Detecting Collaborative ZMap Scans

Detection of distributed ZMap scans in network telescope data using an algorithmic approach

Fatih Acikkollu
Supervisor(s): Harm Griffioen, Georgios Smaragdakis
[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: Fatih Acikkollu
Final project course: CSE3000 Research Project
Thesis committee: Harm Griffioen, Georgios Smaragdakis, Kubilay Atasu

Abstract

Detecting distributed scans is crucial for understanding network security threats. This research uses an algorithmic approach to identify collaborative ZMap scanning activities in the network telescope data from TU Delft. ZMap is a high-speed network scanner capable of scanning the entire IPv4 address space. The main research question centers on creating an algorithm for detecting these distributed scans. The research method includes analyzing network telescope data, examining ZMap packets and modifying the set cover algorithm to detect collaborative ZMap scans. Key contributions include adapting the set cover algorithm to find sources that perfectly cover the entire destination address range without overlaps, which is a unique feature of ZMap scans. Results indicate that this method effectively identifies coordinated scanning activities. It is concluded that utilizing network telescope data and an adapted version of the greedy set cover algorithm significantly improves the detection of distributed scanning operations using ZMap.

1 Introduction

In today's world, almost everything from heavy machinery to household items is connected to the Internet. In 2024, the number of IoT devices is above 15 billion which was around 8.5 billion in 2019 [1]. Figure 1 displays the constant increase in the number of IoT devices since 2019 with projections extending to 2030. Due to this increase in such devices and services, attackers have a wide range of targets. They often perform Internet-wide scans to find vulnerabilities using scanners which are programs used to scan IP addresses for accessible ports. Such scans provide these malicious parties an overview of potential vulnerabilities and system weaknesses, which they can exploit. Consequently, detecting these port scans as early as possible offers valuable information about the attackers and the specific services they are targeting [2].

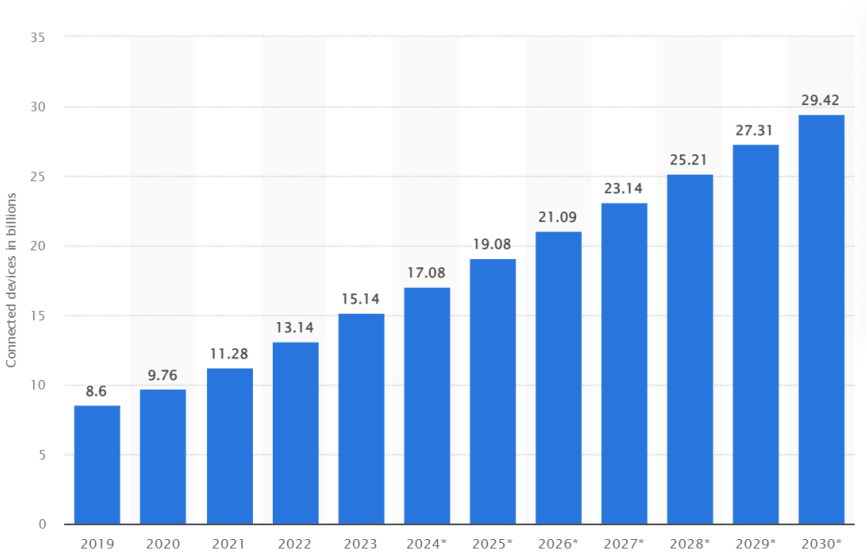

Figure 1: Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030 (in billions) [1]

While it is trivial to detect and block scans from a single device due to the high number of packets they are sending, detecting coordinated scans from multiple devices is significantly harder [2]. This is the main problem addressed in this research. Detecting these distributed scans is important because they can lead to large-scale cyber attacks and data breaches. Identifying and stopping these scans can help protect sensitive information and improve overall network security. This research focuses on the detection of distributed scans using a scanner tool called ZMap[1] using an algorithmic approach to the network telescope data provided by the TU Delft. The main research question is *"How can we detect collaborative ZMap scans in network telescope data using an algorithmic approach?"* with the following sub-questions:

1. What are the characteristics of collaborative scanning activities using ZMap in network telescope data?
2. How can set cover algorithms be applied to distinguish collaborative scanners from other network traffic?
3. What are the main challenges in detecting collaborative ZMap scanners and how can they be addressed?

Firstly, we aim to understand the main features of collaborative scanning activities using ZMap. Secondly, we investigate how set cover algorithms can help separate collaborative scanners from other network traffic using the characteristics identified in the first sub-question. Finally, we explore the main challenges in detecting collaborative ZMap scans and suggest ways to tackle them. These sub-questions help us understand and solve problems with detecting collaborative ZMap scans.

Section 2 contains background information regarding the ZMap tool and the specifications of the network telescope data. Section 3 explains the methodology of this research in detail while results are discussed and analyzed in Section 4. Furthermore, ethical aspects and reproducibility of the methods can be found in Section 5. Finally, discussions and conclusions are included in Sections 6 and 7 respectively.

[1] https://github.com/zmap/zmap

2 Background Information

This section aims to provide essential information for understanding the concepts within this research. Subsection 2.1 explains internet scanning, including its methods and purposes. Subsection 2.2 describes the ZMap tool, its capabilities and usage. Subsection 2.3 covers network telescopes and their role. Subsection 2.4 explains the set cover problem and its relevance to this research. Finally, Subsection 2.5 reviews related research on detecting distributed scanners.

2.1 Internet Scanning

Internet scanning is the process of checking IP addresses on the internet to find open ports, services and potential security gaps. This is done to gather information about active devices and the services they run. The scanning can be done using different methods, like vertical scanning, which checks all ports on one IP address, horizontal scanning, which checks one port across many IP addresses and block scanning, which combines checking multiple ports on multiple IP addresses [2].

The main goal of internet scanning is to understand the security of systems. It can be used by security researchers and network administrators to identify and fix vulnerabilities. However, attackers also use it to find weaknesses they can exploit. Detecting scanning activities early is important because it helps to understand and respond to potential threats quickly [2].

2.2 The ZMap Tool

ZMap is a network scanner tool that can quickly scan the entire public IPv4 address space. It can scan at a rate of 1.44 million packets per second, completing the full scan in under 45 minutes on a typical connection. Enhancements have increased its speed to nearly 10 gigabits per second, allowing it to scan the IPv4 space in under 5 minutes. These improvements allow for faster and more accurate Internet-wide scans which is beneficial for security research but also creates possible risks if misused by attackers [3].

ZMap has an option to distribute the scanning tasks across multiple machines. This can be achieved using the `--shards`, `--shard` and `--seed` flags. It is important to ensure that the seed, shard count and ZMap version are consistent across all machines to avoid overlapping or uncovered destination addresses [4]. In ZMap, a shard represents a segment of the IPv4 address space that can be scanned independently of other shards. Each shard covers a separate subset of the address space without overlaps and together, all shards cover the entire address space [3].

2.3 Network Telescopes

A network telescope is a segment of unused IP addresses that monitors unexpected incoming traffic to detect network security incidents such as DDoS attacks, Internet worm infections and network scanning activities [2, 5]. These telescopes passively collect data without initiating communication, which makes them effective for observing anomalous behavior.

The size of the network telescope (the fraction of address space it covers) is crucial for its effectiveness. A larger address space increases data volume, enhancing the accuracy of detecting and characterizing network events [5].

2.4 Set Cover Problem

The set cover problem is the challenge of finding the smallest number of subsets from a given collection such that their union is equal to the universal set. It is one of the Karp's 21 NP-complete problems meaning it is a challenging problem known to be difficult to solve efficiently [6, 7].

Imagine the universe $U = \{1, 2, 3, 4, 5\}$ and the subsets $s_1 = \{1, 2\}$, $s_2 = \{2, 3\}$, $s_3 = \{2, 3, 4, 5\}$, $s_4 = \{3, 4, 5\}$, $s_5 = \{4\}$. Clearly, the union of all subsets is equal to U. However, we can cover all

elements using only two subsets: $\{\{1,2\},\{3,4,5\}\}$. Thus, the solution to the set cover problem in this case includes the subsets s_1 and s_4. This example is illustrated in Figure 2.

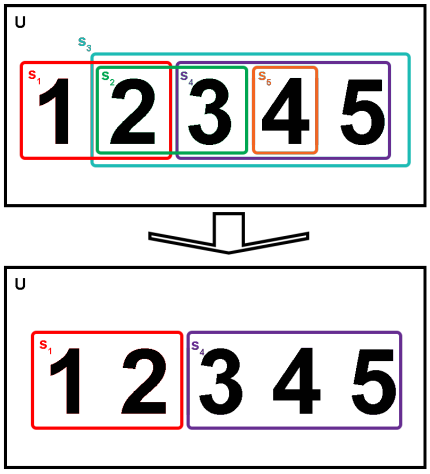

Figure 2: Example set cover problem

Set covering problems appear in various contexts such as operations research, machine learning, planning, data quality and data mining. While discovering the best solution is tough due to its NP-hard nature, greedy algorithms are commonly used and often yield solutions that are nearly optimal [8]. Set cover algorithms are important in our research because they help us find sources that perfectly cover all the addresses the telescope monitors. This is crucial for detecting collaborative ZMap scans effectively.

2.5 Related Research

Most research conducted in detection of the distributed scanners aims to detect collaborative scanning activities in general instead of focusing solely on ZMap packets. One such research proposes a fingerprinting technique to identify distributed scanning activities, but it does not specifically address the detection of ZMap scans [2].

Robertson et al. [9] propose detection of distributed scans by assuming that IP addresses involved in these scans are close together. This may not always hold true since they might be originating from widely distributed botnets. Similarly, Yegneswaran et al. [10] identify coordinated behavior by analyzing destination ports and IP addresses but miss the identification of groups [2].

Similar to this research, Carrie Gates [11] also suggested utilizing the set cover algorithm but again in a general scope. Their algorithm aims to maximize coverage and minimize overlap, which allows it to also detect collaborative ZMap scans. However, allowing overlaps makes this algorithm inefficient in comparison to the algorithm explained in Section 3.3. The main reason behind this is using shards to distribute ZMap scans does not result in overlapping destination IP ranges. This allows us to immediately reject a source if it has an overlap with the currently covered destination IPs and reduce the amount of work.

3 Methodology

The purpose of this section is to explain the methodology of this research. The methodology is divided into five main steps and each step is explained in a separate subsection. Subsection 3.1 discusses the analysis of the network telescope data provided by the TU Delft. Subsection 3.2 contains an analysis of the ZMap packets and the way it performs distributed scans. Subsection 3.3 explains the adaptation of the set cover algorithm to identify collaborative scanning activities. The method for validating the implemented algorithm is discussed in Subsection 3.4. Finally, Subsection 3.5 provides information about the identification of the challenges of this research.

3.1 Analyzing Network Telescope Data

The TU Delft network telescope monitors three IP ranges. The first two ranges are for staff and network equipment, while the third range is for Eduroam[2], mostly used by students and visitors. The exact ranges are not disclosed here due to confidentiality concerns.

IP addresses change over time; when a device disconnects, its IP address goes back into the pool, making traffic to that address unroutable and logged by the telescope. This setup stops attackers from mapping the network but can create noise. Traffic data is captured with Tshark[3] and saved in .pcap[4] files, collecting about 14.4 GB of data daily [2].

For this research, the data captured in February 2024 is utilized, which contains 12.64 billion scans. According to the `src/probe_modules/packet.c` file in the ZMap source code[5], the tool uses 54321 as the default IP identification number. Using this information for filtering telescope data yields 4.55 billion scans, which is 36% of the total captured scans.

As mentioned in the second paragraph of this section, the TU Delft network telescope can be noisy. To tackle this, we check how often each destination address gets scanned. Sorting the destination addresses by scan frequency, we find that the $112,431^{st}$ address is scanned 421 times, while the next one on the list is scanned 14,053 times. Given this notable difference, we set a threshold of 1,000 scans to filter out the noise. Addresses scanned less than 1,000 times are considered noise and are therefore ignored. This is because these addresses might not always be active or could be involved in temporary network activities, contributing to the overall noise in the dataset.

Another important aspect to consider is identifying the most scanned ports. This helps in deciding which ports to prioritize while running the algorithm explained in Subsection 3.3. Figure 3 visualizes the top 10 most scanned ports using ZMap and their respective scan counts. The queries used for this analysis and others in this section are available in Appendix A.

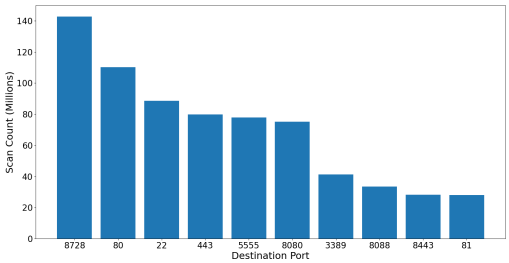

Figure 3: Top 10 most scanned ports using ZMap

[2]https://eduroam.org
[3]https://tshark.dev
[4]https://www.solarwinds.com/resources/it-glossary/pcap
[5]https://github.com/zmap/zmap

Due to the time constraints of this research, only the first four ports are considered. Here is what each port is used for:

- **8728:** MikroTik RouterOS API [12]
- **80:** World Wide Web HTTP [13]
- **22:** Secure Shell (SSH) Protocol [13]
- **443:** HTTP Protocol over TLS/SSL [13]

3.2 Analyzing ZMap Probes

"Probes are the ZMap term for the packets ZMap sends out to potential hosts" [4]. To analyze these probes, `--dryrun` option is utilized which displays the generated packets without actually sending them [4]. Using the `--dryrun` flag, a distributed scan simulation is performed using 4 sources (shards) on the `171.67.70.0/24` IP range and port 80 using the following commands:

```
Source 1: zmap -p 80 171.67.70.0/24 --shards 4 --shard 0 --seed 1453 --dryrun
Source 2: zmap -p 80 171.67.70.0/24 --shards 4 --shard 1 --seed 1453 --dryrun
Source 3: zmap -p 80 171.67.70.0/24 --shards 4 --shard 2 --seed 1453 --dryrun
Source 4: zmap -p 80 171.67.70.0/24 --shards 4 --shard 3 --seed 1453 --dryrun
```

The analysis of ZMap probes aims to explain how ZMap generates packets and distributes scans. By examining these aspects, we try to find patterns in its distributed scanning behavior, which can be used to adapt and optimize the detection algorithm.

3.3 Adapting Set Cover Algorithm to Detect Collaborative ZMap Scans

The greedy algorithm for solving the set cover problem mentioned in Section 2.4, aims to find covers with the fewest subsets. However, applying this algorithm directly is not suitable for this research, as the goal here is to identify all groups, not just those with the minimum number of sources. Consequently, a new algorithm, inspired by the greedy set cover algorithm, was developed to find as many groups as possible that cover the entire address space. Based on the analyses discussed in the previous sections, overlaps are not allowed and noise is removed from the data. The algorithm is explained in the following steps:

1. Begin with a window size of 1 hour.
2. Query data that falls within the window.
3. Create the universal set by collecting distinct destination IPv4 addresses from the queried data.
4. Create a mapping of source IPv4s to the list of destination addresses they scanned.
5. Sort the source IPv4s based on the number of addresses they scan.
6. Initialize the "selected sources" and "current cover" sets.
7. From the source-to-destination map, select the source with the most uncovered destinations.
8. If the source overlaps with the current cover, skip it; otherwise, add it to the selected sources set and update the current cover.
9. Repeat steps 7 and 8 until no more sources can be selected.
10. If the universal set is covered at the end, add the selected sources to the results and remove a randomly selected source from the map. Then, reset the selected sources and current cover sets and return to step 6.
11. If the universal set is not covered, remove the first source from the sorted map and return to step 6.
12. If no more covers are possible for the current window, shift the window by 1 hour and return to step 2.
13. If the window is shifted to the end of the day and the current window size is not 24 hours, increase the window size by 1 hour and return to step 2.

This algorithm works on a specific day and port. It uses shifting windows that increase in size after each round to catch as many groups as possible. Some sources cover the same area, so they can be used in different groups. When a group is found that covers the whole address space, only one random source

is removed from the selected sources set, not all of them (step 10). This way, in the next round, the remaining sources can still be clustered in other groups. Since the algorithm runs on windows ranging from 1 to 24 hours, shifting by 1 hour each time, the chance of the same source being removed every time is low. This means that the algorithm can find almost all covers. The pseudocode of the algorithm and the helper function is provided below:

Algorithm 1 Detect Collaborative Scans

```
1: Initialize covers as an empty set
2: Initialize n to 1
3: for window_size from 1 to 24 do
4:     for start_hour from 0 to (24 - window_size) do
5:         Query data for the given date, port, start_hour, and window_size
6:         Create src_to_dst map from the queried data
7:         while there are sources in src_to_dst do
8:             cover ← SetCover(src_to_dst)
9:             if cover is None then
10:                break
11:            end if
12:            Convert cover to a hashable type
13:            if cover is in covers then
14:                Remove a random source from the cover in src_to_dst
15:                continue
16:            end if
17:            Add cover to covers
18:            Write the cover details to the results file
19:            Remove a random source from the cover in src_to_dst
20:            Increment n by 1
21:        end while
22:    end for
23:    if n is 1 then
24:        Write "No covers found!" to the results file
25:    end if
26: end for
```

Algorithm 2 SetCover (helper function)

```
1: Get all_dstips as the union of all destination IPs in src_to_dst
2: Sort src_to_dst keys by the number of destination IPs in descending order
3: Initialize selected_srcs as an empty set
4: Initialize covered_dsts as an empty set
5: while there are sources in the sorted list do
6:     for each srcip in the sorted list do
7:         if srcip has any destination IPs already covered then
8:             continue
9:         end if
10:        uncovered_dstips ← src_to_dst[srcip] − covered_dsts
11:        if uncovered_dstips is not empty then
12:            Add srcip to selected_srcs
13:            Update covered_dsts with uncovered_dstips
14:            if covered_dsts equals all_dstips and selected_srcs has more than 1 source then
15:                return selected_srcs and size of all_dstips
16:            end if
17:        end if
18:    end for
19:    Remove the first source from the sorted list
20:    Reset selected_srcs and covered_dsts
21: end while
22: return None
```

It is important to note that because the windows are shifted by 1 hour and the algorithm is run daily, this approach cannot detect groups that scan multiple times within an hour or groups that scan slowly over a period longer than one day. Additionally, the algorithm is greedy, meaning that it might not always find all the groups. The query used for step 2 can be found in Appendix B.

3.4 Algorithm Validation

The correctness of the algorithm explained in Section 3.3 is crucial for the validity of the results of this research. Since the TU Delft network telescope data is not labeled, a new method for validation is developed. First, after creating the universal set of distinct destination addresses (step 3), n sources are generated where n is an arbitrary number between 2 and 256. These sources will have IPv4 addresses in the range `0.0.0.0` to `0.0.0.n` to avoid conflicts with the source addresses in the telescope data. Next, the destination addresses in the universal set are randomly distributed to the generated sources. After that, the generated sources and their destination address portion are inserted into the source-to-destination map (before step 5). Finally, the algorithm results are checked to see how many of these injected groups are identified. Due to the greedy nature of the algorithm, not all groups may be caught, but a high percentage is expected.

3.5 Identifying the Challenges of Detecting Collaborative ZMap Scans

To address the third sub-question of this research, any challenges encountered during the first four steps of the methodology, along with the solutions implemented to tackle these problems and their effectiveness, are documented.

4 Results and Analysis

In this section, the results of the methodology explained in Section 3 are presented. Each sub-question mentioned in Section 1 is answered in a separate subsection. Subsection 4.1 contains the answer for sub-question 1 and Subsection 4.2 answers sub-question 2. Finally, sub-question 3 is addressed in Subsection 4.3

4.1 Characteristics of Collaborative ZMap Scanning Activities in Network Telescope Data

In this subsection, results of the analyses described in Section 3.1 and Section 3.2 are examined to answer the first sub-question: *What are the characteristics of collaborative scanning activities using ZMap in network telescope data?*.

The distributed scan simulation results show that the total number of packets is evenly distributed across four sources, with each source scanning 64 addresses. Furthermore, no IP addresses were scanned more than once, confirming that ZMap shards do not cause overlapping scans. However, no clear pattern was found regarding which sources scanned which IP addresses. Additionally, the generated sequence numbers of the packets were random. The detailed outputs of the distributed scan simulation are available in Appendix E.

After filtering out the noise as explained in Subsection 3.1, the number of distinct destination addresses drops significantly from 174,669 to 62,242. Considering this and the limited range of the telescope, it is expected that sources involved in a collaborative scan might not cover an equal number of destinations perfectly. However, they still should not have any overlapping covers. This is indeed the case for the groups identified by the algorithm, which is discussed in detail in the next section.

4.2 Using Set Cover Algorithm to Distinguish Collaborative Scanners from Other Network Traffic

The purpose of this subsection is the answer the sub-question *How can set cover algorithms be applied to distinguish collaborative scanners from other network traffic?* by showing that the algorithm described in Section 3.3 can be utilized to discover collaborative scanning activities. The algorithm is tested using the method explained in Section 3.4. Validation and algorithm results are explained further in the following subsections. Full results, as well as the validation results, are available at `https://github.com/purificateur/collaborative_zmap_scans`.

4.2.1 Validation Results

The validation method described in Section 3.4 was run for the first three days of February 2024 on ports 8728, 80, 22 and 443, identified as the most scanned ports in Section 3.1. For each day and port, groups were injected into the first window of each window size, resulting in 24 injections per day per port. Table 1 shows the number of detected injections for each day and port.

Day	Port 8728	Port 80	Port 22	Port 443
01-02-2024	24	19	20	23
02-02-2024	24	20	19	21
03-02-2024	24	21	22	23

Table 1: Number of injected groups detected each day for each port

The total number of injected groups is 288 (24 window sizes $\times$ 3 days $\times$ 4 ports). According to Table 1, 260 of these groups were detected by the algorithm. This means the algorithm can detect collaborative ZMap scans with an accuracy of more than 90%. Additionally, the algorithm did not mix any injected sources with actual sources.

4.2.2 Results for Port 80

In February 2024, at total 8,527 groups were found that perfectly cover the entire address space without overlaps on port 80. The number of groups found per day is visualized in Figure 4 and the size distribution of these groups is visible in Table 5.

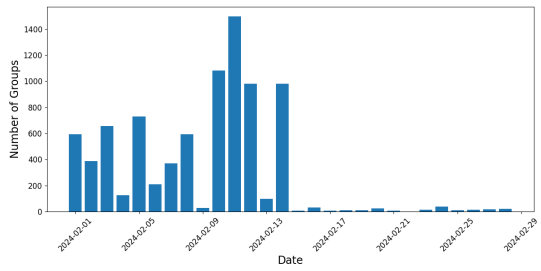

Figure 4: Number of Groups Found per Day in February 2024 (Port 80)

Group Size	Total Number of Groups
2	37
3	328
9	1
23	6
24	23
25	35
26	69
27	361
28	407
29	1461
30	5794
37	2
359	3
Total	**8527**

Figure 5: Number of Groups Found per Group Size in February 2024 (Port 80)

According to Table 5, the total number of sources that are part of a distributed scanning activity should be 242,909. However, the number of distinct sources is only 464. The reason for this significant difference is that there are sources that cover exactly the same address space and multiple groups can be formed by mixing these source addresses. Several examples of such groups are included in Appendix D. One identified reason for these mixed groups is the fact that the telescope monitors three different ranges of IPv4 addresses.

After filtering the noise, the total number of distinct sources that perform a ZMap scan on port 80 is 11,085. This means the algorithm has shown that about 4.19% of these sources are grouped into at least one collaborative scanning activity.

For port 80, the characteristics of sources within each group are examined more closely using AbuseIPDB[6]. This analysis reveals that nearly all sources within each group are from the same Internet Service Provider (ISP). Below are the names of the ISPs along with details of the groups associated with them:

- **DigitalOcean LLC**[7]: Most of the groups of size 3 and all of the groups of size 23 to 30 belong to DigitalOcean ISP. There are also several groups of 2 sources both belonging to DigitalOcean. These groups make up about 99% of the total groups. One main reason for this is the effect of groups with mixed sources mentioned above.
- **Akamai / Linode LLC**[8]**:** Contains groups with size 2. Certain sources from this ISP cover exactly the same destinations as specific sources from DigitalOcean. This results in combinations of sources where one source is from DigitalOcean and the other is from Linode/Akamai.
- **SecurityTrails LLC**[9]**:** One scan on 09-02-2024 with 9 sequential source IPv4 addresses ranging from `195.230.103.242` to `195.230.103.250`. They almost scan the same number of destinations.
- **Gemnet LLC**[10]**:** Two scans using 37 sources on the 13^{th} and 26^{th} of February, using the same sequential IPv4 addresses from `180.149.125.201` to `180.149.125.237`. These sources also scan an almost equal number of destinations.

[6] https://www.abuseipdb.com/check/
[7] https://www.digitalocean.com
[8] https://www.linode.com
[9] https://securitytrails.com
[10] https://www.gemnet.mn

- **Rapid7**[11]**:** Three scans using the same 359 sources. These sources consist of sequential IPv4 addresses from 5 different ranges: `5.63.151.100` to `5.63.151.126`, `71.6.233.3` to `71.6.233.254`, `88.202.190.132` to `88.202.190.158`, `109.123.117.228` to `109.123.117.254` and `146.185.25.164` to `146.185.25.190`.

4.2.3 Results for Port 22

On port 22, there were a total of 212 groups found, each perfectly covering the entire address space without overlaps. Figure 6 illustrates the daily distribution of these groups, while Table 7 provides details on their size distribution.

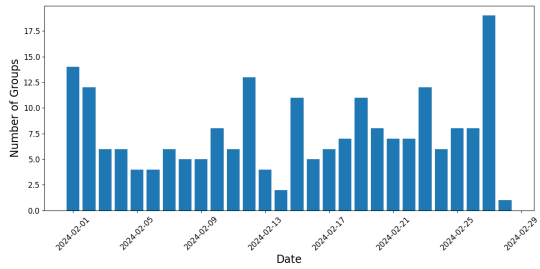

Figure 6: Number of Groups Found per Day in February 2024 (Port 22)

Group Size (Range)	Total Number of Groups
2	41
3	114
168 - 186	7
215 - 280	48
359	2
Total	**212**

Figure 7: Number of Groups Found per Group Size (Range) in February 2024 (Port 22)

There are 792 distinct sources involved in distributed scanning activity on port 22. Out of the total 19,440 distinct sources performing ZMap scans on port 22, only about 4.07% are observed to be grouped into collaborative scanning activities by the algorithm.

The sources are analyzed using AbuseIPDB, similar to what was done for port 80. As observed in the results for port 80, nearly all sources within each group are from the same Internet Service Provider (ISP). The same Rapid7 group with 359 sources, as well as groups with sizes 2 and 3 from DigitalOcean and Akamai/Linode, are also detected on port 22. Additionally, new groups are identified. Below are the details of these new groups:

- **Palo Alto Networks**[12]: All groups with a size range between 215 and 280 belong to Palo Alto Networks. These groups are active on almost every day of February 2024 on port 22 and contain the same sequential source IPv4 addresses from two different ranges, namely `198.235.24.2` to `198.235.24.255` and `205.210.31.2` to `205.210.31.255`.
- **The Shadowserver Foundation**[13]: Groups with a size range between 168 and 186 are from The Shadowserver Foundation. These groups also consist of the same sequential source IPv4 addresses from two different ranges: `64.62.197.3` to `64.62.197.241` and `65.49.1.10` to `65.49.1.121`.

4.2.4 Results for Port 443

A total of 7331 groups were identified for port 443, mostly consisting of various combinations of the same sources covering the same address space, referred to as "mixed groups" above. The daily distribution of these groups can be seen in Figure 8, while Table 9 presents their size distribution.

[11] https://www.rapid7.com
[12] https://www.paloaltonetworks.com
[13] https://www.shadowserver.org

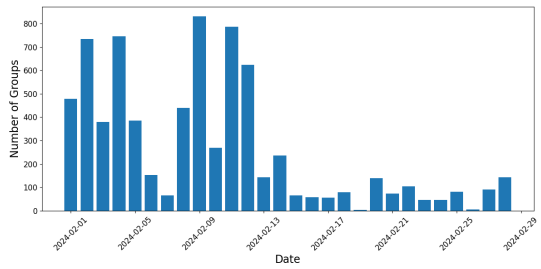

Figure 8: Number of Groups Found per Day in February 2024 (Port 443)

Group Size (Range)	Total Number of Groups
2	94
3	1727
27	68
28	410
29	745
30	4268
37	1
203 - 215	12
254	2
324	1
359	2
360	1
Total	**7331**

Figure 9: Number of Groups Found per Group Size (Range) in February 2024 (Port 443)

A total of 594 distinct sources were detected participating in distributed scanning activities on port 443. Among the 15,623 unique sources that scanned this port using ZMap, approximately 3.8% were found to be engaged in collaborative scanning activities.

AbuseIPDB analysis uncovered several findings on port 443. The Rapid7 group, consisting of 359 sources previously detected on ports 80 and 22, was also identified here. Mixed groups from DigitalOcean and Akamai/Linode, ranging in sizes from 2 to 3 and 27 to 30, were again detected, now also including certain sources from Amazon AWS[14]. Additionally, these mixed groups were observed in new sizes, specifically 324 and 360. The Gemnet group with 37 sources, each with sequential addresses, was also present. The Shadowserver Foundation group, with sizes ranging from 203 to 215, was detected here too. Furthermore, two entirely new groups were identified on port 443, which will be discussed below:

- **Google LLC**[15]: Performed two scans using the same 254 sources. These sources are all from Belgium and originate from six different subnets: `34.0.0.0/8`, `35.0.0.0/8`, `104.0.0.0/8`, `130.0.0.0/8`, `146.0.0.0/8` and `192.0.0.0/8`.
- **Amazon AWS**: Several groups with sizes 2 and 3. As mentioned earlier, some sources from Amazon AWS cover the same space with those from DigitalOcean and Akamai/Linode, leading to their inclusion in mixed groups.

4.2.5 Results for Port 8728

In the network telescope data, the most scanned port using ZMap is 8728. However, the algorithm only found one group on 10-02-2024. This group includes the sources `37.44.238.100` and `37.44.238.144`, registered under FBW Networks SAS ISP.

Further analysis of the telescope data reveals why only one group was found. Statistics show that port 8728 was scanned 142,773,561 times in February 2024. These scans were grouped by sources and sorted by scan frequency. The top 20 sources accounted for 126,215,221 scans, which is 88% of the total. This suggests that the sources primarily scan port 8728 individually rather than collaboratively.

4.3 Addressing Main Challenges in Detecting Collaborative ZMap Scanners

The sub-question addressed in this subsection is *What are the main challenges in detecting collaborative ZMap scanners and how can they be addressed?*. As mentioned in Section 3.5, the challenges encountered during the application of the first 4 steps of the methodology described in Section 3 are explained in this subsection. The challenges and the solutions applied to tackle them are discussed in the following subsections.

[14]https://aws.amazon.com
[15]https://www.google.com

4.3.1 Detecting Groups That Scan Multiple Times per Day

The first version of the algorithm described in Section 3.3 was run once a day. This version treated multiple scans per day as a single scan, creating overlaps. As a result, the algorithm couldn't detect groups that scanned more than once within a day. Figure 10 shows the number of groups found per day on port 80 before the shifting window implementation.

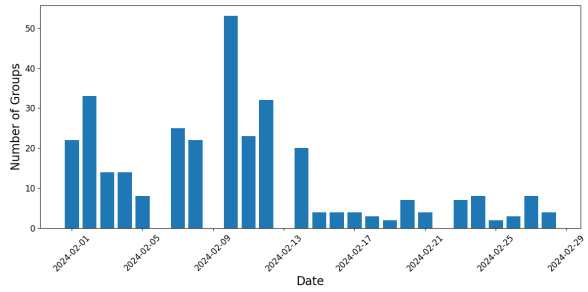

Figure 10: Number of Groups Found per Day in February 2024 (Port 80 & before shifting windows)

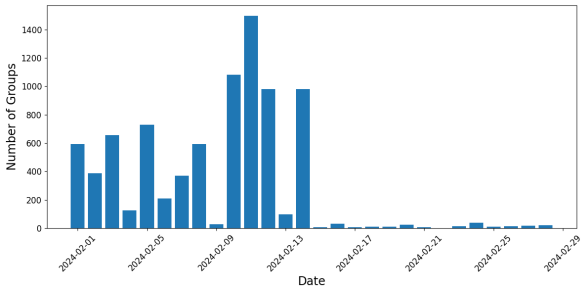

Figure 11: Number of Groups Found per Day in February 2024 (Port 80 & after shifting windows)

To fix this issue, the first attempt was to create a 3-hour window and shift it by 1 hour after each iteration. This found new groups but lost some of the previous ones. The final version of the algorithm uses shifting windows ranging from 1 hour to 24 hours and the result is shown in Figure 11. Comparing Figures 10 and 11, the difference is clear.

4.3.2 Long Execution Time of the Algorithm

Due to the large size of the telescope data, the NP-complete nature of the set cover algorithm and the many iterations caused by shifting windows of varying sizes, the execution time of the algorithm is significantly high. With an Intel Core i7-10750H processor and 16 GB of DDR4 RAM, a single run of the algorithm takes roughly 2 hours.

To tackle this problem, the algorithm is run per day and per port. This allows for parallel runs of the algorithm and reduces the size of the data for each run. After parallelization, the average execution time is reduced to approximately 1.2 hours. Furthermore, the algorithm is designed to save its progress continuously, meaning that even if the execution is interrupted, some results are still available. This is achieved by making the algorithm write the results to a file immediately after a group is found.

4.3.3 Groups with Mixed Sources

As shown in Section 4.2.2, certain sources cover the same address space, enabling the creation of multiple groups by mixing these source addresses. The challenge is identifying the actual groups and eliminating the mixed versions. Unfortunately, a viable solution to this issue could not be found within this research.

4.3.4 Missed Groups Due to Greediness

The algorithm is inspired by the greedy set cover algorithm. Due to being greedy and the randomness introduced in step 10, each run of the algorithm identifies a slightly different number of groups. One possible solution to this problem might be to run the algorithm multiple times for the same day and port, then merge the results.

However, after examining the results, the number of distinct sources remained the same for each run. This indicates that the different groups identified in different runs are just different combinations of the already identified sources. Therefore, this solution was not used in this research.

5 Responsible Research

This section reflects on the ethical aspects of this research and discusses the reproducibility of the applied methods and the results. The use of large language models (LLMs) during this research is also explained in this section.

5.1 Use of LLMs

The help of LLMs, most specifically ChatGPT[16], is utilized at various stages of this research. These include:

- Paraphrasing certain sentences during citation
- Grammar and spelling checking
- As a thesaurus to find synonyms of certain words
- Rewriting a specific sentence with other words
- Formatting references for BibTeX
- Formatting and styling LaTeX tables/figures
- Converting Python code to pseudocode
- Generating SQL queries satisfying certain properties
- Generating markdown file writing format for algorithm results
- Generating Python code for plots
- Finding useful LaTeX packages
- Debugging coding errors
- Summarizing texts to increase reading efficiency

ChatGPT is primarily used for writing assistance, such as rephrasing and formatting. No ideas generated by ChatGPT are directly used to maintain scientific integrity. The structure of the prompts used to receive ChatGPT help is included in Appendix C.

5.2 Reproducibility of the Results

To ensure reproducibility, we have provided the pseudocode of the algorithm. Furthermore, we have included the queries used and the ZMap simulation output in the appendix. The full algorithm results are available at `https://github.com/purificateur/collaborative_zmap_scans`. These can be used to validate the reproduced results.

5.3 Ethical Aspects

All of the content created by others that is used in this research is cited and included in the references. Furthermore, all findings are published and no results are discarded.

For ethical reasons, we used ZMap's simulation mode and did not perform any actual scans. This approach helps to avoid any potential misuse of the tool while still allowing us to demonstrate and validate our methodology.

[16] https://chatgpt.com

6 Discussion

The main goal of this research was to identify collaborative ZMap scanning groups. The results showed that various ISPs and cybersecurity organizations, such as DigitalOcean, Akamai, Gemnet, The Shadowserver Foundation, Palo Alto Networks, SecurityTrails, Amazon AWS, Google and Rapid7, are frequent sources of these scans. This section discusses assumptions about the possible purposes of the groups from different ISPs/organizations.

For instance, Rapid7 is a well-known cybersecurity company that often conducts research to improve network security [14]. Therefore, scans from Rapid7 are likely part of a research and pose no threat. Similarly, SecurityTrails, a data security company, might perform scans to improve their services or ensure reliability [15]. Other cybersecurity companies and organizations identified by this research include Palo Alto Networks and The Shadowserver Foundation, which probably have similar objectives.

On the other hand, scans from ISPs without a clear security or research goal, like Amazon AWS, Google, DigitalOcean, Akamai and Gemnet, might need closer inspection. The purpose of the groups belonging to these ISPs is less transparent and could include potential threats by malicious actors using their infrastructure.

These assumptions provide context to our findings and help differentiate between legitimate and potentially malicious scanning activities. This context supports our approach and suggests new ways to understand collaborative scanning activities.

Another aspect to discuss is the scalability of the method, which is important for its practical use. To make it work on larger datasets and longer periods, the method can use a smaller shifting period (less than 1 hour), larger window size (more than 24 hours) and run in iterations. This helps capture more data and find groups that take more than a day to scan or scan multiple times in an hour. Using parallel processing and distributed computing can manage the increased load. Regular updates to the algorithm can also improve its efficiency, making it possible to run continuously and adapt to new scanning patterns. This way, the method can detect collaborative ZMap scans in real-time on a larger scale.

7 Conclusions, Limitations and Future Work

This section summarizes the main findings of the research, discusses the limitations encountered during the study and proposes directions for future work. Subsection 7.1 covers the conclusions drawn from the study. The limitations that may affect the results are discussed in Subsection 7.2. Finally, potential areas for further work are covered in Subsection 7.3.

7.1 Conclusions

In conclusion, this research aimed to develop an algorithmic approach to identify collaborative ZMap scanning groups, addressing the main research question: "How can we detect collaborative ZMap scans in network telescope data using an algorithmic approach?". The methodology consisted of analyzing the characteristics of collaborative scanning activities using ZMap in network telescope data, applying set cover algorithms to distinguish collaborative scanners from other network traffic and tackling the main challenges in detecting collaborative ZMap scanners.

This research successfully addressed the challenge of detecting distributed ZMap scans by developing a specialized version of the set cover algorithm that uses the unique properties of ZMap's distributed scanning methodology. Findings indicate that this approach significantly improves the accuracy of detecting coordinated scanning activities.

7.2 Limitations

There are several limitations to this study. First, the algorithm relies on specific characteristics of ZMap scans, like non-overlapping destination address ranges, which may limit its applicability to other types of scanning tools. Second, the analysis was conducted on a limited dataset provided by the network telescope at TU Delft, which may not represent all possible scanning behaviors on the Internet. Third, the modified set cover algorithm can be slow due to multiple iterations for each window and window size, especially with large datasets. Additionally, the algorithm cannot detect groups that scan more than once per hour or span more than a day. Finally, while the algorithm successfully identified collaborative scanning activities, verifying the purpose of these groups was beyond the scope of this research.

7.3 Future Work

Future work should focus on optimizing the algorithm to handle fast scanners, larger datasets, longer time intervals and more diverse scanning strategies beyond ZMap. Additionally, integrating machine learning techniques could further enhance detection capabilities. Furthermore, exploring methods to verify the identified groups and understand their objectives is important.

References

[1] L. S. Vailshery, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030," 2023.

[2] H. Griffioen, "Scanners: Discovery of distributed slow scanners in telescope data," 2018.

[3] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zippier ZMap: Internet-Wide scanning at 10 gbps," in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, (San Diego, CA), USENIX Association, Aug. 2014.

[4] ZMap, "Getting started guide." `https://github.com/zmap/zmap/wiki/Getting-Started-Guide`, 2023. Accessed: 29-05-2024.

[5] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Network telescopes: Technical report," 2004.

[6] J. A. Filar, M. Haythorpe, and R. Taylor, "Linearly-growing reductions of karp's 21 np-complete problems," *arXiv preprint arXiv:1902.10349*, 2019.

[7] R. M. Karp, *Reducibility among combinatorial problems*. Springer, 2010.

[8] G. Cormode, H. Karloff, and A. Wirth, "Set cover algorithms for very large datasets," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 479–488, 2010.

[9] S. Robertson, E. Siegel, M. Miller, and S. Stolfo, "Surveillance detection in high bandwidth environments.," pp. 130–, 01 2003.

[10] V. Yegneswaran, P. Barford, and J. Ullrich, "Internet intrusions: Global characteristics and prevalence," vol. 31, 05 2003.

[11] C. Gates and D. U. F. of Computer Science, *Co-ordinated Port Scans: A Model, a Detector and an Evaluation Methodology*. Canadian theses, Library and Archives Canada = Bibliothèque et Archives Canada, 2006.

[12] MikroTik, "Mikrotik documentation." `https://wiki.mikrotik.com/wiki/Manual:IP/Services`, 2022. Accessed: 29-05-2024.

[13] IANA, "Service name and transport protocol port number registry." `https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml`. Accessed: 29-05-2024.

[14] Rapid7, "Rapid7 cybersecurity research." `https://www.rapid7.com/research/`. Accessed: 01-06-2024.

[15] SecurityTrails, "Securitytrails api." `https://securitytrails.com/corp/api`. Accessed: 01-06-2024.

A Network Telescope Data Analysis Queries

A.1 Total Scan Count

```
SELECT Count(*)
FROM tcppackets;
```

A.2 Total ZMap Scan Count

```
SELECT Count(*)
FROM tcppackets
WHERE IPId == '54321'
```

A.3 ZMap Scan Count per Destination Address

```
SELECT IPv4NumToString(DstIP) AS DstIP, COUNT(*) AS EntryCount
FROM tcppackets
WHERE IPId == '54321'
GROUP BY DstIP
ORDER BY EntryCount;
```

A.4 Filter Destination Address Noise

```
SELECT IPv4NumToString(DstIP) AS DstIP
FROM tcppackets
WHERE IPId == '54321'
GROUP BY DstIP
HAVING COUNT(*) > 1000;
```

A.5 Top 10 Scanned Ports Using ZMap

```
SELECT DstPort, COUNT(*) AS EntryCount
FROM tcppackets
WHERE IPId == '54321'
GROUP BY DstPort
ORDER BY EntryCount DESC
LIMIT 10;
```

B Algorithm Query

```
WITH denoised_dstips AS (
    SELECT IPv4NumToString(DstIP) AS DstIP
    FROM tcppackets
    WHERE IPId = '54321'
    GROUP BY DstIP
    HAVING COUNT(*) > 1000
)
SELECT IPv4NumToString(SrcIP) AS SrcIPv4,
    groupArray(IPv4NumToString(DstIP)) AS DstIPv4s
FROM tcppackets
WHERE
    IPId = '54321'
    AND DstPort = '{port}'
    AND toDate(Timestamp) = '{day}'
    AND toHour(Timestamp) >= {start_hour}
    AND toHour(Timestamp) < {start_hour + window_size}
    AND IPv4NumToString(DstIP) IN (SELECT DstIP FROM denoised_dstips)
GROUP BY SrcIP
ORDER BY SrcIP
```

C Used ChatGPT Prompt Structures

- Paraphrase <sentence>
- Check the following text for grammar and spelling errors: <text>
- What can I say instead of <word> in this sentence: <sentence>
- Rewrite using different words, please avoid fancy ones: <sentence>
- Format <reference> for BibTeX
- Format/Style <content> as LaTeX table/figure
- Convert the following code to pseudocode and format for LaTeX: <code>
- Generate SQL query that gets/filters/counts <property>
- Generate code that writes the results of the following code to a markdown table: <code>
- Generate Python code that plots the result of this query/code using bar chart: <query/code>
- What is the LaTeX package for <purpose>?
- How to fix this: <code error>
- Summarize <text>

D Example Groups with Mixed Sources

Source IPv4	Covers
107.170.234.27	11998
192.241.197.44	10496
192.241.219.29	39473

Table 2: Set 3 - Source Count: 3 - Total Cover: 61967 - Window: 4

Source IPv4	Covers
94.102.61.80	10496
107.170.234.27	11998
192.241.219.29	39473

Table 3: Set 5 - Source Count: 3 - Total Cover: 61967 - Window: 2

Source IPv4	Covers
107.170.234.27	11998
137.184.255.7	10496
192.241.219.29	39473

Table 4: Set 6 - Source Count: 3 - Total Cover: 61967 - Window: 4

Source IPv4	Covers
192.241.197.44	10496
192.241.207.8	11998
192.241.219.29	39473

Table 5: Set 8 - Source Count: 3 - Total Cover: 61967 - Window: 4

E ZMap Distributed Scan Simulation Outputs (Dryrun)

E.1 Source 1: 192.168.1.56

```
tcp { source: 51786 | dest: 80 | seq: 3798257696 | checksum: 0X7ED3 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.203 | checksum: 0X33AF }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45597 | dest: 80 | seq: 2689054289 | checksum: 0XF2F8 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.191 | checksum: 0X33BB }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56253 | dest: 80 | seq: 3735041688 | checksum: 0XB02 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.118 | checksum: 0X3404 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56253 | dest: 80 | seq: 3735041688 | checksum: 0XA8B }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.237 | checksum: 0X338D }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45597 | dest: 80 | seq: 2689054289 | checksum: 0XF358 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.95 | checksum: 0X341B }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56253 | dest: 80 | seq: 3735041688 | checksum: 0XB09 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.111 | checksum: 0X340B }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58096 | dest: 80 | seq: 1656753345 | checksum: 0XA966 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.150 | checksum: 0X33E4 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58096 | dest: 80 | seq: 1656753345 | checksum: 0XA9CD }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.47 | checksum: 0X344B }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56253 | dest: 80 | seq: 3735041688 | checksum: 0XA9E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.218 | checksum: 0X33A0 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56253 | dest: 80 | seq: 3735041688 | checksum: 0XB41 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.55 | checksum: 0X3443 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47980 | dest: 80 | seq: 3357628959 | checksum: 0X1E43 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.126 | checksum: 0X33FC }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47980 | dest: 80 | seq: 3357628959 | checksum: 0X1DE2 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.223 | checksum: 0X339B }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 32861 | dest: 80 | seq: 3114248976 | checksum: 0X1735 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.44 | checksum: 0X344E }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56763 | dest: 80 | seq: 1319936220 | checksum: 0X2ED1 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.89 | checksum: 0X3421 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 34918 | dest: 80 | seq: 3023397785 | checksum: 0X5B7B }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.190 | checksum: 0X33BC }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33630 | dest: 80 | seq: 1950960671 | checksum: 0XB7F4 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.179 | checksum: 0X33C7 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 48194 | dest: 80 | seq: 1803573742 | checksum: 0X7A09 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.180 | checksum: 0X33C6 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60984 | dest: 80 | seq: 332708811 | checksum: 0X3E2E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.104 | checksum: 0X3412 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50872 | dest: 80 | seq: 2389664177 | checksum: 0X4519 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.124 | checksum: 0X33FE }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 46400 | dest: 80 | seq: 3531905268 | checksum: 0XDAB7 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.253 | checksum: 0X337D }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 59520 | dest: 80 | seq: 2647947658 | checksum: 0XFB95 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.249 | checksum: 0X3381 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60192 | dest: 80 | seq: 1125129913 | checksum: 0XAEE2 }
```

```
ip { saddr: 192.168.1.56 | daddr: 171.67.70.162 | checksum: 0X33D8 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 59520 | dest: 80 | seq: 2647947658 | checksum: 0XFB9A }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.244 | checksum: 0X3386 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 49749 | dest: 80 | seq: 4225133490 | checksum: 0XD1C2 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.205 | checksum: 0X33AD }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 48514 | dest: 80 | seq: 1235262468 | checksum: 0X584D }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.250 | checksum: 0X3380 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58561 | dest: 80 | seq: 952436680 | checksum: 0XD6B9 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.102 | checksum: 0X3414 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 59520 | dest: 80 | seq: 2647947658 | checksum: 0XFC4A }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.68 | checksum: 0X3436 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58561 | dest: 80 | seq: 952436680 | checksum: 0XD637 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.232 | checksum: 0X3392 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58561 | dest: 80 | seq: 952436680 | checksum: 0XD64E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.209 | checksum: 0X33A9 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58068 | dest: 80 | seq: 88015188 | checksum: 0X1215 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.242 | checksum: 0X3388 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53633 | dest: 80 | seq: 1273436464 | checksum: 0XC5A1 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.52 | checksum: 0X3446 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58670 | dest: 80 | seq: 2875739524 | checksum: 0X1DC3 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.144 | checksum: 0X33EA }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58670 | dest: 80 | seq: 2875739524 | checksum: 0X1DCA }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.137 | checksum: 0X33F1 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58813 | dest: 80 | seq: 3388826133 | checksum: 0XEA03 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.154 | checksum: 0X33E0 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50644 | dest: 80 | seq: 351883776 | checksum: 0XCF8E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.18 | checksum: 0X3468 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53193 | dest: 80 | seq: 708586810 | checksum: 0XD4C5 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.105 | checksum: 0X3411 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58813 | dest: 80 | seq: 3388826133 | checksum: 0XE9FD }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.160 | checksum: 0X33DA }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50644 | dest: 80 | seq: 351883776 | checksum: 0XCEBC }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.228 | checksum: 0X3396 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 48754 | dest: 80 | seq: 877292123 | checksum: 0X9E8E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.200 | checksum: 0X33B2 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 35265 | dest: 80 | seq: 4222468613 | checksum: 0XB429 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.208 | checksum: 0X33AA }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33557 | dest: 80 | seq: 2583427566 | checksum: 0XE549 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.37 | checksum: 0X3455 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33557 | dest: 80 | seq: 2583427566 | checksum: 0XE54E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.32 | checksum: 0X345A }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50339 | dest: 80 | seq: 2982737595 | checksum: 0X8E73 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.211 | checksum: 0X33A7 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33557 | dest: 80 | seq: 2583427566 | checksum: 0XE4D7 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.151 | checksum: 0X33E3 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
```

```
-----------------------------------------------------
tcp { source: 56939 | dest: 80 | seq: 1937378888 | checksum: 0X9BFF }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.65 | checksum: 0X3439 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33557 | dest: 80 | seq: 2583427566 | checksum: 0XE4ED }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.129 | checksum: 0X33F9 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33557 | dest: 80 | seq: 2583427566 | checksum: 0XE52E }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.64 | checksum: 0X343A }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55506 | dest: 80 | seq: 2699472707 | checksum: 0XD326 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.75 | checksum: 0X342F }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50410 | dest: 80 | seq: 3775310419 | checksum: 0XAB83 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.166 | checksum: 0X33D4 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 44800 | dest: 80 | seq: 3802159136 | checksum: 0X12AB }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.2 | checksum: 0X3478 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58126 | dest: 80 | seq: 1242142410 | checksum: 0X3887 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.5 | checksum: 0X3475 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53487 | dest: 80 | seq: 2454684606 | checksum: 0X1525 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.76 | checksum: 0X342E }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 42522 | dest: 80 | seq: 2955027442 | checksum: 0X81A6 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.153 | checksum: 0X33E1 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 35717 | dest: 80 | seq: 3756876206 | checksum: 0X2ECA }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.131 | checksum: 0X33F7 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47697 | dest: 80 | seq: 3008254406 | checksum: 0X3CFB }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.13 | checksum: 0X346D }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55814 | dest: 80 | seq: 2323841952 | checksum: 0X93E7 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.93 | checksum: 0X341D }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43901 | dest: 80 | seq: 3233793834 | checksum: 0XC8D4 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.50 | checksum: 0X3448 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 36991 | dest: 80 | seq: 891640282 | checksum: 0XDCE9 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.6 | checksum: 0X3474 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53206 | dest: 80 | seq: 2165429684 | checksum: 0XD5A3 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.46 | checksum: 0X344C }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55388 | dest: 80 | seq: 933389250 | checksum: 0X88A2 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.11 | checksum: 0X346F }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 49224 | dest: 80 | seq: 2650954550 | checksum: 0X42D6 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.23 | checksum: 0X3463 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53889 | dest: 80 | seq: 2080569720 | checksum: 0XB40C }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.101 | checksum: 0X3415 }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 49224 | dest: 80 | seq: 2650954550 | checksum: 0X4232 }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.187 | checksum: 0X33BF }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60162 | dest: 80 | seq: 3061082942 | checksum: 0XEF9D }
ip { saddr: 192.168.1.56 | daddr: 171.67.70.27 | checksum: 0X345F }
eth { shost: 50:2f:9b:d0:9c:b0 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
```

E.2 Source 2: 192.168.1.57

```
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE60B }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.1 | checksum: 0X3479 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 59648 | dest: 80 | seq: 1987898245 | checksum: 0XB156 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.21 | checksum: 0X3465 }
```

```
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE602 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.10 | checksum: 0X3470 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE58C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.128 | checksum: 0X33FA }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE52F }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.221 | checksum: 0X339D }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE548 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.196 | checksum: 0X33B6 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE60C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.0 | checksum: 0X347A }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE52A }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.226 | checksum: 0X3398 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33163 | dest: 80 | seq: 1249004132 | checksum: 0XE59E }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.110 | checksum: 0X340C }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43090 | dest: 80 | seq: 2031870982 | checksum: 0XF6CE }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.43 | checksum: 0X344F }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50270 | dest: 80 | seq: 1707644781 | checksum: 0X3AB9 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.33 | checksum: 0X3459 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 38534 | dest: 80 | seq: 771297788 | checksum: 0X2544 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.175 | checksum: 0X33CB }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55998 | dest: 80 | seq: 2117777890 | checksum: 0XE91E }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.116 | checksum: 0X3406 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45633 | dest: 80 | seq: 317075799 | checksum: 0X5E9 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.7 | checksum: 0X3473 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45633 | dest: 80 | seq: 317075799 | checksum: 0X568 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.136 | checksum: 0X33F2 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 59229 | dest: 80 | seq: 3556548732 | checksum: 0XA094 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.3 | checksum: 0X3477 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 38534 | dest: 80 | seq: 771297788 | checksum: 0X2539 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.186 | checksum: 0X33C0 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50270 | dest: 80 | seq: 1707644781 | checksum: 0X3A83 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.87 | checksum: 0X3423 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33873 | dest: 80 | seq: 1872616302 | checksum: 0X2D01 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.16 | checksum: 0X346A }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50270 | dest: 80 | seq: 1707644781 | checksum: 0X39F1 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.233 | checksum: 0X3391 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 51754 | dest: 80 | seq: 1306728666 | checksum: 0XCB28 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.94 | checksum: 0X341C }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43872 | dest: 80 | seq: 3102157227 | checksum: 0X6E6C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.15 | checksum: 0X346B }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 39395 | dest: 80 | seq: 1227631176 | checksum: 0XEE44 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.210 | checksum: 0X33A8 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56179 | dest: 80 | seq: 3866624746 | checksum: 0X3711 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.135 | checksum: 0X33F3 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
```

```
tcp { source: 51854 | dest: 80 | seq: 2354785647 | checksum: 0X77D2 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.67 | checksum: 0X3437 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43442 | dest: 80 | seq: 762806714 | checksum: 0XA34B }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.63 | checksum: 0X343B }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 57492 | dest: 80 | seq: 1381757260 | checksum: 0XD812 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.31 | checksum: 0X345B }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37651 | dest: 80 | seq: 21777155 | checksum: 0X1C4F }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.189 | checksum: 0X33BD }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 52497 | dest: 80 | seq: 1211925849 | checksum: 0X614D }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.122 | checksum: 0X3400 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 40172 | dest: 80 | seq: 2729369432 | checksum: 0XDCD6 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.164 | checksum: 0X33D6 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50567 | dest: 80 | seq: 2408071123 | checksum: 0X676E }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.29 | checksum: 0X345D }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37973 | dest: 80 | seq: 3965900710 | checksum: 0XA31A }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.245 | checksum: 0X3385 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37973 | dest: 80 | seq: 3965900710 | checksum: 0XA401 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.14 | checksum: 0X346C }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37973 | dest: 80 | seq: 3965900710 | checksum: 0XA3C7 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.72 | checksum: 0X3432 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55429 | dest: 80 | seq: 343483012 | checksum: 0XEC01 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.234 | checksum: 0X3390 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 52578 | dest: 80 | seq: 2353620105 | checksum: 0X3E17 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.34 | checksum: 0X3458 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50567 | dest: 80 | seq: 2408071123 | checksum: 0X66FA }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.145 | checksum: 0X33E9 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37973 | dest: 80 | seq: 3965900710 | checksum: 0XA310 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.255 | checksum: 0X337B }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56734 | dest: 80 | seq: 4007950585 | checksum: 0XB672 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.127 | checksum: 0X33FB }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56734 | dest: 80 | seq: 4007950585 | checksum: 0XB6B6 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.59 | checksum: 0X343F }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41581 | dest: 80 | seq: 1566137022 | checksum: 0XA0EB }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.254 | checksum: 0X337C }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 34258 | dest: 80 | seq: 575173219 | checksum: 0XE11C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.212 | checksum: 0X33A6 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41581 | dest: 80 | seq: 1566137022 | checksum: 0XA1A4 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.69 | checksum: 0X3435 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 34258 | dest: 80 | seq: 575173219 | checksum: 0XE0F4 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.252 | checksum: 0X337E }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58496 | dest: 80 | seq: 449316695 | checksum: 0XF57E }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.80 | checksum: 0X342A }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 52072 | dest: 80 | seq: 2456534980 | checksum: 0XDE4A }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.139 | checksum: 0X33EF }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45763 | dest: 80 | seq: 3489263201 | checksum: 0X8AA7 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.168 | checksum: 0X33D2 }
```

```
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 35396 | dest: 80 | seq: 1004004859 | checksum: 0X4F68 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.239 | checksum: 0X338B }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 56720 | dest: 80 | seq: 3001379264 | checksum: 0X1C1 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.119 | checksum: 0X3403 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 55177 | dest: 80 | seq: 2833014208 | checksum: 0X1D50 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.248 | checksum: 0X3382 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 53607 | dest: 80 | seq: 1117423166 | checksum: 0X6218 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.22 | checksum: 0X3464 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 53607 | dest: 80 | seq: 1117423166 | checksum: 0X618D }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.161 | checksum: 0X33D9 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 60763 | dest: 80 | seq: 2389170752 | checksum: 0XA58C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.222 | checksum: 0X339C }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 56155 | dest: 80 | seq: 2586093551 | checksum: 0XDE42 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.188 | checksum: 0X33BE }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 34894 | dest: 80 | seq: 1363052498 | checksum: 0X9AE2 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.45 | checksum: 0X344D }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 55467 | dest: 80 | seq: 3403621095 | checksum: 0X351C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.224 | checksum: 0X339A }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 53441 | dest: 80 | seq: 3665153506 | checksum: 0X8312 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.66 | checksum: 0X3438 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 38902 | dest: 80 | seq: 623463314 | checksum: 0XF344 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.120 | checksum: 0X3402 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 59341 | dest: 80 | seq: 1956173405 | checksum: 0XC8BA }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.240 | checksum: 0X338A }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 46905 | dest: 80 | seq: 48928669 | checksum: 0XAA52 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.91 | checksum: 0X341F }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 36923 | dest: 80 | seq: 1436665182 | checksum: 0X507C }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.183 | checksum: 0X33C3 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 36923 | dest: 80 | seq: 1436665182 | checksum: 0X50AE }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.133 | checksum: 0X33F5 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 44435 | dest: 80 | seq: 4053035510 | checksum: 0XF28F }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.192 | checksum: 0X33BA }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 54880 | dest: 80 | seq: 1753773646 | checksum: 0X4646 }
ip { saddr: 192.168.1.57 | daddr: 171.67.70.241 | checksum: 0X3389 }
eth { shost: 50:2f:9b:d0:9c:b1 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
```

E.3 Source 3: 192.168.1.58

```
tcp { source: 50425 | dest: 80 | seq: 2526872581 | checksum: 0X9CB8 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.26 | checksum: 0X3460 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 52235 | dest: 80 | seq: 2119282983 | checksum: 0X9B }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.79 | checksum: 0X342B }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 58506 | dest: 80 | seq: 401994053 | checksum: 0XDD0 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.217 | checksum: 0X33A1 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 58506 | dest: 80 | seq: 401994053 | checksum: 0XE74 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.53 | checksum: 0X3445 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 60275 | dest: 80 | seq: 2358380079 | checksum: 0X7E12 }
```

```
ip { saddr: 192.168.1.58 | daddr: 171.67.70.39 | checksum: 0X3453 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37120 | dest: 80 | seq: 1252608835 | checksum: 0XD52F }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.82 | checksum: 0X3428 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 54787 | dest: 80 | seq: 2056019524 | checksum: 0X4D0D }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.141 | checksum: 0X33ED }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37434 | dest: 80 | seq: 3882798968 | checksum: 0XB2A3 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.169 | checksum: 0X33D1 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47331 | dest: 80 | seq: 2446304364 | checksum: 0XCB0 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.159 | checksum: 0X33DB }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 54787 | dest: 80 | seq: 2056019524 | checksum: 0X4CE8 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.178 | checksum: 0X33C8 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 51475 | dest: 80 | seq: 1426048126 | checksum: 0X1914 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.201 | checksum: 0X33B1 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 51475 | dest: 80 | seq: 1426048126 | checksum: 0X1993 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.74 | checksum: 0X3430 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33159 | dest: 80 | seq: 586357909 | checksum: 0X3ABA }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.165 | checksum: 0X33D5 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33159 | dest: 80 | seq: 586357909 | checksum: 0X3B21 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.62 | checksum: 0X343C }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41948 | dest: 80 | seq: 2008529874 | checksum: 0X2473 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.149 | checksum: 0X33E5 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33159 | dest: 80 | seq: 586357909 | checksum: 0X3AFB }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.100 | checksum: 0X3416 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33159 | dest: 80 | seq: 586357909 | checksum: 0X3ACD }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.146 | checksum: 0X33E8 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33159 | dest: 80 | seq: 586357909 | checksum: 0X3A88 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.215 | checksum: 0X33A3 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33159 | dest: 80 | seq: 586357909 | checksum: 0X3AF4 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.107 | checksum: 0X340F }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 48302 | dest: 80 | seq: 3063901458 | checksum: 0X1B91 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.125 | checksum: 0X33FD }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53245 | dest: 80 | seq: 2823412189 | checksum: 0XA94E }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.251 | checksum: 0X337F }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47506 | dest: 80 | seq: 3943255207 | checksum: 0XA7D }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.174 | checksum: 0X33CC }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 46435 | dest: 80 | seq: 239129420 | checksum: 0X6488 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.246 | checksum: 0X3384 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 46431 | dest: 80 | seq: 1635390525 | checksum: 0XD06B }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.236 | checksum: 0X338E }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43923 | dest: 80 | seq: 3022034997 | checksum: 0X3CC }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.185 | checksum: 0X33C1 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45595 | dest: 80 | seq: 3231799491 | checksum: 0X30C4 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.42 | checksum: 0X3450 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43316 | dest: 80 | seq: 348533191 | checksum: 0XC51 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.92 | checksum: 0X341E }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
```

```
-----------------------------------------------------
tcp { source: 52892 | dest: 80 | seq: 1321619991 | checksum: 0X8CAB }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.73 | checksum: 0X3431 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60623 | dest: 80 | seq: 1884510285 | checksum: 0X46DA }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.36 | checksum: 0X3456 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 44528 | dest: 80 | seq: 483875644 | checksum: 0XD615 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.85 | checksum: 0X3425 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 59277 | dest: 80 | seq: 2308359598 | checksum: 0XC4F0 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.171 | checksum: 0X33CF }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60179 | dest: 80 | seq: 1916337978 | checksum: 0X9F54 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.147 | checksum: 0X33E7 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 35090 | dest: 80 | seq: 4251444301 | checksum: 0X90CE }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.216 | checksum: 0X33A2 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47607 | dest: 80 | seq: 2153867651 | checksum: 0X583C }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.86 | checksum: 0X3424 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 49313 | dest: 80 | seq: 2841994805 | checksum: 0X2C0C }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.38 | checksum: 0X3454 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 54173 | dest: 80 | seq: 2665171660 | checksum: 0X3E7C }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.173 | checksum: 0X33CD }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 51145 | dest: 80 | seq: 2992364379 | checksum: 0XA6A0 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.77 | checksum: 0X342D }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41857 | dest: 80 | seq: 3153778540 | checksum: 0XC513 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.114 | checksum: 0X3408 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 39518 | dest: 80 | seq: 3906620448 | checksum: 0X2C65 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.176 | checksum: 0X33CA }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33623 | dest: 80 | seq: 1539567687 | checksum: 0X2C27 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.229 | checksum: 0X3395 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 52398 | dest: 80 | seq: 398283846 | checksum: 0XC35C }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.96 | checksum: 0X341A }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55271 | dest: 80 | seq: 473872383 | checksum: 0X4F7F }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.202 | checksum: 0X33B0 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47488 | dest: 80 | seq: 204710524 | checksum: 0X93E4 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.90 | checksum: 0X3420 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 44507 | dest: 80 | seq: 1221288121 | checksum: 0XA44D }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.193 | checksum: 0X33B9 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 47477 | dest: 80 | seq: 292119700 | checksum: 0XCC67 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.148 | checksum: 0X33E6 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41219 | dest: 80 | seq: 2488758912 | checksum: 0X546B }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.40 | checksum: 0X3452 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 42794 | dest: 80 | seq: 3135465439 | checksum: 0X31FF }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.130 | checksum: 0X33F8 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 51524 | dest: 80 | seq: 1939254871 | checksum: 0X10A1 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.155 | checksum: 0X33DF }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 54242 | dest: 80 | seq: 3612724677 | checksum: 0X833A }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.54 | checksum: 0X3444 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 34588 | dest: 80 | seq: 940275019 | checksum: 0XC746 }
```

```
ip { saddr: 192.168.1.58 | daddr: 171.67.70.181 | checksum: 0X33C5 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43384 | dest: 80 | seq: 867898915 | checksum: 0X8AE }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.106 | checksum: 0X3410 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55815 | dest: 80 | seq: 2741289869 | checksum: 0XBA99 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.219 | checksum: 0X339F }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 34661 | dest: 80 | seq: 944351116 | checksum: 0X9489 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.170 | checksum: 0X33D0 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43384 | dest: 80 | seq: 867898915 | checksum: 0X914 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.4 | checksum: 0X3476 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55815 | dest: 80 | seq: 2741289869 | checksum: 0XBB07 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.109 | checksum: 0X340D }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43384 | dest: 80 | seq: 867898915 | checksum: 0X843 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.213 | checksum: 0X33A5 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 38572 | dest: 80 | seq: 706521087 | checksum: 0X92F0 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.182 | checksum: 0X33C4 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33707 | dest: 80 | seq: 2162816133 | checksum: 0X303 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.81 | checksum: 0X3429 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37347 | dest: 80 | seq: 3985820934 | checksum: 0XAFE8 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.9 | checksum: 0X3471 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 34332 | dest: 80 | seq: 2165375244 | checksum: 0XF391 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.163 | checksum: 0X33D7 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56952 | dest: 80 | seq: 1270458882 | checksum: 0X2827 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.19 | checksum: 0X3467 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 42701 | dest: 80 | seq: 3938311422 | checksum: 0X8D90 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.84 | checksum: 0X3426 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55742 | dest: 80 | seq: 1095768168 | checksum: 0XC8B1 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.70 | checksum: 0X3434 }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 46369 | dest: 80 | seq: 67260892 | checksum: 0XF102 }
ip { saddr: 192.168.1.58 | daddr: 171.67.70.108 | checksum: 0X340E }
eth { shost: 50:2f:9b:d0:9c:b2 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
```

E.4 Source 4: 192.168.1.59

```
tcp { source: 53406 | dest: 80 | seq: 3100386348 | checksum: 0X4DEC }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.235 | checksum: 0X338F }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50165 | dest: 80 | seq: 731736853 | checksum: 0XA15F }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.103 | checksum: 0X3413 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 38133 | dest: 80 | seq: 2503428845 | checksum: 0X8686 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.206 | checksum: 0X33AC }
eth { shost: 50:2f:9b:d0:9c:bb3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53096 | dest: 80 | seq: 559931344 | checksum: 0X2B1E }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.184 | checksum: 0X33C2 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50165 | dest: 80 | seq: 731736853 | checksum: 0XA0F7 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.207 | checksum: 0X33AB }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53406 | dest: 80 | seq: 3100386348 | checksum: 0X4E01 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.214 | checksum: 0X33A4 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53096 | dest: 80 | seq: 559931344 | checksum: 0X2BA3 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.51 | checksum: 0X3447 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
```

```
tcp { source: 53406 | dest: 80 | seq: 3100386348 | checksum: 0X4E2B }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.172 | checksum: 0X33CE }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53406 | dest: 80 | seq: 3100386348 | checksum: 0X4DF0 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.231 | checksum: 0X3393 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50165 | dest: 80 | seq: 731736853 | checksum: 0XA0EA }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.220 | checksum: 0X339E }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 45777 | dest: 80 | seq: 3512402537 | checksum: 0X753C }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.156 | checksum: 0X33DE }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33068 | dest: 80 | seq: 2302179923 | checksum: 0X76D5 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.225 | checksum: 0X3399 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60059 | dest: 80 | seq: 3515623586 | checksum: 0X1768 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.60 | checksum: 0X343E }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 54910 | dest: 80 | seq: 442326405 | checksum: 0XADB5 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.88 | checksum: 0X3422 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41330 | dest: 80 | seq: 3656606637 | checksum: 0X1C99 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.194 | checksum: 0X33B8 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 60013 | dest: 80 | seq: 3656448002 | checksum: 0X3FD5 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.56 | checksum: 0X3442 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 32849 | dest: 80 | seq: 3760803072 | checksum: 0X4E82 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.113 | checksum: 0X3409 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56042 | dest: 80 | seq: 1457454299 | checksum: 0XCD18 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.177 | checksum: 0X33C9 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41330 | dest: 80 | seq: 3656606637 | checksum: 0X1CBD }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.158 | checksum: 0X33DC }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33313 | dest: 80 | seq: 3814134521 | checksum: 0X838C }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.112 | checksum: 0X340A }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 49174 | dest: 80 | seq: 367666394 | checksum: 0X132 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.98 | checksum: 0X3418 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56338 | dest: 80 | seq: 2388470852 | checksum: 0X64F5 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.197 | checksum: 0X33B5 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 49769 | dest: 80 | seq: 1928128894 | checksum: 0XDCB6 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.227 | checksum: 0X3397 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 39675 | dest: 80 | seq: 1537615312 | checksum: 0XDFF4 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.8 | checksum: 0X3472 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56110 | dest: 80 | seq: 993244318 | checksum: 0X307B }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.243 | checksum: 0X3387 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37876 | dest: 80 | seq: 1593117330 | checksum: 0XFE0C }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.230 | checksum: 0X3394 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 48307 | dest: 80 | seq: 3230398355 | checksum: 0X8711 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.138 | checksum: 0X33F0 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 37428 | dest: 80 | seq: 2184228245 | checksum: 0X39AE }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.198 | checksum: 0X33B4 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50042 | dest: 80 | seq: 4144465711 | checksum: 0XBA38 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.132 | checksum: 0X33F6 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 33665 | dest: 80 | seq: 525248497 | checksum: 0XB134 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.121 | checksum: 0X3401 }
```

```
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 60986 | dest: 80 | seq: 1778540276 | checksum: 0X44B1 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.140 | checksum: 0X33EE }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 40857 | dest: 80 | seq: 3815843109 | checksum: 0X542A }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.20 | checksum: 0X3466 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 40698 | dest: 80 | seq: 3302822378 | checksum: 0X8694 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.24 | checksum: 0X3462 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 34053 | dest: 80 | seq: 107019456 | checksum: 0X73B0 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.152 | checksum: 0X33E2 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 57990 | dest: 80 | seq: 2999200501 | checksum: 0X3C0B }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.35 | checksum: 0X3457 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 40610 | dest: 80 | seq: 3324061649 | checksum: 0X6F92 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.71 | checksum: 0X3433 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 36233 | dest: 80 | seq: 2661246183 | checksum: 0X6B2D }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.49 | checksum: 0X3449 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 35240 | dest: 80 | seq: 1060496511 | checksum: 0X4F00 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.17 | checksum: 0X3469 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 52562 | dest: 80 | seq: 3089158389 | checksum: 0XA5DC }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.41 | checksum: 0X3451 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 38367 | dest: 80 | seq: 1682384701 | checksum: 0XDA3E }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.204 | checksum: 0X33AE }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 55273 | dest: 80 | seq: 24435561 | checksum: 0X4754 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.83 | checksum: 0X3427 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 33738 | dest: 80 | seq: 3774148512 | checksum: 0XA7AB }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.99 | checksum: 0X3417 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 35060 | dest: 80 | seq: 2615529968 | checksum: 0X4FD }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.167 | checksum: 0X33D3 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 39523 | dest: 80 | seq: 465570445 | checksum: 0X3BA0 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.30 | checksum: 0X345C }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 36306 | dest: 80 | seq: 1769960327 | checksum: 0X9134 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.97 | checksum: 0X3419 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 54018 | dest: 80 | seq: 1554010153 | checksum: 0X7C72 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.48 | checksum: 0X344A }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 58174 | dest: 80 | seq: 825872121 | checksum: 0X18AF }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.78 | checksum: 0X342C }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 51259 | dest: 80 | seq: 1774365903 | checksum: 0X1CDE }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.195 | checksum: 0X33B7 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 58027 | dest: 80 | seq: 916931120 | checksum: 0XA024 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.199 | checksum: 0X33B3 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 47984 | dest: 80 | seq: 3297955130 | checksum: 0XAEA1 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.143 | checksum: 0X33EB }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 47473 | dest: 80 | seq: 2323657021 | checksum: 0X86B1 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.142 | checksum: 0X33EC }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
tcp { source: 40110 | dest: 80 | seq: 436207962 | checksum: 0X444A }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.28 | checksum: 0X345E }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-------------------------------------------------------
```

```
tcp { source: 53382 | dest: 80 | seq: 2098055148 | checksum: 0XE6B0 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.61 | checksum: 0X343D }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 50762 | dest: 80 | seq: 355885650 | checksum: 0XBDFE }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.157 | checksum: 0X33DD }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 35424 | dest: 80 | seq: 410064344 | checksum: 0X43B9 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.12 | checksum: 0X346E }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 43981 | dest: 80 | seq: 2664364631 | checksum: 0XB6FF }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.123 | checksum: 0X33FF }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 41460 | dest: 80 | seq: 3684672184 | checksum: 0XDB2A }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.247 | checksum: 0X3383 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 53507 | dest: 80 | seq: 1624098465 | checksum: 0X3C3 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.57 | checksum: 0X3441 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 58942 | dest: 80 | seq: 3201593940 | checksum: 0XE4CC }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.58 | checksum: 0X3440 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 55129 | dest: 80 | seq: 3265447557 | checksum: 0X9B66 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.134 | checksum: 0X33F4 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 36851 | dest: 80 | seq: 498836926 | checksum: 0XA88C }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.117 | checksum: 0X3405 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 36851 | dest: 80 | seq: 498836926 | checksum: 0XA8E8 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.25 | checksum: 0X3461 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 56357 | dest: 80 | seq: 1934302446 | checksum: 0X8F21 }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.238 | checksum: 0X338C }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
tcp { source: 36851 | dest: 80 | seq: 498836926 | checksum: 0XA88E }
ip { saddr: 192.168.1.59 | daddr: 171.67.70.115 | checksum: 0X3407 }
eth { shost: 50:2f:9b:d0:9c:b3 | dhost: c8:54:4b:92:3c:a0 }
-----------------------------------------------------
```