Detection of Botnets Using Network Graph Patterns Analysis

Agatha Tatianingseto - 13524008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Genesha 10 Bandung
E-mail: atia.seto@gmail.com, 13524008@std.stei.itb.ac.id

***Abstract*– Botnets, networks of compromised devices controlled by attackers, are difficult to detect due to their diverse architectures and behaviors. This paper applies graph theory to model network traffic from the CTU-13 dataset, focusing on Scenario 12. By analyzing in-degree, out-degree, and data transfer weights, the research identifies structural patterns typical of botnet activity. Findings reveal centralized communication with star-like topologies and consistent traffic volumes among bots, suggesting a Command-and-Control (C&C) model despite the scenario's P2P label. Graph-based analysis proves effective in highlighting botnet behavior through visual and statistical patterns.**

***Keywords*– Botnet, Graph Theory, Anomaly Detection, Command-and-Control (C&C), Peer-to-Peer (P2P) Botnet**

I. INTRODUCTION

As technology becomes more integrated into daily life, from phones to smart appliances, cybersecurity risks have grown. Many people are unaware that their personal devices, like laptops or smart fridges, can be unknowingly hijacked by cybercriminals. This threat often takes the form of a botnet: a network of malware-infected devices secretly controlled by an attacker, or bot herder. These compromised devices work together to carry out large-scale malicious activities such as DDoS attacks, data theft, or spam distribution [1].

Understanding and preventing botnets is crucial in today's digital landscape. Its attacks are hazardous because they are often persistent and difficult to fully eliminate, especially when multiple devices across a network have been compromised. Even if one infected device is removed, the attacker can still control others, often using encrypted communication to avoid detection. A bot-master can take over thousands of devices at once, using them to launch large-scale attacks or sell access to other criminals. For organizations, falling victim to a botnet can result in serious consequences such as financial loss, data breaches, and service disruptions. These attacks place a heavy burden on security teams and can significantly damage business operations [2].

Botnets are difficult to detect due to their unique structures and behaviors—there's no single template for how they operate. They may use centralized or peer-to-peer architectures and spread via malicious ads, infected emails, or other vectors. Their purposes vary, from launching DDoS attacks to mining cryptocurrency. Unpatched software vulnerabilities make it easy for botnets to infiltrate systems, especially as many users delay updates. IoT devices are particularly vulnerable due to weaker security and infrequent patching, as seen in the widespread URGENT/11 vulnerabilities that left billions of devices exposed long after fixes were available [3].

Graph theory is a powerful tool for detecting botnet activity, especially within large-scale IoT environments. By modeling network traffic as graphs—where devices are nodes and their interactions are edges—researchers can uncover hidden patterns of communication that suggest coordinated or malicious behavior. For example, devices participating in a botnet often form tightly connected clusters or exhibit synchronized traffic to the same destination. These visual patterns can be difficult to spot in raw traffic logs but become more apparent through graph analysis. This structural approach allows for quicker identification of threats and deeper insights into botnet behavior [4].

This paper explores the use of graph theory in identifying abnormal connection patterns associated with botnet activities.

II. THEORY

Understanding botnet detection begins with examining botnet architectures and applying graph theory to model their communication. Botnets can be centralized, decentralized (P2P), or hybrid, each with unique control structures. Graph theory helps represent network traffic, where nodes are devices and edges are communication links. This approach enables the identification of suspicious patterns based on the structure and intensity of connections. The following sections outline key botnet types and relevant graph concepts for detection.

A. Botnet Architecture Types

A botnet is a network of compromised devices (bots) controlled by an attacker (botmaster) to perform malicious tasks such as spamming, DDoS attacks, or data theft. The way these bots communicate with the controller defines the botnet architecture.

Centralized botnets (C&C servers)

A distinguishing feature of botnets, compared to other internet malware, lies in their control communication architecture. Traditionally, most botnets have adopted a centralized model, where infected devices (bots) connect directly to one or more Command-and-Control (C&C) servers. These servers act as intermediaries, receiving commands from the botmaster and distributing them to the bots across the network. This structure, often referred to as a C&C botnet, has been the focus of extensive research, particularly those using Internet Relay Chat (IRC) as the communication protocol. The simplicity and direct communication flow of this model make it relatively easy to manage from the attacker's perspective [5].

However, from a defensive standpoint, this architecture presents critical vulnerabilities. The C&C servers form a single point of failure—if identified and taken down, the entire botnet can be dismantled. Defenders can also extract the C&C server's identity by analyzing network traffic or capturing a single bot that contains server details. Moreover, hijacking a C&C server could potentially expose the entire botnet infrastructure. These weaknesses have driven the evolution of botnet architectures, as attackers seek to develop more resilient and covert communication models to evade detection and takedown efforts [5].

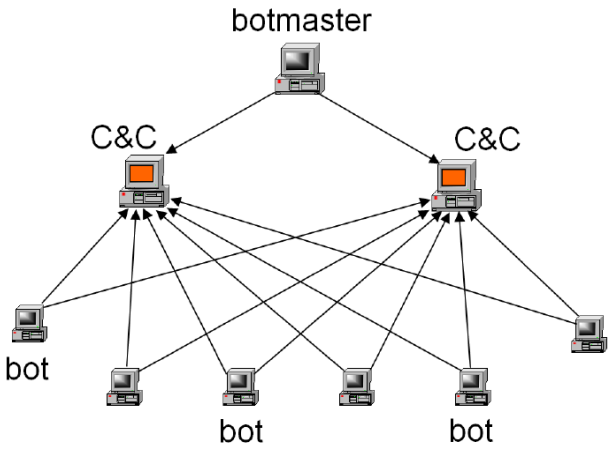

Figure 2.1 Command and control architecture of a C&C botnet
(Taken from usenix.org)

Decentralized (P2P) Models

Peer-to-Peer (P2P) botnets represent a decentralized form of botnet architecture in which each infected device (bot) functions both as a client and a server. Unlike centralized models, P2P botnets do not rely on a single command-and-control (C&C) server; instead, bots exchange data and instructions directly with one another. This decentralized nature enhances their resilience, making them less vulnerable to takedowns by law enforcement. While setting up a P2P botnet is technically more challenging than configuring traditional IRC or HTTP-based botnets, attackers benefit from reduced dependency on static infrastructure and maintain full control over malicious operations through the distributed network [6,7].

The transition to P2P botnets has been driven by the increasing effectiveness of global efforts to dismantle centralized C&C servers. In response, attackers have adopted P2P communication models to avoid detection and disruption. Bots typically discover peers by scanning random IP addresses, exchanging lists of known infected hosts, and relaying updates or commands through the network. This design allows the botmaster to manage the botnet indirectly while minimizing exposure and increasing persistence against defensive measures [8].

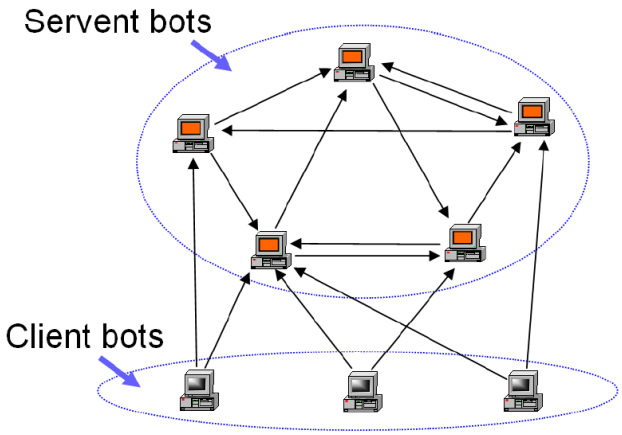

Figure 2.2 Command and control architecture of the proposed hybrid P2P botnet
(Taken from usenix.org)

Hybrid botnets

Hybrid botnets integrate both centralized and peer-to-peer (P2P) communication models to improve flexibility and resilience [7]. Typically, bots connect to a central command-and-control (C&C) server for instructions, but can also switch to P2P communication if the server is taken down. This architecture allows attackers to maintain control and operational continuity even when part of the botnet infrastructure is disrupted. By combining the simplicity of centralized models with the robustness of P2P systems, hybrid botnets are harder to detect and dismantle. Notable examples such as Conficker and some variants of Mirai exhibit hybrid characteristics, making them more adaptive and persistent in hostile environments.

B. Graph Theory Basics

To analyze and detect botnet behavior within network traffic, graph theory offers a powerful and structured

approach. In this context, a network can be modeled as a graph, where nodes (vertices) represent devices such as IP addresses or hosts, and edges represent communication flows like packets or message exchanges. This representation helps identify unusual patterns and relationships among devices, which may indicate malicious activity. The following section explains key components used in graph-based analysis, including vertices, edges, directed and undirected graphs, and weighted graphs.

Graph Nodes and Edges

- Node (or Vertex): A node represents an individual point in a graph, such as a person, device, city, or website. In a diagram, nodes are typically labeled (e.g., A, B, C, D, E, F) and serve as the entities being connected [9].
- Edge: An edge is the link or connection between two nodes, representing relationships or interactions such as friendships, roads, or data transfers. In a graph diagram, edges are shown as lines connecting the nodes [9].

Graph Types

- Directed Pair: A directed pair represents a connection between two nodes, u and v, written as (u, v), where the direction of the connection matters. In this case, (u, v) is different from (v, u), indicating that the edge goes specifically from node u to node v. This is typically used in directed graphs to show the direction of communication or flow.
- Undirected Pair: In an undirected pair, the connection between nodes u and v does not have a direction, meaning (u, v) is considered the same as (v, u). The order of the nodes doesn't matter, and this form is used in undirected graphs, where relationships are mutual.

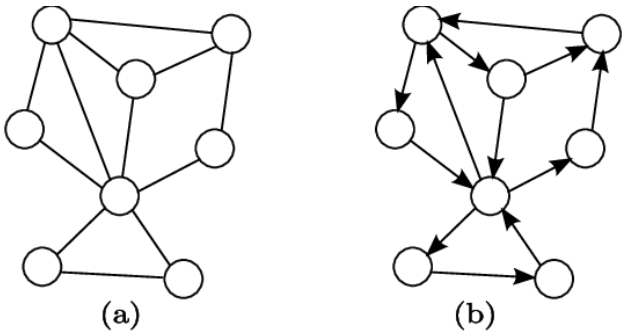

Figure 2.3 (a) Undirected Graph, (b) Directed Graph
(Taken from ResearchGate)

- Weighted Graph: A weighted graph is a graph—either directed or undirected—where each edge carries a numerical value called a weight. These weights usually represent meaningful quantities such as distance, cost, time, or capacity between connected nodes.
- Unweighted Graph: An unweighted graph is one where edges do not carry any specific value. All connections are treated equally, indicating only the presence or absence of a relationship, without assigning any priority or cost to the edges.

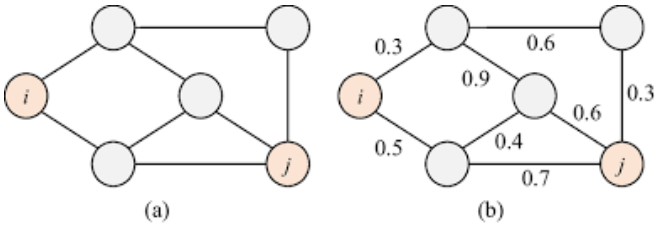

Figure 2.4 (a) Unweighted Graph, (b) Weighted Graph
(Taken from ResearchGate)

III. GRAPH FEATURE FOR DETECTION

In-Degree

In a network graph, in-degree refers to how many other devices are sending data or making connections to a specific device. If many suspected infected devices (bots) are contacting a certain server—such as a command-and-control (C&C) server—that server will have a high in-degree. A higher in-degree means that the device is being contacted frequently, which could be a sign that it is part of a botnet, especially if many bots are trying to reach it at once.

Out-Degree

Out-degree is the number of connections a device makes to other devices. A high out-degree means the device is trying to talk to many others, which can be suspicious behavior, especially if the device is sending data to many IP addresses. Bots often do this to spread the infection or to report information back to the attacker. So, a high out-degree could be a clue that the device is acting as part of a botnet.

In-Degree Weight

In-degree weight counts how much data a device is receiving from others, not just how many connections. In botnets, infected devices may receive commands or updates from the attacker or other bots. If several devices receive similar amounts and types of data, they may be part of the same botnet. This helps identify suspicious communication patterns in the network.

Out-Degree Weight

Out-degree weight measures how much data a device is sending out to others. Bots often send out information, such as stolen data or messages, to other infected machines. If a device is sending large or similar amounts of data to many destinations, this behavior might indicate it's a bot. Looking at both the number of connections and the amount of data sent can help spot these threats.

IV. DATASET

This paper uses the CTU-13 Dataset in its bidirectional network flow (binetflow) format. The CTU-13 is a collection of network traffic data recorded at CTU University in the Czech Republic in 2011. It was designed to provide a comprehensive mix of real botnet activity, normal user behavior, and background traffic. The dataset includes thirteen distinct scenarios, each representing a unique capture where a specific type of malware was executed. These scenarios vary in terms of the protocols used and the types of malicious actions performed, offering a diverse and realistic set of botnet traffic for analysis and research.

Each scenario was originally captured in a PCAP file, which contains all the packets related to the three types of traffic. For analysis purposes, the traffic is also provided in binetflow format, where each record summarizes a bidirectional network flow. Each flow includes 15 data attributes, such as StartTime, Duration, Protocol (Proto), Source IP Address (SrcAddr), Source Port Number (Sport), Direction (Dir), Destination IP Address (DstAddr), Destination Port Number (Dport), Connection State (State), Source ToS (sTos), Destination ToS (dTos), Total Packets (TotPkts), Total Bytes (TotBytes), Source Bytes (SrcBytes), and Label, which classifies the flow as Botnet, Normal, or Background.

This research specifically utilizes Scenario 12 of the CTU-13 dataset. The labeled flows in this scenario enable the construction of communication graphs, where nodes represent IP addresses and edges represent the communication flows. These graph representations allow the extraction of structural features relevant for identifying botnet behavior.

V. DATA PRE-PROCESSING

Label-Based Filtering

In the initial stage of data preprocessing, only records that contain the keyword “Botnet” in the Label attribute were selected. This filtering step is crucial to ensure that the analysis is focused solely on traffic that is known to be associated with botnet activity, thereby excluding irrelevant or benign flows such as normal user behavior and background noise. By narrowing the dataset to only malicious instances, the resulting graph structure and its analysis can better reflect the communication patterns and behaviors typical of botnet-infected systems.

Time Window Selection

To further refine the dataset and reduce complexity, a specific time window from 10:53:00 to 11:05:00 was chosen. This period was selected based on known intervals of botnet activity in the scenario, and serves as a representative sample for analysis. Limiting the data to this time frame allows for more focused graph construction and clearer visualization of interaction patterns. It also ensures that the graph reflects a concentrated burst of activity, which is often useful when identifying behavioral characteristics of bots, such as data exfiltration or command execution.

Directed-based Segmentation

The dataset was then segmented based on the Dir (direction) attribute into two distinct categories: outgoing connections (->) and incoming connections (<-) or bidirectional (<->). This segmentation enables a clearer understanding of how each node behaves in the network—whether it is primarily sending data, receiving data, or both. This step is especially important in graph-based analysis, as it directly impacts the calculation of key features such as in-degree (number of incoming connections) and out-degree (number of outgoing connections). By separating these flows, the resulting graphs more accurately represent directional communication patterns, which are critical for detecting centralized command structures or peer-to-peer interactions within a botnet.

VI. GRAPH

In this stage, the processed dataset is transformed into a graph structure using the Python programming language, specifically with the help of the NetworkX library. The goal is to model network interactions as directed graphs that reflect communication behavior during botnet activity. The following explanation focuses on the conceptual approach to constructing the graph from the network flow data.

Graph Construction

In the graph construction phase, each network flow from the filtered dataset is converted into a directed graph representation using NetworkX. In this graph, nodes represent IP addresses involved in communication, while directed edges represent the flow of data between these IPs. For outgoing flows, edges are constructed from the botnet IPs (as source) to their respective target IPs (as destination), reflecting command propagation or attack attempts. For incoming flows, the direction is reversed, with edges pointing from external IPs to the botnet IPs, representing control commands or data uploads. Each edge is assigned a weight based on the TotBytes attribute, capturing the volume of data transferred between nodes.

TotBytes stands for Total Bytes, representing the total amount of data (in bytes) exchanged during a single network flow between a source IP and a destination IP. It includes the bot payload and metadata transferred in that session. In this paper, TotBytes is used as the edge weight to reflect the intensity of communication between nodes, allowing the graph to capture not just the existence of a connection but also its significance. This helps highlight abnormal patterns of data transfer often associated with botnet behavior, such as large volumes of outbound data or synchronized communication sizes among infected hosts.

To enhance interpretability, the graph visualization features color-coded nodes that correspond to their roles within the network. Known botnet IP addresses are highlighted in red, indicating malicious actors within the communication graph. In contrast, all other IP addresses, typically representing victims, background traffic, or normal users, are shown in sky blue. This color distinction helps to visually isolate the botnet nodes, making it easier to observe their communication patterns and centrality within the graph structure

Graph Visualization

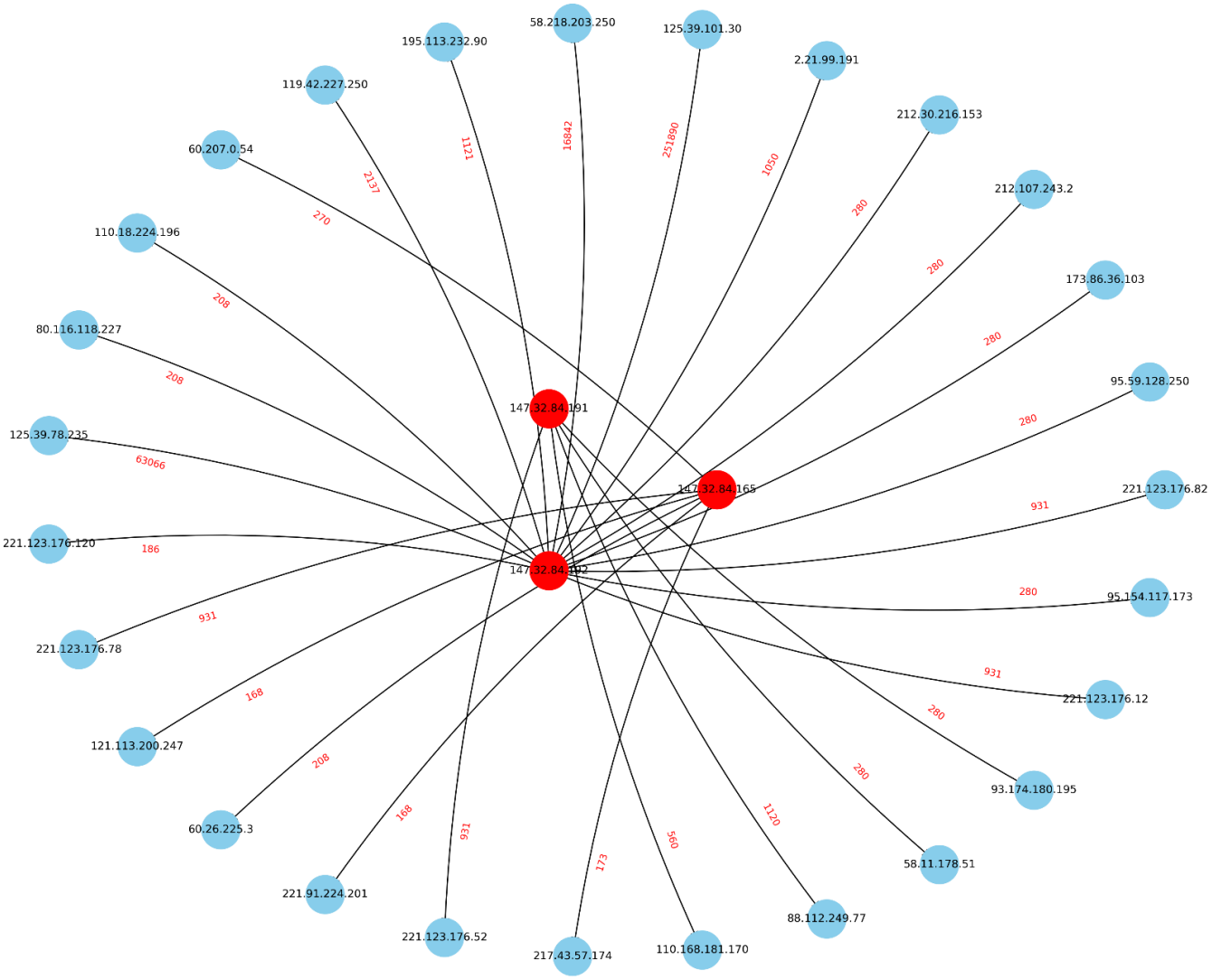

Figure 6.1 Out-Degree for Botnet Activities

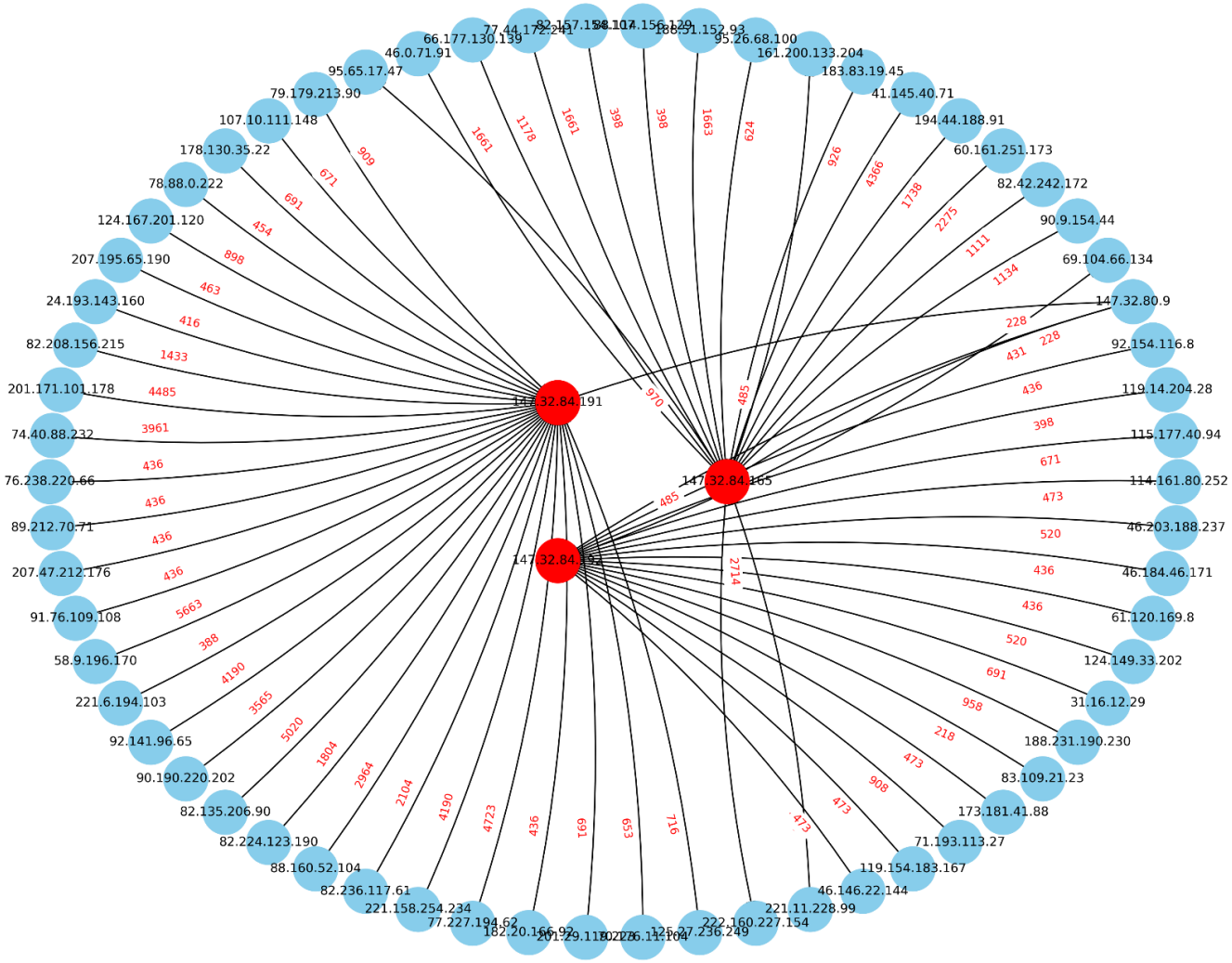

Figure 6.2 In-Degree for Botnet Activities

Botnet Graph Analysis

Figure 6.1 presents a sample result of the graph visualization within the selected time window from 10:53:00 to 11:05:00, illustrating the outgoing communication (out-degree) from botnet nodes to other IP addresses. Figure 6.2 is typically the same as Figure 6.1, but it represents the incoming communication (in-degree) from the other IP addresses to the botnet. The number of in-degree, out-degree, and edge weight for scenario 12 details are represented in figure 6.3 and figure 6.4.

Based on the structure of the outgoing communication graph, where botnet nodes are shown in red, we can see that these nodes send data to many different IP addresses but do not communicate with each other. This creates a pattern that looks like a star, with each bot connecting outward to several targets. This kind of setup is typical of a Command and Control (C&C) botnet, where each infected computer (bot) is controlled by a central server. Since the bots are not talking to one another, it's clear that this is not a Peer-to-Peer (P2P) botnet, which would normally involve bots sharing information in a more equal way.

In the incoming graph, many external IP addresses are seen communicating with the botnet nodes. The arrows between them indicate two-way communication, but the bots mostly appear to be on the receiving end of these connections. This pattern suggests that the bots are likely being contacted by central servers or other infected machines to receive commands or updates. Although there is some return traffic, the structure still follows a Command and Control (C&C) model, where bots are controlled by a central entity. There is no visible communication between the bots themselves, which means this is not a Peer-to-Peer (P2P) botnet.

By observing both graphs presented in Figure 6.1 and Figure 6.2, it can be seen that the number of incoming

connections (in-degree) to the botnet nodes is much higher than the outgoing connections (out-degree). This suggests that the botnet nodes (shown in red) are receiving far more communication from external IP addresses than they are initiating. Such a pattern typically indicates a centralized communication model, where bots receive commands or data from controlling entities. However, according to the official description of the CTU-13 dataset (available at Stratosphere IPS [10]), Scenario 12 is designed to demonstrate a Peer-to-Peer (P2P) botnet, in which infected machines (bots) are expected to communicate directly with one another. This expected behavior is not reflected in the graphs, as the red nodes do not appear to be connected. A possible explanation for this difference is the time window selection used during the graph construction. The chosen time range may only capture interactions between the botnet and external IPs, and not the internal communication between the bot nodes themselves, which could occur outside of the selected interval.

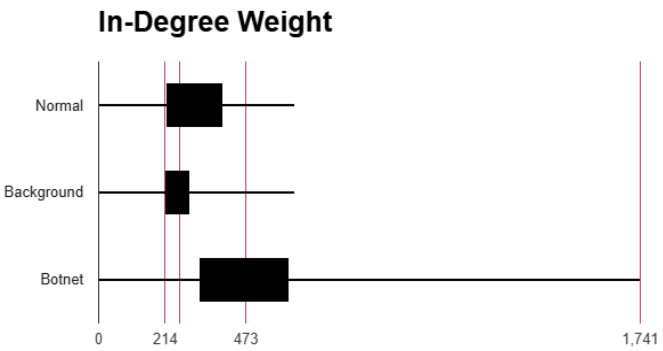

Figure 6.3 In-Degree Weight BoxPlot

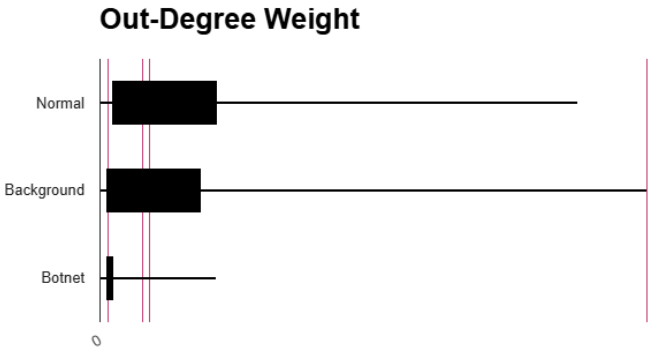

Figure 6.4 Out-Degree Weight BoxPlot

	Normal		Background		Botnet	
	In-coming	**Out-going**	**In-coming**	**Out-going**	**In-coming**	**Out-going**
Sample Size	4566	2548	198766	74734	866	935
Minimum	0	0	0	0	0	0
Q1	218	412	214	194	323	208
Median	260	1717	214	1490	473	280
Q1	400	4066	292	3509	613	488
Maximum	629	16522	629	18911	1741	4026
Mean	298.13053	2957.408948	266.025905	2828.458172	518.89261	614.693048

Table 6.1 Out-Weight and In-Weight distribution

Analysis of Botnet Communication Patterns

This section explains how the structural properties of the graph and the statistical distribution of in-degree and out-degree values help us understand botnet behavior. Two main graph-based metrics were used to study and compare communication patterns across different types of network traffic: in-degree weight and out-degree weight, both measured using the TotBytes attribute from the CTU-13 dataset. These metrics represent the total amount of data (in bytes) either received (in-degree) or sent (out-degree) by each IP address. For each traffic type—Normal, Background, and Botnet—boxplots were generated to visualize the distribution of TotBytes, and a summary table was created including

sample size, minimum, first quartile (Q1), median, third quartile (Q3), maximum, and mean.

The table shows that Background traffic has the largest number of records, with over 198,000 incoming and 74,000 outgoing connections, and generally low data sizes. Normal traffic shows a broader range, with some very large data transfers, especially in the outgoing direction. Botnet traffic, on the other hand, involves fewer connections (866 incoming and 935 outgoing), but the amount of data received is higher and more consistent. This pattern suggests that botnet devices may be receiving similarly sized data repeatedly, indicating structured or automated communication rather than random behavior.

The characteristics of botnet communication in Scenario 12 align with those of a centralized Command and Control (C&C) architecture. Graph visualizations and boxplot analysis show that botnet nodes have significantly higher in-degree and out-degree weights compared to normal and background traffic. The average in-degree weight for botnet traffic is 518.89 bytes, which is noticeably higher than normal (298.13 bytes) and background (266.02 bytes) traffic. Although the average out-degree weight for botnet traffic (614.69 bytes) is lower than its in-degree, it still surpasses background traffic. These values suggest regular, possibly coordinated exchanges, such as commands received from and responses sent to control servers.

Notably, botnet traffic shows a tighter distribution in TotBytes compared to normal and background categories, as reflected in the median and interquartile range (IQR). This consistency implies that bot communication is not only heavier but also more uniform, which could indicate the use of automated scripts or synchronized instruction sets. The outgoing communication graph supports this conclusion: botnet nodes (in red) form a star-like topology with multiple, uniform connections to other IPs, fitting the structure of a typical C&C botnet.

The number of active botnet nodes also plays a role in shaping the observed communication pattern. In Scenario 12, three botnet IPs—147.32.84.165, 147.32.84.191, and 147.32.84.192—participate in the communication graphs. Each initiates connections with multiple destination IPs and receives data from various sources. Among them, 147.32.84.165 shows slightly higher activity in terms of connection count and total bytes sent, though the difference is not extreme. The relatively even out-degree weights across the three bots, as seen in the boxplot, suggest a structured and synchronized operation. Importantly, there are no connections between the bot nodes themselves in either graph, reinforcing that the communication model is centralized rather than peer-to-peer (P2P). The consistent traffic size and structure across the three bots further confirm that they operate independently under a common control mechanism, scaling communication without introducing randomness.

VII. CONCLUSION

Graph-based analysis effectively detects botnet behavior by modeling network flows as directed graphs and analyzing structural features such as in-degree, out-degree, and data volume. Applied to Scenario 12 of the CTU-13 dataset, the approach revealed centralized communication patterns, with botnet nodes exhibiting high and consistent in-degree values and star-like topologies, typical of Command-and-Control (C&C) structures. Although the scenario is labeled as peer-to-peer (P2P), the lack of inter-bot communication suggests that the time window captured only external interactions. Statistical analysis further supports the presence of synchronized and structured data flows in botnet traffic. This method proves valuable for identifying coordinated malicious activity and can be extended to detect more complex or evolving botnet architectures.

VIII. ACKNOWLEDGMENT

The author would like to express deep appreciation to God for the guidance, strength, and blessings that made the learning journey and the completion of this paper possible. The author is also grateful to the lecturers of the Discrete Mathematics IF1220 course at Institut Teknologi Bandung, Mr. Arrival Dwi Sentosa and Mr. Rinaldi Munir, for their dedication in teaching and their valuable insights throughout the semester. Special thanks are also extended to the author's family and friends for their ongoing support, motivation, and encouragement during this academic period.

APPENDIX

This is the source link containing the full code used for data preprocessing, graph construction, and visualization, as well as a video explanation of the paper:
https://github.com/Agatha936/Graph-Construction---Botnet-Detection-CTU-13-Scenario-12-.git

REFERENCE

[1] Palo Alto Networks, “What is a botnet?” Cyberpedia. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-botnet#:~:text=A%20botnet%20(short%20for%20%E2%80%9Crobot,known%20as%20a%20bot%20herder. [Accessed: June 17, 2025].
[2] Darktrace, “Botnet - Cyber AI Glossary.” [Online]. Available: https://www.darktrace.com/cyber-ai-glossary/botnet?utm_source=chatgpt.com. [Accessed: June 17, 2025].
[3] DataDome, “How to detect & mitigate botnets.” [Online]. Available: https://datadome.co/learning-center/how-to-detect-mitigate-botnets/. [Accessed: June 17, 2025].

[4] T. Høiland-Jørgensen, “Identifying malicious IoT botnet activity using graph theory,” APNIC Blog, Jul. 16, 2020. [Online]. Available: https://blog.apnic.net/2020/07/16/identifying-malicious-iot-botnet-activity-using-graph-theory/. [Accessed: June 17, 2025].
[5] P. Wang, S. Sparks, and C. Zou, “An advanced hybrid peer-to-peer botnet,” in Proc. USENIX Workshop on Hot Topics in Understanding Botnets (HotBots), 2007. [Online]. Available: https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/wang/wang_html/. [Accessed: June 18, 2025].
[6] Spiceworks, “What is a botnet?” [Online]. Available: https://www.spiceworks.com/it-security/network-security/articles/what-is-botnet/#_002. [Accessed: June 18, 2025].
[7] Palo Alto Networks, “How botnets work,” Cyberpedia. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-botnet#how. [Accessed: June 18, 2025].
[8] Radware, “Botnet,” Cyberpedia. [Online]. Available: https://www.radware.com/cyberpedia/bot-management/botnet/. [Accessed: June 18, 2025].
[9] GeeksforGeeks, “Mathematics | Graph Theory Basics - Set 1.” [Online]. Available: https://www.geeksforgeeks.org/mathematics-graph-theory-basics-set-1/. [Accessed: June 19, 2025].
[10] B. Grill, V. Valeros, and M. Rehak, “CTU-13 dataset: A labeled dataset for flow-based botnet detection,” Stratosphere IPS, 2015. [Online]. Available: https://www.stratosphereips.org/datasets-ctu13#:~:text=The%20CTU%2D13%20is%20a,normal%20traffic%20and%20background%20traffic. [Accessed: June 19, 2025].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Juni 2025

Agatha Tatianingseto - 13524008