2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) | 978-1-6654-3572-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/DSN48987.2021.00059

Sanitizing the IoT Cyber Security Posture: An Operational CTI Feed Backed up by Internet Measurements

Morteza Safaei Pour, Dylan Watson, Elias Bou-Harb
The Cyber Center for Security and Analytics, University of Texas at San Antonio, San Antonio, Texas
{morteza.safaeipour, dylan.watson, elias.bouharb}@utsa.edu

***Abstract*—The Internet-of-Things (IoT) paradigm at large continues to be compromised, hindering the privacy, dependability, security, and safety of our nations. While the operational security communities (i.e., CERTS, SOCs, CSIRT, etc.) continue to develop capabilities for monitoring cyberspace, tools which are IoT-centric remain at its infancy. To this end, we address this gap by innovating an actionable Cyber Threat Intelligence (CTI) feed related to Internet-scale infected IoT devices. The feed analyzes, in near real-time, 3.6TB of daily streaming passive measurements (≈1M pps) by applying a custom-developed learning methodology to distinguish between compromised IoT devices and non-IoT nodes, in addition to labeling the type and vendor. The feed is augmented with third party information to provide contextual information. We report on the operation, analysis, and shortcomings of the feed executed during an initial deployment period. We make the CTI feed available for ingestion through a public, authenticated API and a front-end platform.**

***Index Terms*—Internet-of-Things (IoT), Cyber Threat Intelligence, Security capabilities, Network telescopes, Data science**

I. Introduction

The Internet-of-Things (IoT) paradigm is no longer just a concept. It has indeed touched and infiltrated various facets of contemporary life. The number and types of deployed IoT devices in households, organizations, critical infrastructure, and cities have increased with rampant speed [1]. Meanwhile, the security posture of these devices did not keep abreast with their applicability and have undoubtedly not received enough security attention dealing with their design, manufacturing, and provisioning by vendors, policymakers, and consumers, all resulting in disastrous consequences [2, 3]. As a result, the IoT paradigm continues to attract malicious actors' attention, through targeted exploitations by evolving malware [4]. While the operational security communities continue to lay down efforts for IoT security monitoring and response, several challenges hinder such endeavors.

First and foremost, patch management is more difficult in the IoT context than conventional systems, given the heterogeneous nature of the IoT devices and their far-from-optimal update mechanisms [2]. In many cases, organizations deploying such IoT devices might not be aware of the exact details of the IoT models, which renders their management quite difficult especially in realms with significant amount of deployed devices (i.e., health and transportation sectors) [5]. Further, a large number of IoT devices are not yet compatible with central management systems for monitoring and management. Therefore, the chances that an IoT device becomes infected and goes unnoticed is quite high.

Second, for continuous and proactive monitoring of the cyber security posture, threat intelligence and security service providers typically require access to the organizations' network traffic and systems' data. This is achieved either by installing special agents/software or setting up hardware to curate and analyze the required data, while processing it on-site or using a cloud-based capability. Nevertheless, organizations are most often reluctant to even share their basic internal information for privacy and liability concerns. This situation has induced the broad lack of available real-world data (including IoT-centric empirical data) to utilize for identifying compromised devices. Additionally, with the IoT paradigm being deployed at an Internet-scale perspective, having microscopic data access is impractical and would definitely not scale well.

Third, the objective of IoT device fingerprinting (i.e., identification) by analyzing network traffic continues to be an open research problem. Along this line of thought, the capability to infer compromised devices is even more challenging, given the increasing sophistication of IoT-malware which execute various actions (i.e., killing running services or altering device characteristics) to avoid being flagged by detection and monitoring systems as well as by competing IoT botnets [6].

Given the aforementioned challenges, coupled with the lack of IoT-centric Cyber Threat Intelligence (CTI) capabilities which would aid the operational security community in identifying and responding to Internet-wide compromised IoT devices, in this work, we make the following contributions:

1) We introduce eX-IoT (for exploited IoT), a first-of-a-kind operational, real-time CTI feed, operating on streaming Internet-scale network telescope data, for fingerprinting (and notifying about) compromised IoT devices deployed in Internet-wide realms. eX-IoT is advantageous as it provides a macroscopic visibility of deployed IoT devices, independent of the host organization. Indeed, while other scan-based threat detection capabilities exist, eX-IoT complements them by innovating scientific and engineering methods to specifically address the IoT insecurity problem.
2) We design eX-IoT, both in terms of its dynamically-updating machine learning methodology to keep track

978-1-6654-3572-7/21/$31.00 ©2021 IEEE
DOI 10.1109/DSN48987.2021.00059

of newly discovered IoT devices in the wild, and its computing architecture, from the data source to the indexing (and augmentation) of the generated threat intelligence. We report on the operation of an initial deployment of eX-IoT, while evaluating it against other CTI feeds in terms of various metrics including latency, accuracy and coverage.

3) We make eX-IoT available to the security community for ingestion through an authenticated RESTful API and a streamlined front-end platform. We also develop eX-IoT to monitor certain IP spaces of interest and to automatically respond to IoT exploitations through email notifications. We validate eX-IoT's initial results by collaborating with US and international operators.

The remainder of this paper is organized as follows. In the next section, we review the relevant literature to demonstrate the state-of-the-art contributions of this work. In Section III, we detail the science and the engineering methodologies employed within eX-IoT. In Section IV, we detail how we make eX-IoT available for ingestion. We evaluate eX-IoT in contrast to other CTI feeds under various data-driven metrics in Section V. The shortcomings and conclusion are respectively discussed in Section VI and VII.

II. Related Works

In this section, we review two topics. The first focuses on efforts pertaining to compromised IoT device fingerprinting. The second summarizes available operational cyber security capabilities for monitoring cyberspace to highlight the need for tools and platforms which are IoT-centric.

A. Fingerprinting Compromised IoT devices

Honeypots. The first line of research leverages IoT-specific honeypots to gain CTI into IoT maliciousness [6–9]. However, often, the vantage size is extremely narrow and limited to a small set of mimicked devices, firmware versions, and services which hinders the completeness and the quality of the results to be employed for operational security. In addition, sophisticated IoT malware and attackers execute reconnaissance and discovery tactics to avoid honeypots [7].

Internet telescope and edge. Another line of research relies on Internet telescope (darknet) [10, 11] or similar passively collected edge network traffic to identify infected IP addresses. However, additional steps are required to identify IoT-specific characteristics. Some studies relied on the unique identifier in the received packets (TCP seq == dst IP) to attribute them to Mirai infected devices [6, 12, 13]. Similarly, Cetin et al. [13] identified Mirai-related infections following an attempt to eradicate them from the hosting network where notification-based remediation efforts were successfully tested. However, not every malware will carry such a profound signature as Mirai [14]. Indeed, works that leverage malware-specific signatures are not generic enough, rendering them impotent to fingerprint devices infected by emerging malware.

An alternative technique relies on correlating a list of identified malicious IP addresses with fingerprinted devices through the application of banner grabbing using `Shodan` [15–17] or actively scanning set of IP addresses to retrieve available service banners. Banners contain text information that needs to be processed to characterize the type and model of the device. Acknowledging the variety of IoT devices in the wild, various learning techniques have emerged to accomplish the classification objective [18–25].

Nevertheless, several challenges related to banner-based IoT fingerprinting techniques exist, including *(i)* the fact that modern malware close ports and services with the intention to avoid reinfection by competing malware and to conceal their identity from Internet scanners [12, 26]. *(ii)* That IoT devices may reside behind a border firewall/NAT where accessing such devices becomes restricted and *(iii)* the fact that some vendors avoid hard-coding device information in clear text which makes banner analysis almost impossible. In the context of eX-IoT, we combined passive Internet traffic with active application banners while feeding the output to a machine learning model to address these IoT fingerprinting challenges, especially for devices where their banners are not available.

ISP level and Internet transit. Some studies leveraged Internet backbone and transit traffic to identify botnet activities and related compromised hosts and devices [27–30]. These proposed detection methods often require intensive computation due to the high volume of curated traffic (benign and malicious) and they generally might possess lower accuracy in contrast to leveraging passive measurement techniques.

Internal network traffic. This line of research is based on analyzing internal network traffic. Meidan et al. [31] proposed a network-based anomaly detection technique that employs deep autoencoders to discover abnormal network traffic generated from compromised IoT devices, while Hafeez et al. [32] introduced IoT-Keeper, a lightweight anomaly detection system at edge gateways. Though noteworthy, these techniques require special software and hardware to curate such data, in addition to possessing a small-scale, microscopic perspective of IoT maliciousness. In contrast, eX-IoT CTI's feed leverages Internet telescope traffic which is a "pure" source of unsolicited/malicious activities while providing a global view towards compromised IoT devices.

B. Operational Cyber Security Capabilities

Although the operational security communities continue to develop capabilities for monitoring cyberspace [33], tools which are IoT-centric remain at its infancy. `Shodan` [34] was among the first search engine for constantly monitoring cyberspace to index all Internet-facing end-hosts. The engine allows users to find specific types of devices and services (webcams, routers, servers, etc.) connected to the Internet. Following closely the same objective, `Censys` [35], developed by the University of Michigan, based upon the open source tools Zmap [36] and Zgrab [37], forked many Internet- and security-based studies [12, 38–40]. `Shodan`, `Censys` and `ZoomEye` [41] are used by organizations to monitor their IP space for publicly accessible services and vulnerabilities related to them. However, actual infections are not visible

to their vantage points and thus they do not cover or report about infected (IoT) devices. `Dshield` [42], in contrast, provides amalgamated statistics about the daily activity of each targeted ports based on IDS crowd-sourcing and reports provided by entities and individuals. The closest work to eX-IoT is `GreyNoise` [43], which provides general lists of Internet-wide scanners. However, its implementation and *modus operandi* are obscure and the engine does not provide IoT-specific CTI.

Thus, it is intuitive to note that eX-IoT will be among the first to focus on the Internet-scale IoT cyber threat landscape by devising integrated scientific and data engineering methodologies while pinpointing and sharing relevant CTI, in near real-time, on hundreds of thousands of newly infected IoT devices, this developed capability is postulated to contribute toward a better IoT hygiene while aiding security operators and hosting organizations with their (IoT) security triad endeavors.

III. Methodology and Architecture

The rationale of eX-IoT is to possess an Internet-scale visibility of deployed IoT devices. To this end, it leverages passive network telescope traffic. A network telescope is set of routable, allocated yet unused IP addresses [10, 44]. Such IPs passively collect incoming packets without sending any replies; they have also not been assigned to any machine (with a legitimate service) and therefore there is no reason for Internet nodes to send packets towards these IPs. Therefore, all the incoming packets are either *(i)* Internet-wide scans, *(ii)* backscatters from DDoS attacks, and *(iii)* results of machine/network malfunctioning. Therefore, Internet telescope data provides a clean, unsolicited dataset (no traffic with legitimate intention) of Internet-scale malicious activities. Compromised IoT devices are often constantly scanning the Internet to discover more vulnerable devices and during this procedure, they inevitably send a packet to Internet telescopes. Following the filtering of backscatter packets based on their flags and other header fields, an employed Threshold Random Walk (TRW) scan detector algorithm [45] identifies scan activities. Since there is no purpose that an IoT device (e.g. IP camera) would perform Internet scanning as part of its normal operation, this would be a strong indicator of its compromise.

The methodology at the core of eX-IoT is illustrated in Figure 1. Following the detection of the scanners ①, the tool immediately probes the scanners for open ports and application banners ②. The returned application banners ③ are checked with a database of text fingerprints of IoT devices ④ to generate labels (IoT vs non-IoT). The label for IP address x.x.x.x along with the traffic samples originated from x.x.x.x will be used to train (update) the machine learning classifier ⑤. Finally, the classifier is applied on newly incoming scan traffic to predict their label ⑥. Particularly, we train a random forest classifier to predict the label (IoT vs non-IoT) of sources that are generating scans towards the network telescope. Each sample in the training and test datasets consists of fields extracted from the received packets with a corresponding label. The output of the machine learning classifier is one of the

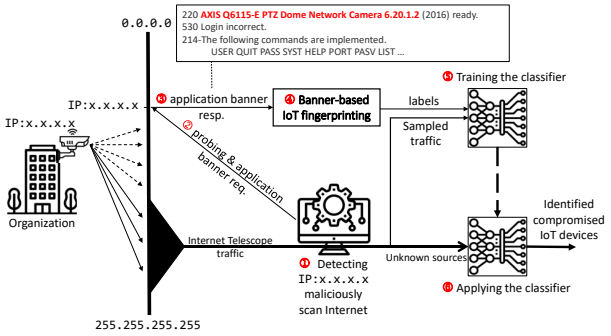

Fig. 1. Methodology of the detection system

binary classes (IoT or non-IoT). Note that the model can adaptively learn the behavior and implementation differences of IoT-specific malware families and evolving IoT botnets. The effect of these differences is indeed reflected in their generated scanning packets and can be observed in various features such as scan packet inter-arrival times [46], and the set of targeted ports (and corresponding assigned weights to each) [47]. Readers that are interested in the inner modus operandi of the employed machine learning methods are kindly referred to the "Annotate" and "Update Classifer" subsections herein as well as to our previous in-depth studies in [47–49].

The overall data engineering architecture of eX-IoT is demonstrated in Figure 2. The numbers highlight the implementation/usage of the same steps as in Figure 1. We leverage a /8 Internet telescope (16M+ IP addresses) from CAIDA [50]. The passive traffic collected by CAIDA consists of approximately 150GB/hour which is on average more than 1M packets per second (with ascending trend). The data is collected hourly, compressed and stored using OpenStack Swift [51]. Access to data is provided through a cluster located at UCSD. Due to the agreement policy and the high volume of data, transferring the entire data stream and processing it locally in real-time is infeasible. Therefore, the flow detection and packet sampling stages are executed on the assigned cluster at CAIDA. The analyzed sampled batch of packets are then sent to the local eX-IoT server through an established secure tunnel for further processing. The processing pipeline is divided into several distinct modules and run separately to achieve high level of parallel processing, high throughput and low latency for real-time stream processing. Besides, eX-IoT depends on three distinct databases; (i) A MongoDB database to store the latest threat information related to compromised IoT devices, (ii) Another historical MongoDB database to store information of compromised devices with a lapsing two-week period, (iii) A Redis in-memory database to store records (OBjectIDs) of MongoDB instances related to compromised devices that are still active; and to use for fast and low-overhead update of the devices coupled with their malicious activities.

Flow detection and packet sampling. This module is developed in C++ and utilizes the `Libtrace` [52] packet handling library to achieve real-time processing. The program constantly checks for newly added data sources (hourly), decompresses and analyzes every single packet. Packets that

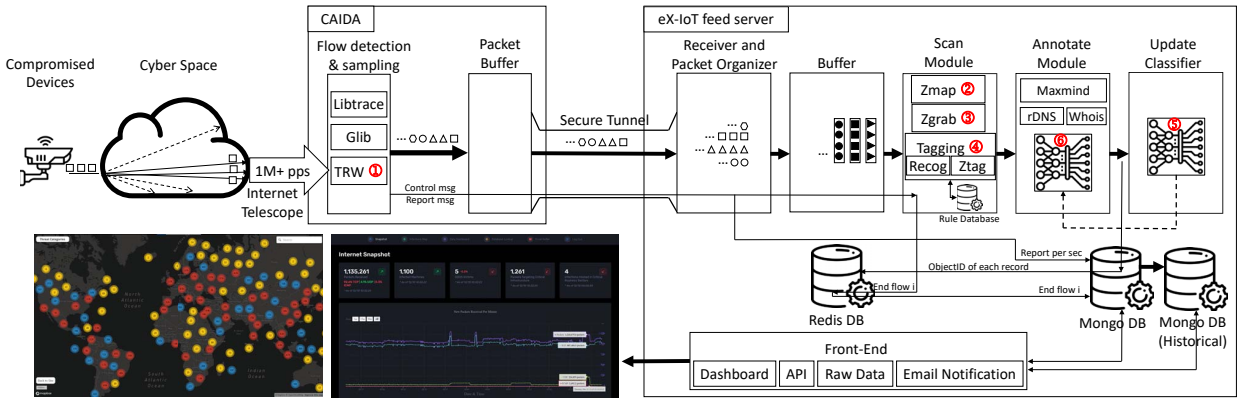

Fig. 2. Architecture of eX-IoT

are not backscatter-related (e.g., packets with only TCP ACK flag set, ICMP packet with unreachable code set, etc.) are considered as potentially scan packets. For packets which pass this step, their state is kept in a GLib Hashtable [53] with the source IP address as the key. The state for source IP x consists of a {start timestamp, timestamp of latest arrived packet from x, number of packets from x, `IsScanner`}.

We then use a TRW-based scan detector [54, 55]. The program considers a source x as a scanner if it receives at least 100 packets without expiration (inter-arrival time between consecutive packets from x should not be more than 300 seconds). Also, its duration should not be less than 1 minute to find and exclude flows that are result of network misconfigurations. Upon detection of a source IP address x as a scanner, the `IsScanner` attribute for x is set, and the number of received packets is reset to zero to start packet sampling. Subsequently, a full list of field values (fields in IP, TCP, UDP, ICMP headers, timestamp, packet and header length) from x is generated for the next *threshold* (200) packets and sent to the next module. The module seizes after the *threshold* sample is acquired.

For the next batch of incoming packets from x, the program ignores the packets and updates the latest timestamp for x. At the end of an hour and before starting to process the new hour, if the latest timestamp for x is more than 1 hour ago, it expires the scan flow, concludes that the scan has ended, and sends a message that the flow generated by x has ended. Another task of this module is to provide packet-level reports every second including total processed packets, number of TCP, ICMP, UDP, number of newly detected scan flows, and number of packets target specific ports. On average, this module spends close to 20 minutes to analyze one hour of data.

The output is sent to a specific local port using Socat [56]. Therefore, if any network communication is disrupted, the flow detection and sampling module will go idle until the next stage can reconnect to that port and be ready to receive the data. Thus, no data will be lost due to network failures.

Receiver. This module establishes and maintains the secure communication as well as analyze the control and report messages. If the secure communication is disrupted for any reason, it will retry to establish a new SSH tunnel and connect to the specific local port on the CAIDA cluster. The receiver then sends the report messages to the MongoDB.

TABLE I
LIST OF SUPPORTED PORTS AND PROTOCOLS

Ports	80, 22, 443, 21, 23, 8291, 554, 8080, 7547, 8888, 5555, 81, 631, 8081, 8443, 9000, 8888, 2323, 85, 88, 8082, 445, 8088, 4567, 82, 7000, 83, 84, 8181, 5357, 1900, 8083, 8089, 8090, 110, 143, 993, 995, 20000, 502, 102, 47808, 1911, 5060, 5000, 60001
Protocols	HTTP(s), TELNET, SMTP(s), IMAP(s), POP3(s), SSH, FTP, CWMP, SMB, MODBUS, BACNET, FOX, SIP, RSTP, SSL/TSL, DNP3

Packet Organizer. This module receives all the sampled packets from different sources, organizes them based on their source IP address and arrival time, packs them and dispatch them to the next module. This way, the module ignores the sources that do not contain enough samples to be used for applying the model or for updating the classifier. These are typically sources that have been erroneously identified as scanners and may be the results of node malfunction on the Internet (which usually send out burst of small number of packets for a very short period). The output of this module is in a `JSON` format piped to buffer.

Buffer. The buffer is an in-memory large FIFO (15GB) to curb the effect of mismatched processing delays among the modules especially due to the rising volumes of data which is sent out from the CAIDA cluster. We leveraged the `mbuffer` implementation [57] for this purpose.

Scan Module. This module performs buffering of batch of identified scanners for 100k records or 60 minutes, run Zmap for target ports (50 ports) with 5K pps rate, followed by running Zgrab for several protocols (16 protocols). Table I enumerates the supported ports and protocols in the initial deployment of the system. These ports/protocols are known empirically to be the most responding; could be easily extended using updated measurements from emerging threats.

The module also prepares an updated database of application banner fingerprints based on `Recog` [58] and `Ztag` [59] and applies it on returned banners using `Beautifulsoup` and regex to add information regarding their vendor, type, model and firmware version which are also used as labels in

the update classifier module. We select `Recog` since it is an open source repository, where individuals contribute actively to add new rules. The scan module also dumps unknown banners that contain the "[a-z]+[-]?[a-z!]*[0-9]+[-]?[-]?[a-z0-9]" regex rule as a generic rule for inferring device-related information in text [21] to a log file for further inspection and for generating rules for new devices that are not covered by the mentioned public resources. Meanwhile, the returned banners are added to the records in `JSON` format. Besides, upon receiving an `END_FLOW` message for a source IP x, the module retrieves the ObjectID of that specific infected IoT device from the Redis database (which contains a list of active infected devices) and use that for updating the status of the device in the MongoDB. Searching the MongoDB based on returned ObjectID is less expensive than finding the latest record for that IP address among all records in the database.

Annotate Module. The annotate module first pre-processes traffic for each identified scanner and then applies the latest updated classifier to the network flow to identify if the flow has been generated by an infected IoT device or not. The pre-processing step consists of *(i)* calculating inter-arrival times, processing the TCP options field and then normalizing every field using MinMax and subtracting it from the mean value of the training dataset; a final list of fields are summarized in Table II, *(ii)* Min, first-Quantile, Median, third-Quantile and Max values for each field over all sequence of packets from each source are calculated which is a tuple of size $24 \times 5 = 120$ that are considered as the final feature set to be fed into the machine learning model for training and classification. The output of the classifier is the predicted label and a value between 0 and 1 which is the prediction score.

TABLE II
LIST OF EXTRACTED FIELDS FROM INCOMING PACKETS

General	Protocol ({TCP, UDP, ICMP}), Dst port, Total length, TCP offset, TCP data length, Inter-arrival time
IP header	Type of Service, Identification, TTL, Src IP, Dst IP
TCP header	Src Port, Sequence, ACK Sequence, Reserved, Flags, Window Size, Urgent Pointer
TCP Options	WSCALE, MSS, TIMESTAMP (Binary), NOP (Binary), SACK-permitted (Binary), SACK (Binary)

In the next step, the module looks up every identified IP address in various databases including `MaxMind` dataset, `IP WHOIS`, and reverse DNS. The results comprise of geo-location data (country, state, city, latitude and longitude co-ordinates), hosted ISP, Autonomous system (ASN), domain address, business sector, resided organization and registered emails related to the hosting entity. In addition, the results of packet-level fingerprinting of IoT malware's scanning module (e.g., Mirai) [12, 60] and Internet scanning tools (ZMap, MASSCAN, Unicorn, Nmap) [61] is appended. Additionally, a list of targeted ports and their distribution, scanning rates and address repetition ratio (ratio of the number of all packets to the number of unique targets) for each flow is estimated [47]. Besides, scanners are labeled as `Benign` if their rDNS records contain domains that can be attributed to legitimate security companies and research institutions such as scanners from University of Michigan, Shodan, Censys, Rapid7, etc.

Update Classifier. Traffic that has the original labels (from the tagging step in the scan module) are passed to this module. The flow pre-processing step is exactly the same as the explained step in the annotation module. The model, which is updated every 24 hours, use data samples during the past 14 days to make sure that the model is always updated based on the latest information and can comprehend the patterns related to emerging IoT malware. Available data is split into training (20%) and testing (80%) datasets and the best Random Forest classifier model (from the `sklearn` package which maximizis roc_auc metric) is selected among 1000 iterations over a set of tuned hyperparameters. All the daily trained models are augmented with training timestamp and stored in a directory to make the results easily reproducible. In preliminary tests, the performance of Random Forrest (RF), Support Vector Machine (SVM) and Gaussian Naive Bayes (GNB) over a wide-range of hyper parameters are compared. Results based on ROC-AUC and F1 score motivated us to leverage the Random Forrest model for implementing eX-IoT.

IV. EXPOSING DATA (USER INTERFACES)

To provide a fast and reliable way of accessing both the raw data and the generated CTI, eX-IoT exposes its data to the operational security communities and researchers in several ways: a web-based interface for data analysis and data visualizations, a RESTful API for programmatic access, raw bulk data and email notifications. Please note that eX-IoT's CTI feed will be available for free to other academic researchers to use through data sharing agreements via DHS IMPACT [62], which addresses legal and logistical concerns. Meanwhile, interested parties can also contact the authors to access the feed.

Web Interface. eX-IoT's web Interface is a hub for data visualizations and raw data searches. It is comprised of 5 parts: (1) an Internet snapshot that provides high-level real-time data, (2) an interactive map of all data points in the past week, (3) a dashboard with data visualizations and the ability to examine specific database fields and (4) a raw database query builder.

Programmatic Access. eX-IoT's REST API is a way for data to be easily filtered and extracted from the database in an ingestible format. The API returns data encoded as `JSON` objects for ease of interpretation or integration by third-party applications. Details about the API could be found on [63].

Raw Data. In some cases, bulk historical data is required and thus eX-IoT can provide this to security operators/researchers, practitioners and government authorities for research/training purposes and cyber situational awareness.

Email Notification. Two mechanisms for email notification are considered. First, organizations and users can set alarms for their IP block and instantly receive notification through their provided email address. In the second mechanism, eX-IoT feed will notify organizations and ISPs who host infected IoT devices by the list of organization's email address available in their WHOIS record.

V. Initial Operation and Evaluation

To report on the operation, analysis, and shortcomings of eX-IoT, we first executed it for two weeks to make sure the model had enough data points for training purposes. We consistently completing all steps on an Intel Xeon W-2145 (16 cores at 3.70GHz) processors, 128GB of DDR4 memory, and RAID 1+0 with an Intel 850 Pro 1TB SSD drives. The cluster at CAIDA is running an Intel processor (Skylake, IBRS) with 8 cores at 2.20GHz and 32GB of RAM.

A. Initial CTI Validation

While we only operated eX-IoT for a short period of time, we made an effort to validate the generated CTI in terms of the exploited IoT devices. Ideally, we would use the email notification capabilities to contact each entity in which eX-IoT have identified a compromised hosted IoT device. Nevertheless, for initial validation purposes and to obtain quite a comprehensive and a convincing response rate, we utilized two approaches. First, we worked with a US-based entity, namely, Bad Packets [64]. Bad Packets deploy and operate large-scale honeypots (including IoT-specific honeypots) distributed across many network providers and spread across multiple countries. We used CTI from Bad Packets to correlate eX-IoT's CTI feed related to US-based IoT exploitations. Second, we worked with a CSIRT in Czech Republic and cross validated eX-IoT's IoT exploitations that are specific to Czech Republic with the CSIRT's scanners' database [65]. For both approaches, we used data from the week extending from March 14$^{\text{th}}$ to March 18$^{\text{th}}$, 2021. Broadly, we were able to validate close to 70% of eX-IoT's detected IoT exploitations from both sources, with the CSIRT in Czech validating close to 83% of the country-based IoT exploitations. Several factors could have affected the validation accuracy, including, the limited/different vantage points used by those sources, the time frame of the conducted validation, the fact that IoT malware continue to avoid honeypots, and eX-IoT's false positives in terms of misclassifying a scanning source to be an IoT device (rather than a generic scanning host, though the learning approach have previously demonstrated high accuracy [47]). Nevertheless, we believe the initial validation results are motivating and we continue to work with local, federal and international collaborators to fine-tune it.

B. Evaluation

Although there exists no similar feed which solely and exclusively focuses on compromised IoT devices, we further evaluate eX-IoT based on a well-defined set of metrics [33] in contrast to the other two scan-based feeds, namely `Dshield` (one of the high volume public feeds) and `GreyNoise` (commercial threat intelligence that tags the records with "Mirai" and "Mirai variant"). In the sequel, we elaborate on such metrics and on the corresponding evaluation.

Volume is the rate of a feed, which quantifies the amount of data appearing in a feed on a daily basis. Table III reports the average number of new daily records in the evaluated feeds over a week period. `GreyNoise`, on average, reports around $215,350$ records which classifies $85,330$ as being malicious, $126,018$ unknown and $4,002$ as benign. `GreyNoise` also tags $20,557$ records with "Mirai" and "Mirai variants". `DShield` have $214,390$ records in general without any information about IoT exploits. eX-IoT in general identifies close to 4 times more threats than the other two feeds. Further, regarding the number of infected IoT devices, eX-IoT detects about 7 times more comparing to `GreyNoise`.

TABLE III
Volumetric comparison of scan-based CTI feeds

	eX-IoT	GreyNoise	DShield
All	757,289	215,350	214,390
IoT-specific	145,989	20,557	N/A

Another metric is the **differential contribution** of one feed with respect to another. That is the number of indicators which appear in the first feed that are not in the second feed over the same measurement time; $\text{Diff}_{A,B} = |A \backslash B|/|A|$. $\text{Diff}_{A,B} = 1$ indicates that the two feeds have no elements in common, while $\text{Diff}_{A,B} = 0$ indicates that every indicator in A also appears in B. It is a measure to characterize how many additional indicators a feed offers relative to one or more feeds. Respectively, **normalized intersection** is defined as $1-\text{Diff}_{A,B}$. Similarly to differential contribution, **exclusive contribution** is defined as the contribution of a feed with respect to a set of other feeds which is the proportion of indicators unique to a feed; $\text{Uniq}_{A,B} = |A \backslash \bigcup_{B \neq A}|/|A|$. To this end, the set of newly infected devices from eX-IoT during the 9$^{\text{th}}$ of Dec 2020 is considered which contains $134,782$ unique IP addresses. The IP addresses are contrasted against `GreyNoise` and `DShield`, where GreyNoise was found to contain information about $28,338$ of them ($28,338$ in it's historical database, $12,282$ have updated in the same time period, $10,460$ tagged with "Mirai" and "Mirai variants"). Further, we matched them with the `DShield` feeds from the same time period which lead to $8,559$ common records. Subsequently, we report on differential contribution, normalized intersection and exclusive contribution of eX-IoT with respect to these statistics. The results in Table IV show the significant contribution of eX-IoT against the other CTI feeds (more particularly, in the compromised IoT-context) where typically devices execute scanning in low rates. First, `DShield` does not provide information about the IoT/non-IoT type of these scanners, and `GreyNoise` tagged $10,640$ of $134,782$ as "Mirai" and "Mirai variants" which confirms the lack of IoT-specific focus in the existing threat feeds. Second, the differential contribution which is close to 1 confirms the high contribution level of eX-IoT feed over the other feeds. The maximum value for the normalized intersection only reached

TABLE IV
Metrics of eX-IoT in contrast to GreyNoise and Dshield. The feeds are compared with 134, 782 IoT records from eX-IoT

	GreyNoise	GreyNoise(Mirai)	DShield
# of indicators	28,338	10,640	8,559
Diff$_{A,B}$	0.78974	0.92105	0.93649
Normalized Intersection	0.21025	0.07894	0.06350
$\lvert A \bigcap (\cup_{B \neq A}) \rvert$	31,563		
Uniq$_{A,B}$	0.76582		

TABLE V
REPORT ON TOP-5 CHARACTERISTICS OF GLOBAL IOT INFECTIONS ON THE 7TH, 8TH AND 9TH OF DECEMBER 2020 TIME PERIOD.

Country	Continent	ASN	ISP	Critical Sector	Vendor	Target Ports
China (43.46%)	Asia (73.31%)	4134 (21.28%)	China Telecom [CN] (21.16%)	Education (649)	MikroTik (11583)	23 (43.25%)
India (10.32%)	S. America (10.82%)	4837 (16.45%)	Unicom Liaoning [CN] (16.23%)	Manufacturing (240)	Aposonic (1809)	8080 (37.40%)
Brazil (8.48%)	Europe (8.62%)	9829 (5.38%)	Vivo [BR] (5.38%)	Government (184)	Foscam (1206)	80 (37.16%)
Iran (5.51%)	N. America (5.57%)	27699 (4.96%)	BSNL [IN] (5.31%)	Banking (80)	ZTE (709)	81 (13.10%)
Mexico (3.52%)	Africa (4.10%)	58244 (3.30%)	Axtel [MX] (3.03%)	Medical (79)	Hikvision (638)	5555 (12.92%)

0.21. In other words, 78.9% of IoT infections flies under the GreyNoise radar without being detected. Finally, about 76% of eX-IoT output are unique and the records are not indexed in other feeds.

Latency related to a feed is the elapsed time between an instance's first appearance in any feed and its appearance in the feed in question. Latency characterizes how rapidly new threats are included in a feed. To this end, by leveraging Zmap, we execute a 3-hour Internet-wide scanning for port 80 with a rate of 1000 pps on Dec 9th 2020 at 7:30:00. It then indeed appeared in eX-IoT as "Desktop (non-IoT)" and tool as "Zmap" at 12:42:04 of the same day, which means that it took 5 hours and 12 minutes from the time that scan started to appear in the feed. The main contributor of this delay is CAIDA's role in collecting, compressing and storing to prepare hourly pcap files which approximately take 3.5 hours. The detected start time and end time for this test scan in eX-IoT is recorded as 7:30:24 and 17:48:59 which respectively have erroneously 24 seconds and 13 minutes difference. Comparing with other feeds, the IP did not appear in DShield, while the record was added to the GreyNoise feed with close to 10 hours of latency since the beginning of the scan and the tool was incorrectly identified as "Nmap".

The **accuracy** in this context is equivalent to precision of a feed which is the percentage of indicators in the eX-IoT that are correctly labeled as IoT. The **coverage** is equivalent to recall in information retrieval contexts [66] and defined as the proportion of the correctly labeled IoT devices contained in the feed. Accordingly, here we consider accuracy and coverage with respect to the assigned labels as IoT not the accuracy and coverage for all scan feeds. Therefore, we compare the labels in the eX-IoT with the labels derived directly from the returned banners. We checked this for all the records on the 7th, 8th and 9th of December 2020 that had the true label (IoT/non-IoT) based on their banners and use them as the grand truth to evaluate eX-IoT. The analysis led to accuracy (precision) of 94.63% and a coverage of (recall) 77.21%.

Next, we report on a snapshot of compromised IoT devices. All the discovered compromised IoT devices that are active during the 7th-9th of December 2020 are selected. Although it is not possible to remove the effect of dynamic IP allocation and IP churn on the reported statistics, we select three days as a trade-off between the completeness of the view and the IP churn effect. eX-IoT generated CTI related to 488,570 instances belonging to 405,875 unique IP addresses. Therefore, only 82,695 ($\approx 16\%$) have redundant IP addresses. Based on Table V, China (43.46%), India (10.32%), Brazil (8.48%), Iran (5.51%) and Mexico (3.52%) are the top 5 countries which host infected IoT devices. Further, the top ASN included 4134 (21.28%), 4837 (16.45%), 9829 (5.38%), 27699 (4.96%) and 58244 (3.30%). By identifying the type of the hosting organizations [67], the existence of compromised IoT devices in Education (649), Manufacturing (240), Government (184), Banking (80) and Medical (79), although proportionally small, is quite alarming. Top targeted ports by the infected IoT devices were 23 [TELNET] (43.25%), 8080 [HTTP alt] (37.40%), 80 [HTTP] (37.16%), 81 [HTTP alt] (13.10%) and 5555 [ADB] (12.92%). Details about the top hosted ISPs and continents, and infected vendors are also provided in Table V.

VI. LIMITATIONS AND FUTURE DIRECTIONS

Gathering fine-grained details about compromised IoT devices (e.g., type, vendor, model, and firmware version) remains challenging. Empirical analysis reveals that less than 10% of the infected hosts return application banners and approximately 3% of them contain textual information which enable us to determine their detailed information. Further, eX-IoT was tested during a short period of initial deployment and needs to be assessed in the long run to gain insights regarding the challenges and opportunities in gaining a more rounded understanding of the IoT security posture.

VII. CONCLUSION

We introduce eX-IoT, a network-independent AI-empowered cyber threat intelligence capability for inferring compromised IoT devices. eX-IoT's feed is implemented to process more than 1M+ packets/sec of scan traffic arriving at our passive sensors, labeling the flows by analyzing the returned application banners, while also utilizing an online, adaptive training/fingerprinting model and by applying it on passive traffic data. The experimental evaluation shows that eX-IoT's CTI feed provides exclusive contribution of more than 0.76 with respect to other scan-based feeds. eX-IoT also reports on 145K+ newly compromised IoT devices daily and its CTI can be fed to organizations and security operators through an API, email notifications, and a visualization dashboard.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the anonymous reviewers and to our shepherd Marc Dacier for their constructive comments. We would also like to thank Bad Packets LLC and Czech Republic's CSIRT (in collaboration with Martin Husák) for for their help in the CTI validation part. This work was supported by a grant from the U.S. National Science Foundation (NSF) (Office of Advanced Cyberinfrastructure (OAC) 1907821).

REFERENCES

[1] Mark Hung. Leading the iot, gartner insights on how to lead in a connected world. *Gartner Research*, pages 1–29, 2017.

[2] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. *Computer*, 50(2):76–79, 2017.

[3] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. Sok: Security evaluation of home-based iot deployments. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1362–1380. IEEE, 2019.

[4] Ahmad Darki and Michalis Faloutsos. Riotman: a systematic analysis of iot malware behavior. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 169–182, 2020.

[5] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. Profiliot: a machine learning approach for iot device identification based on network traffic analysis. In *Proceedings of the symposium on applied computing*, pages 506–509. ACM, 2017.

[6] Harm Griffioen and Christian Doerr. Examining mirai's battle over the internet of things. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 743–756, 2020.

[7] Tongbo Luo, Zhaoyan Xu, Xing Jin, Yanhui Jia, and Xin Ouyang. Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices. *Black Hat*, 2017.

[8] Lionel Metongnon and Ramin Sadre. Beyond telnet: Prevalence of iot protocols in telescope and honeypot measurements. In *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity*, pages 21–26, 2018.

[9] Armin Ziaie Tabari and Xinming Ou. A multi-phased multi-faceted iot honeypot ecosystem. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 2121–2123, 2020.

[10] Claude Fachkha and Mourad Debbabi. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Communications Surveys & Tutorials*, 18(2):1197–1227, 2015.

[11] Elias Bou-Harb, Mourad Debbabi, and Chadi Assi. A novel cyber security capability: Inferring internet-scale infections by correlating malware and probing activities. *Computer Networks*, 94:327–343, 2016.

[12] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1093–1110, 2017.

[13] Orçun Çetin, Carlos Gañán, Lisette Altena, Takahiro Kasama, Daisuke Inoue, Kazuki Tamiya, Ying Tie, Katsunari Yoshioka, and Michel van Eeten. Cleaning up the internet of evil things: Real-world evidence on isp and consumer efforts to remove mirai. In *NDSS*, 2019.

[14] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. Measurement and analysis of hajime, a peer-to-peer iot botnet. In *NDSS*, 2019.

[15] Farooq Shaikh, Elias Bou-Harb, Nataliia Neshenko, Andrea P Wright, and Nasir Ghani. Internet of malicious things: correlating active and passive measurements for inferring and characterizing internet-scale unsolicited iot devices. *IEEE Communications Magazine*, 56(9):170–177, 2018.

[16] Morteza Safaei Pour, Elias Bou-Harb, Kavita Varma, Nataliia Neshenko, Dimitris A Pados, and Kim-Kwang Raymond Choo. Comprehending the iot cyber threat landscape: A data dimensionality reduction technique to infer and characterize internet-scale iot probing campaigns. *Digital Investigation*, 28:S40–S49, 2019.

[17] Sadegh Torabi, Elias Bou-Harb, Chadi Assi, El-Mouatez Billah Karbab, Amine Boukhtouta, and Mourad Debbabi. Inferring and investigating iot-generated scanning campaigns targeting a large network telescope. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[18] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. Acquisitional rule-based engine for discovering internet-of-things devices. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 327–341, 2018.

[19] Talha Javed, Muhammad Haseeb, Muhammad Abdullah, and Mobin Javed. Using application layer banner data to automatically identify iot devices. *ACM SIGCOMM Computer Communication Review*, 50(3):23–29, 2020.

[20] Qiang Li, Xuan Feng, Haining Wang, and Limin Sun. Discovery of internet of thing devices based on rules. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2018.

[21] Kai Yang, Qiang Li, and Limin Sun. Towards automatic fingerprinting of iot devices in the cyberspace. *Computer Networks*, 148:318–327, 2019.

[22] Arturs Lavrenovs and Gabor Visky. Exploring features of http responses for the classification of devices on the internet. In *2019 27th Telecommunications Forum (TELFOR)*, pages 1–4. IEEE, 2019.

[23] Artūrs Lavrenovs, Roman Graf, and Kimmo Heinäaro. Towards classifying devices on the internet using artificial intelligence. In *2020 12th International Conference on Cyber Conflict (CyCon)*, volume 1300, pages 309–325. IEEE, 2020.

[24] Zhaoteng Yan, Shichao Lv, Yueyang Zhang, Hongsong Zhu, and Limin Sun. Remote fingerprinting on internet-wide printers based on neural network. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[25] Deepak Kumar, Kelly Shen, Benton Case, Deepali Garg, Galina Alperovich, Dmitry Kuznetsov, Rajarshi Gupta,

and Zakir Durumeric. All things considered: an analysis of iot devices on home networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1169–1185, 2019.

[26] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50:80–84, 2017.

[27] Agathe Blaise, Mathieu Bouet, Stefano Secci, and Vania Conan. Split-and-merge: Detecting unknown botnets. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 153–161. IEEE, 2019.

[28] Agathe Blaise, Mathieu Bouet, Vania Conan, and Stefano Secci. Detection of zero-day attacks: An unsupervised port-based approach. *Computer Networks*, 180:107391, 2020.

[29] João Marcelo Ceron, Klaus Steding-Jessen, Cristine Hoepers, Lisandro Zambenedetti Granville, and Cíntia Borges Margi. Improving iot botnet investigation using an adaptive network layer. *Sensors*, 19(3):727, 2019.

[30] Roberto Perdisci, Thomas Papastergiou, Omar Alrawi, and Manos Antonakakis. Iotfinder: Efficient large-scale identification of iot devices via passive dns traffic analysis. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 474–489. IEEE, 2020.

[31] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018.

[32] Ibbad Hafeez, Markku Antikainen, Aaron Yi Ding, and Sasu Tarkoma. Iot-keeper: Detecting malicious iot network activity using online traffic analysis at the edge. *IEEE Transactions on Network and Service Management*, 17(1):45–59, 2020.

[33] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. Reading the tea leaves: A comparative analysis of threat intelligence. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 851–867, 2019.

[34] Shodan. The search engine for internet of things. http://shodan.io, 2020.

[35] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J Alex Halderman. A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553, 2015.

[36] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. Zmap: Fast internet-wide scanning and its security applications. In *22nd {USENIX} Security Symposium ({USENIX} Security 13)*, pages 605–620, 2013.

[37] Zmap. Zgrab. https://github.com/zmap/zgrab2, 2020.

[38] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. Measuring {HTTPS} adoption on the web. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1323–1338, 2017.

[39] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J Alex Halderman. Towards a complete view of the certificate ecosystem. In *Proceedings of the 2016 Internet Measurement Conference*, pages 543–549, 2016.

[40] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. You've got vulnerability: Exploring effective vulnerability notifications. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 1033–1050, 2016.

[41] ZoomEye. Zoomeye - cyberspace search engine. https://www.zoomeye.org/, 2020.

[42] Dshield. Dshield. https://isc.sans.edu/, 2020.

[43] GreyNoise. Greynoise. https://greynoise.io/, 2020.

[44] Elias Bou-Harb, Nour-Eddine Lakhdari, Hamad Binsalleeh, and Mourad Debbabi. Multidimensional investigation of source port 0 probing. *Digital Investigation*, 11:S114–S123, 2014.

[45] Jaeyeon Jung, Vern Paxson, Arthur W Berger, and Hari Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 211–225. IEEE, 2004.

[46] Morteza Safaei Pour, Sadegh Torabi, Elias Bou-Harb, Chadi Assi, and Mourad Debbabi. Stochastic modeling, analysis and investigation of iot-generated internet scanning activities. *IEEE Networking Letters*, 2(3):159–163, 2020.

[47] Morteza Safaei Pour, Antonio Mangino, Kurt Friday, Matthias Rathbun, Elias Bou-Harb, Farkhund Iqbal, Sagar Samtani, Jorge Crichigno, and Nasir Ghani. On data-driven curation, learning, and analysis for inferring evolving internet-of-things (iot) botnets in the wild. *Computers & Security*, 91:101707, 2020.

[48] Morteza Safaei Pour, Antonio Mangino, Kurt Friday, Matthias Rathbun, Elias Bou-Harb, Farkhund Iqbal, Khaled Shaban, and Abdelkarim Erradi. Data-driven curation, learning and analysis for inferring evolving iot botnets in the wild. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–10, 2019.

[49] Antonio Mangino, Morteza Safaei Pour, and Elias Bou-Harb. Internet-scale insecurity of consumer internet of things: An empirical measurements perspective. *ACM Transactions on Management Information Systems (TMIS)*, 11(4):1–24, 2020.

[50] CAIDA. Ucsd network telescope—near-real-time network telescope dataset. https://www.caida.org/data/passive/telescope-near-real-time_dataset.xml, 2020.

[51] https://docs.openstack.org/swift/latest/, author=OpenStack, title=Welcome to Swift's documentation!, year=2020.

[52] Shane Alcock, Perry Lorier, and Richard Nelson. Libtrace: a packet capture and analysis library. *ACM*

SIGCOMM Computer Communication Review, 42(2):42–48, 2012.
[53] V GTK-Doc. Glib reference manual, 2010.
[54] Morteza Safaei Pour and Elias Bou-Harb. Theoretic derivations of scan detection operating on darknet traffic. *Computer Communications*, 147:111–121, 2019.
[55] Morteza Safaei Pour and Elias Bou-Harb. Implications of theoretic derivations on empirical passive measurements for effective cyber threat intelligence generation. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.
[56] Debian. Socat. https://packages.debian.org/sid/socat, 2020.
[57] Thomas Maier-Komor. home of measuring buffer. http://www.maier-komor.de/mbuffer.html, 2020.
[58] Rapid7. Recog. https://github.com/rapid7/recog, 2020.
[59] Zmap. Ztag. https://github.com/zmap/ztag, 2020.
[60] Harm Griffioen and Christian Doerr. Discovering collaboration: Unveiling slow, distributed scanners based on common header field patterns. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2020.
[61] Vincent Ghiëtte, Norbert Blenn, and Christian Doerr. Remote identification of port scan toolchains. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2016.
[62] DHS. Information marketplace for policy and analysis of cyber-risk trust. https://www.dhs.gov/science-and-technology/cybersecurity-impact, 2021.
[63] https://github.com/eX-IoTsubmission/eX_IoT_API.
[64] BP. Bad packets cti. https://badpackets.net/threat-intelligence/, 2021.
[65] Václav Bartoš. Nerd: Network entity reputation database. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–7, 2019.
[66] Nima Ebadi, Mohsen Jozani, Kim-Kwang Raymond Choo, and Paul Rad. A memory network information retrieval model for identification of news misinformation. *IEEE Transactions on Big Data*, 2021.
[67] Martin Husák, Nataliia Neshenko, Morteza Safaei Pour, Elias Bou-Harb, and Pavel Čeleda. Assessing internet-wide cyber situational awareness of critical sectors. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–6, 2018.