

Two Visual Computer Network Security Monitoring Tools Incorporating Operator Interface Requirements

William Yurcik James Barlow* Kiran Lakkaraju Mike Haberman

National Center for Supercomputing Applications (NCSA)

University of Illinois at Urbana-Champaign

605 E. Springfield Avenue

Champaign, IL 61820 USA

{byurcik,jbarlow,kiran,mikeh}@ncsa.uiuc.edu

ABSTRACT

The critical role of the human operator in security operations has not been a focus of existing tools created by security developers. In this paper we describe interface requirements for usable and effective security operations tools to assess security situational awareness on large and complex computer networks. We have developed two prototype security monitoring tools based on these interface requirements and are progressing on usability studies to evaluate their effectiveness.

Keywords

security operations, intrusion detection, visualization

INTRODUCTION

Computational security on most networked systems is precarious and getting worse. A small configuration change at any of multiple levels (application, operating system, firewall) can make otherwise secure systems instantly vulnerable to attack. Other systems are already insecure to begin with due to unpatched software vulnerabilities or poor configurations – these systems just await hacker discovery and exploitation using automated scanning tools [7]. These situations consider only known attacks, preparing against unknown future attacks that are discovered daily is an open research question. As if this were not bad enough, corporate surveys consistently report that insider attacks, staff with proprietary knowledge of security operations, are the greatest threat [4].

The state-of-the-art protection provided by security staff is typically the use of multiple tools to monitor different parts of networked systems for security. Examples include firewall logs for monitoring unauthorized access attempts, network intrusion detection systems for attack signatures within traffic, host intrusion detection systems for suspicious changes in operating systems such as file modifications and new accounts, and lastly applications

designed exclusively to maintain authentication and authorization for an organization. While each of these tools may provide unique information, they suffer from drawbacks: (1) they provide information limited to a specific view of a network, (2) operators must develop expertise in multiple cryptic tools that change frequently, and (3) multiple tools do not currently provide cross-cues or fusion for events in complex environments.

Seeking to improve this situation in a dramatic way, we have developed two software tools that provide visual situational awareness of an entire network based on a single operator interface. We do this using a visualization derived from audit logs that are continuously collected. One tool is focused on forensic data mining and the other tool is focused on animation/video playback. Both tools are being extended for real-time monitoring but at present they are near-real-time in that output is derived from input batch files although these files may be available for input in different time increments including month, week, day, hour, minute, or seconds.

The contribution of this research is a tool that allows an operator to visually assess the situational awareness of an entire network for security on one screen. In our specific case, we provide an operator a view of an entire Class B IP address space consisting of 65,536 computers with each computer having 65,536 ports.¹ We do this by leveraging human cognitive processing based on specific interface requirements elicited from security operators: (1) it is estimated that humans can visually process a screen of information at about 150 MB/s, (2) human vision can discriminate tiny but high contrast visual effects (minimum level of discrimination for color or motion or shape), and (3) humans perform well at recognizing visual patterns especially when real-world intuition can be used (ecological design).

The remainder of this paper is organized as follows: Section

¹ Note some IP addresses and ports are reserved or otherwise unavailable.

* corresponding author, NCSA Computational Security Team, telephone: (217) 244-6403, fax: (217) 244-1987.

two states requirements for a situational awareness security operations tools as elicited from security staff. Section three presents prototypes of the two tools we have developed. Section four discusses future directions based on preliminary feedback from security staff and interface developers. In the Section five we end with a summary and conclusions.

OPERATOR INTERFACE REQUIREMENTS

After evaluating state-of-the-art security operations tools and determining that all current tools needed human factor redesign², we decided to create our own tool based on requirements elicited directly from security operators. The security operators who contributed requirements include four NCSA security staff members and several security operators from two different forums of incident response teams (FIRST and CICSWG).³

The primary interface requirement voiced by all security operators is the need for an overall situational awareness view of an entire network. We found good reasons for this: (1) the ability to provide concise reports to upper management either periodically or upon demand, (2) the ability to comprehend status of a network as a whole at different levels, and (3) the ability to continuously monitor for changes anywhere on an entire network.

A concise status report of the state of security for an entire network is elusive. Scanning tools such as Nessus or Internet Security Scanner provide a static risk management profile of known software vulnerabilities on computers within a network but do not report security events. What is needed is an interface report that can efficiently and quickly convey to a human the overall network security status, including both vulnerabilities and security events.

Monitoring a network as a holistic system is important because a malicious software foothold (stepping stone) anywhere within a network perimeter can endanger all machines within the internal network as well as other external networks of computers [9]. There are relationships between individual security events (an intrusion on an individual machine) and network-wide security events (disruptions and/or attacks on multiple machines across the network). An interface ability to compare macro and micro security views of a network simultaneously may provide operator comprehension of these relationships.

Monitoring continuously is important because security protection needs to be dynamic against intelligent attackers that seek stepping stones. Static security protection will eventually be circumvented by persistent attackers just as changes in technology over time will evolve both attack and defense techniques. In fact, the ability to continuously monitor in order to detect security events, even the smallest event in otherwise hidden parts of a network, may be the most effective protection. While this is the function of current intrusion detection systems (IDSs), these tools have a fatal flaw in that while they can detect signatures of known attacks – they are blind to new attacks. What is needed is a way to monitor for security events based on data inherent to network operations (as opposed to relying on importing external attack signatures) that can be concisely represented in an interface.

Other general requirements from operators include:

- A user-friendly interface so the security operator does not have to also be a software developer or system administrator. This includes initial installation and configuration, rendering speed, inputting data, and changing views.
- Flexibility to query all distinguishable features from source data, asking for some of these features may not make initial sense to software tool developers but unusual security events occur that make such searches valuable.
- Dynamic view of network events over different time scales (seconds, minutes, hours, days, months, years) since attacks may be fast-paced or long-latency or anywhere in between.
- Cross-cueing between events since the complete anatomy of an attack often has a sequential sequence (reconnaissance->exploit->new account->root access->rootkit->attacks on other computers).
- Identification and monitoring of critical computers (authentication, clusters, servers) separate from non-critical computers.
- Profiling of distinguishable classes of computers by activity type, activity volume, and time.

Other specific requirements from operators include:

- Raw port activity for well-known ports below 1024 and dynamic ports above 1024 (both source and destination).
- Indications of port activity above defined thresholds.
- Drill-Down views of traffic by IP address (either an individual IP address or as a group of IP addresses – subnet, geographical, or source/destination IPs).
- Monitoring traffic exclusively to/from the Internet.
- Monitoring traffic exclusively within the intranet.
- Network mapping awareness (pre-attack reconnaissance).
- Port scanning awareness (pre-attack reconnaissance).

² While evaluating human factors in state-of-the-art security operations tools is a valid paper topic in itself, we prefer in this paper to instead focus on our positive results rather than constructive criticism of other security operations tools. We encourage anyone interested in pursuing constructive criticism of state-of-the-art security operations tools to contact the corresponding author for potential collaboration.

³ FIRST = Forum of Incident Response and Security Teams <www.first.org>, CICSWG = Committee on Institutional Cooperation – IT Security Working Group (the academic consortium of Big Ten Universities and the University of Chicago) <www.cic.uiuc.edu/groups/ITSecurityWorkingGroup/>.

- Alerts for connections with suspicious IP addresses (unusual IP addresses where previous attacks have come).

PROTOTYPE SECURITY MONITORING TOOLS

We have developed two tools based on the operator interface requirements contained in the previous section. Resource limitations in terms of time and manpower constrained implementation of all interface requirements such that prioritization of effort occurred. We did not find any of the operator interface requirements to be impossible due to current technology.

Both tools we developed use the NetFlow application as the data source. The NetFlow application, a defacto standard, records lowest level packet flows from a router or a computer anywhere on a network into a log. There is an ability to configure collection of different flow features as well as sampling of flows in order to reduce log volume and application processing load. Although NetFlow is not geared specifically for security, it provides pure (unfiltered) information about network activity that can be used to identify security events. Location of the platform executing NetFlow within a network topology determines which flows will be logged.

The first tool we developed is a forensics data mining tool we will refer to as **NVisionIP:D2K:NetFlow**. Although this tool is independent of source data, we will focus exclusively on NetFlow source data for the purposes of this paper. This tool was created within a Data-to-Knowledge (D2K) software tool for the advantages of its software development environment [2].

NVisionIP:D2K:NetFlow highest level “galaxy view” shown in Figure 1 represents an entire class B IP address space as a grid of 255 X 255 boxes (each box is 2 pixels by 2 pixels) with each box representing an IP address. There are two levels of zoom capabilities available via mouse input: (1) from the galaxy view to a subset of computers within the network and (2) from a subset of computers to an individual computer. Each zoom view shows port activity. As shown in Figure 2, the interactive drill-down capability provides a simultaneous macro-micro view as desired by security operators.

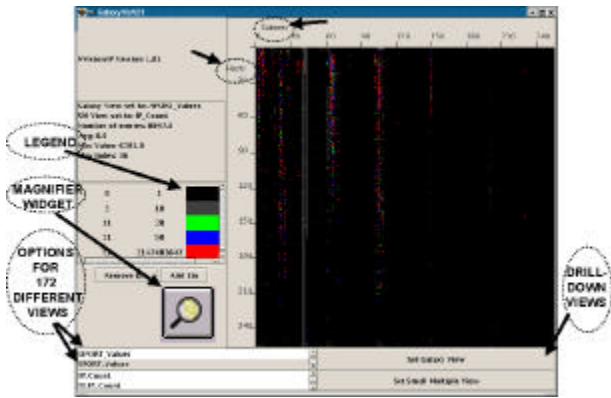


Figure 1: **NVisionIP:D2K:NetFlow** Operator Interface

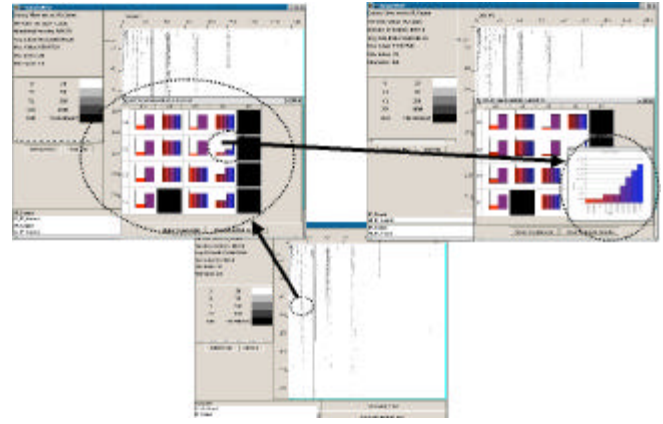


Figure 2: Drill-Down Views

In order to analyze all source data features per operator requirements, **NVisionIP:D2K:NetFlow** has 162 distinct views selectable by the operator as shown in Table 1. This “visual debugging” flexibility promotes finding semantic relationships between features for forensic purposes [3].

Table 1: **NVisionIP:D2K:NetFlow** Selectable Features

IP Addresses (3 options)	Activity Type (2 options)	Protocols (3 options)	Ports (9 options)
all IPs [default]	number of flow connections [default]	all protocols [default]	all ports (source&destination) [default]
only source IPs	number of bytes transmitted	specific protocols	specific ports
only destination IPs	---	all protocols minus specific protocols	all ports minus specific ports
---	---	---	all Destination ports
---	---	---	specific Dports
---	---	---	all Dports minus specific Dports
---	---	---	all Source ports
---	---	---	specific SPorts
---	---	---	all SPorts minus specific Sports

The second tool we developed is an animation/video playback tool we will refer to as **NVisionIP:NetFlow**. This tool is dependent exclusively upon NetFlow source data for graphic processing but is independent of D2K. The interface has no operator interactivity but instead focuses on highlighting dynamic changes over time. The interface is close to 100% content containing a clock, legend, and vertical lines for aligning subnet computers.

Figure 3 shows a **NVisionIP:NetFlow** animation of a Denial-of-Service (DOS) attack scanning a Class B IP address space. The colored dots represent “delta” change in byte flows to/from specific machines during five minute intervals. The NetFlow source data used for this animation

is sampled 1:100 due the effect of GB/s router connection speeds on input file size and processor load.

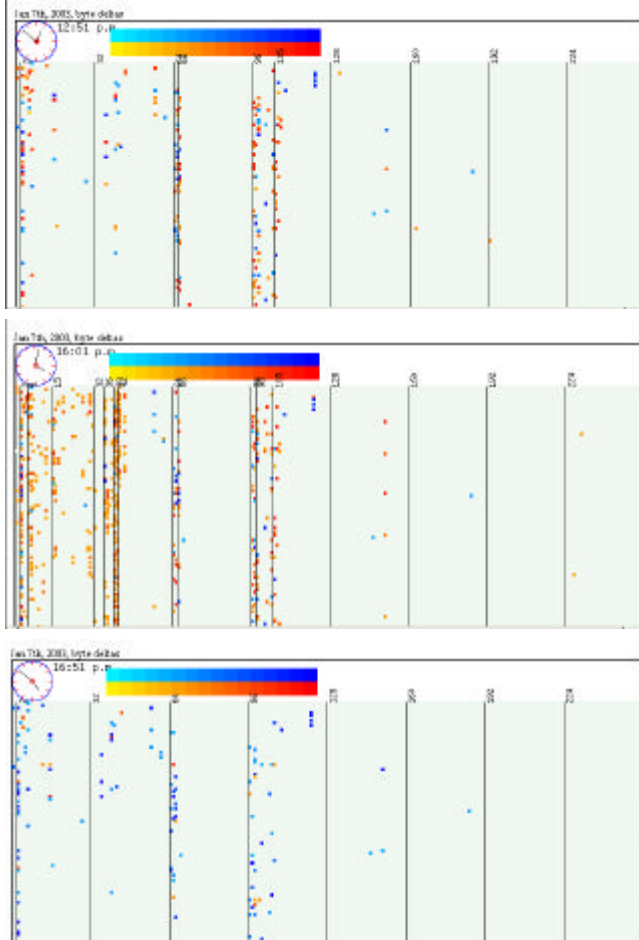


Figure 3: A Before/During/After DOS Scan Animation

PRELIMINARY QUALITATIVE FEEDBACK

There are additional interface requirements after initial use that we plan to incorporate in future releases:

- Natural language query processing (limited input question search space).
- Visual comparison of the statistical history of a feature to the current feature.
- Visual labeling to highlight security events for human comprehension, a proposed set of Information attack icons can be found in [6].
- Connectivity diagrams as an alternate feature view.
- Projection of feature views by large, high resolution, devices such as a tiled wall, immersadesk, and/or CAVE.

The density we have created in the galaxy view of **NVisionIP:D2K:NetFlow** is less than the maximum human visual ability to distinguish 625 points per square inch but it is still small such that a magnification capability is attractive. We have already developed widgets for linear magnification and axis manipulation with future plans for non-linear, fisheye, or 3D/virtual reality capabilities.

As noted in [1], designers have often focused on theoretical threats rather than likely threats and many security products are too complex to use. For the next three months our two tools will be tested by experts in a production environment to measure actual threats and gather quantitative feedback on usability.

CONCLUSIONS

Although visualizing Internet attacks for security operations has been postulated in [5,8], no prototype systems have been developed to test usability and effectiveness. In this paper we describe the elicitation of interface requirements from security operators which were incorporated into a pair of visual security monitoring tools: (1) **NVisionIP:D2K:NetFlow** (forensic data mining) and (2) **NVisionIP:NetFlow** (animation/video playback).

Computer network security is an absolute game where one detail can make all the difference. We propose the use of visualization to leverage human cognitive ability in security operations tools and plan to provide empirical support for this position with results from expert testing with our pair of visual security monitoring tools.

ACKNOWLEDGMENTS

We thank SIFT/NCSA research colleagues for significant indirect support especially: Loretta Auvil, Ruth Aydt, Randy Butler, Dora Cai, David Clutter, Doru Marcusiu, Hrishikesh Raje, Jeff Rosendale, Duane Sears Smith, David Tchong, and Michael Welge.

REFERENCES

1. Anderson, R. Why Cryptosystems Fail. *Communications of the ACM* 37, 11, 32-40.
2. Automated Learning Group, NCSA. *D2K Getting Started Tutorial*. (April 2002).
3. Crossno, P. and Rogers, D. Visual Debugging. *IEEE Computer Graphics and Applications*, (Nov/Dec 2002).
4. *CSI/FBI Computer Crime and Security Survey*. (2002). Available at <<http://www.gocsi.com/>>.
5. Dourish, P. and Redmiles, D. An Approach to Usable Security Based on Event Monitoring and Visualization, *ACM New Security Paradigms Workshop*, (2002).
6. Hosmer, H. Visualizing Risks: Icons for Information Attack Scenarios. *National Information System Security Conference*, (2000).
7. Staniford, S., Paxson, V., and Weaver N. How to Own the Internet in Your Spare Time. *Usenix Security Symposium*, (2002).
8. Varner, P. and Knight, J. Security Monitoring, Visualization, and System Survivability. *SEI/CERT Information Survivability Workshop* (2001).
9. Zhang, Y. and Paxson, V. Detecting Stepping Stones. *Usenix Security Symposium*, (2000).