UNIVERSITY OF AMSTERDAM
SYSTEM AND NETWORK ENGINEERING

Research Project 2:

Identifying Patterns in DNS Traffic

Using Visual Analytics to Discover DNS Abuse

Author:
Pieter Lexis
pieter.lexis@os3.nl

Supervisor:
Matthijs Mekking
matthijs@nlnetlabs.nl

Abstract

In this research, a visual analytics approach is used on a large set of DNS packet captures to gain insight into ways that authoritative name servers are abused for denial of service attacks. Several tools were developed to identify patterns in DNS queries and responses. These patterns revealed that source port selection by recursive name servers is not uniformly distributed and that attackers are using a diffuse pattern of query names to defeat anti-amplification measures implemented in nameservers.

Keywords: Domain Name System, Visualization, Response Rate Limiting, Abuse

July 9, 2013

1 Introduction

The Domain Name System (DNS) [18, 19] has recently been abused to perform the biggest Denial of Service (DoS) attacks in the history of the Internet [22]. By using a so-called reflected DNS amplification attack on open resolvers, the attackers managed to sustain a rate of 300 Gigabits per second.

DNS amplification attacks are a form of distributed denial of service (DDoS) attacks, where name servers are abused to send large answers to DNS queries to the victim [28]. This is accomplished by sending a small query that will result in a large response, using a spoofed source IP address to a name server. The ratio between the response and query size is the amplification factor. The name server then responds with this large answer to the spoofed IP address, potentially flooding the link towards that host.

Note that amplification attacks can be achieved using (almost) all stateless, UDP based protocols. For example, SNMP amplification attacks [27]. This paper limits itself to DNS amplification attacks.

DNS amplification has a long history. The recent growth in DNSSEC secured domains takes this problem to a new level, however. A small query can return an even larger response, now also containing the associated DNSSEC signatures, yielding a higher amplification factor. As for instance shown by Daniel Bernstein in 2010, where he had a 51× amplification in an in vivo experiment [3]. The greatest amplifiction is achieved by using the query type `ANY`, as the name server returns many types of records associated with the name. For instance, an `ANY` query for the zone-apex will return, among others, `DNSKEY`, `SOA`, `MX` and `NS` records with their associated `RRSIG` records.

A graphical overview of this kind of attack abusing recursive name servers is shown in figure 1.

Both recursive and authoritative name servers are abused in a similar fashion to achieve these kinds of attacks. There are several methods that allow a DNS server operator to mitigate the abuse by detecting patterns in the incoming queries.

Preventing reflected amplification attacks on resolvers is fairly straightforward, the operator of the servers have to configure the name server to only respond to IP addresses that should be allowed to query it. This practice is described in RFC 5358 [6].

Authoritative name servers, on the other hand, should by design respond to any valid query from any IP address. Several re-active and pro-active techniques exists to mitigate these attacks on these servers.

The simplest reactive method that name server operators can apply is IP filtering. The operator of an authoritative name server could be notified by the victim and then block (for a certain amount of time) block requests from the victim's IP address. Another way the firewall can be utilized is to prevent attacks with well-known patterns in the packets, provided these patterns are static and not very complex.

To prevent the spoofing of source IP addresses, network-operators can implement BCP 38 [10]. Unfortunately, BCP 38 is not widely implemented as it is of no direct benefit to the network operator implementing it, but for their peer network operators. Hence, there is no (economic) incentive to implement it. However, some national telecom regulators have made this filtering mandatory [11, Section 9].

DNS dampening [8] is a pro-active anti-amplification measure whereby the requesting server collects "penalty points" for every query it makes based on the query type and the size of the response. When the points are above a certain threshold, the name server stops responding to queries from the "dampened" requesting server. These points decrease slowly over time and the name server will respond to queries from the IP address. This effectively stops the name server from participating in the attack. However, it may be that the dampened traffic is legitimate: the victim could be a wrongly configured resolver or have an an empty cache.

Another pro-active technique is Response Rate Limiting (RRL) [29]. It limits the number of unique responses sent by the authoritative server. Roughly, it works by keeping track of of several pieces of information of the responses. With every subsequent request, the name server checks whether the response that would be sent exceeds the set limit of responses per second per set of information. If this is the case, it either responds only once in a number of queries (configurable) or it sends a truncated (TC-flag set) answer, forcing a legitimate resolver to retry the query over TCP. RRL is currently the most promising technique and is implemented in the most popular name server software like BIND [14], NSD [20] and Knot [17]. The effectiveness of RRL is debated, it stops unsophisticated attacks using reflection.

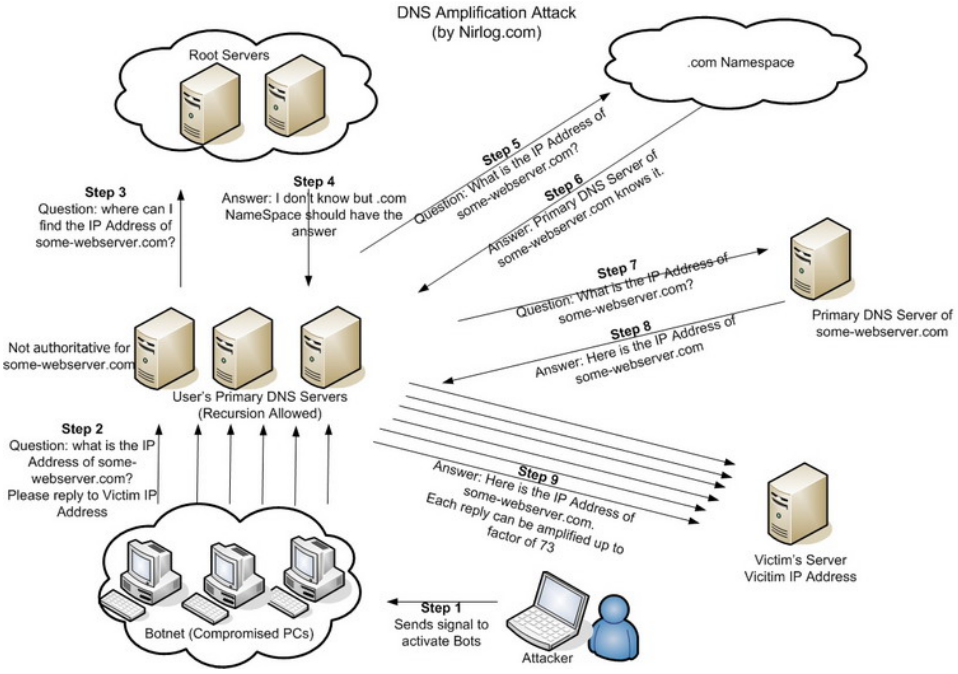


Figure 1: A graphical overview of a Reflected DNS Amplification attack (Source: nirlog.com). The attacker uses compromised machines that send DNS requests to resolvers, using the victim's IP address as the source address (steps 1 and 2). The resolvers do a normal lookup of the data requested (steps 3 through 9) and sends the responses to the victim (step 9). These responses will use bandwith in the victim's network, thereby denying legitimate network traffic for the victim to reach it's destination.

1.1 Research Question

One of the open problems with respect to to DNS amplification attacks is how to detect amplification attack patterns as they are initiating or in progress, as well as detection of other ways that DNS servers are abused.

The main question this research tries to answer is:

What types of behaviour can be detected in traffic to and from authoritative DNS servers and how can this detection be used to mitigate denial-of-service attacks?

1.2 Outline

Section 2 describes the approach taken during this research, related work and background information on the methods used. Section 3 describes the data that was used, its storage, manipulation and the processing of data for analysis. Section 4 contains the findings of this research. Describing how these were found, why these anomalies occur and what their significance is. Finally, section 5 contains several conclusions are drawn about the findings and future work is discussed.

2 Approach

In order to answer the research question, a visual approach was used to identify patterns in DNS messages.

2.1 Related Work

Rozekrans and de Koning measured the effectiveness of RRL [25] and found out that more complex attacks, e.g. attacks where the queries are spread out over multiple names and zones, are not detected nor stopped by RRL.

In their 2006 paper "Visualizing DNS Traffic" [23], Ren et al. describe several related interactive visualizations that were created to gain insight into logs created by resolvers to identify potential security incidents with the clients of those resolvers. They visualized the changes in client behaviour over short periods of time to give a quick overview to allow the operator to monitor the health of their clients. In one of their case studies, they detected a client that was actively participating in SSH bruteforce attacks by seeing the number of `PTR` by that client queries rise quickly.

Many nameservers come with the facilities to show statistical information from the requests they

receive. Either by having the name server aggregate this data or by having the name server log information on queries for later processing by external tools.

The visual approach used by Ren et al. works for caching resolvers. As authoritative name servers use the same protocols and data as resolvers (albeit in a slightly different fashion) it makes sense to apply visual techniques to this research.

2.2 Visual Analytics

Visual Analytics (VA) is a technique that uses the power of the human cognitive system to recognize patterns in images. This allows the researcher to quickly create and test hypotheses.

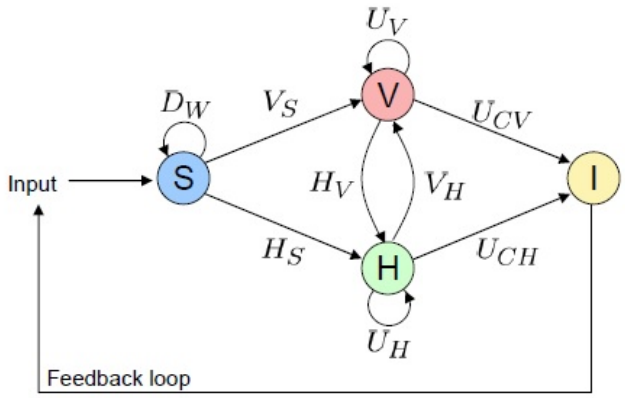


Figure 2: The Visual Analytics Process as described in Keim et al. [16, p. 5]

In 2008 Keim et al. defined a process to gain insight into large data sets using a visual approach [16]. This process revolves around an iterative process between the data source S, the visualizations V, the hypotheses H and the insight gained (I). The interactions between these can be seen in figure 2. In this model, hypotheses are created and confirmed or invalidated while insight (I) is gained by using visualizations (H_V and V_H), going back and forth between them.

3 System

To gain insight into the data a system consisting of several tools were developed. The first tool was developed to convert the acquired data into a format suitable for import into the data store. A second set of tools then create visualizations based on data retrieved from this data store.

3.1 Data Acquisition

The data for analysis required for this research consisted of both queries and responses from a authoritative name serve. This data was supplied by SURFnet[1] from their authoritative DNS server `ns1.surfnet.nl` that is authoritative for over 4000 zones, several of which are high-profile/high-volume zones such as `ac.be`, `ac.uk` and `gov.uk`.

The data used spans several days in the period from May 24th until June 13th 2013. The data set contains almost 630 million DNS messages. The average query rate on the name server is 900 queries per second.

The data was provided under a Non-Disclosure Agreement (NDA) as it may contain sensitive or private information in the form of queries or data sent inside the packets. Hence some images contain anonymized or blanked-out query names, IP addresses etc. and methods in which the several tools were used are described in general.

SURFnet actively blocks certain known attack patterns in the DNS traffic towards their authoritative name servers. The attacks that are blocked have very specific signatures, making it possible to filter them using host-based firewall rules on the authoritative name servers. These queries are filtered before they reach the name server software, consequently no responses to the queries are present in the data set.

3.1.1 Data Storage and Processing

The data was stored in an ElasticSearch [9] cluster. ElasticSearch is an Open Source search server built on top of Apache Lucene [2], an information retrieval library, that is scalable, distributed and offers high-availability in cluster setups. The cluster consisted of 20 data-nodes with the hardware specifications described in appendix A.1. These nodes were installed and configured in the same way as is described by the author and Fiebig et al. in "Scalable Large Data Cluster Services for Visual Analytics" [12]. This means that Debian GNU/Linux 6.0 ("Squeeze") [7] was used with ElasticSearch version 0.90.0.

The supplied data was in packet capture (PCAP) format and contained all DNS traffic to and from the

[1] SURFnet develops, implements and maintains the national research and education network (NREN) of the Netherlands

[2] http://lucene.apache.org/

name server. Only the DNS message sent over UDP were stored in the data store, as spoofing the source IP address with TCP is hard due to its connection-oriented design [21].

ElasticSearch stores data in a so-called "document" in JavaScript Object Notation (JSON) [4] format. ElasticSearch allows the data to have a typed schema, known as a "mapping" [2], where the fields are described. This typed schema allows for efficient storage in ElasticSearch and restricts the insertion of wrongly typed data, e.g. inserting a string in an integer field. The mapping used for this research can be found in appendix A.2. an overview of the fields in the data can be found in appendix B.1.

An anonymized example of a single packet in the cluster can be found in appendix B.2.

3.2 Data Retrieval and Visualization

To retrieve the data from the data store and to visualize this data, two types of tools were developed. First, tools that run batched and generate images that show an overview of the data. Secondly, several small, browser-based tools were created to allow the user to filter on the fields in the data and select the fields of interest to show in the visualization in several ways. The reason for this approach is twofold. As the data store contains a large amount of information, an overview is required before the user can zoom in. Secondly, this approach adheres to the "Visual Information-Seeking Mantra" [26] defined by Shneiderman:

> "Overview first, zoom and filter, then details-on-demand."

The overview, zoom and filter are implemented in proof-of-concept prototype tools. The details-on-demand is only partially implemented.

3.2.1 Source Ports and Query IDs

In the DNS, the source port of the query and the 16-bit query identification number should be chosen at random [13, section 9.2] The former (mostly) by the operating system and the latter by the DNS software.

When using a scatter plot to plot these two fields in a 2-dimensional space, one should expect to see only a fully uniform distribution of points in this visualization.

Any variation or pattern could mean several things. Simple Denial of Service tools that do not adhere to the DNS specifications might re-use query IDs or select them from a smaller range than specified to increment the rate of sending queries to the server used for amplification. A non-random selection of source ports could indicate a problem with the resolver, the operating system the resolver runs on or a tool that, for instance, sends the same packets many times to achieve amplification.

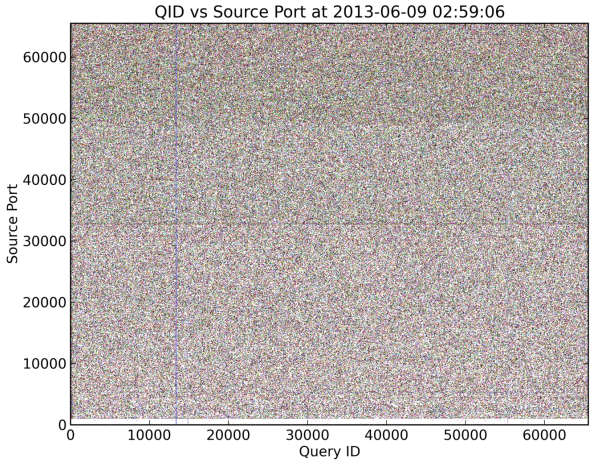


Figure 3: An example of a scatter plot showing the source port and query ID, a larger version can be found in Appendix C.2

The first batch tool generates such scatter plots. One plot is created for all the data in an 1800 second interval. As the average query-rate is 900 queries per second, the scatterplots will contain around 1.2 million points. This resulted in images as show in figure 3. The query ID and source ports mapped on the X and Y axes, respectively. Every point in the image is a single DNS query made to the server starting at the date in the title until 30 minutes later. To identify whether emerging patterns are from similar IP addresses, every point was colored based on the IPv4 /24 subnet. This allows the user to see if the pattern was caused by the same network or host. As IPv6 is not widely deployed, these data points were plotted as black dots. A second batch tool was created to draw a histogram of the distribution of these fields. This complements the previous visualization, allowing the user to see whether the patterns seen in the scatter plot show a non-uniform distribution for a single dimension in the scatter plot.

3.2.2 Average and Maximum Buffer Sizes in Queries

The EDNS standard [5] introduces a mechanism to increase the payload size of UDP DNS packets beyond the original 512 byte limit [19]. EDNS allow the client to specify the maximum response size it can handle. Common buffer sizes are 512, 4096 bytes [1].

Extremely high buffer sizes could mean an attacker is trying to flood a victim with huge packets of DNS data. A legitimate resolver could use high buffer sizes, but it has been discovered that this leads to fragmentation of DNS packets that could lead to slow response times from the resolver to the client as fragments are often lost in intermediate networks [24].

This second batch visualization tool creates a single graph that shows the mean and maximum buffer sizes for all requests in a five minute period over time, see figure 4.

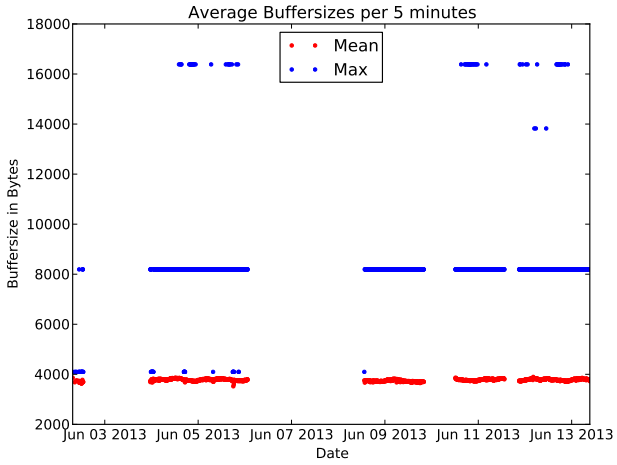


Figure 4: The mean and maximum advertised buffer sizes per 5 minutes, plotted over time, in the data set.

3.2.3 Aggregated Information

As the data store contains a lot of information, an aggregated view of parts of the data is required. One of the visualizations allows the user to show aggregrated information (counts of different fields) before and after filtering on values or ranges in other fields. This allows for quick examination of the data and drilling down to interesting entries. Figure 7 shows a graph created by this tool.

The page keeps the results of previous queries displayed so the user can see the results of previous queries to keep filtering and gaining more insight into the data.

3.2.4 Relationships Between Various Fields in DNS Queries

Parallel coordinates are an efficient way to show the relationships between multiple dimensions in the data [30]. As there are many fields in a DNS messages, the user should be able to see the relationships between these fields on a sub-set of the data.

This tool tries to be a Swiss-army knife to see relationships between many of the fields in the data. It allows the user to filter (AND-wise) on all the fields in the database. It has separate controls where the user can select which fields should be shown in the visualization. The application then shows a parallel coordinate image of the selected fields. The tool also allows the user to make subselections on the axes after the initial filters have been applied, as to gain more insight into subsets of the filtered data. The visualization allows the user to also drag and drop the axes in different orderings as to further explore the data. An example is show in figure 11 and Appendix C.1.

4 Results

Using the tools described in the previous sections, several things were discovered in the data. This section explains several of these and their significance. As stated before, the data contained privacy sensitive information. Because of this, some parts of the images have been blacked out.

4.1 Badly Configured Resolvers or Network Equipment

In all scatterplots that map the Query IDs to source ports (Appendix C.2), a line is seen around source port 32768 (2^{15}), see figure 5. As seen by the colors in the line, there is not a single host responsible for this behavior.

As randomness in selecting the source port is a necessary security measure against falsified answers [13], this behaviour could lead to security incidents, as this weakness can be exploited for cache poisoning [15]. While researching the cause of this line, an IP address was discovered that belonged to

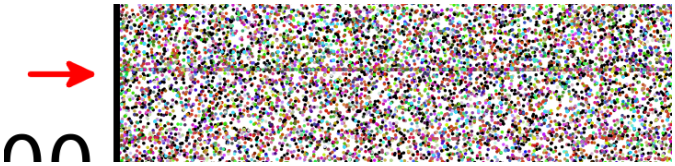


Figure 5: Detail of a non-uniform distribution of source ports as shown in the scatter plot of source ports versus Query IDs.

a customer of SURFnet. SURFnet operators then remarked that the DNS resolver was behind a NAT firewall that changes the source port to the first free port starting at 32768. Another, smaller, bias that can be seen in the image exists between the source ports 50,000 and 62,000. Both biases are visually confirmed by the histogram shown in Appendix C.3.

4.2 New Amplification Attack Patterns

In many of the scatter plots created, several lines were visible that showed several ranges of Query IDs were over-represented for several subnets/hosts. In figure 6, three vertical lines are clearly visible at Query IDs near 12000, 15000 and 55000. Note that the ones near 15000 and 55000 have the same color, indicating that these anomalies can be attributed to hosts inside the same /24 IPv4 subnet.

Figure 7 and 8 show the distribution of IP addresses responsible for queries with Query IDs in those ranges. These IPs are labeled 1, 2a and 2b. These figures come from the tool described in section 3.2.3. The date filter was the same interval as the one used in the scatter plot.

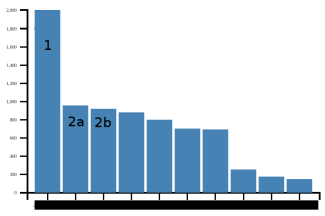


Figure 7: The graph of requests per IP in a 1800 second time frame where the Query ID is between 12000 and 16000

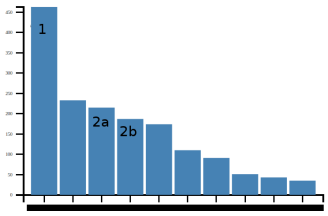


Figure 8: The graph of requests per IP in a 1800 second time frame where the Query ID is between 54500 and 55500

The same tool is used to only show the summary of the queries done by these IPs. There are an extreme amount of `PTR` queries and a fair number of ANY queries as well (figure 9).

When looking at the queries done by IP addresses 2a and 2b (not shown), they nearly only perform `PTR` queries indicating that the reverse zone of SURFnet is scanned. However, the same name is requisted very often and over 3.5 million requests are made within the 1800 second time period.

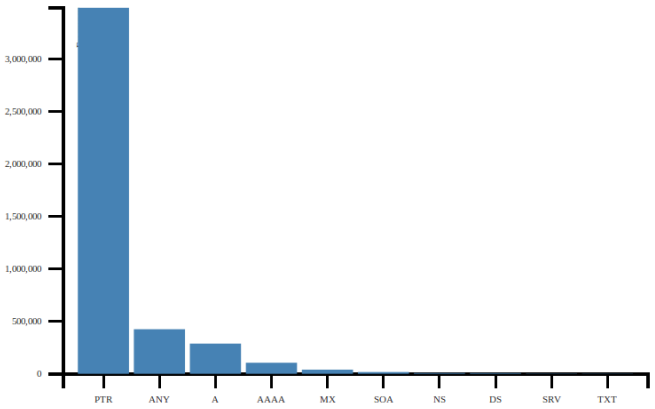


Figure 9: Distribution of query types for the 3 suspicious IP addresses

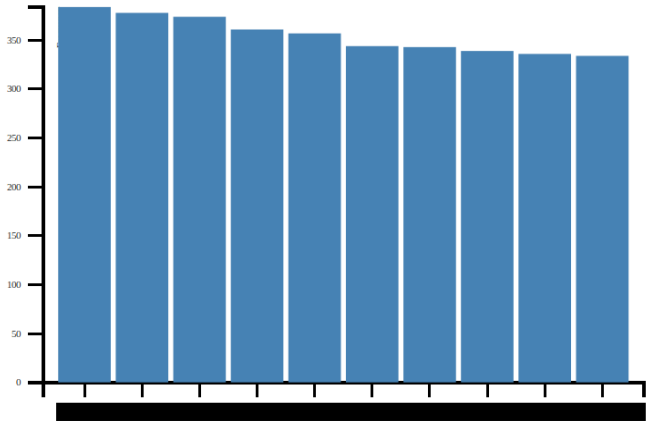


Figure 10: Distribution of query names for suspicious IP addresses 1

A reverse lookup of these two addresses reveals them to belong to a large provider of recursive DNS services.

When looking at the query names and types requested by the remaining IP address (1) only `ANY` queries are seen, which is usually a sign of abuse. The most interesting feature of the requests is the distribution of query names. These are quite uniform as shown in figure 10. This process and associated visualization shows one of the techniques that defeats RRL are used by the attackers.

A reverse lookup and a WHOIS of the address reveals that the address is registered to a hosting provider boasting to be "Providing DDoS Protected Secure Virtual/Dedicated Server". The machine hosted at IP address 1 is almost certainly under at-

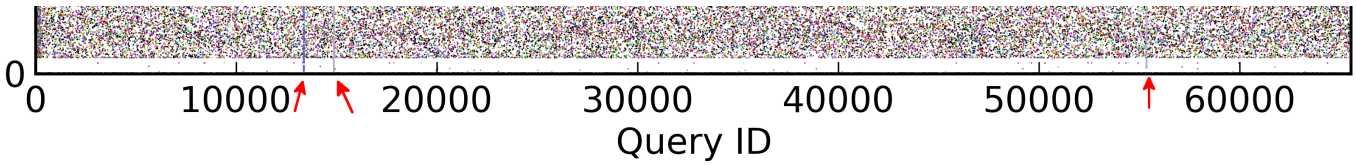


Figure 6: Detail of a non-uniform distribution of Query IDs as shown in the scatter plot of source ports versus Query IDs.

tack and the SURFnet name server is abused as a reflector.

These attacks are somewhat sophisticated as they spread the amplification over multiple names. However, EDNS is not used in the request (figure 11). This means that the maximum size of the response will be 512 bytes and that DNSSEC related records will not be included in the answers. As such, they do not utilize the full amplification potential that could be achieved.

5 Conclusions and Future Work

This research started out as an attempt to apply a visual approach to identify possible abuse of authoritative DNS servers for reflection and amplification attacks based on the full DNS messages sent to and from the server. Several tools were developed to explore the data using an iterative approach that allow the user to follow the steps of the Shneiderman mantra.

In the course of developing these tools, several patterns were discovered in the data that indicated that the name server was being abused to attack victims. These attacks went unnoticed prior to this research. By using a combination of the developed tools, it was possible to pinpoint the victim and the manner by which it was attacked. The distributed and diffuse nature of the discovered amplification attack that defeat the current RRL implementation are cause for further analyses of these attacks.

Another pattern that was discovered was a bias in the selection of UDP source ports either by legitimate resolvers themselves or by on-path network devices. These discoveries show that applying visual techniques to a large data set of DNS messages can lead to new insights into large amounts of DNS messages.

The second part of the research question was if the discovery of these patterns could mitigate the denial of service attacks. The visual detection will not directly lead to new mitigation techniques. However, an in-depth analysis on these patterns could be used to create new and more effective mitigation algorithms against reflected amplification attacks.

This paper only describes patterns found in the data set that was used. Undoubtly are there more patterns to be discovered using these techniques. The following sections lists expansions and improvements to the research done in this project.

5.1 Improvements to the Tools

The tools and script created in the course of this research were for proof of concept purposes only. But as they do provide new insights into DNS queries and answers, improvements could make them useful in a production environment. An operator could use these tools to gain a quick understanding of an attack after it has happened, or even during should the data be already available, and create countermeasures based on the insights gained.

5.1.1 Generalization

The tools were created for very specific use cases. Making these more general could increase the effectiveness of them. An example of such an improvement is the tool that creates the graphs of the mean and maximum buffersizes. There are several other fields in the messages, e.g. the UDP payload size, that could benefit from being visualized.

5.1.2 Interactivity and Details on Demand

The visualizations created by the tools, both the batch and GUI tools, don't allow for filtering using the generated visualization. This makes them "static". Adding interactivity like zooming in, getting details on single datapoints or filtering by field values in the existing visualization would be an improvement.

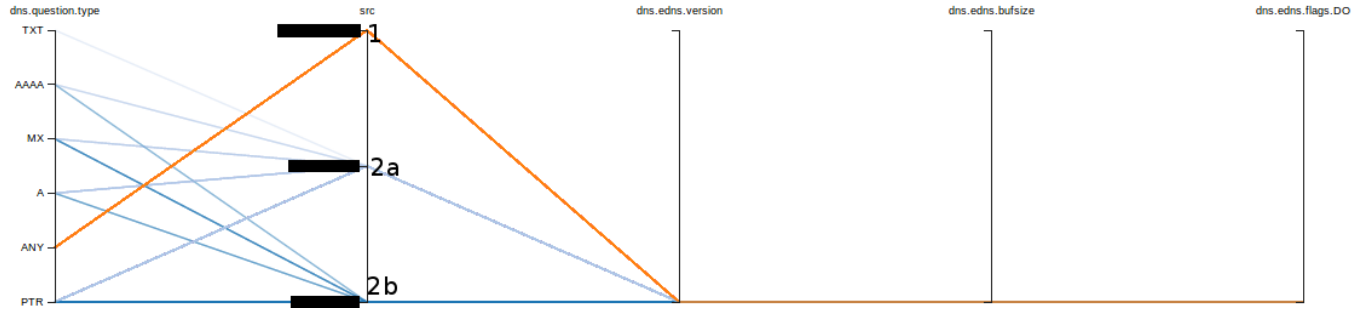


Figure 11: The relationship between the fields in the queries made by the 3 suspicious IP addresses. The fields shown are query type, IP address, EDNS version (top = 0, bottom = non-existent in query), EDNS Buffersize (bottom = non-existent in query) and DNSSEC OK flag (top = set, bottom = non-existent in query)

5.1.3 (Near-) Real Time Tooling

Most DNS operators are continuously monitoring their name servers. When an incident occurs, the analysis is done afterward with the data that was captured while the incident was already in progress. Keeping the last 24 hours of data in a data store as described in this paper would allow for an operator to identify the initial stages of the incident and create detection and prevention mechanisms.

5.2 Data Fields

While investigating the source of spoofed traffic, the TTL field (or hop count in IPv6) in the IP header can give an indication where the packets are coming from. It could help to filter and show these for further insight.

Other fields in the current data store have not received scrutiny. The different (combinations of) flags within DNS messages could be the subject of an investigation. The response code, their associated message sizes, answer types and destination addresses of responses in the data would make for an interesting combination of dimensions that could give insights into new RRL mechanisms.

5.3 Resolver Behaviour

The observed bias in source port selection should be of concern for DNS resolver operators. A follow-up research should be done to identify the extend of this problem and how vulnerable these servers are to cache poisoning compared to resolvers implementing RFC 5452.

References

[1] Daniel Karrenberg et al. *Measuring DNS Transfer Size – First Results*. Accessed: 10-Jun-2013. 2010. URL: https://labs.ripe.net/Members/dfk/content-measuring-dns-transfer-sizes-first-results.

[2] ElasticSearch Authors. *Guide > mapping*. Accessed: 15 Jun 2013. URL: http://www.elasticsearch.org/guide/reference/mapping/.

[3] Daniel J. Bernstein. *The DNS Security Mess*. 2012. URL: http://cr.yp.to/talks/2012.06.04/slides.pdf.

[4] D. Crockford. *The application/json Media Type for JavaScript Object Notation (JSON)*. RFC 4627 (Informational). Internet Engineering Task Force, July 2006. URL: http://www.ietf.org/rfc/rfc4627.txt.

[5] J. Damas, M. Graff, and P. Vixie. *Extension Mechanisms for DNS (EDNS(0))*. RFC 6891 (INTERNET STANDARD). Internet Engineering Task Force, Apr. 2013. URL: http://www.ietf.org/rfc/rfc6891.txt.

[6] J. Damas and F. Neves. *Preventing Use of Recursive Nameservers in Reflector Attacks*. RFC 5358 (Best Current Practice). Internet Engineering Task Force, Oct. 2008. URL: http://www.ietf.org/rfc/rfc5358.txt.

[7] *Debian Project Website*. URL: http://www.debian.org/.

[8] Lutz Donnerhacke. *DNS Dampening*. Accessed: 19 Jun 2013. 23 Sept 2012. URL: http://lutz.donnerhacke.de/eng/Blog/DNS-Dampening.

[9] *ElasticSearch Website*. Accessed: 10-Jun-2013. URL: http://www.elasticsearch.org/.

[10] P. Ferguson and D. Senie. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. RFC 2827 (Best Current Practice). Updated by RFC 3704. Internet Engineering Task Force, May 2000. URL: http://www.ietf.org/rfc/rfc2827.txt.

[11] Finnish Communications Regulatory Authority (FICORA). *Regulation on information security and functionality of internet access services*. URL: http://www.ficora.fi/attachments/englantiav/5B37hthyt/FICORA13A2008M.pdf.

[12] Tobias Fiebig et al. *Scalable Large Data Cluster Services for Visual Analytics*. 2013. URL: https://www.os3.nl/_media/2012-2013/students/pieter_lexis/fiebig_katz_lexis_yassem_-_va_clusters.pdf.

[13] A. Hubert and R. van Mook. *Measures for Making DNS More Resilient against Forged Answers*. RFC 5452 (Proposed Standard). Internet Engineering Task Force, Jan. 2009. URL: http://www.ietf.org/rfc/rfc5452.txt.

[14] *ISC BIND Website*. URL: http://www.isc.org/downloads/bind/.

[15] Dan Kaminsky. "Black ops 2008: It's the end of the cache as we know it". In: *Black Hat USA* (2008).

[16] Daniel A Keim et al. *Visual analytics: Scope and challenges*. Springer, 2008.

[17] *Knot DNS Website*. URL: https://www.knot-dns.cz/.

[18] P.V. Mockapetris. *Domain names - concepts and facilities*. RFC 1034 (INTERNET STANDARD). Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936. Internet Engineering Task Force, Nov. 1987. URL: http://www.ietf.org/rfc/rfc1034.txt.

[19] P.V. Mockapetris. *Domain names - implementation and specification*. RFC 1035 (INTERNET STANDARD). Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604. Internet Engineering Task Force, Nov. 1987. URL: http://www.ietf.org/rfc/rfc1035.txt.

[20] *NSD Website*. URL: http://www.nlnetlabs.nl/projects/nsd/.

[21] J. Postel. *Transmission Control Protocol*. RFC 793 (INTERNET STANDARD). Updated by RFCs 1122, 3168, 6093, 6528. Internet Engineering Task Force, Sept. 1981. URL: http://www.ietf.org/rfc/rfc793.txt.

[22] Matthew Prince. *The DDoS That Almost Broke the Internet*. Accessed: 11-Jun-2013. URL: http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet.

[23] Pin Ren, John Kristoff, and Bruce Gooch. "Visualizing DNS traffic". In: *Proceedings of the 3rd international workshop on Visualization for computer security*. ACM. 2006, pp. 23–30.

[24] Roland van Rijswijk Deij. *DNSSEC and Fragmentation – A Prickly Combination*. Given at ICANN 45 in Toronto, 17 Oct 2012. 2012. URL: http://toronto45.icann.org/meetings/toronto2012/presentation-dnssec-fragmentation-17oct12-en.pdf.

[25] T Rozekrans and J de Koning. *Defending against DNS reflection amplification attacks*. 2013. URL: http://www.nlnetlabs.nl/downloads/publications/report-rrl-dekoning-rozekrans.pdf.

[26] Ben Shneiderman. "The eyes have it: A task by data type taxonomy for information visualizations". In: *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE. 1996, pp. 336–343.

[27] Chris Thompson. *SNMP DDoS Vector - Secure Your Network NOW!* URL: http://www.spamhaus.org/news/article/678/.

[28] Randal Vaughn. *DNS Amplification Attacks*. 2006. URL: http://www.isotf.org/news/DNS-Amplification-Attacks.pdf.

[29] Paul Vixie and Vernon Schryver. *DNS Response Rate Limiting (DNS RRL)*. URL: http://ss.vix.su/~vixie/isc-tn-2012-1.txt.

[30] Edward J Wegman. "Hyperdimensional data analysis using parallel coordinates". In: *Journal of the American Statistical Association* 85.411 (1990), pp. 664–675.

A ElasticSearch

A.1 Hardware Specifications

Vendor	SuperMicro
Type	X7DBT
CPU	Intel Xeon E5420 @2.50GHz
Memory	16 GigaByte (8 * 2GB) DDR2, 667 MHz

Table 1: Cluster node hardware specifications

A.2 `dns-mapping.json`

```
{
  "mappings": {
    "data": {
      "properties": {
        "dns": {
          "properties": {
            "additional": {
              "properties": {
                "data": {
                  "index": "not_analyzed",
                  "type": "string"
                },
                "name": {
                  "index": "not_analyzed",
                  "type": "string"
                },
                "ttl": {
                  "type": "integer"
                },
                "type": {
                  "index": "not_analyzed",
                  "type": "string"
                }
              }
            },
            "answer": {
              "properties": {
                "data": {
                  "index": "not_analyzed",
                  "type": "string"
                },
                "name": {
                  "index": "not_analyzed",
                  "type": "string"
                },
                "ttl": {
                  "type": "integer"
                },
                "type": {
                  "index": "not_analyzed",
                  "type": "string"
                }
              }
            },
            "authority": {
              "properties": {
                "data": {
                  "index": "not_analyzed",
```

```
                "type": "string"
              },
              "name": {
                "index": "not_analyzed",
                "type": "string"
              },
              "ttl": {
                "type": "integer"
              },
              "type": {
                "index": "not_analyzed",
                "type": "string"
              }
            }
          },
          "edns": {
            "properties": {
              "bufsize": {
                "type": "integer"
              },
              "flags": {
                "properties": {
                  "DO": {
                    "null_value": "false",
                    "type": "boolean"
                  }
                }
              },
              "version": {
                "type": "integer"
              }
            }
          },
          "flags": {
            "properties": {
              "AA": {
                "null_value": "false",
                "type": "boolean"
              },
              "AD": {
                "null_value": "false",
                "type": "boolean"
              },
              "CD": {
                "null_value": "false",
                "type": "boolean"
              },
              "RA": {
                "null_value": "false",
                "type": "boolean"
              },
              "RD": {
                "null_value": "false",
                "type": "boolean"
              },
              "TC": {
                "null_value": "false",
                "type": "boolean"
              }
            }
          },
          "opcode": {
            "type": "string"
          },
          "qid": {
            "type": "integer"
```

```
            },
            "question": {
              "properties": {
                "data": {
                  "index": "not_analyzed",
                  "type": "string"
                },
                "name": {
                  "index": "not_analyzed",
                  "type": "string"
                },
                "type": {
                  "index": "not_analyzed",
                  "type": "string"
                }
              }
            },
            "rcode": {
              "type": "string"
            }
          }
        },
        "dport": {
          "type": "integer"
        },
        "dst": {
          "index": "not_analyzed",
          "type": "string"
        },
        "sport": {
          "type": "integer"
        },
        "src": {
          "index": "not_analyzed",
          "type": "string"
        },
        "timestamp_unix": {
          "type": "double"
        },
        "udp_len": {
          "type": "integer"
        }
      }
    }
  }
}
```

B DNS Data

B.1 Table of DNS fields in the cluster

Fieldname	Type	Description
src	IP address	The source IPv4 or IPv6 address
dst	IP address	The destination IPv4 or IPv6 address
sport	Integer	The source port of the packet
dport	Integer	The destination port of the packet
timestamp_unix	Floating Point	The timestamp when the packet was captured
udp_len	Integer	The number of bytes in the DNS packet
dns.opcode	String	Textual value of the OPCODE field
dns.rcode	String	Textual value of the RCODE field
dns.qid	Integer	Value of the Query Identification field
dns.flags.AA	Boolean	Set to true if the AA flag is set
dns.flags.AD	Boolean	Set to true if the AD flag is set
dns.flags.CD	Boolean	Set to true if the CD flag is set
dns.flags.RA	Boolean	Set to true if the RA flag is set
dns.flags.RD	Boolean	Set to true if the RD flag is set
dns.flags.TC	Boolean	Set to true if the TC flag is set
dns.edns.bufsize	Integer	The advertised EDNS(0) buffersize in bytes
dns.edns.version	Boolean	The EDNS version
dns.edns.flags.DO	Boolean	Set to true if DO flag is set
dns.query.name	String	The name of the DNS record in the query section
dns.query.type	String	The textual representation of the type of this record in the query section
dns.answer.name	String	The name of the DNS record in the answer section
dns.answer.data	String	The date for this DNS record in the answer section
dns.answer.ttl	Integer	The Time-To-Live for this DNS record in the answer section
dns.answer.type	String	The textual representation of the type of this record in the answer section
dns.authority.name	String	The name of the DNS record in the authority section
dns.authority.data	String	The date for this DNS record in the authority section
dns.authority.ttl	Integer	The Time-To-Live for this DNS record in the authority section
dns.authority.type	String	The textual representation of the type of this record in the authority section
dns.additional.name	String	The name of the DNS record in the additional section
dns.additional.data	String	The date for this DNS record in the additional section
dns.additional.ttl	Integer	The Time-To-Live for this DNS record in the additional section
dns.additional.type	String	The textual representation of the type of this record in the additional section

Table 2: The data inserted into the cluster. For more information on the fields inside the DNS packets, see [18, 19, 5]

B.2 Record example

```
{
 "dns": {
  "additional": [],
  "answer": [
```

```
    {
    "data": "2002:440:0:1::45",
    "name": "ns1.examplezone.net.",
    "ttl": 7200,
    "type": "AAAA"
    }
   ],
   "authority": [],
   "edns": {
    "bufsize": 4096,
    "flags": {
     "DO": true
    },
    "version": 0
   },
   "flags": {
    "AA": true,
    "CD": true,
    "QR": true
   },
   "opcode": "QUERY",
   "qid": 1872,
   "question": [
    {
    "name": "ns1.examplezone.net.",
    "type": "AAAA"
    }
   ],
   "rcode": "NOERROR"
  },
  "dport": 42208,
  "dst": "1.2.3.4",
  "sport": 53,
  "src": "5.6.7.8",
  "timestamp_unix": 1370304171.54113,
  "udp_len": 84
}
```


Listing 1: An example of a DNS packet as inserted into the cluster

C Images

C.1 Parallel Coordinates Visualization

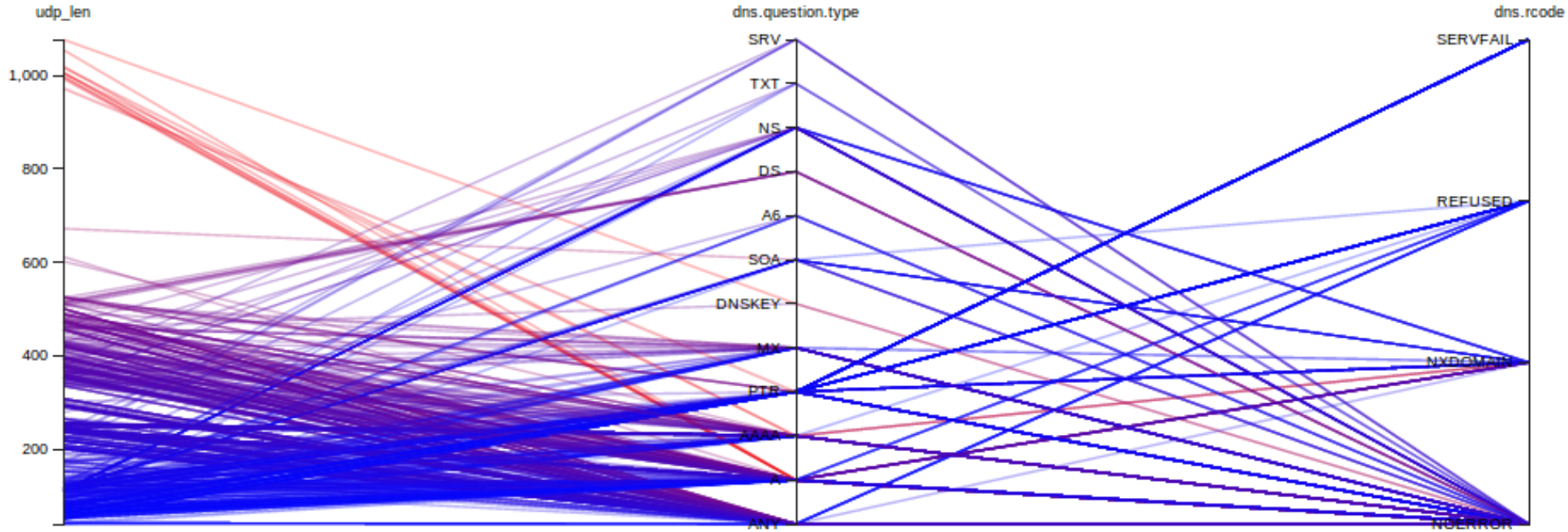
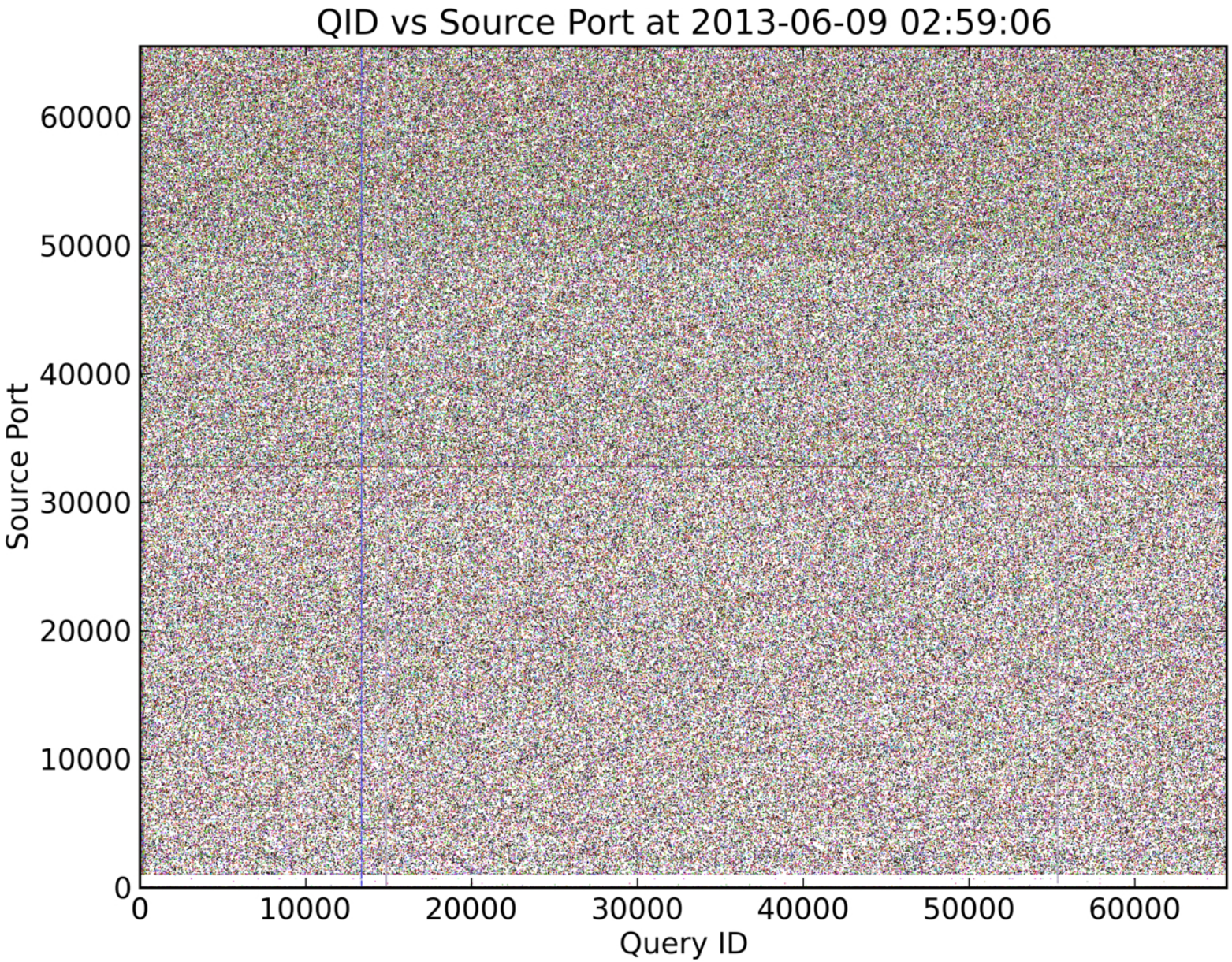


Figure 12: The relations application showing the relationship in parallel coordinates of the UDP packet sizes, Question Type and the response code of the server.

C.2 Scatter plot of the Source Port vs. the Query ID



C.3 Histogram of Source Port Selection in the Full Data Set