NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness

Kiran Lakkaraju
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
605 E. Springfield Ave.
Champaign, IL 61821
kiran@ncsa.uiuc.edu

William Yurcik
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
605 E. Springfield Ave.
Champaign, IL 61821
byurcik@ncsa.uiuc.edu

Adam J. Lee
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
605 E. Springfield Ave.
Champaign, IL 61821
adamlee@ncsa.uiuc.edu

ABSTRACT

The number of attacks against large computer systems is currently growing at a rapid pace. Despite the best efforts of security analysts, large organizations are having trouble keeping on top of the current state of their networks. In this paper, we describe a tool called NVisionIP that is designed to increase the security analyst's situational awareness. As humans are inherently visual beings, NVisionIP uses a graphical representation of a class-B network to allow analysts to quickly visualize the current state of their network. We present an overview of NVisionIP along with a discussion of various types of security-related scenarios that it can be used to detect.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*invasive software*

General Terms

Security, Management, Human Factors

Keywords

NetFlows, security system state, security visualization, situational awareness

1. INTRODUCTION

In recent years, the number of security incidents reported per annum has increased at an exponential rate [3]. Each of these incidents may involve multiple sites and within each site, it is highly likely that a large number of machines may fall under attack. The job of the security analyst is complicated at best, involving the use of multiple tools to attempt to gain an understanding of the current state of the network. As the number of security incidents continues to increase, this task will become ever more insurmountable, requiring extreme vigilance on the part of security personnel.

Perhaps the main reason that the task of network security monitoring is so difficult is the lack of tools to provide a sense of network situational awareness. The Department of Homeland Security defines situational awareness as "the ability to identify, process, and comprehend the critical elements of information about what is happening to the team with regards to the mission [28]." While the notion of situational awareness has been around for some time in military combat scenarios, it is a relatively new development in the field of computer security.

The best means of presenting a security analyst with enough data to form a cohesive picture of what is happening in their network is through the use of visualization. Humans are by nature visual beings and are capable of processing large amounts of data through maps and data plots. In fact, there is no more powerful method of presenting large amounts of information than through visual data maps [27].

While people have mapped the structure and content of the Internet since its inception [4], applying these maps to security analysis is a relatively new endeavor. The tool that we discuss is called NVisionIP [12, 31, 13]. NVisionIP and its sister application VisFlowConnect [29] have been developed over the past two years as part of the Security Incident Fusion Tool (SIFT) project at NCSA [22]. The SIFT project aims to discover security incidents from source data assembled from heterogeneous sources through the use of visualization and data-mining techniques.

NVisionIP utilizes Argus NetFlow data [1] to present a visual representation of the traffic on an entire class-B IP network on a single screen. The visualization presented is based upon either the number of bytes transmitted or the number of flows to or from the hosts on the network and can be filtered based upon a number of attributes useful in categorizing security incidents. In addition to this global view of the network, the user can "drill-down" to view statistics about a small subset of hosts and again from there to view detailed information about a single host. This functionality

VizSEC/DMSEC'04, October 29, 2004, Washington, DC, USA.

is all presented through a single user interface that allows the security analyst a single place to look for an overview of the current state of the network.

The rest of this paper is organized as follows: Section 2 details necessary background and provides an introduction to important related work; Section 3 discusses the implementation of NVisionIP, including an overview of the system architecture and deployment scenarios; and Section 4 presents a sampling of some attack and misuse scenarios that our tool can be used to detect. In Section 5 we will present our conclusions and then end the paper with a discussion of future work in Section 6.

2. BACKGROUND AND RELATED WORK

2.1 Situational Awareness and Visualization

As a means of detecting what is currently happening in the network, administrators have traditionally relied on intrusion detection systems (IDSes). There are two classes of IDSes that are in widespread use: network-based IDSes and host-based IDSes. Network based IDSes, such as Snort [24] work by monitoring packets as they traverse a network. Host-based IDSes, as in [18] and [10], monitor the events taking place on a single host. Depending on the particular IDS deployed, these systems could either raise alerts when a known attack signature is recognized or when an anomalous usage pattern is detected. If the IDS finds an exceptional event, a notification is raised. This can take the form of an email to an administrator or an event written to a log file.

There are several shortcomings to relying on IDSes to provide a sense of situational awareness in a large network. First, the range of attacks that can be detected is usually limited to known attacks that match a certain signature. Second, in systems that detect anomalies, the rules used by the system need to be tuned to minimize the number of false positives while not allowing too many attacks to go undetected. Lastly, if a large network is under heavy attack, the number of alerts that are sent to the administrator of the system can be overwhelming. For instance, at our installation, a group of computers recently came under attack and their administrator received in excess of 4000 alerts in a one-day period from a highly-tuned IDS. This amount of information is unnecessary and hinders the productivity of an administrator trying to manage the systems being attacked. To help address this problem, there are commercial products available to visualize these events, such as [25, 20].

In addition to the work done on visualizing IDS events, researchers have put much effort into visualizing events that take place on a single system of interest. In [6] and [7] the authors present a system to visualize the traffic into and out of a host of interest. Different types of connections are represented as various types of graphic arrows pointing into a host node. Based upon the characteristics of the arrows, users can determine if usage patterns are potentially attack related.

In [11], the authors present a tool to help automate the task of analyzing intrusions to a single host. Starting from a single event of interest (deletion of a file, creation of an account, etc.), the tool creates a graph of events that precede the event of interest. By backtracking through connected events, it is possible to start with the symptom of an intrusion and use the graph to find the point at which the intrusion took place. There is, however, quite a high logging overhead when using this system.

People have been mapping the Internet since it was incarnated as the ARPANET [4]. Some of these maps document the physical locations of computers and communication lines, while others focus on connectivity patterns and traffic volumes. To date, many researchers and corporations use tools such as OPNET [17] and ns–2 [16] to analyze the networks that they administer and model protocols that are under development. While these tools are useful for modeling and simulation, they do not provide a means for real-time analysis of the traffic currently in the system.

In addition to mapping networks, numerous research projects have focused on monitoring networks to visualize traffic. In [26], BGP routing data has been visualized to look for security incidents on the Internet. This work does not provide a sense of situational awareness of a particular network as it analyzes traffic between autonomous systems.

A new tool to enhance situational awareness is the Spinning Cube of Potential Doom [14]. This tool represents network traffic as points in 3D space. The addresses of the network being monitored lie on one axis, all possible source IP addresses lie on a second axis, and the third axis represents port numbers. The color of the points represent different characteristics of the traffic flows on the network. This presentation is similar to that of NVisionIP, though it tends to be more "busy." While it manages to visualize large trends in the network, users cannot interact with the visualization to refocus the display or drill down to explore events of interest in greater detail.

2.2 NetFlows

NetFlows are records that represent aggregate traffic between two hosts. The information saved in a NetFlow record includes the IP address and port numbers of the source and destination, the protocol type of the traffic, the volume of traffic sent and various other attributes.

NetFlow data is collected at a granularity that is optimal for tools that aim to enhance network security or provide network situational awareness. Rather than becoming overwhelmed by trying to examine each packet that traverses the network, NetFlows allow the security analyst to look at higher-level trends of traffic flow across the network. These trends can reveal interesting patterns that may otherwise be "lost in the noise" if analyst was to try to examine raw packet traces. At the same time, they provide enough information to be useful, rather than a completely aggregated summary of traffic on the network such as "we saw 10 million requests today."

Some uses of NetFlow information for security monitoring have previously been explored and documented [5]. To this end, several tools have been developed to aid in exploiting this information source, such as those discussed in [8], [15], and [19]. These tools tend to work well for analysis of security properties on some level, but do not provide a sense of situational awareness. Some present information at too high of a level, while others will find nearly every minutiae– so long as the analyst knows *exactly* what they are looking for. In the next section we will explain how our tool, NVisionIP, is an improvement upon previous tools and allows the security analyst to get a "big picture" view of what is happening in their network and proceed from there to find any details of interest.

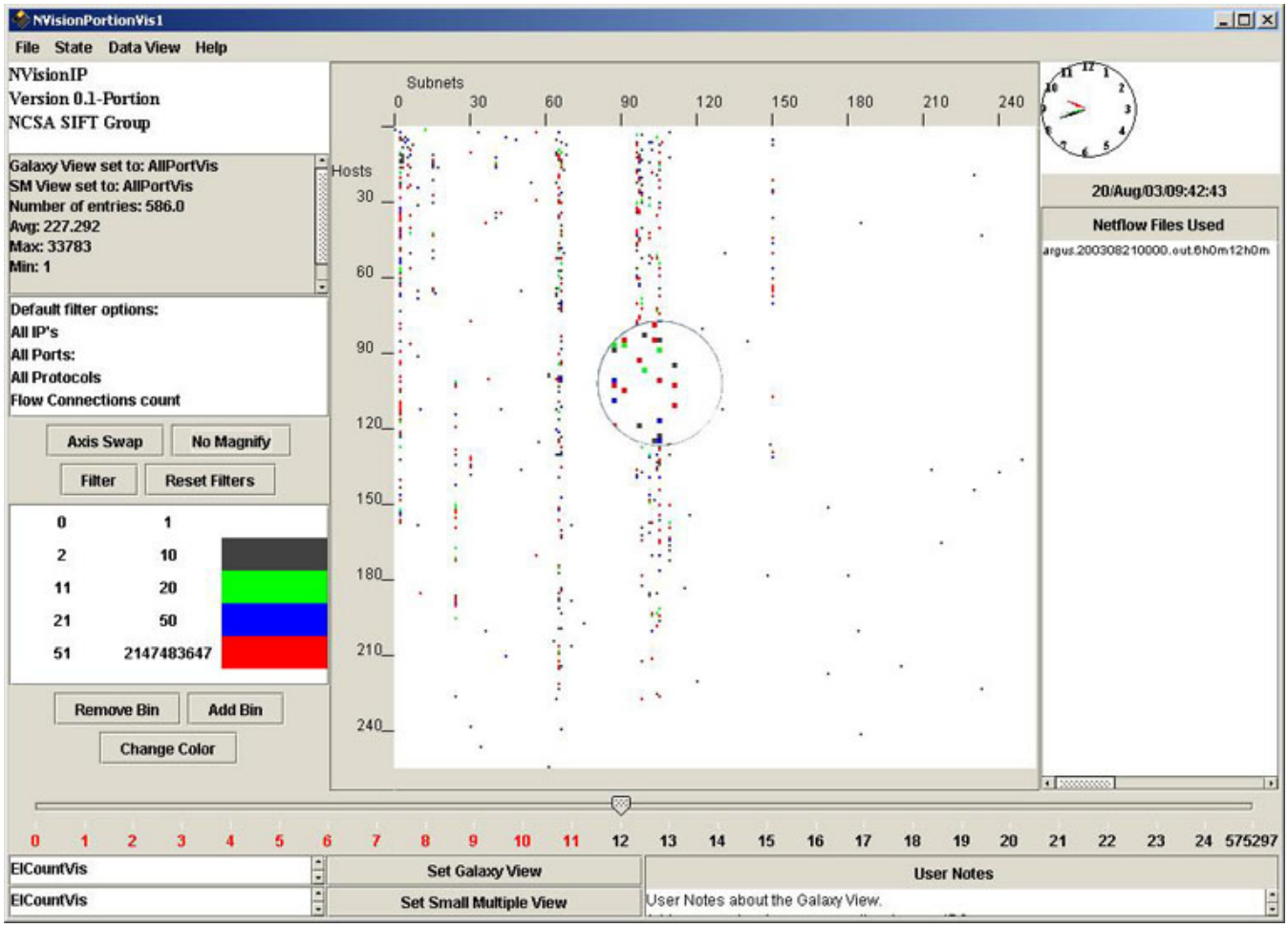

Figure 1: The NVisionIP user interface (with magnifier activated in galaxy view)

3. NVISIONIP IMPLEMENTATION

In this section we present our tool for visualizing the state of a network, NVisionIP, in detail.

3.1 System Architecture

NVisionIP was developed in Java and runs inside of the Data-to-Knowledge (D2K) data mining software package [9]. D2K provides a drag-and-drop programming environment where components developed by the user can be integrated with data mining and visualization components provided as part of the system. In NVisionIP, we use D2K to read the NetFlow data from the flow file of interest. D2K then passes this information to a component developed in-house which computes statistics of interest about the data. After the statistics have been computed, we create and display the visualizations of interest, using the D2K framework. As the user interacts with the program, parts of this data-path can be repeated, allowing more data to be processed and visualized. For further detail about the system architecture of NVisionIP, see [2].

3.2 Deployment Scenario

Typically, networks will have multiple routers capable of recording NetFlow data. In this instance, flows are recorded

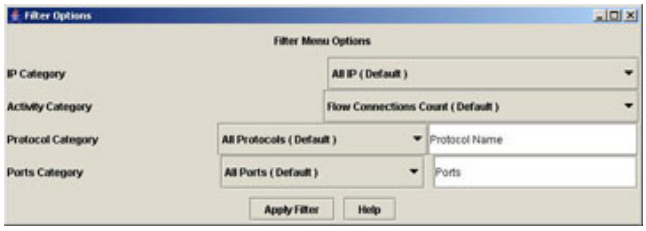

Figure 2: The galaxy view filtering options

at each router and sent over UDP to a central collection point that aggregates the flow data into a single flow file. By combining flows from multiple points in the network, security analysts are able to get a picture of what is happening throughout their network, rather than focusing on isolated points. For this reason, NVisionIP uses aggregate flow files to visualize the activity on the network.

3.3 User Interface

The main strength of NVisionIP lies in its user interface. The network can be visualized at three levels of granular-

ity, each of which will be discussed in greater detail below. Figure 1 shows the NVisionIP user interface. Through this interface, the user can activate a filtering dialog that limits which host's data flows are visualized. Flows can be filtered based upon any combination of IP address (source, destination, either, or subset), ports (source, destination, either, or subset), protocols (all or subset) and display type (number of flow records or byte count). Uses of these various filtering capabilities to detect attacks and misuses of the network are discussed in Section 4. Figure 2 shows NVisionIP's filtering dialog box.

Galaxy View

The broadest possible view of the network is the galaxy view, shown in Figure 1. The galaxy view gives a visual picture of the current state of an entire class-B network. All subnets of the network are listed along the top axis of the galaxy view, while the hosts in each subnet are listed down the vertical axis. For instance, if the top two octets of the network being monitored are 141.142, then the point located at $(122, 45)$ would represent some traffic characteristic of the host whose IP address is 141.142.122.45. In addition to visualizing the current state of the network, the galaxy view can animate traffic collected over some period of time.

When designing the galaxy view layout, we considered a number of different potential layouts. Three such layouts are shown in Figure 3. The first representation among the three possibilities is the current galaxy view, in which hosts are laid out in a subnet $\times$ host coordinate system. Each host is colored depending upon some characteristic of interest. This presents an intuitive view of the network, in which any visual anomaly that can be detected can be mapped to the corresponding host or hosts immediately.

The second option is a more logically oriented view of the network. Rather than using the coordinate system described above, hosts are clustered according to function (eg., file servers, DHCP assigned address space, computer labs, etc.). This way, the administrative domain that a misbehaving host belongs to can be easily determined. At that point, the appropriate people can be contacted to aid with the containment or patching of the rogue or compromised host.

The third option represents a more abstract approach to visualizing the network, similar to the work presented in [21]. In this view, hosts with some particular characteristic of interest are represented larger or smaller based upon that characteristic. Characteristics of interest include traffic volume, number of flows, or flows on a particular port.

There are two main reasons for using our current galaxy view over the other two options. The first of these is the desire of security analysts. In interviews with security personnel at our site, it became apparent that they were interested in a consistent IP-space based visualization of the raw NetFlow data [30]. As the visualization offered by our third option is inconsistent with respect to physical network layout and changes depending on which trait is monitored, it was eliminated from our choices. The first option is clearly more based on the IP-space being monitored than the second and can offer the same clarity of presentation if subnets are laid out intelligently (eg., file servers on one subnet, human resources machines on another subnet, etc.). For these reasons, the first galaxy view was chosen.

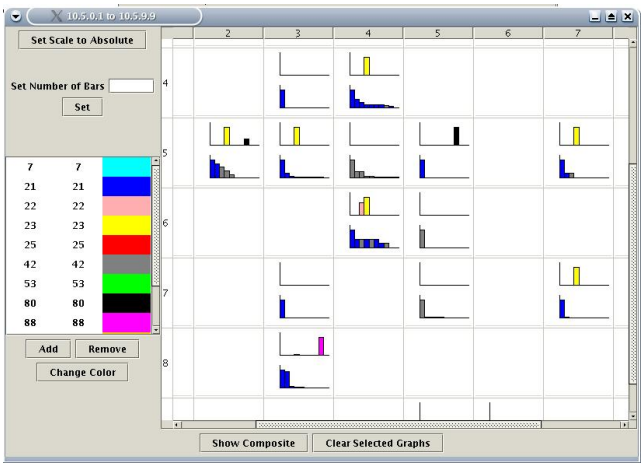

Figure 4: The small multiple view

Small Multiple View

If the analyst using NVisionIP sees a particular area of the galaxy view that is beginning to show signs of abnormal traffic patterns, they can zoom in on that region. Selecting a rectangular region of the galaxy view will open a small multiple view (SMV) window for the selected region. In this view the analyst is presented with more detailed information about the set of hosts selected. Figure 4 shows a small multiple view window encompassing many hosts.

In the SMV, each host in the region selected is represented by two bar charts. The top chart represents traffic on a number of well-known ports while the lower chart represents traffic on all other ports including the ephemeral ports (those greater than 1024). The charts are drawn on a relative basis by default, in which traffic levels are compared on a per-host basis. Absolute comparison levels can be defined, which allows for easier comparison across multiple hosts. To aid in cross-host traffic comparison, user can assign colors to traffic on different ports, making it easier to pick out traffic levels for flows of specific interest.

Machine View

The most detailed view that can be presented by NVisionIP is the machine view. By clicking on a single host in the small multiple view, its machine view is exposed. In the machine view, a wide variety of visualizations are available, including:

- Byte and flow counts for all protocols used
- Byte and flow counts for all ports used
- Byte and flow counts for all TCP traffic
- Byte and flow counts for all UDP traffic
- Byte and flow counts for all traffic using a protocol other than TCP or UDP

In addition to these visualizations, the host's raw NetFlow records are accessible, as are some basic statistics regarding the host's flow counts. Figure 5 shows an example machine view window. Figure 6 illustrates graphically the relationships between the galaxy view, small multiple view, and machine view within NVisionIP.

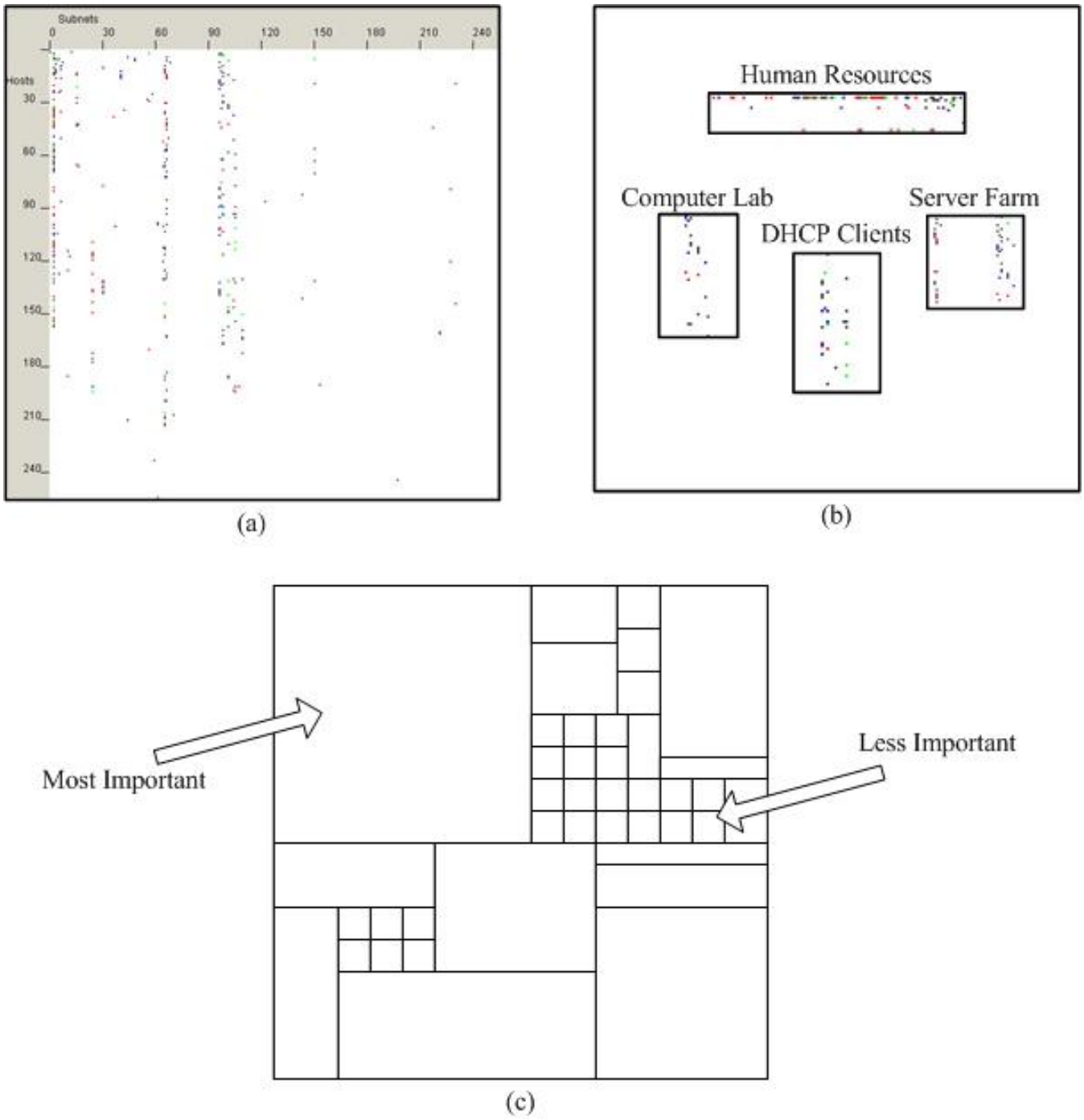

Figure 3: In (a), hosts are listed in a subnet $\times$ host coordinate system and colored based upon the value of some characteristic of their traffic. Picture (b) shows a labeled cluster view of a network that using the same coloring scheme as (a). In (c), hosts are represented by boxes whose size reflects the characteristic of interest.

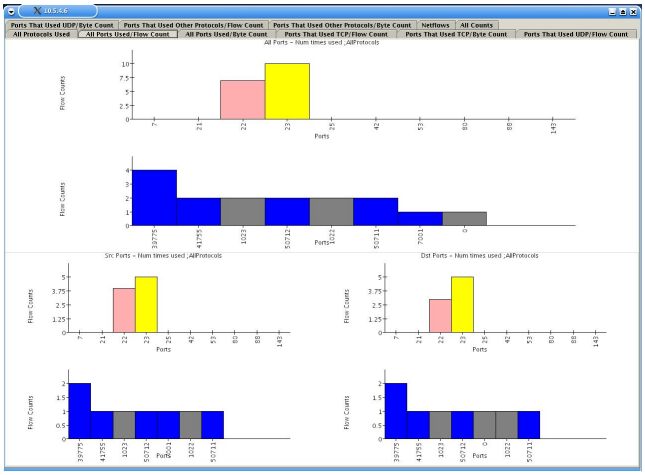

Figure 5: The machine view

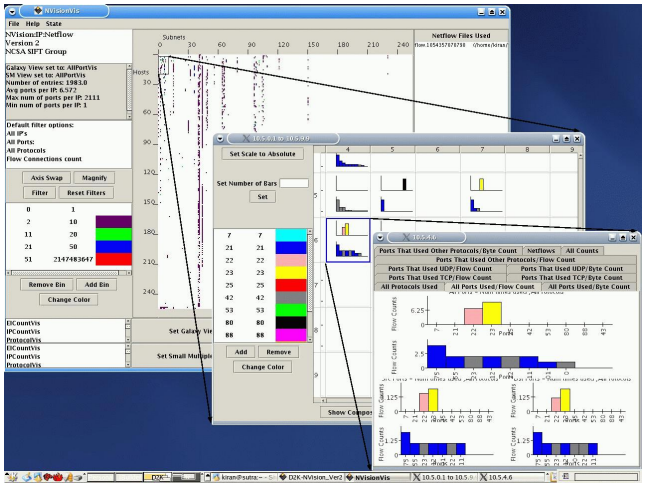

Figure 6: The relationships between the galaxy, small multiple, and machine views

4. ATTACK AND MISUSE SCENARIO DETECTION

As was shown in Section 3, NVisionIP can give very meaningful visualizations of NetFlow data records. The network analyst can zoom in from a high-level profile of the network to a more detailed view of multiple machines. From there, the user can drill down to the individual machine level, gaining access to many statistics, graphs, and even the individual flow records. In this section, we present how NVisionIP allows the user to detect various types of attack and misuse through the use of visualization.

Worm Infection

Perhaps the most basic security function that NVisionIP can perform is the location of hosts infected with a virus or worm. Many types of worm spread by probing for other hosts to infect. For instance, the Slammer worm sent 376–byte packets to UDP port 1434 of random hosts in an attempt to propagate [23]. After noticing an increase in the number of flows leaving their network, an analyst could filter the displayed traffic to only include flows originating in the network with a destination port of 1434 transmitted using UDP. Hosts with large numbers of flows meeting these criteria would be indicated in red in the Galaxy view and could alert analysts that these machines are possibly infected and should be taken offline and patched.

Compromised Systems

Many times, when a host is compromised, the attacker will install software that allows remote access to the machine. In this way, compromised hosts can act as DDoS zombies or file servers. For instance, it is not uncommon to find IRC "bots" installed on compromised hosts to turn them into file servers. NVisionIP can aid in the detection of such hosts easily. If the network analyst detects large volumes of traffic in regions of the network where they were previously absent, they can zoom in to a small multiple view of that region of the network. If traffic is detected on ports 6667 or 6668 (typical IRC ports), this could alert the network analyst that the machine is potentially compromised.

Misuse

Misuse of computer networks (as defined in an organization's "acceptable use policies" or similar document) can be detected using NVisionIP similarly to the above two scenarios. Abnormal high volumes of traffic can easily be detected visually by the analyst. From this point, they can drill into the detailed information for the machine in question and locate what services are running and explore exactly how it is that the system in question is violating the policies of the organization.

Port Scans

Port scans are perhaps one of the most easily detected types of patterns that can be detected though the use of NVisionIP. Port scans of an individual host can be detected by noticing a large number of ports in use on a particular host. The most noticeable type of port-scan, however, occurs when an attacker scans each host on a particular subnet. Figure 7 illustrates what this type of scan looks like in NVisionIP's galaxy view. In reality, the entire class-B address space being monitored is unlikely to be populated with hosts, so bizarre striping patterns are usually indicative of some form of probing or scanning activity.

Denial of Service Attacks

Denial of service attacks can also be readily detected by NVisionIP. In the instances where hosts on the network under surveillance are launching a distributed denial of service attack, the network analyst will notice sudden spikes in traffic volume from these hosts, indicated by a color change in the pixels representing those hosts. However, if the DDoS attack is sufficiently distributed, NVisionIP will not assist in its detection, as the volume of traffic sent by each participating node will be too low to distinguish from normal traffic.

In the case where hosts on the network under surveillance are under attack, NVisionIP will change their color to indicate the rising amounts of traffic received. However, care must be taken to ensure that a valid attack is taking place and that the increase in traffic does not simply correspond

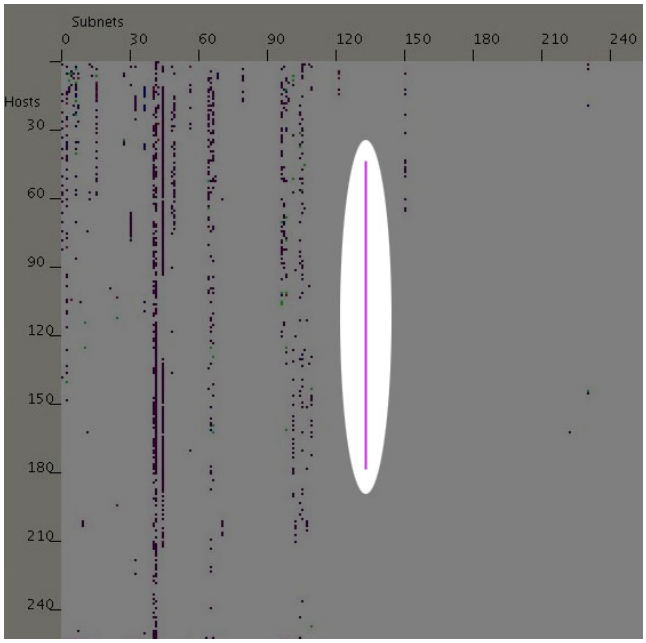

Figure 7: IP-scan activity

to the release of a new software package or start of an automated backup, for instance.

5. CONCLUSION

The number of attacks on computer systems connected to the Internet is growing at an alarming rate. One of a security analyst's best defenses is the knowledge of the current state of their network at all times. Using this situational awareness, they can detect the first traces of abnormal network behavior and patch holes to stop attacks before they have a chance to cause serious damage. Unfortunately, there is a severe lack of tools that can provide today's security analysts and systems administrators with the sense of situational awareness that they need.

NVisionIP is a visualization tool that can present a wide range of information about the characteristics of an entire class-B network on a single screen. NVisionIP reads NetFlow data records that provide a snapshot of the activity on the network. This data can be filtered and aggregated based upon a number of attributes that are important for security analysis. The data can be visualized at the level of the entire class-B network, a range of hosts of interest, or at the level of extremely detailed information about a single host. All of this visualization takes place through one intuitive user interface.

Humans can quickly recognize patterns and trends in graphically presented data. It is hoped that through the use of NVisionIP, network administrators will be able to learn what their network looks like under "normal" conditions. At the first signs of attack, the visual representation of the network will change, alerting the administrator that the network is in a non-normal state. Through the use of the NVisionIP user interface, the administrator will be able to drill-down into the abnormal areas of the network to learn in greater detail exactly what is happening and how to combat this threat.

6. FUTURE WORK

Future work on NVisionIP is divided into three main areas: extracting the NVisionIP functionality out of the D2K environment, allowing new NetFlows to be imported "on the fly," and the addition of other security related functionality.

Members of our team are currently working on porting NVisionIP out of D2K to run as a standalone Java application. Removing NVisionIP from the D2K framework would make NVisionIP more accessible to a wider variety of users. As a security tool, we would like to see NVisionIP available to any user that feels the need to use it. Freeing end users from the licensing agreements associated with D2K is one way to do this. In addition, this would reduce our maintenance overhead, as we would not need to change our code each time that the architecture of D2K changes.

Currently, NVisionIP requires the NetFlow logs of interest to be loaded when the program is started. While this allows the user to navigate flow information for some archived period of time, it does not allow for a streaming view of the network. By allowing NetFlow records to be imported in a streaming mode, analysts will be kept up to date with regards to the current activity on the network. For instance, a large projection screen with the current view of the network could become a fixture in network operation centers, allowing each analyst at the NOC to keep a constant eye on the state of the network, while still processing other information at their individual workstation.

Section 4 of this paper briefly presented some scenarios in which NVisionIP allows network analysts to locate attacks against their systems or misuse thereof. In the future, we plan to conduct a formal analysis on the types of attacks that NVisionIP can help to detect. We anticipate finding more types of attacks that could be detected or located more easily with the help of new visual functionality. It is anticipated that new filtering and aggregating characteristics may help our visualizations present more information in different ways that could be of use to security analysts.

7. ACKNOWLEDGMENTS

We would like to acknowledge the significant intellectual input of our SIFT colleagues whose work and insights indirectly contributed to this paper: Cristina Abad and Adam Slagell. We would also like to acknowledge NCSA collaborators Jim Barlow, Tim Brooks, Jeff Rosendale, and Aashish Sharma of the NCSA Security Operations and Incident Response team.

8. ADDITIONAL AUTHORS

Ratna Bearavolu (NCSA, email: `ratna@ncsa.uiuc.edu`), Yifan Li (NCSA, email: `yifan@ncsa.uiuc.edu`), and Xiaoxin Yin (NCSA, email: `xiaoxin@ncsa.uiuc.edu`).

9. REFERENCES

[1] Argus – metrics. Web Page, Mar. 2001. ⟨`http://www.qosient.com/argus/metrics.htm`⟩.

[2] Ratna Bearavolu, Kiran Lakkaraju, William Yurcik, and Hrishikesh Raje. A visualization tool for situational awareness of tactical and strategic security events on large and complex computer networks. In

IEEE Military Communications Conference (Milcom), 2003.

[3] CERT/CC Statistics 1988–2003, Jan. 2004. ⟨http://www.cert.org/stats/⟩. (Jun. 2004).

[4] Martin Dodge and Rob Kitchin. *Atlas of Cyberspace*. Addison Wesley, Harlow, England, 2001.

[5] Jana Dunn. Security applications for cisco netflow data. Technical report, SANS, Jul. 2001. ⟨http://www.sans.org/rr/papers/index.php?id=778⟩.

[6] Robert F. Erbacher and Deborah Frincke. Visual behavior characterization for intrusion and misuse detection. In *SPIE '2001 Conference on Visual Data Exploration and Analysis VIII*, pages 210–218, Jan. 2001.

[7] Robert F. Erbacher, Kenneth L. Walker, and Deborah A. Frincke. Intrusion and misuse detection in large-scale systems. *Computer Graphics and Applications*, 22(1):38–48, Jan.–Feb. 2002.

[8] Mark Fullmer and Steve Romig. The osu flow-tools package and cisco netflow logs. In *14th Systems Administration Conference (LISA 2000)*, Dec. 2000.

[9] NCSA Automated Learning Group. *D2K Toolkit User Manual*. National Center for Supercomputing Applications, Apr. 2003. ⟨http://algdocs.ncsa.uiuc.edu/TU-20030425-1.pdf⟩.

[10] Gene H. Kim and Eugene H. Spafford. The design and implementation of tripwire: a file system integrity checker. In *Proceedings of the 2nd ACM Conference on Computer and communications security*, pages 18–29. ACM Press, 1994.

[11] Samuel T. King and Peter M. Chen. Backtracking intrusions. In *Proceedings of the 2003 Symposium on Operating Systems Principles (SOSP)*, Oct. 2003.

[12] Kiran Lakkaraju, Ratna Bearavolu, and William Yurcik. Nvisionip – a traffic visualization tool for security analysis of large and complex networks. In *International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communications Systems (Performance TOOLS)*, 2003.

[13] Kiran Lakkaraju, William Yurcik, Ratna Bearavolu, and Adam J. Lee. NVisionIP: An Interactive Network Flow Visualization Tool for Security. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2004.

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

[15] John-Paul Navarro, Bill Nickless, and Linda Winkler. Combining cisco netflow exports with relational database technology for usage statistics, intrusion detection, and network forensics. In *14th Systems Administration Conference (LISA 2000)*, Dec. 2000.

[16] The network simulator – ns–2. Web Page, May 2004. ⟨http://www.isi.edu/nsnam/ns/⟩.

[17] OPNET Technologies, Inc. Web Page, Jun. 2004. ⟨http://www.opnet.com⟩.

[18] Adam G. Pennington, John D. Strunk, John Linwood, Griffin, Craig A.N. Soules, Garth R. Goodson, and Gregory R. Ganger. Storage-based intrusion detection: Watching storage activity for suspicious behavior. In *USENIX Security Symposium 2003*, 2003. ⟨http://www.pdl.cmu.edu/PDL-FTP/Secure/usenix03.pdf⟩.

[19] Dave Plonka. Flowscan: A network traffic flow reporting and visualization tool. In *14th Systems Administration Conference (LISA 2000)*, Dec. 2000.

[20] Secure decisions. Web Page, Jun. 2004. ⟨http://www.securedecisions.com/⟩.

[21] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[22] Security incident fusion toolkit SIFT, Jun.

[23] CERT Advisory CA-2003-04 MS-SQL Server Worm. Web Page, Jan. 2003. ⟨http://www.cert.org/advisories/CA-2003-04.html⟩.

[24] Snort: The open source network intrusion detection system. Web Page, Jun. 2004. ⟨http://www.snort.org⟩.

[25] Security threat manager. Web Page, Jun. 2004. ⟨http://www.open.com/products/threatmanager/threatmanager.shtml%⟩.

[26] Soon Tee Teoh, Kwan-Liu Ma, S. Felix Wu, and Xiaoliang Zhao. Case study: Interactive visualization for internet security. In *IEEE Visualization*, 2002.

[27] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, P.O. Box 430, Cheshire, CT 06410, Second edition, Jan. 2001.

[28] United States Department of Homeland Security. *Team Coordination Training, Student Guide*, May 2004. ⟨http://www.cgaux.info/g_ocx/training/tct/⟩.

[29] Xiaoxin Yin, William Yurcik, Yifan Li, Kiran Lakkaraju, and Cristina Abad. Visflowconnect: Providing security situational awareness by visualizing network traffic flows. In *Workshop on Information Assurance (WIA04) held in conjunction with the 23rd IEEE International Performance Computing and Communications Conference (IPCCC)*, 2004.

[30] William Yurcik, James Barlow, Kiran Lakkaraju, and Mike Haberman. Two visual computer network security monitoring tools incorporating operator interface. In *ACM CHI Workshop on Human-Computer Interaction and Security Systems (HCISEC)*, 2003.

[31] William Yurcik, Kiran Lakkaraju, James Barlow, and Jeff Rosendale. A prototype tool for visual data mining of network traffic for intrusion detection. In *3rd IEEE International Conference on Data Mining (ICDM) Workshop on Data Mining for Computer Security (DMSEC)*, 2003.