

# Visualizing DNS Traffic

Pin Ren  
Northwestern University,  
Department of EECS  
p-ren@cs.northwestern.edu

John Kristoff  
Neustar  
jtk@ultradns.net

Bruce Gooch  
Northwestern University,  
Department of EECS  
bgooch@cs.northwestern.edu

## ABSTRACT

This paper proposes a visualization approach to address Domain Name System (DNS) security challenges, such as distributed denial of service (DDoS) and cache poisoning attacks.

We present *Flying Term*, a new perceptually motivated visual metaphor for visualizing the dynamic nature of DNS queries. The addition of visual metaphors such as *Stacking Graphs*, *Two Tone Pseudo Color*, and *Chernoff Face Glyph* within the same application framework provide enhanced monitoring capability and situational awareness for visualizing DNS queries. We demonstrate our visualization's capability to help administrators identify and understand DNS querying behavior due to anomalies such as misconfiguration and security events with DNS query data acquired from a diverse set of caching servers on the Internet.

## Keywords

Domain Name System, Information Visualization, Network Security Visualization, Visual Metaphor

## 1. INTRODUCTION

The Domain Name System (DNS) is a general, distributed naming service, widely used in TCP/IP networks to refer to resources. All hosts can run a complete resolving name client application, but generally, end systems use a lightweight "stub resolver", which sends requests to local caching name servers to perform the lookups on their behalf. On the Internet today, almost all users and applications make extensive use of DNS through the use of lookups from a client resolver to one or more name servers.

While there have been some studies on DNS traffic over the years, we believe a number of insights into the operation of the Internet can be gained by looking at DNS traffic in non-traditional ways. In this paper we aim to help fill this gap by proposing novel visualization methods to depict DNS

data in order to better identify anomalies, misconfiguration, security events and overall trends.

In the long history of DNS, security concerns have often been central to many proposed or implemented changes in design and implementation both past [20] and present [10, 18]. Weaknesses in the DNS protocols or operation of name servers has given rise to a number of malicious attacks including two recent threats:

- 1. Reflection and Amplification attacks:** By their very nature, name servers tend to be widely accessible so that any client can perform a lookup for data that the server is authoritative for. Currently, there is a large population of open recursive name servers accessible on the public Internet [3]. DNS messages are typically delivered using a single request and response over UDP with no widely deployed authentication mechanism between clients and servers. It can also be shown that a very small DNS request can solicit a disproportionately large response. The capability for many clients on many networks to be able to forge their source address coupled with the deployed DNS attributes previously mentioned, make it possible for someone to launch a devastatingly amplified, reflective attack against an unwitting victim [4, 8].
- 2. Cache poisoning attacks:** Caching name servers, including many open recursive name servers, may be susceptible to cache poisoning. One notorious incident involved a DNS implementation that was too trusting with the data it received in the additional section of a response message, making it easy for a vulnerable server to associate incorrect IP addresses with names [2]. More recently, it has been shown that cache poisoning is a threat shared by nearly all implementations if the attacker has enough patience and some knowledge about queries a caching server makes [21].

The two DNS threats mentioned above are our main motivations for this research. We propose a methodology, which leverages visualization and human visual perception capabilities in the process of identifying, detecting, classifying, and analyzing the abnormal DNS querying behaviors. Our methods provide techniques for visualizing DNS queries and may allow the operations community to identify and solve the corresponding DNS issues.

## 2. PREVIOUS WORK

The DNS operations community strives to address the security challenges. But up until now, there has been little algorithmic detection and reaction solution available. To minimize threats, the common approaches are: a) restrict recursive DNS service to a limited, trusted subset of networks and hosts, b) separate a recursive name service from authoritative name servers, and c) broaden the availability and split the load between multiple name servers using any-cast.

A number of recent tools and systems have been built to help limit the use of names used for malicious purposes, but these are reactionary and often require manual identification before they can be entered into the system [5, 15]. Tools such as *dnstop* [6] simply aggregate general statistics including top domain names queried during a period. Others like *dsc* [7] collect similar statistics and provide time series graphs useful for long term trending of general status and health. However, such tools only provide rudimentary counters of certain fields and variables in DNS data. As such, they lack the ability to convey relationships between fields and cannot easily identify interesting trends that occur outside standard minimum/maximum counters.

DNS query log data supplied by popular DNS implementations, such as ISC BIND [1], contain much of the key information to address many of the security issues we mentioned above. However, manually inspecting these log files can be a tedious and daunting job. Existing log monitoring systems tend to focus on counts for specific fields or watch for a hand-crafted list of known malicious remote hosts. Simple counters are only useful if a security event creates abnormally high counts in one of the monitored values or generates a uncommon log that signals investigation. Since many of the security issues fall under the radar with these simplistic methods, we propose visualization methods that exploit the innate human ability to quickly process visual anomalies or trends.

Although there have been some static visualization effort in displaying DNS data statistics [16], to the best of our knowledge, no effort has been made to interactively visualize DNS traffic.

In the information visualization research community, visual exploratory data analysis is an active research domain. Fitzpatrick et al. [13] developed *BreakingStory* to highlight keywords in text and line graphs to show trends in online news. While these visual metaphors are effective in showing particular data properties, static visualizations may not be well suited for the task of illustrating the dynamic and evolving nature of DNS queries over time.

Albrecht-Buehler et al.'s *TextPool* [9] used motion to visualize trends among text-theme relationships and allowed user interaction of the temporal controls and theme relations. Brandes et al. [11] used animation to illustrate the dynamics of international political and military conflicts. Our work visualizes change via animation, leveraging motion perception to effectively illustrate changing dynamics. Although these projects share our design goal of visualizing the evolving dynamics, our work places more emphasis on identifying

ing correlated querying patterns. Additionally, we address situational awareness by providing effective monitoring capabilities.

Livnat et al. [17] introduced a visual correlation paradigm that employs concentric circles for situational awareness. They advocate the use of spatial reasoning, where radial distances correspond directly to relevance and importance. We address information relevance and importance by leveraging both spatial reasoning and motion perception.

## 3. DATA SOURCE

The DNS query logs from one of the primary name servers in a large US university serve as our example data source. The DNS query dataset we use averages 12 million queries daily. Each log message contains a time stamp, the query name to be resolved, the querier's source IP address, the source port number used by the querier, the query class, and the query type. For example:

```
Jun 30 00:00:00 servername named[14235]:  
client 167.156.183.123:32768:  
query: www.google.com IN A6
```

Query logs can provide key insights in understanding DNS traffic, especially for detecting anomalies. Repeated and/or high volume queries from a single source IP address may be a sign of a source spoofed, reflective distributed denial of service attack. If one query name is suddenly being requested repeatedly within a short time window, it might suggest a distributed denial of service attack against the service the name resolves to. We would expect a slow rate of change for queries of popular websites such as "google.com" for example. Any sudden shift in query patterns should draw our attention to look into the reasons behind the changes.

We store those logs into a MySQL database. In order to make interactive data retrieval possible, we build several database indexes on the data fields that require frequent querying and aggregation such as query string, source ip address, and source port. In addition to using database indices, we also leverage data preprocessing to generate information such as top aggregated queries list sorted by occurrence frequency for each hour and store them into a data table for fast access.

The DNS server administrator who provided our data has compiled a list of suspicious query names to monitor. However, generating such a list requires information exchange with peer administrators and is not automatic. We tried to visualize the queries on this list for evaluation purpose only.

## 4. VISUALIZATION DESIGN

In this section we introduce the visualization design for the DNS visualization tool we built. There are several design considerations for DNS query data visualization. Based on our design goal, we want the visualization tool to:

1. **Aid the administrator in understanding overall patterns and trends.**
2. **Provide situational awareness by assisting the monitoring of data change over time.**

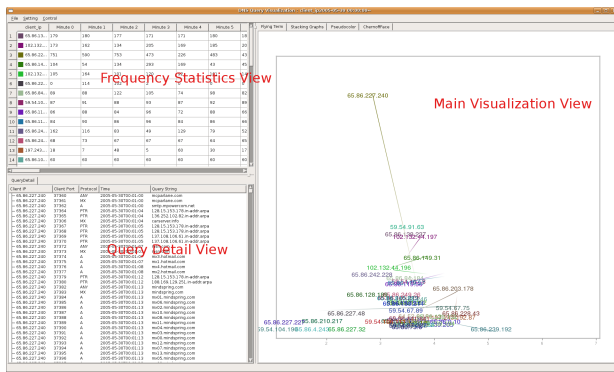


Figure 1: The user interface of our DNS visualization tool. We divide the screen space into three different panels: the main visualization window contains four different visualizations organized for tabbed browsing, the statistics panel displays the aggregated frequency data generated from database, and the query detail panel displays the detailed information for selected queries. In this figure, we visualize the source IP addresses and their aggregated querying frequency around the midnight of May 30, 2005. The same data also drives the visualizations seen in Figures 4, 5, and 6

3. Allow the user to filter and select subsets of information to perform deep analysis.
4. Allow the user to inspect information at the log level with key information highlighted.
5. Allow the user to employ different visualization metaphors to analyze different aspects of the same dataset

## 4.1 Visual Metaphor

*Visual Metaphor* can be defined as the representation of a new system by means of visual attributes corresponding to a different system that is familiar to the user and behaves in a similar way. A visual metaphor should facilitate discovery by presenting data in an intuitive manner, consistent with the user’s perceptual and cognitive abilities [14]. In this project, we focus on the discovery of DNS query patterns and their dynamic evolution over time. We designed a perceptually motivated visual metaphor, *Flying Term*, for assisting such tasks.

### 4.1.1 Flying Term

We built the *Flying Term* visual metaphor following the guidelines proposed by Albrecht-Buehler et al. [9] in order to leverage human’s spatial reasoning and motion perception capabilities for visualizing changing information landscapes.

*Flying Term* is a new visual metaphor that uses motion to visualize frequency of occurrence within a moving time window. The basic idea of *Flying Term* is to visualize target objects (in this DNS application, it can be a query string, source IP, port, and query type) inside a rectangular area, where each object’s spatial location is mapped using the frequency data generated from the aggregation counter and the

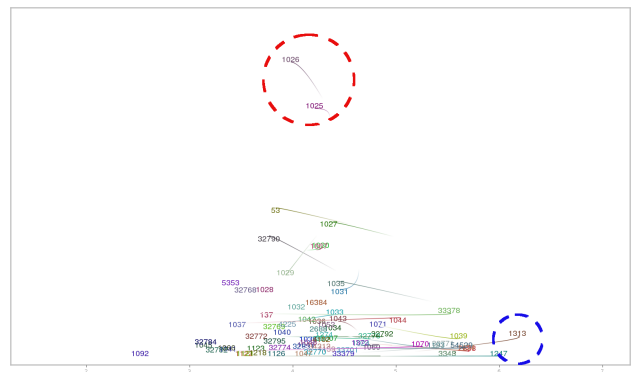


Figure 2: The *Flying Term* metaphor employs motion to visualize the dynamic relationships of DNS queries, using frequency to regulate an object’s vertical position and frequency distribution over time to decide the horizontal position. Animated tail curves highlight the important (selected) queries, and indicate change as well as history. In this figure, our visualization objects are those source port numbers and their frequency data. we can notice that port number 1025 and 1026 are the ports with highest aggregated frequency of occurrence (highlighted using red circle). Indicated by the direction of curly tail, user can notice that the frequency of port 1026 is still moving upwards. Besides those hot ports, moving dynamics of other ports can also be easily observed. For example, near the bottom of the screen, port 1313 is moving towards right (highlighted with blue circle), which means there are (many) queries issued using port 1313 in the most recent time bin of this time window

distribution of such frequency within the current time window. The visual presentation can be the textual name of the object or a small glyph. By using a moving time window and updating the spatial mapping of the objects based on changing of data, we can animate the objects and visualize the dynamic nature of the DNS traffic, which provides a detailed and clear view of the dynamic evolution over time. The limit of how many objects can appear at the same time is an adjustable parameter, we use 30 to 100 for most visualizations, and the default selection criterion is to show objects with higher frequencies within current time window. For example, with a setting of 50 visible objects, by default we will show objects with top 50 frequency values in the current time window. The minimal updating interval for our moving window by default is one minute and the commonly used length for the moving time window is 7 to 10 minutes.

The vertical axis of *Flying Term* represents the accumulated frequency. For any given time window, we calculate the maximum and minimum frequency of all the objects being visualized, map the maximum value to the top of the rectangle and the minimum value to the bottom, and then we calculate each object’s vertical coordinate normalization using linear mapping which linearly map the frequency value range to the vertical screen space range.

The horizontal axis of *Flying Term* represents the frequency

distribution of the visualized subject over time. The horizontal X coordinate of  $Obj_i$  in the time window starting from  $t_s$  and ends at  $t_e$  can be calculated using the following formula:

$$X_{Obj_i} = Width_{window} \cdot \frac{\sum_{t=t_s}^{t_e} Freq(Obj_i, t) \cdot (t - t_s)}{(t_e - t_s) \cdot \sum_{t=t_s}^{t_e} Freq(Obj_i, t)}$$

where function  $Freq(Obj_i, t)$  returns the frequency count of the given  $Obj_i$  and the time bin  $t$ , and  $Width_{window}$  is the width of the current viewing window.

The horizontal coordinate indicates how recent the queries are within the current time window. Query objects from the newly arrived query records appear from the right-most side of the screen. If there is no follow-on occurrence of this query as time goes by, it will gradually age and move to the left side of the screen, eventually moving out of the screen. Objects with constant frequency of occurrence remain in the middle of the screen. The immediate benefit of such mapping is that both the occurring frequency and the occurrence distribution over time are represented intuitively. To answer many analytical questions, users want to keep track of both properties at the same time, and they are generally more interested in high frequency count and more recent occurrences. The right-top region has higher importance and priority than the left-bottom, facilitating fast and efficient visual search.

As the time window advances, we update the screen location of each visible query object. We interpolate the intermediate location between keyframe locations calculated from the data to produce smooth animations.

To highlight the movement of objects, we implemented stylized curly tails for moving objects using quadratic bezier curves. Each tail of a moving object is formed by the path defined by the function  $B(t)$ :

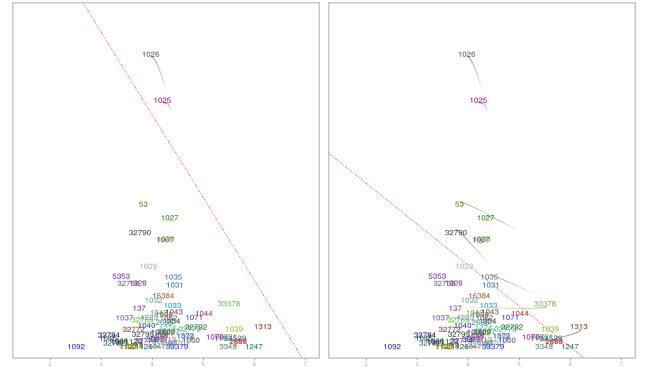
$$B(t) = (1 - t)^2 \cdot P_0 + 2t(1 - t) \cdot P_1 + t^2 \cdot P_2$$

where  $t$  is the parameter varying from 0 to 1,  $P_0$  is the current location of the object, while  $P_2$  represents the previous position of the object at exactly one unit time interval before. The control point  $P_1$  in the middle is the object position at the keyframe in-between <sup>1</sup>.

The animated tail curve serves two purposes: It highlights the objects visually, and makes their moving path apparent. Since the length of the moving tail is mapped to the speed of the movement, the user can thus acquire a sense of rate of change in the frequency of occurrence. By showing object movement and the animated tail at the same time, our system visually assists the comparison of evolving patterns of different objects. Correlated moving patterns can also be easily spotted. Also when used as an ambient display, the animated *Flying Term* with curly tail highlighted can be well suited for monitoring tasks.

We have implemented a space divider widget (Figure 3) to assist with selecting and highlighting objects which are more

<sup>1</sup>Since we update the moving window discretely and interpolate the position trace in between, the keyframe is the frame where the object position is actually calculated by just updated data input instead of interpolation.



**Figure 3: Using space divider widget to select and highlight important queries. The space divider widget is the red dashed line when visually activated. In the left figure, only the top two ports are selected and highlighted with tails. In the right figure, by repositioning the space divider, the user can select and highlight more objects according to the time and value range he specified using the divider. The divider is also a method to control visual complexity.**

recent and/or occurring with higher frequency. When visually activated with mouse button down, the divider widget is a dashed line to divide the rectangular area. Using the mouse, a user can easily rotate and translate this line within the window to interactively select and highlight the objects appearing at the top/right side of the line, and by default only those highlighted objects will have curly tails attached when moving.

#### 4.1.2 Additional Visual Presentations

Although our *Flying Term* visualization facilitates the depiction and monitoring of dynamic information changes, additional data analysis is necessary. To make the visualization system more useful and to provide support for additional visual analytics, we implemented three existing visual presentations within the same visualization framework: Stacking graphs (Figure 4), Two-Tone Pseudo Color visualization (Figure 5) and Chernoff Face glyphs (Figure 6).

Stacking Graphs has already had many successful information visualization applications, such as the *Baby Name* visualization by Wattenberg [22]. Stacking Graphs clearly display broad data trends. By stacking time series data, the frequency stream over time in our DNS application, users can easily observe those evolving pattern of data change and compare the pairwise relationships of selected data streams by filtering out other streams.

Two-Tone Pseudo Color visualization, first introduced by Saito et al. [19], efficiently uses small display space for accurately presenting data values. A quick glance at this visualization can reveal global trends and the big picture, ideal for ambient display monitoring. We map the time series data of frequency to drive the visualization.

Chernoff Face glyphs [12] visualize multivariate data by displaying  $n$  variables on a two-dimensional face. In our implementation, we used the following ten facial features: head

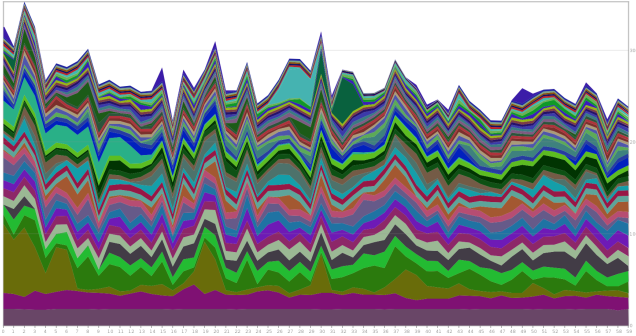


Figure 4: Stacking Graph visualization for the source IP address data used in Figure 1. The color of each stream band is consistent with the originally randomly assigned color in the Flying Term view and Statistic Panel in Figure 1, and the graphs are stacking from bottom up. The global trends of data streams are easy to observe. Pairwise stream comparison can be quickly done by filtering out other streams as seen in Figure 7

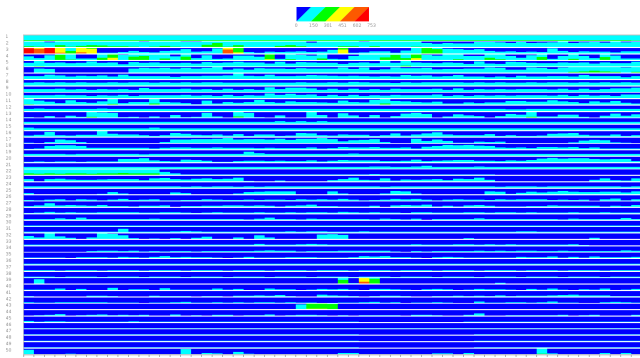


Figure 5: Two Tone Pseudo Color view for visualizing the data value change. Note the value can be read out fairly accurately even with very small display estate, and the value outliers can be easily noticed. For example, there is a high volume of query occurrences in the first three minutes for the third source IP address. The highest frequency in day 1 and day 3 is around 750 just by looking at the visualization. (In the value-color meter above, full red color corresponds to 753). This can be confirmed by looking at top-right corner of the statistics panel in Figure 1, the actual values for day 1 and day 3 are 751 and 753 respectively. Since the color is used to present value, we label each object using the index number to the left of the visualization in order to establish correspondences.



Figure 6: Chernoff Face View for monitoring. Note that the two faces in the first column are reference faces for showing two extremes, The first happy face is the reference for the all zero data input, and the second angry and unhappy face is for an imaginary input data with highest values input for each time bin within the current time window. The color of each face is consistent with previous views. As we discovered in Figure 5 the high value data outlier is the third object. In this visualization, we can also pick out this outlier easily, the third face in the second column is indeed an outlier visually.

eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening. We chose Chernoff Face glyphs mainly for its exceptional capability for enabling passive monitoring due to the human visual systems sensitivity to the changes in a face, especially when the face is associated with emotion. When a face is drastically different from a previous faces, it indicates that something abnormal may be happening and thus requires more attention. In our visualization, we use each query object's time series data of frequency as the multivariate input for the face feature generation. The angry/unhappy faces are the easiest to notice and thus we map angry and unhappy facial features to high frequency values. High frequency values correspond to the most suspicious values in many of our security detection tasks.

## 4.2 User Interface and Interaction

The user interface of our tool is divided into three panels: the main visualization window, the statistics panel, and the query detail view panel. The time window is common to all visualization modules and changes in data selection or time advancement in one visualization module will be reflected in the others (Figure 1). Our visualization has standard playback controls built-in, such as stop, pause, rewind and step-forward. The space divide widget can be used to select regions and highlight the keywords or terms in that region. In the main visualization window, a standard tabbed browsing technique is employed for navigation, and we use consistent color for the same keyword across visualization modules. In the frequency statistics panel, we display the numerical value of the frequency within the time window. The data statistics panel also serves as the object selector for the main visualization window. In the query detail panel, we list all the relevant queries to the current visualization. This list can be filtered by selected query name, source IP address, port number, query type and/or by the time window range.

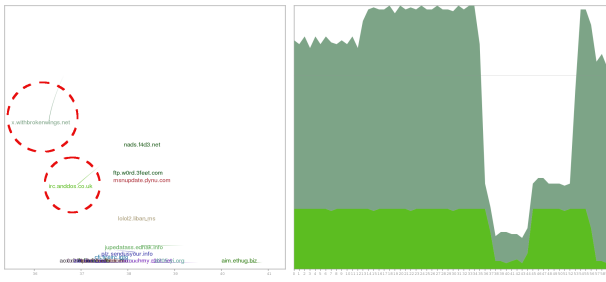


Figure 7: Visualizing the queries for "bad names" on a blacklist. Notice that two host name strings are moving together in the visualization to the left (highlighted using red circles). After filtering out other query names, the user can conduct pairwise comparison of the two names in question in the Stacking Graph visualization to the right. Their data patterns were similar in this hour, and for some unknown reason, they both had a sudden drop of activity in the middle and recovered later. Further investigation using query detail view proved that those two "bad" host names did share one common querying source IP address. The names may be associated with the same malicious server or similar malware, which results in congruent query activity patterns.

Visualization objects such as textual names in *Flying Term* are clickable. The default action is to select the clicked object and query database for detailed records filtering by the selected name and then update the query record view <sup>2</sup>.

## 5. CASE STUDIES

In this section, we are going to demonstrate the utilities of our visualizations using two case studies.

### 5.1 "Bad Names" Querying Behaviors

As we mentioned in Section 3, administrators may have a hand-crafted blacklist recording potential malicious servers or "bad names". Often those names are associated with remote servers that control global botnets. The infected hosts need to contact those servers from time to time, and one way to find the infected bots is by monitoring DNS queries.

In Figure 7 the user found an interesting correlated querying pattern of two "bad" query names by monitoring the *Flying Term* visualization. With the interaction provided with our system, the user conducted pairwise comparison on their data patterns in the Stacking Graph View to the right of Figure 7. The comparison confirmed the similarities. The user was also interested in what hosts were querying them at that time. By clicking on the query name in visualization, the user selected and filtered out irrelevant querying details. As a result, a common infected host that kept querying both servers was identified. Our visualization helped the user to form the hypothesis that those two servers are related malicious servers.

<sup>2</sup>Visit our project website [http://www.cs.northwestern.edu/~pren/dns\\_vis](http://www.cs.northwestern.edu/~pren/dns_vis) for more information, screen shots and video clips demonstrating our interactive system.

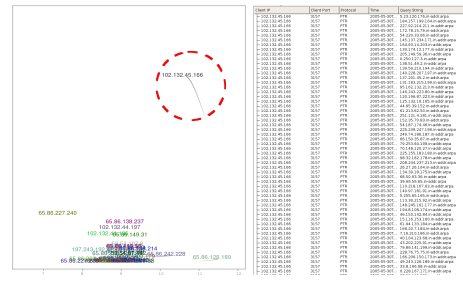


Figure 8: Visualizing the source IP addresses and their querying frequency. By monitoring the *Flying Term* visualization to the left, the user noticed the high-speed sudden rise of one IP address (highlighted using red circle). In order to investigate, the user clicked this IP address's name in *Flying Term* visualization and the detail of its querying behaviors in this time window were revealed. They were all PTR type queries (reverse lookup), a well-known brute force password attack to an SSH server.

### 5.2 Brute Force Attack on SSH Server

In Figure 8, we visualized a source IP address' querying behaviors. Via monitoring the *Flying Term* visualization, the user found a suspicious source IP address with a sudden rise of querying frequency, in just couple of minutes because it quickly rose to the top of the visualization. Note that the long trail indicates its fast rising speed. When the user looked at the statistical view, he found this IP address suddenly issued thousands of DNS queries in a 3 to 4 minute time window. It was quiet before the time window in question, and stayed silent after the burst of queries. The user then click and show detailed query records made by this source IP address in the query detail view in the right half of Figure 8. All the queries are of the type PTR, which falls into a well-known scenario that this IP address (to be specific, the SSH server) was bearing a brute force password guessing attack. As each SSH connection is made, the SSH server attempts to resolve the source IP address in order to further verify that the name maps back to the source IP address.

## 6. DISCUSSIONS AND FUTURE WORK

In all the visualizations, we use the time series data of frequency as input data, varying the length of the moving time window. In *Flying Term*, the length is flexible and usually varies from five to ten. For Chernoff Face glyphs, the time window defaults to ten, and for Stacking Graphs and Two Tone Pseudo Color visualization, the time window is set to 60 (one hour). As a result, the static snapshots of the four visualizations show data features over different time ranges. However, all of these visualizations can be updated at every time interval to actively reflect the newest changes, and thus are well suited for an ambient display that monitors real-time DNS traffic.

In many cases, we found that using DNS query log visualization alone does not provide conclusive evidence to find out the real reason behind those visual anomalies. Those perceived anomalies might be security related attacks or just a misconfiguration. Visualizing DNS query log data alone

may not be sufficient to answer all the questions. There are additional DNS fields and characteristics about the underlying protocols that may be useful. Furthermore, the server response may include invaluable information such as response codes, TTLs and RR data. However, due to practical limitations and in some cases sensitivity of the data, such information is not always readily available. We would like to further investigate integrating the visualizations we have now as components into existing DNS analysis tools to take better advantage of available information and analytical functionalities in those existing tools.

We found that it is helpful to integrate different visualization techniques together within the same application framework, so that user can easily see a different views of their data. With simple interaction, techniques such as filtering, linking and brushing, users can benefit from using multiple visual presentations as a package, and greatly augment their analytical abilities. Another benefit for such integration is in the development process, by taking advantage of existing functionalities such as database module, data processing module, and following the rule of "separating data and view", the effort to implement and integrate new visual presentation can be very small. We would like to investigate other useful visual presentations and to construct a suite of visualization software, which can be used in solving a broader range of problems.

## 7. CONCLUSIONS

In this paper, we introduced the DNS threats and challenges which have been largely overlooked in the VizSec community. We propose a novel, perceptually motivated visual metaphor *Flying Term*, together with the implementation of three existing visual presentations. We demonstrated that our visualizations are effective in revealing many important aspects of DNS traffic. Our preliminary experimental results indicate that our visualization attempt is not a complete solution to those challenging security problems that motivated us. However, we believe that our efforts are aimed in the right direction and improve current monitoring and identification techniques. Integrating effective visualization methods as components into existing security tools is a promising way to address DNS management and security challenges.

## 8. REFERENCES

- [1] Internet systems consortium bind. <http://www.isc.org>.
- [2] Cert advisory ca-1997-22 bind- the berkeley internet name daemon, 1997. <http://www.cert.org/advisories/CA-1997-22.html>.
- [3] The measurement factory:dns survey, May 2005. <http://dns.measurement-factory.com/surveys/sum1.html>.
- [4] Ana spoofer project, 2006. <http://spoofer.csail.mit.edu/>.
- [5] Dns providers blacklist, 2006. <http://www.dnsbl.org/>.
- [6] The measurement factory: dnstop tool, 2006. <http://dns.measurement-factory.com/tools/dnstop/>.
- [7] The measurement factory: dsc-a dns statistics collector, 2006. <http://dns.measurement-factory.com/tools/dsc/>.
- [8] Report from the icann security and stability advisory committee, Mar. 2006. <http://www.icann.org/committees/security/dns-ddos-advisory-31mar06.pdf>.
- [9] C. Albrecht-Buehler, B. Watson, and D. A. Shamma. Visualizing live text streams using motion and temporal pooling. *IEEE Comput. Graph. Appl.*, 25(3):52–59, 2005.
- [10] S. M. Bellovin. A look back at "security problems in the tcp/ip protocol suite". In *ACSAC*, pages 229–249, 2004.
- [11] U. Brandes, D. Fleischer, and J. Lerner. Highlighting conflict dynamics in event data. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 103–110, 2005.
- [12] H. Chernoff. Using faces to represent points in k-dimensional space. *Journal of the American Statistical Association*, 68:361–368, 1973.
- [13] J. A. Fitzpatrick, J. Reffell, and M. Aydelott. Breakingstory: visualizing change in online news. In *CHI Extended Abstracts*, pages 900–901, 2003.
- [14] S. Havre, E. G. Hetzler, P. Whitney, and L. T. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. Vis. Comput. Graph.*, 8(1):9–20, 2002.
- [15] J. Kristoff. An automated incident response system using bind query logs, 2006. <http://public.oarci.net/files/jtk-dnsbotmon.pdf>.
- [16] J. Kristoff. A brief look at some dns query data, 2006. <http://www.nanog.org/mtg-0602/lightning.html>.
- [17] Y. Livnat, J. Agutter, S. Moon, and S. Foresti. Visual correlation for situational awareness. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 95–102, 2005.
- [18] V. Ramasubramanian and E. G. Sirer. Perils of transitive trust in the domain name system. In *Proceedings of the Internet Measurement Conference (IMC), Berkeley, California*, October 2005.
- [19] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 23, 2005.
- [20] C. Schuba. Addressing weakness in the domain name system protocol, 1993. Master Thesis, Purdue University.
- [21] J. Stewart. Dns cache poisoning - the next generation. <http://www.lurhq.com/dnscache.pdf>.
- [22] M. Wattenberg. Baby names, visualization, and social data analysis. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 1, 2005.