Detecting Collaborative Scanners based on Shared Behavioral Features

Andrei-Iulian Vișoiu[1]

Supervisors: Georgios Smaragdakis[1]**, Harm Griffioen**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Andrei-Iulian Vișoiu
Final project course: CSE3000 Research Project
Thesis committee: Georgios Smaragdakis, Harm Griffioen, Kubilay Atasu

Abstract

Port-Scanning is a popular technique that helps detect open ports to connect to on the internet, with both benign and malicious applications. While methods have been developed to detect scans coming from one source, adversaries have started to distribute their scans among multiple hosts. Detecting these collaborative scanners, however, is a difficult task, and no established method has emerged yet. In this paper, we propose a method to cluster scanning groups from network telescope data based on the assumption that scanners working together will have similar behavioral features. An evaluation framework based on address space coverage and overlap is introduced and the performance of two clustering algorithms, HDBSCAN and DBSCAN, is compared. A clear indication was found that only relying on behavioral features and applying unsupervised clustering techniques can result in incomplete groups or clusters with more than one group inside, with HDBSCAN generally performing better than DBSCAN. The effectiveness of the evaluation framework is also discussed, as some of the identified groups indicated overlaps, but manual analysis reveals that they have the potential to belong to a single party.

1 Introduction

Internet-Wide Scanning (also called Port Scanning) is a widespread method that has many applications. It works by sending probe packets to the full public IPv4 address range and monitoring the responses. Tools that can scan the entire internet in less than an hour, such as ZMap [1], are readily and easily available to any party on the web.

This technique has many security applications: it was used in the past to research the depletion of the allocated IPv4 address space [2] and to assess the impact of newly-discovered vulnerabilities. The most popular example is Heartbleed, a severe vulnerability in the OpenSSL implementation, that was quickly analysed using fast port scanning [3].

Alternatively, scanning can also be used with malicious intent. It can provide an overview of the active hosts on the internet and the services available on these hosts. Therefore, it can be a useful tool during the "Reconaissance" stage of a cyber attack [4], during which the adversary decides on the targets and the methods to use.

Countermeasures have been employed in response to the emerging threat. Certain firewall rules can block scanning traffic exceeding set thresholds. This provides good protection against aggressive scanners [5], but more advanced adversaries can now distribute their scans among multiple systems and stay below the threshold on each system.

Adversaries employing silent distributed surveillance would likely posses considerable knowledge and resources and therefore pose significant risk. For defenders to better understand the threat landscape and adapt their countermeasures, it is important that these scanning groups are detected and understood.

While there are multiple proposed methods to tackle the problem, currently no established method has emerged. For example, Robertson et al. [6] proposed a method that detects groups as long as the scans originate from the same /24 subnet. This detection technique can be circumvented by adversaries that distribute their scans across multiple ISPs and countries. Another method proposed by Gates [7] follows a set covering approach and will detect groups that scan more than 95% of the network range. Other approaches, such as the one proposed by Tanaka et. al [8] use packet fingerprints to detect coordinated scans. This identifies groups using the same tool, but offers no way to determine exactly which IPs are working together.

These approaches work well enough for specific types of scanners, but attackers are becoming aware of them and are adapting continuously. In the arms race against internet adversaries, more flexible methods that can preferably adapt to many tools are needed. Threat actors that distribute their scans are inherently more skilled and resourceful than the ones doing single-machine aggressive scans. For this reason, one can assume that they will use a well-established framework of Tactics, Techniques and Procedures (TTPs) [9] that is consistent across their scanning machines.

Therefore, the aim of this paper is to assess the degree to which network telescope data [10], consisting of unsolicited TCP packets, can be used in combination with unsupervised learning techniques to detect collaborative scanning groups by using behavioral features. Therefore, the paper aims to answer the following question:

> *"How can we detect collaborative scanners in network telescope data using clustering algorithms, based on behavioral features?"*

The assumption is that scanners belonging to the same group will exhibit similar scanning patterns and as such, by choosing an appropriate feature set and clustering algorithm, scanning groups can be unveiled. In order to answer the question above, answers to the following sub-questions are needed:

1. What are the most relevant behavioral features to identify a group?
2. What does the group composition look like?
3. How can the performance of such an approach be measured?

The structure of the paper is as follows. The chosen methodology will be presented and justified in Section 2. Section 3 will showcase the experimental setup and the results reached. Section 4 tackles responsible research, focusing on the ethical aspects of the research conducted and the reproductibility of the results. In section 5, the results will be discussed and interpreted, in order to understand the broader context. Section 6 will present the final conclusions of the research and some recommendations for future research.

2 Methodology

This section presents the methods chosen for each step in the research process. Section 2.1 briefly showcases how data was collected. In section 2.2, data aggregation is discussed, presenting the features extracted from the dataset. Section 2.3

introduces the clustering algorithms chosen to perform the experiments, with additional motivation. Section 2.4 details the subsets of features used for the experiments, together with the hyperparameters. Section 2.5 presents the evaluation methodology. Section 2.6 introduces the idea of post-processing the resulting groups using a greedy algorithm in order to improve group quality.

2.1 Data Collection

Secondary data was used to conduct the research, consisting of unsolicited TCP packets captured by a TU Delft network telescope [11]. The number of IPs that listen daily on the telescope is on average $64\,000$, depending on address availability, with spikes in the $110\,000$ range in some periods. The packets were captured throughout between the 1st and 20th February 2024 and have been enriched with additional data, such as the network operator and country. The data is stored in a ClickHouse database hosted on a TU Delft server.
A total of $7\,046\,799\,014$ packets were analysed, corresponding to $1\,668\,141$ distinct source IPs. After removing backscatter traffic and IPs that hit the telescope less than once a day, the remaining number of IPs used to perform the experiments was $921\,846$.

2.2 Pre-Processing: Feature Selection, Aggregation and Scaling

We assume that a typical scanning group would have source IPs from subnets of the same network operator or cloud provider. There is also a chance that the country varies, but the cloud provider is the same. In terms of activity, IPs belonging to the same group should scan at the same inter-packet frequency and on the same number of days. The length of their scans, expressed as the time difference between the first and last packets, should also be similar. Moreover, a scanning group is very likely to use the same toolset across their machines and also the same IP generation algorithm. They should also share similar behavior in terms of source and destination port numbers and also on their most scanned port, as an organised group will scan for a fixed set of vulnerabilities to exploit. An assumption that is useful to test is whether the threat actors divide their scans evenly across machines, or introduce randomness. In the case of uniform distribution, machines in the same group should hit the telescope uniformly in terms of distinct destination IPs and number of packets.
Packet data needs to be aggregated to calculate features that will help in clustering the IPs with the assumptions that were made above. An overview of the extracted aggregated features can be found below.

Subnet The /24 subnet the IP originates from.

Active Days Number The number of days the IP was active.

ASN The assigned number of the autonomous system the IP originates from. Can be interpreted as the network operator.

Country Originating country.

Scan Length Average time difference, in seconds, between the first and last packets seen on a day.

Number	Fingerprint
1	ZMap [1]
2	Masscan [12]
3	Hajime [13]
4	Mozi [14]
5	Atk03
6	Atk05
7	Atk06
8	Surv01
9	Surv02
10	Mirai [15]
11	Unknown

Table 1: Tools for each number. Determined as presented by Tanaka et al. [8]

Median Inter-Packet Time The average of the median time difference between two consecutive packets. Chosen to provide a more accurate estimate of scanning speed, rather than the ratio between packets and time.

Distinct Source Ports Average number of distinct source ports used on a day.

Distinct Destination Ports Average number of distinct destination ports hit on a day.

Distinct IPs Average number of distinct IPs targeted on a day.

Total Hits Average number of packets sent on a day.

Payload Length Average payload length of packets originating from the source IP.

Top Destination Port The port that was the most scanned by the IP in most days. Example: if on days 1-3 the most scanned port is 80, and on day 4 it is 443, this will be 80, irrespective of the number of packets.

Tool Fingerprint Prediction of the tool used, based on fingerprints found by Tanaka et al. [8]. Numbered 1 through 11. Full description can be found in Table 1.

Distinct Fingerprints Average of the number of distinct fingerprints found in the packets originating from the IP.

Quartiles of Consecutive IP Distribution $Q1$, $Q2$ and $Q3$ of the distribution of the difference between each 2 consecutive destination IPs.

Before training any clustering models, the resulting features were scaled to ensure that no feature dominates the distance calculations.

2.3 Choosing a Clustering Algorithm

This subsection shortly explains the clustering algorithms chosen to perform the analysis. An important observation is that the number of clusters (scanning groups) present in the data is unknown. Moreover, the data has a noisy nature. A prospective clustering algorithm for the problem should take both of these prerequisites into account: it should not require the number of clusters to be specified a priori and it should have a concept of noise. For the purpose of this research,

three prospective algorithms were analysed: DBSCAN[16], OPTICS [17] and HDBSCAN [18]. After the analysis, two of them were chosen to be compared. A short presentation for each of them can be found below, followed by the final choice.

DBSCAN
Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [16] works by finding core samples of high density and deriving clusters from them. From a theoretical perspective, this is a strong candidate for a solution. However, its outcome is highly dependent on finding an appropriate value for ϵ, the distance threshold between two points to be considered in the neighbourhood of each other. This can be done either manually by using domain knowledge, or heuristically [19]. Moreover, the resulting clusters will be of similar densities, which the ϵ value influences.

OPTICS
Ordering Points To Identify Cluster Structure [17] is based on the same idea as DBSCAN and tries to deal with the challenge of choosing the ϵ. It does this by introducing the concept of *Reachability Distance*. This can be interpreted as how easy it is to reach a specific points from other points in the dataset, in terms of distance. Points with similar distances will therefore likely be in the same cluster and clusters of varying densities can be found. The ϵ parameter only serves as an upper-bound to improve speed.

HDBSCAN
HDBSCAN [18] is also closely related to DBSCAN and tries to solve the challenge of choosing the right parameter: the ϵ parameter is removed entirely. It performs DBSCAN over a varying ϵ and selects the most stable result. This also allows it to find clusters of varying densities. It is also less sensitive to hyperparameters than both DBSCAN and OPTICS. However, the runtime complexity of the algorithm is higher than that of the previously-mentioned ones.

Selecting a Clustering Algorithm
For the purpose of this paper, HDBSCAN and DBSCAN will be chosen for a comparison. The choice of *HDBSCAN* is motivated by its robustness to parameter selection, combined with the ability to find clusters of varying densities within the data. For the used dataset, domain knowledge is limited and HDBSCAN provides the most appropriate starting point. The other algorithm chosen for performing the experiments, *DBSCAN* is on the opposite end of the spectrum, being heavily influenced by its hyperparameter, ϵ. This allows a more exploratory stance on the dataset and provides a good contrast with the other algorithm choice.

2.4 Performing the Clustering

Hyperparameters
HDBSCAN requires the minimum number of points (IPs) for a cluster to be considered. The algorithm will be run with 2 and 10 set for this parameter and each outcome will be assessed separately.
For DBSCAN, ϵ is chosen empirically, training the algorithm multiple times on incrementally increasing ϵ-values, starting from *0.05*. The parameter will be chosen such that the clustering does not become too relaxed.

Features
The effectiveness of considering some of the behavioral features will be determined experimentally. For the purpose of this research, different subsets of the features detailed in Section 2.2 were chosen and clustering was performed on them. Features can be categorised in *fixed features*, that were fixed for all the experiments, and *test features*, that were varied throughout experiments.
The features that were fixed for all experiments were: subnet, ASN, scan length, median inter-packet time, distinct source ports, distinct destination ports, top port, top fingerprint, the IP distribution quartiles, the number of active days and average payload length. For the purpose of this paper, these features are considered to be defining for a scanning machine. It is evident too see that these features define the overall behavior of a scanning machines.
Different subsets of features were experimented with throughout the process and two test features were selected to be presented here. One very important detail of the behavior of scanners is the amount of traffic each one generates. Adversaries can choose to distribute scans evenly (this is also the default behavior of running a distributed ZMap scan) or can introduce randomness. As such, two feature sets will be considered for clustering, to determine whether the traffic influences the quality of the featuring:

Feature Set 1 (FS_1) Include the total hits and distinct hit IPs.

Feature Set 2 (FS_2) Only include the fixed features.

2.5 Evaluation

The baseline metric used to validate cluster quality will be the Calinski-Harabasz Index [20]. The nature of the problem at hand demands closer evaluation and validation than clustering metrics can offer. It is important in the broader context that the groups found have a certain significance, as their analysis can help offer new insights into how coordinated groups are evolving. Due to this reason, additional evaluation is needed. Besides the CH Index, the methods will also be evaluated based on two other factors:

Number of Heavy Hitters We consider a group to be a *heavy hitter* if the total average number of packets per day of activity is higher than 30 000.

Partial Cover - Destination IP Overlap Starting from the assumption that members of the same group will likely not scan the same IP:port combination multiple times, a group of source IPs with a high coverage of the telescope address space and a low number of overlaps is likely to belong to the same actor. We consider an overlap to happen when two IPs in the same group scan the same IP:port in the timespan of three hours. A group will be called to perform a *partial cover* if it hit more than 50% of the total active telescope IPs with no overlap. This translates to more than 32 000 distinct IPs. For groups with more active days, the average number of overlaps per day should be considered.

Algorithm 1 Greedy algorithm to find sub-groups with no overlap.

Input: $ipList$ - $ip_1 \dots ip_N$
Output: $G_1 \dots G_M$ - new groups; ns - unattributed points

```
k ← 1                                   ▷ Current number of groups
idx ← 1                                 ▷ Current index in ipList
initialize boolean set V, list ns.
while idx ≤ N do
    if ip_idx in V then continue
    end if
    add ip_idx to V, G_k
    nxtIdx ← idx + 1
    while nxtIdx ≤ N do
        add ip_nxtIdx to G_k
        if getOverlaps(G_k) > 0 then
            remove ip_nxtIdx from G_k
        else
            add ip_nxtIdx to V
        end if
        nxtIdx ← nxtIdx + 1
    end while
    if sizeof(G_k) = 1 then
        remove ip_idx from G_k
        add ip_idx to ns
    else
        k ← k + 1
    end if
    idx ← idx + 1
end while
```

A complete evaluation of the results can be found in Section 3.2.

2.6 Post-Processing: No Overlap Sub-Groups

Apart from the groups with high coverage and no overlaps, groups with high coverage and high overlap can also be of interest. Such groups are likely to contain 2 or more groups that behave similarly, but overlap in their scan. As an initial post-processing step, we propose a greedy algorithm that is intended to find sub-groups of no overlap within a group.
The full pseudocode for the algorithm can be found in Algorithm 1. It starts from the first unattributed IP and tries to find another IPs to group with it, while maintaing 0 overlap. If it can find a match, a new group is made. If not, the IP is marked as noise within the group. The algorithm then continues to the next IP that is unattributed yet.

3 Experimental Setup and Results

A brief overview of the technical setup used to run the experiments is given in Section 3.1. This is followed by the evaluation of the results, which can be found in Section 3.2.

3.1 Setup

The experiments and their analysis were run on multiple Jupyter Notebooks, running on a Python kernel. Notebooks were organised corresponding to different stages:

1. Data was aggregated and pre-saved to be reused.

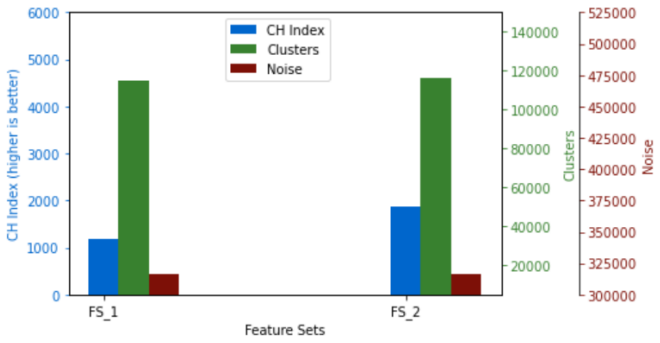

Figure 1: HDBSCAN metrics, $minPts = 2$

Feature Set	minPts	Clusters	Noise	CH Index
FS_1	2	115249	316442	1171.7446
FS_2	2	116333	316265	1868.9511
FS_1	10	10398	493000	3649.6426
FS_2	10	10440	493720	3671.3023

Table 2: HDBSCAN metrics

2. HDBSCAN and DBSCAN models trained with different feature sets, after scaling.
3. Results were analysed and evaluated.
4. Post-Processing was applied and the final results were re-evaluated.

The Pandas[1] library was used to manage and analyse results. For the clustering algorithms: sci-kit learn[2] implementation of DBSCAN was used, together with the HDBSCAN library [3].

3.2 Results

In this subsection, the general metrics for both clustering algorithms will be presented, followed by a comparison based on the amount of heavy hitters with no overlap each approach can identify, first without and then with post-processing.

HDBSCAN

The metrics captured from applying HDBSCAN on the two feature sets, with *minPts* as 2 and 10 are available in Table 2 and in Figure 1 and 2 respectively. There is a sizeable difference depending on *minPts*. Setting the parameter to 10 greatly reduces the number of identified clusters and increases the points marked as noise, with an overall increase in the CH Index as well. Moreover, removing the total hits and distinct IPs had a greater difference on the score when the minimum cluster size was lower.

The geographical distribution chart of the resulting clusters for FS_2 can be found in Figure 3 for $minPts = 2$ and in Figure 4 for $minPts = 10$. When the limit is set higher, the majority of the groups will span more than 3 subnets, but

[1] https://pandas.pydata.org/
[2] https://scikit-learn.org/stable/
[3] https://hdbscan.readthedocs.io/en/latest/index.html

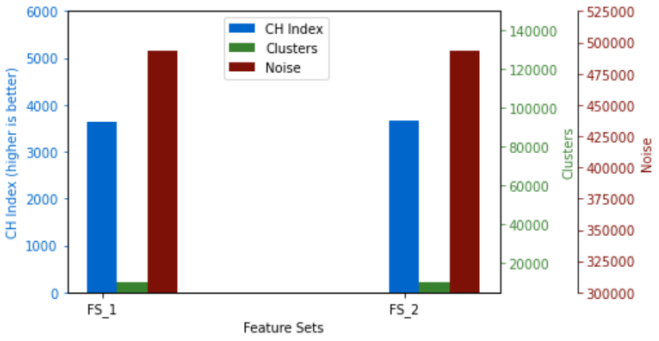

Figure 2: HDBSCAN metrics, $minPts = 10$

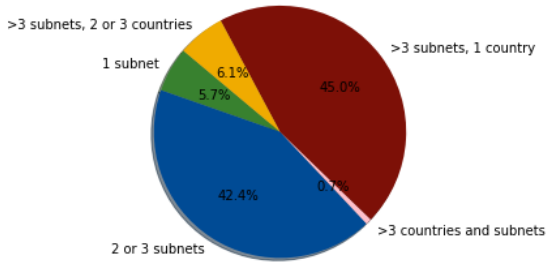

Figure 3: HDBSCAN distribution, FS_2, $minPts = 2$

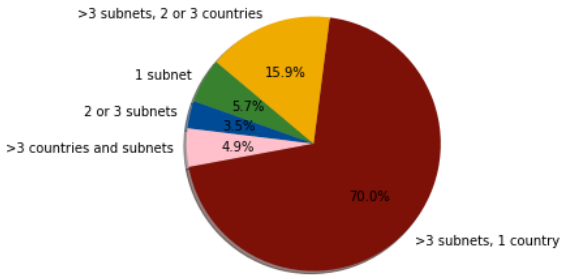

Figure 4: HDBSCAN distribution, FS_2, $minPts = 10$

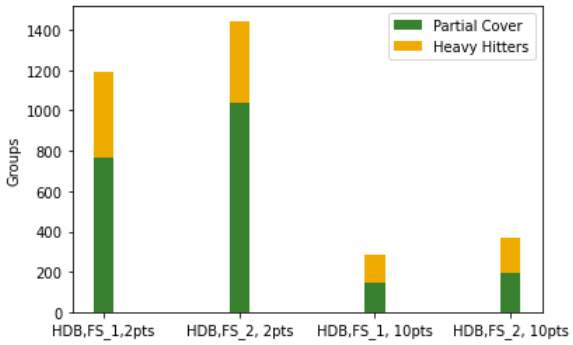

Figure 5: Address Range Coverage, HDBSCAN

one country, while the lower limit results in a higher percentage of clusters being found with 2 or 3 subnets and the same country. In general, the groups resulted from considering a higher minimum of IPs per cluster result in more geographical distribution inside a group.

Figure 5 showcases the amount of heavy hitters and the groups performing a partial cover of the address range among them. The graph shows a clear indication that only a small percentage of the total groups can be considered *heavy hitters*, according to the definition given in Section 2.5. Most of the heavy hitters found, however, hit more than half of the address space without overlaps.

DBSCAN

The DBSCAN algorithm was trained multiple times on both feature sets, with varying ϵ values. An overview of the metrics and the corresponding ϵ can be found in Figure 6 for FS_1 and in Figure 7 for FS_2. The chosen ϵ to be analysed further is 0.1.

The geographical distribution inside the groups resulting from DBSCAN applied on FS_2 can be found in Figure 8. Here, groups with IPs origination from 2 or 3 subnets in the same country are the majority. Notably, every approach resulted in different geographic distributions, but the share of groups made up of one subnet remained stable.

In terms of address range coverage, this technique exhibits the same weak point as previously presented approaches: only a small portion of the found clusters can be labeled as heavy hitters. A visual representation of the result can be found in Figure 9.

Post-Processing

The effect of applying the post-processing technique described in Section 2.6 to the groups found by applying HDBSCAN on FS_2, with $minPts = 2$ can be seen in Figure 10 and Table 3. 232 new groups doing a partial cover were identified, a 30.32% increase from the initial 765.

B/A	Heavy Hitters	Partial Covers	Increase(%)
Before	1444	765	-
After	1508	997	30.32%

Table 3: Increase in Partial Cover Groups after Post-Processing

4 Responsible Research

This section will discuss some of the principles presented in The Netherlands Code of Conduct for Research Integrity [21] that were considered throughout the research process, followed by a discussion on the reproductibility of the results and the ethical aspects of the research.

Transparency was one of the most important guiding principles while conducting the research. For this reason, steps were taken so that the code used throughout and all the intermediary data is readily available to be consulted. This can be done through a Github repository [4]. Moreover, significant context and explanations were given for all of the introduced methods and terms. Public access to the code and data files also ensures *honesty*.

[4] https://github.com/KruZZy/detecting-collaborative-scanners

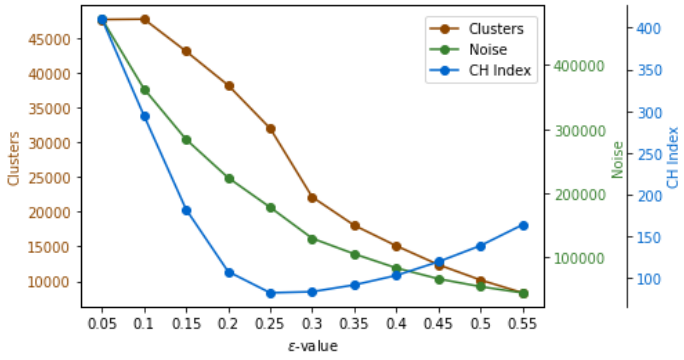

Figure 6: Dependency between ϵ and metrics for DBSCAN on FS_1

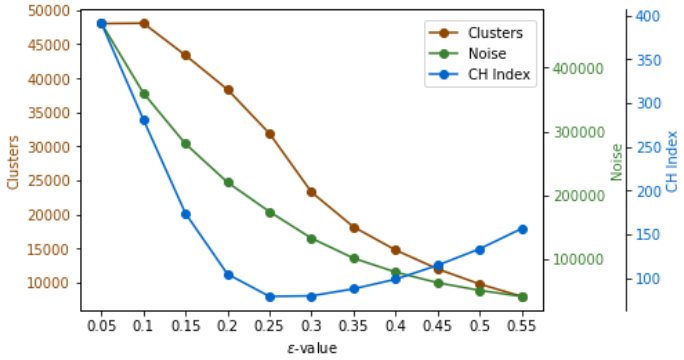

Figure 7: Dependency between ϵ and metrics for DBSCAN on FS_2

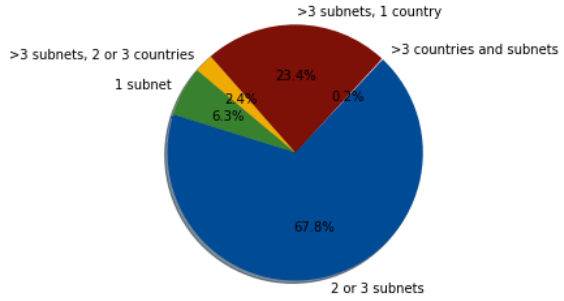

Figure 8: DBSCAN distribution, FS_2, $\epsilon = 0.1$

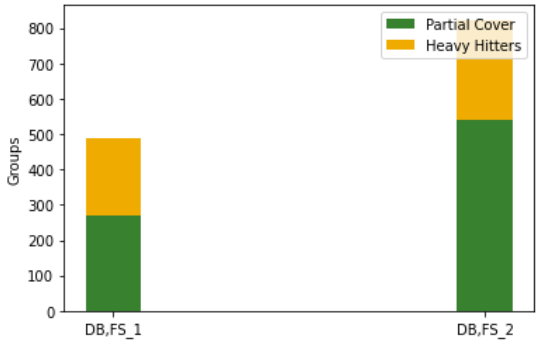

Figure 9: Address Range Coverage, DBSCAN

In terms of reproductibility, every step in the research process has been carefully documented. With the descriptions provided in this paper, any interested party can perform the experiment on the raw data provided by the network telescope. Moreover, if they wish so, they can also repeat any step in the process with the available data exports done at every step in the research.

The data used for this research is inherently anonymous, therefore privacy is not a concern. The subject of the research is also both scientifically and socially relevant: it aims to help in the detection and understanding of malicious parties on the internet, which is becoming ever more important for the good of society as a whole.

5 Discussion

The results of the experiments show a clear indication that the proposed technique of relying on the behavioral features introduced in Section 2.2 does not generally put *all* the IPs working together in the same group, and this is the primary limitation of the approach. Moreover, a significant number of groups also contain IPs that overlap in their search, but using post-processing techniques that analyse groups can mitigate this issue.

After looking at the results, another interesting point of discussion is whether the overlap model is indeed the appropriate criterion for a group to be considered. In order to put its potential limitations into perspective, we will analyse the composition of some of the groups found. The groups described below are extracted from the results of HDBSCAN on FS_2, with $minPts = 2$

Group #1 - Label 3025

This group hits a 62173 distinct IPs, with a very high average overlap of 57028. It is made of 7 IPs that use ZMap, all but one coming from the same subnet in Hong Kong. The other IP comes from Tokyo. Looking at the behavior of the IPs separately, the ones coming from Hong Kong scanned for 7 days, all hitting an average of exactly 3.1428 destination ports per day. The most active port was 22, accounting for an average of 1.172 probes per IP. Their ASN is 16509 - Amazon. The IP in Japan scanned for 8 days in the same time frame, but only hit an average of 1.5 destination ports per day, with the most active port being 4460. Its ASN is 45102 - Alibaba. A full account of other features of the IPs can be found in Table 4.

IP	Inter-Packet Time	Source Ports	Total Hits
JP	2000	3931.12	11983.75
HK-1	2714	4259.57	16897.85
HK-2	3285	4246.28	16696.85
HK-3	3142	4243.14	16717.57
HK-4	3000	4253.28	16750.85
HK-5	3000	4250.28	16851.28
HK-6	3285	4144.14	12761.00

Table 4: Composition of Group Label 3025

After looking at the features, it is reasonable to consider

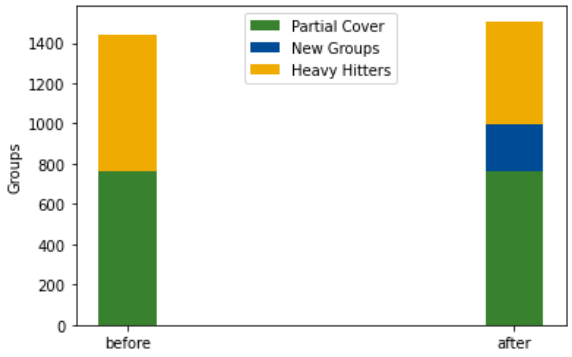

Figure 10: Effect of post-processing on HDBSCAN, FS_2, $minPts = 2$

that the IP in Japan is a false positive and disregard it as part of this group. However, applying post-processing to this group to find non-overlapping subgroups puts JP, HK-1 and HK-6 in the same group and is unable to find any non-overlapping sub-group between the other 4 groups in Hong Kong. Moreover, they also have overlaps when trying to combine them manually. Aside from port 22 (SSH), they also hit port 443 (HTTPS), 3389 (Remote Desktop), 3306 (MySQL). Interestingly, port 443 is used with ports that usually require credentials. A potential reason for this could be that they are trying to scan for IPs that support SSH over the HTTPS port. Some firewall configurations do not allow access through port 22 and organisations opt to have it open through 443, the usual HTTPS port. [22]
A potential reason for the overlap of these IPs could be different sets of credentials for each scanning IP. Therefore, in the case of this group, the result of applying the overlap criterion for the group selection is up for debate.

Group #2 - Label 38
This group contains, among other IPs that can be disregarded for this discussion, 4 IPs from China, in 2 different subnets from the same ASN, using an unclassified tool. An overview of the IPs and the average number of destination ports and distinct IPs each targeted each day can be found in Table 5, with the 2 different subnets marked as [X] and [Y]. In the case of this cluster, IPs in one subnet have overlap in their scanning. When combining an IP from one subnet with an IP from the other one, they together hit 64997 IPs in the telescope with no overlap. Moreover, it does not matter which IPs are chosen from each subnet. Any of the two IPs from *[X].0* can be grouped with any IP from *[Y].0* and they yield the same result. These IPs could belong to an adversary that programs machines on each subnet to scan a specific part of the internet and does multiple runs. By the overlap criterion, this should be considered as two groups, but the manner in which these scans are orchestrated indicates that they are the same group.

6 Conclusions and Future Work

To conclude, this paper proposed a new approach to cluster collaborative scanners in network telescope data, based on aggregated behavioral features, together with an evaluation framework based on the overlap of targeted IPs within the group. The approach proves effective in discovering patterns in the data and identifying subsets of the scanning groups, with HDBSCAN having better performance than DBSCAN for the chosen hyperparameters.
The main limitation of this approach, as shown by the results, is that most identified clusters do not cover enough of the address range to be considered full scanning groups. This is a limitation of the data set, as there is no ground truth data available to allow for supervised learning that would encourage the creation of heavy-hitting clusters. Manual analysis has releaved, however, that general patterns and part of the groups are discovered through this method. These results can be used to conduct further threat intelligence analysis.
Moreover, a closer analysis of the group composition indicates that relying on the overlap of IPs for group evaluation is not effective against all adversaries. This hypothesis provides interesting grounds for new research. It is apparent that some have started scanning the same target multiple times and are sharding the IP ranges their machines scan in ways that are not uniform - possibly to avoid detection. The majority of the identified groups do not cross country boundaries yet and, often, machines within a group use the infrastructure of only one ASN.
There are several research directions to be explored from these results. Domain knowledge could be used to perform clustering with constraints, for example by using C-DBSCAN [23]. As the scanning landscape is understood better, the constraints could evolve and guide to better results. The entirety of the dataset was used for the research, irrespective of the tool. Another direction for future research can be exploring the behavior of scanners using different tools, starting from the recognizable fingerprints. Afterwards, different feature sets and algorithms can be trained per tool, to decrease the complexity of the problem.

#	IP	Days	Dest Ports	Distinct IPs
1	**[X]**.170	1	53	39168
2	**[X]**.171	1	40	39168
3	**[Y]**.186	2	53	12914
4	**[Y]**.187	2	40	12914

Table 5: Composition of Group Label 38

References

[1] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications", in *22nd USENIX Security Symposium (USENIX Security 13)*, Washington, D.C.: USENIX Association, Aug. 2013, pp. 605–620, ISBN: 978-1-931971-03-4. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric.

[2] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, "Census and

survey of the visible internet", in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08, Vouliagmeni, Greece: Association for Computing Machinery, 2008, pp. 169–182, ISBN: 9781605583341. DOI: 10.1145/1452520.1452542. [Online]. Available: https://doi-org.tudelft.idm.oclc.org/10.1145/1452520.1452542.

[3] Z. Durumeric, F. Li, J. Kasten, *et al.*, "The matter of heartbleed", in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14, Vancouver, BC, Canada: Association for Computing Machinery, 2014, pp. 475–488, ISBN: 9781450332132. DOI: 10.1145/2663716.2663755. [Online]. Available: https://doi.org/10.1145/2663716.2663755.

[4] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains", *Leading Issues in Information Warfare Security Research*, vol. 1, Jan. 2011.

[5] A. Anand, M. Kallitsis, J. Sippe, and A. Dainotti, *Aggressive internet-wide scanners: Network impact and longitudinal characterization*, 2023. arXiv: 2305.07193 [cs.NI].

[6] S. Robertson, E. Siegel, M. Miller, and S. Stolfo, "Surveillance detection in high bandwidth environments", in *Proceedings DARPA Information Survivability Conference and Exposition*, vol. 1, 2003, 130–138 vol.1. DOI: 10.1109/DISCEX.2003.1194879.

[7] C. Gates, "Co-ordinated port scans: A model, a detector and an evaluation methodology", 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:110563792.

[8] A. Tanaka, C. Han, and T. Takahashi, "Detecting coordinated internet-wide scanning by tcp/ip header fingerprint", *IEEE Access*, vol. 11, pp. 23 227–23 244, 2023. DOI: 10.1109/ACCESS.2023.3249474.

[9] F. Maymí, R. Bixler, R. Jones, and S. Lathrop, "Towards a definition of cyberspace tactics, techniques and procedures", in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 4674–4679. DOI: 10.1109/BigData.2017.8258514.

[10] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope", in *Proceedings of the Internet Measurement Conference*, ser. IMC '19, Amsterdam, Netherlands: Association for Computing Machinery, 2019, pp. 144–157, ISBN: 9781450369480. DOI: 10.1145/3355369.3355595. [Online]. Available: https://doi.org/10.1145/3355369.3355595.

[11] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope", in *Proceedings of the Internet Measurement Conference*, ser. IMC '19, Amsterdam, Netherlands: Association for Computing Machinery, 2019, pp. 144–157, ISBN: 9781450369480. DOI: 10.1145/3355369.3355595. [Online]. Available: https://doi.org/10.1145/3355369.3355595.

[12] R. D. Graham, *Masscan: Mass ip port scanner*, https://github.com/robertdavidgraham/masscan, Accessed: May 23rd, 2024.

[13] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet", Jan. 2019. DOI: 10.14722/ndss.2019.23488.

[14] J. Sahota and N. Vlajic, "Mozi iot malware and its botnets: From theory to real-world observations", in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021, pp. 698–703. DOI: 10.1109/CSCI54926.2021.00181.

[15] M. Antonakakis, T. April, M. Bailey, *et al.*, "Understanding the mirai botnet", in *Proceedings of the 26th USENIX Conference on Security Symposium*, ser. SEC'17, Vancouver, BC, Canada: USENIX Association, 2017, pp. 1093–1110, ISBN: 9781931971409.

[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[17] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure", in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '99, Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1999, pp. 49–60, ISBN: 1581130848. DOI: 10.1145/304182.304187. [Online]. Available: https://doi-org.tudelft.idm.oclc.org/10.1145/304182.304187.

[18] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates", in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172, ISBN: 978-3-642-37456-2.

[19] J. Esmaelnejad, J. Habibi, and S. H. Yeganeh, "A novel method to find appropriate ϵ for dbscan", in *Intelligent Information and Database Systems*, N. T. Nguyen, M. T. Le, and J. Świątek, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 93–102, ISBN: 978-3-642-12145-6.

[20] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis", *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974. DOI: 10.1080/03610927408827101. eprint: https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

[21] KNAW, NFU, NWO, TO2-federatie, V. Hogescholen, and VSNU, *Nederlandse gedragscode wetenschappelijke integriteit*, 2018. DOI: 10.17026/dans-2cj-nvwu. [Online]. Available: https://doi.org/10.17026/dans-2cj-nvwu.

[22] H. Dwivedi, *Implementing SSH: strategies for optimizing the secure shell*. John Wiley & Sons, 2003, p. 123.

[23] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, "C-dbscan: Density-based clustering with constraints", in *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, A. An, J. Stefanowski, S. Ramanna, C. J. Butz, W. Pedrycz, and G. Wang, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 216–223, ISBN: 978-3-540-72530-5.