Visual Analytics for Root DNS Data

Eric Krokos, Alexander Rowden, Kirsten Whitley, and Amitabh Varshney, *Fellow, IEEE*

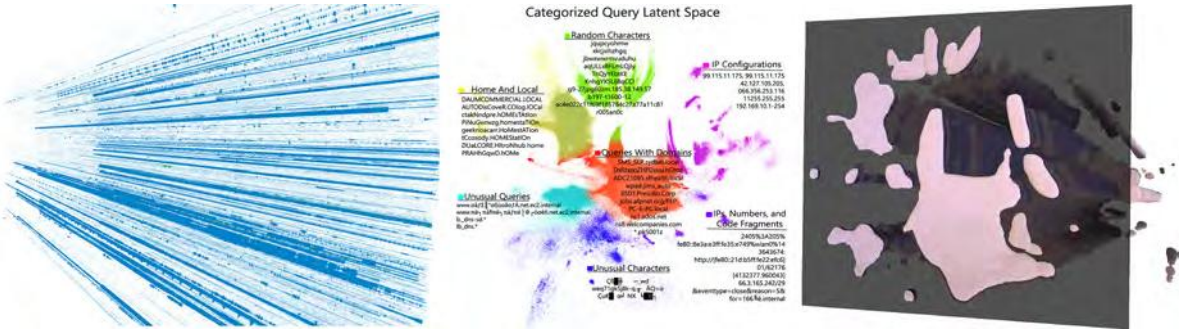

Fig. 1: Our root-DNS dual-visualization that provides both high and low-level overviews and interactions of the IP and query spaces. IP packet traffic is visualized on the left, revealing hidden patterns, IP distributions, and a real TCP-SYN flood. A two-dimensional query-space generated using deep learning portrays a spatial distribution of received queries and counts. The right image portrays the spatial distribution of queries as they change over time, revealing the diminished number of received queries due to a DDOS.

Abstract— The analysis of vast amounts of network data for monitoring and safeguarding a core pillar of the internet, the root DNS, is an enormous challenge. Understanding the distribution of the queries received by the root DNS, and how those queries change over time, in an intuitive manner is sought. Traditional query analysis is performed packet by packet, lacking global, temporal, and visual coherence, obscuring latent trends and clusters. Our approach leverages the pattern recognition and computational power of deep learning with 2D and 3D rendering techniques for quick and easy interpretation and interaction with vast amount of root DNS network traffic. Working with real-world DNS experts, our visualization reveals several surprising latent clusters of queries, potentially malicious and benign, discovers previously unknown characteristics of a real-world root DNS DDOS attack, and uncovers unforeseen changes in the distribution of queries received over time. These discoveries will provide DNS analysts with a deeper understanding of the nature of the DNS traffic under their charge, which will help them safeguard the root DNS against future attack.

Index Terms—Visualization, cyber security, deep learning, big data, dns, 3d, graphics

1 Introduction

There are two schools of thought on how to deal with cyber-attacks, automatic detection methods, and human-driven investigation. Automatic detection methods work by modeling normal and abnormal behavior, through prior knowledge of the behavior of malware and since by definition a cyber-attack is abnormal. However, humans and the machines are also capable of abnormal behavior, causing these automatic detection algorithms to throw many false alarms. In addition, these models take months to create, are created using attack data that occurred on average at least six months prior, and generally only detect the presence of old cyber-attacks, and are therefore unprepared for ever changing and newer attacks. In addition, these methods generally only report the presence of an attack, but give no details on specifics. In contrast, and in part in response to automated mechanisms, many cyber-security analysts prefer manual investigation and analysis of attacks. There is an overall mistrust of automated systems by those who perform cybersecurity analysis [35]. Generally, analysts tools consist of listing packets and related information, whereby data is inspected line by line. In contrast, many tools often only provide very high-level abstractions of the data, typically in the form of histograms, where the histogram consists of the number of received packets.Few tools fill the gap between very high and low-level analysis, as well as provide distinct informative views of the underlying data.

One major target of cyber-attacks is the Domain Name System (DNS) infrastructure, responsible for converting human-understandable URL queries into machine-understandable IP-addresses.The ubiquity and central importance of DNS make it a tempting target for attack and exploitation. If these domain name systems go down, it would create unprecedented chaos and instability on the internet as IP addresses change, caches expire, and queries remain unresolved. Finding, characterizing, understanding, and the mitigation of these attacks on DNS is of utmost importance.

One core aspect of maintaining and defending the DNS is providing DNS analysts with a method of monitoring the queries received. Those DNS analysts that are able to easily comprehend the variety and scope of the queries that pass through their system will be better able to characterize attacks, anomalies, and normal behavior. The challenge is the sheer amount of root DNS traffic which ranges from 100 to 300 GB per server, per DNS letter, per day. The primary systems in use today typically focus on packet counts and origination (source IPs). Modern packet analyzers generally present every aspect of every packet in tabular lists, with query information buried deep in expandable subsections in those lists, or as a single column among many. These techniques, while providing unparalleled detail, do not leverage our innate ability to process spatially organized data to find patterns and anomalies. In addition, they have little emphasis on the aspect of DNS that makes it so important, the queries themselves. By presenting DNS queries and IP-activity in a spatially and temporally coherent manner, with cross-

- *Eric Krokos is with University of Maryland College Park. E-mail: EKrokos@umiacs.umd.edu.*
- *Alexander Rowden is with University of Maryland College Park. E-mail: Alrowden@terpmail.umd.edu.*
- *Kirsten Whitley is with the US Department of Defense. E-mail: Visual@tycho.ncsc.mil.*
- *Amitabh Varshney is with the University of Maryland College Park. Varshney@umiacs.umd.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

visualization interactivity, enabling high and low-level investigation, DNS analysts will be able to more effectively process and organize that data. In this paper, we present our visualization which has been designed to take the vast amounts of root DNS queries, organize them in a spatially comprehensible manner, and facilitate easy investigation to not only answer existing questions, but to help DNS analysts discover new questions. We validate our presented approach on data from one of these 13 root DNS providers, namely the D-Root.

In summary, this paper makes the following contributions to immersive analytics and visualization for network security:

- We have designed a dual-interactive-visualization system for DNS query and IP data which leverages 3D graphics techniques to convey that data in a novel representation.
- We visualize an order of magnitude more DNS data than previous systems, providing analysts with high-level situational awareness, while preserving low-level details and nuance, without the need for switching between multiple different applications.
- We organize abstract DNS queries in an easy to interpret spatial layout using a deep learning variational autoencoder, such that co-located queries are semantically similar. We also leverage volume rendering to provide analysts with a high-level spatiotemporal understanding of how the distribution of DNS queries change.
- We identify and characterize distinct DNS anomalies and attacks through an informal empirical evaluation and discussion of discovered trends and clusters with industry experts.

In the following section, we present a review of the challenges, standard practices, as well as present and characterize new techniques.

2 Background

Originating in the days of ARPANET, the DNS can be considered as a simple list of host names with their mappings of to and from addresses, maintained in a frequently-updated host table. However, the open nature of DNS makes it vulnerable to a wide variety of attacks and abuses.

The constant attack by malicious sources has necessitated the need for automated intrusion detection systems (IDS).However, many industry operators have observed that modern IDS, although useful, are not optimal or trustworthy [8], in-part due to the large presence of false positives and inability to detect the latest threats [35].Often these systems require a human-in-the-loop to review these detection alerts, and to contextualize the alerts with additional information [37], often manually with separate visualization tools from the IDS [12]. Therefore, visualizations that can provide summary and precise representations of the data is of utmost importance [14]. In the remainder of this section, we review techniques with visualization as a core-component of the analytic and investigation process.

2.1 Traditional 2D Network and DNS Visualization

Traditional techniques for visualizing network data include charts (histograms), line-plots (including parallel coordinate views), graphs (including node-link diagrams), among others. The challenge is the enormous and always increasing amount of data to portray. Visualizing all aspects of the data at once is untenable. Tools such as Excel, NetStat [39] and Wireshark [29] (Figure 3), outline every aspect of every packet. This gives analysts an unprecedented level of detail, but hinders finding trends, correlations, and anomalies over time [10]. Traditionally, an analyst will write queries to explore their data, leveraging their background knowledge of the dataset. This process is extremely tedious and labor intensive, and generally requires a known starting point [9]. Generally, DNS analysts are interested in monitoring the health of their system, the flows of traffic, patterns, and anomalies.

Histograms are very commonly used for quick analysis of overall trends, such as direct comparisons between adjacent periods of time [2], the count of a particular feature, such as the number and type of alerts [44], and portray counts of packets [2], query types [43], and severity [42]. Histograms are arranged in 1D, by stacking elements to simultaneously show different properties [2], or with curved and circular representations [44], and in 3D [27]where the direction and orientation of the histogram along the z-axis provides additional information.

Similar in function to histograms, line-plots can convey counts over time [42]. One common implementation is parallel-coordinates, used to find botnets [19]and anomalies [27]in DNS traffic by plotting packet attributes along each axis such as IP-address, time, and attribute counts. Circular representations, such as those used for network intrusion detection [22], can reveal patterns providing what happened, where, and when. Theme rivers, akin to stacked histograms, have been used to visualize changes and anomalies in DNS query traffic [33]. One problem with parallel-coordinate visualizations, including traditional line-plots, are intentional and unintentional obfuscation (Windshield Attacks) [28]. Similarly, as the number of axes and data-points grow, the data elements can self-occlude and hide lingering patterns.

Network graphs, representing IPs, AS, domains, machines, queries, or users, connected via edges (shared traffic, association, or other connections) [35] have been used to visualize communities of hosts in DNS traffic [16], changes in DNS routing and look-up behaviors [20], and anomalous behavior in failed DNS queries [18].While network graphs are useful, previous research [11] found that their effectiveness decreases dramatically if the graph exceeds approximately twenty vertices, limiting their effectiveness for fine-level network analysis.

Many new visualizations leverage TreeMaps [31], which color-code packet counts and anomalies in IP-address bins. Other visualizations correlate geospatial aspects of DNS traffic and overlay packet counts [24, 34]. More creative visualizations, such as glTail (http://www.fudgie.org/) and Logstalgia [5] use interactive graphics to render a log file as dynamic 2D simulations.

The previously mentioned visualizations generally trade scalability for fine-level detail, and focus either on high-level summary overviews for large amounts of data, or detailed views for small amounts of data. Therefore many analysts use multiple tools to gain a complete picture, but this creates an unnecessary context switch and overhead. Additionally, many approaches layout their information with a focus on aesthetic qualities such as maintaining symmetry with uniform glyph positions, potentially compromising latent global and local data structures [40]. In our visualization, we preserve and show both precise and high-level representations for vast amounts of DNS data. Previously, the focus has been on the evolution of source IP packet counts over time, with little to no emphasis on the messages in the packets. The DNS system exists to handle queries, so enabling analysts to explore the changing distributions of queries is of critical importance. Such a visualization would be infeasible using traditional visualizations due to the arbitrary and high dimensionality of the queries, in addition to the irregular behavior of their transmission. Our work portrays a spatiotemporal distribution of packets and queries over time, revealing patterns and anomalies difficult to identify through earlier means.

2.2 3D Network Visualization

While 2D visualizations are regarded as easier to create and understand (in terms of time required for comprehension), recent research has shown there are many benefits to 3D visualizations over 2D for abstract data visualization [13, 30, 36], including clearer spatial separation, reduced over-plotting, and enabling faster construction and deeper mental models. In addition, many visualizations rely heavily on spatialization, encoding information in the location of data-elements. The addition of a third dimension makes available more insightful relative positioning [32].

One of the earliest uses of 3D for cyber-security visualization was by Stephen Lau [21]. The visualization uses a 3D scatter-plot to reveal patterns associated with vulnerability scan attacks. To minimize clutter from 2D parallel coordinate visualizations, many are expanding into the third dimension [41]. P3D, a 3D parallel coordinate network security visualization [28] creates multiple 3D planes, each with a set of either IPs, packet counts, ports, or other information along the x and y-axes, with lines connecting these planes representing connections or FTP transfers, to detect port scans while preventing the occlusion attacks such

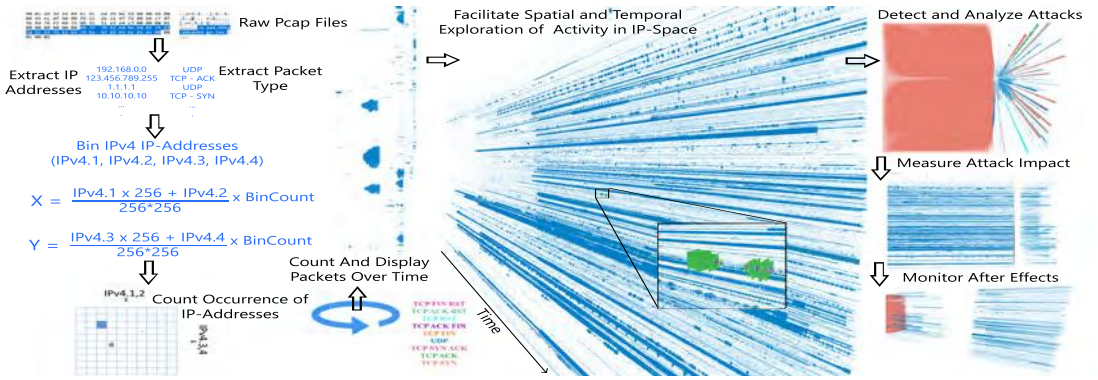

Fig. 2: Overview of the process from raw pcap files to The Flow-Map IP-Space visualization. Starting from a binary pcap file, we extract and count the occurrence of each IPv4 IP-Address and type of packet. Next, the IPs are converted from a 4D to a 2D grid representation, with glyphs scaled and colored based on the number and type of packets. This process repeats for each time slice, with slices stacked along the z-axis. The result is then visualized using 3D accelerated rendering, which allows for high-level structure and low-level analysis, to help analysts establish a sense of normalcy (central blue image), identify outliers (green TCP burst), classify and characterize attacks (top right), measure attack impacts (middle right), and monitor after effects (lower right).

as Port Source Confusion and Windshield Wiper attacks [7]. Another example is Daedalus-Viz [17], which consists of several circular rings, corresponding to various monitored organizations, in orbit around a central sphere representing the complete IPv4 space, with connecting lines indicating the transfer of packets.

One main drawback of the previous systems is the relatively small amount of data they can visualize. We expand upon these ideas by combining elements of scatterplots and parallel coordinate visualizations. Rather than just plotting one element per cell, we interleave multiple data points within a given spatiotemporal cell using 3D transparency, enabling more information to be presented, as well as a direct comparison between similar elements. Lastly, plotting many discrete points temporally increases the overall visual complexity. Instead, we have clumped together spatially coherent groups of points into mesh surfaces to minimize the visual clutter, to reveal structural patterns and changes within the original query point cloud.

3 Problem and Solution

3.1 The Challenge

As part of our development process, we interviewed DNS analyst experts from the University of Maryland D-Root. One of the challenges they face is the scope and enormity of the data they manage. Over the course of an average day at just one of their 131 global facilities, they process over 100 GB of traffic, with a peak traffic around 300 GB. When under attack by a typical DDOS, one server can process roughly 600 GB in one day.

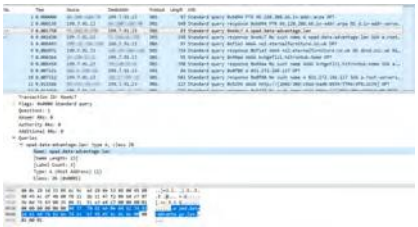

Fig. 3: An example analysis in Wireshark, a widely used pcap analyzer.

Traditional query visualization tools are very limiting and typically omit the queries and contents of the packets as part of the investigation, emphasizing packet counts and the distribution in the IP-space. As stated earlier, most `pcap` (packet capture) tools present packets line-by-line. An example of queries presented in a commonly used tool (Wireshark [29]) is presented in Figure 3. While this level of detail can be very useful, it limits the ability of an analyst to generalize and discover trends due to the lack of a global, visual, and temporal coherence as well as eliciting a sense of information overload.

There have been many root DNS attacks, historically lasting three [25] and five hours [15]. DDOS attacks outside the realm of the root-DNS on average last less than twenty hours [23].To ensure we can cope with the largest of attacks, we visualize 24 hours of traffic in our case study. However, our visualization is capable of showing larger durations of time.

For the purposes of this paper, we explore one recent root DNS attack. On June 25th 2016, a moderately sized DDOS attacked all root DNS authorities in a coordinated attack. A report published by the root DNS authorities on this specific attack can be found here (http://root-servers.org/news/events-of-20160625.txt).According to the official report, all DNS root name servers received a high rate of TCP SYN packets in a SYN flood attack for nearly four hours. The source addresses appeared to be randomized and uniformly distributed throughout the IPv4 address space. The observed traffic volume due was up to approximately 10 million packets per second (approximately 17 GB/s), per DNS root name server letter. Our goal is to provide analysts with a sense of normalcy over the course of a day, contextualize attacks when and if they occur, to aid in subsequent investigation and mitigation strategies, and help develop a characterization of current and future attacks for comparison.

Previous network visualizations have visualized up to approximately 350 million packets [3, 4, 20].In our presented visualization, we visualize, in real-time using 3D accelerated rendering techniques, over 2.4 billion packets, consisting of over 487 million unique queries, spanning 24 hours, from the McLean Virginia D-Root DNS site.

3.2 Approach Overview

There were three design considerations driving our development, to display an entire day of query traffic from a Root DNS server, show high-level structures and patterns in an intuitive manner with interactions enabling finer investigation, and display time as a spatial dimension. In this paper, we present two complementary visualizations of the activity in both the IP and query domains. An overview of the IP-space and query-space construction processes from raw packets to visualization are presented in Figure 2 and Figure 4.

3.3 Flow-Map IP-Space Visualization

The IP-space visualization uses what we call a Flow-Map, spatially presenting information regarding packet counts and types over time. The IP-space consists of 4-octets, resulting in over 4 billion unique values/indexes. To properly visualize such a space would require a four-dimensional cube, or a very tightly indexed 1D histogram. The compromise reached with our DNS experts, to maintain a fine-level of

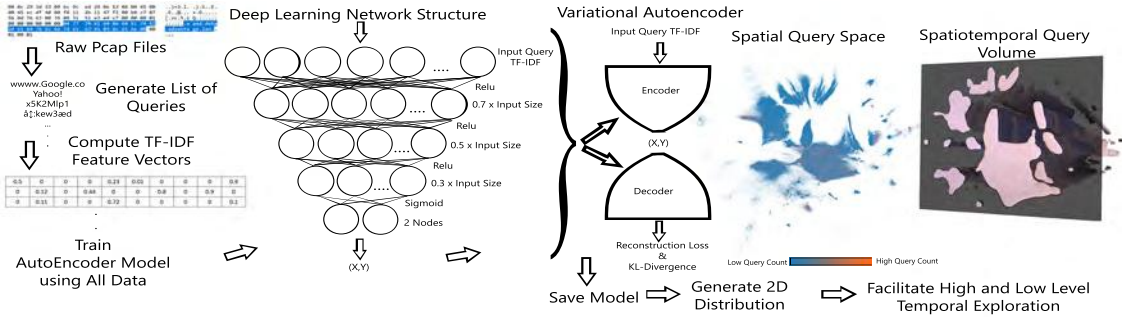

Fig. 4: A overview of the process from raw pcap files to Query-Space visualization. Starting from a binary pcap file, we extract and count each query. Next, each query is converted to a TF-IDF (Term Frequency, Inverse Document Frequency) character-level feature vector. A deep learning autoencoder is trained using all queries, to generate a visually coherent spatial distribution of queries when projected into a 2D space. This distribution of queries is then visualized using 3D accelerated rendering, which allows for high-level temporal structure (top-right) and low-level query analysis (bottom right).

detail without overwhelming the user, as summarized in Figure 2, is to reduce the IP space from over 4 billion to $10,000$ values, by combining the first and second octet pairs into a single value. These values are used to bucket (bin) the IP-addresses of the packets into a 2D IP-space. Our current scheme does not take into account autonomous domains. Therefore, some IP addresses that belong to very different autonomous domains will be sent to the same bucket while IP addresses that belong to the same corporation could be sent to different buckets. It should be possible in future, to add an Autonomous Domain layer above it, showing only those IP-bins belonging to a particular autonomous domain, or by reorganizing the IP-space into an autonomous domain space. Due to irregular packet arrival times, packets are temporally binned into 5 second chunks. Our discussions with DNS experts revealed that knowing the precise IP and arrival time of particular packet was insignificant and that gaining a general understanding of the distribution of packet sources is preferable. A novel characteristic of our visualization is that each bin/cell contains multiple glyphs, seamlessly linked together, sized based on the number of received packets (within a given time it represents), and colored transparently based on the type of packet received (UDP, TCP SYN, etc.). The transparency and sizes of the glyphs can be adjusted to aid in minimizing occlusion for certain views, revealing hidden information, and to emphasize different bins with certain counts of packets. The advantage of this Flow-Map representation is that time is represented as a spatial component, removing temporal animations or scrolling through individual time slices, thereby providing a globally and temporally coherent model for the analyst. Within this constructed visualization, analysts may freely move and rotate their view to get a high-level overview of the space, zoom in close for an in-depth analysis, and change the current transparency and glyph scaling levels using the keyboard and mouse. Hovering the cursor over any given element presents additional information, among which are the range of the IP bin, the number of packets, and the time.

3.4 IP-Space Observations

Three general observations can be made using the Flow-Map representation as shown in Figure 2. First, most of the space is empty, suggesting that most queries fall into relatively few IP-bins. Second, for most filled bins, the glyphs are relatively small, suggesting that most of the queries received are singletons (customers send just one or a few packets). Third, there are a few persistent high-packet count buckets that send out thousands of queries in just a few seconds. From this IP-space representation, an analyst can grasp the nature of the changing volume of traffic.

Within the first few hours of data, we found a small selection of interesting patterns. First is the anomaly in highlighted in the middle of Figure 2 which shows an instance of high TCP-based packet activity, as indicated by the green color. This was a large burst of packets, as indicated by the large size of the glyphs, and was distinct in that no TCP activity preceded or followed this period lasting roughly a minute. In the left Figure 5, we have extracted three IP bins that have regularly repeating, self-similar, internal patterns of traffic. This subset of data can be seen near the top-right of the blue Flow-Map in Figure 2. From our discussion with D-Root experts, this traffic might be from external monitoring sources, who periodically query the root DNS, resulting in this regular pattern. After a closer look at the queries from these IP-bins in the query-visualization, we found that the received queries were generally of the form `*.trafficmanager.net`. The top-right of Figure 2 shows the distribution of IPs used in the TCP-SYN flood attack (half of the first IP Octet, and the entirety of the other octets, contrary to the official report indicating the totality of the IPv4 space was spoofed) , the decreased volume of traffic from lost customers, and the resulting hard-drive failure as a result. The right of Figure 5 shows a high inter-bin temporal similarity found across all IP-bins involved in the TCP-Syn flood. It is possible that all these characteristics could serve as an attack signature. Thanks to the preservation of the low-level of detail, which would otherwise be summarized or abstracted away by other tools, these interesting patterns and anomalies were identified, driving further investigation.

3.5 Deep Learning Driven Query Space Visualization

Our query-space visualization provides analysts with a deeper understanding of the distribution of received queries. We organize the non-spatial queries into a spatial representation, enabling easy detection of patterns and structure from the large amount of data using deep learning. Each query is visualized as a 3D sphere, positioned near similar queries, and colored to indicate the number of times it was received. Using this visualization, an analyst can see the high-level distribution and volume of queries, then drill down to discover the precise queries received and draw observations, panning and zooming the camera with the keyboard or mouse, and obtain details for a specific data-element by hovering over it with the cursor. The goal is to provide high-level information, in the form of natural and easy to interpret geometric categorical structures as generated by deep learning and to facilitate low-level investigation by providing an analyst to get close and personal with the raw query

Fig. 5: Interesting self-similar patterns of intra-IP-bin queries and across-IP-bin traffic (from the TCP-Syn DDOS) over time in D-Root traffic.

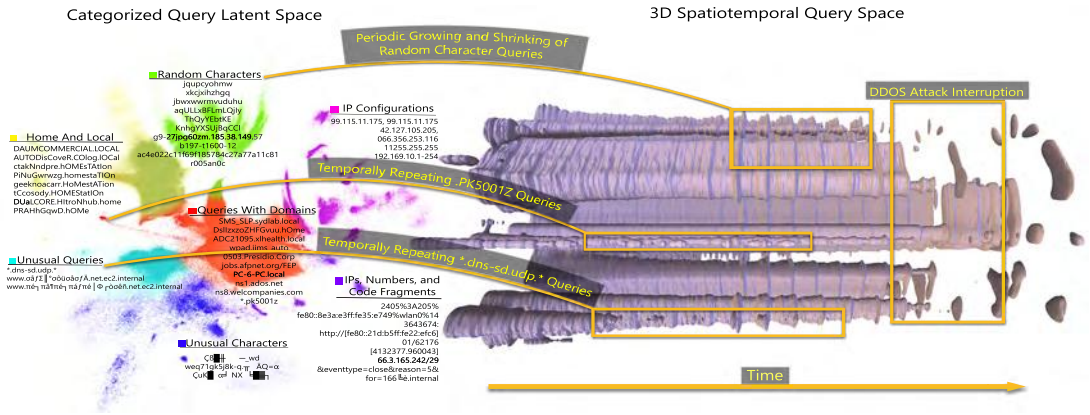

Fig. 6: In the lower-left, the 2D query-space clustered into semantic query categories. There is a general trend with alphabetic-based queries towards the left, and numeric-based queries the right, along with a general trend to normal characters at the top, and unusual characters at the bottom. The lower-right shows the temporal query visualization, portraying a high-level temporal overview of the distribution of queries. The top reveals a selection of interesting observations, such as, rapidly diminishing groups of queries at the start, temporally repeating groups of queries, the large reduction in queries during the attack, and an overall decrease in the number of queries over time.

data.

To generate the distribution of queries, we use deep learning to find patterns within the set of queries, and then project those queries into a 2D plane. Previous techniques, such as spectral clustering and PCA were unable to generate adequate representations, less cope with the huge quantities to be projected, due to the requirement of a full similarity matrix between all pairs of elements. In addition, the lack of a ground truth or implicit notion of similarity inhibits the use of other traditional methods. Therefore, we use deep learning, which is able to handle the data in chunks, maintain a global conception of the data-space, and learn its own metric of similarity from the data rather than have one imposed. Each query string is converted to a vector of TF-IDF (Term Frequency, Inverse Document Frequency) values, resulting in a 256-length character feature vector. TF-IDF is commonly used in natural language processing algorithms for converting text into a machine-understandable format by converting each word/letter into a value based on its importance to the local sentence/word relative and the entire corpus. All query feature vectors are used to train an unsupervised variational autoencoder, which learns to project the queries into a latent space, with the constraint that it must then accurately recover the original input feature vector from that projection, resulting in similarly structured queries as spatially co-located. An example of the queries projected into a semantic latent space is presented in the lower-left of Figure 6.

The deep learning network uses four decreasingly sized dense layers, to capture interesting co-occurrences of features, each using a Rectified Linear Unit activation function [26] (Relu), defined as $\max(0,x)$, except the last output layer, which uses a Sigmoid, defined as $\frac{1}{1+exp(-x)}$. This network structure is embedded into a Variational Autoencoder, which refines the internal weights. A visual overview of the model is presented in Figure 4. The model is trained using all queries as a pre-processing stage before the visualization starts. The libraries TensorFlow [1] and Keras [6] are used to build, train, and process the deep learning network.

A coarse labeling of the query-space visualization with examples of queries within each group are presented in the left of Figure 6. There is a general trend with alphabetic-based queries appearing on the left, and numeric-based queries appearing on the right, along with a general usage of normal characters on the top, and unusual characters on the bottom. Through our investigation, we have identified several interesting general groups of queries. The most common form of queries are those primarily composed of random configurations of alphabetical characters. Other discovered groups include configurations of IP addresses and numbers, invalid queries which include distinct domains such as `.com` and `.org`, device names with local and home domains, and more strangely, queries consisting of unusual (non-alphanumeric) characters (most likely binary and code fragments).

3.5.1 3D Query Flow Visualization

The enormous scale of the DNS data pushes the limits of dimensionality projection. Earlier solutions would, in general, independently project down segments of the overall high-dimensional data, creating separate local models, resulting in projections that could not be directly compared due to differences in projection axes, or result in a gradual change in axes as more data is projected. Both solutions are undesirable due to the increased cognitive burden of constantly updating one's understanding of the local axes and interpretation of the projection. With deep learning, we are able to remove this effect by training a globally consistent model, providing analysts with a cohesive and consistent representation. In addition, provided a set of projected points, displaying a time-varying scatter-plot is a challenging problem. A common solution is to use animation to convey time. However, research has shown animation incurs higher cognitive loads, can be often difficult to comprehend, and is generally less effective in communication than a static visualization [38]. Therefore we opted for a static approach for visualizing temporal information similar to the IP-space. However, unlike the IP-space which consists of regular interval IP-addresses, the queries have arbitrary distances between each other, requiring a different visualization method. Another solution is to stack individual 2D scatter-plot time-slices, but for large datasets, this leads to an overwhelming amount of information, as well as a large amount of visual clutter. Therefore, we use a volume-rendered marching cubes approach using CUDA and OpenGL. This view is designed, in contrast to the earlier blanket temporal 2D view, to provide analysts with a structural high-level overview of how the query-space changes over time, naturally revealing temporally repeating, absent, and anomalous queries. Similar to the IP-space visualization, the camera in the temporal query-space visualization may be translated, rotated, and zoomed for closer inspection using the keyboard and mouse. An example of this temporal query visualization is presented in Figure 6.

Among many observations, one unexpected finding is that as time continues, the overall number/volume of queries, particularly on the periphery involving unusual queries, decreases. Starting from the beginning of the day, many groups of queries have high volume and variety, in particular, those belonging to the `.pk5001z` group. However, as the day continues, this cluster of queries decreases, with a regularly repeating increase and decrease in volume. Other groups, such as those consisting of invalid IP-address fragments, start with a high volume of traffic, but quickly drop-off, as shown in the top-left of Figure 6.

Another emergent observation is the large quantity of temporally repeating queries, which fade in and out, growing and shrinking in

volume and diversity. A large group of these, consisting of random character queries, have temporally aligned high and low points, as shown in the middle of Figure 6. Others, such as the aforementioned `.pk5001z` group, along with many groups of unusual character queries, such as the group of queries of the form `*_dns-sd.udp_*`, also share an intermittent and oscillatory emission. The discovery of such patterns would have been almost impossible without such a visualization. Due to the oscillatory nature of these queries, it is therefore likely they originate from an erroneous program.

Lastly, the TCP-Syn flood DDOS, as visually portrayed distinctively from the norm in the IP-space visualization as a large increase in activity in TCP-Syn packets, is measured as an absence of queries in the 3D spatiotemporal visualization, notably as the reduction in traffic before the large empty gap in the right of Figure 6. Although we see that some queries are processed, the majority of traffic, particularly on the periphery, has ceased. The large gap, similar to its portrayal in the IP-space, corresponds to a hard-ware failure. In the IP-space visualization, we learned that there was an overall reduction in the number of queries after the attack. In this view, we can also see that only some of the traffic has returned, but now we are informed that primarily those queries consisting of domains are processed, and unexpectedly, those groups of queries on the periphery are mostly absent.

3.6 Dual IP-Query Visualization Interaction

In contrast to previous cyber-visualizations, which focus primarily on the counts of packets, we present a visualization capable of providing analysts with a well-rounded and complete representation of the DNS data. This involves a dual representation, namely an IP and query space. In our system, the IP-space view is presented on the left of the screen, with the query-space shown on the right. Users simply move their cursor from one view to the other to direct their input focus. Until now we have focused on the construction, interaction, and discoveries made with these visualizations independently. In this section we review the interactions and discoveries made when analyzing the D-Root DNS data in a dual-representation.

Suppose the analyst desires to know from which IPs and times a particular query originated. In the query-space visualization, an analyst can double-click on the particular query, to highlight the corresponding bins and times in the IP-space view. We are not filtering out the response packets from the D-Root DNS, so most queries will have at least two IP-space occurrences. To select multiple queries, an analyst holds control+click to brush-select the queries. One selection of queries, presented in the top-left of Figure 7 originate from a wide range of IP-bins and time periods. The selected queries are of the form `*.pk5001z`, whose occurrence in the query-space visualization surprised our DNS experts. Initially, they thought these queries were from one or a few IPs (old hardware), but in the dual interaction, we see that these packets come a wide range of IP addresses for a long duration of time. At the bottom-left of Figure 7, the selected group of queries arrives in three IP-bins, one corresponding to the D-Root, and the others corresponding to sources, indicating that these queries are an anomaly due to the small number of sources. Interestingly, one source is consistent, with the other intermittent. One query was "211.67.67.217.", with the remainder of a similar structure. Using our visualization and interaction methodology, analysts may also to do the reverse, and select individual IP-Time-Bins and visualize the corresponding queries, or select an entire duration by double-clicking the IP bin, as shown in the bottom-right of Figure 7, revealing this temporally oscillating IP-bin (the middle-left from Figure 5 primarily consists of `*.trafficmanager.net` queries. The developed visualizations along with the dynamic interactions enable analysts to visually identify new behavior, develop hypotheses, and overall gain a deeper understanding of the network flow.

4 Empirical Validation

The primary motivation of our visualization is to help analysts understand the distribution of queries they receive, how it changes over time, identify anomalous behavior, and help them explore new questions. Throughout our investigation, we found and reviewed several trends

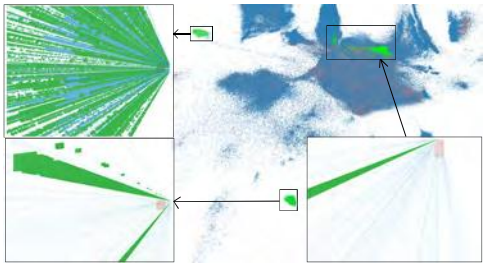

Fig. 7: Two selected regions of queries. The top selection indicates the queries originate from a wide range of IPs, while the bottom selection indicates those queries came from very few IPs. Non-included IPs may be set transparent (bottom) or left un-transparent(top). The bottom-right image shows the resulting queries included from an entire IP-bin.

and anomalies using our visualization. We present a few insights that were discovered by three DNS experts, who will be named A, B, and C. The discoveries made would have been difficult, if not impossible, to make through traditional analysis tools that lack organizing DNS information in an intuitive spatial and temporal representation. The discoveries presented here originate from multiple joint discussions and sessions with both the authors and DNS experts engaging with the visualizations.

4.0.1 DNS Expert A

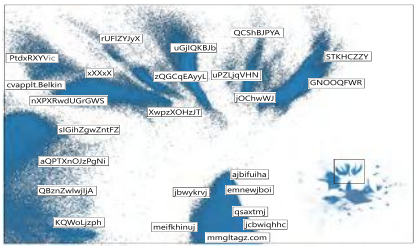

Fig. 8: The region of the query-space consisting of different distributions of random characters.

DNS expert A stated that they would often monitor the overall health with a `pcap` analysis program, and look at a small random selection of packets to see how those queries looked. As expert A was interacting with our visualization, he zoomed into a cluster of points and noted that there was a high volume of lower-case random character queries. He pointed out it was interesting that our visualization could cluster such queries which could come from the Chrome internet browser. Upon further discussion, we learned that when Chrome starts, it tries to learn the nature of the DNS it sits behind by issuing multiple random queries, as ISPs tend to wildcard DNS servers to catch all domains and load advertisements. If the result of the random queries is a valid response, then Chrome knows something is playing strange with the DNS. For those who do not sit behind one of these particular ISPs, these random queries end up at the root to be resolved. While we cannot attribute all of the random lower-case queries to Chrome, it is likely responsible for a large majority. In our visualization, there is a large chunk of the distribution space dedicated solely to sets of random characters, with small differences between them (typically the frequency and capitalization of individual letters), as can be seen in Figure 8.

In addition, DNS expert A was able to learn that many random character queries contained valid domains. When the root encounters such queries, it forwards those queries to the authority domain listed as part of the query. This could lead to a kind of DDOS attack from re-directed queries. With this new information and our visualization, it may be possible to establish filters to mitigate the effects of such queries.

Lastly, DNS expert A noticed a large collection of queries from different routers and modems. Human error or malfunctioning machines often result in erroneous queries. One example of this unusual behavior was the presence of a large number of queries of the form `*.pk5001z`. This initially struck our DNS experts as very unusual, and after some investigation on their end, they found that these types of requests are typically associated with a particular model of modems, namely the PK5001Z flavor of modem. The presence of these queries at the root indicates that someone somewhere has a miss-configured or infected modem sending erroneous queries.In addition, there was an entire distinct cluster dedicated to queries of the form `*.Home`, `*.Belkin`, and `*.local`, indicating erroneous configurations of home routers and devices. The presence of router and modem based queries, while known to our analysts, surprised them by their variety and the age of the originating hardware. In particular, DNS expert A found a set of queries belonging to a 20 year old version of OS VXworks. Using our visualization tool, our DNS experts have been informed on the scope of this problem, and that this traffic can lead to intense bursts when an outdated system desperately searches for a valid DNS response.

4.0.2 DNS Expert B

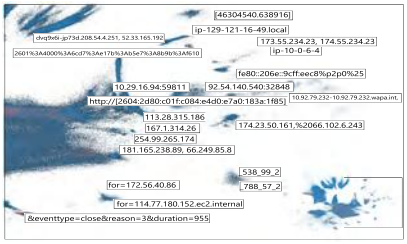

Fig. 9: The region of the query-space consisting of different distributions of IP addresses, fragments, and expressions.

Rarely, people directly enter an IP address into the web-browser to directly connect to a specific IP-enabled device. However, expert B noted queries containing IP addresses, which often contain different mistakes. Our visualization is able to identify and cluster these mistakes. Previously, IP-based investigations primarily use the source IP in the packet header, rather than look at IPs in the query itself. Using our visualization, we can see the distribution of queried IPs and the mistakes made when querying them. DNS expert B found that the most common mistake is the usage of an invalid IP-octet (> 255), or an incorrect port address, often using brackets, dashes, or parenthesis to delineate port. More elaborate mistakes include entering too few or too many octets, surrounding the IPv4 address with brackets and other formatting, or enter two or more IP addresses at once, separated in many different ways. Other errors include partial URLs followed or preceded by IP addresses, erroneous bit masks, IP addresses which replace different numbers with letters (perhaps in an attempt to use IPv6), strange hybrid combinations of URLs with IPv6 IPs, generally in the form of `http://`, and IPs containing many percent symbols, perhaps in an attempt to use a regular expression, or as a fragment from a `printf` statement. An instance could be programs erroneously copying code or URL fragments into a browser DNS query packet. As a result, many of these queries reach the root DNS. In our query-space visualization, there are a few clusters dedicated to these types of queries, as shown in Figure 9.

In addition, we also often find command and instruction segments or simple statements, such as a large occurrence of `for=` statements, boolean expressions, and variable assignments. For other queries, we find many instances of queries structured as `www.` followed by a random collection hexadecimal and unusual characters. We believe these are instances of broken applications going through random permutations of URLs trying to resolve to a valid response.In addition, sending commands through DNS is a common way to control bot-nets. Learning of the occurrence and distribution of these queries have reinforced their belief that the majority of traffic they receive is machine rather than human generated. Just as with the random characters, knowing the types of queries containing IP addresses, and knowing that they cannot be resolved, would allow automatic filtering of such traffic earlier.

4.0.3 DNS Expert C

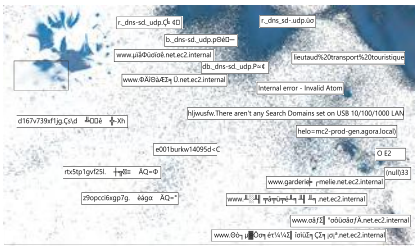

Fig. 10: The region of the query-space consisting of unusual characters and queries.

Expert C noted that there were a large number of queries that contain unusual characters as shown in Figure 10. These characters are those that cannot be interpreted by normal ASCII.Therefore, for our parsing purposes, we relied on the ISO/IEC 8859-1:1998 (also known as Latin-1) character encoding to properly decode and display the queries. The very existence of these queries is unusual, as people do not generally perform queries using such characters. Expert C has theorized that these queries are data and code binary fragments, likely erroneously copied from an invalid buffer. Another theory is they consist of exfiltrated data exploiting the DNS system (DNS tunneling). Possible examples of this were the large occurrence of long queries containing sequences of `prodID=` and other delineated information. Another source of these unusual characters could be bad Unicode translation in software applications. In our IP and temporal query-space visualization, these particular sets of queries tend to fluctuate depending on the hour of the day (with many only being issued in the early morning and late at night as indicated by our temporal query visualization), suggesting that a machine is likely the initiator. Expert C noted that these observations would have been very difficult to make without the usage of our visualization.

5 Conclusions

The goal of our visualization was to provide a natural and easy to use interface for working with large amounts of real-world DNS IP and query data, for providing analysts with a general overview of the distribution of the packet traffic and queries, while also allowing them to investigate small temporal events, individual queries, and find correlations between the IP and query spaces. We have shown that using deep learning to generate spatial representation of non-spatial queries is a very effective method of presenting such data. By working closely with real-world root DNS experts, we have been able to find new and interesting anomalies, groups, and patterns that were previously unknown, and have led to further investigation. As the internet of things is set to grow exponentially, the number of erroneous, malformed, and junk queries is set to explode, as well as increasing complexity and scale of future attacks. Having knowledge of the different types of queries and packet behaviors, what they tend to look like, and how they change over time, may allow for DNS analysts to start automatically filtering these packets as the traffic gradually increases, to keep operations functioning normally. Prior to our visualization, the DNS experts would often only look at a handful of queries at a time, not fully grasping the variety and dynamics of the queries flowing through their network. With this new knowledge and capability, closer inspections of their vast quantities of DNS data may now be conducted, and a greater preparedness for the future may now begin with greater confidence.

6 Acknowledgements

We would like to extend our thanks to Karl Reuss, Bruce Crabill, and Tripti Sinha from the University of Maryland D-Root DNS authority

for their help in providing real-world capture data, and providing continuous guidance in the development and analysis of our visualization and findings.

REFERENCES

[1] M. Abadi et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, vol. 16, pp. 265–283, 2016.

[2] K. Abdullah, C. Lee, G. Conti, and J. A. Copeland. Visualizing network data for intrusion detection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pp. 100–108. IEEE, 2005.

[3] C. Amin, M. Candela, D. Karrenberg, R. Kisteleki, and A. Strikos. Visualization and monitoring for the identification and analysis of DNS issues. In *Proceedings of the Tenth International Conference on Internet Monitoring and Protection*, 2015.

[4] M. Aupetit, Y. Zhauniarovich, G. Vasiliadis, M. Dacier, and Y. Boshmaf. Visualization of actionable knowledge to mitigate DRDoS attacks. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pp. 1–8. IEEE, 2016.

[5] A. Caudwell. *Logstalgia*, 2014. http://logstalgia.io/.

[6] F. Chollet et al. Keras: Deep learning library for Theano and Tensorflow. *URL: https://keras. io/k*, 2015.

[7] G. Conti, M. Ahamad, and J. Stasko. Attacking information visualization system usability overloading and deceiving the human. In *Proceedings of the 2005 symposium on Usable privacy and security*, pp. 89–100. ACM, 2005.

[8] S. G. Eick. Engineering perceptually effective visualizations for abstract data. In *In Scientific Visualization Overviews, Methodologies and Techniques, IEEE Computer Science*. Citeseer, 1995.

[9] G. Fink, C. North, A. Endert, and S. Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pp. 45–56, Oct 2009.

[10] G. A. Fink, C. L. North, A. Endert, and S. Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pp. 45–56. IEEE, 2009.

[11] M. Ghoniem, J.-D. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pp. 17–24. Ieee, 2004.

[12] J. Goodall, W. Lutters, and A. Komlodi. The work of intrusion detection: rethinking the role of security analysts. *AMCIS 2004 Proceedings*, p. 179, 2004.

[13] A. Gracia, S. González, V. Robles, E. Menasalvas, and T. Von Landesberger. New insights into the suitability of the third dimension for visualizing multivariate/multidimensional data: A study based on loss of quality quantification. *Information Visualization*, 15(1):3–30, 2016.

[14] V. T. Guimaraes, C. M. D. S. Freitas, R. Sadre, L. M. R. Tarouco, and L. Z. Granville. A survey on information visualization for network and service management. *IEEE Communications Surveys & Tutorials*, 18(1):285–323, 2016.

[15] ICANN. Factsheet - root server attack on 6 february 2007, 2007. https://www.icann.org/en/system/files/files/factsheet-dns-attack-08mar07-en.pdf.

[16] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese. Network monitoring using traffic dispersion graphs (tdgs). In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 315–320. ACM, 2007.

[17] D. Inoue, M. Eto, K. Suzuki, M. Suzuki, and K. Nakao. Daedalus-viz: novel real-time 3D visualization for darknet monitoring-based alert system. In *Proceedings of the ninth international symposium on visualization for cyber security*, pp. 72–79. ACM, 2012.

[18] N. Jiang, J. Cao, Y. Jin, L. E. Li, and Z. Zhang. Identifying suspicious activities through dns failure graph analysis. In *The 18th IEEE International Conference on Network Protocols*, pp. 144–153, Oct 2010. doi: 10.1109/ICNP.2010.5762763

[19] I. Kim, H. Choi, and H. Lee. BotXrayer: Exposing botnets by visualizing DNS traffic. In *KSII the first International Conference on Internet*, 2009.

[20] Q. Lai, C. Zhou, H. Ma, Z. Wu, and S. Chen. Visualizing and characterizing DNS lookup behaviors via log-mining. *Neurocomputing*, 169:100–109, 2015.

[21] S. Lau. The spinning cube of potential doom. *Communications of the ACM*, 47(6):25–26, 2004.

[22] Y. Livnat, J. Agutter, S. Moon, R. F. Erbacher, and S. Foresti. A visualization paradigm for network intrusion detection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pp. 92–99. IEEE, 2005.

[23] S. Mansfield-Devine. The growth and evolution of DDoS. *Network Security*, 2015(10):13–20, 2015.

[24] S. McKenna, D. Staheli, C. Fulcher, and M. Meyer. Bubblenet: A cyber security dashboard for visualizing patterns. In *Computer Graphics Forum*, vol. 35, pp. 281–290. Wiley Online Library, 2016.

[25] G. Moura, R. d. O. Schmidt, J. Heidemann, W. B. de Vries, M. Muller, L. Wei, and C. Hesselman. Anycast vs. DDoS: Evaluating the november 2015 root DNS event. In *Proceedings of the 2016 Internet Measurement Conference*, pp. 255–270. ACM, 2016.

[26] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[27] T. Nunnally, K. Abdullah, A. S. Uluagac, J. A. Copeland, and R. Beyah. Navsec: A recommender system for 3D network security visualizations. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pp. 41–48. ACM, 2013.

[28] T. Nunnally, P. Chi, K. Abdullah, A. S. Uluagac, J. A. Copeland, and R. Beyah. P3D: A parallel 3D coordinate visualization for advanced network scans. In *Communications (ICC), 2013 IEEE International Conference on*, pp. 2052–2057. IEEE, 2013.

[29] A. Orebaugh, G. Ramirez, and J. Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[30] J. Poco, R. Etemadpour, F. V. Paulovich, T. Long, P. Rosenthal, M. C. F. d. Oliveira, L. Linsen, and R. Minghim. A framework for exploring multidimensional data with 3d projections. In *Computer Graphics Forum*, vol. 30, pp. 1111–1120. Wiley Online Library, 2011.

[31] R. Romero-Gomez, Y. Nadji, and M. Antonakakis. Towards designing effective visualizations for DNS-based network threat analysis. In *Visualization for Cyber Security (VizSec), 2017 IEEE Symposium on*, pp. 1–8. IEEE, 2017.

[32] H.-J. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE transactions on visualization and computer graphics*, 17(4):393–411, 2011.

[33] G. Shan, Y. Wang, M. Xie, H. Lv, and X. Chi. Visual detection of anomalies in DNS query log data. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pp. 258–261. IEEE, 2014.

[34] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani. Botviz: A memory forensic-based botnet detection and visualization approach. In *Security Technology (ICCST), 2017 International Carnahan Conference on*, pp. 1–8. IEEE, 2017.

[35] H. Shiravi, A. Shiravi, and A. A. Ghorbani. A survey of visualization systems for network security. *IEEE Transactions on visualization and computer graphics*, 18(8):1313–1329, 2012.

[36] M. Tavanti and M. Lind. 2d vs 3d, implications on spatial memory. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pp. 139–145. IEEE, 2001.

[37] A. Torres. Building a world-class security operations center: A roadmap. *SANS Institute, May*, 2015.

[38] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: can it facilitate? *International journal of human-computer studies*, 57(4):247–262, 2002.

[39] G. Vigna and R. A. Kemmerer. Netstat: A network-based intrusion detection approach. In *Computer Security Applications Conference, 1998. Proceedings. 14th Annual*, pp. 25–34. IEEE, 1998.

[40] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, vol. 30, pp. 1719–1749. Wiley Online Library, 2011.

[41] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.

[42] A. Yelizarov and D. Gamayunov. Visualization of complex attacks and state of attacked network. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pp. 1–9. IEEE, 2009.

[43] B. Yu, L. Smith, and M. Threefoot. Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 258–271. Springer, 2014.

[44] Y. Zhao, F. Zhou, X. Fan, X. Liang, and Y. Liu. IDSRadar: a real-time visualization framework for IDS alerts. *Science China Information Sciences*, 56(8):1–12, 2013.