Computers & Security 91 (2020) 101707

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

On data-driven curation, learning, and analysis for inferring evolving internet-of-Things (IoT) botnets in the wild

Morteza Safaei Pour[a,*], Antonio Mangino[a], Kurt Friday[a], Matthias Rathbun[b], Elias Bou-Harb[a], Farkhund Iqbal[c], Sagar Samtani[d], Jorge Crichigno[e], Nasir Ghani[f]

[a] The Cyber Center for Security and Analytics, University of Texas at San Antonio, Texas, USA
[b] Florida Atlantic University, Florida, USA
[c] College of Technological Innovation, Zayed University, Abu Dhabi, UAE
[d] Department of Information Systems and Decision Sciences, University of South Florida, Florida, USA
[e] Integrated Information Technology, University of South Carolina, Columbia, USA
[f] Department of Electrical Engineering and Cyber Florida, University of South Florida, Florida, USA

ARTICLE INFO

Article history:
Received 26 July 2019
Revised 28 November 2019
Accepted 24 December 2019
Available online 27 December 2019

Keywords:
Data science
Cyber forensics
Internet-of-things
IoT Security
Internet measurements

ABSTRACT

The insecurity of the Internet-of-Things (IoT) paradigm continues to wreak havoc in consumer and critical infrastructures. The highly heterogeneous nature of IoT devices and their widespread deployments has led to the rise of several key security and measurement-based challenges, significantly crippling the process of collecting, analyzing and correlating IoT-centric data. To this end, this paper explores macroscopic, passive empirical data to shed light on this evolving threat phenomena. The proposed work aims to classify and infer Internet-scale compromised IoT devices by solely observing one-way network traffic, while also uncovering, reporting and thoroughly analyzing "in the wild" IoT botnets. To prepare a relevant dataset, a novel probabilistic model is developed to cleanse unrelated traffic by removing noise samples (i.e., misconfigured network traffic). Subsequently, several shallow and deep learning models are evaluated in an effort to train an effective multi-window convolutional neural network. By leveraging active and passing measurements when generating the training dataset, the neural network aims to accurately identify compromised IoT devices. Consequently, to infer orchestrated and unsolicited activities that have been generated by well-coordinated IoT botnets, hierarchical agglomerative clustering is employed by scrutinizing a set of innovative and efficient network feature sets. Analyzing 3.6 TB of recently captured darknet traffic revealed a momentous 440,000 compromised IoT devices and generated evidence-based artifacts related to 350 IoT botnets. Moreover, by conducting thorough analysis of such inferred campaigns, we reveal their scanning behaviors, packet inter-arrival times, employed rates and geo-distributions. Although several campaigns exhibit significant differences in these aspects, some are more distinguishable; by being limited to specific geo-locations or by executing scans on random ports besides their core targets. While many of the inferred botnets belong to previously documented campaigns such as `Hide and Seek`, `Hajime` and `Fbot`, newly discovered events portray the evolving nature of such IoT threat phenomena by demonstrating growing cryptojacking capabilities or by targeting industrial control services. To motivate empirical (and operational) IoT cyber security initiatives as well as aid in reproducibility of the obtained results, we make the source codes of all the developed methods and techniques available to the research community at large.

1. Introduction

With the escalating adoption of the Internet-of-Things (IoT) paradigm in critical infrastructure, smart homes, transportation, and numerous other realms, an increasing number of devices are becoming directly Internet-facing. Although IoT devices deployed behind a Network Address Translation (NAT) gateway might be less vulnerable to Internet-enabled attacks, a plethora of such devices are directly connected to the Internet and/or employ port-forwarding for simplified provisioning and management (Da Xu et al., 2014). Unfortunately, such devices often lack basic security protocols and measures, rendering them easy targets for exploitations and hence recruitment within coordinated IoT botnets (Bertino and Islam, 2017). Additionally, there exists several inher-

* Corresponding author.
E-mail address: morteza.safaeipour@utsa.edu (M. Safaei Pour).

https://doi.org/10.1016/j.cose.2019.101707
0167-4048/

ent IoT factors such as their heterogeneous nature and limited processing resources which further complicate addressing necessary security requirements. At the same time, subpar attention is being paid to IoT security aspects by their manufacturers and users, on top of an overwhelming lack of maturity of IoT-specific update procedures for patch management (Bertino and Islam, 2017).

Indeed, IoT security has been an emerging area of focus after `Mirai` (Antonakakis et al., 2017) infected more than 200,000 devices to conduct debilitating Distributed Denial of Service (DDoS) attacks in late 2016, demonstrating the sheer malicious capabilities by way of exploiting IoT devices. Thereafter, botnets consisting of IoT devices have consistently been evolving, incorporating new devices and services. Hence, the IoT botnet environment has expanded to include several more players who ultimately compete for control over insecure IoT devices by means of newly-exposed vulnerabilities. In June 2019, the `Echobot` (Cashdollar, 2019; Nigam, 2019) campaign was identified as operating "in the wild". Derived from `Mirai`'s source code, `Echobot` has compromised millions of IoT nodes. Exploiting more than 20 unique (software and firmware) IoT-centric vulnerabilities, the campaign has infected devices across more than 10 diverse vendors. Indeed, this IoT threat phenomena will undoubtedly continue to display highly dynamic behavior as they attempt to propagate and exploit a higher number of devices, making the inference, attribution, and assessment of compromised IoT devices and their coordinated illicit activities significantly challenging.

Despite efforts to mitigate the ever-increasing IoT security issues, challenges exist due to the heterogeneity of IoT devices and the emergence of anti-honeypot techniques to avoid discovery (Dowling et al., 2018; Luo et al., 2017; Nawrocki et al., 2016). Moreover, acquiring IoT-centric empirical data to be curated and analyzed for maliciousness is problematic, given the large-scale deployments of such devices in Internet-wide realms. While network telescope (darknet) traffic (Fachkha and Debbabi, 2016) (i.e., Internet-scale traffic targeting routable yet unused IP addresses) has proven to be a reliable and effective source for generating insights related to Internet-wide maliciousness (Fachkha and Debbabi, 2016), its exploration for addressing IoT security issues is still in its infancy. Broadly, a major challenge related to the inference of IoT maliciousness through the analysis of network telescope traffic is the lack of sound data-driven artifacts which can be analyzed to confirm that the perceived one-way traffic is in fact originating from IoT devices and not from typical machines. Further, successful darknet-driven methodologies should accommodate the evolving nature of IoT botnets, leveraging their empirical specifications as perceived by the (somehow limited) vantage point of the network telescope.

Correlating darknet-inferred IP addresses with databases such as `Shodan` (Shodan, 2019) or `Censys` (Team, 2017) has proven to be a successful use-case for classifying IoT-centric data and Internet-scale exploitations (Shaikh et al., 2018; Torabi et al., 2018). Both `Shodan` and `Censys` use IP crawlers, active scanning, and banner grabbing to collect and index open ports and available services on billions of Internet-facing IoT devices. While this strategy provides large-scale device information, the limited scope of services reachable by `Shodan` and `Censys` scanners makes them incapable for identifying the complete Internet-wide set of active IoT devices. Such generated probes and active measurements are typically filtered by firewalls. Additionally, upon infection, IoT malware tend to block ports, modify banner information and disable common outward facing services (i.e., Telnet, CWMP, ADB, etc.) (Antonakakis et al., 2017; Herwig et al., 2019). When the aforementioned events occur, the indexing of IoT devices is significantly impeded.

Having noted this, a number of darknet-related technical challenges exist which further hinder IoT-centric fingerprinting efforts. Indeed, perceived packets on the network telescope (that have been generated by IoT bots) solely resemble scan activities (i.e., do not include payload information and are unidirectional), which limits the amount of available data to analyze. Furthermore, only a small portion of unsolicited IoT-generated traffic actually targets deployed network telescopes, rendering time-based analysis non-trivial and complicates the process of extracting effective and robust features to infer orchestration behaviors of compromised IoT devices.

Motivated by the aforementioned limitations coupled with the lack of thorough measurement-based studies on the insecurity of the IoT paradigm at large, this paper contributes by proposing a multi-threaded, generic methodology for scrutinizing macroscopic darknet data to design, develop and evaluate:

- A novel darknet-specific, formal sanitization model that systematically identifies and filters out misconfiguration traffic to permit the storage and processing of network telescope data. The proposed darknet sanitization model does not rely on arbitrary cut-off thresholds, but instead provides likelihood models to distinguish between misconfiguration and other forms of darknet traffic, independent from the nature of the traffic sources. As a result, the proposed model neatly captures the natural behavior of darknet-targeted misconfiguration traffic.
- An IoT-centric fingerprinting approach rooted in deep learning and active measurements methodologies to infer Internet-scale compromised IoT devices by exclusively operating on network telescope data. The addressed problem herein is illustrated in Fig. 1a. Using more than 3 TB of recent darknet data, the outcome of such a proposed approach exposes more than 400,000 compromised IoT devices from very well-known vendors. The results highlight that more than 75% of all the inferred IoT bots do not match the typical `Mirai` signature (Antonakakis et al., 2017), concurring the evolving nature of this threat phenomena and highlighting the added-value of the proposed methodology.
- An IoT-specific botnet inference methodology based upon effective and lightweight (darknet) data-driven features and hierarchical agglomerative clustering. The addressed problem herein is shown in Fig. 1b. The results from instrumenting such an approach uncover more than 300 "in the wild" IoT botnets, where close to 25 campaigns contain over 1000 exploited, well-coordinated IoT bots. Moreover, IoT botnet-specific traits are investigated, including scanning modules, probing rates and their geo-distributions. While the results shed light on previously documented IoT botnets that were found to still be active, the outcome also uncovers new IoT botnets such as those possessing cryptojacking capabilities (which were shown to be coordinated by the same "player" due to the usage of the same key) and those that were inferred to be targeting industrial control systems. To facilitate the reproducibility of the results in addition to motivate passive Internet measurements for IoT security, we make all the developed methods and techniques available to the research and operational communities at large via https://github.com/COYD-IoT/COYD-IoT.

The remainder of this paper is organized as follows. Section 2 reviews the literature related to network telescope research, IoT device fingerprinting and IoT botnet analysis to demonstrate the state-of-the-art contributions of this work. In Section 3, we detail the darknet pre-processing model, the studied machine/deep learning models for fingerprinting compromised IoT devices, in addition to elaborating on the IoT-centric botnet inference methodology. In Section 4, we report and discuss the results derived from executing the proposed approach. Finally, Section 5 summarizes the contributions of this paper and paves the way for future work by addressing a number of limitations.

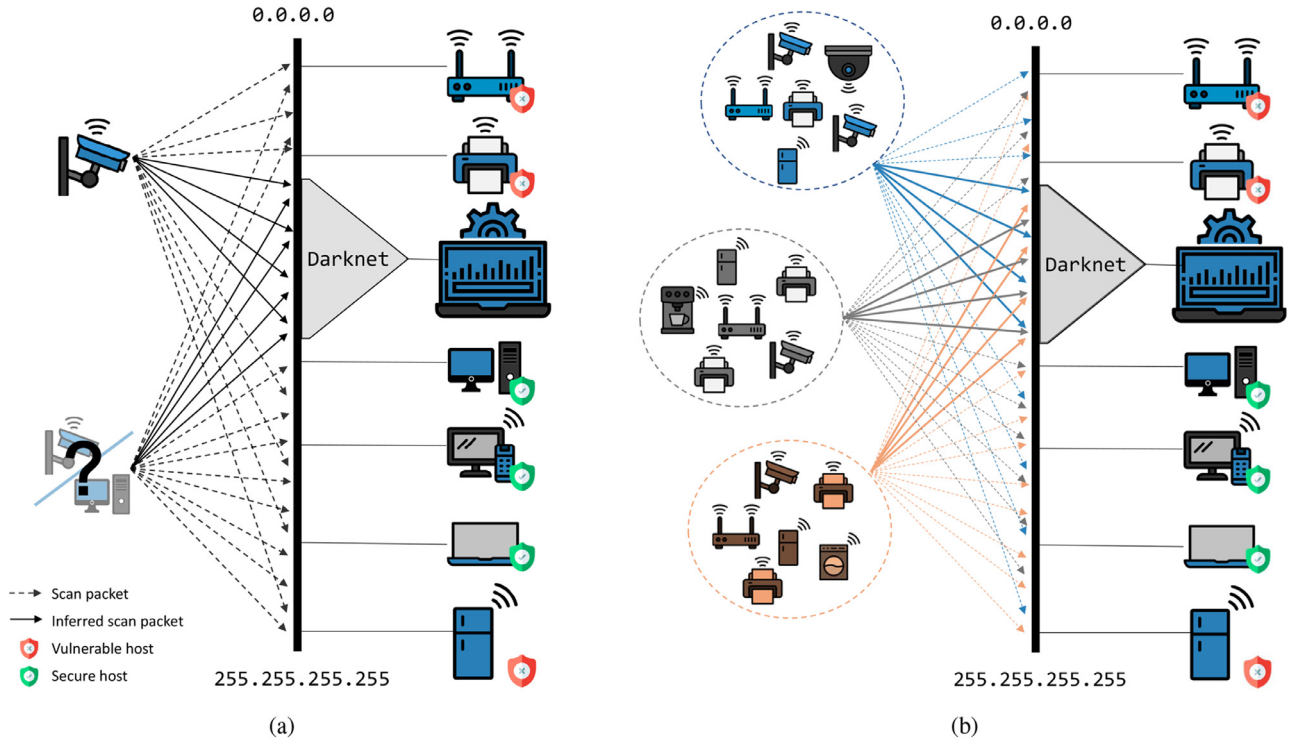


Fig. 1. Leveraging network telescopes to (a) devise learning techniques for fingerprinting IoT devices; and (b) develop clustering methods for identifying campaigns of orchestrated IoT devices.

2. Related work

In this section, we review three topics central to the contemporary IoT security landscape. The first focuses on network telescopes as a powerful mechanism to capture IoT-specific, illicit network traffic. The second summarizes efforts pertaining to IoT device fingerprinting. Finally, we enumerate the literature related to IoT-specific botnet analysis.

2.1. Network telescopes and IoT security

A network telescope (i.e., darknet), is a set of routable, allocated, yet unused IP addresses deployed in order to passively observe incoming Internet-scale traffic (Fachkha and Debbabi, 2016). Since these IP addresses are not associated with any services, traffic targeting them is unsolicited (Bou-Harb et al., 2016). Such traffic originates from infected devices scanning the Internet space, victims of Denial of Service (DoS) attacks, or misconfiguration caused by hardware/software errors or improper routing. Network telescopes are reliable sources for investigating large-scale, Internet-wide activities, which is supported by recent examples of successful applications including studies on probing activities (Dainotti et al., 2015) and DDoS attacks (Fachkha et al., 2015; Moore et al., 2006). In the context of assessing the maliciousness of IoT devices through network telescopes, Torabi et al. (2018) recently conducted large-scale correlations between passive measurements and IoT-relevant information to investigate and disclose malicious activities associated with more than 26,000 IoT devices, including those within critical infrastructure. Leveraging a large-scale network telescope, the authors categorized the ports and services targeted by scans within the network telescope, attributing them to infected IoT devices and identifying active threats (i.e, IoT devices launching brute-force SSH attacks). Similarly, by correlating active measurements with collected network telescope data, Shaikh et al. (2018) examined nearly 14,000 compromised IoT devices and extracted malicious signatures for mitigation in IoT hosting environments. Further investigation of the collected network traffic revealed that nearly 20% of the identified IoT devices were related to DDoS attacks. Moreover, by means of applying filters to network telescope data in order to discern `Mirai`-relevant traffic, Antonakakis et al. (2017) were able to gather IoT-related information pertaining to roughly 1.2 million `Mirai`-infected IP addresses during 8 months. Their work revealed crucial details of the `Mirai` malware's attack vectors, such as targeted ports (TCP/Telnet:23 and TCP/Telnet:2323). Furthermore, after correlating their results with `Censys` (Durumeric et al., 2015a) scans, the authors fingerprinted the device types of `Mirai`-infected bots - confirming the IoT-centric composition of the botnet. In a related study, Cetin et al. (2019) collected network traffic across a 300,000 IP darknet space to conduct empirical studies focusing on IoT malware cleanup efforts and remediation rates in a medium-sized Internet Service Provider (ISP). Combining network traffic received within their darknet with malware binaries retrieved from an IoT-based honeypot and IP addresses retrieved from Internet scanners `Censys` and `Nmap`, the authors tracked the success (and failure) of remediation efforts, reporting device reinfection rates.

While such contributions are noteworthy, several shortcomings exist. First, previous works rely on a specific IoT malware signature (e.g., the `Mirai`-specific signature of tcpSeq == dstIP). Not every IoT bot will follow such an easily identifiable signature, preventing comprehensive identification of Internet-scale botnets "in the wild". In fact, our measurements have revealed that less than 25% of all the inferred IoT bots match the `Mirai`-specific signature. Second, the majority of these related works solely depend on databases gathered by Internet scanning services (e.g., Censys and Shodan), which might not be able to accurately identify every infected IoT device at a global scale. Antonakakis et al. discovered that upon infection, the `Mirai` malware closed a number of ports and services on newly exploited IoT bots to prevent infection by competing malware (Antonakakis et al., 2017). As a result

of such territorial nature, Internet search engines are unable to discover a large portion of infected devices due to the similarities between their discovery methods and malware scanning trends. In contrast, we aim to create a more comprehensive view of the aforesaid IoT bot populations by proposing a novel approach that synergistically leverages passive, darknet-centric assessments coupled with active, Internet-scale measurements and machine/deep learning techniques.

2.2. IoT device fingerprinting

Previous works that propose IoT inference methods rely on text information retrieved from service banners, gathered by active measurements (e.g., port scanning and banner grabbing), or provided by Internet scanning services similar to Shodan (Shodan, 2019), Censys (Durumeric et al., 2015a) and ZoomEye (ZoomEye, 2019). For example, Kumar et al. (2019) leveraged predefined text rules from Censys to fingerprint consumer IoT devices, designing an ensemble of four supervised classifiers on UPnP and DNS responses, HTTP data banners, and network-layer information. Alternatively, several research efforts have elected to attempt IoT device fingerprinting by solely observing network traffic. For instance, Guo and Heidemann (2018) postulated that because IoT devices regularly exchange data with servers managed by their manufacturers, IoT device type and vendor can be fingerprinted by observing the exchanged flow-level network traffic between devices and servers. After discovering nearly 200 candidate servers accessed by 26 devices across 15 vendors, their methodology successfully identified IoT devices connected across the University of Southern California (USC). In another work, Meidan et al. (2017) explored a localized lab environment to extract TCP packet features from a variety of IoT devices, including baby monitors, IP cameras, and printers. Extracted features were employed to train a supervised learning algorithm in order to distinguish between IoT devices and non-IoT. Moreover, Miettinen et al. (2017) leveraged network traffic generated by IoT devices during their setup process for capturing device-specific traits. A number of automatic requests and updates were collected and subsequently mapped as signatures, detailing the device type by way of random forest classification. Improving upon anomaly detection premised on device types, Nguyen et al. (2018) have recently implemented a machine learning algorithm which discriminates between the corresponding classes of devices, but exhibited remarkable performance with detection rate 95.6%. Similarly, Thangavelu et al. (2018) developed a machine learning-based methodology capable of fingerprinting distributed devices. Their work overcomes previous limitations hindering centralized approaches by offering a scalable and dynamic methodology.

Pinheiro et al. (2019) developed algorithms for distinguishing between IoT and non-IoT devices based upon packet specifications. The mean and standard deviation of the packet length were combined with the number of bytes transmitted by each device in one-second time intervals to accurately profile devices. Further, Siby et al. (2017) detected devices in a local network by passively intercepting and recording wireless signals. Extracting the encapsulated MAC addresses from investigated flows allowed for IoT device identification. In an alternative approach, Acar et al. (2018) implemented a web script to identify the presence of IoT devices running local HTTP servers. Once identified, IoT vulnerabilities are disclosed, specifically the unauthorized accessing of such IoT devices through DNS rebinding.

A shortcoming of the aforementioned literature is that their scope is limited to local IoT networks. Therefore, they do not present an Internet-wide perspective; hence, their proposed approaches are not applicable on one-way scan flows arriving at network telescopes. In contrast, we leverage a large-scale network telescope to collect Internet-wide network traffic, followed by the deployment of a strict rule set used to fingerprint hosts that respond with banners when probed. Additionally, we devise learning techniques to identify unreachable infected hosts and predict their device types using innovative features extracted from sequences of TCP SYN packets arriving at the network telescope.

2.3. IoT botnet analysis

The discovery and analysis of IoT-centric botnets reveal crucial cyber threat intelligence relating to the discovery of malware attack vectors and disclose possible vulnerabilities or intrusion points within globally-deployed IoT devices. Within the context of botnet analysis through tailored honeypots, Pa et al. (2016) inferred several malware families by constructing a honeypot to analyze attacks against Telnet services. Dubbed as IoTPOT, the honeypot was specifically tailored to mimic the CPU architectures of various IoT devices, while learning how to respond to command interactions. Furthermore, Guarnizo et al. (2017) designed the IoT-centric Scalable high-Interaction Honeypot (SIPHON) which demonstrated effectiveness to attract a tremendous amount of malicious IoT botnet-generated traffic (ranging from 50,000 to 600,000 attempted TCP connections) through a combination of worldwide wormholes and a small number of deployed IoT devices. The deployed SIPHON honeypot recorded hundreds of brute-force password attacks and retrieved credential dictionaries used for these attacks. Moreover, Metongnon and Sadre (2018) reported on a large number of exploited IoT protocols, based on an in-depth analysis of network traffic from IoT-centric honeypots and network telescopes.

While such works provide crucial IoT-centric botnet analysis, given the copious amounts of IoT hardware in the wild and their accompanying heterogeneity, we note that honeypot-based methodologies frequently fail at mimicking the entirety of IoT device and firmware vulnerabilities. However, capturing such characteristics are essential to characterizing and attributing large-scale IoT botnets. Additionally, the vantage points of honeypots are typically quite small, hindering their effectiveness in tracking Internet-scale IoT botnets as well as accurately estimating their population size.

Rather than deploying honeypots, alternative works attempted to identify compromised IoT devices (bots) in local networks. For example, Meidan et al. (2018) proposed a novel, host-based intrusion detection system (IDS) that monitors a device's typical behaviors through analyzing its network traffic using autoencoders. The IDS creates a snapshot of what the device is expected to be communicating, and will subsequently raise an alarm if any deviations or anomalies are detected. To evaluate the effectiveness of the proposed IDS, nine commercial IoT devices were deployed in a controlled environment to generate benign traffic. These devices were then infected with two very notorious IoT malware, Mirai and BASHLITE, and tested the IDS capabilities for identifying the newly corrupted traffic flows. Further, Nguyen et al. (2019) present an autonomous self-learning distributed system for detecting compromised IoT devices, leveraging federated learning technique to aggregate IoT device fingerprints. These fingerprints are then clustered and categorized based on device type and models. Next, a K-Nearest Neighbors classifier identified abnormal network traffic to discover compromised IoT devices. Evaluated on 30 different devices and selected Mirai malware for the real-world case study, ultimately, the methodology achieved a 94% detection rate and 257 ms average detection time.

Alternatively, other literature works consider a macroscopic approach for generic botnet analysis by aggregating information from various sources. To this end, Gu et al. (2008) proposed a system for correlating aggregated IDS log files with extracted features from network flows to detect botnet activities. The system

Table 1
Summary of selected recent botnet detection literature.

Publication	Employed Data Type	Botnet Class	Methodology and Evaluation
Metongnon and Sadre (2018)	Honeyfarm/ Network Telescope	IoT	Passive measurement and analysis of real-world IoT honeyfarm traffic.
Meidan et al. (2018)	Two-way Traffic	IoT	Training on normal behavior using autoencoders. Evaluated on 9 IoT devices in a lab environment tested with `Mirai` and `Bashlite`.
Nguyen et al. (2019)	Two-way Traffic	IoT	Federated learning. Evaluated on 30 IoT devices in a lab environment tested with `Mirai`.
Gu et al. (2008)	NetFlow/ Aggregated IDS log	Generic	Correlation and two-step clustering. Evaluated on real-world dataset.
Homayoun et al. (2018)	NetFlow	Generic	Autoencoder/CNN. Evaluated on combined botnet traffic ISCX UNB (2019).
Araki et al. (2019)	xFlow	Generic	Two-step Subspace Clustering. Tested on real-world ISP traffic and MAWI (Fontugne et al., 2010).
Ozawa et al. (2019)	Network Telescope	Generic	Association rule mining evaluated on real-world /16 network telescope.
Antonakakis et al. (2017)	Network Telescope/ Censys/ Passive DNS/ Telnet Honeypots/ Malware binaries	`Mirai`	`Mirai` signature tested on /10 network telescope.
Herwig et al. (2019)	Active Scanning/ Root DNS backscatter traffic	`Hajime`	Bug in P2P infrastructure while observing the effect on samples of all queries to the D-root DNS root server.

assumed that hosts infected with the same malware behave in a similar manner and that bot-generated flows captured within the IDS logs will share many of the same characteristics. Subsequently, they were clustered in a two-stage, high dimensional classifier for identifying botnet campaigns. The evaluation results on a university campus reveal a botnet detection rate of 99.6%. Similarly, Homayoun et al. (2018) proposed `BoTShark`, primarily using deep learning models such as autoencoders and convolutional neural networks relying on captured netflows to efficiently detect malicious traffic. With a true positive rate of 0.91 and a false positive rate of 0.13, `BoTShark` successfully detected malicious traffic signatures of botnet campaigns.

Additionally, different works attempted to identify and characterize IoT botnets through passively collecting one-way network traffic. Araki et al. (2019) proposed a methodology that not only detected bots, but classified their primary behaviors and characteristics. Utilizing a two-step subspace clustering method to cluster botnets and clarify partial characteristics (such as low-size flows or high TCP-SYN packet rates), the methodology was evaluated on two real-world datasets, collected by upstream, backbone ISPs. Similarly, Ozawa et al. (2019) studied IoT botnet characteristics by way of analyzing their scan activities. The authors applied associate rule learning (Agrawal et al., 1993) on network telescope features such as destination ports, ToS, and TCP window sizes to discover the activities of bots that were infected with well-known malware. Their work reported interesting observations on the evolution of IoT botnet characteristics before and after the release of the `Mirai`'s source code.

In contrast, other works specifically focused on a single IoT botnet family to retrieve relevant attack vectors and behavior, such as the `Mirai` botnet (Antonakakis et al., 2017). Another example includes, Herwig et al. (2019) who provided a comprehensive investigation related to the Hajime IoT botnet, revealing crucial insights such as Hajime's atypical infrastructure. Deviating from typical command and control infrastructures that rely on bots to communicate with infected servers to receive orders, instead relying on peer to peer connections between bots and utilizing the BitTorrent protocol to transfer payloads. To summarize such contributions, Table 1 provides a brief classification of recent selected works using different dimensions.

The aforementioned works present significant and important analysis of IoT botnets; however, a number of limitations prevent them from offering a generic, Internet-wide perspective of global IoT botnet populations. Target-specific studies designed to investigate a singular IoT botnet take advantage of known botnet infrastructure or signatures, and in turn cannot be replicated or generalized to study other IoT botnets. Furthermore, the vantage point offered by the honeypots, results in limited exposure when compared with one offered by a network telescope. To this end, in this work, we complement and expand upon previous contributions by developing a purely passive methodology to not only identify Internet-scale compromised IoT devices, but also to infer ongoing IoT botnets by capturing their orchestrated artifacts.

3. Proposed methodology

This section details the proposed approach as depicted in Fig. 2. Its core components include *(i)* data collection and dataset preparation, which introduces the darknet sanitization probabilistic model to filter out misconfiguration traffic along with the inference of Internet-scale probing activities and labeling their sources; *(ii)* the systematic evaluation of state-of-the-art machine learning and deep learning classifiers for fingerprinting compromised IoT devices; and *(iii)* the feature engineering process coupled with executing hierarchical agglomerative clustering to infer and report on IoT botnets. These steps are subsequently detailed.

3.1. Network telescope sanitization model

As previously noted, network telescopes, most commonly known as darknets (Fachkha and Debbabi, 2016), constitute a set of allocated and routable, yet unused, IP addresses. Since these addresses do not operate legitimate services, any traffic targeting them is considered unsolicited. From a deployment perspective, network telescopes are commonly distributed on specific Internet IP subspaces operated by Internet Service Providers (ISPs), educational entities and corporate backbone networks. Darknet IP addresses are, by nature, indistinguishable from other routable addresses, rendering them an effective technique to amalgamate Internet-wide, one-way unsolicited network traffic.

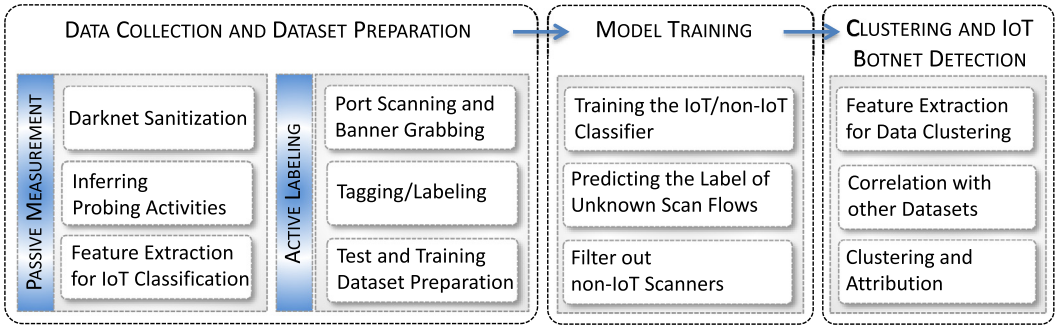


Fig. 2. The components of the proposed approach.

Although network telescope (darknet) data predominantly consists of malicious packets originating from probes, backscattered packets from victims of DDoS attacks, and malware propagation attempts, it also contains misconfiguration traffic. The latter Non-malicious packets frequently result from network, routing, hardware, or software faults that were erroneously directed towards a darknet. Such traffic might also be an artifact of improper configurations during darknet deployment. Misconfiguration traffic (Fachkha and Debbabi, 2016) impedes the proper functioning of cyber threat intelligence algorithms operating on darknet data, which often yields numerous undesirable false positives and false negatives and waste of valuable storage resources. Given the lack of formalism in addressing this problem, the objective herein is to elaborate on a probabilistic model that is specifically tailored towards the preprocessing of darknet data by way of fingerprinting, and in turn, filtering out embedded misconfiguration traffic.

In brief, the model formulates and computes the probability metrics of misconfigured traffic, while capturing the behavioral perspective of misconfiguration flows as they target the darknet space. Regarding the natural tendencies associated with typical network flows, the model initially estimates the rarity of hosts attempting to access the destination address. Secondly, the scope of access is considered, accounting for the number of distinct darknet IP addresses that a specific remote source has accessed, preserving the unique characteristics of the misconfiguration flow. Subsequently, the joint probability is formulated, computed, and compared. If the probability of the source generating a misconfiguration flow is higher than that of the source being malicious (or unsolicited), then that particular source is deemed to be generating misconfiguration traffic, flagged, and the corresponding flows are filtered out. In the following, we detail the notions of both rareness and scope of access.

Let $D = \{d_1, d_2, d_3, \cdots\}$ represent the set of darknet IP addresses, with D_i being a subset of those accessed by source s_i. First, the model captures how unusual these accessed destinations are. The underlying idea in doing so stems from the fact that misconfigured sources target destinations seldom called upon by others (Ford et al., 2006). Thus, the model estimates the distribution of a darknet IP d_i as being accessed by such a source as

$$P_{misc}(d_i) = \frac{n_s(d_i)}{\sum_{\forall d_j \in D} n_s(d_j)}, \tag{1}$$

where $n_s(d_i)$ is the number of sources that have accessed d_i; in contrast, a malicious darknet source will target a given destination at random. Typically, defining a suitable probability distribution to exemplify the randomness of a malicious source taking aim at a specific darknet destination is quite tedious; therefore, a simplistic assumption is often applied to resolve this potential headache. In this context, Durumeric et al. (2014) demonstrated that sources probe their darknet targets following a Gaussian distribution. By adopting that assumption, one can model the probability of a darknet destination being accessed by a malicious source as $P_{mal}(d_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$ where σ is the standard deviation, μ is the mean, σ^2 is the variance, and x is the location of the darknet destination following the aforementioned distribution. Recall that not only does the model capture how unusual the accessed destinations are, but it also considers the number of darknet destinations accessed by a particular source, which we subsequently describe. Given a set of D_i darknet destinations accessed by a specific source s_i, the model ultimately measures two probability distributions, namely, $P_{misc}(D_i)$ and $P_{mal}(D_i)$; the former being the probability that D_i has been generated by a misconfigured source and the latter originating from that with a malicious intent towards darknet D_i. For example, if the darknet addresses accessed by s_1 are $D_1 = \{d_{i1}, d_{i2}, d_{i3}\}$, $P(D_1)$ equates to the probability of s_1 accessing the specific combination of addresses $\{d_{i1}, d_{i2}, d_{i3}\}$ given three targeted destinations, multiplied by the probability of s_1 accessing any three destinations. In turn, we can generalize $P(D_1)$ as

$$P(D_i) = P(D_i = \{d_{i1}, d_{i2}, \cdots d_{in}\} \mid |D_i| = n) \times P(|D_i| = n). \tag{2}$$

For both a misconfigured and malicious source, the first term of Eq. (2) can be modeled as

$$P(D_i = \{d_{i1}, d_{i2}, \cdots\} \mid |D_i|) = \frac{1}{K} \prod_{\forall d_j \in D_i} P(d_i) \tag{3}$$

where K, acting as a normalization constant and solely being used as a means of summing the probabilities to 1, could be defined as $K = \frac{|D|!}{n!(|D|-n)!} \times \frac{1}{|D|^n}$. K is a standard normalization constant often employed in Bayesian probability (Gelman et al., 2014). Moreover, n encompasses all sources in the data, whereas $|D|$ represents the darknet IP space. Consequently, the likelihood that a source will target a certain number of darknet destinations (i.e., the second term of Eq. (2)) depends upon whether it is malicious or misconfigured. Characteristically, misconfigured sources access one or few destinations while those with malicious intent target a larger pool. Accordingly, we model such distributions as

$$P_{misc}(|D_i|) = \frac{1}{(e-1)|D_i|!} \tag{4}$$

$$P_{mal}(|D_i|) = \frac{1}{|D|}, \tag{5}$$

where the term $(e-1)$ in Eq. (4) ensures the distribution's summation equals 1. Eq. (4) guarantees a significant decrease in the probability as the number of targeted destinations increases. In contrast, Eq. (5) captures that of a random number of darknet addresses being accessed by a malicious source. Thereby, via plugging in of Eqs. (4) and (5) into (3), respectively, we can represent the probability of a source being either misconfigured or malicious,

given a set of darknet destination addresses, as

$$P_{misc}(D_i) = \frac{1}{K(e-1)|D_i|!} \prod_{\forall d_j \in D_i} P_{misc}(d_i) \tag{6}$$

$$P_{mal}(D_i) = \frac{1}{K|D|} \prod_{\forall d_j \in D_i} P_{mal}(d_i). \tag{7}$$

Eqs. (6) and (7) provide two distinct likelihood models to distinguish between misconfiguration and malicious, darknet-bound traffic, which enables their simplified and systematic post-processing. Furthermore, as the proposed model generalizes and formalizes the concepts of misconfiguration and malicious darknet traffic, it does not make any assumptions regarding the nature of the sources from which the given types of traffic are originating. Thus, the method deems a source and its corresponding flows as misconfiguration traffic if $\ln P_{misc}(D_i) - \ln P_{mal}(D_i) > 0$. To effectively employ the proposed network telescope sanitization model, we present Algorithm 1, which provides a simplistic yet effective mechanism to flag misconfigured sources.

Algorithm 1 Network Telescope Sanitization Algorithm.

Input: Darknet Flows, *DarkFlows*
Output: Flag, *MiscFlag*, indicating that the respective flow is originating from a misconfigured source
for *DarkFlows* **do**
  MiscFlag ← 0
  i ← *DarkFlows.getUniqueSources()*
  Amalgamate $DarkFlows_i$ originating from a specific source s_i
  Update $s_i(D_i)$
  Compute $P_{misc}(D_i), P_{mal}(D_i)$
  if $P_{misc}(D_i) > P_{mal}(D_i)$ **then**
    MiscFlag ← 1
  end if
end for

Since the field of Internet measurements for cyber security heavily relies on processing network telescope data (Fachkha and Debbabi, 2016; KoronIoTis et al., 2019), we make the model's code available to researchers and security operators at large from https://github.com/COYD-IoT/COYD-IoT/tree/master/Darknet%20Sanitization.

3.2. Data collection and dataset preparation

This section summarizes the methodology used to infer probing activities captured at a network telescope. We also detail the proposed mechanisms for feature engineering and active measurements, in order to fingerprint IoT devices through data-driven learning.

3.2.1. Inferring probing activities

After employing the aforementioned pre-processing steps to sanitize misconfiguration traffic, the aim is to dissect the malicious traffic in order to extract probing flows as perceived by a network telescope as indicators of exploitation. This is achieved through a Threshold Random Walk (TRW)-based probing detection algorithm (Jung et al., 2004). The TRW algorithm searches for subsequent packets from the same source IP address for a duration of 300 seconds. If the time-based threshold is exceeded without receiving a packet, the given counter is reset. If the threshold has held and the duration has not expired, the counter is incremented. If the counter reaches a threshold of 64 (Rossow, 2014), the flow is deemed as a probing event.

3.2.2. Feature extraction for IoT classification

Following the amalgamation of packets into flows, the first t consecutive packets are extracted from each. Given that the majority of the observed scanning traffic are TCP SYN scans, the applicable features reside in the TCP and IP header fields (i.e., ToS, Total Length, Identification, TTL, Dst IP Address, srcPort, dstPort, TCP SEQ, TCP ACK SEQ, TCP offset, TCP DATA Length, TCP Reserve, TCP Flags, TCP Win, TCP URP, TCP options, Packet Inter-arrival Time). Overall, along with the inter-arrival time of the consecutive packets within a flow, $d = 17$ features are gathered for each packet. In turn, the data samples for each scanner IP address would consist of a $t \times d$ matrix. To elaborate on the model's training procedure, we subsequently detail the labeling process.

3.2.3. Port scanning and banner grabbing

In order to annotate decidedly accurate labels for the training dataset, it was imperative to perform the procedure herein upon detection of a scan activity to circumvent any potential complications due to the dynamic reallocation of the associated device's IP address (through DHCP, for instance). To accomplish this, we utilized the gigabit open-source Internet scanning tool ZMap (Durumeric et al., 2013) as well as the high-speed application scanner ZGrab (Durumeric et al., 2015b), in tandem, to provide comprehensive results necessary for guaranteeing the versatility of the classification task. Specifically, ZMap was used to probe 45 ports[1] of the IP addresses (that were previously inferred as probing sources) that were found to be active. The port list is selected based on reports by ZoomEye (ZoomEye, 2019) during a one month analysis of returned banners, chosen to cover most of the default ports of various devices in order to maximize the number of captured banners. Furthermore, via ZGrab, we obtained banner fields and application handshakes from various protocols such as HTTP(s), CWMP, TELNET, SMTP(s), IMAP(s), POP3(s), SSH, FTP, SMB, DNP3, MODBUS, BACNET, FOX, Siemens S7 and SSL certificates. Additionally, we designed and developed two custom scanning modules to extract RTSP and SIP banners.

3.2.4. Tagging and labeling

To label discovered IoT devices, we amalgamated a comprehensive list of keywords related to major Internet-facing IoT devices and vendors. As previously noted, these are typically the devices that are most targeted by IoT botnets. This list consists of devices provided by `Nmap` along with results from ZoomEye Internet Scanner[2] (ZoomEye, 2019) and ZTag, Censys's tagging module.[3] Although it is unrealistic to claim that we cover all IoT products from every manufacturer and vendor, we carefully leverage information from various sources and focused on widely deployed Internet devices. In addition, we implemented a parsing algorithm which extracts useful keywords from banners and SSL certificates such as the combination of letters, digits, "-" and "_" signs, which typically represent device models (Feng et al., 2018) to enrich our list of devices. We further considered devices running multi-purpose OSs as non-IoT, which were identified using keywords such as "Win64", "Ubuntu", "Microsoft IIS" and "CentOS", etc. while we deemed other specialized devices as IoT where their OS types were indicated as being "embedded", "RouterOS", "FritzOS" etc. For example "TD-W8960N" is a sample keyword in the database related to a TP-LINK router that is marked as IoT. The prepared database consists of 3286 patterns related to 1121 vendors. We make this list publicly available at https://github.com/COYD-IoT/COYD-IoT/blob/master/devices.txt. Given that not all scanning events are illicit in

[1] https://github.com/COYD-IoT/COYD-IoT/blob/master/Port-List.txt.
[2] https://www.zoomeye.org/component.
[3] https://github.com/zmap/ztag/tree/master/ztag/annotations.

nature (e.g., academic institutions conducting research) such entities will incorporate information into their banners. Thereby, we leverage banners, coupled with a Greynoise list (GreyNoise, 2019), to filter out these benign scanners[4].

3.3. Model training for fingerprinting compromised IoT devices

We propose herein a learning approach for the extraction of embedded features within unsolicited scan flows for the training of a binary classifier which distinguishes between traffic originating from both malicious IoT and non-IoT devices. The underlying methodology is based upon determining similarities in network traffic that are exclusively associated with IoT devices and their corresponding IoT malware in order to fingerprint flows originating from them. Additionally, it is known that IoT products manufactured by the same vendor possess a uniform, low-level architecture such as sharing a similar network card, operating system, etc., and happen to share the same TCP/IP stack information, including but not limited to TTL value and initial TCP window size, thus permitting the fingerprinting of IP addresses that Internet scanning services (i.e., Shodan) may have overlooked or could not identify.

To select a suitable and sound learning technique, we compare and contrast the performance of five models to permit the classification of compromised IoT devices in order to distinguish them from compromised, multi-purpose hosts. The first three are based on Convolutional Neural Network (CNN) deep learning models. Deep learning is an emerging branch of machine learning that use multiple layers of neural networks, backpropagation, and error correction to automatically learn features (i.e., representations) from a given data input. CNN is a state-of-the-art deep learning algorithm that uses dynamic kernels along a given data input to automatically extracts (i.e., pool) features. To this end, we asses a two-dimensional CNN (2D-CNN), a one-dimensional CNN (1D-CNN) (Collobert et al., 2011) and a multi-window one-dimensional CNN (MW-1D-CNN) (Cheng et al., 2016) in addition to two "shallow" learning methods rooted in Random Forest (RF) models.

In this context, an input sample consists of a matrix representation $\mathbf{X}$ of a flow with t packets and the number of extracted fields d from a packet is considered, yielding $\mathbf{X} \in \mathbb{R}^{t \times d}$. Namely, the i^{th} packet in a given flow is $\mathbf{x}_i \in \mathbb{R}^d$. Convolution operations are also defined by applying local kernels $\mathbf{w} \in \mathbb{R}^{h \times w}$ on the input to extract spatially local correlations in the data. In terms of the 2D-CNN model, it contains L number of consecutive two dimensional convolutional layers (with k kernels of size $w \times w$) and max pooling, followed by two dense hidden layers of sizes 64 and 32, respectively, and a Softmax classifier at the end (Fig. 3a). The 1D-CNN model has a similar architecture to the 2D-CNN, but instead, the convolution kernels have a fixed kernel width equal to the input sample width (i.e., $h \times d$) (Fig. 3b). Further, the MW-1D-CNN model mixes the outputs of various kernel heights h to capture the features. In turn, the output of the first layer of the proposed model is given by $c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$, where $\mathbf{x}_{i:i+h-1}$ defines the notation for a sequence of packets $\mathbf{x}_i, \mathbf{x}_{i+1}, .. \mathbf{x}_{i+h-1}$, b representing the bias, and f denoting the non-linear activation function. The filter is applied to each 2D sample instance to produce a feature map $\mathbf{c} = [c_1, \ldots, c_{t-h+1}]$. Subsequently, max pooling is applied over the feature map $\mathbf{c}$, taking the value $\max \mathbf{c}$. We used kernels $\mathbf{w}$ of different window heights h ($h = [2, 4, 6, \ldots, h_{\max}]$) to enable the capture of varying dynamics specific to darknet packet flows (Fig. 3c).

We also devise two RF models. The first was constructed based on raw packet features. The second operates on statistical features. We define feature statistics as the 5-tuple {*min*, 1-*quantile*, *median*, 3-*quantile*, max} of each field in flows of packets, which overall produces 85 features. These statistics can be considered as an estimation of the probability distribution function related to each field of the packet sequence in each flow. Please note that we make available the source code of the developed models, including their specificities from https://github.com/COYD-IoT/COYD-IoT/tree/master/IoT%20classifier%20models.

3.4. IoT Botnets: Features' extraction and campaign inference

Following the binary classification of IoT-generated scanning activities while filtering out non-IoT sources by employing the developed model, we conduct a thorough investigation on each individual flow's characteristics $\mathbf{Flow}_{IP}$. Such flows are comprised of at least 500 ($t \geq 500$) sequential packets, originating from a particular unsolicited IoT device. We proceed by extracting the corresponding feature set from aggregated flows $\mathbf{F}_{IP} = < \mathbf{Ports}_{IP}, \boldsymbol{\pi}_{IP}, \text{Flag}_{IP}, \text{ARR}_{IP} >$. $\mathbf{Ports}_{IP}$ is the grouping of the targeted transport protocols paired with their associated ports in ascending order (e.g., $\mathbf{Ports}_{IP_x} = \{\text{TCP:23, TCP:80, TCP:8080}\}$). In turn, $\boldsymbol{\pi}_{IP}$ is the corresponding discrete probability distribution function which represents the frequency of appearance of each of these ports within the given flow of packets (e.g., $\boldsymbol{\pi}_{IP_x} = [0.15, 0.70, 0.15]$). This is relevant since IoT devices typically possess a limited supply of resources. As a result, in the midst of conducting illicit scanning activities, they are often allocated to different ports and weighted based on the expected return. Flag_{IP} is Boolean, holding a value 1 if the IoT device conducting the scanning has the signature $tcpSeq == dstIP$ and 0 otherwise. This inference provides insights about a `Mirai`-like behavior, possibly indicating a variant or a code-reuse practice. Lastly, the Address Repetition Ratio, or ARR_{IP}, is the ratio of the total number of packets sent by a source IP address over the number of unique destination IP addresses, and is defined as $\text{ARR}_{IP} = \frac{|\mathbf{Flow}_{IP}|}{|\{dstIP | dstIP \in \mathbf{Flow}_{IP})\}|}$. Such scenarios as an ARR_{IP} greater than one are a consequence of the sending of multiple packets to a particular destination in order to compensate for packet loss and/or the probing of multiple ports at each destination. Note that, each instance of the same probing campaign will exhibit an equivalent ARR_{IP} due to the underlying IoT orchestrated probing machinery.

3.5. Minimum number of packets (t) for robust feature estimation

To derive an accurate estimation of the discrete probability distribution $\boldsymbol{\pi}$, we perform statistical analysis to compute a suitable t. By generating a lower bound on the number of packets, we can guarantee a minimum error of 5% within a confidence level of 0.5. Note that within the /8 network telescope, scan packets arrive with a random probability of 1/256, resembling the random sampling procedure. We consider the population of scan packets originated by a compromised IoT device, and adopt a simple random sampling mechanism, as shown in Eq. (8) (Cochran, 2007), to derive a lower bound on the sample size (equivalently, the number of received packets within the network telescope). The method herein is thus used to estimate the minimum sample size necessary to find the lower bound. By leveraging the requirements of a population proportion interval (Cochran, 2007), we perform the estimation at a $1 - \alpha$ confidence level, margin of error E and a planned proportion estimate p. By selecting more than n_0 samples, we assure that the probability that the actual error to be larger than E is not more than a small value α, i.e., $Pr(|p - P| \geq E) = \alpha$; where $z_{\alpha/2}$ is the $100(1 - \alpha/2)$ percentile of the standard normal distribution.

$$n_0 = \frac{z^2_{\alpha/2} p(1-p)}{E^2} \tag{8}$$

Since the product $p(1 - p)$ increases as p moves toward 0.5, a conservative estimation of the sample size is obtained by choosing $p = .5$, regardless of the actual estimated value of p.

[4] https://github.com/COYD-IoT/COYD-IoT/blob/master/Benign-Scanners.txt.

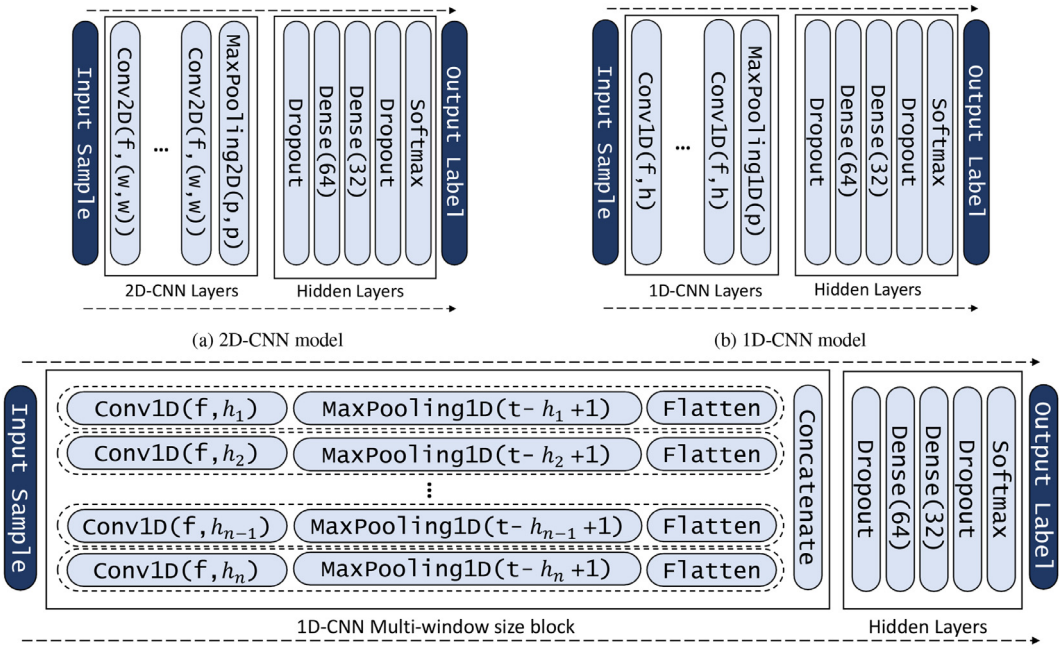


Fig. 3. CNN models for IoT/non-IoT binary classification.

Therefore, using a 0.5 planned portion estimate, the sample size needed to achieve a 5% margin of error at 95% confidence level is computed at 385. In this work, we select the number of packets equal to 500 (≥ 385) to significantly minimize the risk of errors in the extracted features, namely $\boldsymbol{\pi}$, to avoid subsequent issues in clustering and campaign detection.

3.6. Clustering mechanism

We hierarchically divide the IP addresses of the IoT scanners into separate groups $\boldsymbol{G}_i$ based on the given $\mathbf{Ports}_{IP}$, Flag_{IP} and ARR_{IP} of their feature set $\mathbf{F}_{IP}$. Upon completion, we cluster members of each group $\boldsymbol{G}_i$ to identify those scanning for the same set of ports but with a different probability distribution function $\boldsymbol{\pi}$. This enables us to leverage hierarchical agglomerative clustering (Xu and Tian, 2015), which determines the proximity matrix by calculating the distance between every pair of probability distribution functions $\{\boldsymbol{\pi}_{IP} | IP \in \boldsymbol{G}_i\}$ based upon the Jensen-Shannon Divergence (JSD) (Cha, 2007) distance metric. JSD, defined in (9), estimates the distance between two discrete distribution functions, and is the symmetrized version of the well-known Kullback-Leibler Divergence (KLD).

$$JSD(\boldsymbol{\pi}_i || \boldsymbol{\pi}_j) = \frac{1}{2} KLD(\boldsymbol{\pi}_i || \boldsymbol{M}) + \frac{1}{2} KLD(\boldsymbol{\pi}_j || \boldsymbol{M}), \tag{9}$$

where $\boldsymbol{M} = \frac{1}{2}(\boldsymbol{\pi}_i + \boldsymbol{\pi}_j)$. and $KLD(\boldsymbol{P}||\boldsymbol{Q}) = -\sum_i \boldsymbol{P}(i) \log(\frac{\boldsymbol{Q}(i)}{\boldsymbol{P}(i)})$ for discrete PDF $\boldsymbol{P}$ and $\boldsymbol{Q}$.

We select hierarchical agglomerative clustering due to the fact that based on statistical analysis, the estimated $\boldsymbol{\pi}$ values are within the specific distance from the actual distribution (cluster centers) with high confidence. Therefore, it can correctly identify centers by merging close samples and executing a bottom-up approach. In addition, other clustering methods (such as *k*-means) assume equal cluster sizes which is not correct in the context of IoT campaigns while density-based techniques (such as DBSCAN (Khan et al., 2014)) are only suitable when the density of the data is non-uniform and the clusters can be shaped arbitrarily. As noted, hierarchical agglomerative clustering operates in a bottom-up fashion. Each observation forms its own cluster and begins moving up the distance-based hierarchy, subsequently merging with the clusters. To designate appropriate consolidation, we use a distance threshold (0.05) in which merging only occurs if the distance between the two given cluster centers falls beneath.

4. Empirical evaluation

The evaluation was executed using 3.6 TB of darknet traffic that was collected throughout a 24-h period on December 13th, 2018. This data is provided by the Center for Applied Internet Data Analysis (CAIDA) /8 network telescope. While this specific dataset *per se* is subject to MOUs and thus cannot be shared as is, interested readers can request access to CAIDA's real-time darknet data through DHS IMPACT (Policy and Trust, 2019). Additionally, while we make available a sample collected at another /13 darknet IP space available through the `GitHub` repository for experimentation purposes, the developed and open-source methods are generic enough to be applied on any darknet data within any desired time frame.

4.1. Results of the darknet sanitization model

By executing the proposed model of Section 3.1, the distribution of malicious and misconfiguration traffic with respect to the number of packets was found to be 88.21% and 11.79%, while the distribution of source IP addresses was 26.17% and 73.83%, respectively. Validation of such outcome revealed that close to 90% of the misconfiguration traffic defines packets that hit the /8 network telescope only once, while the remaining appeared to be malformed packets. Further, it can be observed that even though misconfiguration traffic is relatively low (11.79%), it is responsible for a large proportion of the source IP addresses (73.83%). These findings shed more light on the problematic nature of misconfiguration traffic with regards to Internet measurements via network telescopes and emphasize the effectiveness of the proposed pre-processing model (Table 2).

Table 2
Distribution of malicious and misconfiguration traffic in the /8 network telescope dataset.

	Malicious	Misconfiguration
Traffic	88.21%	11.79%
Sources	26.17%	73.83%

In terms of runtime, the implementation heavily relied on the Linux-derived libpcap C++ library while running on an Ubuntu 18.04 system. Testing our model on a machine with a quad core Intel i7-8550 at 1.80GHz processor and 16GB of RAM, the developed approach processed 8GB files containing close to 67.5 million packets with an average 636 second execution time, consuming close to 11.6GB of RAM. We believe runtime can be considerably improved by using SSD storage (since most of the delay was I/O related) and adopting multithreading.

4.2. Results of dataset preparation

Regarding the data collection and dataset preparation steps of Section 3.2, and by immediately scanning back about 1.7M Internet scanners inferred through the network telescope, about 25.84% of them were found to have at least one open port. Further, amongst the total 543,392 gathered banners, the majority were HTTP (54.11%), FTP (11.10%), SSL Certificate (10.50%), TELNET (10.19%), RTSP (7.00%), and CWMP (2.60%). We were able to distinguish between 45,184 IoT and 7763 non-IoT devices to generate the training dataset. At this juncture, the label and corresponding metadata were incorporated into $t \times d$ training and test data matrices of IoT and non-IoT devices. We shuffled the training dataset and then performed normalization by way of the Min-Max method (García et al., 2015). Subsequently, we computed and removed the mean. To evaluate the proposed model, we trained it using a prepared dataset captured in November 2018 and then tested it using our dataset from December 2018. The one month gap between the training and test datasets ensured that there exists no correlation between them for sound evaluation. The test dataset consisted of 34,974 IoT and 7193 non-IoT sources.

4.3. Evaluating the IoT classification models

The proposed CNN models were implemented in Keras (Chollet et al., 2015). To address the problem of class imbalance within the training dataset, cost-sensitive learning was applied (Thai-Nghe et al., 2010). The number of epochs was found to be 30 to avoid over-fitting. Further, we performed a search on subspaces of hyper-parameters as presented in Table 3, leveraging Tree-structured Parzen Estimator (TPE) (Bergstra et al., 2011) in Hyperas (Pumperla, 2019), and selected the best model (out of 100 trials) with regards to the loss. RF models were implemented and trained using the scikit-learn (Pedregosa et al., 2011) package. The best model was retrieved based upon random search (using the RandomizedSearchCV method) in the search space as summarized in Table 4. In Tables 3 and 4, parameter ranges are reported with `begin:step:end` format. For evaluating the CNN models, we leverage an NVIDIA GeForce RTX 2070 GPU with 8GB of memory, 2304 CUDA cores and 288 Tensor cores to accommodate for parallelization.

To compare the performance of the different models, we rely on standard machine learning metrics such as precision, recall, F-measure and AUC-ROC for the IoT class. Precision is the ratio of correctly classified IoT devices over all the instances that have been designated as IoT using the proposed model ($precision = \frac{tp}{tp+fp}$). Recall is the ratio of correctly classified IoT devices over the total number that is actually existing within the test data ($recall = \frac{tp}{tp+fn}$). Recall demonstrates the model's ability to find all relevant cases within a given dataset, whereas precision gives the model's ability to designate only the actual relevant cases as relevant. In order to bring these two metrics together, often F-measure is employed which takes the weighted average of precision and recall, i.e., the harmonic mean ($\boldsymbol{F} - measure = 2.\frac{precision.recall}{precision+recall}$). The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a threshold-independent performance measurement for classification. It measures the entire two-dimensional area underneath the ROC curve (i.e., true positive rate vs false positive rate at all classification thresholds) from (0,0) to (1,1).

We report the results in Figs. 4 and 5. We can note that the AUC-ROC score for the RF model trained on quantiles is slightly higher than that of the other models. Further, both of the figures reveal that the CNN-based models result in higher recall and lower precision scores in contrast to the RF models. The outcome also shows that the multi-window 1D-CNN (MW-1D-CNN) outperforms the 1D-CNN and the 2D-CNN; this is quite expected, since packet fields (unlike image pixels) lack temporal or spatial relationships with one another. Therefore, moving the kernels over the horizontal dimension would not lead to better learning. Furthermore, the multi-window 1D-CNN can capture varying dynamics given that only a portion of the scan packets actually hit the /8 darknet.

4.3.1. Feature importance

To shed light on which features were most decisive in the learning process, and given that the RF models performed the highest, we illustrate the features' scores (derived from the RF model on quantiles) in Fig. 6. As expected, the distribution of destination ports which typically reveals the scans' intentions plays the most noteworthy role for fingerprinting IoT devices. This is closely followed by other fields such as the total packet length and the total header length, in addition to the TCP/IP stack and OS-related fields including the TCP window size, option fields and the TTL.

4.3.2. Effect of number of packets ($\boldsymbol{t}$) on the classifiers

Figs. 7 and 8 illustrate the impact of the number of packets within the input sample $\mathbf{X} \in \mathbb{R}^{t \times d}$ on the AUC-ROC and processing time (loading and training data). To quantify the effect for each value of t, we execute the training process 10 times using parameters taken from the best models, retrieved from Tables 3 and 4. Although it is expected that increasing the number of packets will increase the total amount of information to be processed, subsequently increasing a model's performance, it is not consistently proven. Reviewing the results in Fig. 7, when a RF model is trained using raw features, adding an increasing amount of packet data will eventually confuse the model, lowering the AUC-ROC of the RF model. In contrast, when a RF model is trained on quantiles, increasing the amount of input packet data actually lead to an improvement in the AUC-ROC of the model, with diminishing returns. In addition, it can be seen that changing t has no significant trending effect on the AUC-ROC of CNN-based models.

Fig. 8 reveals that an increased sample size, containing a larger number of packets, will generally increase the processing time. Most evidently perceived in the MW-1D-CNN model, its high complexity leads to a significant increase in processing time as the sample size is increased. However, an increased sample size leads to a slight decrease in processing time for a quantile-trained RF model. Ultimately, the results depict that maximum AUC is achieved through training an RF model with $t = 90$. Furthermore, the non-RF models have an acceptable performance and AUC value at $t = 90$. Therefore, to facilitate future implementations and experimentation, $t = 90$ is found to be a suitable choice for efficient and accurate classification.

Table 3
Tuned hyperparameters of the selected CNN models.

Parameters	Space	2D-CNN	1D-CNN	MW-1D-CNN
Optimizer	SGD, Adam, RMSProp	RMSProp	RMSProp	RMSProp
Num. of kernels (k)	32,64,128	32	128	64
Kernel size ($w \times w$)	(2,2),(3,3)	(2,2)	-	-
Kernel height (h)	2,4,8,16,32,64	-	64	-
Max kernel height ($h_{\max}$)	40:10:80	-	-	80
Pool size (p)	2,3	2	3	-
Batch size	128, 256	128	256	256
Activations	Relu, Sigmoid, Tanh	Sigmoid	Tanh	Sigmoid
Dropout	$U(0.1, 0.3)$	0.195	0.296	0.298
learning rate	0.001	0.001	0.001	0.001
Num. CNN layers (L)	1:1:4	4	3	-

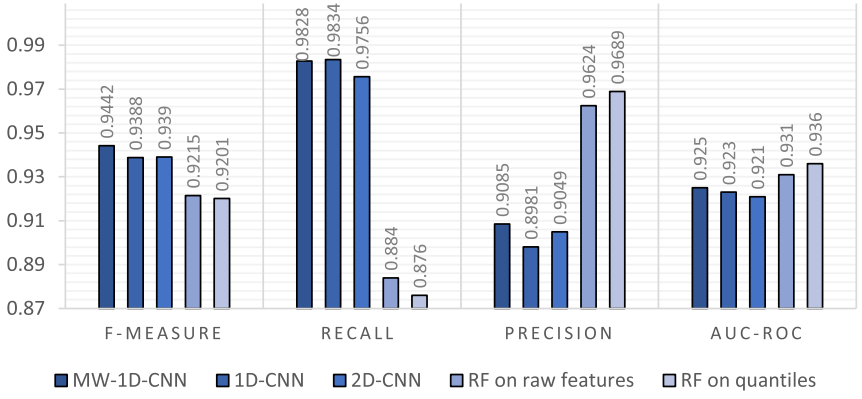


Fig. 4. Performance metrics of the devised models.

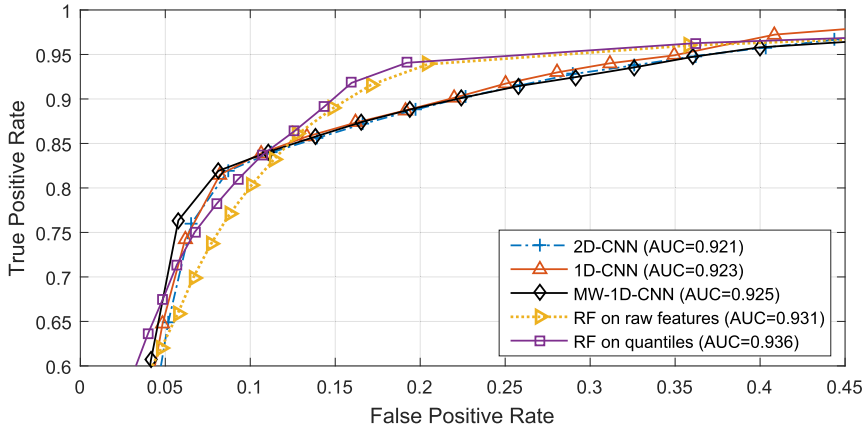


Fig. 5. AUC-ROC curves to evaluate the devised models.

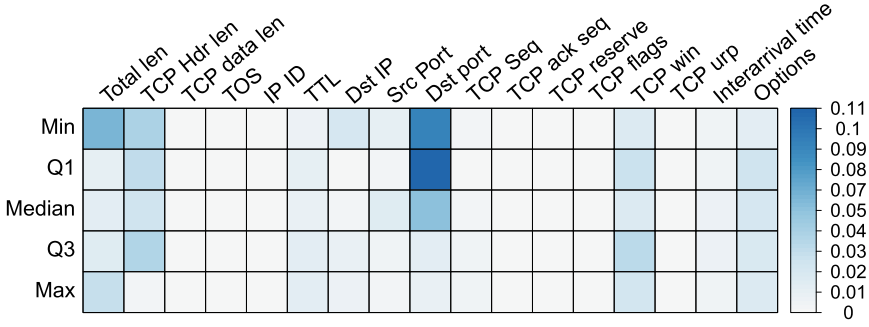


Fig. 6. Ranking of features' importance.

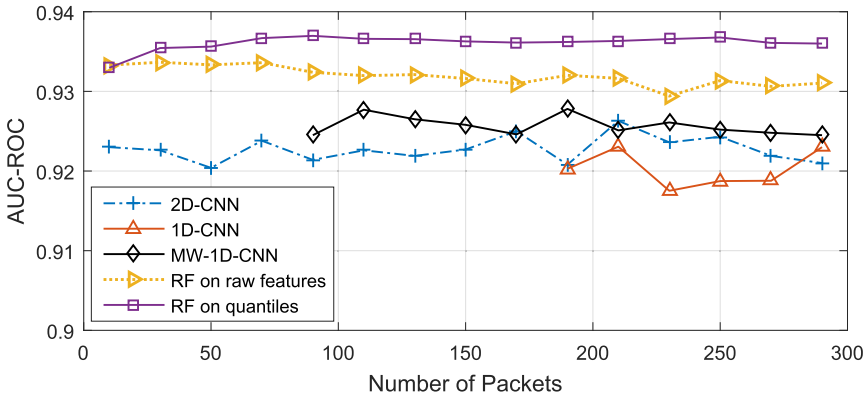


Fig. 7. Effect of packet number (t) on AUC-ROC.

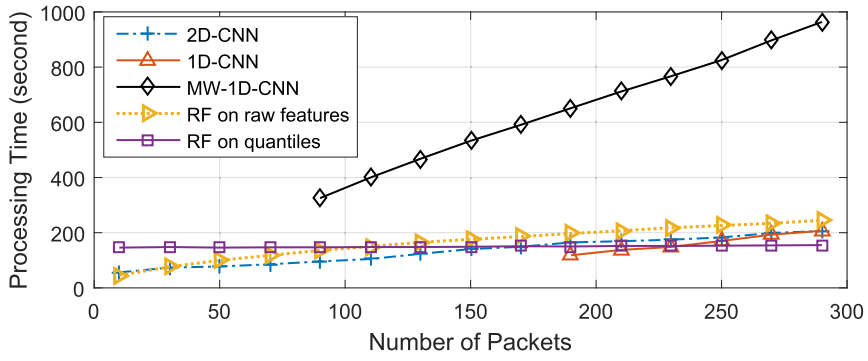


Fig. 8. Effect of packet number (t) on processing time.

Table 4
Tuned hyperparameters of the RF models.

Parameters	Space	RF on raw fields	RF on Quantiles
Num. estimators	20:20:100	60	60
Max depth	4:4:20	12	12
Min samples leaf	2:10:102	52	52
Min samples split	$U(2, 10)$	6	4
Bootstrap	True, False	False	False
Criterion	Gini, Entropy	Entropy	Gini

Table 5
Top countries hosting infected IoT devices.

Country	(%)	Country	(%)
Brazil	41.93	Greece	1.70
Iran	10.17	Italy	1.60
China	5.14	United States	1.42
Russian	3.59	Indonesia	1.25
Egypt	3.36	Mexico	1.24
India	2.47	Ukraine	1.21
Turkey	2.32	Korea (south)	1.07
Taiwan	2.13	United Kingdom	0.83
Vietnam	1.91	Thailand	0.72
Argentina	1.83	Spain	0.66

4.4. Inferring and characterizing compromised IoT devices and campaigns

Given the aforementioned classification results, we selected the MW-1D-CNN model since it provided the highest true positive rate while limiting the false positive rate to around 0.08 (Fig. 5). We further re-trained the model on recent data from December 2018 to accompany for any evolving dynamics.

By applying the binary classifier on 24 hours of darknet data of December 13th, it was capable of fingerprinting 441,766 out of the 1,787,718 unique scanners to be originating from compromised IoT devices. Although previous works solely considered those with a `Mirai` signature as IoT-related (Antonakakis et al., 2017), we inferred that in fact, they make up less than 25% of the IoT scanner population that the proposed model was able to uncover, leaving a whopping 75% to go about their malicious activities without any semblance of an adequate attribution.

Table 5 summarizes the location of these exploited devices, where Brazil (41.93%) was found to be hosting a significant portion, followed by Iran (10.17%), China (5.14%), Russia (3.59%), Egypt (3.36%), India (2.47%) and Turkey (2.32%).

Furthermore, the top three ISPs hosting the largest number of compromised IoT devices were Vivo (134,021), TE Data (11,804) and Iran Telecom Co. (9912).

While the extensive presence of IoT scanners alone gives pause for concern, a relatively significant proportion residing within the telecommunication and ISP sectors is rather expected; conversely, their existence within sectors including but not limited to critical sectors is quite alarming. In Table 6, critical sectors which appear in lists provided by the U.S. Department of Homeland Security (DHS) and the European Union (EU) are highlighted (Husák et al., 2018). Amongst the inferred instances, quite a few were found to be located within that of medical infrastructures (87), government

Table 6
Top Sectors hosting infected IoT devices.

Sector	Count
Telecommunications	175,642
Internet Service Provider	82,238
Private Service	1,319
Internet Hosting Services	780
Education	485
Internet Colocation Services	314
Data Services	99
Health	87
Government	86
Manufacturing	74
Finance	57
Lodging	44
Professional Service	38
Transportation	29

entities (86), manufacturing realms (99), and commercial businesses (38).

Along those lines, the lengthy list of 50 identified vendors reveals a broad range of manufactures and device types that IoT botnets demonstrate preference for exploitation. Amongst them, `MikroTik` (14,090), `Aposonic` (2,222), `Huawei` (732), `Foscam` (594) and `Hikvision` (417) are the topmost five targeted by the tagged compromised devices. Routers (53.64%) and IP Camera/DVR (28.93%) continue to be the most frequently infected devices.

Moreover, the most commonly targeted ports based upon the number of scanning packets generated by the compromised IoT devices are reported in Table 7. The top targeted ports include 23 (41.9%), 80 (23.9%), 8080 (19.7%), 5555 (4.9%), 81 (3.2%), 2323 (1.7%) and 22 (1.3%). Intriguingly, we identified the presence of non-IoT targeted ports such as 2480 (OrientDB), 5984 (CouchDB), 3389 (RDP), 7001 (Oracle), 5900 (VNC) and 2004 (Drupal), as well as that of uncommonly used IoT ports 32,764 (router backdoor), 37,215 (UPnP in SOHO routers) and 52,869 (UPnP in wireless chipsets). Set {23,80,8080} is the most prevalent target port combination; 54% of devices actually only scan this port combination.

4.4.1. Inferring and reporting on orchestrated IoT botnets

Among the 441,766 IoT scanners that were detected on Dec. 13th, 2018, based on the results in Section 3.5, those that sent less than 500 packets were filtered out to exclude any of those that can degrade the estimation of the probability distribution function $\boldsymbol{\pi}$.

Subsequently, the respective features were extracted and the clustering method described in Section 3.4 was executed. In roughly 40,000 scan flows, we witnessed less than 0.01% of packets in each scan flow arriving at specific UDP ports (e.g., 5998, 43922, 48715, 31,869 etc.). After analyzing such occurrences, we deduced they resulted from associated bugs or attacks on P2P networks such as BitTorrent; an observation that is also consistent with previous works (Benson, 2016). As a result, in order to avoid the ill-effects of uncorrelated incidents, the identified packets were removed prior to clustering. Regarding the inferred campaigns, the proposed approach detected over 350 orchestrated IoT botnets. Since the size of each IoT probing campaign translates to its given severity, we summarize those botnets possessing more than 300 coordinated IoT bots in Table 8. Interestingly, in solely considering IoT scanners that targeted the set of ports {23, 80, 8080}, we detected 30 distinct botnets with differing distributions, Flag (i.e., `Mirai`-like signature/behavior), and ARR.

4.4.2. Packet inter-arrival time analysis generated from the inferred botnets.

Following the investigation of scan-based behaviors of the inferred compromised IoT devices, we deduced two separate classes of unique scan traits. Class 1 includes devices that present periodic behavior in the time series of their packet Inter-Arrival Time (IAT). For rate limiting purposes (Ceron et al., 2019), such devices seem to generate a fixed number of packets then wait exactly 1 second to re-confirm their desired scanning rate (in packets per second (pps)). This leads to high peaks in their histograms of packet IATs as seen in Fig. 9a. In contrast, the members of the Class 2 do not portray any related periodic behavior when analyzing their packet IAT. Fig. 9b portrays the IAT of the packets generated by the IoT devices of Class 2, demonstrating an exponential distribution. To detect the aforementioned behaviors, we first calculate the histogram of packet IATs and then identify the peaks with an auto-correlation coefficient (Figs. 9c and 9 d). To reveal the population of such inferred classes in the context of the identified probing campaigns, Fig. 10 illustrates the fraction of scanning classes in each campaign.

4.4.3. Scan rate analysis of the inferred IoT botnets

Fig. 11 presents the distribution of scanning rates extracted from the inferred IoT botnets, as perceived by the network telescope. Campaigns in which their scan rates follow a normal distribution with a single peak and a narrow width (such as #1, #2,

Table 7
TCP port distribution determined by quantifying the number of compromised IoT scan packets received by each included port. Grey cells highlight unconventional, rarely probed ports and services.

Port	Service	(%)	Port	Service	(%)	Port	Service	(%)
23	Telnet	41.912	8181	HTTP-alt	0.114	83	HTTP-alt	0.028
80	HTTP	23.917	88	HTTP-alt	0.057	443	HTTPS	0.025
8080	HTTP-alt	19.784	21	FTP	0.056	3389	RDP	0.023
5555	ADB	4.995	7547	TR-064	0.053	8090	HTTP-alt	0.018
81	HTTP-alt	3.288	8081	HTTP-alt	0.050	8089	HTTP-alt	0.018
2323	Telnet-alt	1.705	8888	HTTP-alt	0.047	139	SMB	0.006
22	SSH	1.391	37215	UPnP	0.045	7001	WebLogic	0.005
9000	MCTP	0.470	2480	OrientDB	0.041	52869	UPnP	0.005
445	SMB	0.315	5984	CouchDB	0.040	8291	Winbox	0.004
5358	Telnet	0.238	82	HTTP-alt	0.029	1433	MS-SQL	0.004
8000	HTTP-alt	0.197	8001	HTTP-alt	0.029	5900	VNC	0.003
2222	SSH	0.165	8088	HTTP-alt	0.028	2004	Drupal	0.003
8443	HTTP	0.121	84	HTTP-alt	0.028	1900	UPnP	0.002
32764	Linksys Vuln.	0.117	85	HTTP-alt	0.028	Other	-	0.596

Table 8
Orchestrated IoT botnets in the wild.

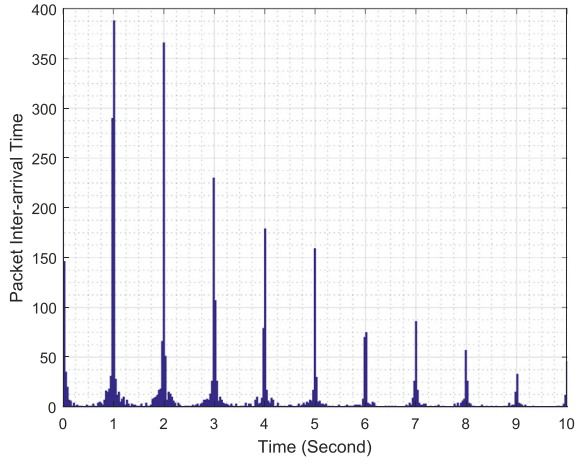
Id	Ports	Flag	ARR	#Bots	π	Crypto Miners (Coinhive, xmrMiner)	Compromised Devices
1	23, 80, 8080		1	139,858	[0.33 0.33 0.34]		MikroTik, Hikvision, Foscam, Vivotek, Huawei, Aposonic, Intelbras, Ubiquiti, Netgear, Mitrastar, Askey, Archer
2	23, 80, 8080		1	55,139	[0.294 0.295 0.411]		MikroTik, Hikvision, Intelbras, TP-LINK, D-Link, Huawei, ZTE, Foscam, QNAP, ZyXEL, Cisco, SERCOMM, Vivotek
3	23, 2323	✓	1	36,464	[0.9 0.1]		Huawei, Aposonic, Foscam, Hikvision, Mikrotik, Cisco, TP-LINK, CIG Shanghai, ZTE, Ubiquiti
4	80	✓	1	12,895	[1.]		Huawei, Hikvision, MikroTik, AvTech, ZTE, Foscam, Cisco, Ubiquiti, NUUO
5	5555	✓	1	11,050	[1.]		Huawei, TP-Link, Hikvision, Aposonic, Foscam, MikroTik, Sagemcom, iGate, VNPT, Trendchip
6	23, 81		1	9805	[0.495 0.505]		Aposonic, Foscam, Huawei, Hikvision, ZTE, Lilin, Sagemcom, Netgear
7	23, 80, 8080		2	7610	[0.171 0.650 0.179]		MikroTik, TP-LINK, Hikvision, AvTech, Foscam, D-Link
8	23	✓	1	7200	[1.]		Huawei, Hikvision, TP-Link, AvTech, TP-LINK, Aposonic, ZEM800, ZTE
9	23, 80, 8080		1	5971	[0.242 0.244 0.514]		MikroTik, ZTE, Hikvision, TP-LINK, Foscam
10	23		3	5491	[1.]		DZS, Foscam, MikroTik, Synology, ZyXEL, Hikvision
11	80, 8080		1	5162	[0.492 0.508]		MikroTik, Foscam, Hikvision, Huawei, TP-LINK, Ubiquiti
12	23		1	4689	[1.]		D-LINK, Hikvision, Aposonic, MikroTik, Broadcom, ASUS, AVM, Netgear
13	23, 80, 8080		1	4468	[0.442 0.032 0.526]		MikroTik, TP-LINK, Hikvision, D-Link
14	23		4	3911	[1.]		GPON (DZS), Hikvision, Huawei, MikroTik, Dasan, Foscam, Mercusys
15	22, 2222	✓	1	3783	[0.897 0.103]		QNAP, Huawei, Hikvision, ASUS, Foscam, SERCOMM, MikroTik, Intelbras, Ubiquiti
16	23, 2323, 5555	✓	1	3545	[0.249 0.032 0.719]		ZyXEL, MikroTik, Avtech, Broadcom, Foscam, TP-LINK, Hikvision, D-Link
17	23, 2323	✓	1	2727	[0.967 0.033]		Huawei, ZTE, Hikvision, MikroTik, Aposonic, ZEM800, Foscam
18	23		2	2146	[1.]		TP-LINK, Hikvision
19	23, 32764, 80, 8000, 8080, 8081, 8089, 8090, 81, 8181, 8443, 8888, 9000	✓	1	2140	[0.034 0.122 0.153 0.02 0.154 0.02 0.019 0.02 0.068 0.123 0.122 0.022 0.121]		NUUO, Foscam, Hikvision, Huawei, AVM, MikroTik, Aposonic
20	23, 8080		1	1591	[0.48 0.52]		MikroTik, Hikvision, D-Link, TP-LINK
21	23, 80, 8080		1	1286	[0.384 0.319 0.298]		MikroTik, SERCOMM, Foscam
22	80, 8080+rnd		1	1247	[0.45 0.45]		MikroTik
23	23, 81		1	1191	[0.095 0.905]		Aposonic, Foscam, Hikvision
24	23, 80, 8080		1	1083	[0.226 0.5 0.274]		MikroTik, D-Link, Foscam, Aposonic
25	23, 5358		1	1059	[0.5 0.5]		Hikvision, Foscam, Intelbras
26	23, 2480, 5555, 5984, 80, 8080+rnd	✓	1	783	[0.126 0.120 0.134 0.121 0.128 0.121]		Foscam, Huawei, Aposonic, Hikvision
27	80, 8080		3	756	[0.814 0.186]		MikroTik, TP-LINK
28	443, 80, 8000, 8001, 8080, 8081, 8088, 81, 82, 83, 84, 85, 88, 8888	✓	1	723	[0.071 0.071 0.071 0.071 0.071 0.071 0.071 0.072 0.072 0.072 0.071 0.072 0.071 0.072]		Synology, Hikvision

(*continued on next page*)

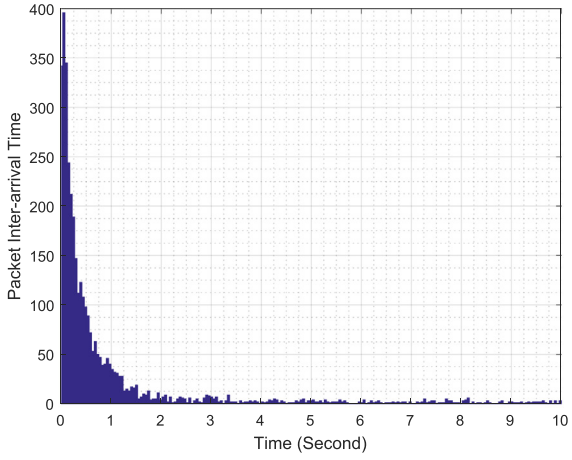
Table 8 (*continued*)

29	23, 2323	✓	1	691	[0.794 0.206]		Huawei, Aposonic, Hikvision
30	23, 9000		1	677	[0.49 0.51]		Sagemcom, SERCOMM, Hikvision, Cisco, Huawei, Aposonic, AVM, Mikrotik, Foscam
31	23		5	642	[1.]		ZTE, Hikvision, Foscam, MikroTik, Netgear
32	80		1	616	[1.]		MikroTik, Hikvision, Huawei
33	23, 80, 8080		1	544	[0.15 0.3 0.55]		MikroTik, Huawei, ZTE, Hikvision, Vivotek, Foscam, Aposonic
34	23, 81		1	541	[0.291 0.709]		Aposonic, Huawei, Hikvision, Foscam, TP-LINK
35	23, 445, 80, 8080		1	376	[0.3142 0.0587 0.3155 0.3115]		MikroTik, Aposonic
36	23, 7547, 80, 8080, 8291		1	340	[0.334 0.002 0.33 0.331 0.002]		MikroTik, Hikvision

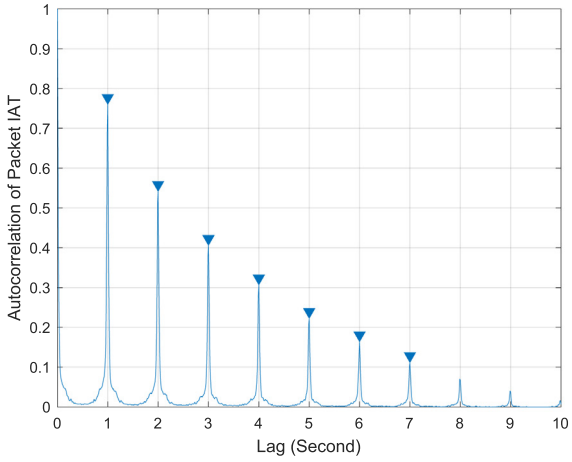
#6, #11, #14, #15, #18 and #20 of Table 8) were found to be exhibiting strong rate limiting policies regardless of their identified scanning class. In contrast, the rates of botnets #3, #8 and #17 are distributed over a wider range, showing no artificial rate limiting behaviors. In fact, this inference matches the released source code of the `Mirai` malware (which botnet #3 is attributed to), demonstrating the lack of rate limiting practices. For classes that were discovered to have no artificial rate limiting usages, each individually-compromised device sent a maximized number of scan packets based on their processing power and throughput. By focusing on the scanning rates' distributions of the inferred campaigns coupled with their population of scanning classes (Fig. 10), we can deduce some facts about the purity of the clustered campaigns. This enable further scrutiny of such campaigns to



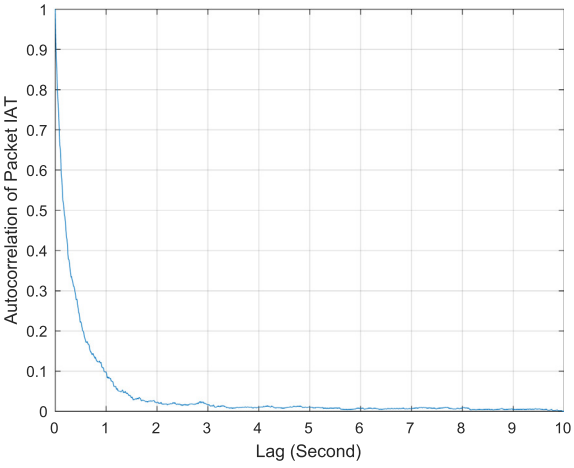
(a) Sample histogram of Class 1. Originated from an IoT device in Campaign 6



(b) Sample histogram of Class 2. Originated from an IoT device in Campaign 4



(c) Autocorrelation of (a) followed by peak detection.



(d) Autocorrelation of (b) followed by peak detection.

Fig. 9. Sample of packet IAT's histograms for Class 1 (a) and Class 2 (b) scanning practices. (c) and (d) respectively show the auto-correlation of (a) and (b) and the detected peaks.

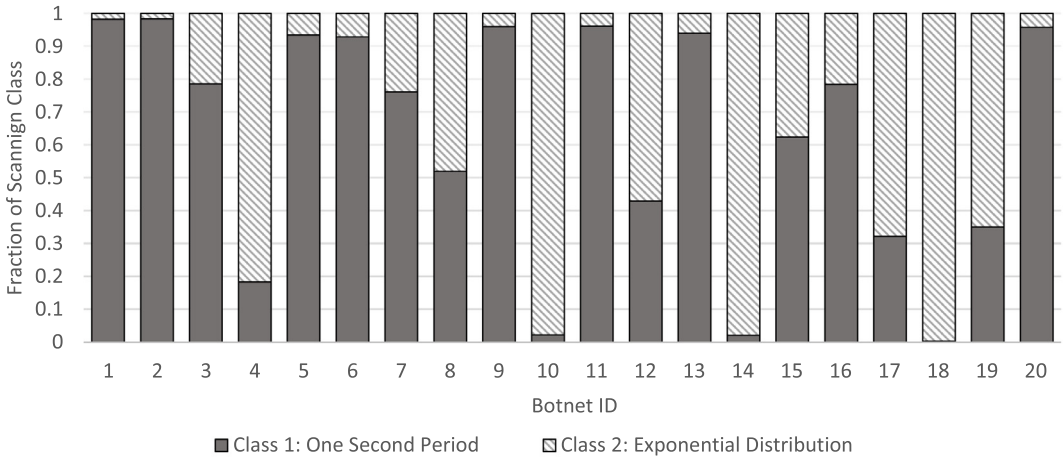


Fig. 10. Fraction of the derived scanning practices within the top populated campaigns (as detailed in Table 8).

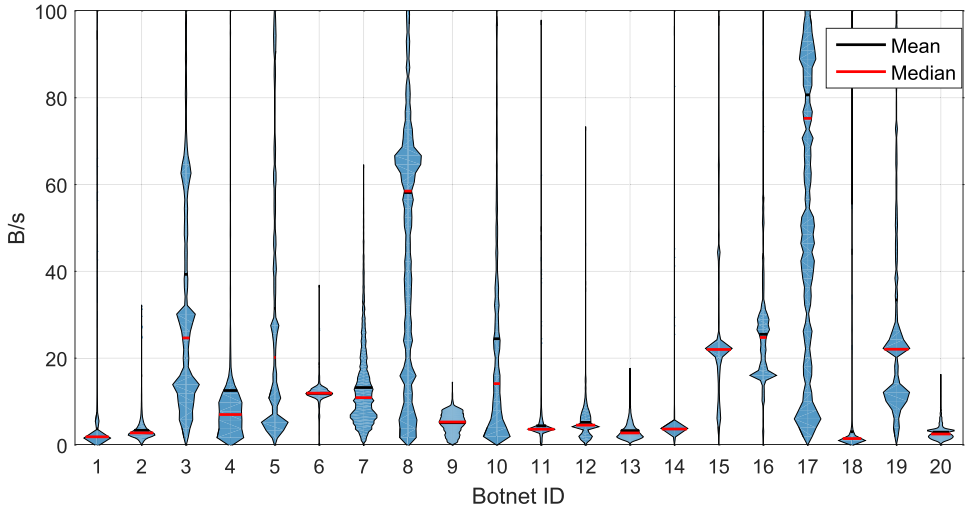


Fig. 11. A Violin illustration of the top 20 populated campaigns' scanning rated as observed on the /8 network telescope. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

determine if they contain a singular or multiple botnets. Note that IAT-related features were not included during the clustering mechanism of Section 3.4. With this in mind, we further examined the identified campaigns looking for Class 1 scan traits (those employing rate limiting practices) while their rates' distributions showing negative outcomes related to following a single normal distribution. The outcome, for instance, showed that although botnet #5 was identified as possessing Class 1 scan traits, its distribution of scanning rate shows 3 distinct peaks. Therefore, we postulate that there may be three unique botnets that were misidentified within a single cluster. Nevertheless, this is expected for campaigns with a single target port (and eventually $\boldsymbol{\pi} = [1]$), similar to campaign #5.

4.4.4. Geo-distributions of the inferred IoT botnets

By examining the geo-distribution characteristics of the identified botnets, we observe significant differences. Fig. 12 depicts the geo-distribution of the most populated campaigns generating scan events from multiple continents. Indeed, geo-distribution characteristics are likely a direct result of the popularity differences related to the adopted device types and manufacturers, which is known to be unique to each region (Kumar et al., 2019). Since botnets leverage definitive attack vectors, they are typically customized to target specific vendors; coherently, the vendor's popularity will also attract botnets, which is reflected in the concentration of such popular devices/bots in certain geo-locations. For further analysis, Fig. 13 displays the cumulative distribution of campaigns over Autonomous Systems (AS). Despite a few cases with highly similar distributions (e.g., #2 and #9), other specially less populated botnets are discovered to have a larger difference between distributions, as shown in Fig. 13b. Furthermore, amidst the inferred campaigns, there exists campaigns whose geo-distributions do not comply with that of the global distribution of infections (Table 5). For instance, with respect to campaigns #7, #13, #24 and #27, over 98% of infected IoT devices are located in Iran. Campaign #30 has upwards of 50% and 10% of compromised IoT devices located in USA and UK, respectively. Further, campaign #28 shows a 40% infection rate in North America, and a 21% infection rate in Europe. Spread across multiple geographic regions, these campaigns contradict the global distribution of infections.

4.4.5. IoT botnets with cryptojacking capability

Aside from the dominant monetization method for IoT botnets performing DDoS attacks, cryptojacking has emerged as a critical IoT botnet capability (ISTR, 2018; Tuttle, 2018). In essence, compromised routers have become responsible for injecting JavaScript crypto-currency miners into the HTTP pages requested by devices on their network (Zimba et al., 2019). JavaScript miners such

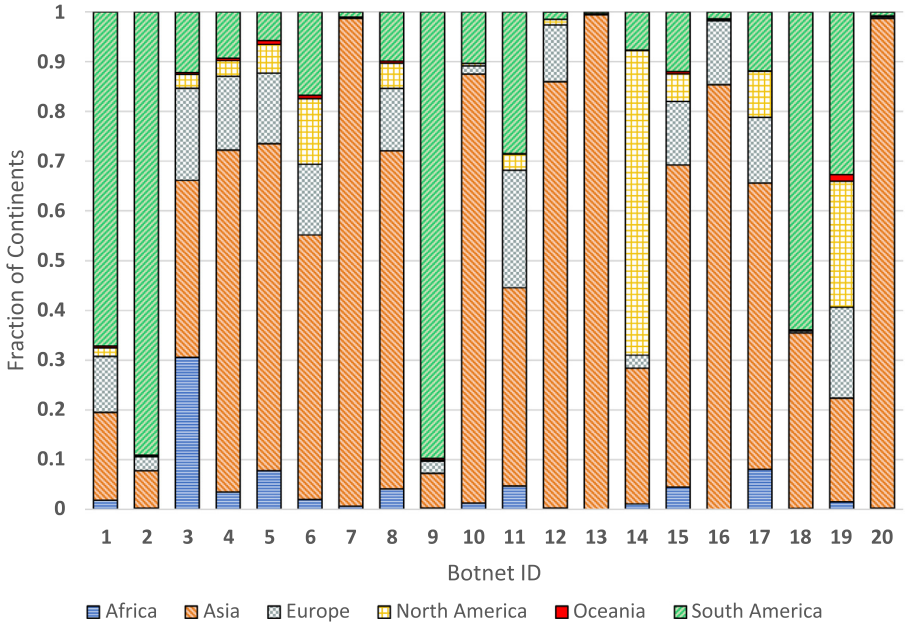
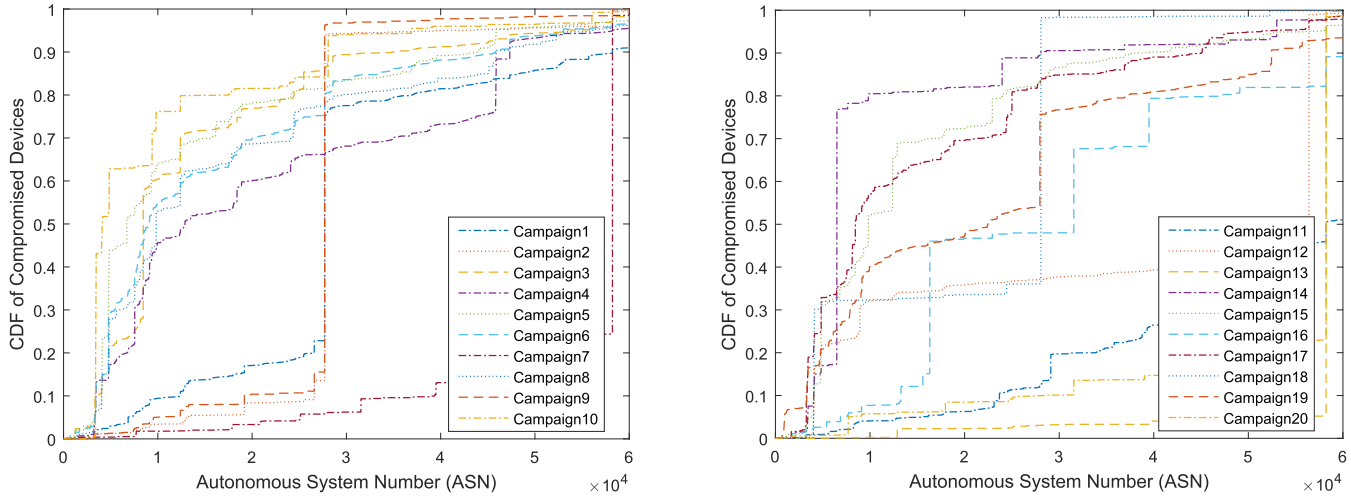


Fig. 12. Geo-distribution of the top 20 populated campaigns over different continents.



(a) Campaigns 1-10 (b) Campaigns 11-20

Fig. 13. Cumulative distribution of the top 20 populated campaigns (of Table 8) over different Autonomous Systems.

as `Coinhive` (Coinhive, 2018) and `xmrMiner` (xmrMiner, 2019) strive for Monero altcoin in particular. To this end, we examined the responses to HTTP requests derived from the IoT scanners, tagging those that contain the `xmrMiner` or `Coinhive` JavaScript modules, and exporting their corresponding keys. By doing so, we discovered 1134 `xmrMiner` and 923 `Coinhive` instances with 23 and 30 distinct keys, respectively. The campaigns designated as containing members with cryptojacking capabilities are highlighted in Table 8. The relation of crypto-mining keys appearing in each campaign is illustrated using a 3-partite graph as in Fig. 14. We analyzed the graph to find the number of components to uncover any further relations between the campaigns. Interestingly, the graph is connected. This demonstrates that salient connections exist between all the campaigns involved in cryptojacking activities.

In addition, we uncovered large campaigns maintaining cryptominer instances with and without the presence of `Mirai`-like signatures. Moreover, 943 out of 1134 devices, belonging to a total of 18 campaigns (#1, #2, #4, #5, #7, #8, #11, #13, #15, #16, #19, #20 #21, #24, #32, #33, #35 and #36), share the same xmrMiner-related key "4983e34ef01b4b579725b3a228e59e79" (red edges in Fig. 14). In other words, large portions of significant IoT campaigns could be reported to be attributed to the same "player". Additionally, upon exploring the key within `Censys`, 54,743 `Mikrotiks` were shown to possess it. In total, these campaigns equate to approximately 250,000 compromised IoT devices, or 54% of all the identified compromised devices.

4.4.6. A closer look at other campaigns of interest.

Campaign #3 with 36,464 bots was inferred to be targeting ports 23 and 2323 with a proportion of 9:1, which is the same as instructed within the `Mirai` released code. Another interesting observation pertains to botnet #26 (of Table 8) where packets to random TCP and UDP ports were sent in addition to targeting the defined set of ports of {23, 2480, 5555, 5984, 80, 8080}. Additionally, this campaign targeted port 2480 (OrientDB) and 5984 (CouchDB), as well as other common IoT-related ports including

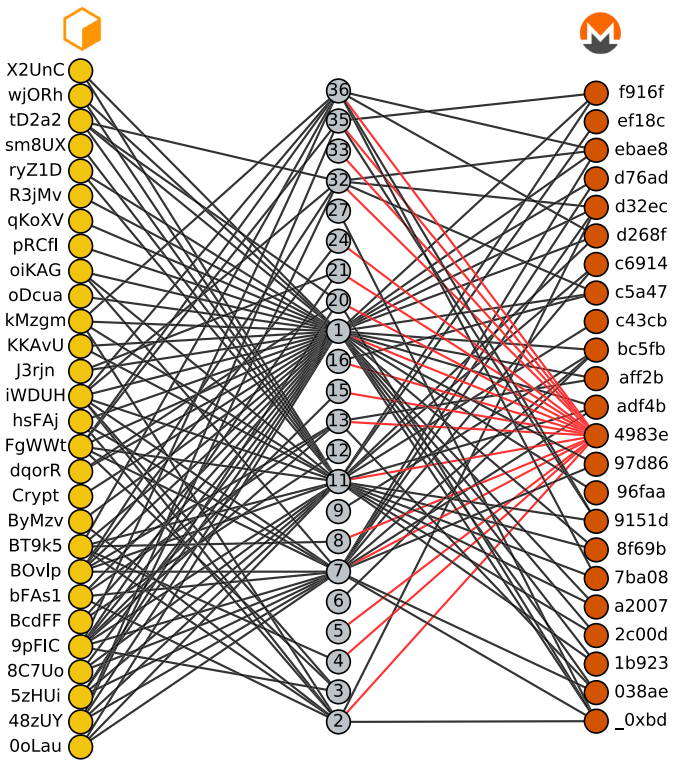


Fig. 14. A 3-partite graph of discovered cryptomining-related keys within each campaign. Each key is represented by its first 5 characters. Red threads highlight the reuse of the same key "4983e34ef01b4b579725b3a228e59e79" in 18 different campaigns. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

23, 5555 (ADB) and 8080. Upon further analysis, this behavior could be attributable to the infamous `Hide and Seek` botnet (Rootkiter, 2018b).

For port 32,764 which is related to a backdoor vulnerability (Thomas, 2018), the proposed IoT botnet clustering approach revealed a campaign of substantial size (#19 in Table 8), consisting of 2140 active IoT scanners with the signature of $<$\{23, 32764, 80, 8000, 8080, 8081, 8089, 8090, 81, 8181, 8443, 8888, 9000\}, Flag=1, ARR=1$>$. We did not come across any previously reported botnet families that scan such ports. As a result, we postulated that this campaign is either new or specific ports have been recently added to the target list of a previously known IoT botnet. Another aspect is that this is the only large campaign that exploited a relatively significant number of `NUUO` products, which is a common indicator of the `Reaper` IoT botnet. The `JenX` botnet (Exchange, 2018), which scans ports 37,215 and 52869, was also disclosed. Moreover, a botnet with $<$\{2004, 80, 8080, 81\}, Flag=0, ARR=2 $>$ was also discovered and consisted of 35 coordinated IoT scanners, all of which compromised QNAP NAS. This campaign strongly resembles the `Muhstik` botnet (Rootkiter, 2018a), with the exception of the substitution of port 7001 with 81 in the target port set.

With the prevalence of IoT botnets, port 5555 (Android debug bridge) has become a popular target port. We found 23 IoT botnets that include port 5555 as part of their target set. Based on the reports on ADB miner (Netlab, 2019a) and the similarity of its scanning module to `Mirai`, we can attribute the inferred large IoT botnet (#5 in Table 8) to `Mirai` or its variant `Fbot` (Netlab, 2019b). Additionally, we found `xmrMiner` instances with the same previously noted key (of Fig. 14 in the latter campaign and in campaign #16. Based off the set of target ports pertaining to campaign #25 (port 23, 5358), it seems to be highly likely attributed to that of the `Hajime` (Herwig et al., 2019; Radware, 2018) IoT botnet. In total, this campaign possessed 1059 active IoT scanners (made up of IP cameras/DVRs).

4.4.7. A note on Industrial Control Systems (ICS).

We also inferred an IoT botnet of 25 bots with the signature $<$\{102, 8888, 993\}, Flag=0, ARR=1$>$, probing Siemens S7 (heavily used in SCADA systems), IEC 61,850 and ICCP (both are mostly used in utility/electric substations) on port 102. To provide additional insights, we also actively scanned each of the identified compromised IoT devices for ICS open ports on TCP and UDP 102 (S7), 502 (MODBUS), 20,000 (DNP3), 47,808 (BACNET) and 1911 (FOX) and found 100, 101, 465, 70 and 85 devices with open ports, respectively. We note that we have also inferred close to 40 devices having simultaneously all the above-mentioned ICS ports open, which we thought are related to ICS honeypots. Nevertheless, the appearance of compromised IoT devices within ICS setups is alarming.

5. Concluding remarks

With the continuous adoption of the IoT paradigm in critical infrastructure and consumer sectors, their security and privacy concerns are becoming quite serious, leading to devastating consequences. This work compliments current IoT-centric research by offering a macroscopic, generic and passive methodology to infer Internet-scale compromised IoT devices and to report on ongoing IoT botnets. The work initially introduces a novel darknet-specific sanitization model that contributes to the field of Internet measurements at large. Subsequently, by devising a binary classifier based upon a CNN in conjunction with active measurements, the proposed work is capable of fingerprinting compromised IoT devices by solely operating on darknet traffic. Consequently, by automating the generation of signatures related to the ports being probed coupled with their distribution in addition to other simplistic yet effective features, the proposed approach provides the capability to infer ongoing orchestrated botnets. The results demonstrate the significant security issue with the IoT paradigm by exposing more than 400,000 exploited IoT devices during only a 24-hour period, some of which have been deployed in critical sectors such medical and manufacturing. Additionally, the outcome provides evidence-based indicators related to ongoing IoT botnets such as those of `Mirai`, `Hide and Seek`, and `Reaper`, to name a few. More interestingly, the results demonstrate evolving IoT botnets with cryptojacking capabilities, where many of those seem to be attributed to the same mastermind by exposing the same employed key.

Future work will address current limitations. This includes addressing the misidentification of two distinct IoT botnets (as one larger campaign) by including packet IAT related features, and improving the tagging/labeling procedure. We will also examine IoT-specific malware samples and devising formal methodologies between the traffic they generate from one side and the corresponding darknet traffic from the other side, to fortify the attribution evidence. Moreover, extracting a number of features from IoT malware binaries will empower the attribution of each malware sample to their respective campaigns, enabling agile IoT botnet inference and characterization for mitigation and remediation purposes.

Declaration of Competing Interest

The authors declare that they do not have any financial or non-financial conflict of interests.

Acknowledgment

The authors would like to express their sincere gratitude in advance to the anonymous reviewers and the editors for their constructive feedback. This work was supported by two grants from

the U.S. National Science Foundation (NSF) (Office of Advanced Cyberinfrastructure (OAC) #1907821 and #1917117).

References

Acar, G., Huang, D.Y., Li, F., Narayanan, A., Feamster, N., 2018. Web-based attacks to discover and control local IoT devices. In: Proceedings of the 2018 Workshop on IoT Security and Privacy. ACM, pp. 29–35.

Agrawal, R., Imieliński, T., Swami, A., 1993. Mining association rules between sets of items in large databases. In: ACM Sigmod Record, 22. ACM, pp. 207–216.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., et al., 2017. Understanding the Mirai Botnet.

Araki, S., Takahashi, K., Hu, B., Kamiya, K., Tanikawa, M., 2019. Subspace clustering for interpretable botnet traffic analysis. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, pp. 1–6.

Benson, K., 2016. Leveraging Internet Background Radiation for Opportunistic Network Analysis. UC San Diego Ph.D. thesis.

Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems, pp. 2546–2554.

Bertino, E., Islam, N., 2017. Botnets and internet of things security. Computer (2) 76–79.

Bou-Harb, E., Debbabi, M., Assi, C., 2016. A novel cyber security capability: inferring internet-scale infections by correlating malware and probing activities. Computer Networks 94, 327–343.

Cashdollar, L., 2019. Latest Echobot: 26 Infection Vectors. https://blogs.akamai.com/sitr/2019/06/latest-echobot-26-infection-vectors.html

Ceron, J.M., Steding-Jessen, K., Hoepers, C., Granville, L.Z., Margi, C.B., 2019. Improving IoT botnet investigation using an adaptive network layer. Sensors 19 (3), 727.

Cetin, O., Ganán, C., Altena, L., Kasama, T., Inoue, D., Tamiya, K., Tie, Y., Yoshioka, K., van Eeten, M., 2019. Cleaning up the internet of evil things: real-world evidence on ISP and consumer efforts to remove MIRAI.

Cha, S.-H., 2007. Comprehensive survey on distance/similarity measures between probability density functions. City 1 (2), 1.

Cheng, Y., Wang, F., Zhang, P., Hu, J., 2016. Risk prediction with electronic health records: adeep learning approach. In: Proceedings of the 2016 SIAM International Conference on Data Mining. SIAM, pp. 432–440.

Chollet, F., et al., 2015. Keras. https://keras.io

Cochran, W.G., 2007. Sampling Techniques. John Wiley & Sons.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12 (Aug), 2493–2537.

Coinhive, 2018. https://coinhive.com/. [Online; accessed 01-March-2019]

Da Xu, L., He, W., Li, S., 2014. Internet of things in industries: a survey. IEEE Trans. Ind. Inform. 10 (4), 2233–2243.

Dainotti, A., King, A., Claffy, K., Papale, F., Pescapé, A., 2015. Analysis of a /0 stealth scan from a botnet. IEEE/ACM Trans. Netw. 23 (2), 341–354.

Dowling, S., Schukat, M., Barrett, E., 2018. Using reinforcement learning to conceal honeypot functionality. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 341–355.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A., 2015a. A search engine backed by internet-wide scanning. In: 22nd ACM Conference on Computer and Communications Security.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A., 2015b. A search engine backed by internet-wide scanning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 542–553.

Durumeric, Z., Bailey, M., Halderman, J.A., 2014. An internet-wide view of internet-wide scanning. In: USENIX Security Symposium, pp. 65–78.

Durumeric, Z., Wustrow, E., Halderman, J.A., 2013. Zmap: fast internet-wide scanning and its security applications. In: USENIX Security Symposium, 8, pp. 47–53.

Exchange, I.X.-F., 2018. Jenx Botnet. https://exchange.xforce.ibmcloud.com/collection/JenX-Botnet-c47476c5e6fafd7df487cecd1110a761 [accessed 01-March-2019]

Fachkha, C., Bou-Harb, E., Debbabi, M., 2015. On the inference and prediction of DDoS campaigns. Wirel. Commun. Mob. Comput. 15 (6), 1066–1078.

Fachkha, C., Debbabi, M., 2016. Darknet as a source of cyber intelligence: survey, taxonomy, and characterization. IEEE Commun. Surv. Tut. 18 (2), 1197–1227.

Feng, X., Li, Q., Wang, H., Sun, L., 2018. Acquisitional rule-based engine for discovering internet-of-things devices. In: 27th {USENIX} Security Symposium ({USENIX} Security 18), pp. 327–341.

Fontugne, R., Borgnat, P., Abry, P., Fukuda, K., 2010. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In: Proceedings of the 6th International COnference. ACM, p. 8.

Ford, M., Stevens, J., Ronan, J., 2006. Initial results from an ipv6 darknet13. In: International Conference on Internet Surveillance and Protection. IEEE. 13–13

García, S., Luengo, J., Herrera, F., 2015. Data Preprocessing in Data Mining. Springer.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2014. Bayesian Data Analysis, 2. Taylor & Francis.

GreyNoise, 2019. https://viz.greynoise.io/.

Gu, G., Perdisci, R., Zhang, J., Lee, W., 2008. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection.

Guarnizo, J., Tambe, A., Bunia, S. S., Ochoa, M., Tippenhauer, N., Shabtai, A., Elovici, Y., 2017. Siphon: towards scalable high-interaction physical honeypots. arXiv:1701.02446.

Guo, H., Heidemann, J.S., 2018. Detecting IoT Devices in the Internet (Extended).

Herwig, S., Harvey, K., Hughey, G., Roberts, R., Levin, D., 2019. Measurement and Analysis of Hajime, A Peer-to-Peer IoT Botnet.

Homayoun, S., Ahmadzadeh, M., Hashemi, S., Dehghantanha, A., Khayami, R., 2018. Botshark: a deep learning approach for botnet traffic detection. In: Cyber Threat Intelligence. Springer, pp. 137–153.

Husák, M., Neshenko, N., Pour, M.S., Bou-Harb, E., Čeleda, P., 2018. Assessing internet-wide cyber situational awareness of critical sectors. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. ACM, p. 29.

ISTR, S.I.S.T.R., 2018. Cryptojacking: A Modern Cash Cow. Technical Report.

Jung, J., Paxson, V., Berger, A.W., Balakrishnan, H., 2004. Fast portscan detection using sequential hypothesis testing. In: Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on. IEEE, pp. 211–225.

Khan, K., Rehman, S.U., Aziz, K., Fong, S., Sarasvady, S., 2014. Dbscan: past, present and future. In: The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014). IEEE, pp. 232–238.

KoronIoTis, N., Moustafa, N., Sitnikova, E., 2019. Forensics and deep learning mechanisms for botnets in internet of things: a survey of challenges and solutions. IEEE Access 7, 61764–61785.

Kumar, D., et al., 2019. All things considered: an analysis of IoT devices on home networks. 28th USENIX Security Symposium (USENIX Security 19). USENIX Association, Santa Clara, CA.

Luo, T., Xu, Z., Jin, X., Jia, Y., Ouyang, X., 2017. Iotcandyjar: Towards an Intelligent-Interaction honeypot for IoT Devices. Black Hat.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y., 2018. N-baIoT–network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. 17 (3), 12–22.

Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y., 2017. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis.

Metongnon, L., Sadre, R., 2018. Beyond telnet: Prevalence of IoT protocols in telescope and honeypot measurements. In: Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity. ACM, pp. 21–26.

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., Tarkoma, S., 2017. Iot sentinel: automated device-type identification for security enforcement in IoT. In: Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on. IEEE, pp. 2177–2184.

Netlab, 2019a. Adb.miner: More Information. https://blog.netlab.360.com/adb-miner-more-information-en/ [Online; accessed 01-March-2019].

Netlab, 2019b. Fbot, a satori related botnet using block-chain DNS system. https://tinyurl.com/yavvhf4v [Online; accessed 01-March-2019].

Moore, D., Shannon, C., Brown, D.J., Voelker, G.M., Savage, S., 2006. Inferring internet denial-of-service activity. ACM Trans. Comput. Syst. 24 (2), 115–139.

Nawrocki, M., Wählisch, M., Schmidt, T. C., Keil, C., Schönfelder, J., 2016. A survey on honeypot software and data analysis. arXiv:1608.06249.

Nguyen, T. D., Marchal, S., Miettinen, M., Dang, M. H., Asokan, N., Sadeghi, A.-R., 2018. D\" IoT: a crowdsourced self-learning approach for detecting compromised IoT devices. arXiv:1804.07474.

Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.-R., 2019. Dïot: A federated self-learning anomaly detection system for IoT. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 756–767.

Nigam, R., 2019. New Mirai Variant Adds 8 New Exploits, Targets Additional IoT Devices. https://unit42.paloaltonetworks.com/new-mirai-variant-adds-8-new-exploits-targets-additional-IoT-devices/

Ozawa, S., Ban, T., Hashimoto, N., Nakazato, J., Shimamura, J., 2019. A study of IoT malware activities using association rule learning for darknet sensor data. International Journal of Information Security 1–10.

Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C., 2016. Iotpot: a novel honeypot for revealing current IoT threats. J. Inf. Process. 24 (3), 522–533.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Pinheiro, A.J., Bezerra, J.d. M., Burgardt, C.A., Campelo, D.R., 2019. Identifying IoT devices and events based on packet length from encrypted traffic. Comput. Commun. 144, 8–17.

Policy, I. M. F., Trust, A. O. C.-R., 2019. https://impactcybertrust.org/.

Pumperla, M., 2019. https://github.com/maxpumperla/hyperas.

Radware, 2018. Hajime Botnet Friend or FOE?. https://security.radware.com/ddos-threats-attacks/hajime-IoT-botnet/ [Online; accessed 01-March-2019]

Rootkiter, Y., 2018a. Botnet Muhstik is Actively Exploiting Drupal cve-2018-7600 in a Worm Style. https://blog.netlab.360.com/botnet-muhstik-is-actively-exploiting-drupal-cve-2018-7600-in-a-worm-style-en/. [Online; accessed 01-March-2019]

Rootkiter, Y., 2018b. Hns Botnet Recent Activities. https://blog.netlab.360.com/hns-botnet-recent-activities-en/ [Online; accessed 01-March-2019]

Rossow, C., 2014. Amplification hell: Revisiting network protocols for DDoS abuse. NDSS.

Shaikh, F., Bou-Harb, E., Neshenko, N., Wright, A.P., Ghani, N., 2018. Internet of malicious things: correlating active and passive measurements for inferring and

characterizing internet-scale unsolicited IoT devices. IEEE Commun. Mag. 56 (9), 170–177.
Shodan, 2019. The Search Engine for Internet of Things. http://shodan.io
Siby, S., Maiti, R.R., Tippenhauer, N.O., 2017. Iotscanner: detecting privacy threats in IoT neighborhoods. In: Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security. ACM, New York, NY, USA, pp. 23–30. doi:10.1145/3055245.3055253.
Team, C., 2017. Internet-wide scan data repository. Retrieved 2017.
Thai-Nghe, N., Gantner, Z., Schmidt-Thieme, L., 2010. Cost-sensitive learning methods for imbalanced data. In: Neural Networks (IJCNN), The 2010 International Joint Conference on. IEEE, pp. 1–8.
Thangavelu, V., Divakaran, D.M., Sairam, R., Bhunia, S.S., Gurusamy, M., 2018. Deft: a distributed IoT fingerprinting technique. IEEE Internet Things J.
Thomas, S.L., 2018. Backdoor Detection Systems for Embedded Devices. University of Birmingham Ph.D. thesis.
Torabi, S., Bou-Harb, E., Assi, C., Galluscio, M., Boukhtouta, A., Debbabi, M., 2018. Inferring, characterizing, and investigating internet-scale malicious IoT device activities: a network telescope perspective. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, pp. 562–573.
Tuttle, H., 2018. Cryptojacking. Risk Manag. 65 (7), 22–27.
UNB, 2019. ISCX Botnet Dataset. http://www.unb.ca/research/iscx/dataset/ISCX-botnetdataset.html#Botnet%20Data%20set
xmrMiner, 2019. Monero Web Miner. https://xmrminer.cc/
Xu, D., Tian, Y., 2015. A comprehensive survey of clustering algorithms. Ann. Data Sci. 2 (2), 165–193.
Zimba, A., Wang, Z., Mulenga, M., 2019. Cryptojacking injection: a paradigm shift to cryptocurrency-based web-centric internet attacks. J. Organ. Comput. Electron. Commerce 29 (1), 40–59.
ZoomEye, 2019. http://www.zoomeye.org/.

Morteza Safaei Pour is currently a Ph.D. candidate and a member of the Cyber Center for Security and Analytics at University of Texas at San Antonio. He received his B.S. and M.S. in electrical engineering (communications) from Sharif University of Technology, Tehran, Iran, in 2013 and 2016, respectively. He has executed several projects in various fields including side channel attacks on embedded systems, stochastic analysis of malware propagation, and detection of IoT probing campaigns, to name a few. His research interests include operational cyber security, IoT security, security of critical infrastructure and smart cities, AI and machine learning.

Antonio Mangino is currently pursuing a master degree in Information Systems and Cyber Security. He received his B.S. in Computer Science from Florida Atlantic University (FAU) in 2019. As a member of the Cyber Threat Intelligence Laboratory at Florida Atlantic University, Antonio has worked on the development of various network and cyber security projects, with a focus on the IoT paradigm. His research interests include network analysis, operational cyber security and information security.

Kurt Friday is currently a Ph.D. student with an emphasis on cyber security at University of Texas at San Antonio (UTSA). He previously completed his B.S. in computer science at Florida Atlantic University. He is additionally an active member of the Cyber Center for Security and Analytics, where he conducts research within the domains of security, software defined networking (SDN), and data science.

Matthias Rathbun is currently a high hchooler at Boca Raton Community High School and a member of the Cyber Threat Intelligence Laboratory at Florida Atlantic University. He is the founder of his previous school Hacking Team which applied different aspects of Machine Learning to challenges at Hackathon Competitions. He has applied machine learning to multiple disciplines including bioinformatics and physics along with currently, IoT cybersecurity. His research interests include Machine Learning, Data Visualization, and Big Data.

Elias Bou-Harb is currently an associate professor at the department of Computer Science at University of Texas at San Antonio. Previously, he was a visiting research scientist at Carnegie Mellon University (CMU) under the sponsorship of Professor Bruno Sinopoli. He is also a permanent research scientist at the National Cyber Forensic and Training Alliance (NCFTA) of Canada. The latter is an international organization which focuses on the investigation of cyber-crimes impacting citizens and businesses. Dr. Bou-Harb holds a Ph.D. degree in computer science from Concordia University in Montreal, Canada, which was executed under the supervision of Professors Mourad Debbabi and Chadi Assi. His research and development activities and interests focus on the broad area of operational cyber security, including, attacks detection and characterization, malware investigation, cyber security for critical infrastructure and big data analytics.

Farkhund Iqbal holds the position of associate professor and Director Advanced Cyber Forensics Research Laboratory in the College of Technological Innovation, Zayed University, United Arab Emirates. He holds a Master (2005) and a Ph.D. degree (2011) from Concordia University, Canada. He is using machine learning and Big Data techniques for problem solving in healthcare, cybersecurity and cybercrime investigation in smart and safe city domain. He has published more than 80 papers in high ranked journals and conferences. He is an affiliate professor in School of Information Studies, McGill University, Canada and Adjunct Professor in Faculty of Business and IT, University of Ontario Institute of Technology, Canada. He is the recipient of several prestigious awards and research grants. He has served as a chair and TPC member of several IEEE/ACM conferences and is the reviewer of high rank journals. He is the member of several professional organization including ACM and IEEE Digital Society.

Sagar Samtani is an assistant professor with the Muma College of Business' Information Systems and Decision Sciences Department at the University of South Florida. Samtani earned a PhD and a master's degree in management information systems from the Univesity of Arizona and a bachelor's degree from the same institution. He served as National Science Foundation Scholarship-for-Service Fellow from 2014 to 17. His research interests are within the cybersecurity domain, specifically on cyber threat intelligence and large-scale vulnerability assessment. Samtani employs and develops novel data/text/web mining algorithms and systems, specifically, deep learning and network analysis within these contexts. His work has been published in several journals, including the Journal of Management Information Systems and IEEE Intelligent Systems. He has presented at several conferences including Woman in Cybersecurity, IEEE Intelligence and Security Informatics and the INFORMS annual meeting.

Jorge Crichigno received his Ph.D. in electrical and computer engineering from the University of New Mexico, Albuquerque, USA, in 2009. Prior to that, he received his M.Sc. and B.Sc. in electrical engineering from the University of New Mexico and from the Catholic University of Asuncion in 2008 and 2004, respectively. He is currently an associate professor in the Department of Integrated Information Technology, College of Engineering and Computing, University of South Carolina, Columbia, SC. He was also a visiting professor at the Florida Center for Cybersecurity, Tampa, FL, in 2016. His research interests include highspeed networks, network security and science DMZs, and STEM education. He has served as a reviewer for numerous journals and a TPC member for many conferences, including IEEE Transactions on Mobile Computing and IEEE Globecom. He has also been an invited panelist for various NSF STEM education initiatives. He is a member of the IEEE Computer Society.

Nasir Ghani is a professor in the Department of Electrical Engineering at the University of South Florida and Research Liaison for Cyber Florida. Earlier he was Associate Chair of the Electrical and Computer Engineering Department at the University of New Mexico, USA, and prior to that, a faculty member at Tennessee Tech University. He also spent several years working at large corporations (including Nokia, IBM, and Motorola) and several hi-tech startups. He has co-authored over 220 publications and received the NSF CAREER award in 2005. He has co-chaired the IEEE Technical Committee on High Speed Networks (TCHSN) and has served as an associate editor for IEEE Communications Letters, IEEE/OSA journal of Optical Communications and Networking, and IEEE Systems. He has also co-chaired and organized many symposia and workshops for leading IEEE ComSoc conferences, including IEEE ICC, IEEE Globecom, IEEE Infocom, and IEEE ICCCN. His research interests include high-speed cyberinfrastructure design, cybersecurity, cyberphysical and IoT systems, cloud computing, and disaster recovery. He received his Bachelors from the University of Waterloo, his Masters from McMaster University, and his Ph.D. from the University of Waterloo.