Tracking IoT P2P Botnet Loaders in the Wild

Hatem A. Almazarqi*, Mathew Woodyard†, Troy Mursch†, Dimitrios Pezaros*, Angelos K. Marnerides*
*School of Computing Science, University of Glasgow, Glasgow, Scotland, UK
h.almazarqi.1@research.gla.ac.uk, [dimitrios.pezaros, angelos.marnerides]@glasgow.ac.uk
†Okta, Inc., San Francisco CA, USA
[mathew.woodyard, troy.mursch]@okta.com

***Abstract*—Evidently, centralised botnets are nowadays considered as easy targets for take-down efforts by law enforcement and computer security researchers. Hence, malicious actors transitioned towards the implementation of Peer-to-Peer (P2P) IoT botnets such to solidify their infrastructures, avoid single points of failure and further evade back tracking. Consequently, due to the highly distributed persona of modern P2P botnets, the detection of critical nodes to aid for the effective capturing of emerging threat vectors in such setups evolved into a challenging task. In this work, we conduct a novel 24-month longitudinal study based on real Internet measurements from globally distributed honeypots focusing on propagation trends of P2P IoT botnets. In order to achieve this, we develop graph-based centrality metrics to attribute AS-level connectivity characteristics to botnet and malware propagation as well as relating AS-level tolerance for botnet malware hosts we refer to as loaders. In general, we argue that the proposed methodology and outcomes of the herein study, can significantly benefit security experts and network operators towards the design of mitigation measures against present and future P2P botnets.**

***Index Terms*—IoT botnets, Internet measurements, cybersecurity, malware, cyber threat intelligence.**

I. Introduction

The IoT market expansion in synergy with botnets targeting IoT devices and modern cyberwarfare techniques evolution has resulted to be a significant and challenging threat to confront in networked systems. In general, IoT botnets can be described as a group of compromised IoT devices ('bots') which are infected with malware and controlled via a single entity ('a malicious actor') or organised groups of 'hacktivists' [1]. Such devices include, but are not limited to, home routers, smart meters, Internet-enabled DVRs, wearables and programmable logic controllers. Each device runs multiple pieces of applications, or software, which are increasing in complexity, and can have vulnerabilities that can be exploited, resulting in a diverse set of exploits. Hence, compromised IoT devices can be instrumented by malicious actors to perform various malicious activities, including Distributed Denial of Service (DDoS), ransomware campaigns, phishing and spamming.

IoT-based botnets expand their network over the Internet via infecting a potential victim. Thus, analysing the the dependencies of structural entities of an IoT-based botnet can reveal communication characteristics and attribute their evading strategies. Since the development of the first botnet in 1999 (Pretty Park botnet), botnet communication architectures emerged in response to the growing effort to identify botnets using their communication structure and communication patterns [2]. The primitive Pretty Park botnet implementation was able to download and execute a file on the victim through using IRC server as a remote-control server. Nonetheless, such schemes have significantly changed with the convergence of IoT technologies and the pervasiveness invoked by the services they operate or access.

IoT botnet architectures mainly fall into two broad categories: (i) centralised, and (ii) Peer-to-Peer (P2P) also known as decentralised. Centralised architectures consist of the simple setup where compromised IoT receive commands from a centralised command and control (C&C) server. However, this approach is prone to a single point of failure and relatively easily detected by authorities [3]. Due to the weaknesses of the centralised IoT botnet architecture and by virtue of widely accessible IoT botnet source code with the Mirai botnet, botnet developers shifted in favour of decentralised setups through the P2P paradigm such as to increase their resilience [4]. Within a P2P botnet architecture, a malicious actor orchestrates control commands to more than one bot who subsequently relay them to their neighbouring IoT nodes. In general, a P2P IoT botnet can operate with little or no central coordination and even if a single host is taken offline by the defence, the botnet still remains under the command of the malicious actor and it could span across multiple Internet Autonomous Systems (ASes).

As highlighted by many studies and security vendors, P2P botnets are hard to backtrack. Therefore, profiling their structural characteristics across the global Internet is a challenging task. Overall, there is still a lack of mechanisms on tracking critical components in charge of instrumenting the formation of P2P botnets such as botnet loaders, or vital supernodes in charge of coordination [5]. Hence, the development of generic methods to identify critical nodes is still an open issue and it would surely benefit future botnet mitigation strategies.

In this work we present a novel longitudinal analysis using real Internet measurements to profile critical structural properties and associations in the context P2P IoT botnets. The herein analysis considers AS-level routing as well as real Cyber Threat Intelligence (CTI) feeds gathered for a period of 24 months from Regional Internet Registries (RIRs) and Okta/Auth0's globally distributed attack honeypots. In order to explore the the geographic distribution of the examined botnets and map important nodes such as botnet loaders to their originating ASes, we leverage Internet geolocation

CTI data				
Observation Period				
01/07/2020 - 11/07/2022				
IP addresses	**ASes**	**Origin Countries**	**Payloads**	**Botnet loaders**
205,618	4,536	190	466,000	18,476

TABLE I. Summary of CTI feeds gathered from the global distributed honeypots as well as the number of ASes that originate bots traffic and host botnet loaders.

data[1] as well as Internet topology information [2]. In general, this work aims to profile the strategy adopted by IoT based botnets in expanding their network and increasing their level of resilience. We term resilience as a measure for how difficult it is for defenders to disrupt the botnet network. The main contributions of this paper are:

1) A novel method on describing the AS-level propagation strategy adopted by modern IoT P2P botnets.
2) Attributing the AS-level tolerance over P2P botnet loaders using graph-based centrality metrics.
3) Insights on the structural properties of botnet loaders with respect to the distribution of malware binaries of various strains.

The remainder of this paper is structured as follows: Section II provides background information on the structure of IoT-based botnet and an overview of related work. Section III describes the datasets and methodology used in this work. Section IV is dedicated on presenting our findings. Finally, Section V summarises and concludes this work.

II. Background & Related Work

A. IoT Botnet architectures

As already mentioned, botnets can be stratified in terms of their communication architecture that can be either (i) centralised, (ii) P2P, and (iii) hybrid. In the centralised setup, the botmaster instructs a centralised Command and Control (i.e., C&C) server to send a command to the bots. By contrast, P2P bots are distinct from conventional bots since their command and control module is designed through a relay-type paradigm adhering to P2P principles. There is no central point for a C&C server in such design and any host in the network can work as a client and a server at the same time. In this scenario, the botmaster can communicate directly with a bot and the commands are relayed among the bots. In order to employ a better structure with respect to bot orchestration, the P2P architecture has recently evolved towards a hybrid scheme. Compromised devices in this architecture are categorized into two groups: (i) servant bots, and (ii) client bots. The first group are called servant bots, as they act as both servers and clients, and have static IP addresses (routable IPs) which are simply accessible from the entire Internet. Conversely, bots in the second group do not accept incoming connections and consists of bots operates behind firewalls that are inaccessible from the global Internet as well as bots with dynamically assigned IP addresses (non-routable IPs).

[1]MaxMind: https://www.maxmind.com/en/home
[2]RIPEstat: https://stat.ripe.net

Country	Number of botnet loaders	Number of ASes
China	4922	25
India	2430	66
United States	528	76
Albania	444	2
South Korea	290	26

TABLE II. The geographical distribution of botnet loaders with respect to hosted ASes.

Furthermore, P2P architectures in IoT botnets can be categorised in terms of how bots are distributed. Hence we could have both structured and unstructured setups with loose hierarchy across bots. In structured bot setups, compromised devices are able to interact with one another via the use of the crafted P2P protocol in order to update its neighbour peer information. Such botnet-related P2P protocols are commonly based on Distributed Hash Table (DHT) maintained by botmasters. Through the functionalities offered by DHTs, botmasters are able to search running botnet service using hash table (key, value) pairs and storing information in the DHT instance running on every bot. In unstructured bot setup, the compromised devices do not maintain a seed list, and scan the network to collect information in order to identify potential bots. No specific network topology is defined in such setup and does not support key lookups function. Therefore, the difference between structured and unstructured systems relays in the method of adding peers to the botnet network.

Botmasters implement an external server known as a botnet loader in charge of hosting a variety of malware strains. Hence, loaders can serve multiple botnets that operate over distinct malware strains. In general, a loader is considered as one of the most crucial components of a botnet that is able to facilitate its propagation. A loader is used to instruct vulnerable devices to reach particular DNS domains in order to download specific malware strains. Hence, loaders recruit new bots and enable the dissemination of executables that are responsible for targeting various IoT platforms such as MIPS and ARM by directly communicating with potential victims.

B. Related Work

Several research efforts have focused on detecting the structure of P2P botnets (e.g., [1], [6]–[8]). Work described in [1] provided a novel insight on the evolving of Mozi, a new P2P-based Mirai-like variant with regard to infection and scanning strategies. The work in [7] proposed a model for detecting evolving P2P botnet communities in dynamic communication graphs. In parallel, the work described in [8] proposed an approach to detect P2P bots in network traffic by employing machine learning in synergy with dynamic group behaviour analysis (DGBA). However, the dependency of botnet infrastructures with botnet loaders are not covered in any of the aforementioned pieces of work. In addition, and

by contrast with these studies, we provide an insight on the propagation strategy adopted by IoT botnets.

There have also been studies exploiting graph properties to identify the presence of P2P botnets (e.g. [2], [4], [5], [9], [10]). Work conducted in [2] developed a dynamic monitoring approach that leverages graph-theoretic properties of botnet network to defeat exfiltration attempts by modern botnets, including P2P botnets. Moreover, the work in [5] leveraged graph-theoretic metrics in order to develop detection mechanisms aimed to detect sensor nodes on P2P botnet. Similarly, the work in [10] identifies bot communities based on characterising the communication amongst network nodes using metrics distilled by undirected graph definitions such as node degree and conductance. Evidently, most such studies focused on detecting anomalies in dynamic or static graphs, however, they have not adequately attributed the criticality of specific botnet nodes and their behaviour with respect to their AS-level distribution as we do herein. In addition, most of the previous studies did not capture the propagation characteristics of P2P botnet with respect to global distribution as we do in this work.

Algorithm 1: Identification of IoT botnet propagation strategy.

Input: IP addresses, URLs
Output: self_propagate, centralised
$centralised \leftarrow \emptyset$;
$selfPropagate \leftarrow \emptyset$;
i=0
while $i < No.\ of\ URL\_IP$ **do**
  temp $\leftarrow\emptyset$
  if $src\_IP[i]$ == $URL\_IP[i]$ **then**
    Add $URL\_IP[i]$ *to centralised*
  else
    $j = 0$
    while $j < No.\ of\ src\_IP$ **do**
      if $URL\_IP[i] == URL\_IP[j]$ *and* $src\_IP[j] \notin temp$ **then**
        Add $src\_IP[j]$ *to temp*
      end
      j = j+1
    end
    Add {key: $URL\_IP[i]$, values: temp} to $self\_propagate$
    i = i +1
  end
end

III. DATASET DESCRIPTION & METHODOLOGY

A. Dataset Description

During our observations, we have collected 435K payloads generated by different IoT botnet families, including Mirai, Bashlite and Gafgyt that were attempting to enslave potential victims and expand the botnet network. As summarised in Table I our observations were stemmed from 205,618 distinct IP addresses located across 4,536 Autonomous Systems (ASes) spanning 190 countries between July 2020 and July 2022.

The received payloads contain embedded URLs and shell commands that are used to direct the victim to download malicious binaries. We have extracted the URLs from all gathered payload by leveraging regular expressions such as to locate botnet loaders. Code listing 1 represents a sample of malicious payload received by honeypot targeting an AVTech IoT device. As shown, the sample payload contains a URL used to direct the victim to download binaries from a specific domain that we anonymise. In addition, the payload embeds the piping of a chmod command (i.e., chmod 777) after the device downloads the binary via the wget instruction such as to provide full read/write/execute privileges to the downloaded binary. Hence, the malware taking full access control over the infected system. Through appropriate parsing of URLs we compared IP addresses with active loader instructions with source IP addresses in our feeds such as to identify the propagation strategy adopted by the examined botnet strains as per algorithm 1. If the source IP (src_IP) does not match the extracted URL (URL_IP), the proposed algorithm classify the source IP as a malicious bot that adopts a P2P architecture ($self\_propagate$). In this case, the malicious bot acts as a C&C server and instructs the potential victim to download malicious binaries from a loader server. On the other hand, if the source IP (src_IP) matches the extracted URL (URL_IP), the proposed algorithm will classify the source IP as a loader server that is controlled by a C&C server ($centralised$). The malicious actors instruct the loader server to to login to vulnerable IoT devices and download botnet malware. Based on our proposed algorithm, we have totally detected 1,955 bot loaders participating in forming P2P botnet.

The geographical analysis of botnet loaders shows that China, India, United States, Albania and South Korea are preferred countries for malicious actors to setup a botnet loader since 84% of servers are located in these five countries. The remaining botnet loaders are spread over 861 ASes and 74 countries. Evidently, some botnets conform to a certain architecture and loaders are distributed across multiple networks and countries. Table II describes the geographical distribution of botnet loaders used by attackers to expand their P2P botnet networks.

Code Listing 1: Sample of a malicious payload received by our honeypots.

```
GET/cgi-bin/supervisor/CloudSetup.cgi?exefile=
    cd /tmp;rm -rf *; wget http://X.X.X.X/bins
    /ayylmao420kekuaintge -O 27.x; chmod 777
    27.x; ./27.x avtech; echo keksec HTTP/1.1
```

B. Methodology

Centrality measures have been used in our study as a decision-making tool in order to address a variety of issues pertaining to network security. Such metrics have been used in the past to identify critical nodes in an effort to mitigate or prevent computer viruses or malware spreads [11]. In addition, they were used to quantify the potential threat of websites exposing API vulnerabilities [12].

In the herein described work, we study the centrality properties of botnet loaders from a graph-theoretical perspective. In particular, the concept of centrality is applied to determine node significance with respect to its graph connectivity. Furthermore, through the centrality measure we assess the level of influence or significance of vertex in a graph and reflect on specific Internet topology properties. Hence, we employ metrics associated to centrality such as degree centrality, betweenness centrality, closeness centrality and local clustering coefficients to profile critical nodes in a botnet P2P network and analyse its robustness.

C. Graph connectivity

A graph G consists of a finite set of vertices or nodes and a finite set E of edges or links. A set of nodes representing all bots on a botnet network G is written as:

$$V(G) = \{v1, v2, v3, ..., vn\} \tag{1}$$

The edges (e) represent neighborhood relations between the nodes and are defined as:

$$E(G) = \{u_a v_a, u_b v_b, ..., u_n v_n\} \tag{2}$$

where each pair $e = (u, v)$ denotes a connection between two nodes in $G(V)$. For example, the edge is added to the set of edges E, when communication is observed between node u_a and v_a. Moreover, if a communication detected between vertice (v_i) and (v_j), then edge $(e_i j) = (v_i, v_j)$ is added to to the set of edges E. Eventually, a botnet communication graph is generated from monitoring the traffic between bots and botnet loaders. Similarly, we construct a graph representing the connectivity among ASes embracing bots and ASes hosting botnet loaders.

D. Centrality measures

Degree centrality: represents the total number of edges connected to a certain node. By using the following formulation, we can define the degree centrality of each node in the P2P botnet network.

$$CD(v) = \frac{df}{(|N| - 1)} \tag{3}$$

Where d_v is the degree of node which is the number of connected edges, and N is the total number of edges on the graph. It can take a value from 0 when the node does not have a connection with its neighbours, to $N - 1$ when a node is connected to all its neighbours. A high degree centrality indicates high node significance in the network.

Betweenness Centrality: reflects the fraction of shortest paths that go through the node relative to the total number of shortest paths in the graph. It also quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. Thus, the betweenness centrality C_B of node i can be computed as follows:

$$C_B(i) = \sum_{j \neq k \neq i} \frac{g_{jk}(i)}{g_{jk}} \tag{4}$$

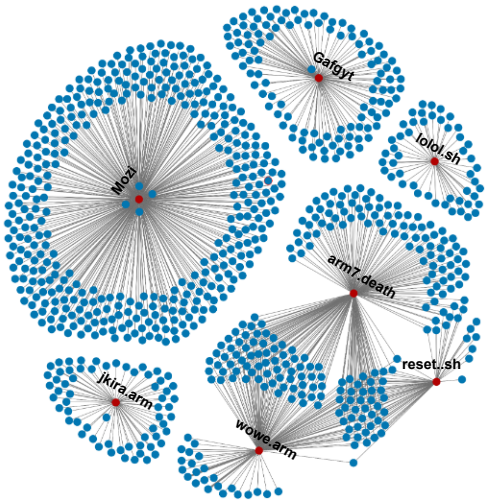


Fig. 1: Connectivity graph of the top botnet loaders instrumenting specific malware strains (red) with their corresponding bots (blue).

where the sum is performed on all pairs of nodes j and k distinct from i and from each other, $g_{jk}(i)$ indicates the number of shortest paths connecting jk passing through i, and g_{jk} indicates the total number of shortest paths from vertex j to vertex k. Hence, the contribution of the pair (j, k) to the betweenness of i is 1, if all shortest paths between j and k pass through i. The contribution take a zero value if no shortest path between j and k pass through i.

Closeness centrality: of vertex i is defined as the mean distance from vertex i to every other reachable vertex.

The closeness centrality of node i in graph G is given by:

$$C_C(v) = \frac{1}{\sum_{v \neq u} d(v, u)} \tag{5}$$

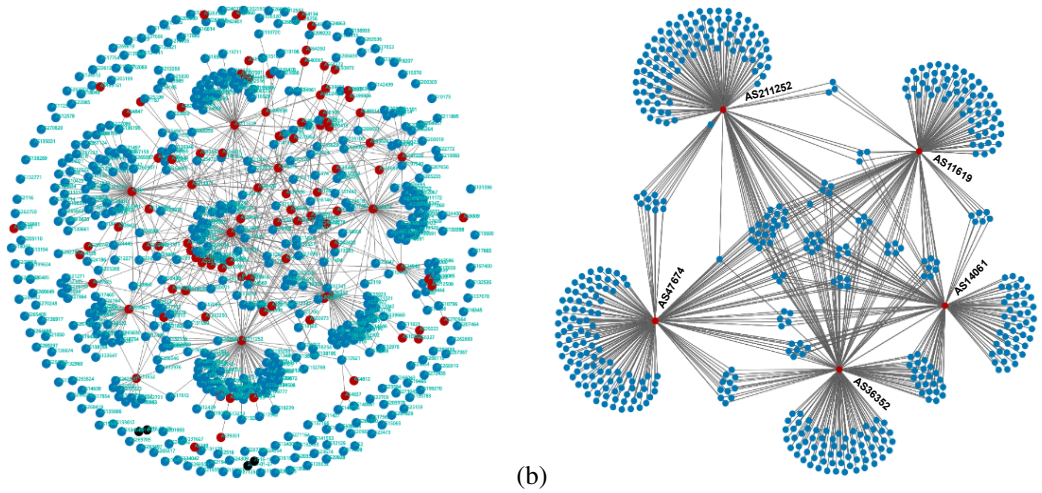
Local clustering coefficient (LCC): of a node i, and l_v is the number of edges among neighbors of v where d_v is the number of neighbors to node v.

$$LCC_v = \frac{2l_v}{d_v(d_v - 1)} \tag{6}$$

Therefore, $LCC = 0$ if none of the neighbors of a node v are connected and 1 if all of the neighbors are connected.

IV. Results

Fig. 1 depicts the connectivity graph of the top seven nodes with high betweenness degree. High betweenness centrality nodes are often gateway nodes or nodes bridging different clusters in a network. The removal of such nodes can cause the botnet network to become partitioned and the betweenness feature in the graph reflects the significant extent of botnet loaders. By ranking all the nodes in the graph based on their betweenness centrality, we identify that botnet loaders responsible for distributing Mozi binaries has the highest betweenness degree. Measuring the degree centrality of Mozi malware servers led to identify 3954 different bots connected to a single server. Our tracking process also reveals that bots



(a) (b)

Fig. 2: (a): The connectivity among ASes embracing bots and ASes hosting botnet loaders. (b): Network topologies for ASes that have a high betweenness centrality, suggesting that nodes identified by centrality metrics are more effective at spreading malicious content throughout the Internet

related to Mozi variants are spread across 125 ASes and 41 countries. Hence, it indicates that Mozi botnet designers spread as much as possible in order to avoid single points of failure and thus increase the botnet's resilience. In addition, our analysis for botnet loaders responsible for distributing arm7.deathh binaries shows that such loaders have an average connection with 270 bots distributed over 18 countries. It indicates that the attackers tend to form a P2P botnet network that is geographically widespread and through its distributed nature across the global Internet could have higher guarantees in terms of its resilience. By analysing the betweenness centrality of nodes that connected to such servers it was revealed that bots are also instructed by attackers to download binaries from three different botnet loaders such as wowe.arm and reset.sh (Fig. 1). Our analysis shows that bots randomly form P2P like networks and direct to download binaries from different sources in order to promote redundancy and increase the robustness of their formation with additional edges.

As summarised in Table III, we have detected 16,473 edges forming the communication setup between bots and loaders in P2P botnets. The identified botnet loaders composing P2P

number of links	mean	std	min	max
16473	0.000116	0.003165	0.000061	0.40238

TABLE III. Summary of centrality degree for the connectivity between botnet loaders and malicious bots.

botnet networks are distributed over 1,011 ASes and 80 countries. Measuring the degree centrality among ASes embracing bots and ASes hosting botnet loaders can reveal the structural properties of a given botnet. Our analysis shows that ASes with the highest degree of centrality are reachable by bots residing in more than 100 ASes. For instance, AS211252 exchanges the binaries with bots distributed across 160 ASes. It therefore demonstrates that such botnet loaders have a strong influence on malware spreading across the Internet.

As shown in Fig. 2(a), some ASes do not have edges, as the bots download the binaries from botnet loaders located within their home AS. Our analysis also highlights that certain malware residing in specific ASes promote communication with bots that are globally distributed. For example, AS47674 hosts four different botnet loaders and demonstrates connectivity with bots in 25 different countries. Fig. 2(b) shows the 5 top ASes that have a high betweenness degree. Evidently, such a high degree can be used as an indicator for identifying nodes that are effective at spreading malicious content throughout the network. Via analysing the betweenness centrality for such ASes, we found that 70% of ASes have a 0 betweenness value as shown in Fig. 3. Hence, malicious bots target nearby vulnerable devices to interact and download malicious binaries from a botnet loader that is co-located within the same AS. For example, AS17622 has a 0 betweenness value since bots interact with botnet loaders within its own domain. We argue that botmasters identify ASes with weak routing policies to achieve this, and they adopt such behaviour to hide the visibility of their botnet's traffic over the Internet in general.

The outcome of degree centrality analysis for loaders and bots indicates that a small number of loaders have influence over the botnet network as shown in Fig. 4. Evidently, malicious actors adopt such behaviour in order to hide the presence of botnet loaders and evade detection. The distribution of cluster coefficients in Fig. 5 shows that the majority of botnet loaders have a clustering coefficient value between 0 and 0.2. Such values indicate that botmasters tend to connect compromised machines with a few number of botnet loaders. However, the LCC degree implies that some bots exhibit different behaviour by having a connection with multiple loaders. Botmasters deploy such architectures to avoid a single point of failure and in parallel assume that inter-AS collaboration is not present to entirely track their critical loaders. Moreover, our analysis of the closeness centrality among botnet loaders and bots shows that approximately 4% of botnet loaders have a

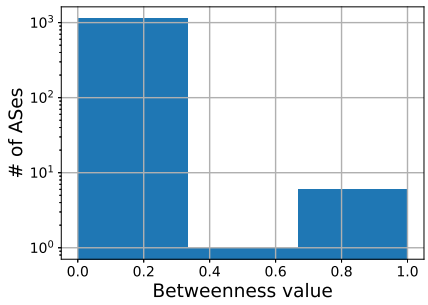


Fig. 3: Normalized betweenness value of ASes hosting botnet loaders; a 70% of ASes have a zero value, as they tend to perform intra-AS traffic routing to bots.

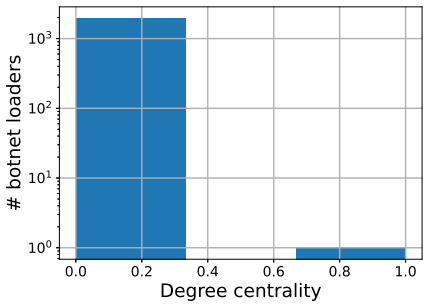


Fig. 4: Normalized degree centrality of botnet loaders, where only 27% of loaders have a centrality degree > 0, and plays a critical role in spreading malicious binaries.

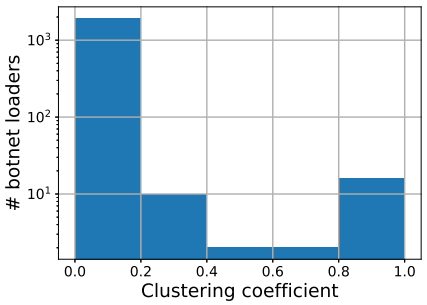


Fig. 5: Clustering coefficient distribution indicates that 90% of botnet loaders have a connection with bots that do not have a relationship with other loaders.

high closeness degree (e.g. $0.4 > C_C < 1$) as depicted in Fig. 6. It indicates that given nodes have close links with several other nodes. Thus, detecting these nodes by defenders will effectively aid in reducing the propagation of botnet, as they have a significant impact on the botnet network.

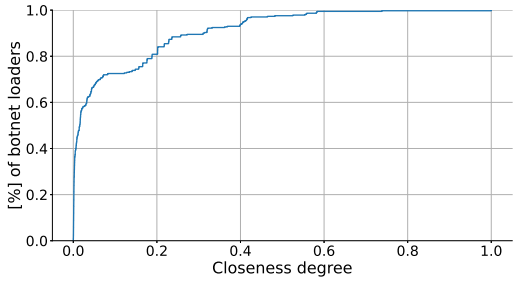


Fig. 6: Cumulative distribution of closeness centrality degree shows that a small proportion of botnet loaders have a high closeness degree.

V. Conclusion

Botmasters have moved towards the full deployment and maintenance of P2P IoT botnets that enable the composition of new large-scale attack vectors with improved resiliency. The propagation and detection evasion of such botnets is strongly related to the design properties of botnet loaders since they constitute the basis for directing compromised devices to download malware strains dictating a botnet's attack vector. In this work, we provide a novel measurement study with the use of real Internet measurement feeds captured for a period of 24 months. Through a graph-based methodology we capture the propagation characteristics and attribute attack strategies through tracking the behaviour of IoT P2P botnet loaders. Through quantitative graph-based metrics we demonstrate that botnet loaders in some instances communicate with bots that are distributed over 25 countries and some botnets tend to conduct all their malware downloading instrumentation within a single AS. In general, we argue that the proposed methodology can act as a cornerstone to assist legal and cybersecurity entities to track and detect botnets. Thus, aid towards the prevention of large-scale and highly evolving attack vectors.

VI. Acknowledgement

The authors would like to thank Okta, Auth0, Max Mind and RIPEstat for providing their datasets. This work has been supported in part by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, which has been funded by the UK EPSRC under grant number EP/S035362/1.

References

[1] Hatem A Almazarqi, Angelos K Marnerides, Troy Mursch, Mathew Woodyard, and Dimitrios Pezaros. Profiling iot botnet activity in the wild. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2021.

[2] Sridhar Venkatesan, Massimiliano Albanese, George Cybenko, and Sushil Jajodia. A moving target defense approach to disrupting stealthy botnets. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pages 37–46, 2016.

[3] Mohammad Alauthaman, Nauman Aslam, Li Zhang, Rafe Alasem, and M Alamgir Hossain. A p2p botnet detection scheme based on decision tree and adaptive multilayer neural networks. *Neural Computing and Applications*, 29(11):991–1004, 2018.

[4] Yaoyao Shang, Shuangmao Yang, and Wei Wang. Botnet detection with hybrid analysis on flow based and graph based features of network traffic. In *International Conference on Cloud Computing and Security*, pages 612–621. Springer, 2018.

[5] Shankar Karuppayah. *Advanced Monitoring in P2P Botnets: A Dual Perspective*. Springer, 2018.

[6] Tirthankar Sengupta, Sanghamitra De, and Indrajit Banerjee. A closeness centrality based p2p botnet detection approach using deep learning. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2021.

[7] Harshvardhan P Joshi and Rudra Dutta. A reinforcement approach for detecting p2p botnet communities in dynamic communication graphs. *arXiv preprint arXiv:2203.12793*, 2022.

[8] Qiben Yan, Yao Zheng, Tingting Jiang, Wenjing Lou, and Y Thomas Hou. Peerclean: Unveiling peer-to-peer botnets through dynamic group behavior analysis. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 316–324. IEEE, 2015.

[9] Sudipta Chowdhury, Mojtaba Khanzadeh, Ravi Akula, Fangyan Zhang, Song Zhang, Hugh Medal, Mohammad Marufuzzaman, and Linkan Bian. Botnet detection using graph-based feature clustering. *Journal of Big Data*, 4(1):1–23, 2017.

[10] Harshvardhan P Joshi and Rudra Dutta. Identifying p2p communities in network traffic using measures of community connections: Ieee cns 20 poster. In *2020 IEEE Conference on Communications and Network Security (CNS)*, pages 1–2. IEEE, 2020.

[11] Mark EJ Newman, Stephanie Forrest, and Justin Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.

[12] Dohoon Kim. Potential risk analysis method for malware distribution networks. *IEEE Access*, 7:185157–185167, 2019.