UNIVERSITY OF AMSTERDAM

A visual analytics approach to network security hygiene

Tarik El Yassem
System and Network Engineering, University of Amsterdam
(Dated: July 17, 2013)

This paper describes a visual analytics approach to situational awareness of network hygiene. It answers the research question of which methods and techniques can be used to visualize network hygiene. Three visualizations were combined in a single dashboard to visualize three main data categories: state information, communications and network information. The network information and its corresponding hygiene is visualized at various abstraction levels by using a Hilbert curve in a novel way. The system architecture needed to implement the visualization is described, and further research topics are suggested.

I. INTRODUCTION

Security of networks has become an important issue. To keep networks secure, security specialists and IT administrators rely on various sources. These sources come from commercial services, internal security systems and security researchers.

Discussions on disputed hotspots of malicious activity occur within internet communities on a regular basis. In some cases this has resulted in the disconnection of networks such as Attrivo and McColo[1][2] and recently this has even resulted in large scale network attacks and consequently the involvement of law enforcement[50]. These hotspots are sometimes refered to as bad internet neighbourhoods.

Organisations not only have to deal with external information about security issues concerning their public IP space, they also have to deal with information from internal sources with regards to the organizations internal IP space. For example, an organization may receive IDS logging, vulnerability reports and other information with regards to the security state of corporate systems. It is largely up to the network administrators and security personnel to decide if and how to act upon these security notifications. For large networks the administration and monitoring of this information is becoming a bigger challenge of increasing importance. Not having a full situational awareness of the network security status can have dire consequences[51].

We define the overall status of various aspects of a network's security as network security hygiene. We use the term hygiene because it is a fitting metaphor for information security. Much like hygiene in medicine, information security hygiene reflects to practices that are implemented in a preventative way to reduce incidents and spreading of harmful things. Information security hygiene is a quality aspect of an IT infrastructure. A network with a good information security hygiene will be less vulnerable, spread less harmful things, have less abuse notifications and when incidents occur, they will be easier to solve and have a smaller impact.

Visualization may help to inform a wide variety of stakeholders about the overall information security hygiene of a network. Network administrators, ISP sales representatives, policy makers and security personnel on strategic, tactical and operational levels can all benefit from a visualization that allows them to be informed about the status of security hygiene of a network. How can we create a visualization that may accomplish this? What methods and techniques are available? The research question this report answers is *What methods and techniques can be used to visualize network hygiene?*

When it comes to visualizing data, visual analytics[3][4] provides scientific methods to analyze, process and visualize information in ways that enable users to ask relevant questions and gain insights from the data. Visual analytics is a multidisciplinary field and uses a variety of techniques in order to tackle problem areas where the size and complexity of the data require analysis from machines as well as humans. Visual analytics focuses on the data structure and ways this data can be transformed and represented visually. Visual analytics also focuses on techniques of analytical reasoning facilitated by interaction to enable a user to form hypotheses and gain insights. Visual analytics applies very well to the domain of network security hygiene because the data is vast and complex often incomplete or uncertain and various layers of abstraction play a role. The problem domain requires more then computer based analytics, but also requires human intuition, cognition and reasoning. This report describes a way of applying visual analytics techniques on a variety of data with relevance to network security hygiene.

II. RELATED RESEARCH

This section describes research that is related to the topic of network security hygiene and it's visualization. We focus on methods and techniques that can be used to determine and visualize the network security hygiene level of networks.

A. Network security hygiene

Van Eeten[5] has researched the role of Internet Service Providers in botnet mitigation. This study used an empirical analysis based on SPAM data. One of the findings was that infected machines display a highly concentrated pattern. The networks of just 50 ISPs accounted for around half of all infected machines worldwide. Furthermore, the bulk of the infected machines were not located in the networks of obscure or rogue ISPs, but in those of established, well-known ISPs. Another interesting finding was that countries with active Telecom regulators have lower infection rates. Van Eeten[6] further studied this phenomenon with an in-depth study of the Dutch market. The focus of this study lay on infected machines and the results show a clear distinction between different ISPs. The work of van Eeten is mostly relevant because it makes a clear argument that especially the large ISPs play a large role in internet security. According to van Eeten, most industry insiders lack good signals about security problems, except for anecdotal evidence and speculative claims within the security community about the performance of a certain ISP.

Moura[7] has conducted an in depth, systematic and multifaceted study on the concentration of malicious hosts on the Internet. Some conclusions of Moura's work are that the top 20 Autonomous Systems concentrate almost 50% of all spamming IP addresses and that bad neighbourhoods are mostly application-specific and may be located in neighbourhoods one would not expect. Another important finding was the importance of context with regards to the specific type of abuse or security issue and the network where it has been reported. SPAM typically comes from infected clients in ISP networks whereas phishing sites are hosted on reliable infrastructure such as cloud providers or hosting companies. Furthermore Moura has determined that the number of attacks vary per application and bad neighbourhoods are therefore application-specific. The work of Moura is relevant to our research because Moura's methods can be used to aggregate notifications of malicious activity of individual hosts

Kalafut et al[8] have explored whether some ASes indeed are safe havens for malicious activity. They looked for ISPs and ASes that exhibit disproportionately high malicious behaviour using 12 popular blacklists. Their findings were that some ASes have over 80% of their routable IP address space blacklisted and others account for large fractions of blacklisted IPs. Their conclusion is that examining malicious activity at the AS granularity can unearth networks with lax security or those that harbour cybercrime. This research is relevant to us because it makes the case that it is worthwile to abstract the information at the AS level.

Venkataraman et al[9] researched discovering changes in malicious activity across the Internet. They developed algorithms that can automatically infer how malicious IPs, aggregated at AS level, evolve over time. This paper is relevant because the algorithms used also show how information on malicious IPs can be aggregated at the AS level.

HostExploit[52] has released a periodic World Hosts Report since 2009. These quarterly reports provide a ranking of publicly-routed Autonomous System data based on the number of infected websites, botnets, spam and other malicious activity. The ranking algorithm of HostExploit is an example of how malicious activity information can be processed to indicate a certain network security hygiene level.

The research that was referenced above shows clearly that bad networks can be identified and by looking at ISPs and specifically at the autonomous system level. Aggregating information from malicious IPs to AS level helps in this regard.

B. Visualization

While there is a decent amount of research available on analysis of bad hosts and rogue networks, the visualization of this specific area has not been well researched. However, visualization methods for specific threats have received proper attention from the research community. The most relevant visualizations with regards to bad hosts are referenced in this section and more specific visualizations are referenced in later sections.

Roveta et al[10] have developed a visualization of malicious networks at the AS level. They created an interactive visualization that displays autonomous systems exhibiting rogue activity. This helps in finding misbehaving networks through interactive exploration. The paper describes a few limitations, introduced by the use of a bubble chart and a flawed migration heuristic. Unfortunately the visualization is only available as a demo and the data it uses comes from the FIRE[11] system which has been discontinued.

A Pixel-oriented Treemap visualization which visualizes the health and status of about a million devices has been described by Chung et al[12]. The visualization shows many details at once, to make this possible the visualization makes use of multiple displays. This has consequences for access to the system by multiple users and makes user interaction with the visualization more difficult.

Harrison and Lu[13] describe a number of related security visualizations and conclude that the visualizations fall short in relation to the scalability of their visual

metaphors, and the lack of explicit representations of network topology and heterogenous network data.

Another common visualization of networks uses a Hilbert curve to display a network map. This was first used by Randall Munroe[53], and has since been used in a number of academic papers such as [14][15][16][2].

Lalanne[17] et al propose to go beyond the standard visualization for day-to-day monitoring. They suggest a pyramidal model with various levels of time and data granularity, in order to support security engineers, analysts and managers.

Drawing from the research above, we can envision a visualization that is scalable, visualizes the state of an AS, represents the full address space of a network and presents the data in various levels of aggregation.

III. APPROACH

The introduction has described two problems relating to security issues in networks. The first one takes the external perspective of badness emanating from networks, while the second one takes the internal perspective of the situational awareness of network security hygiene. These two seemingly different problems influence each other and are similar in nature when viewed from a higher level of abstraction. The main similarities between these two problems are that unwanted things happen with a certain IP address or network at a given time, and that different of these unwanted things can have a varying impact. With large networks of interest and large event data sets available, the challenge is in the representation of the data.

A visual analytical approach was taken in order to design a proof of concept visualization system that can be used to visualize security status of networks. The aim is to provide a general proof of concept visualization that is applicable to a wide array of security related events occurring on various networks.

Keim's[4] visual analytics process was followed with regards of data sources, visualizing and the theoretical formation of hypotheses the visualization should assist a user in forming. To make this possible, the focus lay in part on the data analysis and implementation of the backend system. The insight process described in Keim's model remained out of scope because the visualization was not completely implemented. The used model which was based on Keim's is illustrated in figure 1.

The model depicts data preprocessing as S, visualization as V and hypotheses forming as H. The data preprocessing stage includes data selection, data cleaning, data transformation and data integration. Hypotheses were formed from the data directly, from the created visualizations as well as through reasoning upon earlier hypotheses. Visualizations were created from analysis of the data and from formed hypotheses. Visualizations were improved which allowed in turn for new hypotheses to be formed. The model shows the process flow between

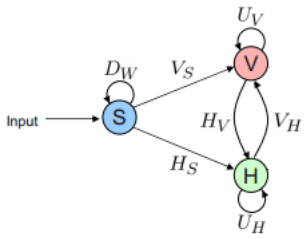


FIG. 1: Visual analytics process as applied in this research project.

data preprocessing, hypotheses formation and visualization.

IV. DATASOURCES

This section describes the available datasources. Two basic datasources are relevant: the network of interest and security events.

A. network

At the highest level we consider the Autonomous System. Each Autonomous System covers one or more netblocks. These netblocks can have a varying size. This covers the external perspective. However, an organisation can have multiple ASs. Netblocks get subdivided into subnetworks. Finally the lowest level of a network is a single IP address.

- AS
- Netblock
- Subnet
- IP

These networks fall within the responsibility of various levels of an organization. The administration of this varies between organizations and usually this information is not maintained very well and is not publicly available. Therefore we focus on AS, netblock and IP address information which is available in Whois databases and through BGP and will not consider subnets and other possible aggregation levels. This data could be highly structured, given the clear hierarchy of organization, AS, netblock and IP address. In theory this could be represented as a tree. However, it depends on the level of administration and the source of this data if such a structure is practically usable. To keep a more generic approach, the visualization should be able to deal with the structure of network data in a flexible way. A characteristic of netblocks, subnets and IP addresses is that there is a meaning in their numbering. Numbers that

are close to each other imply that the netblocks, subnets and IP addresses are also close to each other in a network. This closeness could reflect that IP addresses belong to the same subnets for example.

B. security events

We define security events at the highest level as something that applies to some IP address that has occurred at a given time, has a certain risk involved with it and comes from a certain source. Security events can have the following attributes:

- Risk
- Time
- Source
- Type
- IP

Risk is an expression of the seriousness of a given security issue. In many cases it is a combination of the chance of something happening and its impact. For convenience we discern three levels of chance and impact, low, medium and high. The risk calculations are out of scope since they are not very relevant for the proof of concept system. Time is an interesting aspect because in many cases the notification time, or time when some event first occurred is known. However, it is sometimes unknown if a certain event has seized to exist. The issue may remain or may have been solved after it has first been seen. This causes possible problems when the visualization contains time lines for multiple events. Type specifies the category of security events. Security events can be notifications of SPAM, IDS logs, vulnerability reports, notice and take down requests from law enforcement, intelligence feeds on botnet activity and so on. We can categorize these events with the following categories:

- Vulnerability
- Attack
- Abuse
- Notice and take down request

A vulnerability is something that is reported on by a vulnerability scanner for example. It signals a misconfiguration or vulnerable software on a given host. There is usually a chance and impact score given to a certain vulnerability, together they form the risk. An attack is something that can be signalled by firewalls, intrusion detection and prevention systems. It signals an active exploitation attempt of a certain vulnerability. We define abuse as unwanted actions that emanate from a given system. This usually happens after a system has been compromised. But it could also be a misbehaving user. Communicating with a botnet command and control server, or sending SPAM are examples of abuse. Notice and take down(NTD) are requests to remove content, for example because the material is illegal. Any of the types within these categories can have a certain importance. The importance of these events depends on policy. This importance can be included in the risk calculation. These security events can be categorised by security state events or communication events. Security state events are vulnerabilities, abuse, and notice and take down requests. Attacks always fall in the communication events category because there is always a source and destination involved in an attack. This results in the following data categorization:

Type	Category	Examples
Vulnerability	state	patch information, vulnerability scan logs
Attack	communication	IDS, IPS, firewall logs, network traffic captures
Abuse	state	SPAM notification, phishing
NTD	state	requests for removal of content

V. VISUALIZATION

The ultimate goal of the visualization is to gain insight into the network security hygiene of networks. To design a system that could make this possible, we consider both internal, corporate networks as well as networks from an internet perspective. The system must be able to handle vast amounts of heterogeneous data. The system should be flexible in the support of various data formats because security and network related information can quickly change. The formats of these data sources also tend to vary and change over time. The users of the system operate on operational, tactical and strategic levels. Therefore the visualization should work on various abstraction levels. Because the user base varies, accessibility to the system is important as well.

The previous section has described the data sources. This section will motivate the chosen visualization for each data source category.

Visualizing both network information, security state information and communication might allow the user to formulate hypotheses and gain insight into the state of a network's information security hygiene level. An interactive dashboard was chosen as it can show the most important information to the user in a single screen as argued by Few[18]. It could be argued that the visualization is not a dashboard because dashboards may not be interactive but instead the visualization is better described as an interactive multiview visualization. There is no clear established definition of information dashboards. Dashboard design rules do apply to this visualization and therefore dashboard seems to be a proper description. Within this dashboard we aim to visualize each of the categories. Now follows a description and motivation of the visualization for each category but the main focus lies on the network information visualization.

A. Network security hygiene visualization

For each organization of interest an overview must be visualized to show the information security hygiene level of autonomous systems, netblocks or individual hosts, depending on the level of abstraction. The total number of autonomous systems of interest, and the size of the netblocks can vary. The challenge is to present the information in a single view which should remain the same size independent of the host or networks shown. A common approach to this is the usage of treemaps[19][20][21]. However, proper usage of treemaps requires data that has a tree structure. As mentioned in the data sources section, this might not always be the case. The approach should be widely applicable and the structure of network information may not always be evident. Therefore treemaps are not an option. Another popular approach is the use of pixel based mapping[22][23][24][25]. The downside of this approach is that the image in which the pixels are mapped can vary according to the amount of information available. Space filling curves[26], a type of fractal, can solve this problem. Of the various types of space filling curves, the Hilbert curve[27][28][29], has suitable qualities. It allows to draw a matrix of a varying number of squares in a fixed space. The sequential layout allows for a visual map of values that are logically close to each other. In other words, a Hilbert curve allows for the preservation of the locality of the original data items. IP addresses that are numerically close to each other are close to each other within the visualization as well. This property is the main reason for choosing a Hilbert curve as opposed to other space filling curve such as the z-curve(also known as Peano, Morton encoding, quad code, bit interleaving or N-order) or gray-curve as demonstrated by Moon et al[30] and Mokbel[31]. The reason this property is important is that it allows a user to find the location of a given IP address within the visualization. Hilbert curves have been used to visualize the complete IPv4 space by mapping each /8 to a tile in a 256 tile grid. Randall Munroe, author of the XKCD web comic was the first to use this method[32]. Since then a number of scientific works[16][2][29][15] have made use of this approach. The properties of the Hilbert curve allow us to create one visualization for organizations, autonomous systems and netblocks of varying sizes. We can map organizations, autonomous systems, netblocks and IP addresses in the squares. One soon notices that some Hilbert curves map neatly to the different network classes, but not as neatly to many CIDR prefixes. This has been described by Irwin and Pilkinton[33]. The research into the usage of Hilbert curves for IPv4 CIDR networks and IPv6 seems to have come to a halt. The technique has proven itself useful in visualising internet scale phenomena for IPv4 classfull addresses. Given the usefulness of the Hilbert curve, an effort will be made to research the applicability of this technique in classless IPv4 and IPv6 networks, of varying prefix sizes.

The following example demonstrates a Hilbert curve implementation in D3. It binds IP addresses and status information to a square. A square represents an organization, autonomous system, netblock or single host. However, it is easy to extend this to more levels in case other hierarchical elements such as subnets or departments need to be represented. Each of these can have a scoring that ranges from none, low, medium to high. The tile is coloured according to the scoring. Figure 2 shows an implementation of Hilbert curves with an order of 1, 2, 4 and 7. Hilbert curves with an order higher then 7 cause performance issues on standard browser configurations. Appendix A contains tables with the appropriate minimal Hilbert order for popular IPv4 and IPv6 CIDR prefixes. Appendix B contains a few examples of these. IPv4 CIDR prefixes of /18 and higher can be represented fully, in some cases the Hilbert curve itself will not be filled fully. Prefixes of /17 and larger cannot be fully represented in our Hilbert curve implementation because the amount of datapoints is larger then the maximum number of items the implemented Hilbert curve allows. For IPv6 CIDR prefixes that map well are /48 and /56. The /24, /32 and /48 prefixes cannot be represented fully in our Hilbert curve implementation. For the prefixes that cannot be mapped, we must make use of filtering and abstraction.

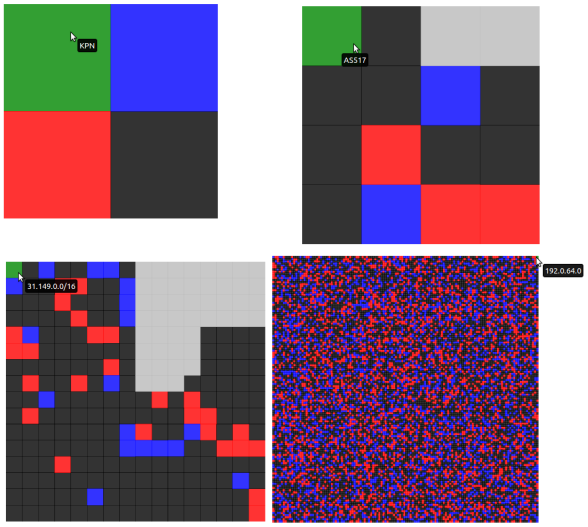


FIG. 2: The Hilbert curves of various orders.

Initially the designs included an embedded visualization within each Hilbert tile to visually communicate the levels of the various categories. However, this approach would not work well with high order Hilbert curves as the individual squares become too small. The network security hygiene level is visualized simply by using a coloured block to represent one of three different levels, low, medium and high. These levels correspond to what is commonly used to communicate information security risk levels. Keeping it simply at three levels was chosen over using more sophisticated scoring systems. The reasoning behind this is that abstract scores with a high

granularity might not entice the user to take direct action because it may be unclear what the policy for a given score is. The color palette of black, blue and red was chosen after some experimentation. Most color combinations would work at lower Hilbert dimensions but would become too unappealing at higher dimensions. Furthermore the experiments showed that having two colors contrast the black gave the colors a visual distinction. The obvious choice was to pick red to represent high risk, and black for low risk, leaving blue to represent medium risk. We implement these three levels as a cumulative of the risk levels for that specific network. When there is not enough data to fill the Hilbert curve, tiles are given a grey color. White was used first but because of the patterns of the Hilbert curve this causes an unpleasant user experience. This is likely due to the fact that the visual perception and thinking system of humans[34] tends to search for recognizable patterns. Giving the non-data squares a light grey color however does facilitate visual closure. The concept of closure is part of the Gestalt laws, more information on the Gestalt laws can be found in [35] and [36].

B. Status display

The status display should contain a visualization suitable for multivariate data. The most relevant visualizations are statistical visualizations. Usable visualizations include barcharts, stacked barcharts, bullet graphs and spark lines for example. Barcharts were chosen as they can show a bar for each category, and within each bar we can show the amount of each risk level with a certain colour. The barchart should be interactive[37] and the information it shows should correlate with the data selection in the other parts of the dashboard. Due to time restrictions, this interaction has unfortunately not yet been implemented.

C. Communication display

There are many visualizations for network communication that could be used in a dashboard of this kind. Hierarchical Edge Bundling[38] provides a way to visualize network traffic in an abstracted form. The main advantage of Hierarchical Edge Bundling is that it can be used to visually communicate traffic flows to and from hosts. Parallel coordinates[39][40][41] can also be used, for example to visualize netflow data[42]. More complex visualizations are also popular, for example to display intrusion detection data as demonstrated by Visalert[43][44]. Another approach is to use graphs in a way such as used by Tsigkas et al[45]. This method visualizes various attacks by representing the associated infrastructure as nodes in a network. It relies on information about a given infrastructure to be known.

A Hierarchical Edge Bundle was chosen to visualize communication. The main reason for this choice was that it visually maintains the logical locality of IP addresses in the visualization. This allows a user to quickly identify subnets which are in trouble. Another reason to choose the Hierarchical Edge Bundle over the other suitable visualizations is that it can be applied for complex as well as simple source data. Finally, the radial Hierarchical Edge Bundle is aesthetically pleasing, an important aspect for inclusion in a view and part of Few's[18] dashboard design criteria. While arguably not a dashboard, the A work in progress is to implement user-interaction[37] to allow for filtering, selection, marking of various risk levels with an appropriate color and to present details on demand. A downside of the radial hierarchical edge bundle that was used in the proof of concept visualization is that it has trouble showing large amounts of different hosts. This also results in unreadable labels. These are implementation issues that can be solved with a filter which can be presented to the user or which can be implemented in the visualization system code. Another way to partially solve this problem is by extending the visualization with a fisheye distortion[54].

D. Dashboard

Figure 3 shows the proof of concept dashboard with a network overview with status information, categories display and communication display.

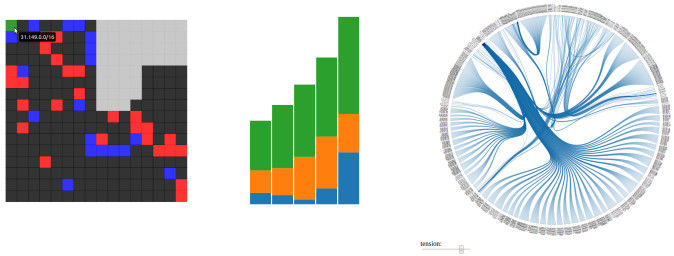


FIG. 3: The proof of concept dashboard.

VI. IMPLEMENTATION

This section describes the system architecture and some of the implementation details and choices.

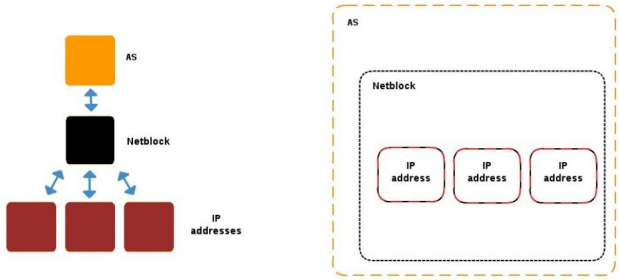
The system must be able to handle a variety of data sources and should be flexible. Furthermore, the system must be easy to use and accessible to users. The choice was made to implement the basic architecture as a three-tier architecture in which the data management, application logic and presentation are implemented as separate layers.

The choice of a database system for the backend was to either use a traditional relational database or a modern

NoSQL solution. Relational databases such as MySQL provide good support for operations on IP addresses, but are not as flexible in storage of various data formats and present many challenges in dealing with vast amounts of data. Therefore, NoSQL was chosen. Within the NoSQL category there are two main database type systems: key-value oriented and document oriented. Because the main source of data are reports of various kinds, the obvious choice was to select a document oriented database. Popular NoSQL, document oriented databases are ElasticSearch, CouchDB and MongoDB. MongoDB[46] was chosen because it has decent security features and plenty of available documentation. Furthermore sharding provides horizontal scalability by spreading the workload over multiple machines. Another reason to choose MongoDB is the feature known as capped collections. Capped collections allow for fixed size, circular collections. When a collection runs out of space, it will overwrite the oldest documents. This allows for flexibility in the choice of data retention versus limited resources. Another important reason to choose MongoDB was the fact that storage and queries are done in JSON format. Furthermore, MongoDB allows for JavaScript applications to be executed on the database. This feature could be used to calculate scores on the database when new data is loaded. An important design choice is the database schema for the different data collections. While MongoDB is very flexible, it is primarily meant to store documents. The challenge lay in storing the network information. Each AS, netblock and all IP addresses should be stored in such a way that the security level can be stored with it. MongoDB schemas are mostly implemented as embedded documents[55]. Embedded documents are flat document structures where data is presented hierarchically within a single document. Using an embedded document schema design causes problems when large volumes of IP addresses are stored under a single netblock or AS. When a query is made for a single IP, MongoDB returns the whole document where the given IP address is found in. The ratio of wanted output versus unwanted output would be completely disproportionate and would require filtering and thus unnecessary complexity and system resources. The right way to implement this is by using a normalized schema with references. Figure 4 shows a referenced approach on the left and an embedded approach on the right.

The middleware has been implemented in Node.js[56]. Node.js is a server side JavaScript framework. The main reason for choosing Node.js is that it works well with MongoDB and because its programming language is also JavaScript, it would allow for easy transfer of code between database and middleware. Node.js can run a webserver and we chose the Express web application framework[57] to provide a model-view-controller setup. Templating functionality is provided by ejs[58]. A basic REST service was implemented which allows browsers to issue queries through the middleware. The REST API can call a filter function on the database output.

FIG. 4: MongoDB schemas, referencing v.s. embedding.



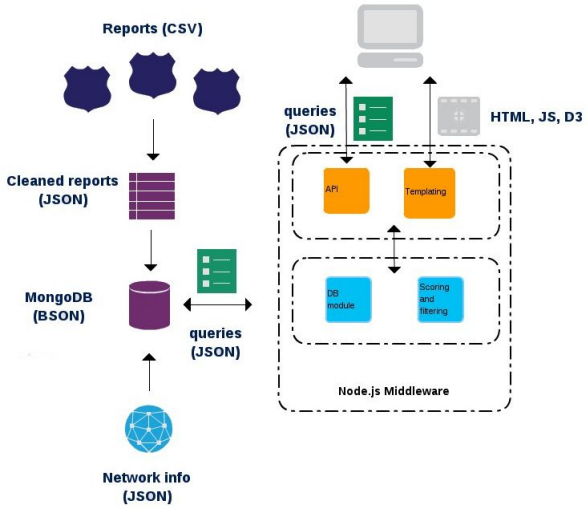
The presentation layer is handled by the client browser which executes JavaScripts via the D3[47] visualization framework. D3 was chosen mainly because of the availability of many visualizations. The Hierarchical edge bundle of Mike Bostock[59] was adapted so that it shows IP addresses and groups them well. The stacked barchart was adapted from Murray[48]. The Hilbert curve implementation was based on [60][61][62][63]. The Hilbert curve is interactive, clicking on a certain square will result in a newly drawn Hilbert curve with a higher order. This new Hilbert curve will display the appropriate data. Hilbert curves with orders of 5, 6 and 7 are used to display IP addresses of a given netblock. After this, the Hilbert curve will cycle through to level 1. Upon drawing of the Hilbert curve, the appropriate API call is made to the middleware in order to retrieve the data. From Hilbert curve order 5, the size of the retrieved data determines if a 5th order Hilbert curve is sufficient or if order 6 or 7 is more appropriate.

Filtering can be implemented at the middleware by sorting the database output on the risk level, placing the results in a fixed size buffer and then reordering this buffer on IP addresses. This way, a filter is implemented that returns the IP addresses with the highest risk level, sorted on IP address. This makes sure that the locality is preserved in the Hilbert curve. Appendix D demonstrates the use of locality preservation. It is also possible to fully filter on the client side or even on the database. Client side filtering has the drawback of increased browser resource requirements and the advantage of flexibility with regards to transformations due to user interaction. Filtering on the database level has the advantage of offloading load to the database, which is scalable and the disadvantage of being less flexible with regards to the visualization. It would result in the client having to issue more GET requests to the middleware API which might cause unwanted delays. These alternative filtering strategies have not been tested. Filtering on the middleware layer was chosen because it offered the required flexibility while providing good performance. Appendix C contains a flowchart which illustrates the client side process of drawing the Hilbert curve.

Figure 5 displays the system architecture. Data input is delivered in various CSV formats, a Ruby script is used

for data cleaning and transformation to JSON. Network information was gained from Whois databases. For test and demonstration purposes bash scripts were used to generate test data.

FIG. 5: A schematic overview of the system architecture.



VII. CONCLUSION

In this report we have looked at the methods and techniques which can be used to visualize network hygiene at various abstraction levels. From related research we have learned that bad network neighbourhoods exist, found ways to discover them and explored existing visualizations that could be applied to visualize network security hygiene. A number of suitable visualizations were found, but no single visualization would be sufficient. Therefore a new approach was taken. To come up with a useful visualization we combined three visualizations in a single dashboard to visualize the three main data categories. For state information we use visualized statistics in the form of a stacked barchart. For communications we used a hierarchical edge bundle. The network information and it's corresponding network security hygiene level was visualized at various abstraction levels by an interactive Hilbert curve.

This Hilbert curve implementation was tested for various popular IPv4 and IPv6 CIDR prefixes. We found that this approach works in most cases for IPv4 prefixes, and for some IPv6 prefixes. This part of the dashboard functions very well, both for small as well as big networks. It's usefulness extends beyond the problem of network security hygiene visualization.

A three-tier web application that visualizes network security hygiene using the three visualizations has been implemented. The system is highly flexible when it comes to it's data sources and the system is easy to access and adapt.

The visualizations have not been fully implemented due to time constraints. User interaction as described by Yi et al[37] was not fully implemented. This is an important aspect of the visualization which will be developed further. Without these interactions the full power of the visualization, the combination of the three separate visualizations, cannot be harnessed. Therefore, the visualization has not been evaluated[49] and no conclusions about the effectiveness of the visualization can be drawn. This remains for future work.

The main contributions of this paper are the description of the current research related to network hygiene and its visualization, the novel approach of using the Hilbert curve in an interactive way to visualize CIDR prefixes and the suggested system architecture and proof of concept visualization.

First impressions paint a positive picture and provide sufficient reasons to further develop and research this system.

VIII. FURTHER RESEARCH

The main subject that remains to be further researched is the effectiveness of the system. To measure what insights a user might gain from the system, it first needs to be developed further. The features which should be developed further are mainly the interactions between the user and the visualization and the interactions between the three separate visualizations.

Another research subject could be an in depth look at various filtering mechanisms that can be applied to the Hilbert curve or the data it is provided. This is necessary in cases where there is too much data for the higher order Hilbert curves, which apply to low IPv4 CIDR prefixes and IPv6. Another possible solution could be to introduce subnets as an intermediate abstraction layer. This would limit the total number of IP addresses that need to be plotted in a given Hilbert curve to the IP addresses of the given subnet. The challenge in this approach lies in the method of choosing relevant subnet sizes, especially if information on which subnets are used is not available.

Finally, the varying temporal aspects of security state information could be researched. How can this information be visualized with regards to time? If there is no information available on when a given security issue seizes to exist, what strategies could we apply to deal with this visually? The same questions apply to the risk level of events for which there is no end time known. One way to deal with this could be to solve this using a temporal factor in the mathematical risk calculation model. Another way could be integration with other systems, such as ticketing systems used by IT helpdesks, abuse desks or incident response teams.

IX. ACKNOWLEDGEMENTS

I wish to thank Jaya Baloo, Martijn van der Heide, Folkert Visser, Mandy Kaandorp, KPN's CISO office, KPN-CERT and KPN-SOC for the hospitality, enthusiasm, feedback and assistance.

My thanks also go out to Wouter Katz for giving me valuable pointers when I got stuck implementing the visualization.

I also wish to thank Marcel Worring for valuable advice and guidance on this research project. The OS3 team also deserves my gratitude, especially Karst Koymans and Jaap van Ginkel, for providing a fun and inspiring learning environment, guidance and interesting opportunities.

I am grateful to the National Cyber Security Centre of the Netherlands for providing me the opportunity to pursue this master. My colleagues at the NCSC are thanked for taking care of business while I was not around. I want to thank Aart Jochem for his patience and understanding. To Elly van den Heuvel I owe the greatest of thanks. Thank you for making this possible and encouraging me to seek out, challenge and develop myself.

Finally I wish to thank my family and friends for supporting me through tough times over the past two years.

References

[1] Richard Clayton. How much did shutting down mccolo help. *Proc. of 6th CEAS*, 2009.

[2] Steve DiBenedetto, Dan Massey, Christos Papadopoulos, and Patrick J Walsh. Analyzing the aftermath of the mccolo shutdown. In *Proceedings of the 2009 Ninth Annual International Symposium on Applications and the Internet-Volume 00*, pages 157–160. IEEE Computer Society, 2009.

[3] James J Thomas and Kristin A Cook. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, 2005.

[4] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. *Visual analytics: Scope and challenges*. Springer, 2008.

[5] Michael Van Eeten, Johannes Bauer, Hadi Asgharia, Shirin Tabatabaie, and David Rand. The role of internet service providers in botnet mitigation an empirical analysis based on spam data. TPRC, 2010.

[6] Michel JG van Eeten, Hadi Asghari, Johannes M Bauer, and Shirin Tabatabaie. Internet service providers and botnet mitigation. 2011.

[7] GC Moreira Moura. Internet bad neighborhoods. 2013.

[8] Andrew J Kalafut, Craig A Shue, and Minaxi Gupta. Malicious hubs: detecting abnormally malicious autonomous systems. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE, 2010.

[9] Shobha Venkataraman, David Brumley, Subhabrata Sen, and Oliver Spatscheck. Automatically inferring the evolution of malicious activity on the internet.

[10] Francesco Roveta, Luca Di Mario, Federico Maggi, Giorgio Caviglia, Stefano Zanero, and Paolo Ciuccarelli. Burn: Baring unknown rogue networks. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec)*, page 6, 2011.

[11] Brett Stone-Gross, Christopher Kruegel, Kevin Almeroth, Andreas Moser, and Engin Kirda. Fire: Finding rogue networks. In *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pages 231–240. IEEE, 2009.

[12] Haeyong Chung, Yong Ju Cho, Jessica Self, and Chris North. Pixel-oriented treemap for multiple displays. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 289–290. IEEE, 2012.

[13] Lane Harrison and Aidong Lu. The future of security visualization: Lessons from network visualization. *Network, IEEE*, 26(6):6–11, 2012.

[14] Alistair King, Bradley Huffaker, Alberto Dainotti, et al. A coordinated view of the temporal evolution of large-scale internet events. *Computing*, pages 1–13, 2012.

[15] Alberto Dainotti, Alistair King, Ferdinando Papale, Antonio Pescapè, et al. Analysis of a/0 stealth scan from a botnet. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 1–14. ACM, 2012.

[16] Jan Stanek and Lukas Kencl. Sipp-dd: Sip ddos flood-attack simulation tool. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1–7. IEEE, 2011.

[17] Denis Lalanne, Enrico Bertini, Patrick Hertzog, and Pedro Bados. Visual analysis of corporate network intelligence: abstracting and reasoning on yesterdays for acting today. In *VizSEC 2007*, pages 115–130. Springer, 2008.

[18] Stephen Few. *Information dashboard design*. O'Reilly, 2006.

[19] Joseph H Goldberg and Jonathan I Helfman. Enterprise network monitoring using treemaps. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 671–675. SAGE Publications, 2005.

[20] Florian Mansmann, Daniel A Keim, Stephen C North, Brian Rexroad, and Daniel Sheleheda. Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1105–1112, 2007.

[21] Florian Mansmann, Fabian Fischer, Daniel A Keim, and Stephen C North. Visual support for analyzing network traffic and intrusion detection events using treemap and graph representations. In *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, page 3. ACM, 2009.

[22] Soon Tee Teoh, Kwan Liu Ma, S Felix Wu, and Xiaoliang Zhao. Case study: Interactive visualization for internet security. In *Proceedings of the conference on Visualization'02*, pages 505–508. IEEE Computer Society, 2002.

[23] Ben Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. In *SIGMOD Conference*, pages 3–12, 2008.

[24] Daniel A Keim. Designing pixel-oriented visualization techniques: Theory and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):59–78, 2000.

[25] David Barrera and PC van Oorschot. Security visualization tools and ipv6 addresses. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 21–26. IEEE, 2009.

[26] Martin Wattenberg. A note on space-filling visualizations and space-filling curves. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 181–186. IEEE, 2005.

[27] Barry Vivian William Irwin. *A framework for the application of network telescope sensors in a global IP network*. PhD thesis, Rhodes University, 2011.

[28] Olivier Thonnard, Wim Mees, and Marc Dacier. Behavioral analysis of zombie armies. *The Virtual Battlefield: Perspectives on Cyber Warfare*, 3:191–210, 2009.

[29] Ward Van Wanrooij and Aiko Pras. Filtering spam from bad neighborhoods. *International Journal of Network Management*, 20(6):433–444, 2010.

[30] Bongki Moon, Hosagrahar V Jagadish, Christos Faloutsos, and Joel H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *Knowledge and Data Engineering, IEEE Transactions on*, 13(1):124–141, 2001.

[31] Mohamed F Mokbel, Walid G Aref, and Ibrahim Kamel. Analysis of multi-dimensional space-filling curves. *GeoInformatica*, 7(3):179–209, 2003.

[32] R Munroe. xkcd: Map of the internet, 2006.

[33] Barry Irwin and Nick Pilkington. High level internet scale traffic visualization using hilbert curve mapping. In *VizSEC 2007*, pages 147–158. Springer, 2008.

[34] Colin Ware. Visual queries: The foundation of visual thinking. In *Knowledge and information visualization*, pages 27–35. Springer, 2005.

[35] Dempsey Chang, Laurence Dooley, and Juhani E Tuovinen. Gestalt theory in visual screen design: a new look at an old subject. In *Proceedings of the Seventh world conference on computers in education conference on Computers in education: Australian topics-Volume 8*, pages 5–12. Australian Computer Society, Inc., 2002.

[36] Karen Smith-Gratto and Mercedes M Fisher. Gestalt theory: a foundation for instructional screen design. *Journal of Educational Technology Systems*, 27:361–372, 1999.

[37] Ji Soo Yi, Youn ah Kang, John T Stasko, and Julie A Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, 2007.

[38] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[39] Sebastien Tricaud and Philippe Saade. Applied parallel coordinates for logs and network traffic attack analysis. *Journal in computer virology*, 6(1):1–29, 2010.

[40] Gabriel D Cavalcante, Sebastien Tricaud, Cleber P Souza, and Paulo Licio de Geus. Interactive analysis of computer scenarios through parallel coordinates graphics. In *Computational Science and Its Applications–ICCSA 2012*, pages 314–325. Springer, 2012.

[41] Kevin T McDonnell and Klaus Mueller. Illustrative parallel coordinates. In *Computer Graphics Forum*, volume 27, pages 1031–1038. Wiley Online Library, 2008.

[42] Xiaoxin Yin, William Yurcik, Michael Treaster, Yifan Li, and Kiran Lakkaraju. Visflowconnect: netflow visualizations of link relationships for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 26–34. ACM, 2004.

[43] Yarden Livnat, Jim Agutter, Shaun Moon, Robert F Erbacher, and Stefano Foresti. A visualization paradigm for network intrusion detection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 92–99. IEEE, 2005.

[44] Stefano Foresti and James Agutter. Visalert: From idea to product. In *VizSEC 2007*, pages 159–174. Springer, 2008.

[45] Orestis Tsigkas, Olivier Thonnard, and Dimitrios Tzovaras. Visual spam campaigns analysis using abstract graphs representation. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, pages 64–71. ACM, 2012.

[46] Reuven M Lerner. At the forge: Mongodb. *Linux Journal*, 2010(193):5, 2010.

[47] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[48] Scott Murray. *Interactive Data Visualization for the Web*. O'Reilly Media, 2013.

[49] Chris North. Toward measuring visualization insight. *Computer Graphics and Applications, IEEE*, 26(3):6–9, 2006.

[50] http://arstechnica.com/security/2013/03/spamhaus-ddos-grows-to-internet-threatening-size/

[51] http://www.reuters.com/article/2012/02/10/kpn-idUSL5E8DACNB20120210

[52] http://hostexploit.com/

[53] http://xkcd.com/195/

[54] http://bost.ocks.org/mike/fisheye/

[55] http://docs.mongodb.org/manual/core/data-modeling/

[56] http://www.nodejs.org

[57] http://expressjs.com/

[58] http://embeddedjs.com/

[59] https://gist.github.com/mbostock/1044242

[60] http://en.wikipedia.org/wiki/Hilbert_curve

[61] http://bit.ly/biWkkq

[62] http://bl.ocks.org/597287

[63] http://www.jasondavies.com/hilbert-curve/

Appendix A: Hilbert curve orders for popular CIDR prefixes

Hilbert curve orders for IPv4 popular CIDR prefixes:

Prefix	IPs	Hilbert order	Max items	Fill	Comment
16	65535	7	16382	large overflow	
17	32766	7	16382	overflow	
18	16382	7	16382	100.00%	browser limit
19	8190	7	16382	50.00%	
20	4094	6	4094	100.00%	
21	2046	6	4094	50.00%	
22	1022	5	1024	100.00%	
23	510	5	1024	50.00%	
24	254	4	256	100.00%	
25	128	3	256	50.00%	
26	64	3	64	100.00%	
27	32	3	32	100.00%	
28	16	2	16	100.00%	
29	8	2	16	50.00%	
30	4	1	4	100.00%	
31	2	1	4	50.00%	
32	1	1	4	25.00%	

Hilbert curve orders for popular IPv6 CIDR prefixes:

Prefix	IPs	Hilbert Order	Max items	Fill
24	16,777,216 subscriber sites	7	16382	overflow
32	65,536 /48 networks	7	16382	large overflow
48	256 /56 networks	4	256	100.00%
48	65, 536 64 LANs	7	16382	large overflow
56	256 64 LANs	4	256	100.00%

Appendix B: Hilbert curve examples

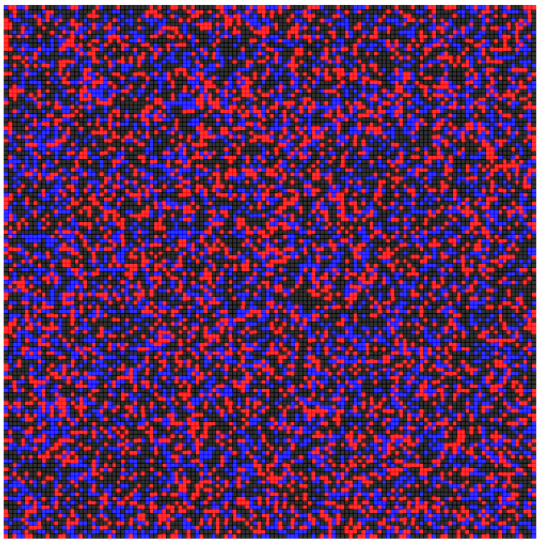


FIG. 6: Mapping a /16 in a 7th order Hilbert curve.

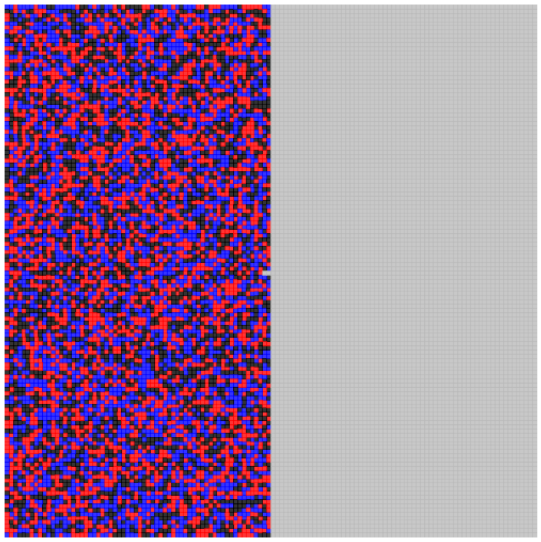


FIG. 7: Mapping a /19 in a 7th order Hilbert curve.

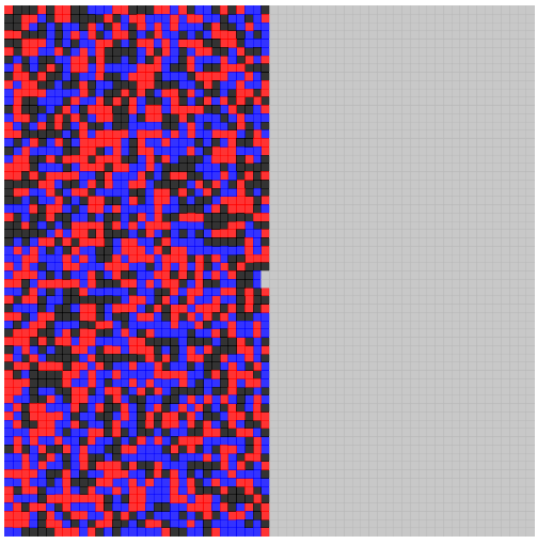
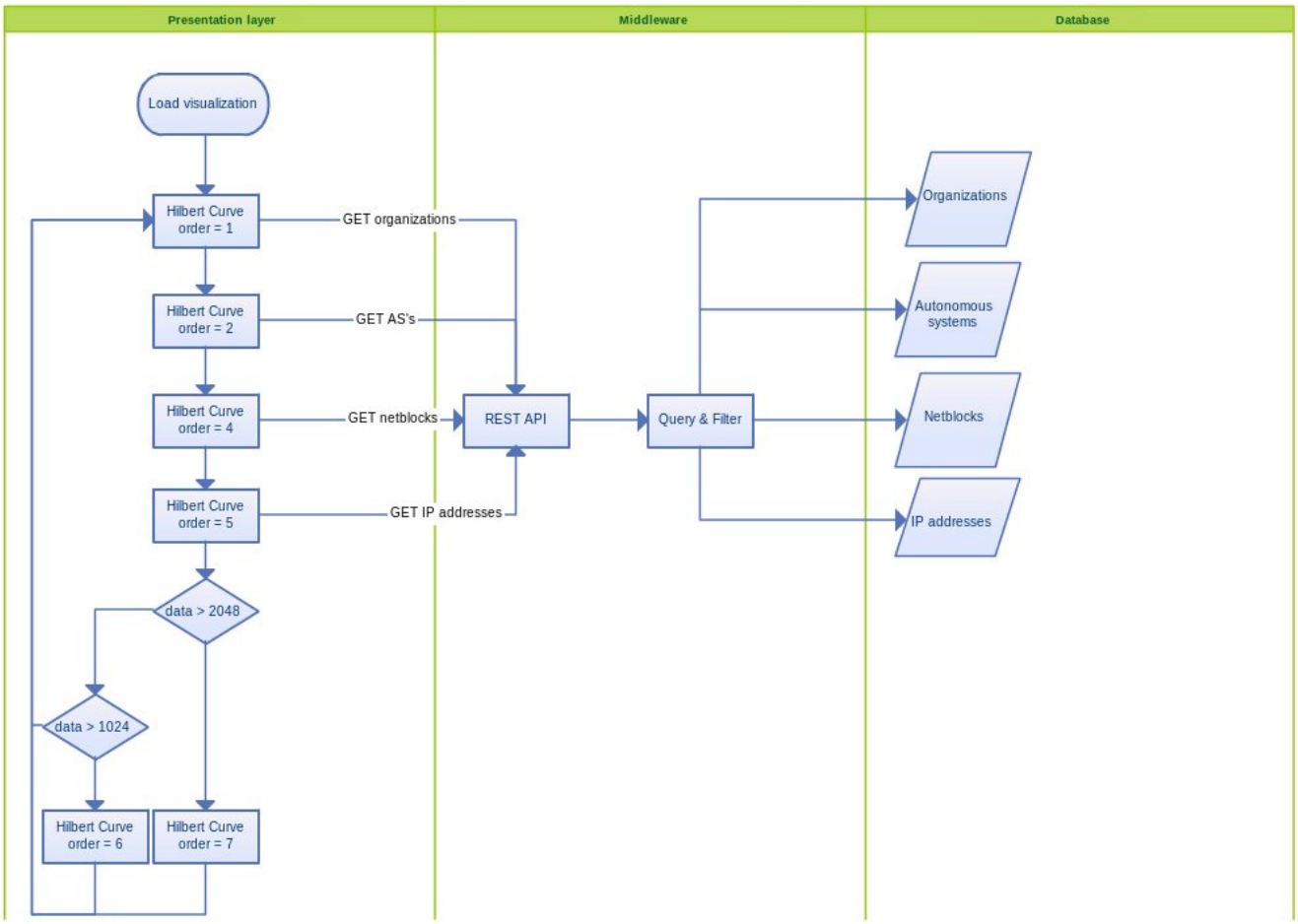


FIG. 8: Mapping a /21 in a 6th order Hilbert curve.

Appendix C: Client side Hilbert curve process



Appendix D: Hilbert curve demonstration

The following images demonstrate the practicality of the interactive Hilbert Curve and its location preserving properties. The images demonstrate the representation of network security hygiene at various abstraction levels and show that a group of IP addresses that are in the same range are drawn near to each other in the Hilbert curve. This allows a user to hypothesize on the function of the systems (a low address range might indicate servers), possible causes and relations between the network security hygiene levels of different systems.

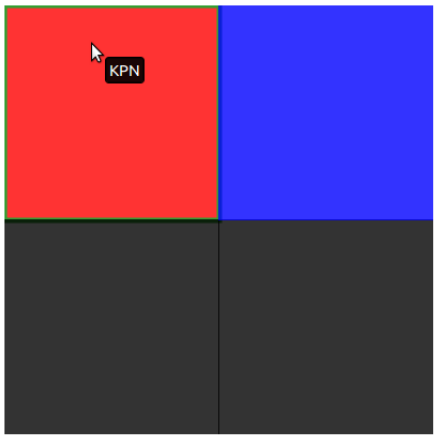


FIG. 9: Demo: organization in trouble.

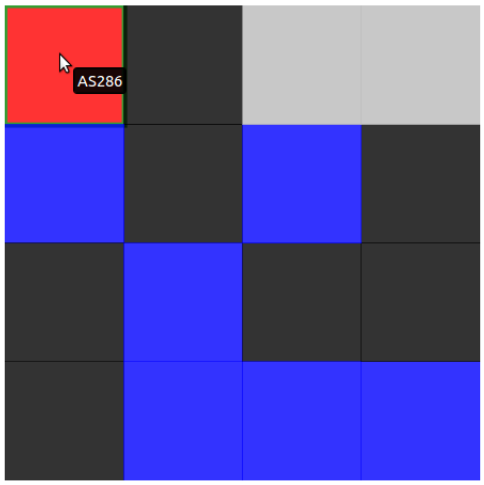


FIG. 10: Demo: zooming in to the troublesome AS.

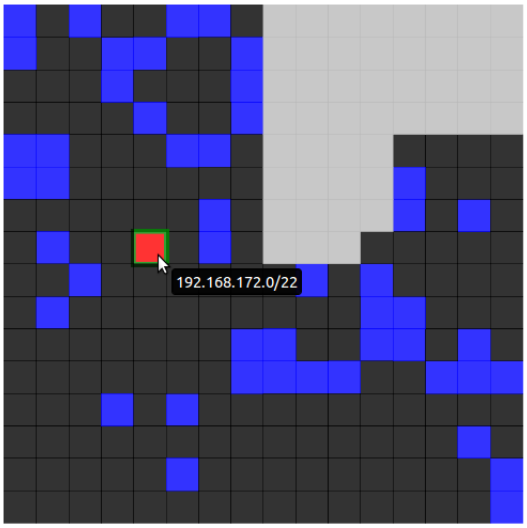


FIG. 11: Demo: zooming in to the troublesome /22 netblock.

The network security hygiene level was generated for demonstration purposes and does not reflect KPN's true network security hygiene.

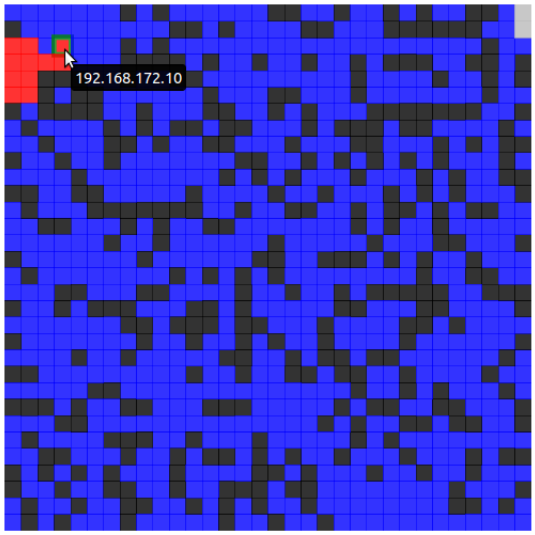


FIG. 12: Demo: troublesome IP 192.168.172.10.

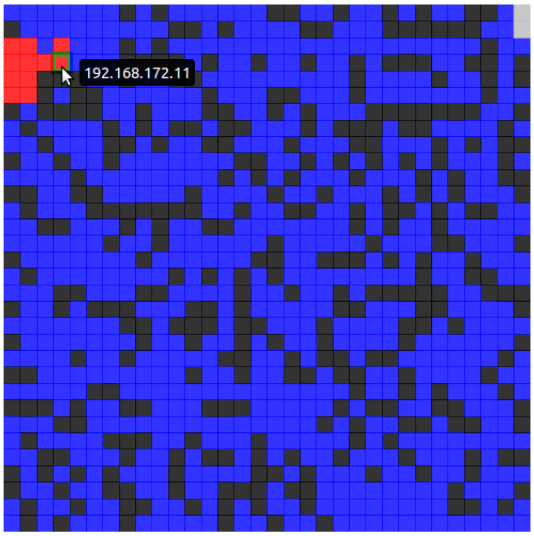


FIG. 13: Demo: troublesome IP 192.168.172.11.

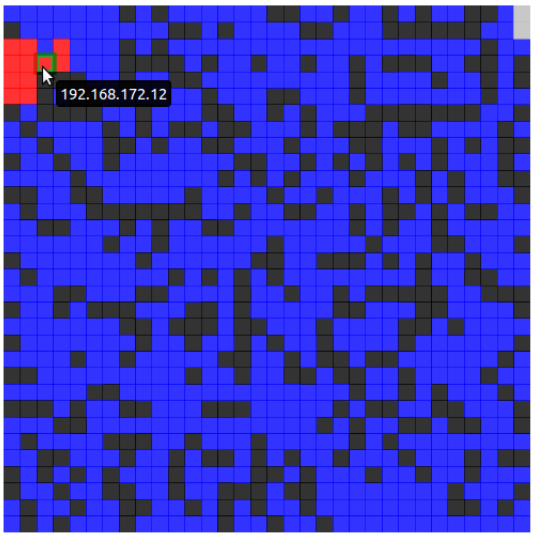


FIG. 14: Demo: troublesome IP 192.168.172.12.

The network security hygiene level was generated for demonstration purposes and does not reflect KPN's true network security hygiene.