Open for hire: attack trends and misconfiguration pitfalls of IoT devices

Conference Paper · November 2021

DOI: 10.1145/3487552.3487833

3 authors, including:

Shreyas Srinivasa
Aalborg University
16 PUBLICATIONS **157** CITATIONS
SEE PROFILE

Emmanouil Vasilomanolakis
Technical University of Denmark
73 PUBLICATIONS **1,307** CITATIONS
SEE PROFILE

Open for hire: attack trends and misconfiguration pitfalls of IoT devices

Shreyas Srinivasa
Aalborg University, Denmark

Jens Myrup Pedersen
Aalborg University, Denmark

Emmanouil Vasilomanolakis
Aalborg University, Denmark

ABSTRACT

Mirai and its variants have demonstrated the ease and devastating effects of exploiting vulnerable Internet of Things (IoT) devices. In many cases, the exploitation vector is not sophisticated; rather, adversaries exploit misconfigured devices (e.g. unauthenticated protocol settings or weak/default passwords). Our work aims at unveiling the state of IoT devices along with an exploration of the current attack landscape. In this paper, we perform an Internet-level IPv4 scan to unveil 1.8 million misconfigured IoT devices that may be exploited to perform large-scale attacks. These results are filtered to exclude a total of 8,192 devices that we identify as honeypots during our scan. To study current attack trends, we deploy six state-of-art IoT honeypots for a period of 1 month. We gather a total of 200,209 attacks and investigate how adversaries leverage misconfigured IoT devices. In particular, we study different attack types, including denial of service, multistage attacks and attacks from infected online hosts. Furthermore, we analyze data from a /8 network telescope covering a total of 81 billion requests towards IoT protocols (e.g. CoAP, UPnP). Combining knowledge from the aforementioned experiments, we identify 11,118 IP addresses (that are part of the detected misconfigured IoT devices) that attacked our honeypot setup and the network telescope.

ACM Reference Format:
Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. 2021. Open for hire: attack trends and misconfiguration pitfalls of IoT devices. In *ACM Internet Measurement Conference (IMC '21), November 2–4, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3487552.3487833

1 INTRODUCTION

With the adoption of IoT, there is an increase of misconfigured devices on the Internet. Some are incorrectly configured or left with default configuration, thereby making them vulnerable [28]. Misconfigured IoT devices are exploited on a large scale by malware like Mirai that infect vulnerable devices with bots [44]. A device is considered to be *misconfigured* if its incorrect configuration leads to vulnerabilities. NIST defines misconfiguration as *"An incorrect or suboptimal configuration of an information system or system component that may lead to vulnerabilities"* [58]. Moreover, attacks like denial-of-service, ransomware, or data leaks can be purchased and facilitated through botnets. For instance, many variants of the Mirai botnet and newer IoT malware like GitPaste-12 [13], Kaiji [9], RHOMBUS [49] continue to look for vulnerable devices on the Internet [44]. Furthermore, recent research shows the possibilities of DoS attacks through messaging protocols like MQTT [87, 88] and CoAP [91].

According to the ENISA Threat Landscape Report 2020, malware attacks are the leading and emerging threats worldwide [16]. While it is known that botmasters look for vulnerable devices with misconfigured protocols of Telnet and SSH, research suggests that bot deployments are now possible with IoT-based protocols like MQTT, AMQP, and UPnP [4, 31, 51, 82]. With the increasing adoption of IoT in diverse sectors like Industry 4.0, healthcare, and critical infrastructure, we argue that this poses a significant threat.

Heretofore, there has been research on the underlying IoT vulnerabilities and proposing honeypots to analyze the threat actors for specific protocols [32, 46, 63, 99]. However, to the best of our knowledge, no work combines an active search for misconfigured devices with an analysis of the attack trends in IoT by deploying multiple honeypots and studying the traffic flow received on a network telescope. In this paper, we unveil the vulnerable aspects of misconfigured services on IoT devices and emphasize the importance of authentication and authorization in IoT protocols and devices.

Our contributions are summarized as follows:

- We perform Internet-wide scans on six protocols: Telnet, MQTT, CoAP, AMQP, XMPP, and UPnP. As a result, we unveil 1.8 million misconfigured IoT devices that can either be infected with bots or be leveraged for a (D)DoS amplification attack. In addition, we use open datasets to complement our findings. Furthermore, our scan takes into account the existence of honeypots. To deal with the lack of ground truth knowledge for deployed honeypots on the Internet, we analyze the response banners from our scan and the static banners returned by open-source honeypots. Hence, we filter out from the results 8,192 systems that we classify as honeypots.
- We deploy six SOTA *IoT honeypots*, to capture and analyze the attack vectors on the protocols scanned. Moreover, we analyze data from a /8 *network-telescope* with 16 million IP addresses to better understand Internet scanning trends in IoT protocols.
- Combing knowledge from the IPv4 scan, the honeypot deployment and the network telescope traffic analysis, we discover 11,118 (out of the 1.8 million) misconfigured IoT devices that attacked our honeypot setup and the network telescope.

The rest of the paper is organized as follows. Section 2 introduces the related work in detecting vulnerable IoT devices on the

IMC '21, November 2–4, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9129-0/21/11.
https://doi.org/10.1145/3487552.3487833

Internet and IoT honeypots. In Section 3 we describe our methodology of finding misconfigured devices on the Internet, detection of honeypots and deploying state-of-the-art honeypots in our lab environment to learn the attack vectors and analysis of FlowTuple data from a network telescope. Section 4 shows the results obtained from our methodology. In Section 5 we discuss the attack trends and findings of our research. Section 6 concludes the paper and discusses potential future work.

2 RELATED WORK

This section discusses the related work in the area of Internet-wide scanning for finding vulnerable IoT devices, IoT honeypots, and IoT honeypot fingerprinting.

2.1 Internet-wide scanning for vulnerable IoT devices

The widespread increase of IoT devices on the Internet has called upon various kinds of research, focusing on their security and trust [98]. The majority of the research in this area includes fingerprinting IoT devices to facilitate exploitation based on their type. However, there is less research that follows the approach of scanning the Internet to find vulnerable devices. Markowsky et al. [50] demonstrate how to scan the Internet for vulnerable IoT devices using the Shodan scan engine [73] and scanning tools like Masscan [29], NMap [47], and PFT [84]. The authors describe multiple ways of finding vulnerable devices on the Internet using banners of known services. The scan finds more than 1.6 million vulnerable devices on the Internet. Although we make use of a similar methodology, i.e, we utilize ZMap and Shodan in our scanning approach, we leverage open datasets and run the scans with custom probes for both TCP and UDP protocols. Furthermore, unlike Markowsky et al. we do not try to connect to the devices after the scanning process. We also use the banners and the initial response received from the hosts from our scans. In addition to results from Shodan, we combine datasets from open projects that do not index the scan results based on banners or responses.

Neshenko et al. [57] make an exhaustive survey of IoT vulnerabilities by an empirical study of the published research work on IoT. Their analysis proposes a taxonomy of IoT vulnerabilities, including their technical details and consequences. The authors also evaluate IoT exploits through analysis of a passive network dataset obtained by a network telescope. The evaluation provides good insights into the number of vulnerable IoT devices by country, infected devices, and malicious IoT traffic. To sum up, there is significant research on fingerprinting of IoT devices using passive data sets. However, there is scarce work on scanning the Internet with custom probes to discover misconfigured IoT devices.

The work of Springall et al. [74] is the closest to ours. The authors attempt to find FTP servers on the Internet that accept anonymous logins and investigate real-world attacks by deploying FTP honeypots. Springall et al. detect more than 20,000 public FTP servers that allowed write access. The authors focus mainly on the FTP protocol and the anonymous login misconfiguration that allows remote users to authenticate without any access information.

Honeypot	Telnet	MQTT	CoAP	AMQP	XMPP	UPnP	Open-source
IoTPot (2016)	●						
ThingPot (2018)					●		●
U-Pot (2018)						●	●
IoTCandyJar (2017)		●	●		●		●
HosTaGe (2020)	●	●	●	●			●
Conpot (2020)	●						●
Cowrie (2020)	●						●
Dionaea (2020)		●					●
MQTT and CoAP Honeypots (2019)		●	●				
Anglerfish	●	●				●	

Table 1: IoT Honeypots

2.2 IoT-Honeypots

The use of honeypots and network telescopes to monitor attacks is not new. Honeypots are deception-based entities that simulate the services of a target system. All connection attempts to a honeypot can be considered malicious as there is no real reason for accessing a honeypot system. Over the years, many honeypots have been proposed, both open-source and research-based, to understand the threats to IoT protocols. The Honeynet Project [64] offers a number of open-source honeypots such as: Conpot [69], Dionaea [83] and HosTaGe [90] that simulate IoT protocols (e.g. Telnet, MQTT, CoAP and AMQP). Other honeypots include ThingPot [99], IoTPot [63], UPot [32] and IoTCandyJar [46].

Table 1 lists IoT-honeypots and the protocols the simulate. IoTPOT [63] proposes a honeypot and a sandbox environment for capturing Telnet-based attacks. Through IoTPOT, the authors were able to identify four distinct DDoS malware families targeting Telnet-enabled IoT devices based on the attacks gathered. Wang et al. propose ThingPot [99] that emulates the XMPP protocol. The authors also implemented the Philips Hue smart home lighting system profile into ThingPot that emulates the Hue devices like the bridge and the smart lamps. During the evaluation of ThingPot, the authors discovered attacks that tried to gain control of the system and some fuzzing attempts. Hakim et al. propose U-Pot [32], a UPnP-based honeypot framework for capturing attacks on IoT devices that use Universal Plug and Play (UPnP) protocol. The authors claim that U-Pot offers high-interaction capabilities and is agnostic of device type. The authors deploy the profile of the Belkin Wemo smart switch [11] into U-Pot and evaluate its performance by trying to measure the response times from the honeypot. The results are observed to have near similar response times to real devices.

Luo et al. propose IoTCandyJar [46], a machine learning-based honeypot that learns the behavioral knowledge of IoT devices by continuous Internet-wide probing. The honeypot sends Internet-wide probes as seed requests to get response information from devices with specific open ports. The honeypot responds to the attacker queries, using the saved responses and the requests in its training database. HosTaGe [89, 90] is a low-interaction mobile honeypot that emulates many protocols, including IoT protocols like MQTT, CoAP, and AMQP. Further, the honeypot offers device profiles like Arduino, a smoke-sensor, and a temperature sensor for simulation. Shimada et al. implemented MQTT, and CoAP honeypots [72] to observe the possible attack vectors on the IoT messaging protocols. The authors observed a large number of MQTT requests on the honeypot and requests from unknown protocols.

Lastly, we discover the Anglerfish honeypot from the results of our honeypot detection methodology which is described in Section 3. The honeypot is managed by Netlab 360 [1], a commercial security organization.

2.3 Network Telescopes

Data from network telescopes has been utilized in some research to study the scanning trends. Durumeric et al. [23] use the data from an extensive network telescope to gain insights in scanning traffic, behavior, and patterns. The authors reveal many attacks detected from Darknet IP sources and derive many statistical patterns from the scanning data. Similarly, Heo et al. [33] analyze the connection-level log data of a large-scale campus network to study the trends in scanning. The log data used for analysis is acquired from the firewalls deployed in the campus network. The authors provide an in-depth analysis and classification of the scan traffic.

Jonker et al. [43] use four independent datasets that include honeypots and a network telescope to perform a comprehensive analysis of the gathered attacks and introduce a new framework to enable a macroscopic characterization of attacks, attack targets, and DDoS Protection Services. The authors present significant results regarding the global problems caused by DoS attacks and the most targeted types of servers.

Lastly, Richter et al. analyze the unsolicited traffic at firewalls from 89,000 hosts across 1,300 networks of a significant Content Distribution Network [68]. Their findings indicate that localized scanning campaigns likely target narrow regions in the IP address space. Their characteristics vary compared to the Internet-wide scanning services in terms of target selection. The authors further compare the suspicious traffic received on the firewalls to the UCSD Darknet Network Telescope [54] and provide a comprehensive analysis of the scanning services.

2.4 IoT-Honeypot Fingerprinting

Honeypot fingerprinting is the process of detecting if a target system is indeed a honeypot. The fingerprinting process may involve either active, passive, or both fingerprinting techniques. Some examples include banner-based, static-response, the use of low-interaction libraries, and response times. Honeypot fingerprinting can help adversaries in avoiding any interaction with a honeypot either directly or through malware propagation. Research on honeypot fingerprinting has increased over time. Early works on honeypot fingerprinting started in 2005 by Holz et al. [35] who queried the target system for known static responses from honeypots. More recent works include Vetterl et al. [92] who systematically detected known open-source honeypots by analyzing the deviation in response from that of honeypots. The authors considered open-source honeypots that emulate Telnet, SSH, and HTTP protocols.

A first approach towards the detection of IoT honeypots was proposed by Surnin et al. [80]. The authors detect honeypots that emulate SSH and Telnet protocols by performing multiple checks through tests that determine if the target is a honeypot. Based on the results of each test, the authors assign a probability for the target. In this paper, we also use static banners sent by known IoT honeypots to detect and filter them from our scan results. For this, we extend our previous work on honeypot fingerprinting [75].

3 METHODOLOGY

This section describes the methodology for unveiling vulnerable devices and the attack trends.

3.1 Detection of misconfigured IoT-devices

We follow two approaches for the detection of misconfigured IoT devices that are exposed to the Internet. *First*, we perform Internet-wide scans for six protocols. In particular, *MQTT, CoAP, AMQP, XMPP* and *UPnP* are chosen on the basis of their adoption and usage in IoT [10]. In addition, *Telnet* is selected as it has been significantly targeted by malware in the past [5, 6, 93]. We subsequently examine the received banners for known vulnerabilities and misconfigurations, e.g. accepting the authentication in plain text. *Second*, we use the available and open network datasets to search for vulnerable devices.

3.1.1 Internet-wide scanning: In this approach, we scan the Internet for six protocols (Telnet, MQTT, CoAP, AMQP, XMPP, and UPnP). We utilize ZMap [24] along with ZGrab [21] to capture the banners of the responding hosts for further analysis. We use one of the servers running Ubuntu 20.04-LTS OS with a fixed static IP address in our lab as the scanning host. For the scan of UDP protocols like CoAP and UPnP, we used custom scripts that requested a response from the target host. For example, the UDP scan for CoAP protocol included the query "/.well-known/core" in the scan request. Note that CoAP responds to all requests if there is no authentication configured. Similarly, for UPnP, we send an "ssdp:discover" request. The scans for all the six protocols were completed in a week between March 1-5 2021 (see Table 9 in the Appendix for the specific scan dates for each protocol). The information retrieved from the scans, such as IP address, port, response, banner, were stored in a database for further analysis to identify the vulnerable hosts. The scans followed the default blocklist provided by ZMap [100] and the European blocklist from the FireHOL Project [25]. We discuss the ethical aspects of scanning in Appendix Section A.3.

3.1.2 Open datasets: Open datasets of Internet-wide scans are provided by projects like Project Sonar from Rapid7 [67] and Shodan [73]. These datasets contain essential information like IP address, port, protocol, headers, and banner information of the host with the open ports identified through the scan. We utilize the datasets from Project Sonar and Shodan to search for misconfigured IoT devices in Telnet, MQTT, CoAP, AMQP, XMPP, and UPnP. The information from the datasets assists us in verifying the results obtained from our scans. The aforementioned datasets vary by scan frequency, and hence we correlate the results identified in all the datasets.

3.1.3 Identifying misconfigured hosts: The protocols considered in our work involve both TCP and UDP protocols. We consider vulnerabilities associated with the misconfiguration of protocols in IoT devices. We focus on devices that prominently lack any authentication, authorization, and encryption configurations. Furthermore, we derive that many devices with default configurations also use default parameters for authentication. To identify vulnerable hosts from the scan data obtained from the above approach, we classify our methodology into two: *Banner-based* and *Response-Based.*

Banner-based (TCP):. This approach involves the analysis of the banners received on a successful connection with the target host. Banner grabbing is a technique that is used to retrieve more information from the target host. The information in the banners may help know the type, version, username, and even the session-related metadata. Based on the scanned protocol, the banners vary in the information sent. We use the ZGrab tool in our scan to fetch the banner information from the connected target. This approach is followed for the Telnet, MQTT, AMQP, and XMPP protocols. In Table 2 we list sample banners that indicate misconfiguration of the protocol on the target device and are explained below.

- **Telnet:** We examine the banners received from the Telnet scan. The scan tries to establish a session with the target host to discover an open Telnet port, either 23 or 2323. Upon connecting, the target host sends a banner to our scanning host with basic server information. While the Telnet protocol itself can be exploited for active banner grabbing, we instead use our ZMap Telnet scan probe to get essential information on the target host. The banners received from the hosts provide us with information like the protocol, server, version, and some headers. We examine the banners received for established connections with unauthenticated console access. In case of finding certain characters like "$", "root@xxx:~$" and "admin@xxx:~$" in the response banners, we infer that the target hosts accept unauthenticated connections.
- **MQTT:** The MQTT (Message Queuing Telemetry Transport) protocol scan investigates the possibility of connecting to port 1883 without any authentication. The banner received upon connection establishment with a target host provides information about supported authentication methods or connects to the target directly. After a successful connection, all the topics and channels on the target host are listed. We examine the received banners for "MQTT Connection Code:0" which specifies unauthenticated access to MQTT servers.
- **AMQP:** The AMQP (Advanced Message Queuing Protocol) scan involves scanning the Internet for port 5672. The probe retrieves metadata from the target host like version, product, and the supported authentication mechanisms on connection. The AMQP protocol has many open-source implementations like RabbitMQ [95], Apache Qpid [27] and Apache ActiveMQ [26]. We refer to the CVE [53] and NIST NVD [59] database to search for known vulnerable versions of the protocol used in the devices detected from our scan. The findings are listed in Section 4.
- **XMPP:** The XMPP protocol (Extensible Messaging and Presence Protocol) is widely used in IoT devices for message passing and communication. The XMPP protocol is scanned for both client (5222) and server ports (5269). We primarily scan for devices that support non-TLS connections on these ports. Then, we examine the banners received from the hosts for known vulnerabilities and misconfigurations, like accepting the authentication in plain text. Furthermore, as XMPP supports anonymous logins, it is possible to establish connections with the servers without any authentication. The banner provides information like version, features and supported authentication-mechanisms. The information received from the banners is used to determine the potential vulnerabilities on the device.

Protocol	Banner Response Indicator	Misconfiguration
Telnet	$	No auth, console access
Telnet	root@xxx:~$	No auth, root console access
Telnet	admin@xxx:~$	No auth, root console access
MQTT	MQTT Connection Code:0	Connection Accepted with no auth
AMQP	Version: 2.7.1	No auth
AMQP	Version: 2.8.4	No auth
XMPP	MECHANISM<PLAIN>	No encryption
XMPP	MECHANISM<ANONYMOUS>	No auth

Table 2: Misconfiguration indicators: TCP protocols

Response-based (UDP):. The protocols using UDP as the transport layer do not respond with banners and therefore have to be explicitly queried for any information on the service. We target two UDP-based protocols, namely CoAP and UPnP, employed in IoT devices on the Internet to search for any misconfigurations and known vulnerabilities. We use the ZMap tool to scan for open CoAP and UPnP ports. The methodology followed for each of the protocols is described below.

Protocol	Response	Misconfiguration
CoAP	x1C	Full Access
CoAP	220	Connected Session
CoAP	220-Admin	Admin access connection
CoAP	CoAP Resources	Resource Disclosure
UPnP	upnp:rootdevice USN: uuid:5a34308c-1a2c-4546 -ac5d-7663dd01dca1::upnp:rootdevice EXT: SERVER: Ubuntu/lucid UPnP/1.0 MiniUPnPd/1.4 LOCATION: http://192.168.0.1:16537/rootDesc.xml	Resource Disclosure

Table 3: Misconfiguration indicators: UDP protocols

- **CoAP:** The CoAP (Constrained Application Protocol) is a web-based transfer protocol used in constrained environments like IoT devices for machine-to-machine communication. CoAP supports multicast and uses UDP as the transport layer. We scan the Internet for CoAP port 5683 and query the end systems for "/.well-known/core". The query triggers a response from the host, based on the configuration set by the administrators. Since CoAP can easily translate to HTTP, it responds with responses like "x1C" that indicate full access to the target system. Table 3 summarizes some of the responses received from misconfigured devices and their misconfiguration details. The sample responses listed in the table show the indicators in the response that denote a specific misconfiguration. However, having systems with CoAP exposed to the Internet itself is a vulnerability and can be recruited for DoS amplification attacks [8].
- **UPnP and SSDP:** The UPnP (Universal Plug and Play) protocol enables device discovery and control in a network. Internet providers use UPnP forwarding on routers to deploy network configuration. The UPnP protocol uses SSDP (Simple Service Discovery Protocol) for the advertisement and discovery of devices on a network. SSDP has been used extensively in smart-home and industrial IoT environments

for automation and control of IoT devices. We scan the Internet for devices with SSDP enabled on port 1900 and trigger a response to a query. Table 3 shows a sample response obtained from a device exposed to the Internet and SSDP enabled. The devices exposed to the Internet could be recruited by malware or botmasters or adversaries for DDoS attacks [17].

The banners and the responses received from active scanning and querying are stored in a database to perform further analysis. Furthermore, we analyze the responses for known high-severity vulnerabilities from the CVE database. The results are correlated with the open datasets analyzed from Subsection 3.1.2. We find a total of 1, 832, 893 unique, vulnerable hosts exposed to the Internet and present our findings and analysis in the results section.

3.2 IoT-Honeypot Fingerprinting

From our Internet-scanning methodology, we expect that some of the misconfigured devices may be honeypots and can poison our result dataset. Thus, we perform honeypot fingerprinting to identify honeypots in our dataset and filter them. Honeypots are widely used deception-based network monitoring systems that proactively detect attacks. They work by simulating protocols and services on the target system and classified based on their simulation levels into low, medium, and high interaction. We filter honeypots from our scan results by following banner-based honeypot fingerprinting. This technique is adapted from existing research methodology proposed by Morishita et al. and Vetterl et al. [55, 92] and is extended to detect IoT-based honeypots.

Honeypot fingerprinting is the technique used to determine if a vulnerable target system is a honeypot [55, 80, 92]. This may assist honeypot developers improve the simulation capabilities, or help adversaries evade honeypots. The techniques are based on banners, response-deviation, static content, lack of simulation, and interaction capabilities. We leverage our previous work on multistage honeypot fingerprinting that is based on banners and responses received from the honeypots [75]. The framework performs sequential checks based on the services discovered on the target host and the response received is analyzed to determine if the target is a honeypot. We deploy open-source and widely used honeypots in our lab to determine the unique characteristics that differentiate them from existing systems. These characteristics can be static banners, response, or content. For the purposes of this paper we only attempt fingerprinting for honeypots emulating Telnet. These include the HoneyPy [48], Cowrie [61], MTPot [19], Telnet IoT honeypot [42], Conpot [69], Kippo [20], Kako [3], Hontel [76] and Anglerfish [1] honeypots.

3.3 IoT Honeypot Deployment

The scans from our methodology reveal a large number of misconfigured devices. To determine the potential attack vectors and to study the attack trends, one of the obvious ways is to deploy honeypots. Honeypots have been a valuable resource for studying the attack trends. We choose open-source honeypots and deploy them in our lab setup, where they are configured to face the Internet without any firewall (see Appendix Section A.3 for details about how we ensured that our honeypots were not used for malicious purposes). The network traffic gathered on all these honeypots is analyzed to understand the attack trends. We describe the IoT honeypots and their deployment in the following subsections.

3.3.1 IoT Honeypots. We choose Cowrie [61], HosTaGe [90], Dionaea [83], ThingPot [99], U-Pot [32], and Conpot [69] honeypots in our methodology as we find these honeypots relevant to our study based on emulated protocols and because they are open source and widely used [55, 92]. Furthermore, these honeypots are capable of simulating IoT-based device profiles. For example, the HosTaGe honeypot can simulate a CoAP-based smoke sensor or, an Arduino board running IoT protocols. The protocols emulated by these honeypots are listed in Table 1.

3.3.2 Deployment Setup. The honeypots are deployed in our lab environment with an unfiltered network. Moreover, the honeypots are grouped based on the emulated protocols as shown in Figure 1. By grouping them in this way, we ensure no overlap of the protocols emulated by the honeypots. Each group is assigned a public IP address with port-forwarding enabled on the routers. This way, the honeypots are independent of their network and are exposed to the Internet. All the honeypots, except HosTaGe, run as containers on a system with Ubuntu 18.04 LTS Server. The HosTaGe honeypot is deployed on a rooted Samsung S10 Galaxy device to emulate services on ports below 1024. All the attacks gathered on the honeypots are exported daily and imported into the database. We record the attacks on all the honeypots for one month in April 2021 on a day to day basis. The findings are summarized in the Section 4.

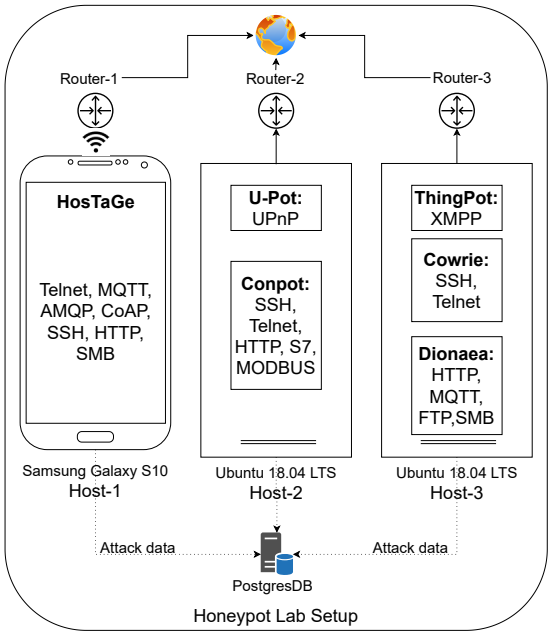

Figure 1: Honeypot experimental setup

3.4 Network-Telescope Analysis

The honeypots deployed in our lab environment provide us with traffic on a limited IP address space. To address this limitation and

get a more holistic view of the attack landscape, we analyze the FlowTuple data from a network telescope. A network telescope is a portion of routed IP address space in which no legitimate traffic exists [14]. Telescopes contain massive data that is captured across large number of routed IP address space. This data helps us to understand the attack landscape across the large network, in addition to the traffic we receive on our honeypots. An analysis of the traffic received on the telescope provides information about the remote network events such as flooding DoS attacks, infection of hosts by Internet worms, and network scanning [54]. Studying these networking events assists us in further understanding the latest scanning and attack trends employed by adversaries. In addition to the data from the honeypots, we analyze the data from the CAIDA UCSD Network-Telescope scanners dataset [14]. The UCSD network telescope consists of a globally routed /8 network that carries almost no legitimate traffic. The captured data provides us with a snapshot of anomalous 'background' traffic to 1/256th of all public IPv4 destination addresses on the Internet. Unlike honeypots, telescopes do not simulate any protocols and hence do not respond to any requests. A significant part of the addresses are unused, and any traffic on this network is potentially suspicious.

The traffic to CAIDA UCSD Network Telescope is captured and offered in three forms; *FlowTuple* data, *Raw pcap* data, and *Aggregated Daily RSDoS Attack Metadata*. The FlowTuple data is captured hourly and consists of elementary information about the suspicious traffic. The information includes source and destination IP address, ports, timestamp, protocol, TTL, TCP flags, IP packet length, TCP-SYN packet length, TCP-SYN window length, packet count, country code, and ASN information [77]. Furthermore, additional metadata like *is_spoofed* and *is_masscan* provide information if the source IP address may be spoofed and if the Masscan tool [29] is used for the scan. The files are stored on a minute basis, and hence there are 1,440 files generated per day. We use the FlowTuple data provided by CAIDA and parse the records for April 2021 and requests targeting the Telnet, AMQP, MQTT, XMPP, CoAP, and UPnP protocols. Furthermore, we analyze and classify the suspicious sources into scanning and malicious traffic based on the results we obtain from our honeypot deployment and the ground truth from threat intelligence repositories GreyNoise [30], and Virustotal [94].

4 RESULTS

This section presents our findings primarily on misconfigured devices on the Internet and the attack trends observed through our honeypots. The section is divided into the results obtained through the Internet-wide scan, honeypot detection and the observations from the deployed honeypots.

4.1 Results from Internet-wide scanning

4.1.1 Exposed devices. Upon scanning the Internet with ZMap [24] for six protocols namely Telnet, MQTT, AMQP, XMPP, UPnP and CoAP, we find a total of 14 million hosts with open ports. We compare our scan results with the Project Sonar [67] Internet-wide scan dataset and Shodan [73]. The total number of unique hosts exposed to the Internet by the protocol identified through our scan is listed in Table 4. The Project Sonar does not provide datasets for AMQP and XMPP protocols.

Protocol	ZMap Scan	Project Sonar	Shodan
AMQP	34,542	NA	18,701
XMPP	423,867	NA	315,861
CoAP	618,650	438,098	590,740
UPnP	1,381,940	395,331	433,571
MQTT	4,842,465	3,921,585	162,216
Telnet	7,096,465	6,004,956	188,291
Total	14,397,929 (14M)	10,759,970 (10M)	1,709,380 (1M)

Table 4: #Exposed systems on the Internet by protocol and source

The number of hosts listed from Project Sonar and Shodan was from the same period when our scans were performed. The total number of exposed hosts detected by our scan is higher than the Project Sonar dataset and Shodan. We argue that this could be because of possible allow-listing performed by these scanning services. Another reason could be that our methodology involves scanning the Internet for multiple ports for one protocol. For example, we perform scans with both ports 23 and 2323 for the Telnet protocol, while Project Sonar performs the scans only with port 23. This leads to having a higher number of detected hosts.

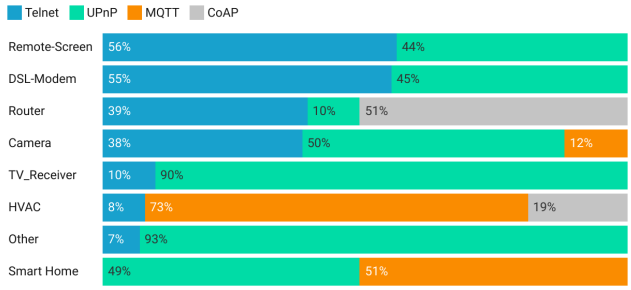

Figure 2: Top IoT device types by protocol (%)

4.1.2 Exposed Device Types. From Table 4, we observe that the number of devices exposing Telnet (7M) is higher than the other protocols. Telnet is highly targeted by botnets to infect with malware. From the banners and the responses received, we attempt to detect the device type. The device types are identified by matching specific text from the banners and the response. For example, the HiKVision Network Camera responds with a banner *"192.0.0.64 login:"* for Telnet connections. The IP address is assigned to the camera as a default configuration and hence responds with this banner [34]. We discover many device types upon performing a similar approach to find consistent banner and response patterns across the scan results. We use the results obtained from the scanning of the protocols to identify device types. We list the major device types and the protocols on which they were detected in Figure 2. We observe that most of the device types are identified through the Telnet and the UPnP responses. The IoT devices were identified with responses from the Telnet, UPnP, MQTT and CoAP protocols. The response received from XMPP and AMQP services were not sufficient to label the target as an IoT device. The basis on which the device types are identified is listed in Appendix-Table

11 for every protocol. Furthermore, other device types like NAS, micro 3D printers and so on are also listed. To facilitate automated detection, we leverage ZTag [22], a tool for annotation of raw data with additional metadata that facilitates tagging and automation of the data from our scans. The banners and static responses are used as metadata for tagging the device types.

Protocol	Vulnerability	#Devices found
CoAP	No auth, admin access	427
AMQP	No auth	2,731
Telnet	No auth	4,013
XMPP	No encryption	5,421
CoAP	No auth	9,067
Telnet	No auth, root access	22,887
MQTT	No auth	102,891
XMPP	Anonymous login	143,986
CoAP	Reflection-attack resource	543,341
UPnP	Reflection-attack resource	998,129
Total		**1,832,893**

Table 5: Total misconfigured devices per protocol

4.1.3 Misconfigured Devices. We consider the misconfigurations for the protocols listed on Tables 2 and 3 for identifying the vulnerable devices. A misconfigured device is a device with no authentication, no encryption, or no authorization configured. We analyze the response received from the scans of all the protocols and find a total of 1, 832, 893 misconfigured devices that satisfy at least one of the conditions. The number of misconfigured devices identified by the protocol are listed in Table 5. The table shows the vulnerability identified in each of the protocols scanned and analyzed by us. In TCP protocols, we see that there are devices exposed with no authentication configured. This means that with a simple connection request, the adversary could connect to the device. There is also a lack of authorization configured in devices that allow the end systems to respond to queries from unknown hosts. Furthermore, we detect many UDP-based devices that respond to discovery queries and can be leveraged in denial of service attacks. We further discuss this type of attack in Section 5. Table 10 in the appendix lists the number of misconfigured devices distributed by country on the six protocols. The source location of the attacks are determined by using the ipgeolocation database [40]. We observe a large number of countries including USA (27%), China (13%), Russia (9.1%), Taiwan (8.9%), Germany (7.8%), Philippines(6.2%), UK(5.8%), Brazil (3.3%), India (3.2%), Thailand (2.7%) , Hong Kong (2.7%), South Korea (2.5%), Israel (2.1%), Canada (1.9%), Bangladesh (1.1%), France (0.9%), Japan (0.7%), and other (1.3%).

4.2 Honeypot Detection

The misconfigured devices identified from our methodology could contain honeypots that can lead to poisoned results. We use the honeypot detection approach, described in Subsection 3.2, to filter out the honeypots from our results. To fingerprint honeypots, we initially perform a search for open-source and research-based IoT-based honeypots. We deploy these honeypots in our lab and capture the banners obtained through a Telnet session from the ZMap client. Then, we systematically search the responses received from our scanning process to filter the honeypot instances. Table 6 lists the honeypots detected using the Telnet banners and the response identified from honeypots [1]. Overall, with this approach we were able to detect a total of 8, 192 honeypots. The results are validated on the basis of our previous work on honeypot fingerprinting [75].

Honeypot	Telnet Banner	#Detected Instances
HoneyPy	Debian GNU/Linux 7\r\r\nLogin:	27
Cowrie	\xff\xfd\x1flogin:	3,228
MTPot	\xff\xfb\x03\xff\xfb\x01\xff\xfd\x1f\ xff\xfd\x18\r\nlogin:	194
Telnet IoT Honeypot	\xff\xfd\x01Login: Password: \r\nWelcome to EmbyLinux 3\.13\.0-24-generic\r\n #	211
Conpot	Connected to [00:13:EA:00:00:0]	216
Kippo	SSH-2.0-OpenSSH_5.1p1 Debian-5	47
Kako	BusyBox v1.19.3 (2013-11-01 10:10:26 CST)	16
Hontel	BusyBox v1.18.4 (2012-04-17 18:58:31 CST)	12
Anglerfish	[root@LocalHost tmp]$	4,241
Total		8,192

Table 6: Detected honeypots through Telnet banner signatures

4.3 Attack trends from honeypots and network telescope

4.3.1 Honeypots. We deploy six honeypots as depicted in Figure 1 at our lab environment. The total number of attack events detected by each honeypot by protocol over one month is listed in Table 7. We observe a total of 200, 209 attack events from all the honeypots. Even though any interaction with honeypots is considered an attack, we argue that recurring scans from known sources (e.g. Shodan [73]) can be considered benign traffic. The attack events consist of both benign and malicious traffic. Scanning-service traffic involves internet-wide scanning events from known sources like Shodan [73], Censys [86], Project Sonar [67], BinaryEdge [38], ZoomEye [62], Fofa [81] and educational organizations like RWTH Aachen University [85]. Malicious traffic involves attacks from unknown scanning sources or attacks with malicious payloads. The packets include both scanning probes and malicious payloads.

Scanning service traffic. We perform a reverse lookup of the source IP addresses of the suspicious traffic received on the honeypots. We identify a total of 10, 696 unique IP addresses that are registered to known scanning services shown in Figure 3. Table 7 lists the total unique IP addresses registered to scanning services, detected per honeypot. Figure 3 shows the scanning-services received on each honeypot. It lists the percentage of total scanning traffic distributed between the identified services. The suspicious traffic that does not resolve to the scanning-services is classified as unknown and is not included as a scanning service. Furthermore, we observe that the IPs from the scanning services scan the Internet periodically and thus are recurring, unlike suspicious one-time scans. The prominent scanning services identified are Stretchoid.com [78], Censys, Shodan, Bitsight [12], BinaryEdge [38], Project Sonar [67], Shadow Server [70], Interne TTL [39], Alpha

[1]The Anglerfish honeypot is not open-source, but was detected retrospectively as a result of large number of suspicious static banners observed in the scan results.

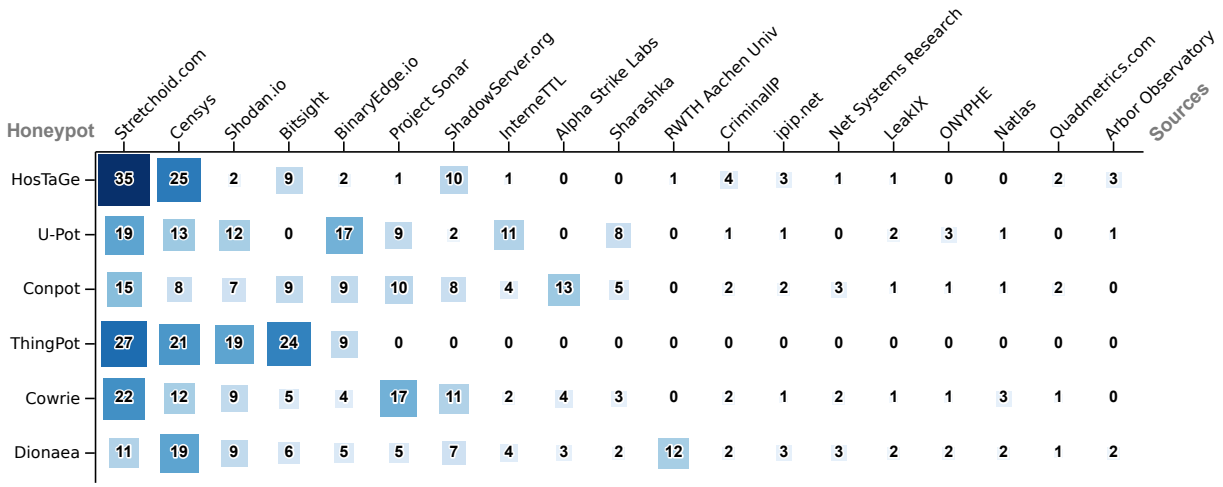

Figure 3: Scanning-service traffic on honeypots (%)

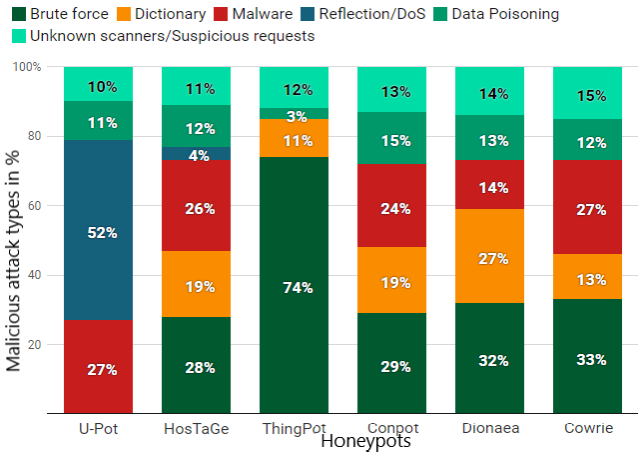

Figure 4: Attack types in different honeypots (%)

Honeypot	Simulated Device Profile	Protocol	#Attack events	Scanning service*	Malicious*	Unknown/ Suspicious*
HosTaGe	Arduino Board with IoT Protocols	Telnet MQTT AMQP CoAP SSH HTTP SMB	19,733 2,511 2,780 11,543 19,174 16,192 1,830	2,866	21,189	2,347
U-Pot	Belkin Wemo smart switch	UPnP	17,101	1,121	7,814	1,786
Conpot	Siemens S7 PLC	SSH Telnet S7 HTTP	12,837 12,377 7,113 11,313	1,678	11,765	1,876
ThingPot	Philips Hue Bridge	XMPP	11,344	967	2,172	963
Cowrie	SSH Server with IoT banner	SSH Telnet	15,459 14,963	2,111	12,874	1,113
Dionaea	Arduino IoT device with frontend	HTTP MQTT FTP SMB	11,974 1,557 3,565 6,873	1,953	13,876	1,694
Total			**200,209**	**10,696**	**69,690**	**9,779**

Table 7: Total attack events by type and protocol on honeypots (* unique source IPs)

Strike Labs [79], Sharashka [71], RWTH Aachen University [85], CriminalIP [18], ipip.net [41], Net Systems Research [66], LeakIX [45], ONYPHE [60], Natlas [56], Quadmetrics.com [65] and Arbor Observatory [7].

Malicious traffic. Since honeypots have no production value, all traffic that is not coming from a known scanning service is considered malicious. These interactions include brute-force attempts, dictionary attacks, malware droppers. Besides, the traffic that does not match the scanning attributes of known scanning tools is malicious. The malware classification is based on the received payloads. The requests are examined for port scans from recognized scanning tools like ZMap. Furthermore, we classify the source as malicious upon receiving recurring requests with malicious payloads. Figure 4 shows the malicious requests received per honeypot and type. We also observe reflection attack attempts on the CoAP and UPnP protocols. The malware attacks listed in Table 7 were classified based on the requested content. The requests included URLs used for downloading the malware and messages with the malicious payload. We also observed data poisoning attacks on the honeypots. For example, there were CoAP requests that changed the data by publishing messages. The malware are identified by analysis of the pcap files stored on the honeypots for unusual content. Upon finding any unusual content, for example a file or script in the payload, we check the file with VirusTotal. Regarding poisoning attacks, we observe if the data has been modified or deleted from the services simulated by the honeypots. For example, we check for any modifications attempted on the data in the MQTT queues. We further discuss some interesting cases in Section 5. The honeypots further encountered non-recurring scanning traffic from unknown sources and suspicious requests that were not identical to any known attack types. Such type of suspicious traffic is grouped under the unknown scanners or suspicious requests.

4.3.2 Network-Telescope: The UCSD CAIDA network telescope consists of 16 million IP addresses. Upon parsing the FlowTuple dataset captured from the telescope, we observe an average of 78

Protocol	Daily Avg. Count	Unique IP	Scanning-service	Unknown/Suspicious
Telnet	2,554,585,920	85,615,200	4,142	85,611,058
UPnP	131,794,560	1,8633	2,279	16,354
CoAP	68,353,920	2,342	627	1,715
MQTT	17,072,640	5,572	1,248	4,324
AMQP	13,907,520	7,132	2,256	4,876
XMPP	6,429,600	4,255	1,973	2,282
Total	**2.7 Bil.**	**85.6 Mil.**	**12525**	**85.6 Mil.**

Table 8: Telescope suspicious traffic classification

billion requests per day. An average of 2.7 billion is targeted towards the Telnet, MQTT, AMQP, CoAP, XMPP, and UPnP protocols. Table 8 shows the average number of suspicious requests received on each protocol daily and the number of IPs that belonged to scanning-services and unknown scanners. We observe that the Telnet protocol dominates the number of suspicious traffic in comparison to the other protocols. This could be because of the presence of many systems infected with malware like Mirai that constantly scan for vulnerable systems on the Internet. For deeper analysis into the attack sources, we check the source IPs to known scanning services and classify them into known and suspicious sources. Table 8 lists the number of known scanning-services and the unknown suspicious scans.

4.3.3 Suspicious traffic classification. We validate our findings on classification of attack sources i.e. scanning services and malicious with [30], and Virustotal [94] databases. Greynoise offers a classification of the attack sources observed on its honeypots into benign, malicious and unknown. The unique source IP addresses of the traffic received on the honeypots and the telescope are searched and corroborated with the classification from Greynoise database. Figure 5 shows the comparison between the total number of attack sources classified as scanning service by our classification and Greynoise. We find that a majority of the sources were identified to be from scanning services by both our method and Greynoise, however, there were 2, 023 IP addresses that were not identified by Greynoise. We also observe that the number of scanning services detected by our method is higher for the AMQP, Telnet and MQTT protocols, which is because we received traffic from multiple cybersecurity risk rating platforms. We suspect that these scans were limited to the European continent or were country-specific.

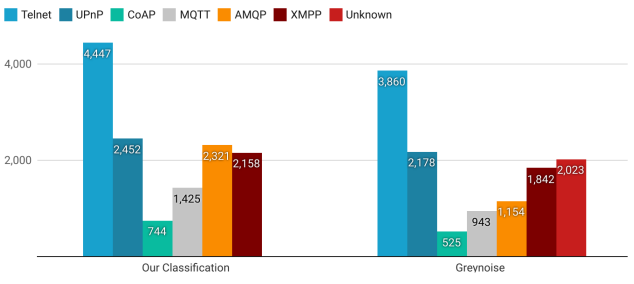

Figure 5: Classification of scanning-services

The source IP addresses are further examined with the VirusTotal threat database. We perform a search of the IP addresses from unknown suspicious requests received on the honeypots and the telescope. Upon performing a search for an IP address, VirusTotal provides a positive score attribute that indicates the number of security vendors that have flagged them as malicious. Note that we consider the IP to be a malicious actor if there is at least one security vendor to label them as malicious (VirusTotal has other labels like phishing). The results are summarized in Figure 6 that lists the percentage of IPs indicated as malicious by protocol as classified by Virustotal. The protocols from the honeypot are indicated by (H) and the telescope as (T). The details about specific malware detected in the traffic are elaborated in Section 5. We observe that the attack sources of the SMB from the honeypots have the highest classification of malicious actors. This is because many well known malware propagate via SMB and hence the detected numbers are higher.

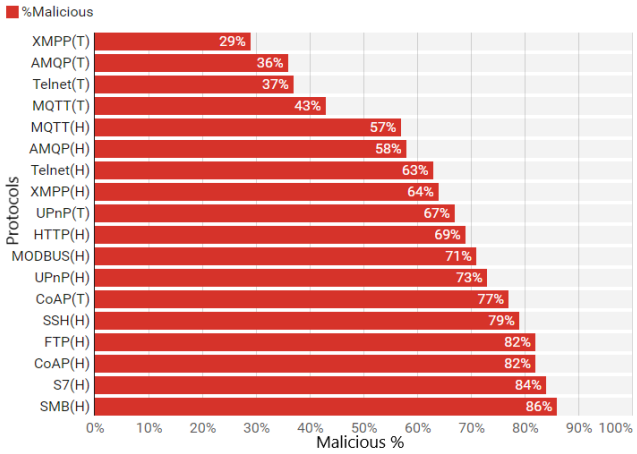

Figure 6: Malware classification by Virustotal (%)

5 DISCUSSION

This section summarizes the attack trends observed from analyzing the attacks on honeypots and the suspicious traffic from the network telescope. We then discuss the impact of listing vulnerable honeypot hosts by scanning services like Shodan. Finally, we investigate the attacks observed from infected hosts and the multistage attacks on honeypots.

5.1 Attack trends by protocol

In the following, we provide an overview of the attack trends on the protocols simulated by the honeypots. In addition to the logs, the network traffic is captured with *tcpdump* on the hosts where the honeypots are deployed and the pcap files are further analyzed to determine the attack vectors. Moreover, we discuss the findings from the analysis of the pcap files from the honeypots by protocol.

5.1.1 Telnet and SSH Attacks. The Telnet protocol (simulated by HosTaGe, Conpot, and Cowrie honeypots) received the highest number of attacks, with a total of 47, 073 attacks, of which 12, 709 were the result of known scanning services. The remaining suspicious traffic received can be further categorized into scans from unknown scanning actors and malware. We examine the pcap files with the Virustotal database for signs of malware signatures and discover 113 Mirai variants. The hashes of the malware identified

are listed in Appendix Table 13. Upon tracing the sources of the malware, we discovered that one of the sources had a valid domain registration as a website for a restaurant in the UK. Beyond Mirai variants, we identified *BrickerBot.2, BrickerBot.1, Hehbot* and *Luabot* malware that brute-force into a target with default credentials. The Appendix Table 12 lists the default most used credentials that were recorded for Telnet and SSH. Moreover, we observe a large number of brute-force attacks with default passwords targeting routers and modems.

The SSH protocol was simulated by HosTaGe, Conpot, and Cowrie honeypots. We observe a high number of brute-force and dictionary attacks on all honeypots. The honeypots received many recent crypto-mining malware like LemonDuck and FritzFrog, among other prominent malware variants. The hash of the malware samples is listed in Appendix Table 13.

5.1.2 MQTT, AMQP and XMPP Attacks. The MQTT protocol was simulated by the HosTaGe and Dionaea honeypots. The attacks mainly aimed at accessing and changing data in the topics. A majority of the attacks tried to access the *'$SYS'* topics. Some attacks tried to poison the data in the topics available while others subscribed to receive messages from specific topics.

The AMQP protocol, simulated by HosTaGe, received similar attacks to that of the MQTT protocol. The adversaries aimed at poisoning the data in the queue through publishing data and subscribing to receive new messages. We also observed a large number of messages published by the adversaries, causing a flood leading to a Denial Of Service.

The XMPP protocol, simulated by the ThingPot honeypot, received brute-force attacks where the adversaries tried to log in to the Philips Hue Bridge system. In addition, we detected some dictionary attacks on the protocol. Lastly, we recorded attempts from malware trying to log in as anonymous users to change the configured state of the lights on the device. We speculate that the malware was trying to examine their write privileges.

5.1.3 CoAP and UPnP attacks. The primary attacks on the CoAP protocol, simulated by HosTaGe, involved discovery requests. However, after the reconnaissance, we observed returning threat actors, especially after being listed on scanning engines like Shodan and Binary Edge (see also Section 5.2). The number of attacks increased, followed by poisoning attacks. Moreover, we detected flooding attacks from unknown malicious actors which resulted in a DoS attack against the honeypot. We observed that the flooding attacks originated from two different sources at the same time. A reverse lookup of the IP addresses showed the existence of duplicate DNS entries for both the IP addresses, which leads to the possibility of reflection or amplification attacks. The webpages of the IPs pointed to an Apache2 Ubuntu Default Page. Other sources of the DoS attacks appeared to originate from Italy, Taiwan, and Brazil.

The U-Pot honeypot received a large number of discovery requests. Following the discovery, there were many DoS attempts recorded on the honeypot. Similar to the attacks on the CoAP protocol, the adversaries performed UDP flood attacks on the honeypot. More than 80% of the total attacks received were a part of the DoS attacks. Two of the adversaries were first observed scanning for the protocol three days before the attack with the same source IP addresses. The source was traced to have a valid domain registration and addressed to a construction service provider in Taiwan.

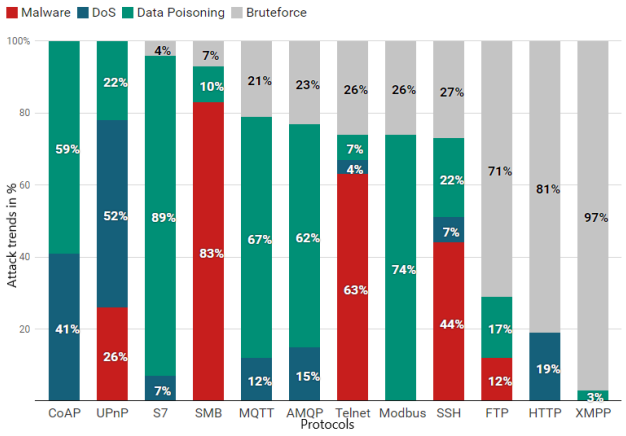

Figure 7: Attack trends by type (%) and protocol

5.1.4 Modbus and S7 attacks. The Modbus and the S7 protocol, simulated by Conpot, received a large number of poisoning attacks where adversaries tried to access and change the values stored in the registers. The attacks targeted three of the nineteen available function codes for reading device identification, the holding register, and the reporting server. Only 10% of the Modbus traffic used valid function codes to access the register data. Furthermore, we observed DoS attacks from attackers that possibly targeted the ICSA-16-299-01 vulnerability for the Siemens S7 protocol [36]. The DoS was performed by flooding the requests with PDU type 1, that results in spawning of a job request in the device.

5.1.5 FTP and SMB attacks. The FTP protocol, simulated by Dionaea, received brute-force and dictionary attacks. In addition, a few attacks deployed malware upon successful authentication to the FTP server. We examined the binary files deployed on the FTP server with Virustotal and found positive results for malware. We discovered multiple deployments of the Mozi and the Lokibot malware. The hash of the malware from Virustotal is listed in Appendix Table 13.

The SMB protocol, simulated by HosTaGe and Dionaea, was largely targeted with the EternalBlue, EternalRomance, and the EternalChampion exploits that attack Microsoft's implementation of the SMB protocol. Among the malware deployed, we find the WannaCry and its variants the most common on the honeypots. The hash of the malware identified via Virustotal is listed on Appendix Table 13.

5.1.6 HTTP attacks. HTTP was simulated by HosTaGe, Conpot, and Dionaea. The honeypots responded with static content and a login page for the simulated device profiles. The protocol was targeted with a large number of web-scraping requests, brute-force, and dictionary attacks. In addition, we observed DoS attacks with HTTP flood packets causing the honeypots to crash. The majority of the DoS attacks came from China, Russia, Israel, USA, and Italy. The attackers also tried to exploit the HTTP protocol by injecting crypto-mining malware. Upon performing a reverse lookup of the

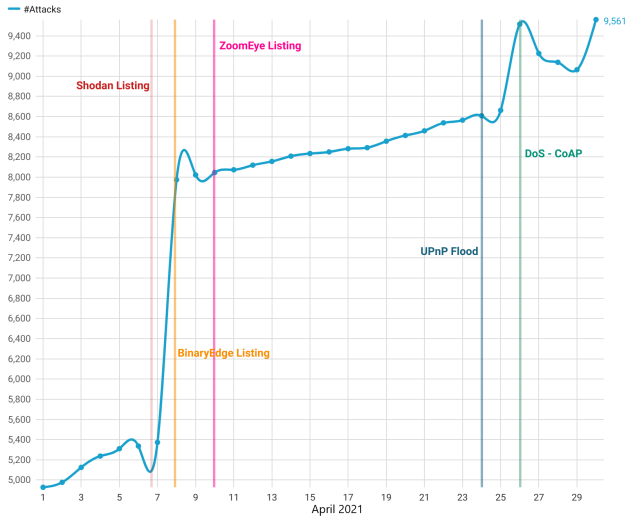

Figure 8: Total attacks by day. We highlight known scanning listings and interesting attack events

attack sources with the Exonerator service [52] we determine a total of 151 unique IPs originating from Tor relays. Furthermore, we observe a daily recurring pattern of scans from these sources and an increasing trend over the month.

5.1.7 Summary. We summarize the attack trends for each protocol emulated by the honeypots for April 2021 in Figure 7. We observe that UDP protocols (CoAP and UPnP) received higher traffic related to Denial of Service in comparison to TCP protocols. Furthermore, the TCP protocols have seen an increase in malware deployment and data poisoning. Our simulated IoT environment suggests that there is an increasing number of attacks concentrating on misusing misconfigured IoT devices.

5.2 Impact of listing by scanning-services

The honeypots received many requests from known scanning-services as listed in Figure 3. We observed an increase in the number of attacks on the honeypots after their listing on scanning-services like Shodan, BinaryEdge and ZoomEye. Figure 8 shows the total number of attacks on the honeypots by day. The attacks include all the requests from scanning-services and other malicious sources. The attacks are distinct by the connection sessions established from the source. The dates at which scanning-services listed the honeypots are also marked in the figure. Furthermore, the figure shows the days on which some major DoS attacks occurred (Day 24, 26). We observe an upward trend in the number of attacks after being listed by scanning-services.

5.3 Attacks from infected hosts

From the results of the honeypots and the network telescope, we observe that there is a large number of attacks originating from unknown sources. Furthermore, from the attack trends, we observe many attempts of malware injections from unknown sources. To determine attack sources originating from infected IoT devices, we search how many of the identified misconfigured devices (see Table 5) are present as attack sources against our honeypots and the telescope. We identify a total of 11, 118[2] unique IP addresses that originate from misconfigured IoT devices. Furthermore, all of the aforesaid IP addresses were flagged as malicious by at least one scanning vendor in Virustotal.

We extend the detection of infected IoT devices by searching the remaining source IP addresses in the Censys database [15]. The Censys database has a labelled dataset of IoT devices and returns an "*iot*" tag if the IP address was identified as an IoT device from its periodic Internet-wide scans. We identify an additional 1, 671[3] IoT devices from the Censys database. A further analysis to determine the type of these IoT devices reveals that the majority of the attacks originate from cameras, routers and IP phones.

Lastly, we extend the search for attacks from infected hosts from non-IoT devices. Upon performing a simple reverse lookup of all the source IP addresses, we discover a total of 797 registered domains of which 427 have a webpage. The domains were looked up to see if they served additional on additional IP addresses than the one discovered from our analysis. We found the domains registered with /30 and /29 subnets with some unused IP addresses. From this analysis, we also infer that some of the Telnet malware injections originated from an infected URL serving HTML. Upon searching Virustotal for these URLs, we find 346 of them tagged as malicious. The webpages were found serving default wordpress sites, Ubuntu Apache test pages, static ad pages and fake online shopping portals.

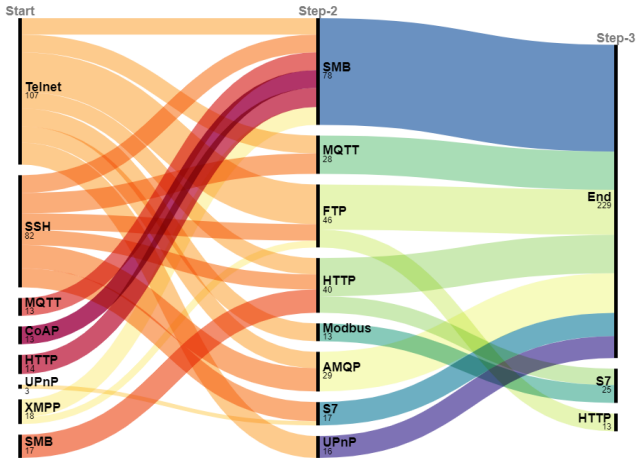

Figure 9: Multistage attacks detected on honeypots

5.4 Multistage attacks in honeypots

We define *multistage attacks* as attacks in which there is a pattern of multiple protocols that are being sequentially attacked by the same adversary. Attackers may employ the multistage attack strategy to amplify an attack or to find alternate sources for malware injection. Although such types of traffic may be observed from scanning services, we filter such sources by checking if they are registered to

[2] In more details, 1, 147 attacked only the honeypots, 1, 274 attacked only the telescope, and 8, 697 both of them.
[3] In particular, 439 attacked only the honeypots, 564 attacked only the telescope, and 668 both of them.

a domain affiliated to a scanning service. The HosTaGe honeypot offers the detection of multistage attacks as a service. For the other honeypots, we group the attacks from distinct source IP addresses and check if multiple protocols are targeted. We note that the attacks are grouped based on the source IP addresses and the time interval between attacks is not taken into account. This entails that a follow up attack from the same adversary may have occurred anytime in the one month experiment period.

We list the protocols targeted by attackers in the identified multistage attacks across the honeypots in Figure 9. The figure depicts the protocols targeted step-wise. The numbers below the protocol indicate the total number of attacks received on that protocol at that stage and the thickness of the bars indicate the amplitude of the attacks. A total of 267 multistage attacks were detected and we observe that the majority of them initiated with Telnet and SSH. Furthermore, the SMB is noticed to be receiving most of the attacks at the second step and the S7 protocol in step three.

6 CONCLUSION

With this work, we combine the search for misconfigured IoT devices on the Internet with an analysis of attack trends in the IoT. To the best of our knowledge, our work is the first to combine the results of a complete IPv4 scan with knowledge gained by honeypot deployment and network telescope data. Beyond the large number of attacks that we received and analyzed, we show that many of the misconfigured devices take themselves the role of the attacker as part of malware propagation campaigns.

In particular, our scans reveal that there is a large number of misconfigured IoT devices that can be leveraged to perform diverse type of attacks on the Internet. Furthermore, the attacks received on the honeypots suggest a trend in attackers searching for vulnerable IoT devices. This is supported by the network telescope data that suggest a global trend. The attacks received from infected IoT hosts show that high magnitude attacks are possible, specifically with devices running CoAP and UPnP. Through this work, we aim at creating awareness about the implications of the misconfigurations of IoT devices by exploring such devices that are making us of six popular protocols. In fact, it is worth noting that by intersecting all of our experiments (IPv4 scanning, network telescope and honeypots) we are able to identify 11, 118 misconfigured IoT systems that are actively attacking the Internet; simultaneously 1.8 million devices are potentially waiting to be exploited by adversaries.

In comparison to previous work on Internet-wide scanning [50, 74], we use custom probes that scan for specific IoT protocols and further use open datasets to verify our findings from the scan. We identify a large of number of misconfigured IoT devices based on specific banner-based and response-based indicators. While Markowsky et al. [50] demonstrate how to scan and find vulnerable devices using Shodan and Masscan, they do not specifically search for misconfigured IoT devices. Our results confirm the methodology of [74], which combines scanning the Internet and deploying honeypots to study the attack trends on the FTP protocol. We instead focus on 6 protocols that are used in IoT. We enhance our methodology by using the data from a network telescope as with Neshenko st al. [57], who use the data to support their proposed taxonomy of IoT vulnerabilities. Furthermore, our work highlights the need for sanitization of Internet-scan data from honeypots. In this context, we identify 8,192 honeypots that would otherwise be classified as misconfigured IoT systems. While individual work on honeypot fingerprinting has shed light into this field [75, 92], no previous work on the Internet measurements has taken honeypots into account.

To summarize our contributions, we scan the Internet, specifically to find misconfigured IoT devices by the use of custom probes on 6 protocols (TCP and UDP). We verify the results from our scan by validating them with open datasets on Internet-scanning. We filter out potential honeypots from our scanning results by using our multistage honeypot fingerprinting techniques [75] to avoid poisoning of the results. Lastly, we deploy 6 IoT honeypots that emulate misconfigurations observed from IoT devices in our scan. Furthermore, the analysis of the data from the telescope compliments our observations on the attacks received on honeypots.

With regard to future work, we plan to extend the scanning scope of protocols to include TR069, SMB, and industrial IoT protocols like DDS and OPC UA. The analysis from the network telescope also motivates us to perform a deeper analysis on raw packet data to uncover new threat actors on Industrial IoT devices and protocols. Lastly, based on the recent work of Wan et al. [96] we see the need for combining geographically distributed scanners, especially for certain protocols (e.g. SSH).

ACKNOWLEDGEMENTS

This research was supported as part of COM^3, an Interreg project supported by the North Sea Programme of the European Regional Development Fund of the European Union.

REFERENCES

[1] Netlab 360. 2021. Anglerfish Honeypot. (2021). https://blog.netlab.360.com/tag/anglerfish-honeypot/
[2] abuse.ch. 2021. MalwareBazaar. (2021). https://bazaar.abuse.ch/
[3] Peter Adkins. 2017. Kako Honeypot. (2017). https://github.com/darkarnium/kako
[4] Syaiful Andy, Budi Rahardjo, and Bagus Hanindhito. 2017. Attack scenarios and security analysis of MQTT communication protocol in IoT system. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE, Yogyakarta, Indonesia, 1–6. https://doi.org/10.1109/EECSI.2017.8239179
[5] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1093–1110. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
[6] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1093–1110. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
[7] Arbor-Bbservatory. 2021. arbor-observatory. (2021). https://www.arbor-observatory.com/
[8] S. Arvind and V. A. Narayanan. 2019. An Overview of Security in CoAP: Attack and Analysis. In *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*. IEEE, Coimbatore, India, 655–660. https://doi.org/10.1109/ICACCS.2019.8728533
[9] Remillano II Augusto, Noel Collado Patrick, and Ivy Titiwa Karen. 2020. XORDDoS, Kaiji Variants Target Exposed Docker Servers. (2020). https://www.trendmicro.com/en_us/research/20/f/xorddos-kaiji-botnet-malware-variants-target-exposed-docker-servers.html

[10] Leonardo Babun, Kyle Denney, Z. Berkay Celik, Patrick McDaniel, and A. Selcuk Uluagac. 2021. A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks* 192 (2021), 108040. https://doi.org/10.1016/j.comnet.2021.108040

[11] Belkin. 2021. Belkin Wemo. (2021). https://www.belkin.com/us/

[12] Bitsight.com. 2021. Bitsight.com. (2021). https://www.bitsight.com/

[13] Alex Burt. 2020. Gitpaste-12: a new worming botnet with reverse shell capability spreading via GitHub and Pastebin. (2020). https://blogs.juniper.net/en-us/threat-research/gitpaste-12

[14] CAIDA. 2021. The CAIDA UCSD Network Telescope "Darknet Scanners" Dataset - April-May2021. (2021). https://www.caida.org/data/passive/telescope-darknet-scanners_dataset.xml

[15] Censys. 2021. Censys Search. (2021). Retrieved May 24, 2021 from https://censys.io/

[16] Douligeris Christos, Raghimi Omid, Barros Lourenço Marco, and Marinos Louis. 2020. ENISA Threat Landscape 2020 - Emerging Threats. *ENISA* ETL2020 (2020), 8–10. https://www.enisa.europa.eu/publications/emerging-trends

[17] Cloudflare. 2021. SSDP DDoS Attack. (2021). https://www.cloudflare.com/learning/ddos/ssdp-ddos-attack/

[18] CriminalIP. 2021. CriminalIP. (2021). https://security.criminalip.com/

[19] Cymmetria. 2016. MTPot. (2016). https://github.com/Cymmetria/MTPot

[20] Decester. 2000. An SSH Honeypot. (2000). https://github.com/desaster/kippo

[21] Zakir Durmeric. 2018. zgrab2. (2018). https://github.com/zmap/zgrab2

[22] Zakir Durumeric. 2017. ZTag. (2017). https://github.com/zmap/ztag

[23] Zakir Durumeric, Michael Bailey, and J. Alex Halderman. 2014. An Internet-Wide View of Internet-Wide Scanning. In *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, San Diego, CA, 65–78. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/durumeric

[24] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *22nd USENIX Security Symposium (USENIX Security 13)*. USENIX Association, Washington, D.C., 605–620. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

[25] fireHOL. 2021. Europe Blocklist. (2021). https://github.com/firehol/blocklist-ipsets/blob/master/ip2location_country/ip2location_continent_eu.netset

[26] Apache Foundation. 2021. Apache ActiveMQ. (2021). https://activemq.apache.org/

[27] Apache Foundation. 2021. Apache Qpid. (2021). https://qpid.apache.org/

[28] Jazib Frahim, Carlos Pignataro, Jeff Apcar, and Monique Morrow. 2015. Securing the internet of things: A proposed framework. (2015).

[29] Robert David Graham. 2014. MASSCAN: Mass IP port scanner. (2014).

[30] GreyNoise. 2021. GreyNoise. (2021). https://viz.greynoise.io/

[31] hackingump. 2020. UpnP – Messing up Security since years. (2020). https://malwareandstuff.com/upnp-messing-up-security-since-years/

[32] Muhammad A. Hakim, Hidayet Aksu, A. Selcuk Uluagac, and Kemal Akkaya. 2018. U-PoT: A Honeypot Framework for UPnP-Based IoT Devices. In *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, Orlando, FL, USA, 1–8. https://doi.org/10.1109/PCCC.2018.8711321

[33] Hwanjo Heo and Seungwon Shin. 2018. Who is Knocking on the Telnet Port: A Large-Scale Empirical Study of Network Scanning. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS '18)*. Association for Computing Machinery, New York, NY, USA, 625–636. https://doi.org/10.1145/3196494.3196537

[34] HKVision. 2021. HKVision Network Camera - User Manual. (2021). https://www.hikvision.com/UploadFile/image/EN-user%20manual%20of%20%20network%20camera%20v3.0.0.pdf

[35] T. Holz and F. Raynal. 2005. Detecting honeypots and other suspicious environments. In *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*. IEEE, West Point, NY, USA, 29–36. https://doi.org/10.1109/IAW.2005.1495930

[36] ICSA. 2016. CISA-ICSA-16-299-01. (2016). https://us-cert.cisa.gov/ics/advisories/ICSA-16-299-01

[37] Fraunhofer IKE. 2021. Malpedia. (2021). https://malpedia.caad.fkie.fraunhofer.de/

[38] Coalition Inc. 2021. BinaryEdge. (2021). https://www.binaryedge.io/

[39] InterneTTL. 2021. InterneTTL. (2021). http://www.internettl.org/

[40] ipgeolocation. 2021. ipgeolocation.io. (2021). ipgeolocation

[41] ipip.net. 2021. ipip.net. (2021). https://en.ipip.net/

[42] Philipp Jeitner. 2018. Telnet IoT Honeypot. (2018). https://github.com/Phype/telnet-iot-honeypot

[43] Mattijs Jonker, Alistair King, Johannes Krupp, Christian Rossow, Anna Sperotto, and Alberto Dainotti. 2017. Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem. In *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. Association for Computing Machinery, New York, NY, USA, 100–113. https://doi.org/10.1145/3131365.3131383

[44] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. 2017. DDoS in the IoT: Mirai and Other Botnets. *Computer* 50, 7 (2017), 80–84. https://doi.org/10.1109/MC.2017.201

[45] LeakIX. 2021. LeakIX. (2021). https://leakix.net/

[46] Tongbo Luo, Z. Xu, Xing Jin, Y. Jia, and Xin Ouyang. 2017. IoTCandyJar : Towards an Intelligent-Interaction Honeypot for IoT Devices. (2017), 11 pages.

[47] Gordon Lyon. 2021. NMap Network Mapper. (2021). https://nmap.org/

[48] Phillip Maddux. 2019. HoneyPy Honeypot. (2019). https://github.com/foospidy/HoneyPy

[49] Malwaremustdie. 2020. Rhombus - Linux DDoS botnet aims VPS & IoT, w/persistence & dropper. (2020). https://otx.alienvault.com/pulse/5e6aacfe61b118f3fc41026a

[50] Linda Markowsky and George Markowsky. 2015. Scanning for vulnerable devices in the Internet of Things. In *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 1. IEEE, Warsaw, Poland, 463–467. https://doi.org/10.1109/IDAACS.2015.7340779

[51] Ian Mcateer, Muhammad Imran Malik, Z. Baig, and P. Hannay. 2017. Security vulnerabilities and cyber threat analysis of the AMQP protocol for the internet of things. In *Australian Information Security Management Conference*. Edith Cowan University, Perth, W.A., 11.

[52] Tor Metrics. 2021. ExoneraTor. (2021). https://metrics.torproject.org/exonerator.html

[53] MITRE. 2021. Common Vulnerabilities and Exposures. (2021). Retrieved May 24, 2021 from https://cve.mitre.org/

[54] David Moore. 2002. Network Telescopes: Observing Small or Distant Security Events. In *11th USENIX Security Symposium (USENIX Security 02)*. USENIX Association, San Francisco, CA, 9. https://www.usenix.org/conference/11th-usenix-security-symposium/network-telescopes-observing-small-or-distant-security

[55] Shun Morishita, Takuya Hoizumi, Wataru Ueno, Rui Tanabe, Carlos Gañán, Michel JG van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. 2019. Detect me if you… oh wait. An internet-wide view of self-revealing honeypots. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, IEEE, Washington DC, USA, 134–143.

[56] Natlas. 2021. Natlas. (2021). https://github.com/natlas/natlas

[57] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. 2019. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Communications Surveys Tutorials* 21, 3 (2019), 2702–2733. https://doi.org/10.1109/COMST.2019.2910750

[58] NIST. 2021. misconfiguration. (2021). https://csrc.nist.gov/glossary/term/misconfiguration

[59] NIST. 2021. NATIONAL VULNERABILITY DATABASE. (2021). https://nvd.nist.gov/

[60] Onyphe. 2021. Onyphe. (2021). https://www.onyphe.io/

[61] Michel Oosterhof. 2016. Cowrie SSH/telnet honeypot. (2016). https://github.com/micheloosterhof/cowrie

[62] ZoomEye Org. 2021. ZoomEye. (2021). https://www.zoomeye.org/

[63] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. 2015. IoTPOT: Analysing the Rise of IoT Compromises. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. USENIX Association, Washington, D.C., 9. https://www.usenix.org/conference/woot15/workshop-program/presentation/pa

[64] The Honeynet Project. 2021. The Honeynet Project. (2021).

[65] Quadmetrics. 2021. Quadmetrics. (2021). https://www.quadmetrics.com/

[66] Net Systems Research. 2021. Net Systems Research. (2021). https://www.netsystemsresearch.com/

[67] Rapid7 Research. 2021. Project Sonar. (2021). https://www.rapid7.com/research/project-sonar/

[68] Philipp Richter and Arthur Berger. 2019. Scanning the scanners: Sensing the Internet from a massively distributed network telescope. In *Proceedings of the Internet Measurement Conference*. Association for Computing Machinery, New York,NY,United States, Amsterdam, Netherlands, 144–157.

[69] Lukas Rist, Johnny Vestergaard, Daniel Haslinger, A Pasquale, and J Smith. 2013. Conpot ics/scada honeypot. (2013).

[70] ShadowServer.org. 2021. ShadowServer.org. (2021). https://www.shadowserver.org/

[71] Sharashka. 2021. Sharashka. (2021). https://sharashka.io/data-feeds

[72] Hajime Shimada, Katsutaka Ito, Hirokazu Hasegawa, and Yukiko Yamaguchi. 2019. Implementation of MQTT/CoAP Honeypots and Analysis of Observed Data. *SECURWARE 2019, The Thirteenth International Conference on Emerging Security Information, Systems and Technologies* 10 (2019), 35–40.

[73] SHODAN. 2021. Shodan. (2021). https://www.shodan.io/

[74] Drew Springall, Zakir Durumeric, and J Alex Halderman. 2016. FTP: The forgotten cloud. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, IEEE, Toulouse, France, 503–513.

[75] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. 2021. Gotta catch 'em all: a Multistage Framework for honeypot fingerprinting. (2021). arXiv:cs.CR/2109.10652

[76] Miroslav Stampar. 2017. HonTel Honeypot. (2017). https://github.com/stamparm/hontel

[77] CAIDA STARDUST. 2021. Flow Level Traffic (FlowTuple). (2021). https://stardust-dev.caida.org/docs/data/flowtuple/

[78] Stretchoid.com. 2021. Stretchoid.com. (2021). http://stretchoid.com/

[79] Alpha Strike. 2021. Alpha Strike. (2021). https://www.alphastrike.io

[80] Oleg Surnin, Fatima Hussain, Rasheed Hussain, Svetlana Ostrovskaya, Andrey Polovinkin, JooYoung Lee, and Xavier Fernando. 2019. Probabilistic Estimation of Honeypot Detection in Internet of Things Environment. In *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Honolulu, HI, USA, 191–196. https://doi.org/10.1109/ICCNC.2019.8685566

[81] FOFA Cyberspace Surveying and Mapping. 2021. Fofa. (2021). https://fofa.so/

[82] Madiha H. Syed, Eduardo B. Fernandez, and Julio Moreno. 2018. A Misuse Pattern for DDoS in the IoT. In *Proceedings of the 23rd European Conference on Pattern Languages of Programs (EuroPLoP '18)*. Association for Computing Machinery, New York, NY, USA, Article 34, 5 pages. https://doi.org/10.1145/3282308.3282343

[83] Dino Tools. 2010. Web Honeypot. (2010). https://github.com/DinoTools/dionaea/

[84] Gray Hat Tools. 2021. PFT Printer Exploration. (2021). http://www.phenoelit.org/fr/tools.html

[85] Communication & Distributed Systems RWTH Aachen University. 2021. RWTH Aachen Scan. (2021). http://researchscan.comsys.rwth-aachen.de/

[86] Stanford University. 2021. Censys Universal IPv4 Internet Dataset. (2021). https://scans.io/

[87] Ivan Vaccari, Maurizio Aiello, and Enrico Cambiaso. 2020. SlowITe, a Novel Denial of Service Attack Affecting MQTT. *Sensors* 20, 10 (2020). https://doi.org/10.3390/s20102932

[88] Ivan Vaccari, Maurizio Aiello, and Enrico Cambiaso. 2020. SlowTT: A Slow Denial of Service against IoT Networks. *Information* 11, 9 (2020). https://doi.org/10.3390/info11090452

[89] Emmanouil Vasilomanolakis, Shankar Karuppayah, Mathias Fischer, Max Mühlhäuser, Mihai Plasoianu, Lars Pandikow, and Wulf Pfeiffer. 2013. This Network is Infected: HosTaGe - a Low-Interaction Honeypot for Mobile Devices. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM '13)*. Association for Computing Machinery, New York, NY, USA, 43–48. https://doi.org/10.1145/2516760.2516763

[90] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max Mühlhäuser, and Mathias Fischer. 2014. HosTaGe: A Mobile Honeypot for Collaborative Defense. In *Proceedings of the 7th International Conference on Security of Information and Networks (SIN '14)*. Association for Computing Machinery, New York, NY, USA, 330–333. https://doi.org/10.1145/2659651.2659663

[91] Alan Tamer Vasques and João J. C. Gondim. 2020. Amplified Reflection DDoS Attacks over IoT Reflector Running CoAP. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, Seville, Spain, 1–6. https://doi.org/10.23919/CISTI49556.2020.9140882

[92] Alexander Vetterl and Richard Clayton. 2018. Bitter Harvest: Systematically Fingerprinting Low- and Medium-interaction Honeypots at Internet Scale. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. USENIX Association, Baltimore, MD, 9. https://www.usenix.org/conference/woot18/presentation/vetterl

[93] Benjamin Vignau, Raphaël Khoury, Sylvain Hallé, and Abdelwahab Hamou-Lhadj. 2021. The evolution of IoT Malwares, from 2008 to 2019: Survey, taxonomy, process simulator and perspectives. *Journal of Systems Architecture* 116 (2021), 102143. https://doi.org/10.1016/j.sysarc.2021.102143

[94] Virustotal. 2021. Virustotal. (2021). https://www.virustotal.com

[95] VMWare. 2021. RabbitMQ. (2021). https://www.rabbitmq.com/

[96] Gerry Wan, Liz Izhikevich, David Adrian, Katsunari Yoshioka, Ralph Holz, Christian Rossow, and Zakir Durumeric. 2020. On the Origin of Scanning: The Impact of Location on Internet-Wide Scans. In *Proceedings of the ACM Internet Measurement Conference (IMC '20)*. Association for Computing Machinery, New York, NY, USA, 662–679. https://doi.org/10.1145/3419394.3424214

[97] Gerry Wan, Liz Izhikevich, David Adrian, Katsunari Yoshioka, Ralph Holz, Christian Rossow, and Zakir Durumeric. 2020. On the Origin of Scanning: The Impact of Location on Internet-Wide Scans. In *Proceedings of the ACM Internet Measurement Conference (IMC '20)*. Association for Computing Machinery, New York, NY, USA, 662–679. https://doi.org/10.1145/3419394.3424214

[98] Jianxin Wang, Ming K. Lim, Chao Wang, and Ming-Lang Tseng. 2021. The evolution of the Internet of Things (IoT) over the past 20 years. *Computers & Industrial Engineering* 155 (2021), 107174. https://doi.org/10.1016/j.cie.2021.107174

[99] M. Wang, Javier Santillan, and F. Kuipers. 2018. ThingPot: an interactive Internet-of-Things honeypot. (2018). arXiv:arXiv:1807.04114

[100] ZMap. 2020. ZMap Block and Allow Lists. (2020). https://github.com/zmap/zmap/wiki/Block-and-Allow-Lists

A APPENDIX

A.1 Scanning dates by protocol

The Internet-wide scans on all the 6 protocols were performed in a span of one week. Table 9 lists the dates on which the scans were started for the corresponding protocols.

Protocol	Scan Date
CoAP	1 March 2021
UPnP	2 March 2021
Telnet	2 March 2021
MQTT	4 March 2021
AMQP	4 March 2021
XMPP	5 March 2021

Table 9: Scan dates per protocol

A.2 Misconfigured IoT devices by country

Based on our scan, we detect a total of 1, 832, 893 misconfigured devices over the response received from six protocols. Table 10 lists the distribution of misconfigured devices by country.

Country	Count
USA	494,881 (27%)
China	238,276 (13%)
Russia	166,793 (9.1%)
Taiwan	163,127 (8.9%)
Germany	142,966 (7.8%)
Philippines	113,639 (6.2%)
UK	106,308 (5.8%)
Brazil	60,485 (3.3%)
India	58,653 (3.2%)
Thailand	49,488 (2.7%)
Hong Kong	45,822 (2.5%)
South Korea	45,822 (2.5%)
Israel	38,491 (2.1%)
Canada	34,825 (1.9%)
Other countries	23,828 (1.3%)
Bangladesh	20,162 (1.1%)
France	16,496 (0.9%)
Japan	12,830 (0.7%)
Total	**1,832,893**

Table 10: Misconfigured devices by country

A.3 Ethical Considerations

The Internet-wide scans were performed through a set of dedicated IP address provided in the University network. The motivation to perform our own scans with ZMap is because some networks blocklist Shodan, Censys and other scanning services. However, we wanted to include datasets from scanning-services to cover the networks that may have blocked our scanning IP. Furthermore, because of the CoAP and the UPnP protocols in our scanning portfolio, we ran custom scripts to fetch specific response from the hosts that helps us in identification of misconfigured devices. Moreover a recent study shows the impact of location on Internet-wide scans, which presents certain limitations of scanning services [97]. We were motivated by this study to perform our own scans.

Information regarding the misconfigured devices, like the source IP addresses are not shared or disclosed. The data will be stored for a period of three months from the date of collection, followed by anonymization of IP addresses to follow the local privacy regulations. Furthermore, a webpage stating the purpose of the scan and research was setup to ensure transparency and indicate intent of the scanning process. The scans included a default blocklist from the ZMap repository [100] and the European blocklisted provided by the FireHOL project [25].

The samples identified by Virustotal as malware will be shared on online threat repositories like Malpedia [37] and Malware Bazaar [2] to facilitate research from the open source community. Lastly, the destination IP addresses in the UCSD CAIDA network telescope have not been disclosed or shared and are anonymized in our database to facilitate the purposes of the telescope.

A.3.1 Honeypot sandboxing. We want to emphasize that our setting focused only on collecting attacks from the Internet and in principle did not allow for honeypots to attack back a system or entity. Furthermore, we use state of the art honeypots (HosTaGe, Conpot, Cowrie, Dionaea, ThingPot and U-Pot) for which, to the best of our knowledge, there is no scientific publication suggesting the possibility of an adversary being able to hack their way out of them and attack systems on the Internet through them. Moreover, we want to highlight that we are only utilizing low/medium interaction honeypots. In contrast to high interaction honeypots, which are real systems and thus may be compromised, low/medium interaction honeypots only partially emulate protocols. Adding to this, each honeypot (except HosTaGe that runs on a mobile device) was deployed as a container for better managing and as an additional security layer. HosTaGe is safeguarded by the device's firmware (Samsung's Linux Container (LXC) sandboxing). Furthermore, HosTaGe's implementation of the various protocols does not allow the attacker for a lot of interaction (the reason for this, as with most low interaction honeypots, is the utilization of protocol emulation libraries that are incomplete it terms of capabilities). In regards to measures against reflection attacks (i.e., on CoAP and UPnP) we would like to note the following. The CoAP implementation of HosTaGe is implemented using the *JAVA mbed-coap* library and only responds to service discovery requests with static information. Hence, it does not allow for an attacker to attack other devices. Similarly, for U-Pot we utilized a low interaction image of the IoT device that responds to only service discovery requests (by using a limited UPnP library, i.e., GUPNP).

In addition, for all the honeypots, we performed continuous monitoring on a daily basis. That is, we examined what kind of attacks and communication was taking place and whether anything looked overly suspicious. Moreover, note that all containers had egress rules to limit any traffic attempting to leave the network. As we write on the paper, the majority of the observed attacks come as the result of automated attacks (e.g. via malware). Lastly, the IP space used for the honeypots is part of our monitored university network; we can confirm that we have not received any complaints with regard to the IP addresses of the honeypots neither from our NOC nor our ISP.

A.4 Most common Device-type identifiers with banners/response

Device	Protocol	Device-Type	Banner/Response
HiKVision Camera	Telnet	Camera	"192.168.0.64 login:"
Polycom HDX	Telnet	Camera	"Welcome to ViewStation"
D-Link DCS-6620	Telnet	Camera	"Welcome to DCS-6620"
D-Link DCS-5220	Telnet	Camera	"Network-Camera login:"
Avtech AVN801	UPnP	Camera	"Server: Linux/2.x UPnP/1.0 Avtech/1.0"
Panasonic BB-HCM581	UPnP	Camera	"Friendly Name: Network Camera BB-HCM581"
Anbash NC336FG	UPnP	Camera	"Model Name: NC336FG"
Beward N100	UPnP	Camera	"Friendly Name: N100 H.264 IP Camera - 004B1000E3E2"
Io Data TS-WLC2	UPnP	Camera	"Model Name: TS-WLC2"
Io Data TS-WPTCAM	UPnP	Camera	"Model Name: TS-WPTCAM"
Io Data TS-WLCAM	UPnP	Camera	"Model Name: TS-WLCAM"
Io Data TS-WLCE	UPnP	Camera	"Model Name: TS-WLCE"
G-Cam EFD-4430	UPnP	Camera	Friendly Name: G-Cam/EFD-4430
Seyeon Tech FW7511-TVM	UPnP	Camera	"Model Name: FW7511-TVM"
ZyXEL PK5001Z	Telnet	DSL Modem	"PK5001Z login"
ZTE ZXHN H108N	Telnet	DSL Modem	"Welcome to the world of CLI"
Technicolor modem	Telnet	DSL Modem	"TG234 login:"
ZTE ZXV10	Telnet	DSL Modem	"F670L Login"
Datacom DM991	Telnet	DSL Modem	"DM991CR - G.SHDSL Modem Router"
TP-Link TD-W8960N	Telnet	DSL Modem	"TD-W8960N 6.0 DSL Modem"
Cisco C11-4P	Telnet	DSL Modem	"MODEM : C111-4P"
TP-Link TD-W8968	Telnet	DSL Modem	"TD-W8968 4.0 DSL Modem Router"
BelAir 100N	Telnet	Router	"BelAir100N - BelAir Backhaul and Access Wireless Router"
Tenda Wireless Router	UPnP	Router	"Manufacturer: Tenda"
Totolink N150	UPnP	Router	"Friendly Name: TOTOLINK N150RA"
ZTE H108N	UPnP	Router	"Model Name: H108N"
OBSERVA BHS_RTA 1.0.0	UPnP	Router	"Model Name: BHS_RTA"
DASAN H660GM	UPnP	Router	"Model Name: H660GM"
Huawei HG532e	UPnP	Router	"Model Name: HG532e"
ASUSTeK RT-AC53	UPnP	Router	"Friendly Name: RT-AC53"
NDM	CoAP	Router	"/ndm/login"
QLink	CoAP	Router	title: Qlink-ACK Resource
Signify Philips hue bridge	UPnP	Smart home	"Model Name: Philips hue bridge 2015"
EQ3 HomeMatic	UPnP	Smart Home	"Model Name: HomeMatic Central"
Hyperion 2.0.0	UPnP	Smart Home	"Model Description: Hyperion Open Source Ambient Light"
Home Assistant	Telnet	Smart Home	"Home Assistant: Installation Type: Home Assistant OS"
Home Assistant	MQTT	Smart Home	"homeassistant/light/"
Emby	UPnP	TV Receiver	"Friendly Name: Emby - DS720plus"
Dedicated Micros Digital Sprite 2	Telnet	TV Receiver	"Welcome to the DS2 command line processor"
Roku	UPnP	TV Receiver	"Server: Roku UPnP/1.0 MiniUPnPd/1.4"
Realtek RTL8671	UPnP	Access Point	"Model Name: RTL8671"
Synology DS918+	UPnP	NAS	"Friendly Name: DiskStation (DS918+)"
Sonos ZP100	UPnP	Smart Speaker	"Model Number: ZP120"
Octoprint	MQTT	3D Printer	"octoPrint/temperature/bed"
Gozmart	MQTT	HVAC	"gozmart/sonoff/CC50E3C943CC110511/app"
Advantech	MQTT	HVAC	"Advantech/"
Emerson	Telnet	Remote Display Unit	"Emerson Network Power Co., Ltd."
Trimble SPS855	UPnP	Remote Display Unit	"Friendly Name: SPS855, 6013R31531: Trimble"

Table 11: Most common device-types with identifiers in banners/response

A.5 Top Telnet and SSH credentials used by count

Protocol	Credentials	Count
Telnet	admin,admin	9,772
Telnet	root,root	1,721
Telnet	root,admin	1,254
Telnet	telnet,telnet	689
Telnet	root,xc3511	556
Telnet	admin,admin123	467
Telnet	root,12345	456
Telnet	user,user	321
Telnet	admin,12345	267
Telnet	admin,polycom	217
Telnet	admin,(blank)	198
SSH	admin, admin	11,543
SSH	root, root	3,432
SSH	root, admin	1,943
SSH	zyfwp, PrOw!aN_fXp	1538
SSH	cisco, cisco	629
SSH	cisco, cisco	629
SSH	admin, ssh1234	254

Table 12: Top Telnet and SSH credentials used by adversaries

A.6 SHA256 Hash of Malware variants

SlNo	SHA256 Hash	Malware Variant Type
1	27870ada242e0f7fd5b1e7fc799f503004b3fd2c0f971784208cae31880b9950	Mirai
2	f05b1018a6fb23154885f55e27a7d20c36c186df5f4d08bd061a5666fdb05be9	Mirai
3	ad9d20dd5159975e4c192a335a41eabcb0bc10e3110d894416a025ac9955f7e7	Mirai
4	dd86acf2bd99afd9da305bb9a4c3da320df617e36f53f206fcf161c04152eca4	Mirai
5	c0571eee3ef8830218dd7bbfd7b915cf5516ba91691e1019b2699191ab3a332c	Mirai
6	88511349498f79eaccfab8c9dd39a8d37560a016d00796c70699023fc76938fc	Mirai
7	5552ee40fdb037c9b64be8e43c19bcee05b92578ce52a6998a90c2f1fca5c5b2	Mirai
8	5657f3003c50b602c15054d9fa7dfb2519a43413885c40ab1a617fc19275f913	Mirai
9	f489758839fb6afb5431ca7dff377b6c86168d251300328d0e6a135105233b3f	Mirai
10	5b9d2c6415873feb6b98ca963bd4b61059056087d5010eb096ce00a2726c983f	Mirai
11	5bb032bda8cc48150744fc08684fcf2c898abf0816f1479cfac02fe729cfa637	Mirai
12	dbebd8e8c11f9e06c1a1ab3019015157f1c82ccdda44f0f0707c69ae721c6890	Mirai
13	72455f499bb407cd090fd079616eb7055824f321d90cbb86bb2f53a757f02c6e	Mirai
14	378df341cea00d8c7838744959fab950d15ae443d14b770cfa2998ae7daf5190	Mirai
15	ae75c29f5f7d3bc602d9cfd355ab6dbcd466c96282fa8ae93a187470ddd34c50	Mirai
16	b8c05074193134695fb975549124835b8f3d1a1ccd24865a2531ad8a90059c7f	Mirai
17	51167f36c3355359a873b19b1aa038fd0772e87b192c8f69b20336d48f980eb6	Mirai
18	bfdd172a08860b7fbfd278e6757f9219d90c25ff47cdf94b57bd3037e81470a1	Mirai
19	652589c71720af72f3566c978fa314408ab12a1286b798f2bec2a4f8525e629d	Mirai
20	e4fafd804c7c9cf29326d4203a74333b211799798cb49d87adb45b9c52938bec	Mirai
21	030b477706540babbfd5997d6afffe47a5cfd3f846521f03873a391a839853c5	Mirai
22	ededadd2a14910547f7dc3d63505b9c03cbf93cecebd302de2e10a75259b13d6	Mirai
23	9b8b0ad1b6f3fa068eec2ddfcb711739b131f4ea5199697a025821729d24ea5b	Mirai
24	4f12ad1c5faa5e43bf17d1906e928e3c7291daa097f9011043582827340604cf	Mirai
25	08fcac8bd754b5b38bad7cb2d17f4347462bc3711a1d82f88da010524ba83f5b	Mirai
26	32b22639b5562d8ef9aa20057053c824ab767cc750a9b17b386f97f829dcdcb3	Mirai
27	94db041c5f1a70c755db90d54c72fb3dfa842729b2d158fb284b3dd90a47491d	Mirai
28	a73ffc17dce716dceb0da272f73d3c6781100aed40565fc601909ef76e908dba	Mirai
29	cda2b6de339a145e6bae502ce3aa71c26de3da7f59547a5764707afdc98fd24e	Mirai
30	acae3ef96626d6b674ca9879419b2fcdc2875bbcc6483f9b4c6057f6374eacde	Mirai
31	e60f7b11d9e26c4a105ca434a2b60bbbd77d69cb13a38b3d2d8aaff0794c9502	Mirai
32	6332c9baecf13d4d9aed26e8d0f14915e0052f34e2cbd84392a3648a0e61fb23	Mirai
33	79d78b3b1aab8e36228f1570659f08c7efc862abc8293291346c837306b3244c	Mirai
34	ec62a759455911c621efb7d6c6aac0b781deabb42931967b712de23ced214589	Mirai
35	3a1063f0af803f8ec5a51076fd5758e1ff784d4eb75645bb81e86cd6fd2504ad	Mirai
36	1925f7a2b715b4af5ff66221447cc5ed135d1b9f9aff2dee8ea1acb62d0dc0a0	Mirai
37	a897bfcd40d42e6d9d8d0b490310a4d21afe4da83bf107f9adc680b52bb09ad9	Mirai
38	f2c7a185f63f76b49c06479b754431b3c897b1e8b47073b0b6e87a49da6db056	Mirai
39	1947ab53faace7d095341791cd2583bcef5419c09b6de6b9052277a3b77e0a14	Mirai
40	bd59588546fe611472c611f46c1a94fd563d59673fa286b7e1d30344bd6cd64b	Mirai
41	0c49abe389cb5f3e59d9f0950468714a68f15c4d1eb1a1c65c9b346ec30471b6	Mirai
42	f064edd2cbb8ab8e0abcfc54406d076390d454b156a6bb71988ebe57b3a3af55	Mirai
43	ce1de869640398a0e51f0f8ad798db97ecfac0b62a3095e823b4ad16f1ef5440	Mirai
44	4cd74e1b5d0441e3b44f4f22c85d41a38dc15ee7de45c6a88b3cadca3c144ef9	Mirai
45	7ba175cd5650ed0d9220003340aae62ee7dec51fea10bc3bf2204dc0899a3873	Mirai
46	d3c865bda24fff7a86d6f70c6909527561097ab7f83db9118dbdd8244dded9b5	Mirai
47	78b6d223f22ed8bf2b628b308eed80a641d415c8a73fdb31994607f3e5e1b570	Mirai
48	b89e37012f39d5abfedf07221cbb1e47e77229210362ad06185f042748118ede	Mirai
49	c06f048b5facaf690ca6bb29f7de30f8cb25803fdeb98e41dc700b1e114b367c	Mirai
50	82080712e408cbeba704ebb29cfd4d1f85cf1f07086008c451331287aa902a16	Mirai
51	5acad83b6314ff5800b5131902a3790d32d9bae5c8a642a23e2936509197072d	Mirai
52	222a737ed1ea068fbc48b3df47627ab9b1f9b06dbe0f0303d38d2546f0afef65	Mirai

Continuation of Table 13

SlNo	SHA256 Hash	Malware Variant Type
53	181a7eca48bef9e356287680dd4a8dd1657662722d26f21305e7939e0a4d96ad	Mirai
54	3c725081c68aed61e1ce646f665865f7b171b379ec3241a0f8a0ed4ab717d728	Mirai
55	88a5b54b9281a7c4b421786af35ff2b7e1107712a027f8f07ad3c28224bacc29	Mirai
56	2e687dc6895cd29e515fe81cecf0fad92530d0d2f18a47b7fb92090b7234e0e0	Mirai
57	5d756eb57c9cff97407a699c96219423061a39ff33b36ac3ff2b4563e4a506f9	Mirai
58	9fc6591bbdd807413dac29d5589ce6b8a1d59c7591fb14affba44a5b91add167	Mirai
59	0d2a1e914747bd6ca919180a491839506c90f2c86b1b1fab543569493389accb	Mirai
60	354ea3fc68c4c745d67417554099d0fa523cd6028ce6d9bac66e67c9739a4325	Mirai
61	179933aa4c9b520b636f1aa49f05b922f7d80b7ec252cb485764508704fc7321	Mirai
62	afae979dc58e9b601a75cfc5af9d2764bbb88d9042e984f2b89c417978ab3a4f	Mirai
63	321cb08441c3a780a1247760c348b5a142e66013be3b3e194a2471d51f7f5891	Mirai
64	ee8ae18792c45b4e1ccede856e30fb141ad000142e90eea7c0a80f4ea9da0322	Mirai
65	4fafd982fd204e1549acdb7653cd4532acaada0fe3475f498387649b5211a852	Mirai
66	5ba6803107fc5d942c158ecfb2eedf7d1b620620574789a8244aca3a58608b66	Mirai
67	87cd6daa315466b7260b1e023da2b6dff926c6418592cfdcb6dc10f2bf323901	Mirai
68	4678be773edfec69238f6352033ef27ce0c78c63828434c06ed69d6128a57d73	Mirai
69	3324e5ccd6b28bbb18cf2d7f0e19b48c1603c29a8b562c12d40137b08f7b8725	Mirai
70	1aec808dc691fee0bf8de862cb088f97f3ab637fb7746668f04fe25798955c8a	Mirai
71	9bb73bc9981ee9bdd3f0f628b0e727b6bf8ac06240e56608517487667a2e9f51	Mirai
72	0434c27a45ee62accfc00ca5fabe07d1d730575cca91df1efef17201a90fde29	Mirai
73	6aacfe5ffbc9808d585bfc623d1fec14ae22b9d8eca8e535583c76ef119fa071	Mirai
74	6f7199d4c55b4006c9f451e48ddcd1f80535660927d0aebe1374ad7598929218	Mirai
75	8f144c1cc3a37120a00abafb28091b3e399b4f65b9b798cbea5a123867eeffb2	Mirai
76	f3c2d7da375ed1afc88c1bc79787675603bed1bb82a67d360300bc7e77b5b6b4	Mirai
77	d4a6e144b49a5e16befa2974384a59bcf68da14ef394948bdf1780bbc589ba67	Mirai
78	7c7c7b54beb1bd503ebdc472b08ed35b0c4291fb465bcad34c26a80b92cb682f	Mirai
79	224f2df0563584885aa637f71077ccce8bb4dab9d7e82dcb12dce92d4e0d704c	Mirai
80	0c45b6faed996600eb05585c532fe7e9d34dc85526afffc08b2fe0fda204f0e9	Mirai
81	c91712f66f9522b6219808b0721baf5f309f627be6025b148f8688a89150cdbf	Mirai
82	9e55a50e619d7f3724e0750449097c387601c255839a7e80676f0c25b4217efc	Mirai
83	a99d6d088071bd216de1fb7dc104bc9fa0b5447debf63958ac4ebf904ac8da45	Mirai
84	1b3bb39a3d1eea8923ceb86528c8c38ecf9398da1bdf8b154e6b4d0d8798be49	Mirai
85	faccd187812a48c7911fb1b643bd346c74f4bc7ddcda2c84e97033e0385ff458	Mirai
86	71adda1a01f2a779796673ec08b1155aa55ffb3f40bdd8752b5a3955684d272e	Mirai
87	82df7a015470179794acc9dc60868ea11221525090f5beecb1c98cdba8510389	Mirai
88	57744761595c2dccdf76560c4e0fe7ea33ea85be281d1b7a9c9b4e9e9dbb0221	Mirai
89	9de1b56f76a47fb1aadc6a78f20e0906bdd9dfcf5379f28fa2927fbdf15bd73b	Mirai
90	2ca71e114a5388aa4d17bb0727bb668dca590b81a063670a44d2dab3adc05af0	Mirai
91	c56fafd9207e18c83d2bfb26550aa00fdba64e05bf5b2aea61629d4108c86517	Mirai
92	3321535dc19687c1d2fd5705012d2653fd6a828733302e4b5932780e7637c084	Mirai
93	8e3fb9f382c1a3136da6e83361464e694d77502b483907eb3f9c55890372e66c	Mirai
94	c052c89438af51f7b8af26b3c5864650d0f2c2199653edc76671d62258f234cf	Mirai
95	1f56bc65381ff6e095c5aa0c84de5d368c08f3a8ee12a0e84c67fcd80626b4fa	Mirai
96	b4623f517f49a825f2f53e4497f944fe10fe9368b3c0db1d30b3ebc63c120962	Mirai
97	b279115318cd447823c8410ee3a318a8c531733404394ec8336184102854c554	Mirai
98	46823451734e47b845bff6d3440ffbd096f8742d9bd8962b1a7c18bf0144d9b4	Mirai
99	8976173ee948c64e89657f734eaea431c5e7a49d5ab7528c676a8d50f1306157	Mirai
100	b79f967aece83b7eda2db4feb08c2eec352eff9d5802f6ea9214064b128987d9	Mirai
101	a05a6affb61f1c84dc30cc0578ddf5aa32b833fc5a772f3abac613293ff89b06	Mirai
102	865c8d9c31a120a92524cf24a8961bc16fe521bea6e72702afc8ac1a0ea9b4ad	Mirai
103	3ef73e98076dc49d83f733eceab93dcffeeefdbdf6f0a36bef756d3448b5d9dc	Mirai
104	5ce6a9bb4daca8f2d6624c654a445d68b2c2f28440c648adb16d0546b9299ecd	Mirai

Continuation of Table 13		
SlNo	**SHA256 Hash**	**Malware Variant Type**
105	46d8fb0b1e46ff8ee0d65697080af8f7ee11d0a741ae0ca662aedad63a716ebd	Mirai
106	6c5679eb7bd905b3ee86ea5770dbfd8fb50be013c6e93ad1df8fd75a6689d523	Mirai
107	057b9b5d11a4500bc0f46b9c2317ef8f82beaa6d95d5babe0194d4a7379d4f6f	Mirai
108	439dc5e3183a9f4316472691791ca6c33c9e56bce480d88ed5a82c28481f6bc2	Mirai
109	de82fb3927bb9357cdba7f8c595bb87940e7502acb8b605ea7eb0e876ac2808b	Mirai
110	7965a27369b329db4004b871429432beb5c0301c03a48bf64d0961e951e712cf	Mirai
111	b936597d0d868607e45478b9be01a9365078d33bdda2a8c053500c729a8cbaf6	Mirai
112	0c1472a800bdaaad840110f93f3c4b248509f7505fc2a1330af2cdc7c2eccfc4	Mirai
113	bfee2d1d34214a93024447bc054c1eea2b05111a74508bd74997eea6a7c4ef65	Mirai
114	f060058462bfaef0bd9382de38d238b96ff4f886967b70020406dab38190bff6	LuaBot
115	0206efba7fc13700efd59354e9c6ca4d1ffe34f6689bd195798181824d46b83d	LuaBot
116	e1f6d967db61ee131dc32b817a9285f5da3ebe3e1f9a4281c8fac9339e2b4521	LuaBot
117	9866b4f2f533de7d742251915802dab355a59f10a51a8bf7d146fd4cb015cd5a	Brickerbot
118	4f9b895a2785f9788fcae8743ab04a24b62e0962b1f8a28dc1206c52327b7916	HEHbot
119	8a2a28d164a6d4011e83ae3f930de8bf1e01ba2e013bee43460f2f58bdaf4109	Photominer
120	01d8e2bcf22422e9c995d43c403c63477389fc9f4a141ef3bbd31c8f5c6ef7e6	Mozi
121	01d8e2bcf22422e9c995d43c403c63477389fc9f4a141ef3bbd31c8f5c6ef7e6	Mozi
122	c9038e31f798119d9e93e7eafbdd3e0f215e24ee2200fcd2a3ba460d549894ab	Lokibot
122	b47e281bfbeeb0758f8c625bed5c5a0d27ee8e0065ceeadd76b0010d226206f0	WannaCry
123	0a73291ab5607aef7db23863cf8e72f55bcb3c273bb47f00edf011515aeb5894	WannaCry
124	d7d0f18071899c81ee90a7f8b266bd2cf22e988da7d0e991213f5fb4c8864e77	LemonDuck
125	d1e82d4a37959a9e6b661e31b8c8c6d2813c93ac92508a2771b2491b04ea2485	FritzFrog
End of Table		

Table 13: Malware hashes detected by our honeypots as found in Virustotal