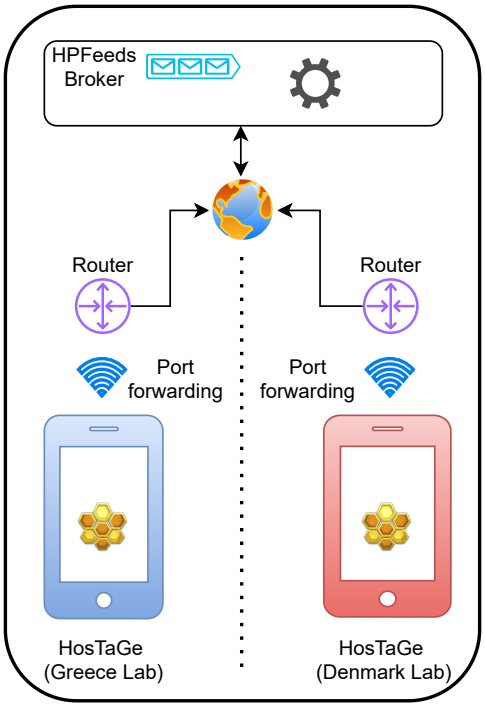

Fig. 3: Experimental setup

database and are received by a HPFeeds publisher integrated in the app. Additionally, the attacks are captured in two different formats, consisting of the *JSON* and the pcap format. More specifically, the *JSON* format displays more details about each packet, since additional information could be added. This additional information consists of the profile, which was active when the packet was captured, and a unique identifier, distinguishing them from different profile captures in the database.

The only drawback of the *JSON* format is the lack of packets' header. In contrast, the pcap format contain the original full packets including the header, which facilitates the conducted analysis. Furthermore, the full analysis of the experiments is specified under a Jupyter Notebook, including more details regarding the number of attacks, the pre-processing of the data and the features selected from the packets.

4 Evaluation

Our evaluation includes two experiments that were conducted with different deployments to collect malicious traffic and perform statistical analysis. The experimental setup and the results are described in the following sections.

4.1 Experimental Setup

The deployment setup of the honeypot experiments is shown in Figure 3. The experiments were conducted in two lab environments in Greece and Denmark. The honeypots were exposed to the Internet for a period of six months from 19 August 2020 to 10 February 2021. Each honeypot is exposed to the Internet through a public IP address on an unfiltered network in the lab environment. The public IP address to the honeypot is configured through port-forwarding from the router. Additionally, some assumptions were considered for the experiment. The first assumption made was that every received packet from the same IP equals to an attack. The second is that the attacks are indeed coming from the source IP's geolocation. However, we acknowledge that many attacks originate from different countries other than what the IPs suggest, due to the fact that attackers may use VPN, Tor or other ways of IP spoofing.

4.2 Results

An overview of the results from the attack data collected from the honeypot is presented in the following sections.

4.2.1 Attack Sources

The data collected from the honeypots is analyzed to extract IoT attack specific information from the two locations. HosTaGe stores all the ingress traffic received in the *JSON* format. Using Jupyter, packets are extracted from the *JSON* file and converted to data frames for analysis.

A noteworthy data point extracted from the captured attack data, is the number of attacks for each source country. Out of the 359,080 total attacks, the distribution of the top attackers: China (261,935), Vietnam (43,921), USA (22,209), United Kingdom (5740) and Russia (4033). Following are India (3955), France (1748), Netherlands (1105), Brazil (1105), Thailand (1102). Regarding the protocols affected during each attack, the distribution appears on Figure 4 and Figure 5.

Preliminary observations from collected data suggested a high number of attacks against SSH and Telnet protocols, in favor of SSH since Telnet is an older protocol with less use in modern systems. Still, Telnet attacks are prevalent and often include legacy home routers and IoT smart devices [14]. As for SSH, published literature identifies that more than 65% of over twenty million SSH servers on the public internet have a password

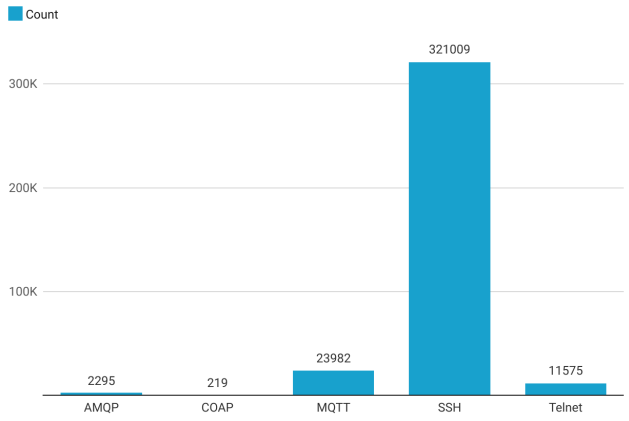

Fig. 4: Total number of attacks per protocol

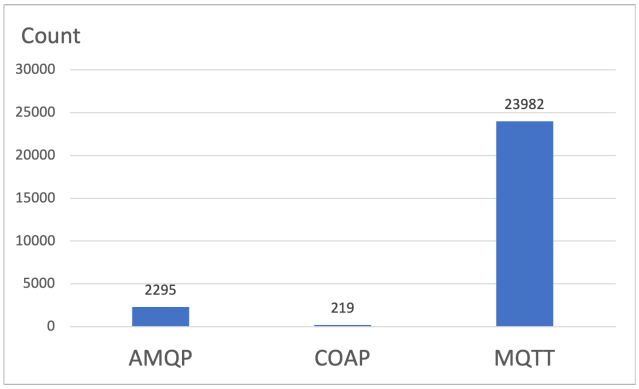

Fig. 5: Total number of attacks per IoT-specific protocol

authentication vulnerability [6]. The Fritz-Frog peer-to-peer botnet, attacked millions of SSH servers found in routers and IoT devices [37]. Preliminary data also suggest an increased number of MQTT attacks in comparison to other IoT protocols. MQTT and CoAP are machine-to-machine communication protocols although they are often lacking secure configurations, or they are misconfigured. For this reason, they are targeted from various attackers [39]. MQTT has been more popular for creating novel attack such as SlowITe [45] a new type of slow DDoS attack designed for MQTT. Other protocols seem to have "less" attacks in our produced dataset. Furthermore, the AMQP and CoAP are newer protocols compared to MQTT [27] and thus less devices currently utilize them. Nevertheless, data do suggest that these protocols are also common targets for bad actors.

During the experiment we noticed that, for both labs (Greece and Denmark Lab environments), most attacker IPs originated from Asia. The IPs that originated from China, are found in blocklist databases, and most commonly attack SSH protocols [2]. The second most common IP (Vietnam) from the Denmark Lab test results can also be found in the same databases with brute force login attempts in SSH protocols [10].

4.2.2 Statistical analysis

This section contains the statistical analysis of the attacks per protocol collected during the five-month period that the experiment was run. Critical assumptions were made to rationalize the data and provide a clarity over the first approach on analyzing protocol attacks collected on honeypot and specify limitations of this study. These can be summarized as follows

- Statistical analysis is restricted by the large standard deviation attack protocol values, which hinders the ability to make certain and solid assumptions.
- The time interval of one week was decided to represent the granularity for the five-month reporting period. This eliminates noise from short-term time frames (e.g. legitimate scans or hit-and-run reconnaissance scans that only occur for a very short amount of time).
- We compared the aggregate data from the attacks in terms of mean, median, maximum and minimum per protocol, whilst comparisons on the attack protocol ratios were made to further evaluate any trends behind the attacks. Analysis did not include false positive/negative rates or relevant F1 scores due to the lack on information behind the context of each scan detected by the IoT network. We assume all attack indicators to involve *true positive* samples of attacks/scans against IoT devices.

The overall number of SSH attacks is almost a factor higher than the the sum of all collected protocol attacks during the five-month reporting period. The SSH protocol represents the only target that constantly collected attacks on a weekly basis and a minimum of nine (weeks), during the twenty-five weeks that the experiment was conducted.

Concerning IoT-specific protocols, the highest number of attacks targeted the MQTT protocol with a maximum number of weekly attacks reaching $8,262$, the second highest attack between all protocols (IoT and generic whilst Telnet protocol collected 12.002 attacks. Finally, the least targeted protocols as clearly portrayed above are the AMQP and CoAP with a sum of $2,289$ and 219 respectively and maximum of only 251 and 45 attacks collected per week.

In Table 1, the display of aggregated attack protocol ratios reflects the order of magnitude between the attack collected per protocol. The SSH protocol

Attack Protocol	Mean of weekly attacks	Maximum weekly attacks	Overall count
AMQP	91.56	251	2.289
CoAP	8.76	45	219
MQTT	957.56	8.262	23.939
SSH	12925.12	29.788	323.128
Telnet	480.08	2.502	12.002

Table 1: Aggregate Protocol Attack Statistics

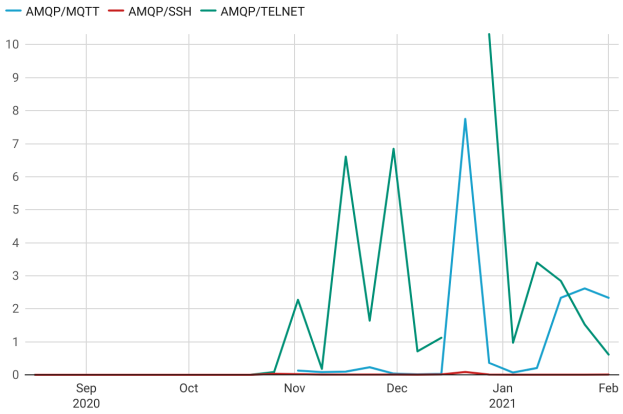

Fig. 6: Attack Ratio - AMQP

is 143 times and 1000 times more targeted than the AMQP and CoAP protocols respectively, whilst it collected 13.5 and 27 more than the MQTT and Telnet protocols respectively. For some reason, there was a Telnet targeted attack that began early 2021 which skyrocketed Telnet data, but this attack was an isolated incident rather a constant event throughout the tested time period. Finally, the ratio between the least targeted protocols (CoAP – AMQP, Figure 7) was by a factor of ten.

Figures 6 through 9 detail a better image of the attack ratio against IoT protocols per week. Indicative examples include that the least targeted protocol CoAP, collected 3.5 times more attacks than MQTT in the first week of the experiment, whilst in December the AMQP protocol received 248 times more attacks than Telnet, despite that Telnet collected five times more attacks during the five-month period. The Telnet protocol collected higher number of attacks especially at the first two months of the experiment, receiving more attacks by almost a factor of 100 of MQTT and the only protocol to surpass SSH in attacks per week, reaching 1.5 higher number of attacks in the second week of the study.

From non IoT-specific protocols, SSH was identified to be the most targeted across all five protocols which follows since SSH RPC calls are prevalent in many IoT

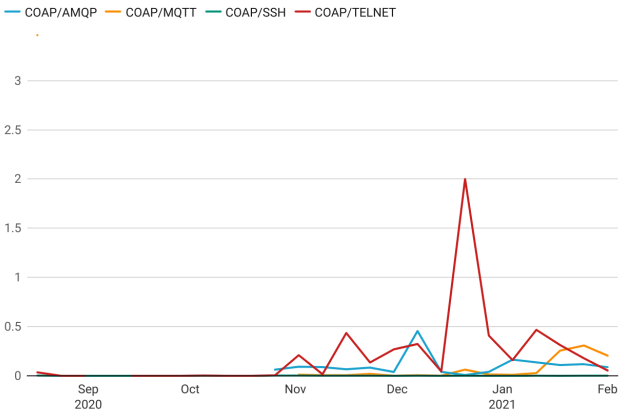

Fig. 7: Attack Ratio - CoAP

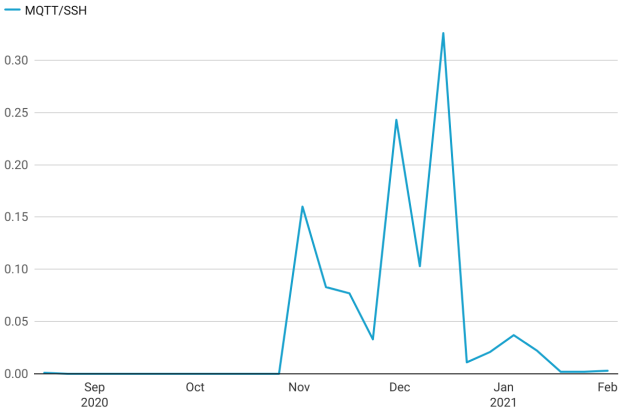

Fig. 8: Attack Ratio - MQTT/SSH

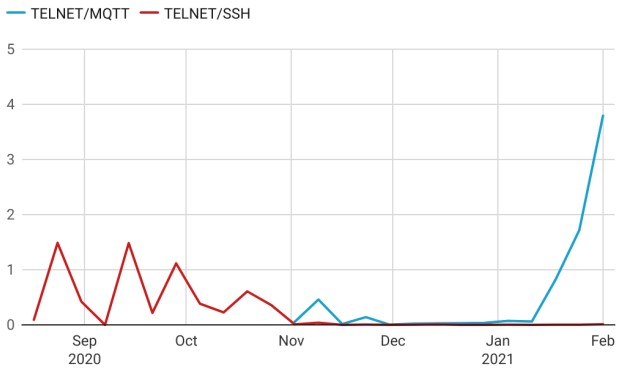

Fig. 9: Attack Ratio - Telnet

devices and remote sensors. Thirdly, the location of the Honeypot was presumed to be a determining factor of the attacks collected. More specifically, MQTT protocol was targeted at a higher rate when the honeypot was activated in Greece, whereas attacks focused on the Telnet protocol were quite high when the honeypot was

activated in Denmark. Finally, it can be assumed that the least favorable protocols to attack were the AMQP and CoAP, due to the significantly lower overall number of attacks and the fact that they have been recently created, which though might change if the honeypot is activated in other countries than Greece and Denmark.

5 Conclusion

With emerging novel attacks targeting IoT devices each day, new security measures and mechanisms should be developed. In our research, we present a potential set of modifications to the HosTaGe honeypot, which will assist in preparing it for the next generation of IoT attacks. IoT functionality was added as a core feature and with three new protocols MQTT, AMQP and CoAP.

Overall, the honeypot attracted many IoT attacks. The majority of discovered attackers were part of known IoT botnets, targeting SSH devices. The HosTaGe IoT honeypot was evaluated and validated in two experiments conducted in Denmark and Greece. The results demonstrate that HosTaGe is capable of effectively simulating the implemented IoT profiles and protocols by attracting malicious traffic. The majority of attacker's IPs existed in blacklist databases. The source location of the attacks matches the common locations from already conducted research and reports of other deployed IoT honeypots around the world [18]. Overall, we argue that the honeypot may be used as an additional defense mechanism in vulnerable IoT networks.

The HosTaGe transformation and the inclusion of an IoT mode for protocol simulation creates new paths for our research. New IoT protocols can be added like XMPP, along with developing new simulated systems, which may expand on other research fields. Additionally, it is recommended that the honeypot is also placed in different locations, to further enhance and confirm the total number of attacks per protocol. The focus will not only be in the enhancement of the app, but also in the data that it collects.

Competing Interests None of the authors have received any research grants. None of the authors have received a speaker honorarium from any company. All authors declare that none of them has any conflict of interest.

Research Data Policy and Data Availability Statements All data generated or analyzed during this study are included in this published article. Compliance with Ethical Standards This study was not funded.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. M. Abomhara and G. M. Køien, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, pp. 65–88, 2015.
2. AbuseIPDB. (2020) Abuseipdb. [Online]. Available: https://www.abuseipdb.com/check/218.92.0.189
3. P. Adkins. (2017) Kako honeypot. [Online]. Available: https://github.com/darkarnium/kako
4. H. M. Ahmed, N. F. Hassan, and A. A. Fahad, "Designing a smartphone honeypot system using performance counters," *Karbala International Journal of Modern Science*, vol. 3, no. 1, pp. 46–52, 2017.
5. . Alriksson. (2020) Critical internet of things (iot) connectivity is ideal for a wide range of time-critical use cases across most industry verticals, and mobile network operators are uniquely positioned to deliver it.
6. R. Andrews, D. A. Hahn, and A. G. Bardas, "Measuring the prevalence of the password authentication vulnerability in ssh," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.
7. M. Bansal *et al.*, "Performance comparison of mqtt and coap protocols in different simulation environments," *Inventive Communication and Computational Technologies*, pp. 549–560, 2021.
8. G. Bernieri, M. Conti, and F. Pascucci, "Mimepot: A model-based honeypot for industrial control networks," in *2019 ieee international conference on systems, man and cybernetics (smc)*. IEEE, 2019, pp. 433–438.
9. Bevywise. (2022) An exhaustive iot simulator for iot/mqtt application testing. [Online]. Available: https://www.bevywise.com/iot-simulator/
10. Blackhat. (2020) Black hat directory. [Online]. Available: https://blackhat.directory/ip/171.240.199.80
11. Bosch-s. (2022) Mqtt simulator. [Online]. Available: http://documentation.bosch-si.com/iot/RM/v7/en/101937.htm
12. D. I. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer, "Cryplh: Protecting smart energy systems from targeted attacks with a plc honeypot," in *International Workshop on Smart Grid Security*. Springer, 2014, pp. 181–192.
13. C. Cimpanu. (2020) The coap protocol is the next big thing for ddos attacks. [Online]. Available: https://www.zdnet.com/article/the-coap-protocol-is-the-next-big-thing-for-ddos-attacks/
14. ——. (2020) Hacker leaks passwords for more than 500,000 servers, routers, and iot devices. [Online]. Available: https://www.zdnet.com/article/hacker-leaks-passwords-for-more-than-500000-servers-routers-and-iot-devices/
15. CloudAMQP. (2020) Amqp. [Online]. Available: https://www.cloudamqp.com/docs/amqp.html
16. G. Communications. (2022) Mimic mqtt simulator - for iot simulation. [Online]. Available: https://www.gambitcomm.com/site/mqttsimulator.php
17. D. Demeter, M. Preuss, and Y. Shmelev. (2019) Iot: a malware story. [Online]. Available: https://securelist.com/iot-a-malware-story/94451/

18. ENISA. (2020) Enisa threat landscape 2020 - botnet. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2020-botnet
19. O. Erdem, A. Pektas, and M. Kara, "Honeything: A new honeypot design for cpe devices," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 12, no. 9, pp. 4512–4526, 2018.
20. A. Ericsson, "Cellular networks for massive iot–enabling low power wide area applications," *no. January*, pp. 1–13, 2016.
21. M. . C. Franik. (2020) Serious flaws found in multiple smart home hubs: Is your device among them? [Online]. Available: https://www.welivesecurity.com/2020/04/22/serious-flaws-smart-home-hubs-is-your-device-among-them/
22. GreenDAO. (2021) greeendao. [Online]. Available: https://greenrobot.org/greendao/
23. M. A. Hakim, H. Aksu, A. S. Uluagac, and K. Akkaya, "U-pot: A honeypot framework for upnp-based iot devices," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2018, pp. 1–8.
24. A. Higgins, "Adaptive containerised honeypots for cyber-incident monitoring," *Integrated Masters in Computer Engineering (MAI)*, 2018.
25. S. Hilt, F. Maggi, C. Perine, L. Remorin, M. Rösler, and R. Vosseler, "Caught in the act: Running a realistic factory honeypot to capture real threats," *Trend Micro, Shibuya City, Japan, White Paper*, 2020.
26. C. C. Ho and C.-Y. Ting, "A conceptual framework for smart mobile honeypots," *Academia, http://www. academia. edu/download/31058450/KasperskyConferen cecchocyting. Pdf*, 2014.
27. S. P. Jaikar and K. R. Iyer, "A survey of messaging protocols for iot systems," *International Journal of Advanced in Management, Technology and Engineering Sciences*, vol. 8, no. 2, pp. 510–514, 2018.
28. M. M. Kendrick and Z. A. Rucker, "Energy-grid threat analysis using honeypots," Naval Postgraduate School Monterey United States, Tech. Rep., 2019.
29. P. Krishnaprasad, "Capturing attacks on iot devices with a multi-purpose iot honeypot," *INDIAN INSTITUTE OF TECHNOLOGY KANPUR*, 2017.
30. S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos, and R. Beyah, "Rethinking the honeypot for cyber-physical systems," *IEEE Internet Computing*, vol. 20, no. 5, pp. 9–17, 2016.
31. T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices," *Black Hat*, vol. 2017, 2017.
32. mqtt.org. (2021) Mqtt: The standard for iot messaging. [Online]. Available: https://mqtt.org/
33. Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "Iotpot: A novel honeypot for revealing current iot threats," *Journal of Information Processing*, vol. 24, no. 3, pp. 522–533, 2016.
34. D. Pliatsios, P. Sarigiannidis, T. Liatifis, K. Rompolos, and I. Siniosoglou, "A novel and interactive industrial control system honeypot for critical smart grid infrastructure," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2019, pp. 1–6.
35. Z. Qing and B. Guangdong. (2017) 3g/4g intranet scanning and its application on the worm hole vulnerability.
36. B. Safaei, A. M. H. Monazzah, M. B. Bafroei, and A. Ejlali, "Reliability side-effects in internet of things application layer protocols," in *2017 2nd International Conference on System Reliability and Safety (ICSRS)*. IEEE, 2017, pp. 207–212.
37. T. Seals. (2020) Fritzfrog botnet attacks millions of ssh servers. [Online]. Available: https://threatpost.com/fritzfrog-botnet-millions-ssh-servers/158489/
38. S. Sentanoe, B. Taubmann, and H. P. Reiser, "Sarracenia: enhancing the performance and stealthiness of ssh honeypots using virtual machine introspection," in *Nordic Conference on Secure IT Systems*. Springer, 2018, pp. 255–271.
39. S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, "Open for hire: Attack trends and misconfiguration pitfalls of iot devices," in *Proceedings of the 21st ACM Internet Measurement Conference*, ser. IMC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 195–215. [Online]. Available: https://doi.org/10.1145/3487552.3487833
40. ——, "Riotpot: a modular hybrid-interaction iot/ot honeypot," in *26th European Symposium on Research in Computer Security (ESORICS) 2021*. Springer, 2021.
41. SSH. (2020) Ssh protocol. [Online]. Available: https://www.ssh.com/ssh/protocol/
42. . Stahie, S. (2020) ver 500,000 credentials for telnet exposed iot devices and servers leaked online. [Online]. Available: https://www.bitdefender.com/box/blog/iot-news/500000-credentials-telnet-exposed-iot-devices-servers-leaked-online/
43. S. Stahie. (2020) Iot botnet attacks on the rise in 2020. [Online]. Available: https://www.bitdefender.com/box/blog/iot-news/iot-botnet-attacks-rise-2020/
44. A. Z. Tabari and X. Ou, "A first step towards understanding real-world attacks on iot devices," *arXiv preprint arXiv:2003.01218*, 2020.
45. I. Vaccari, M. Aiello, and E. Cambiaso, "Slowite, a novel denial of service attack affecting mqtt," *Sensors*, vol. 20, no. 10, p. 2932, 2020.
46. E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Hostage: a mobile honeypot for collaborative defense," in *Proceedings of the 7th International Conference on Security of Information and Networks*, 2014, pp. 330–333.
47. B. Wang, Y. Dou, Y. Sang, Y. Zhang, and J. Huang, "Iotcmal: Towards a hybrid iot honeypot for capturing and analyzing malware," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.
48. H. Wang, H. He, W. Zhang, W. Liu, P. Liu, and A. Javadpour, "Using honeypots to model botnet attacks on the internet of medical things," *Computers and Electrical Engineering*, vol. 102, p. 108212, 2022.
49. M. Wang, J. Santillan, and F. Kuipers, "Thingpot: an interactive internet-of-things honeypot," *arXiv preprint arXiv:1807.04114*, 2018.
50. F. Xiao, E. Chen, and Q. Xu, "S7commtrace: A high interactive honeypot for industrial control system based on s7 protocol," in *International Conference on Information and Communications Security*. Springer, 2017, pp. 412–423.