NetVis: A network traffic visualization tool

SUBMISSION ID
AFFILIATION ID

May 22, 2013

Abstract

Computer network traffic visualizations attempts to deliver improved understanding of traffic on a network to an observer. Many existing tools opt for graph or plot-based visualizations to detect patterns or outliers in the data, but still largely provide a segmented view of any data feed. In this paper, we present a novel network traffic visualization framework that makes use of a variety of complementary visualizations to obtain better situational awareness. Our proposed solution is to look at different..

1 Introduction

Analysing and monitoring network traffic is essential for ensuring network security. Since the amounts of data involved in this task are extraordinarily large, it can be hard to make effective use of them. Visualizations offer a solution: They give a compact representation of data to a human analyst who can use them to detect patterns in the network. An effective visualization of network traffic will make it easy to identify outliers while making clear the general patterns of network usage, and it will facilitate detection of intrusions and of malicious activity.

We present here an application which provides a framework for analysing network activity in an interactive and hollistic manner. Time series data in the form of packet captures are processed in real-time, and simultaneously rendered in multiple interconnected visualizations. The user can switch between the available representations of the data and dynamically influence both the visualization's layout and the amount and structure of the data displayed. The software is designed to make it easy to spot irregular activity and then to investigate it from multiple angles.

The framework developed here aims to provide an environment in which situational awareness can more effectively be obtained. It is therefore built to be easily extendable to fit the demands of the specific situation. New visualizations will in a natural way complement the existing ones: The underlying data processing engine provides a standard, or *normalised*, data basis which is shared by all displays.

The purpose of the tools presented in this paper is to help a human observer to make sense of what is happening in a given network. They emphasize exceptions, show comparisons, and try to answer a wide array of question in a concise and simple fashion. We want the user to generate good hypotheses in response to the visualized information. To achieve this, many aids are given to the user in order to avoid cluttering displays with irrelevant data. In particular, we have implemented a powerful filtering system which can be dynamically adjusted in response to changing situations.

The application's main purpose is to monitor the current activity in the network. However, the same architecture can be used to forensically analyse past time-series data. The system will simulate the data records as if they represented a live network. This exploits the important dimension of time, and makes it easier to understand what is actually happening.

2 Related Work

JH

3 NetVis Architecture

Describe the architecture, how traffic is read and processed, the format of the CSV, why was it designed the way it is? Add diagram How real-time is it? What filtering do you support + protocols? How modular is the code base, can one simply script in another vis?

Normalisers: A normaliser is a class responsible with taking a packet and returning a normalised value of a certain attribute of that packet and the other way around. The Source IP normaliser provides a method that takes a packet and returns a value between 0 and 1 corresponding to its Source IP (0.0.0.0 returns 0, 255.255.255.255 returns 1). It is also capable of taking a value from 0 to 1 and returning a human readable IP address. Each normalising class is able to create a temporary filter in the application that will filter its corresponding attribute on a certain range. Since the normalising class is in control of its filter it can create a zooming effect based on the value of the filter. (If there is a filter on the Source Port normaliser from 0,5 to 1 then port 30000 will be normalised to 0 instead of 0,5.) // You can extend this + make it pretty

Filtering: The application supports two types of filters - filters that the user explicitly defines, and filters applied on-the-fly from within the visualisations. Both types are applied to all visualisations and information displayed, and can be adjusted at any time without losing prior data. From the right panel, the user has access to a variety of different filter controls. First, there is a menu for filtering by transit protocol. By default, all protocols are selected and therefore included. Protolcols are sorted into menus by protocol family, appearing in multiple places where appropriate. Second, there is a control to select the range of ports the application uses, which defaults to the maximum

port range. The source and destination ports can be set separately, enabling a user to view all data entering/exiting a port as desired. Next, there are IP and MAC address filters. These work on a blacklist/whitelist system, allowing a user to only view packets to or from a particular set of addresses, or ignore packets going to or from a different set. This enables a user to, for instance, ignore traffic from sources they know are irrelevant or focus only on an address that is causing concern. The second type of filter will be dealt with in more detail in sections 4.2 and 4.3.

Modularity: The application was designed to be extremely extensible. Adding a new visualisation is simply a matter of adding it to the application's list of those available, which will cause it to be included and kept up-to-date as packets come in and are filtered. The same is true for filters and normalisers. Adding a filter would result in its controls automatically being included in the right panel and all packets would then be filtered according to the criteria it specifies. Adding a normaliser would cause it to be integrated with the Spinning Cube, Dataflow and Attribute Distribution visualisations. Ths modularity extends even so far as data input. If a class were written to accept packets from a different source, it would be trivial to switch the application to use this class.

4 NetVis Visualizations

4.1 GUI

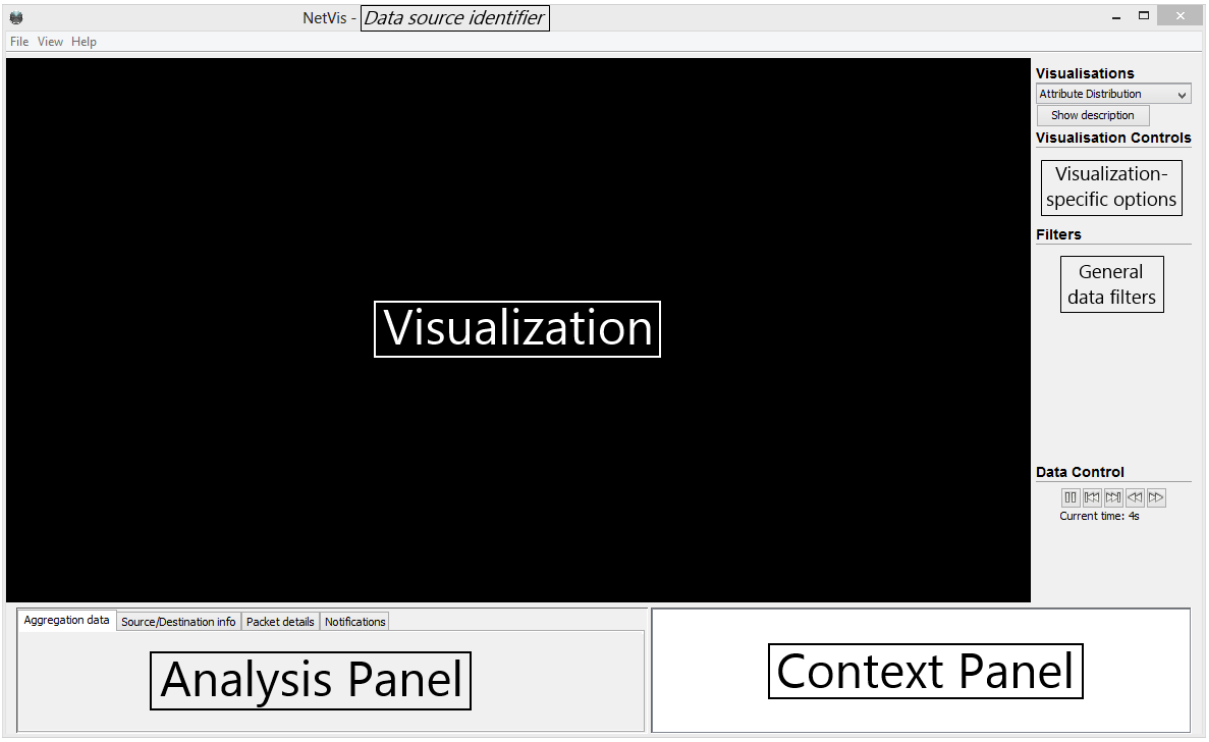
The user interface of the application focusses around the displayed visualization, which takes up the majority of the window, and can be maximised to fill the available space. Other panels occupy the right and bottom sections of the window, and are as follows.

Choice of visualization is presented to the user by a list in the top of the right panel, followed underneath by options specific to that visualisation, and then by general filters to refine the processed data. Lastly, time controls are included on the right panel to allow the user to fine-tune the speed at which data is processed and displayed. The right panel is ultimately aimed at providing the user with options to adapt the shown data.

The function of the bottom panel is to show fine and aggregate details of the processed data. To fit the large amount of data available into such a limited space, a tabbed pane is used on the left to categorise distinct types of data. The right section of this panel contains a 'Context Pane', the purpose of which is to show even further detail of selected data on demand. The left 'Analysis' panel and the right 'Context' panel are split in two by a split pane, so the user can modify the proportion of each they wish to see.

Less-frequently used program functions, as well as window tools, are included in the form of a main menu bar. Common window tasks are available via the menu bar, as well as an exclusive-mode full screen option and the facility to select a new data source.

Messages to the user come in the form of warning dialogues for user or program errors, notifications in the bottom 'Analysis' panel for program notifications, and text messages in the bottom 'Context' panel for help messages and suggestions. A diagram of the application layout is included below.



4.2 Attribute Distribution

It displays a graph of the distribution of some packet attribute (eg. Distribution of port ranges across all packets). The main graph is on a log scale useful for identifying spikes in the volume of data on some attribute interval. Each interval also displays some dots. The area of those dots is an actual representation of the traffic that has that certain attribute property (The dots are also color coded - a red dot is bigger than a blue). In this visualisation you can select a range for an attribute and the visualisation will apply a removable data filter that will only allow data in that range to get to the visualisations.

4.3 Dataflow

Based on the Network Security Visualisation paper by Keith Fligg and Genevieve Max shows the flow of each packet in a parallel coordinate plot. It provides a good view of the whole traffic in the network and the distribution in the attributes of packets. Lines are coloured based on their value in some coordinate and fade out based on how old the packet theyre representing is. Newer packets also have a random offset making it easier to spot if a colorful line represents one or more packets. Adding randomness into the values of the attributes makes it easier to distinguish between concentrated groups of packets with the same characteristics and single packets. Example: If lots of requests to a server come from a single MAC address through the same ports and using the same protocol then they would all seem like one request. Adding randomness makes the pack

of requests more visible without significant impact on the accuracy of what the analyst is viewing.

4.4 Spinning Cube

The spinning cube is an implementation of an existing visualization tool known as the spinning cube of potential doom[1]. Its displays packets as dots inside a spinning cube. Their 3D position is based on 3 packet attributes. (eg. x-axis ⇒ Source Port / y-axis ⇒ Destination Port / z-axis ⇒ Source IP). The 3 coordinates are customisable.

4.5 Traffic Volume

The traffic volume visualization is inspired by the 'FlowScan' graph[2]. It's realised as a stacked bar chart which displays the volume of data arriving in each time interval, with column segment heights proportional to the total number of packets transmitted for each protocol. This allows users to see at a glance if a particular protocol is being exploited on the network. In addition, the column segments are colour-coded and can be cross-referenced with a protocol key underneath the visualization itself.

To help distinguish different protocols, colours are selected from a colour palette which provides colours from a qualitative colour scheme. This visualization provides a control in the right panel which lets the user adjust the number of time intervals displayed at once.

4.6 Heat Map

4.7 Activity Groups

5 Discussion

Each visualisation was chosen to fill a specfic purpose. The Heat Map visualisation fills the need to gain a very quick impression of overall network load and size. The Activity Groups visualisation provides the same information on a server-by-sever basis, providing more data at the expense of simplicity. The Dataflow visualisation provides the ability to see what packets currently coming over the network 'look' like, and the Spinning Cube and Attribute Distribution visualisations allow the user specify which apects of the packets are shown.

As such, no two visualisations fill the same role, each providing something that is not found in any of the others. However, to get a complete understanding of the network no

[1] See `www.kismetwireless.net/doomcube/`

[2] See `www.caida.org/tools/utilities/flowscan/`

one visualisation will suffice - it is the interation of these complementary visualisations that makes the application unique.

5.1 User Workflow

Describe a typical workflow of an analyst, include a workflow diagram (e.g. UML or CONOPS (concept of operations, see))

Describe how the visualisations complement each other

Describe how many alerts it can process at any time and accumulative.

5.2 Advantages

The complementary nature of the visualisations means that vitually all information a user could want about the network in available somewhere. It also makes it very easy to spot a point of interest in one visualisation, then follow it through the others to see more information.

The dynamic filtering makes focusing on a point of interest extremely easy, allowing a user to remove any unnecessary or distracting data. As the filtering is applied retroactively, it allows the user to see past events from different perspectives in order to determine the cause.

As the application works in real-time, the user can see the current state of the network. Hence, they could see a developing network attack and take preventative measures, e.g. blocking the IP of a client attempting a brute-force SSH attack.

Can be used as a learning tool

5.3 Limitations

Doesn't use all data

5.4 Future Work

More advanced data processing Machine Learning Save current configuration and have access to sensible filter packages

6 Conclusion

In this paper we have presented..