6-16-2008

Developing Network Situational Awareness through Visualization of Fused Intrusion Detection System Alerts

Serafin A. Avitia V

Developing Network Situational Awareness through Visualizations of Fused Intrusion Detection System Alerts

THESIS

Mr. Serafin Avitia V

AFIT/GCS/ENG/08-23

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCS/ENG/08-23

Developing Network Situational Awareness through Visualizations of Fused Intrusion Detection System Alerts

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science (Computer Science)

Mr. Serafin Avitia V, B.S.C.S.

June 2008

AFIT/GCS/ENG/08-23

Developing Network Situational Awareness through Visualizations of Fused Intrusion Detection System Alerts

Mr. Serafin Avitia V, B.S.C.S.

Approved:

/signed/ 16 Jun 2008

Lt Col Stuart H. Kurkowski, Ph.D (Chairman) Date

/signed/ 16 Jun 2008

Maj Paul D. Williams, Ph.D (Member) Date

/signed/ 16 Jun 2008

Lt Col J.T. McDonald, Ph.D (Member) Date

AFIT/GCS/ENG/08-23

Abstract

With networks increasing in physical size, bandwidth, traffic volume, and malicious activity, network analysts are experiencing greater difficulty in developing network situational awareness. Traditionally, network analysts have used Intrusion Detection Systems to gain awareness but this method is outdated when analysts are unable to process the alerts at the rate they are being generated. Analysts are unwittingly placing the computer assets they are charged to protect at risk when they are unable to detect these network attacks. This research effort examines the theory, application, and results of using visualizations of fused alert data to develop network situational awareness. The fused alerts offer analysts fewer false-positives, less redundancy and alert quantity due to the pre-processing. Visualization offers the analyst quicker visual processing and potential pattern recognition. This research utilized the Visual Information Management toolkit created by Stanfield Systems Inc. to generate meaningful visualizations of the fused alert data. The fused alert data was combined with other network data such as IP address information, network topology and network traffic in the form of tcpdump data. The process of building Situational Awareness is an active process between the toolkit and the analyst. The analyst loads the necessary data into the visualization(s), he or she configures the visualization properties and filters the visualization(s). Results from generating visualizations of the network attack scenarios were positive. The analyst gained more awareness through the process of defining visualization properties. The analyst was able to filter the network data sources effectively to focus on the important alerts. Ultimately, the analyst was able to follow the attacker through the entry point in the network to the victims. The analyst was able to determine that the victims were compromised by the attacker. The analyst wasn't able to definitively label the attack specifically yet the analyst was able to follow the attack effectively leading to Situational Awareness.

Acknowledgements

I am ever thankful to my advisor, Lt Col Stuart H. Kurkowski, for his patience with me during this arduous process. I would like to thank Luke van der Hoeven and Stanfield Systems Inc., especially Tim Jacobs and Delos Ford, for all the effort they contributed to software engineering development of the toolkit during the past year. I would like to thank Maj Jason McDonald for taking the time to help me setup Latex. I'm sure without your help, I would have been lost. I would like to thank the friends I met here at AFIT especially Nate Tuting, Capt Mark Deyoung, and Lt Col Andrew Hanson. The conversations we shared in the lab kept my spirits high during those rough academic quarters. I would also like to thank my family for keeping me on the road to success. Lastly, I would like to thank Meredith for sticking by me these past 18 months. I know there were some rough moments for us during this process, but without your faith in me, I don't know if I would have had the strength to finish this program.

You have to accept whatever comes and the important thing is that you meet it with courage and with the best you have to give. Eleanor Roosevelt (1884 - 1962)

Mr. Serafin Avitia V

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
DoS	Denial-of-Service	1
IDS	Intrusion Detection System	1
SA	Situational Awareness	2
IA	Information Assurance	4
VIM	Visual Information Management	5
JDL	Joint Director of Laboratories	6
NIDS	Network-Based Intrusion Detection System	9
DMZ	Demilitarized Zone	9
HIDS	Host-Based Intrusion Detection System	9
PIDS	Protocol-Based Intrusion Detection System	10
HTTPS	Secure HTTP	10
APIDS	Application Protocol-Based Intrusion Detection System	10
DBMS	Database Management System	10
WMD	Weapon of Mass Destruction	13
IA	Information Assurance	16
XML	Extensible Markup Language	18
IP	Internet Protocol	26
VIM	Visual Information Management	28
RAM	Random Access Memory	29
VIAssist	Visual Assistant for Information Assurance Analysis	32
SIFT	Security Incident Fusion Tool	33
NCSA	National Center for Supercomputing Applications	33
MMA	Model Matching Algorithm	37
UML	Unified Modeling Language	38
AFRL	Air Force Research Laboratory	43

Abbreviation		Page
PNP	Plug and Play	44
SSH	Secure Shell	66
OSIS	Open Source Information System	77

Developing Network Situational Awareness through Visualizations of Fused Intrusion Detection System Alerts

I. Introduction

The world is increasingly becoming interconnected through networks. The most large and complex networks such as those used by the United States Armed Forces allow enormous amounts of traffic volume to flow through them each day. These networks are constantly threatened by the malicious activities of outside hackers and insider threats. The number of network security incidents reported per annum has increased at an exponential rate [17, 29]. These incidents include, but are not limited to, network attacks on vulnerable services, data driven attacks on applications, host-based attacks such as privilege escalation, unauthorized logins, access to sensitive files, and malware such as viruses, trojan horses, and worms. In 2000, Fullmer et.al observed on a daily basis that the busiest router of the Ohio State University network experienced roughly 400 gigabytes of data traffic flow each day representing nearly 600 million data packets [14]. Furthermore, Fullmer pointed out that during Denial-of-Service (DoS) attacks, especially SYN floods, the network traffic greatly increased from the norm previously mentioned [14].

An analyst's traditional method for monitoring the activities of a network include utilizing a intrusion detection system (IDS). An IDS automates the process of identifying and responding to malicious activity targeted at computing and networking resources [1, 4, 18]. It is a tool that examines the network traffic that passes through it against a user-defined set of rules. If a network packet violates these conditions then an alert is stored in a log file and displayed to the IDS console. The analyst prefers and trusts the IDS because a network packet either is a violation or it is not, there is no guessing with a properly configured IDS. Currently, it is the responsibility of the analyst to view each IDS alert and make a split-second decision on whether the new alert is an isolated event or part of a larger event. With each new alert generated by the IDS, the analyst is developing new, and sometimes changing, situational awareness in the network domain. Situational awareness (SA) is knowing what is happening in the environment around you so that you can act appropriately. SA is important in any domain whether on land, sea, space, or cyber. Additionally, it is important to note that SA only exists in humans. There are no tools presently that can form SA, but there are many that support the analyst's development of SA in his/her mind.

Currently in the network domain, analysts monitor their large networks and computer resources using numerous IDSs to perceive the activities in their networks. It is the analyst's responsibility to comprehend each IDS alert as it is received and integrate that perception into the existing situation that he/she must comprehend entirely. The projection of correct decision-making occurs after the analyst trusts the situation as it has developed over time. The problem in this process is the reliance on the human component to perceive and comprehend. The analysts lose the ability to develop trusted SA when confronted with, as [12] states, "a torrent of data" during the perception phase that they "may be even less informed than ever before."

Fusion of network data and use of visualizations are two methods that offer promising reduction of cognitive processing required by the analyst. With a fusion process, several forms of network data can be integrated and pre-processed in order to clean the data so that the analyst receives pertinent data that is not redundant or false-positive. Fusion process determines the legitimacy of each alert by considering its significance in respect to the client/host configuration, the network's mission and existing cleaned data previously selected as significant. The analyst can trust and have more confidence in the data if it comes from multiple sources, giving the analyst more awareness from the network domain. Visualizations allow the analyst to process information quicker. Data properties can be visualized and processed quicker by the analyst than the traditional, reading method. Also, visualizations can show patterns of activity that would not otherwise be obvious. Lastly, incorporating the fusion process into a visualization would allow the analyst to join different data sources creating a richer data source and greater awareness.

1.1 Purpose and Goals

This research is very important because there is a serious requirement placed on the network security analysts both commercial and military to safeguard computer networks to the best of their abilities. These analysts are tasked with having to transform large volumes of data into sensible information. At this time, their cognitive abilities are stretched beyond human capability as network scalability and traffic flow continue to increase at alarming rates. The analyst's current role in a network environment is a central one as illustrated in Figure 1.1. Based on Figure 1.1, one can see that the human component can be a point of failure.

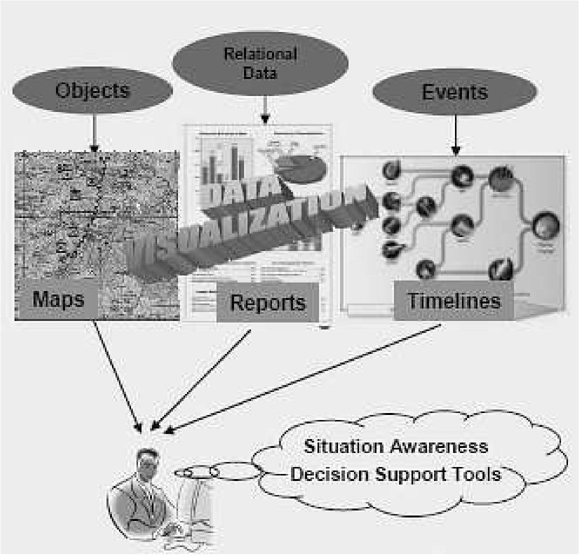


Figure 1.1: Salerno et al. in [23] depict the role of the analyst in today's environment. The massive amounts of data being directed at the analyst are at issue. The analyst is responsible for all processing of the raw data and some information to form SA concept(s) of the current situation. The analyst has moments to do this or else he/she risks losing additional SA.

The IDS is a traditional method for monitoring network activity yet it is not lessening the cognitive strain on the analyst. Often, as mentioned in the introduction, the IDSs implemented on important computer assets contribute to more strain when they generate redundant, false-positive or low-priority alert events. Obviously, the attacker is no help either when he/she purposely generates background noise in hopes that the attack signature remains clandestine.

The responsibility falls on the researchers not only to find effective and efficient solutions to this problem but solutions that the professionals can trust. Trust in the network domain is very important because the analyst must believe in the developing situation as it unfolds so correct decisions are executed in a timely fashion. *The goal of this research is to show the Information Assurance (IA) community that by reducing the quantity of raw alert data into important fused alert data then visualizing the new dataset with a specialized toolkit, that these visualizations are effective in displaying situational awareness and network analysts can trust the information ascertained from this system.*

1.2 Research Assumptions

1. **Fused Alert Data** This research assumes the fusion process correctly generated the fused alert data in the dataset. Each scenario dataset accurately reflects the known parameters of that scenario.
2. **Cyber SA Reference Model** This research effort is solely focused on the Level 2 and above components of the SA Reference Model. We assume that the lower level components exist. Refer to Section 3.2.1 for more details.
3. **Network Analyst** This research assumes the analyst has prior knowledge of the network structure and computer resources. Such knowledge is necessary for determining resource importance, analyst focus, and damage assessment.

1.3 Research Scope

1. **Visualization Toolkit** The Visual Information Management (VIM) toolkit engineered during this research was preselected for its ability to handle the uniqueness of the scenario datasets. Performance metrics will be limited to the toolkit's ability to visualize the necessary network context information versus other toolkits' visualizations of their datasets.
2. **Number of Test Scenarios** Currently, the number of unique scenario datasets is limited and very select. Therefore any results or conclusions from this research will have to be verified in future research efforts for correctness when more datasets become available that include a wider range of network attacks.

1.4 Organization

The remainder of this thesis is divided into four chapters. The next chapter contains four major sections; in-depth discussion of the IDS, other research efforts in the area of IDSs being engineered to aid the analyst in developing network SA, the different SA models and the importance of visualizations when developing SA. Chapter 3 outlines the research methodology to include the boundaries of the problem, the SA Framework utilized in this research, the scenario datasets, the visualization toolkit and the network attack scenarios. Chapter 4 analyzes and interprets the information collected from the visualizations. The last chapter concludes with a summary of the research conducted, discusses research significance and contributions, and suggests areas for future research.

II. Background

This chapter is divided into four major sections, each covering a main topic of interest. The first portion discusses the IDS architecture and classification in detail. The second introduces three models of SA; Endsley, Joint Director's of Laboratories (JDL), and SA Reference. The next section examines the types of visualizations and their uses to convey information more effectively thus enhancing the analyst's situational awareness. The last section will highlight other visualization research currently attempting to improve network SA development in the analyst.

2.1 Intrusion Detection System

In order to monitor a computer network, analysts use a variety of tools yet no tool is more relied upon than the IDS. This tool automates the process of identifying and responding to malicious activity directed at or within a network. Depending on the configuration of the IDS, the alert event could be logged or it can notify the analyst of the activity or both. It should be noted that a majority of the network security community refers to IDS alert events as information while this research will always refer to the alert events as data.

2.1.1 IDS Architecture Schema. An IDS is composed of several major functional components, see Figure 2.1. The next subsections discuss the function(s)

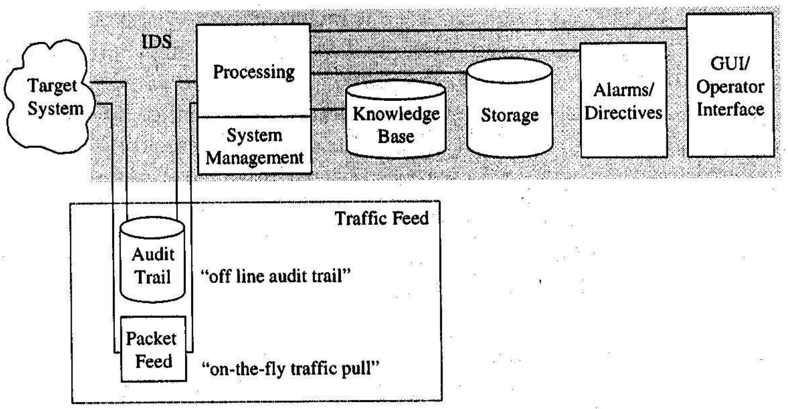


Figure 2.1: In [1], Amoroso et al. illustrate the IDS components and their relationship to each other.

and purpose of the important components in the IDS in order to understand how it generates the data.

2.1.1.1 Target System. The target system refers to the system that is being monitored by the IDS. Logically, the system is important to the analyst otherwise the IDS would not be monitoring it. An example of a target system to monitor would be a internal server hosting a database containing personal medical records of all retired military veterans.

2.1.1.2 Packet Feed. This component according to [1] has two purposes. First, it represents a direct connection from the target system to the IDS's Processing and System Management components. Second, it constitutes a fundamental choice of data processing. Either the traffic flows to the target system are diverted to a centrally located data repository for processing or the traffic flows are analyzed on the target system and the data is sent to the repository post-process.

2.1.1.3 Processing. This component contains the algorithms that exercise and manipulate each data packet in the traffic flow trying to determine if the packet is malicious. These algorithms are also referred to as the attack signatures. They require simplicity in order to process each data packet quickly and avoid network flow bottle-necking. In addition, it is disadvantageous to search for any statistical relationships during the processing phase. First, speed is a necessary factor to have when processing millions of packets each day. Secondly, the rigorous processing required to locate meaningful relationships within the raw data would consume the target system's processor time thereby decreasing its productivity. If this algorithmic processing is off-target, then IDS performance is impacted significantly in terms of responsiveness of the IDS and accuracy of the alert results. The analyst should perform audits on the performance of the IDS at the device location to ensure peak performance.

2.1.1.4 Knowledge Base. The Knowledge Base of the IDS is considered the engine of the system. It stores the information about network attacks in the form of attack signatures, strings and system or user behavior profiles. This component is of great importance to the IDS therefore should have adequate protection from network threats. Most importantly, this component must be accessible by the analyst in order to receive timely signature-based updates for new network attacks.

2.1.1.5 Storage. Storage is very dependent on the size of the network that the IDS monitors. As the network expands so should the storage's capacity. The storage component has a limited cache to store short-term data concerning an open session. Event-related sessions are generally archived for future auditing by the analyst.

2.1.1.6 Alarms/Directives. As the sensor is processing the target system's traffic flow, an alarm will be generated if an attack signature in the Knowledge Base flags the data packet as malicious. The alarm notifies the analyst through the GUI interface. The sensor can be configured to notify other IDSs and IDS Components. The alarm is logged away into a log file for auditing such as network damage assessment.

2.1.1.7 GUI/Operator Interface. Amoroso in [1] states that proper attention must be given to the presentation, combination and representation of the data to the analyst. Additionally, [1] is correct that visualization is critical in aiding the analyst to develop network context and SA. However, visualizations of the raw alert data will not yield large gains for the analyst because without a separate process focused on the reduction of the data into important or semi-informational data related to host/network configuration and the network's mission, the analyst will gain little focus or context essential to the development of SA.

2.1.2 IDS Classification. An IDS is categorized by sensor type, sensor location and engine methodology. Each type is outlined in the next sections.

2.1.2.1 Network-Based Intrusion Detection System. This IDS is commonly referred as a (NIDS). The analyst positions the NIDS sensors at high traffic flow congestion points to capture and analyze every data packet for malicious intent. The most common congestion points of a network are the hubs, network switches and firewalls that guard the entrance/exit into the demilitarized zone (DMZ). An example of a NIDS would be Snort. In [16], Snort.org defines Snort as "an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods." Snort is reportedly the most deployed IDS in the network security community [16].

2.1.2.2 Host-Based Intrusion Detection System. This category of IDS is commonly referred as a (HIDS). Within the HIDS category exists subtypes listed below [6]:

1. **File System monitors** - Systems checking the integrity of files and directories.
2. **Logfile analyzers** - Systems analyzing logfiles for patterns indicating suspicious activity.
3. **Connection analyzers** - Systems that monitor connection attempts to and from a host.
4. **Kernel based IDSs** - Systems that detect malicious activity on a kernel level.

The HIDS consists of a service or agent that searches for intrusions by analyzing a single host machine's state and activities such as system calls, application logs, filesystem modifications. Intrusions such as user privilege escalation would cause this IDS to generate an alert event to the analyst. OSSEC is a scalable, multi-platform, open source HIDS that has a powerful correlation and analysis engine that monitors all the host state and activities mentioned above [8].

2.1.2.3 Protocol-Based Intrusion Detection System. This category of IDS is commonly referred as a (PIDS). A common protocol monitored by this IDS is Secure HTTP (HTTPS). The simpler PIDS will enforce the correct protocol while a more sophisticated PIDS can learn or be taught the accepted protocol language in order to recognize unacceptable language. This system is typically installed on the front-end of the web server in the "interface between where HTTPS is un-encrypted and immediately prior to it entering the web presentation layer" so that it can monitor the behavior and state of the communication protocol between the connected device and the web server [2]. This IDS offers a monitoring technique with greater security than that of filtering on IP addresses or port numbers. A disadvantage of this IDS would be that it increases the computing load on the web server.

2.1.2.4 Application Protocol-Based Intrusion Detection System. This category of IDS is commonly referred as a (APIDS). An APIDS focuses on the monitoring and analysis on a single application protocol. APIDS are setup between the process to be monitored and the targeted system(s) analyzing the behavior and state of the protocol. In [30], Wikipedia.com explains an example of APIDS located between an Apache server and a Database Management System (DBMS) monitoring the SQL protocol between the DBMS and the database.

2.1.3 Section Summary. The IDS is a very useful monitoring tool for network analysts. They can be implemented in a variety of locations and configurations within a computer network. This flexibility gives greater perception coverage of the network domain of which the analyst monitors. The IDS has a significant role in developing the analyst's SA. After explaining what the IDS is and how it works, the next section discusses SA.

2.2 Situational Awareness in the Network Domain

As previously mentioned in Chapter 1, SA is knowing and understanding your environment so that you can react appropriately in the near future. Much of the research in Network Situational Awareness, such as in [10, 11, 27] relies on Endsley in [12] defining SA as “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future.” Currently, SA exists solely in the human observer. Presently, no tool is able to produce SA to lessen the cognitive strain on the human observer. However, the human observer uses tools to aid him in developing or comprehending the ever-changing situation. The SA Agent or system should fundamentally function in the domain as described in Figure 2.2.

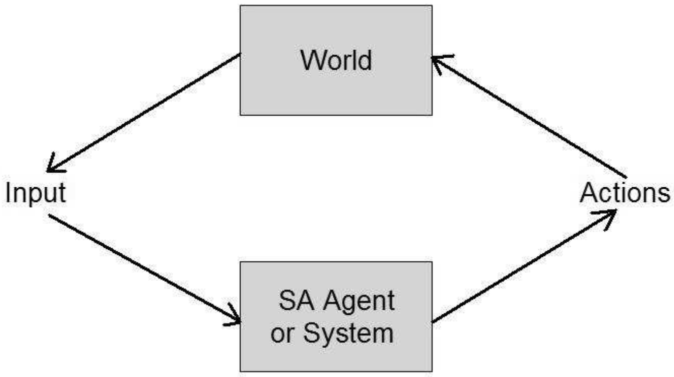


Figure 2.2: Cumiford et al. in [10] display the SA’s role in the feedback loop. In the ntework domain, the inputs could be alert events, network activity logs, other computer resource data and even analyst team members. The analyst processes all the input and makes decisions that create effects in the world. The world responds by generating new inputs and the cycle continues.

Within the SA agent, is a feedback loop termed the OODA Loop. Figure 2.3 illustrates the loop and the SA agent’s goal of making the transitions of each recurring loop quicker and more efficient.

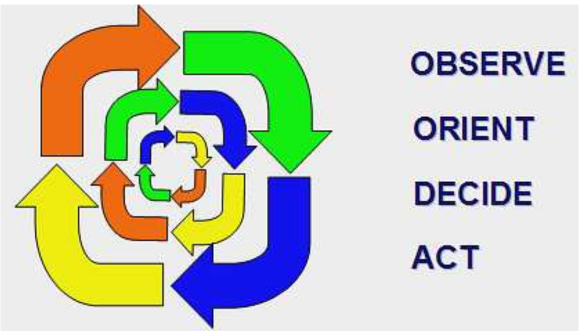


Figure 2.3: Boyd et al. in [7] developed the concept to describe any intelligent system and how it interacts with the rest of the world. The observing and orientation are necessary steps towards developing a situation. Based on the ever-changing situation, decisions and actions are taken. The desired goal of the analyst is to make his or her feedback loop as tight as possible to ensure quicker response. A related goal of the analyst is to have his or her feedback loop tighter than the attacker's feedback loop. The concentric loops illustrate the analyst's OODA loop improving with each revolution. If the analyst achieves this goal then the analyst has a better success rate of recognizing and responding to the attacker's malicious threats. This process should be done as efficiently as possible in order to gain decision and control superiority over adversaries.

Based on Endsley's definition of SA in [12], the following sections will expound on the importance of Perception, Comprehension and Projection in the network domain as it pertains to the model being discussed. In particular, examples will be given to demonstrate the role each plays in the analyst's development of SA.

2.3 Situational Awareness Models

2.3.1 Endsley Model. Figure 2.4 represents the universal SA framework applicable to any domain. The top-down model illustrates the dynamic, cognitive processes. Any component can update any other component that is linked to it. If examined closely, one notices the simple feedback loop described in Figure 2.2 but with added complexity of the concepts; Perception, Comprehension and Projection.

Perception is considered Level 1 SA [12]. It is fundamentally required to be able to perceive the environment. In [22], Salerno relates perception to the basic

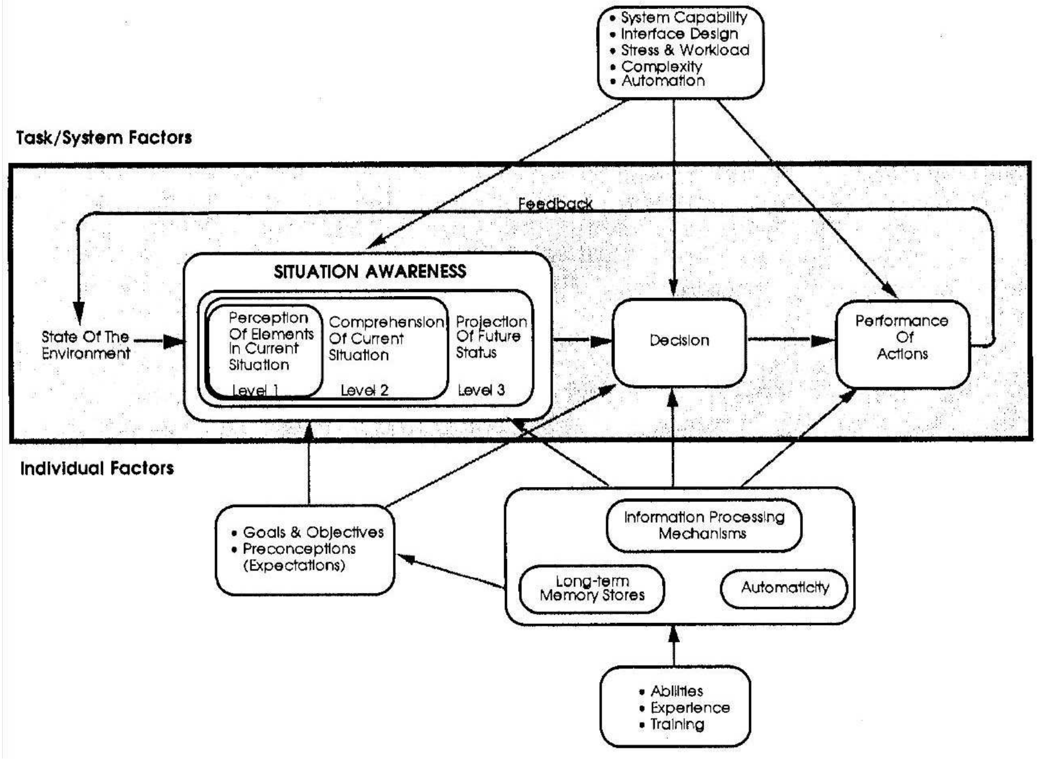


Figure 2.4: Endsley et al. in [12] show the complete SA model in a human perspective. It does not offer any insights into the tools necessary to implement this model in the network domain. The JDL and Hybrid Models are more appropriate for real-world implementation of an SA model.

building blocks of comprehension and projection such that without perception, the probability of developing incorrect SA increases substantially. In the strategic domain, a popular example of inadequate perception leading to faulty comprehension in the 21st century is that Iraq was a serious threat to United States interests because the country had possession of Weapons of Mass Destruction (WMD). In [5], the national security analysts perceived incorrect data leading to lack of understanding of the real threat Iraq posed ultimately leading our decision-makers to declare war on unjustified knowledge.

To perceive the network domain, the analyst utilizes an IDS and other network management devices to monitor activity. IDSs are very useful in this respect but they are still limited in several ways. First, the IDS is a system that only flags data packets that fit one of its signatures or profiles. Most IDSs have no adaptive learning capability but rather a finite list of signatures that must be updated routinely in order to remain effective. Second, the analyst is not guaranteed that the IDS will flag all network attacks because it is impossible to create a finite list of attack signatures that covers the entire network attack space.

To increase perception in larger networks, analysts will employ several IDSs from each classification in order to guarantee wider coverage over the network attack space see Figure 2.5.

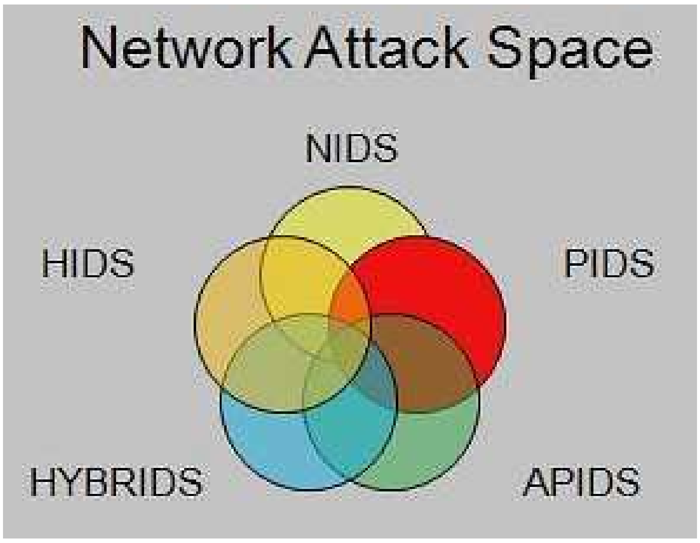


Figure 2.5: The coverage gained in the network attack space by utilizing several IDSs. Overlap that can occur as well. Notice the coverage will never include the entire space.

The strategy has its own disadvantages. First, due to the redundant IDS overlap, the alert event generation rate increases so rapidly it overwhelms the analyst with false-positive and low-priority alert events. The increased alert events can be counterproductive in that they obscure the small subset of high-priority network-warfare threats occurring at that time hindering the analyst's comprehension necessary for situation awareness. The experienced attacker is aware of the analyst's strategy and will employ background scanning and attacking to mask the true intentions which are very precise and small in number. Lastly, given a scenario in which there is a high alert event generation rate and the attacker uses a multi-stage approach to compromising this network, the analyst may notice a single stage but be unable to perceive linking all the stages together in a timely manner thus misunderstanding the situation and missing the small response window to react to the threat.

It is a challenge to find the delicate balance of sensor implementation for adequate perception. Each network is unique in this respect. The analyst should routinely audit the perception sensors for effectiveness because, if not, the analyst can quickly be inundated so much so that he/she loses all awareness. Figure 2.6 illustrates the misconception many analysts have regarding the myth that more sensors is directly proportional to more information.

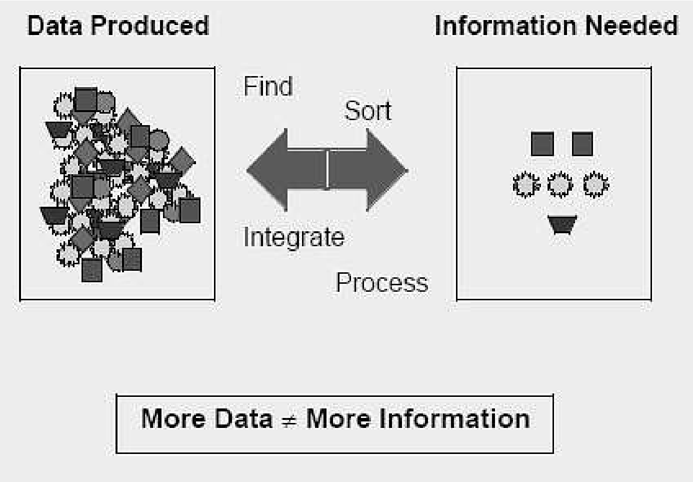


Figure 2.6: Endsley et al. in [12] point out that increased data from sensors at the Level 1 SA does not translate equally into information. At times, the data is redundant and false-positive becoming counter-productive by obscuring the important data.

In Figure 2.4, without the Level 1 perception, there is no Level 2 comprehension. One must experience before finding meaning. Using Figure 2.4, Comprehension is the integration of the current situational awareness with the following:

1. Goals, Objectives and Expectations related to network usage, assets, productivity, efficiency etc.

2. The analyst's abilities, training, a priori knowledge and relevant experience.
3. The analyst's short-term and long-term memory.
4. The network complexity, perception automation, Information Assurance (IA) tool resources.

The Comprehension phase of SA never terminates. The analyst is continuously deriving new meaning from the incoming perceptions attempting to determine operationally relevant information in relation to the current situation. This updating of the network situation must be rapid and seamless because [11] states an analyst "may have as little as 90 seconds to make a decision regarding whether activity is suspicious or not."

The decision-making phase or projection is the Level 3 SA. At this point, the analyst has updated, truthful situational awareness. In [12], Endsley comments that a skilled expert in any field makes timely decisions because he/she relies extensively on the ability to anticipate future events and implications from current events. The skilled operator is also able to make the correct decision and execute it precisely in order to maximize the effect in the domain.

2.3.2 Joint Director of Laboratories Model. Figure 2.7 illustrates the data fusion model. It is a bottom-top model that functions well for the lower level data fusion community but it fails to define SA and Threat Assessment. [23] Levels 0 and 1 perform the perception phase of SA.

Level 0 has physical access to the raw data. It also has the capability for "estimation and prediction of the existence of an object based on pixel signal level data association and characterization [23]. In Level 1, related data is grouped into tracks/containers and each track is given a unique identifier. Together, Levels 0 and 1 perform the duties of the perception phase. The initial levels extract the significant data from the noisy irrelevant data for Level 2, Situation Assessment. Level 2 is the attempt to refresh outdated information concerning the current situation so that,

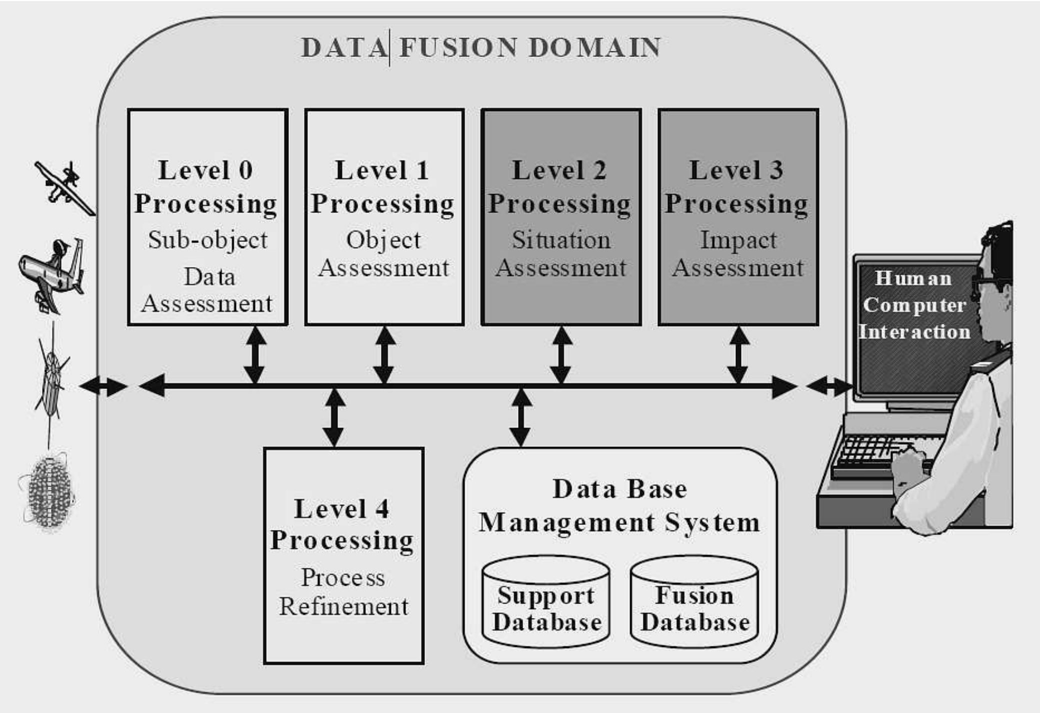


Figure 2.7: Salerno et al. in [23] display the levels of the model and the flow of data from one level to the next. Notice that the data flow is bidirectional meaning any level can progress or regress depending on the Situation Assessment.

Level 3, Impact Assessment accurately predicts the effects of all planned responses. Level 4 is process refinement. This enables feedback between levels in both.

Levels 0 and 1 are the most widely implemented levels of this JDL model. In this research, the fusion process preprocesses the raw alert events into fused data with added metadata and a hierarchial track structure. The SA Reference Model in Figure 2.8 utilizes both the JDL and Endsley models to create a comprehensive Situation Awareness Framework.

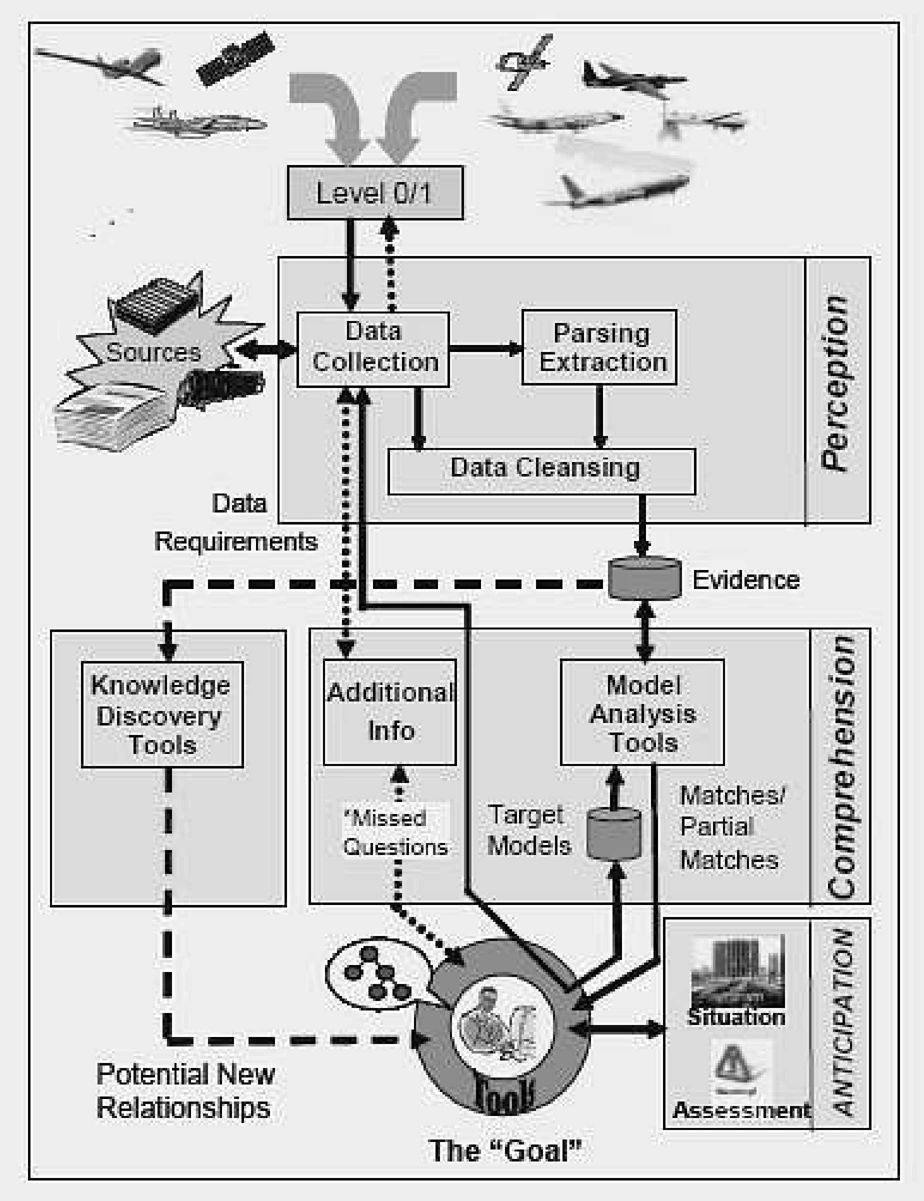


Figure 2.8: Tadda et al. in [27] define an SA model that applies to all domains.

2.3.3 SA Reference Model. Figure 2.8 incorporates both the Endsley and JDL models into a single system. The Data Collection component has the ability and knowledge to gather the raw data. The Levels 0 and 1 provide the interface necessary to collect the raw sensor data and the object events. Once collected, the data is wrapped into a metadata document structure such as Extensible Markup Language (XML). In [23], Salerno et al. explain the metadata captures "various details such as when the information was collected, what source the information came from and the format of the data." In this stage, the parsing and cleansing are necessary steps. Incomplete, redundant or non-essential data should be removed for clarity. The focus and goal of the Perception stage should be data reduction towards presenting important evidence related to the situation to the Comprehension stage.

The Comprehension stage includes the Model Analysis and Knowledge Discovery tools. The Model Analysis component is utilized to determine which portions of a target model currently exist in the evidence. A determination of a target model could be accomplished using a technique such as pattern matching. If evidence of a target model exist beyond a user-defined threshold, this component should provide that model to the analyst. The SA Reference Model defines a real-time system thus the determination of the existence of target models needs to be fast, effective, and efficient. To that end, the Knowledge Discovery component provides the off-line functionality for exhaustive research into learning new models from the evidence. In [23], Salerno et al. state "[t]hese models have the potential to indicate activities, capabilities and group memberships." Any new models would be supplied to the analyst for future usage. The feedback loop connecting the analyst to the Data Collection and Model Analysis Tools exists for important reasons. First, the analyst using prior knowledge, experience or insight may recognize additional, potential models from the current model(s) therefore he/she requires the system capabilities to retrieve such information from the evidence repository and view those models using model analysis. Lastly, the analyst will be an integral part of the system because, as of yet, no successful system can imitate the human's ability to reason and handle incomplete or uncertain information.

Next, the SA develops in the anticipation phase. Any pertinent information to the developing situation will be displayed to the analyst so that he/she can make decisions that are correct and timely. Endsley's projection of actions and JDL's process refinement in Level 4 occur in this stage as well. The analyst is continuously asking questions such as:

1. Is the displayed situation model correct?
2. Was the response appropriate?
3. Were there any key indicators in the situation ahead of time to prevent the situation?

In [23], Salerno et al. illustrates the image of the perfect SA system in any domain in Figure 2.9. This configuration would accelerate the feedback loop described in Figure 2.2 because the analyst is not the main focus of all three phases of SA.

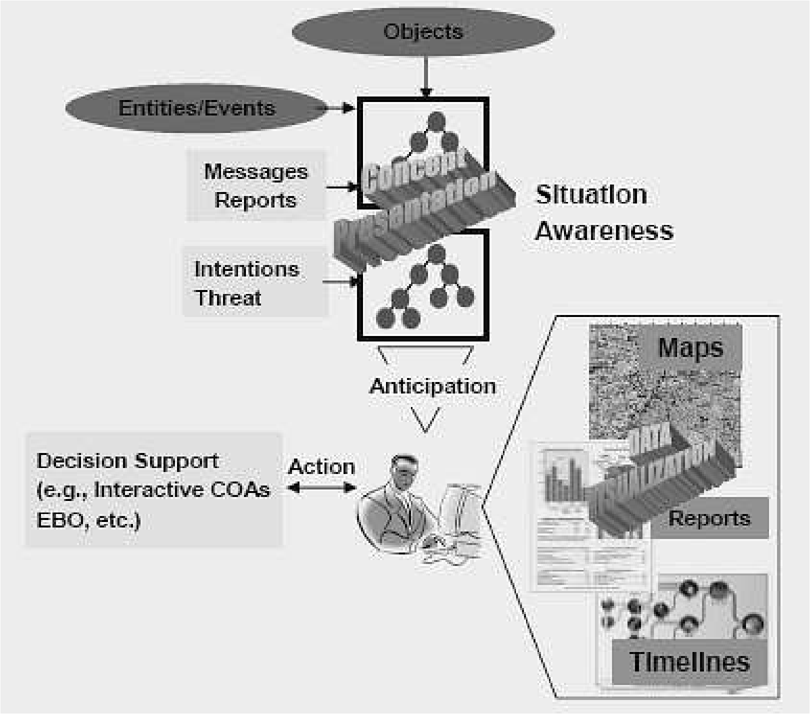


Figure 2.9: Salerno et al. illustrate in [23] that the analyst is supported by a concept presentation engine that is data-driven by a combination of high-level reports, low-level data objects, and entities all of which are associated to threat models. The analyst uses the concepts in conjunction with other knowledge from current world events, maps, reports etc. to make decisions. Take particular note of the analyst's role. He/She is being supported rather than relied on to process raw data, comprehend SA and project decisions. This is more beneficial to the analyst than the current SA environment depicted in Figure 1.1

After explaining the three key models essential to this research, discussion focuses on the benefits of visualizations to display information and current research in IDS visualizations.

2.4 Why Use a Visualization in the Network Domain?

Visualization is a valuable tool for handling sophisticated data in an operational domain. The visualization technique "must be scalable, robust, and effectively and intuitively represent the data and relationships that are relevant to decision making" [13]. In [17, 28], Lakkaraju and Tufte comment that "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." An effective visualization can offload substantial processing by the analyst thus freeing the him/her to think, develop, or manage the new, current or evolving situation. In addition, [11] states that "[V]isual data presentation can facilitate the rapid comprehension of a sequence of interconnected events." The analyst can use a visualization to trace the attacker's path through a network assessing what computer resources may have been compromised through exploitation. This would not be possible for an analyst viewing a monitor of endless alert events from a traditional IDS. In [3], advantages of utilizing a visualization are outlined:

1. It allows more resources to be applied to solving the problem at hand. Resources such as the toolkit discussed in Section 3.2.3 offer significant computing power to establish information relationships to visualize the network data in multiple layouts.
2. It reduces the time spent searching for relevant information.
3. It makes patterns and properties easier to recognize.
4. It makes some things obvious especially problems within the data such as quality control. The analyst using his/her knowledge and experience can resolve

discrepancies in the visualization such as the role of actors based on their alert activities.

5. It facilitates monitoring of multiple events.
6. It provides an alterable medium that is configurable.

The subsections to follow will discuss the important aspects of a visualization that should be considered.

2.4.1 Visualization Classification. The classification of a visualization is based on the ways the analyst uses it. The categories as mentioned in [11] are monitoring, inspecting, exploring, forecasting and communicating. There is no universal visualization that supports every use equally rather each has its strengths and weaknesses in each category. If analysts are responsible for multiple categories, they should expect to use several different visualizations. The visualization categories support Endsley's stages of SA in different ways. Figure 2.10 highlights the stages each category operationally supports.

2.4.1.1 Monitoring. Analysts that use a monitoring visualization are performing real-time analysis at the perception stage. The visualization must update with the new data continuously and be able to present the analyst clear and distinct indicators when activity deviates from established norms. As the deviation from established norms increases, the analyst's attention on the associated actors should increase. The analyst is generally assigned to monitor a specific portion of the network to lessen cognitive strain on the analyst. The focus of the monitoring is dependent on the device being monitored as well as being influenced by the analyst's knowledge and experience. Things that are monitored can include and is not limited to the number of connections for each node in the network, the number and state of the services on each node, the amount of data transferred from an unknown node into the network or vice versa, the overall number of security events from various IDSs in the network.

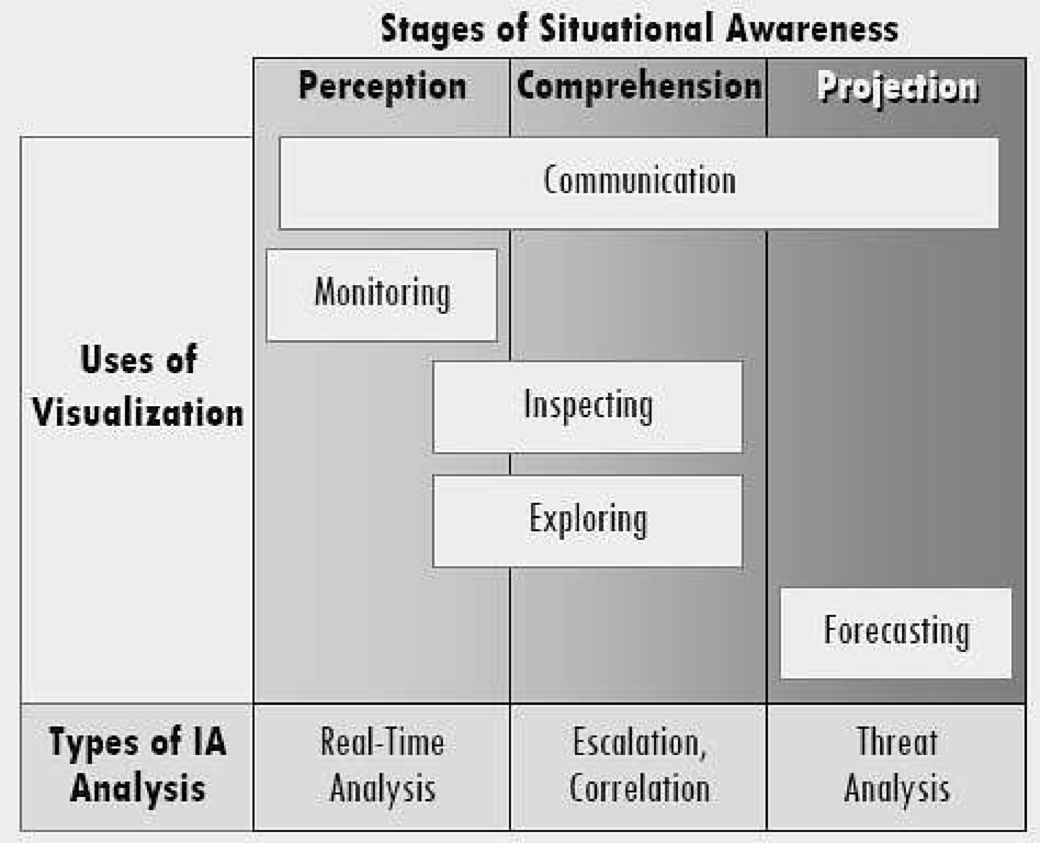


Figure 2.10: In [11], D'Amico et al. illustrate the various classifications are cross-referenced with the stages of SA that they support as well as the type of IA analysis.

2.4.1.2 Inspecting. Inspection visualizations aid the analyst in searching for specific data to test SA hypotheses. This visualization can support both the Perception and Comprehension stages because inspection is closely linked to discovery which leads to an understanding of the situation. Most analysts who are performing Intrusion Detection are inspecting the network in some manner. During this inspection, the analyst is removing nonessential data thereby reducing the dataset in order to focus on important details. Visualizations that support Inspection should be able to display the following as listed in [11]:

1. Many-to-One Connections - many source IPs attempt to connect to one destination IP i.e: Denial of Service.

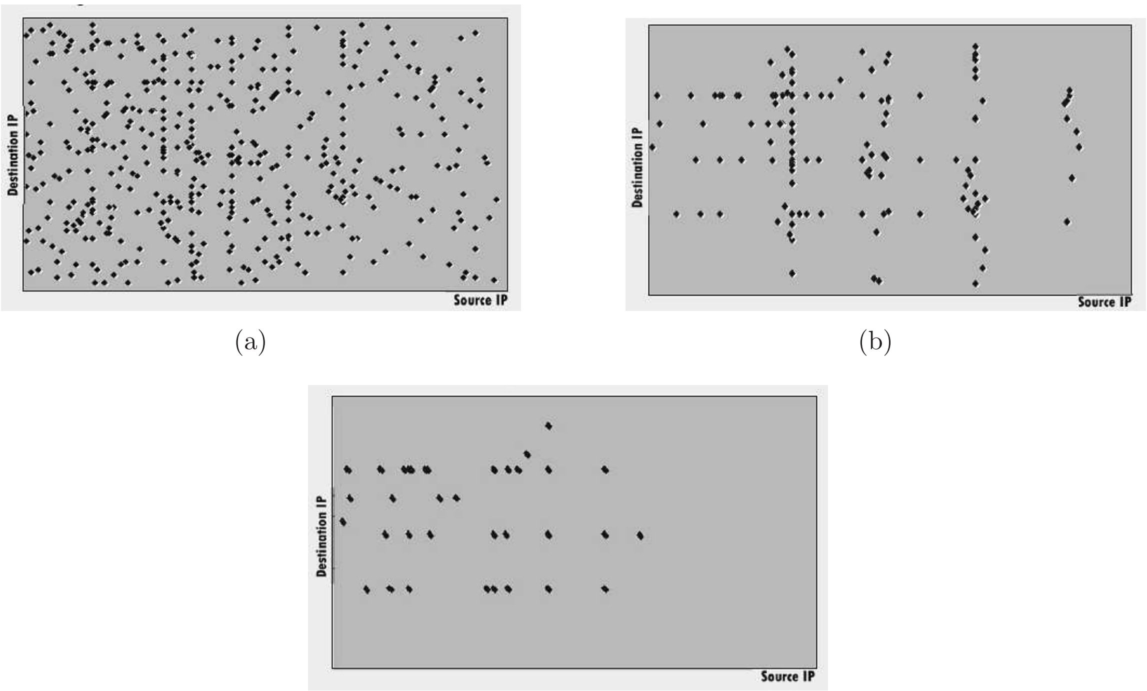
2. One-to-Many Connections - a single source IP attempts to connect to many destination IPs i.e: PortScan.
3. Number of Connections - the number of open connections with an IP. Additionally, the port number of each connection is important to the analyst. An alert generates if the connection is established on a non-standard port.
4. Amount of Data Transferred - The amount of data exceeds a set limit for that IP. The data transfer loads between IPs should be discernable by the analyst.
5. Length of a Connection - time length of a connection between two nodes. Alert generates if the connection time exceeds a set threshold.

2.4.1.3 Exploring. Exploring visualizations support the Perception and Comprehension stages. There is no defined direction or method that the analyst implements to explore the data. The analyst devises new hypotheses without prior knowledge of the data, instead, he/she is seeking new patterns, trends, oddities that would relate to existing SA in the comprehension stage. The analyst is influenced by incoming data and his/her mental model of the cyber domain and the computer resources in that domain. This exploring is not performed during real-time analysis as the analyst has little free-time to do much of anything. Exploration would occur during off-line analysis by correlation analysts for knowledge discovery tools as depicted in Figure 2.8. These analysts tend to observe the network logs spanning several days to a week searching for interesting trends in activity.

2.4.1.4 Forecasting. These visualizations support the Projection stage of SA. Analysts could view a timeline of activity if the current SA progresses without intervention or a number of future states based on the available courses of action the analyst has at that time. The visualization is heavily driven by trending, existing pattern matches and an available model that describes the network. Event sequences that can be overlaid onto the existing visualization can aid the analyst in predicting future actions based on current observations. The sequences can include animations to give the analyst a notion of time, direction, possible victims for an evolving situation.

2.4.1.5 Communicating. Communicating visualizations help the non-analysts, or those not directly knowledgeable of the daily network activity, to make sense of the network analysis that occurred during the situation. These visualizations support the digestible post-analysis IA audits to superiors and subordinates by presenting the review of projected course of actions taken in response to a situation.

2.4.2 Filtering. When developing SA in the network domain, the focus of the visualization should be to minimize the 'noise' within the data. Noise in the network domain is the redundant, false-positive or low-priority alerts. The visualization should be continually minimized until only information remains in view. This minimization processing of the noise is achieved through filtering techniques. Figure 2.11 from [11] illustrates the benefits of filtering non-essential noise from the visualization.



(a)

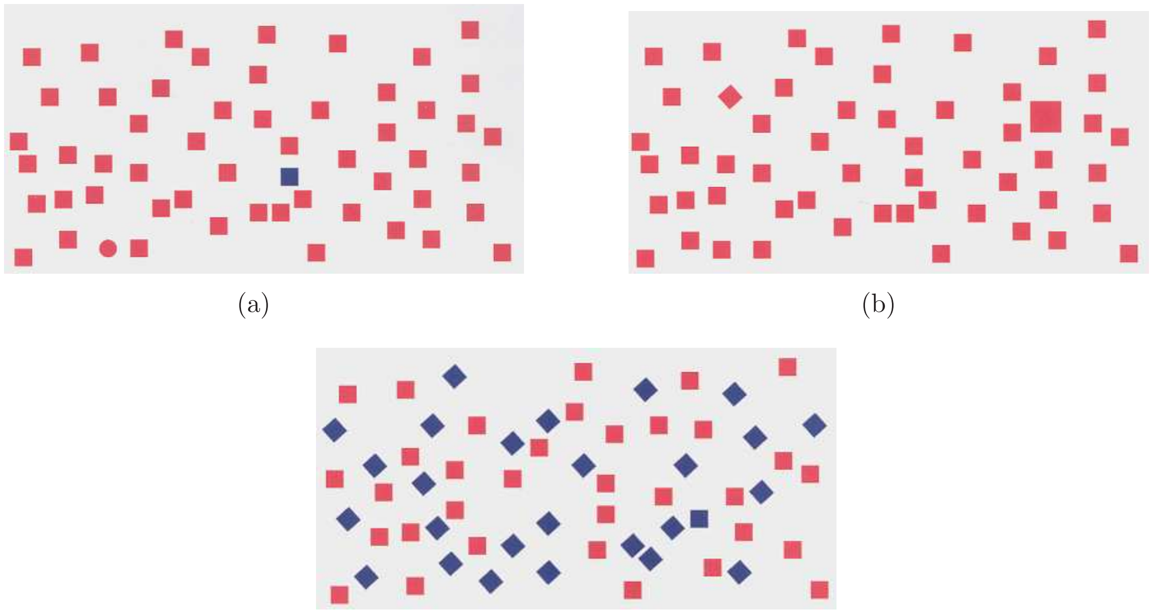
(b)

(c)

Figure 2.11: D'Amico et al. illustrate in [11] the benefits of filtering to reveal patterns within the data what would otherwise remain visually hidden from the analyst. Part(a) shows a noisy graph of the source and destination Internet Protocol (IP) alerts. Part(b) shows the noisy graph filtered of the source and destination alerts where the source IP of the alert received transmitted bytes. Part(c) shows an additional requirement is given to exclude .mil-to-.mil connections.

By using filters, the analyst can locate and focus his/her attention on important information more quickly. Dynamic filtering on a visualization allows the analyst to find patterns in the data that would otherwise be hidden within the myriad of data as seen in Figure 2.11a. Also, the filtering builds SA context because the analyst has further defined the domain space of which he/she is viewing as shown in Figure 2.11c. In this example situation, the analyst gains additional SA by viewing open and active IP connections between friendly and potentially hostile connections.

2.4.3 Preattentive Processing. The visualizations in Figure 2.11 are helpful but are insufficient in a major way. The example visualizations lack preattentive processing as described in Figure 2.12.



(a)

(b)

(c)

Figure 2.12: In [9], Conti et al. define Preattentive Processing as a human's ability to rapidly and accurately analyze visual properties of images at a very low level, without requiring focused attention; it is a fundamental factor in the effectiveness of visualization systems. (a) Preattentive Processing using Shape and Color to distinguish differences in information. (b) Preattentive Processing using size and shape orientation to distinguish differences in information. (c) No Preattentive Processing was used in this visualization. The result is the analyst uses extra cognitive processing to locate the outlier node.

The visual properties for the visualizations in figures 2.12a and 2.12b were selected with greater care than in Figure 2.12c. Figure 2.12c takes significantly more time and cognitive effort to locate the outlier. Preattentive processing can benefit the displaying the network context such as IP address machine-type, friend/foe, alert priority, or time-elapsed since creation. Special attention should be given to deciding on the appropriate visual properties because it can be a real benefit to visualizations that deal with enormous amounts of data. If the toolkit that generated the examples in Figure 2.11 does not incorporate preattentive processing into its visualizations then the analyst will lose valuable time attempting to discover these bits of information by returning to the evidence repository for each IP address. The problem will compound on itself as more data enters the visualization over time or the datasets grow very large.

2.4.4 Focus. Another key factor for visualizations is keeping focus. Focus is defined by [9] as seeking "to display more detail about a user-selected focal point while still maintaining big-picture context." An analyst gains knowledge of important details by focusing on a subarea of the visualization but at a cost of visualization distortion or lost screen space. One method of gaining focus on a subarea of the visualization is to zoom-in. There are some visualizations that display too much information such that the analyst has no recourse but to zoom-in. Immediately after zooming into a subarea, the analyst's mental recollection of the previous visualization begins to fade or be replaced rapidly by the new visualization. An analyst will start to zoom-in and out numerous times to compensate for the context loss. This technique wastes valuable comprehension time. Additionally, zooming into the visualization after two or more iterations becomes completely meaningless. If the analyst loses enough context then he/she can not relate the subarea visualization to the initial visualization.

2.4.5 Speed. There is a requirement for speed in every interactive visualization. The system response must be quick, within 100 milliseconds according to [9], in order to keep the analyst immersed in the visualization. Any tasks such as filtering lasting longer than 10 seconds will cause the analyst to start a new task having lost attention on the task at hand. If the response time exceeds 10 seconds for any reason, a progress bar should be displayed as feedback to the analyst.

The Visual Information Management (VIM) toolkit used in this research was chosen by our AFRL sponsor to visualize the fused alert data. The VIM toolkit allows the user to configure information sources, visualization layouts and properties in order to facilitate the analyst's SA development. To aid the analyst, the VIM toolkit implements a powerful filtering capability that handles structured, hierarchical data across multiple visualizations. The toolkit can filter on any attribute defined within the fused dataset. For example, the toolkit can clean the data of low-priority alerts leaving only the higher priority ones. The toolkit can add and remove filters seamlessly. The visualizations incorporate network context such as IP machine-type and friend/foe using preattentive processing techniques specifically it uses contrasting colors to distinguish friend/foe and differing images for the machine-types. These enhancements lessen the strain on the analyst's cognitive processes which enables the analyst to focus on comprehending the network SA. To keep the analyst's focus, any drill-down into the hierarchial data opens a new dialog box capable of additional drill-down capability if necessary. This is helpful so that the analyst can drill-down into several object entities without losing the knowledge of how those entities interact in the overall situation context. Additionally, the toolkit attempts to render all the entity objects and their relationships in a configuration that utilizes space effectively. If the visualization is larger than the provided window space then the analyst can use scroll bars to navigate the visualization. Furthermore, to allow for focus on individual nodes within the visualization, the nodes selected for focusing are enlarged. This helps the analyst focus on the subarea while not losing the big picture. Lastly, the toolkit responds quickly to the analyst. The toolkit responsiveness is directly proportional to the amount of Random Access Memory (RAM) installed in the operating system.

2.5 Current Research in IDS Visualization

The IA research community has sponsored numerous projects that utilize a variety of network data in an attempt to visualize SA in the network domain. Examples of the network data are IDS alerts, winlog, tcpdump, netflow logs, etc.

2.5.1 VisAlert. In [18], Livnat et al. created a tool that visualizes the correlation of various IDSs logs. In [13], Foresti adds that the "tool facilitates situational awareness in complex network environments by provoiding a holistic view of network security to help detect malicious activities." The tool is capable of visually displaying any past IDS or system alerts. The IA monitoring devices are labeled on the outer ring. The outer ring also functions as a statistical measurement in that percentage of circumference around the outer ring edge is directly proportional to the percentage that a particular alert constitutes in relation to the total number of alerts. The concentric rings represent a user-defined time elapsing for example the outer rings would be older than the most inner ring. Figure 2.13 illustrates the toolkit is operation as it visualizes IDS alerts.

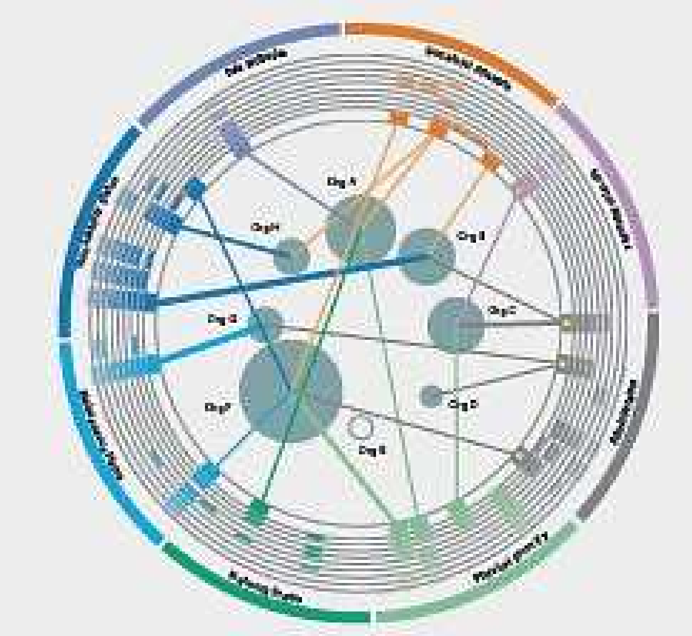


Figure 2.13: In [18], Livnat et al. demonstrate VisAlert reporting a network alert incident.

It does not perform any preprocessing or data cleansing analysis on the alerts or network traffic. As a result, the toolkit's single visualization displays alerts and not information to the analyst. The toolkit's lack of visualization layout diversity "decreases the analyst's ability to create situational awareness and makes a dangerous assertion that a single visualization layout is necessary to display any single or multi-stage attack effectively" [3]. In [11], D'Amico et al. clearly stated that no single visualization can perform as an effective visualization for all three stages of SA. Another potential problem with the toolkit is that the visualization is susceptible to redundant, false-positive, and low-priority alerts cluttering the visualization. The analyst will be mislead or distracted such that the high-priority network threats remain hidden from the analyst. Lastly, the visualization can illustrate single-event network attacks such as DoS on a range of IPs, but it fails to provide the analysis necessary to link a multi-stage attack. A serious limitation of the tool is it must rely on the analyst to bridge the information gap by deciding whether two separate network attacks are related.

2.5.2 PortVis. McPherson et al. in [20] discuss a visualization tool for detecting security events through a port-based method. In [20], McPherson et al. states that the tool uses high-level data for discovering high-level security events. The tool visualizes total count of activities, not substance of the activities thus analysis is limited. The toolkit is beneficial because it is capable of injecting traffic signatures into the network. This offers the analyst the ability to identify suspicious network traffic and develop threat models as it traverses through the network. Unfortunately, [20] offers no examples of high-level network threats the tool can visualize only stating that attackers with small attack signatures would remain clandestine indefinitely. In addition, the tool does not have any drill-down capability to view lower-level details.

2.5.3 VisFlowConnect. In [31], Yin et al. state that the tool was "implemented as a demonstration of an efficient and effective visualization of network flows into and out of a network." The goal was to visualize the relationships, quantity, and direction between internal hosts and external machines. The tool utilizes NetFlow network data [31]. First, [31] claims that the visualization improves situational awareness of current and recent network events through data visualization. The tool visualizes high-level activity, but it has drill-down capability if the analysts requires it. Patterns such as asymmetrical traffic volume between two IP addresses is readily apparent in the visualization. The tool includes a temporal capability in that it can replay past traffic patterns that provides the analyst additional insight into the network environment. The toolkit includes several different views; Global, Domain, Internal and Host Statistics. The toolkit includes advanced filtering techniques on ports, protocol, transfer rates and packet size. Testing of the toolkit against the Blaster virus showed promising results in that the SA required minimal investigation in order to determine the cause of the network attack. The tool generated interesting visualizations with definite patterns when the network attack was DoS. Lastly, the visualization illustrated patterns that depicted hostile reconnaissance such as port scanning. A disadvantage of the tool would be the lack diverse network data integrated into the visualizations. Using additional data sources, the analyst would determine with greater certainty the origin and type of network attack on the network. Figure 2.14 illustrates the tool in operation.

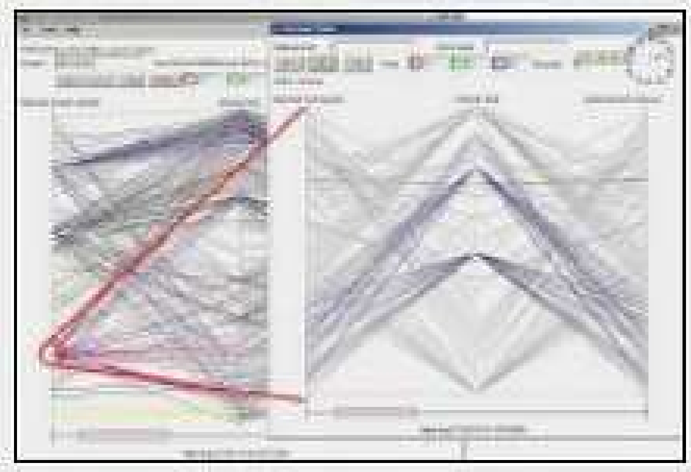


Figure 2.14: In [31], Yin et al. highlight the tool in operation. The visualization appears similar to a parallel-axis graph.

2.5.4 VIAssist. SecureDecisions Inc. in [24] describe the Visual Assistant for Information Assurance Analysis (VIAssist) as aiding the analyst through a software architecture framework that integrates numerous types of visualizations into one toolkit. This research follows the assertions made by [11] that no single visualization can support all the stages of SA. Instead, the framework is attempting to support all three stages of SA by incorporating several different visualizations into a single package. Figure 2.15 illustrates the tool in operation. One can see that there are many tools incorporated into it.

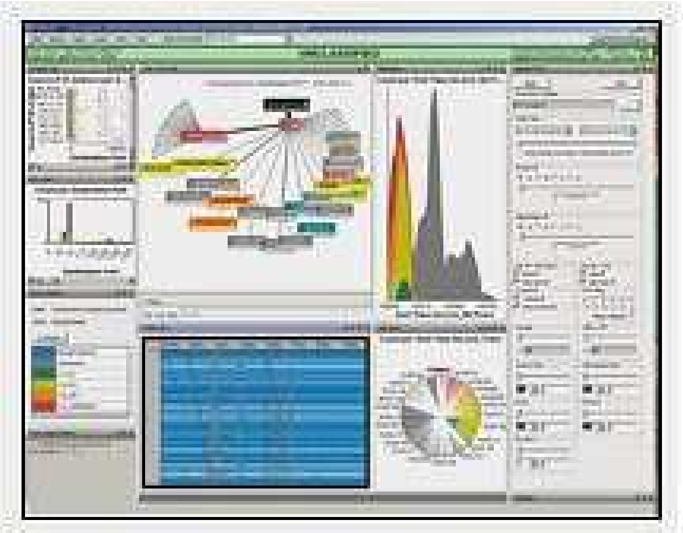
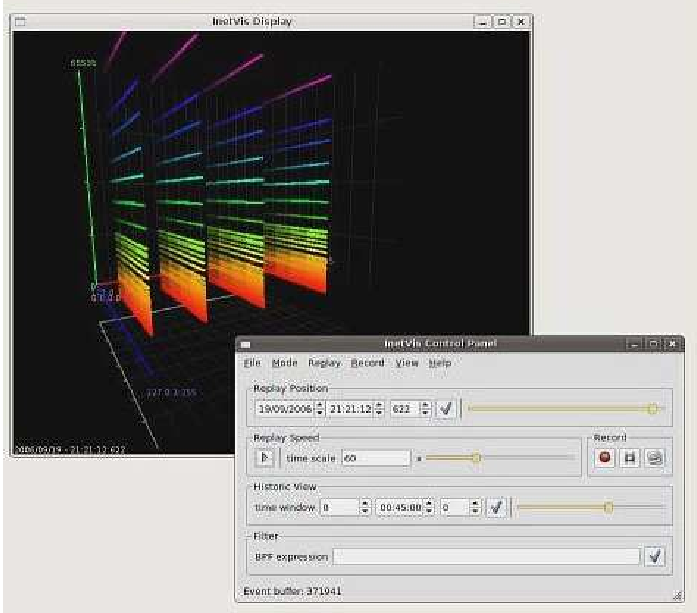


Figure 2.15: In [24], SecureDecisions Inc. highlight the tool's aggregation of various tools.

2.5.5 NVisionIP. Through research funded by Security Incident Fusion Tool (SIFT) at the National Center for Supercomputing Applications (NCSA), [17] developed this toolkit for visualizing network attacks. The tool visualizes network traffic using Argus NetFlow data [17]. The analyst uses network attack attributes to filter the byte-flow data within the visualization. The tool includes a rich set of filtering that gives the analyst the additional control necessary to develop SA. However, the tool's reliance on Netflow data as the single source of network data is limiting. Not all network attacks are captured by Argus Netflow data. Another downside to the toolkit is its reliance on a histogram-like visualization at each level of focus. As discussed earlier in this chapter, SA context is lost with each focus context-switch such as a zoom-in selection.

2.5.6 InetVis. In [21], Riel et al. discuss a 3-D visualization tool that visualizes the network traffic given a source IP, destination IP, and port number. The resulting visualization is a 3-D space with colored points suspended to indicate the data's location. In [21], it is explained "the visualization is intended for viewing events traversing a boundary between an internal (home) network versus the external internet." The toolkit also maps the ICMP packets in the ICMP plane located under the 3-D space. The tool allows the analyst to assign different colors to data variables such as source IP, destination IP, port number, protocol and packet size. The point size can be reduced to help alleviate parts of the 3-D from visualization clutter due to high activity in a specified range. Lastly, an important feature supported by the tool is the playback functionality. This feature allows the analyst to replay any stored data for key cyber attack indicators for future threat model development. To test the tool, NMap was used to generate port scans [21]. The tool visualized the linear port scanning vertical lines showing NMap's attempt to search for vulnerable hosts. Additional testing should be performed on the tool as only scanning techniques have been visualized at this time. Figure 2.16 illustrates the tool in operation.



Replay Position: 19/09/2006 21:21:12 622

time scale: 60 x

time window: 8 00:45:00 0

BPF expression:

Event buffer: 371941

Figure 2.16: In [21], Riel et al. highlight the tool in operation. The analyst controls the visualization through the control panel. The coloring of the background can be modified based on user preference.

III. Research Methodology

This chapter outlines the research methodology used for the design and development of an efficient and effective visualization tool for aiding meaningful visualizations that aid the analyst in developing SA. The problem definition and experimental goals are clearly defined following this introduction. The approach includes specific details of the SA model, the IDS alert data, visualization toolkit and network attack scenarios. The results of testing the visualization toolkit's performance in each network attack scenario is presented in Chapter IV.

3.1 Problem Definition

Computer networks and resources, both military and commercial, continue to grow larger in terms of traffic flow, infrastructure, and available services. While this is advantageous for the user, this trend puts additional strain on the already overworked analysts that monitor the network environment. Traditionally, the analyst uses IDSs to monitor the network traffic attempting to develop SA from the hundreds to thousands of alert events generated each second. This method of network defense and surveillance is quickly becoming unrealistic in practice as the analyst simply can not keep pace with the increasing activity.

3.1.1 Goals and Hypothesis. The goal of this research is to visualize the correct network situation resulting in reduced cognitive strain on the analyst and greater awareness. It is hypothesized that utilizing the Cyber SA Reference Model in Section 3.2.1 can generate trusted SA. If this is true then the toolkit discussed in Section 3.2.3 can accurately present a network situation that allows the analyst to develop SA more effectively than the traditional method of strictly using IDSs.

3.2 Approach

The approach is to implement the necessary components of the Cyber SA Reference Model at the Level 2 of Situation Assessment in the Comprehension stage of SA, see Section 2.3.3, to generate information visualization. Specifically in this research, the VIM toolkit is used visualize the generated fused IDS alert data. The analyst will interact with the visualization(s) to form SA ultimately leading to a decision of whether a network attack has occurred.

3.2.1 Cyber Situational Awareness Reference Model. The Cyber SA Model is the application of the SA Reference Model in the Network Domain, see Figure 3.1.

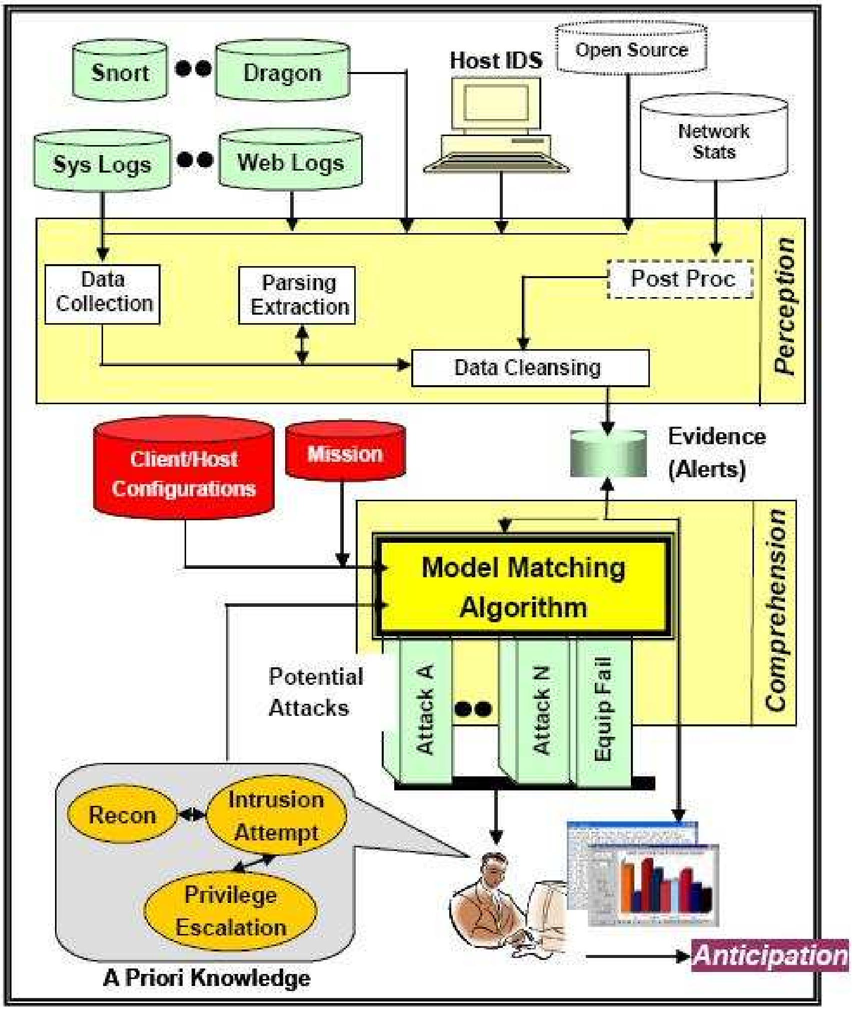


Figure 3.1: In [27], Tadda et al. define the application of the SA Reference Model discussed in Section 2.3.3 applied in the Network Domain.

Multiple IDSs on the network generate the raw sensor data and to send it to data collection repository. From the repository, the data streams into a parsing and data cleansing process that outputs unique, metadata objects that are labeled evidence. This is a necessary step for two reasons. First, IDSs that share a subset of attack signatures will output a similar event alert when triggered. Reducing the data redundancy at this point in the system will ensure the Model Matching Algorithm (MMA) is processing efficiently. Second, the data cleansing is the first opportunity to remove false-positive alert events. The perception stage ends with the queuing of the evidence into a evidence repository accessible by the MMA.

The Comprehension stage begins with the MMA fetching each new evidence and determining the significance of it in relation to the algorithm's threat model, network configuration, and the mission. The decision process by the MMA is continuous until a specific point is reached where the MMA fragments one of the containers. The fracture point occurs when the MMA decides that the current evidence does not support the current network threat stage. After the separation of information, the MMA resumes its normal procedure of fetching, deciding, and placing new evidence in the appropriate container. The separated container of information is sent to the analyst's visualization tool. The tool supports both Situation Assessment, as known as the Level 2 of the JDL model, and the Comprehension stage. The tool will update the visualization appropriately to reflect the new information.

The new information will affect the current SA. The analyst's role is to answer numerous questions:

1. Who caused the effect?
2. Who was impacted?
3. Where did the effect occur?
4. When did the effect occur?
5. How did the effect occur?
6. Is the effect related to past effects?
7. What was the effect?

The analyst is cognitively thinking and answering the above questions while searching for single and multiple stage attacks. The analyst is using the current SA combined with a priori network knowledge of evolving network attacks, education, experiences and known vulnerable targets. In this research, the analyst uses the tool to find truth in the SA by filtering out the noise. The filtering will validate any theories the analyst has about the SA. Any valid theories will motivate the analyst to react appropriately and anticipate.

3.2.2 Fused IDS Alert Data. The fused alert data is the evidence collected during the network attack scenarios discussed in Section 3.2.4. The network attack scenarios were run against a class B mock-network. Network structure details are in Appendix A. IP addresses were changed but the network structure remained unaltered. The alert events were parsed, cleaned and given metadata. This is an essential step to remove alert event redundancy, false-positives and low priority alerts. The formatting of the alert data into metadata alert objects allows the MMA to handle a heterogeneous source of data seamlessly and more efficiently. In this research, the IDS alert sources were Snort, Dragon, IIS, and Apache [15, 16]. Figure 3.2 describes the hierarchial structure of the fused alert dataset in Unified Modeling Language (UML).

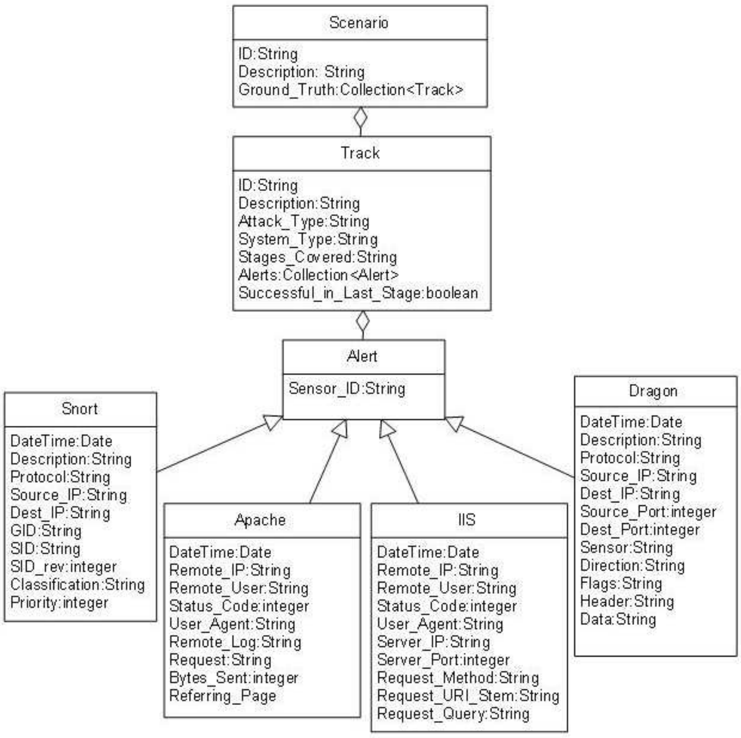


Figure 3.2: The class structure of the fused alert data. Each attribute represents a metadata variable with exception of the attribute collections. The collections represent the alert metadata objects that exist within the current metadata objects i.e: a track object has many heterogeneous alert metadata objects.

The MMA implements the flow diagram illustrated in the Anatomy of a Hack in [19]. With each piece of evidence, the MMA decides which methodology should contain that evidence based on the evidence itself, the existing evidence in all of the methodologies and how that evidence relates to the host/network configuration and network mission. The MMA has the ability to remove, add and combine methodologies as the situation changes. A removal of methodology causes the evidence to be aggregated into a track metadata object and to be sent to the toolkit for visualization.

3.2.3 Visualization Toolkit. This research directly applies to the visualization of Level 2, Situation Assessment, in the JDL model in Section 2.3.2. The level 2 is incorporated in the comprehension stage of the Cyber SA Reference model in Section 3.2.1. The tool helps the analyst in the comprehension stage by offloading considerable cognitive processing through effective visualizations that incorporate additional network context not available in traditional IDS alerts such as friend/foe, IP machine-type or tcpdump data. The analyst can apply his/her network knowledge through the capabilities of the toolkit in order to gain trusted SA.

More specifically, the VIM toolkit was designed by Stanfield Systems, Inc. In [25], Stanfield Systems, Inc. lists the following capabilities that aid the analyst in developing SA in the network domain:

1. Select information sources
2. Apply transformation operations
3. Choose a visual layout
4. Specify relationships between information entities
5. Map information values to graphical attributes

3.2.3.1 System Architecture. Figure 3.3 shows the VIM architecture separated into 3 major sections; Enterprise, Transformation and Visualization. To incorporate a new network data source into the toolkit, a new adaptor is required or modification of an existing adaptor is required. The Enterprise layer handles the data parsing from the fused alert data into the enterprise object model. The transformation layer handles the data operations that the analyst uses to modify the information sources in the layouts. For example, the merge operation transforms two information sources into a single source using a common attribute. After the information source is loaded into the layout, the analyst will use the filter operator to transform the information source according to the filter criteria. The analyst interacts with the Visualization layer when he or she selects a layout or changes the layout. The visualization layer supports the focusing mechanism when the analyst cursors around the visualization.

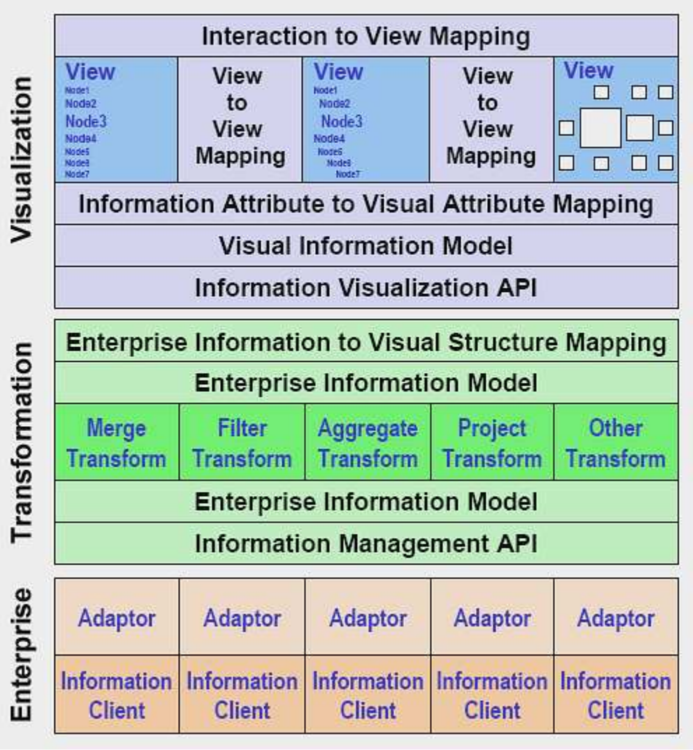


Figure 3.3: Stanfield Systems, Inc. illustrate the Visualization Information Management toolkit's system architecture [26].

3.2.3.2 Data Model. A data model was necessary in order to map the XML metadata of the fused alerts to the enterprise objects required by the toolkit at the adapter level in the system architecture. The data model resembles Figure 3.2. Additionally, the model captured the relationships between the XML objects. During the processing of the XML, the adaptor also builds a network graph representative of the alert activity between the IP addresses. The network graph information allows the analyst to use the graphical information layouts provided by the toolkit.

Early in the research, the tcpdump was the first adaptor created. This adaptor processed the raw tcpdump data of the network attack scenario to visualize the network traffic to and from each IP. The second adaptor was the fused alert adaptor as it was more complicated.

Having multiple data models each with an adaptor allows the analyst to bridge more relationships between disparate network data that can not be accomplished by traditional methods. Future research can be done in the area of implementing more adaptors to see if additional SA is gained.

3.2.3.3 Database. Early into research and development of the toolkit, a requirement for a database solution became apparent. With larger datasets spanning several hours, the toolkit's performance noticeably reduced especially when the analyst used multiple visualizations with other network data such as tcpdump. The data model(s) became so large that the toolkit could not process the analyst's inquiries into the visualization within a desired response time. To reduce the toolkit's reliance on the operating system's RAM, this research developed a MySQL database to store the data models of the tcpdump, fused alert data, and the IP network context data later in development. The rationale for this implementation change is that the analyst can choose specific data within all the network data. The change gives greater control to the analyst by allowing him/her to decide what data, how much of it to load, and having all the data centrally located facilitated usability concerns with multiple data sources. The loading of information sources became less tedious as the analyst configures the toolkit to open a single connection to a database rather than opening a flat file for each information source. The implementation did reduce the toolkit's reliance on main memory to store the data model(s).

Another benefit of the database is related information about the computer resources and network can be stored in the database. The related information is analyst-based meaning the analyst created this information from his/her knowledge and experience on the job and stored this in the database. The analyst-based information can be merged into the other network data to enhance them. Specifically, we created an IP address table that stored information such as friend/foe and machine-type. Other important data such as asset importance could be stored in the database that could help the analyst focus on certain IP addresses more so than others. Additionally, each scenario was stored in its own database so any data modified during the scenario would not effect the behavior or results of other scenarios.

3.2.4 Network Attack Scenarios. We executed the network attack scenarios on a simulated network, see Appendix A, very similar to the class B network corresponding to the Air Force Research Laboratory (AFRL) in Rome, NY. The network included typical activity of a military network such as applications, services and scans.

3.2.4.1 Basic CGI Overflow. This attack represents a buffer overflow exploit against a known script on an Apache web server, www.bprd.osis.gov. The exploited script in this scenario is called petition. The petition script allows people to sign their names in support of some arbitrary statement. The person's signature is stored in a MySQL database. When the cgi is exploited, the tables within the MySQL database are queried.

3.2.4.2 CGI Attack with Chaff. This attack uses the same target and exploit as in Section 3.2.4.1 but the attacker sends sporadic requests from different IPs that produce the same Snort signature as the attack in Section 3.2.4.1.

3.2.4.3 Data Exfiltration. In this attack, the attacker sends a seemingly official-looking email to several inside users. The email contains a word document with embedded macros. When the user opens the word document, the macro runs. The macro locates all of the folders that have files in Word's "Recent Files" list then uploads the folder's contents to a remote ftp server.

3.2.4.4 Phishing with Plug and Play Exploit. In this attack, the attacker sets up a website offering the visitor free "porn" if they sign up. The user is allowed to choose their own username and password. When a user from OSIS visits this site he supplies the same username and password that he uses to login to his machine in the OSIS network.

The attacker sees that she has that information and tries to ssh to the same IP as the request that supplied the username and password, using that same username and password. Once logged in the attacker downloads a Windows exploit from her ftp server. The exploit is a Plug and Play (PNP) exploit that gives the attacker a command shell on a remote Windows machine without needing any login information. Once on the Windows host, the attacker located the user's home folder (under "Documents and Settings") and ftp's all the files located there to a remote server.

3.2.4.5 Firewall Misconfiguration. In this scenario, the system administrator is performing routine system maintenance on the network firewall and accidentally brings down the firewall. After a few minutes, the administrator realizes the error and restores the firewall back to its original settings. No network attack occurs at this time, but background attackers, that are normally unsuccessful, gain access to the IP enclaves inside the network that are supposedly protected by a firewall.

3.3 Test Suite Specifications

Below is a table that describes the necessary requirements of a system in order to implement the visualization solution proposed in this research.

Table 3.1: The hardware and software test environment used to run the toolkit during experiments.

Component	**Value**
Computer Make & Model	Dell Latitude D820 laptop computer
Processor Type	Intel® Core™ Duo processor
Processor Clock Speed	2.16 GHz
RAM Size	4096 MB
Operating System	Windows XP Professional
Integrated Development Environment	Eclipse 3.3.0
Java Compiler	jre1.6.0
	virtual machine arguments: -xmx1024M
Other System Requirements	WinPcap 4.0.1 and Jpcap 0.7
Additional plugins	JAVA-HELP
	MySQL JDBC support plugin
	Fused Alert Dataset adapter plugin
Database	MySQL Server 5.0

3.4 Experimental Design

We tested the methodology in the following way. The researcher, acting as the analyst using the toolkit, determined and outlined the precise steps required to build a visualization that aids the analyst is building SA of the network attack scenario in Section 3.2.4. Initially, the research was tasked with only visualizing the fused alert data to the analyst. However, the data lacked the network context in order to aid the analyst in developing SA from the visualization. Therefore other types of network data were included, such as tcpdump and network host/configuration data, in order to fill-in the information gaps evident in the fused alert data. Each scenario was researched using the additional data to create visualizations that aided the analyst in developing valuable SA.

3.5 Implementation

In order to achieve the goal of an analyst developing SA through the toolkit, it had to be enhanced in several areas for it to be functional in this domain.

3.5.1 Software Engineering the Toolkit. In order to visualize the network situation from the fused alert data, the toolkit was engineered in the following ways. Initially, the toolkit required network data files as input, specifically the fused alert and tcpdump. For each type of network data, an adaptor is necessary in order to parse and create the necessary objects and object relationships in the toolkit. The adaptor is a key component of the toolkit. For example, the fused alert adaptor stores the relationships within the XML metadata and establishes new relationships within the data such as IP address associations. Multiple adaptors allows the analyst to aggregate and visualize different data simultaneously to develop a finer picture of the network activity.

Initially, the toolkit lacked the ability to link multiple visualizations together, but now the analyst can link multiple visualizations together through a shared data attribute. The data attribute most likely to be shared through all network data is the IP address. We found it necessary at least in the fused alert data that a source and destination IP address is required. The visualizations can be linked bidirectionally thus the analyst can highlight any object within the visualizations and the corresponding information in the other linked visualization will highlight.

Next, the toolkit filter was enhanced in several ways. First, the filter was modified to filter on any attribute within the fused alert data. This allows the analyst more variability to manipulate the fused alert data even further for clarity. Second, the toolkit executes an analyst's filter request to all component-linked visualizations. This enhancement provides the analyst with visualizations that include similarly-related information. Before this enhancement, the filter would reduce the current visualization and none of the others. The difference in data presentation was distracting to the analyst. Lastly, the filter was enhanced to include pattern recognition operators to allow the analyst flexibility on filtering based on "key words" rather than specific strings. For example, an analyst using the filter to search and match for specific terms within an alert description versus matching the entire string.

Next, the toolkit was enhanced to handle the notion of time. The fused alert data is parsed to create a linear timeline of alert creation. This is the foundation for track timeline in future research. The analyst is able to set a time constraint to further refine the data that is visualized.

Lastly, the analyst can assign different object shapes, colors, and borders based on the object data attributes. For example, the analyst can assign squares and circles to friendly IP addresses and enemy IP addresses respectively. The analyst may decide to give certain colors to IP addresses or change the object border type of an IP address based on IP address activity. This capability allows the analyst to decide how the visualization should appear. An added benefit is that the analyst is more likely to process information from the visualization faster because he/she designed the color/shape legend for the visualization. The analyst isn't interpreting another person's visualization but rather his/her own visualization.

3.6 Summary

We specified an experimental methodology to determine if the VIM Toolkit can benefit the analyst in developing sufficient SA in order to determine if a network attack is occurring, the network attack progression and possibly label the type of attack. The research would like to see the analyst gain benefits that include reduced cognitive strain in the analyst, quicker response times to valid network attacks and more efficient SA development within the analyst compared to the traditional method of monitoring using IDSs. This chapter focused on the important factors related to the methodology of the research. The next chapter presents the results of the methodology.

IV. Network Attack Scenario Analysis

This chapter presents the operational steps an analyst performs in order to gain SA of the computer network. The goal is to create visualizations that aid the analyst in developing SA. In regards to the network attack scenarios in this research, the goal would be for the analyst to gain insight into the role of the attacker(s), the victim(s) and any other actors in the domain. Another key indicator that the analyst has SA of the computer network is if the analyst can understand the story of the network attack.

This chapter contains step-by-step figures that show the analyst actively participating in a knowledge building exercise. The toolkit, without the aid of the analyst, does not build SA. The analyst uses his/her knowledge, experience, and insight to link and filter multiple visualizations until the underlining, key information is revealed that, up to this point, has been obscured by abundant extraneous data.

The main scenario detailed in this research will be Phishing with Plug and Play because it represents a multi-stage attack with multiple attackers and victims. Additionally, the scenario represents the largest fused alert dataset available in the research. However, there are several visualizations from other scenarios, such as the Firewall Misconfiguration in Section 3.2.4.5, that will be highlighted to show SA of the network. In particular, the analyst using the toolkit can determine if network defenses are performing appropriately.

It is expected that the selected network attack scenario will be outlined in detail starting with the loaded data and ending when the analyst has developed sufficient SA from the scenario. The loading steps are found in the appendix and are not important in this chapter.

4.1 Phishing with Plug and Play Exploit

The Figure 4.1 shows the beginning state of the toolkit. The analyst has connected the toolkit to the database.

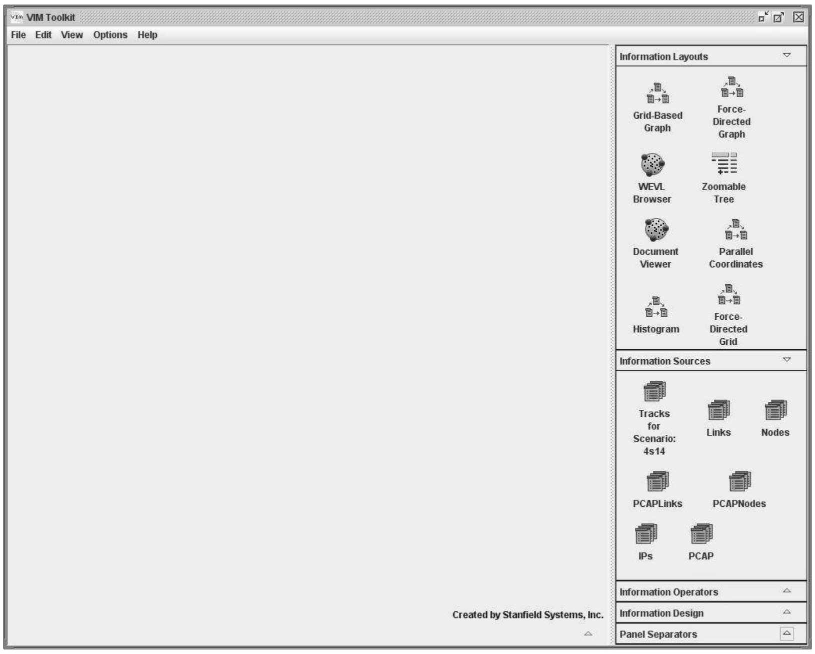
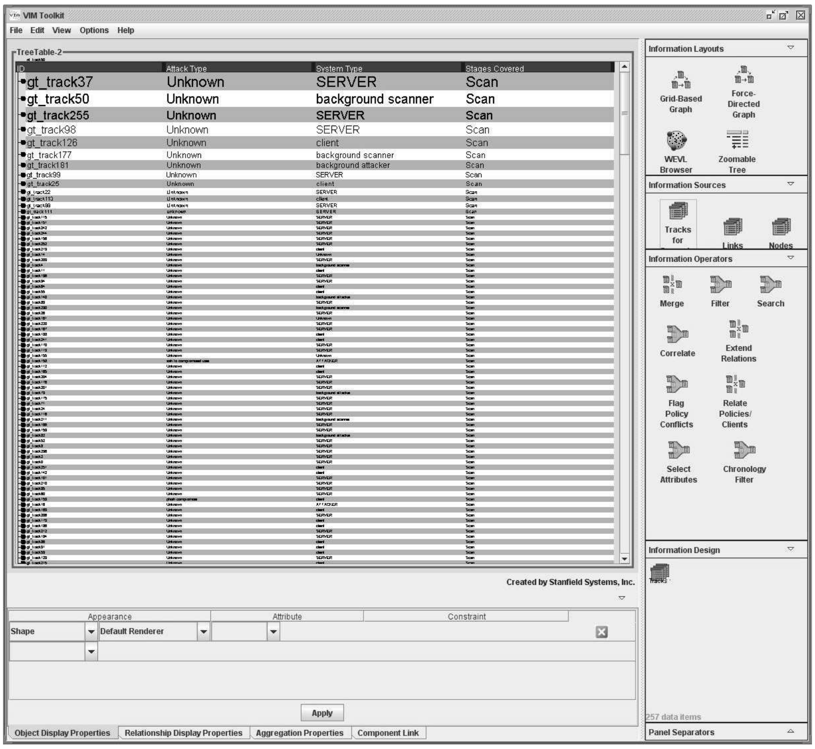


Figure 4.1: The toolkit ready for the analyst to begin building creating visualizations. The toolkit has already connected to the database as evidence that the Information Sources box is populated.

This network attack scenario requires the use of two information layouts; Treetable and Force-Directed Graph. The analyst loads the scenario's fused alert data into a treetable information layout. Figure 4.2 shows the fused alert data loaded into the treetable layout. The analyst does notice a pause in the toolkit as it works to process the 257 tracks. Through proper visualization linking and filtering, the amount of data will dramatically shrink and toolkit responsiveness will increase. At the conclusion of the filtering process, there are 3 tracks associated with the actors involved in this network attack.

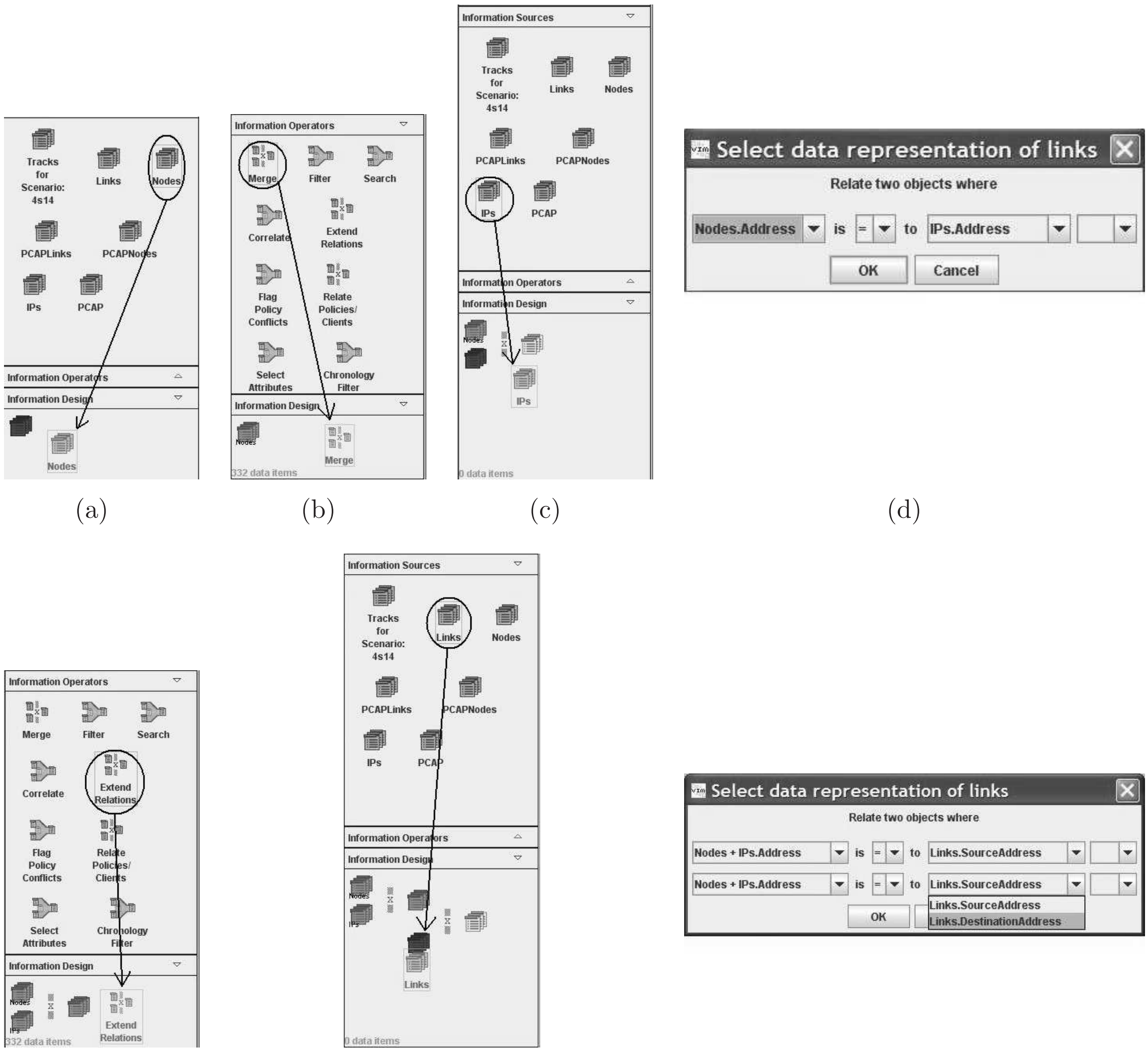
All of the available information sources for visualization are located in the information sources box. It is not a requirement that the analyst utilize every information source, but it should be noted that it is likely that the analyst will gain better understanding with access to more, unique sets of information from his/her environment.



ID	Attack Type	System Type	Stages Covered
gt_track37	Unknown	SERVER	Scan
gt_track50	Unknown	background scanner	Scan
gt_track255	Unknown	SERVER	Scan
gt_track98	Unknown	SERVER	Scan
gt_track126	Unknown	client	Scan
gt_track177	Unknown	background scanner	Scan
gt_track181	Unknown	background attacker	Scan
gt_track99	Unknown	SERVER	Scan
gt_track25	Unknown	client	Scan
gt_track22	Unknown	SERVER	Scan
gt_track113	Unknown	client	Scan
gt_track88	Unknown	SERVER	Scan
gt_track111	unknown	SERVER	Scan

Figure 4.2: The analyst has loaded the fused alert data into the treetable layout. The data that is hierarchical in structure is best represented in this layout.

Next, the analyst creates a second visualization using the force-directed graph information layout. The analyst drags a panel separator into the visualization space to split the space for a second layout. The analyst drags the force-directed graph layout from the information layouts box into the empty visualization space. The selection of which data to load into the force-directed layout requires some understanding of the preprocessing procedures that occurred during the initial loading from the database. The fused alert data was parsed to create two data sources, Nodes and Links, in order to create a proper graph structure. The Nodes collection represents the IP addresses and the Links collection represents the alerts shared between a pair of IP addresses. Figure 4.3 is the progression of choices the analyst makes when defining the force-directed graph information layout necessary to create the foundation of the visualization.



Select data representation of links

Relate two objects where

Nodes.Address is = to IPs.Address

Nodes + IPs.Address is = to Links.SourceAddress

Nodes + IPs.Address is = to Links.SourceAddress

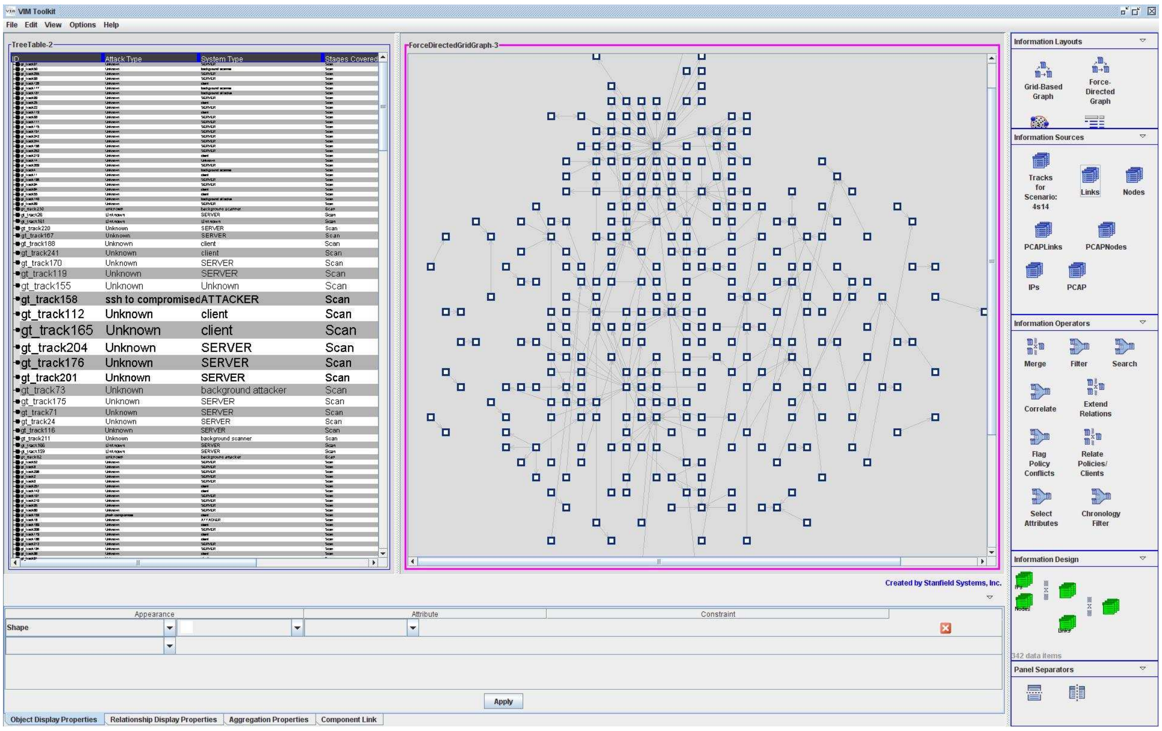
Links.SourceAddress
Links.DestinationAddress

(a) (b) (c) (d)

(e) (f) (g)

Figure 4.3: The information source of the force directed graph must be hand-built by the analyst. The Nodes collection in Figure 4.3a was merged with an outside network data source. Specifically, the IPs collection contains IP address information such as friend or foe. The merging of multiple data sources enhances the analyst's choices when visually defining the objects in the visualization as seen in later figures. Figures 4.3d and 4.3g represent required actions from choosing operations merge and extend relations in Figures 4.3c and 4.3f respectively.

The resulting visualizations are shown in figure 4.4. The next figures outline the process of linking the visualizations together, coloring the force-directed graph and filtering the visualizations properly. The linking and coloring steps can occur in

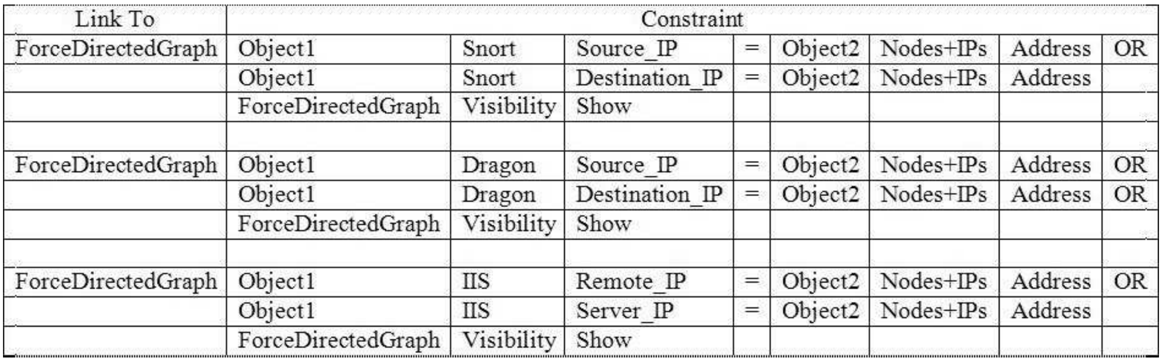


ID	Attack Type	System Type	Stages Covered
gt_track230	Unknown	background scanner	Scan
gt_track26	Unknown	SERVER	Scan
gt_track161	Unknown	Unknown	Scan
gt_track220	Unknown	SERVER	Scan
gt_track167	Unknown	SERVER	Scan
gt_track188	Unknown	client	Scan
gt_track241	Unknown	client	Scan
gt_track170	Unknown	SERVER	Scan
gt_track119	Unknown	SERVER	Scan
gt_track155	Unknown	Unknown	Scan
gt_track158	ssh to compromised	ATTACKER	Scan
gt_track112	Unknown	client	Scan
gt_track165	Unknown	client	Scan
gt_track204	Unknown	SERVER	Scan
gt_track176	Unknown	SERVER	Scan
gt_track201	Unknown	SERVER	Scan
gt_track73	Unknown	background attacker	Scan
gt_track175	Unknown	SERVER	Scan
gt_track71	Unknown	SERVER	Scan
gt_track24	Unknown	SERVER	Scan
gt_track116	Unknown	SERVER	Scan
gt_track211	Unknown	background scanner	Scan
gt_track166	Unknown	SERVER	Scan
gt_track159	Unknown	SERVER	Scan
gt_track82	Unknown	background attacker	Scan

Figure 4.4: In the force-directed graph, the rectangular objects are the IP addresses and the links connecting the IP addresses are the IDS alerts. A link between two IP addresses can represent multiple alerts in the Fused IDS data therefore it is very important that the two visualizations be linked together to display that relationship.

any order. It is left to the analyst's discretion which to do first. In this research, the linking stage was done first because it was a fixed process. There is no deviation from the mechanism that links the treetable visualization to the force-directed graph and vice versa.

4.1.1 Component Linking the Visualizations. Figure 4.5 shows to the required options for the linking of the treetable visualization to the force-directed visualization. The analyst selects the treetable visualization and tabs to the Component linking box at the bottom of the toolkit.

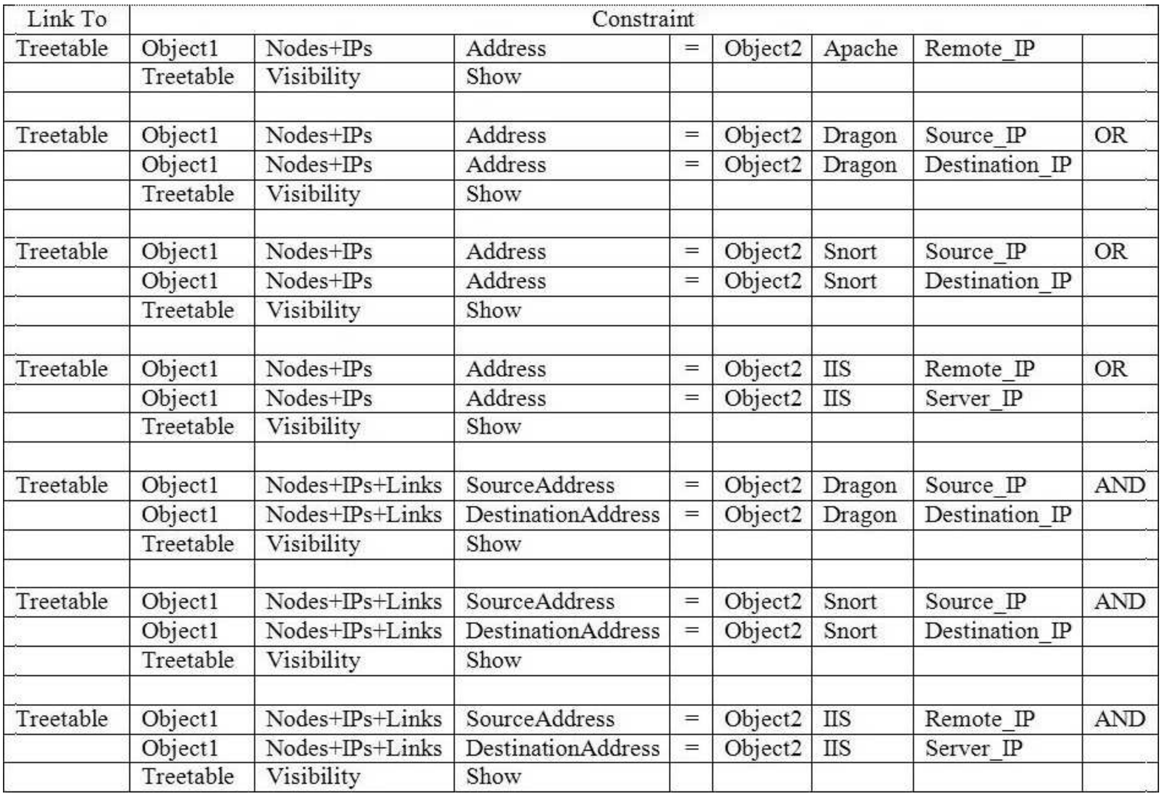


Link To	Constraint							
ForceDirectedGraph	Object1	Snort	Source_IP	=	Object2	Nodes+IPs	Address	OR
	Object1	Snort	Destination_IP	=	Object2	Nodes+IPs	Address	
	ForceDirectedGraph	Visibility	Show					
ForceDirectedGraph	Object1	Dragon	Source_IP	=	Object2	Nodes+IPs	Address	OR
	Object1	Dragon	Destination_IP	=	Object2	Nodes+IPs	Address	OR
	ForceDirectedGraph	Visibility	Show					
ForceDirectedGraph	Object1	IIS	Remote_IP	=	Object2	Nodes+IPs	Address	OR
	Object1	IIS	Server_IP	=	Object2	Nodes+IPs	Address	
	ForceDirectedGraph	Visibility	Show					

Figure 4.5: In the component linking workspace, the analyst selects the visualization he/she wants to link the currently selected visualization to. In this case, the force-directed graph visualization is selected. Next, the analyst associates each alert type source and destination IP address data attribute to the address attribute in the force-directed visualization. After each alert type is specified, the analyst clicks the apply button. Note, the Apache IDS alerts do not specify a server IP address therefore rather than generalize where these alerts came from and possibly skew results, this research chose to leave this data unspecified in the linking process.

Next, the toolkit allows the analyst bi-directional linking of visualizations. The analyst may find key evidence in the force-directed graph visualization and having the two visualizations linked allows the analyst to investigate which tracks are associated with the evidence in the force-directed graph visualization. Having applied the linking from the treetable visualization to the force-directed graph visualization, the analyst should focus on the force-directed graph visualization and click the component linking tab at the bottom of the toolkit in order to link in the opposite direction. Figure 4.6 shows the required options for the linking of the force-directed graph visualization to the treetable visualization.

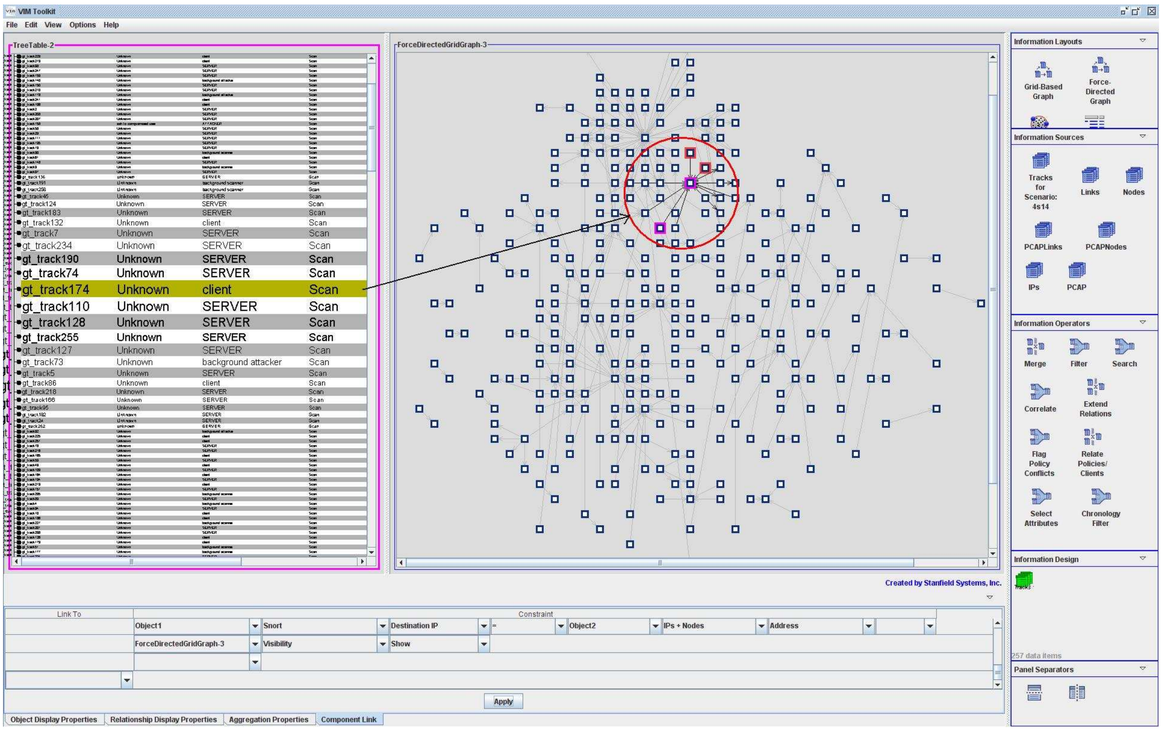
After applying the component linking options in bi-directionally, it is important to test that in fact both visualizations are linked. Simply, select some objects in both visualizations and look to see if the other visualization highlighted any data. At this step, the visualizations still are large requiring the analyst to search the visualization



Link To	Constraint							
Treetable	Object1	Nodes+IPs	Address	=	Object2	Apache	Remote_IP	
	Treetable	Visibility	Show					
Treetable	Object1	Nodes+IPs	Address	=	Object2	Dragon	Source_IP	OR
	Object1	Nodes+IPs	Address	=	Object2	Dragon	Destination_IP	
	Treetable	Visibility	Show					
Treetable	Object1	Nodes+IPs	Address	=	Object2	Snort	Source_IP	OR
	Object1	Nodes+IPs	Address	=	Object2	Snort	Destination_IP	
	Treetable	Visibility	Show					
Treetable	Object1	Nodes+IPs	Address	=	Object2	IIS	Remote_IP	OR
	Object1	Nodes+IPs	Address	=	Object2	IIS	Server_IP	
	Treetable	Visibility	Show					
Treetable	Object1	Nodes+IPs+Links	SourceAddress	=	Object2	Dragon	Source_IP	AND
	Object1	Nodes+IPs+Links	DestinationAddress	=	Object2	Dragon	Destination_IP	
	Treetable	Visibility	Show					
Treetable	Object1	Nodes+IPs+Links	SourceAddress	=	Object2	Snort	Source_IP	AND
	Object1	Nodes+IPs+Links	DestinationAddress	=	Object2	Snort	Destination_IP	
	Treetable	Visibility	Show					
Treetable	Object1	Nodes+IPs+Links	SourceAddress	=	Object2	IIS	Remote_IP	AND
	Object1	Nodes+IPs+Links	DestinationAddress	=	Object2	IIS	Server_IP	
	Treetable	Visibility	Show					

Figure 4.6: In the component linking workspace, the force directed graph visualization differs slightly than the treetable visualization. The force-directed graph visualization has two objects, the nodes and links. It is important to link both because they are focusable. To get the alerts between two linked IP addresses, the analyst would click the link between those IP addresses and not the IP nodes. If the analyst wanted all the alerts associated with a specific IP address, the analyst would click that IP node. Notice that the analyst can link the node address attribute to the Apache Remote IP address attribute. The benefit of this action is that the analyst can determine what tracks hold Apache alerts.

space for the highlighted data. Figure 4.7 shows the analyst that his/her linking options have been successfully implemented.



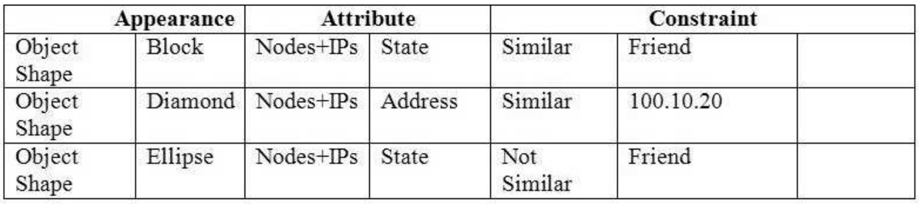
gt_track136	unknown	SERVER	Scan
gt_track191	Unknown	background scanner	Scan
gt_track256	Unknown	background scanner	Scan
gt_track45	Unknown	SERVER	Scan
gt_track124	Unknown	SERVER	Scan
gt_track183	Unknown	SERVER	Scan
gt_track132	Unknown	client	Scan
gt_track7	Unknown	SERVER	Scan
gt_track234	Unknown	SERVER	Scan
gt_track190	Unknown	SERVER	Scan
gt_track74	Unknown	SERVER	Scan
gt_track174	Unknown	client	Scan
gt_track110	Unknown	SERVER	Scan
gt_track128	Unknown	SERVER	Scan
gt_track255	Unknown	SERVER	Scan
gt_track127	Unknown	SERVER	Scan
gt_track73	Unknown	background attacker	Scan
gt_track5	Unknown	SERVER	Scan
gt_track86	Unknown	client	Scan
gt_track218	Unknown	SERVER	Scan
gt_track166	Unknown	SERVER	Scan
gt_track95	Unknown	SERVER	Scan
gt_track182	Unknown	SERVER	Scan
gt_track24	Unknown	SERVER	Scan
gt_track252	unknown	SERVER	Scan

Link To	Constraint							
	Object1	Snort	Destination IP	=	Object2	IPs + Nodes	Address	
	ForceDirectedGridGraph-3	Visibility	Show					

Figure 4.7: The analyst selected a track within the treetable visualization. The nodes and links corresponding to the selected track become highlighted.

4.1.2 Object Configuration in the Force-Directed Graph Visualization. The object configuration in the visualization is highly subjective based on the analyst's preferences. There are no restrictions on what colors, shapes or borders to implement in the visualization however there are choices that follow more logical standards. The goal is to visualize unique and important details that help the analyst quickly process information without him/her focusing on each individual node. Object configurations allow the analyst to focus in on certain details or patterns that would otherwise be hidden.

4.1.2.1 Object Shapes. Figure 4.8 depicts the analyst's object shape selections within the object properties display tab. There are not many shapes available to the analyst due to several factors. First, the visualization is unfocused until the user focuses by clicking on an object within the visualization. When unfocused, objects tend to lose structural characteristics for example the octagon becomes vi-



Appearance		Attribute		Constraint		
Object Shape	Block	Nodes+IPs	State	Similar	Friend	
Object Shape	Diamond	Nodes+IPs	Address	Similar	100.10.20	
Object Shape	Ellipse	Nodes+IPs	State	Not Similar	Friend	

Figure 4.8: The analyst chooses friendly IP addresses as squares and non-friendly IP addresses as ellipses. The diamonds are the important IP addresses that reside in the inner enclave of the network behind the firewall.

sually similar to an ellipse. Future research could develop better, more descriptive shapes. Second, the java object rendering that the toolkit implements requires significant coding in order to build the shapes.

The resulting force-directed graph visualization is shown in Figure 4.9 below. The resulting visualization is an improvement, yet the addition of object colors would greatly enhance recognition of patterns, trends or attacks. The next selection will detail the object colors utilized in this research.

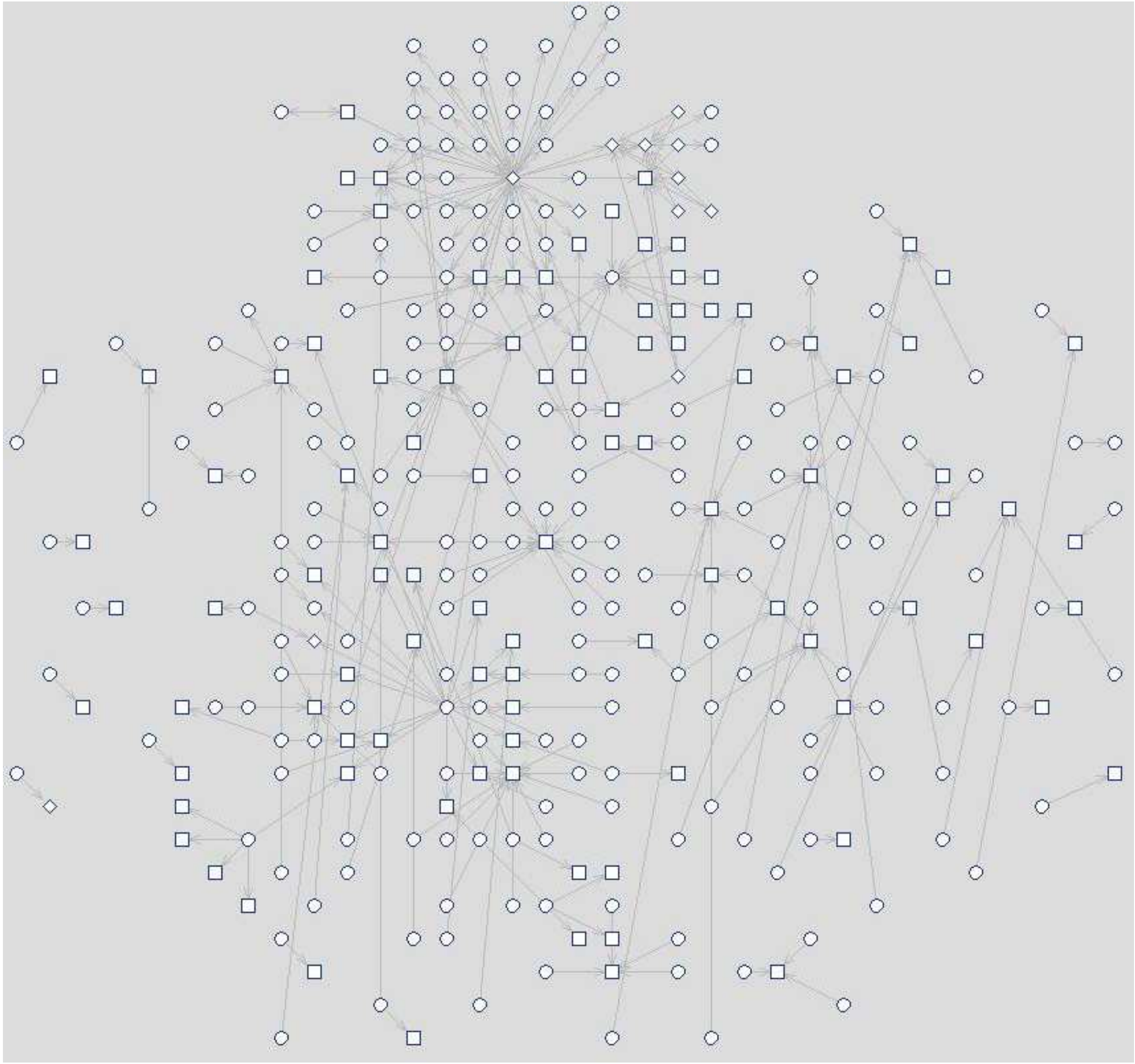
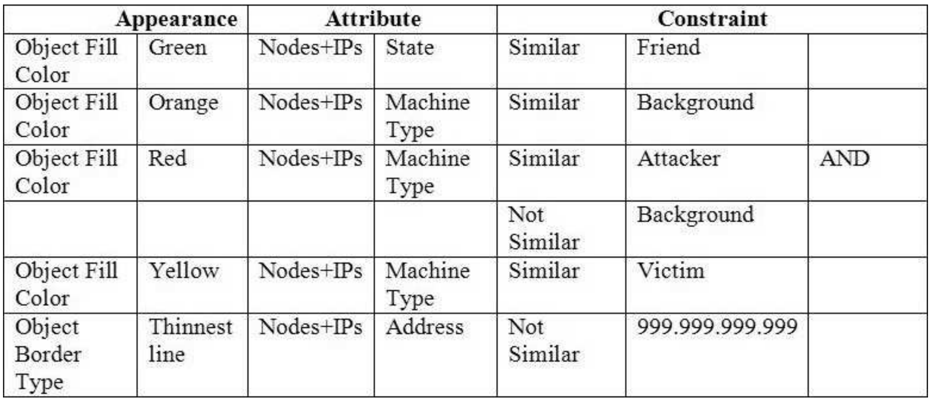


Figure 4.9: The objects are very distinguishable from one another. The analyst is able to visually discern known, trusted IPs from unknown remote IPs. Note that the unknown IP addresses are treated as foe until the analyst determines otherwise. The analyst after analysis on such an IP address can modify the state attribute in the node and the toolkit will make the appropriate object display property changes.

4.1.2.2 Object Colors. For the object colors, the analyst decided on a color scheme that of a threat assessment or vulnerability report. Figure 4.10 outlines the object colors chosen.



Appearance		Attribute		Constraint		
Object Fill Color	Green	Nodes+IPs	State	Similar	Friend	
Object Fill Color	Orange	Nodes+IPs	Machine Type	Similar	Background	
Object Fill Color	Red	Nodes+IPs	Machine Type	Similar	Attacker	AND
				Not Similar	Background	
Object Fill Color	Yellow	Nodes+IPs	Machine Type	Similar	Victim	
Object Border Type	Thinnest line	Nodes+IPs	Address	Not Similar	999.999.999.999	

Figure 4.10: The IP addresses are that friendly were colored green. The background scanners and attackers were colored orange because they are attackers but not successful. The IP addresses they are targeting are non-responsive. The research sees the color Orange as an escalation of concern. The IP addresses colored red are valid, successful attackers. The IP addresses colored in yellow represent cyber attack victims. The IP addresses with no coloring are further emphasis that their status is unknown at this time. If this were real-time the analyst would have little time to research the intentions of unknown IP addresses therefore an off-line research team would do additional information gathering on unknown IP addresses to indicate a shape or color change. Lastly, the analyst chose to set the object shape's border type to the thinnest setting to allow for better shape distinction.

Figure 4.11 shows the resulting visualization. The analyst is gaining more information about the network with each configuration. The next section will define the object border types specific to this scenario.

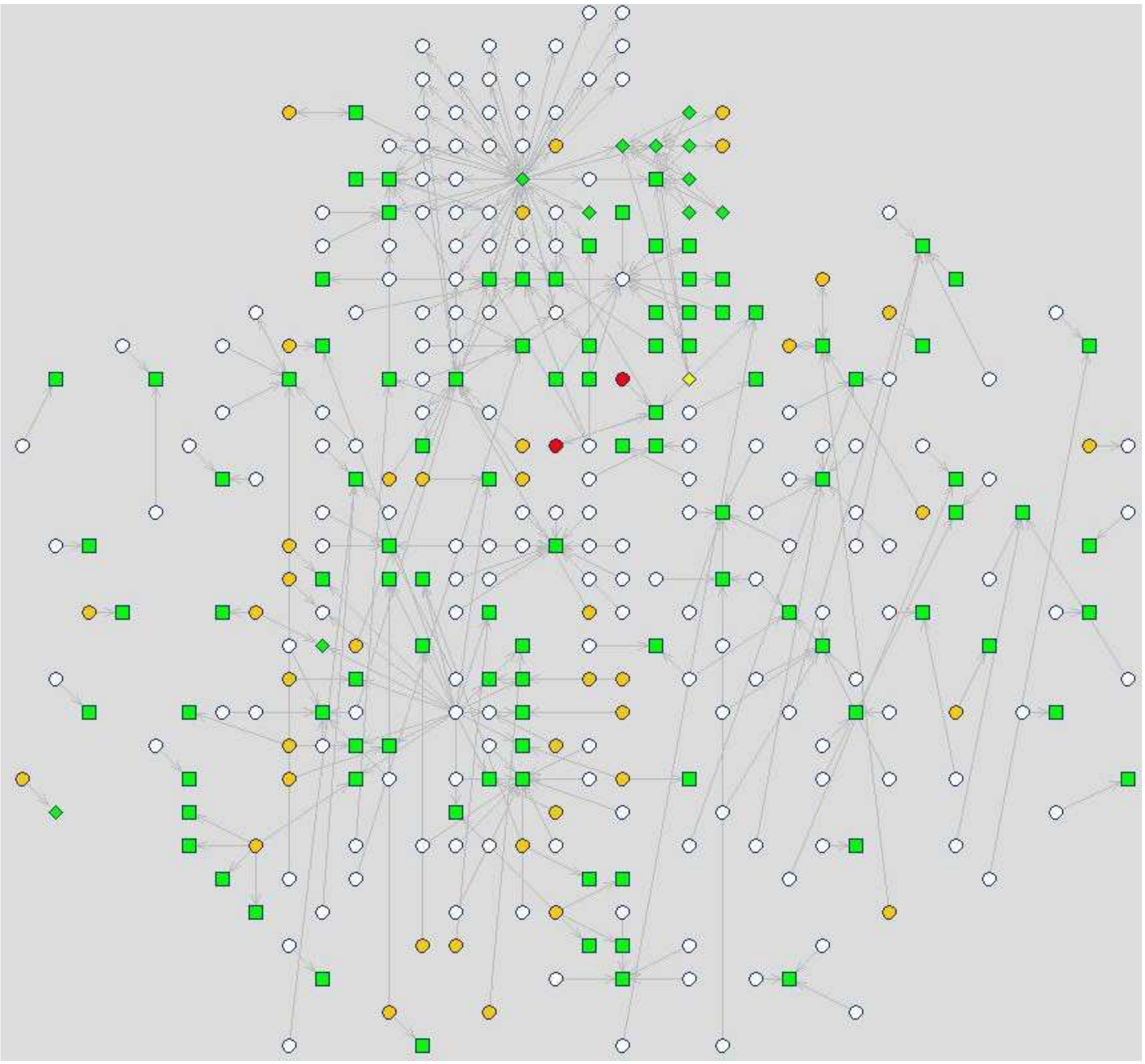
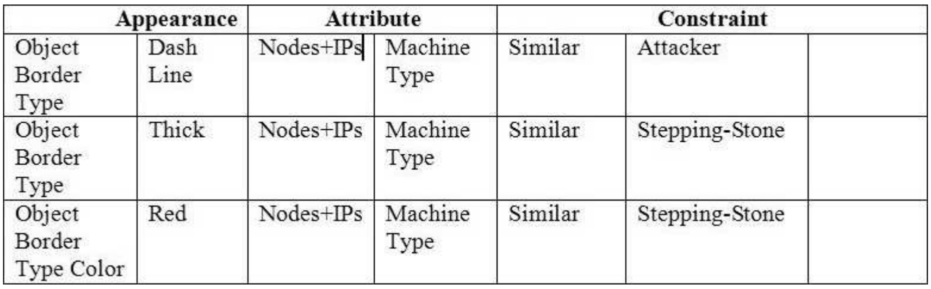


Figure 4.11: After assigning different colors to each actor in the domain, the analyst gains additional insight into the area of the network he/she should be focusing. In particular, there are two known attackers associated to a known friendly of which is associated to the known victim who resides inside the protected enclave. One of the attackers has two-way alert communication with the known friendly, the other attacker has one-way alert communication. Lastly, this scenario is not a single stage attack because of the friendly IP address that resides between the attackers and the victim. Also, the attackers do not have direct communication to the victim giving further evidence that this attack is more sophisticated.

4.1.2.3 Object Border Types. Lastly, object border types were utilized to demonstrate activity of actor. The attackers, background attackers, and stepping-stone friendly IPs in this scenario were given different borders to further distinguish them from the other actors. Figure 4.12 details the border configuration for three actors in the domain. The distinction helps the analyst visually differentiate between background attackers and scanners. The bordering puts additional emphasis on the attackers and the stepping-stone or source of the second stage of this cyber attack. Figure 4.13



Appearance		**Attribute**		**Constraint**		
Object Border Type	Dash Line	Nodes+IPs	Machine Type	Similar	Attacker	
Object Border Type	Thick	Nodes+IPs	Machine Type	Similar	Stepping-Stone	
Object Border Type Color	Red	Nodes+IPs	Machine Type	Similar	Stepping-Stone	

Figure 4.12: Assigning the dashed line border to the background attackers and attackers represents that the actor is an active entity. The enlarging and red coloring of the friendly IP address labeled a stepping-stone to signify the source of the next stage in the attack.

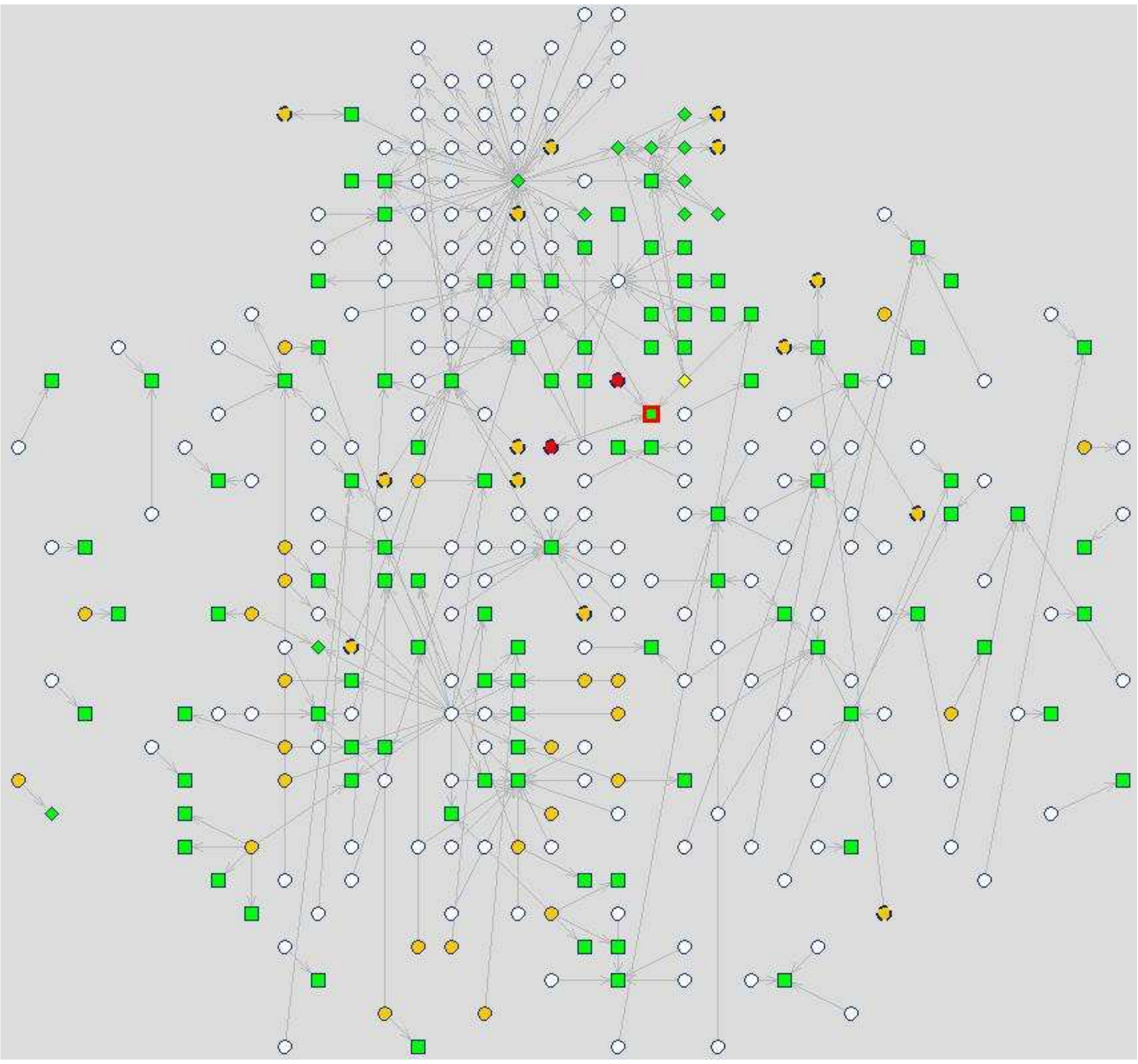


Figure 4.13: Upon examining the visualization, the red-bordered friendly node is to signify to the analyst that the box may or may not be fully compromised but nevertheless it is the source of the next stage in the progression of the cyber attack. The red bordering makes the node in the visualization stand out from all the others further emphasizing its importance. The nodes with dashed borders are attackers. The analyst can determine exactly how many background attackers and scanners there are in the scenario.

Allowing the analyst to assign object display properties based on object node attributes within the visualization is an extremely helpful and effective tool in developing SA. The analyst no longer is required to focus on object within the visualization in order to determine various properties of the object. Instead, the analyst visually processes the object's properties and begins to see activity patterns within the visualization. The next section discusses the toolkit's filtering of the visualizations so the analyst can focus on developing SA.

4.1.3 Filtering the Visualizations. Figure 4.13 is a visualization that represents the 257 tracks in the treetable visualization. Having linked the visualizations together, the analyst filters the treetable visualization. The analyst should be more concerned with the key actors in the domain. They are background scanners, background attackers, attackers, and victims therefore the analyst filters the track data for machine types similar to that criteria. The filter results produces 43 tracks in the treetable visualization. Figure 4.14 shows the resulting force-directed graph visualization after the filtering. The analyst is able to focus immediately on the important part of this visualization seen in Figure 4.15.

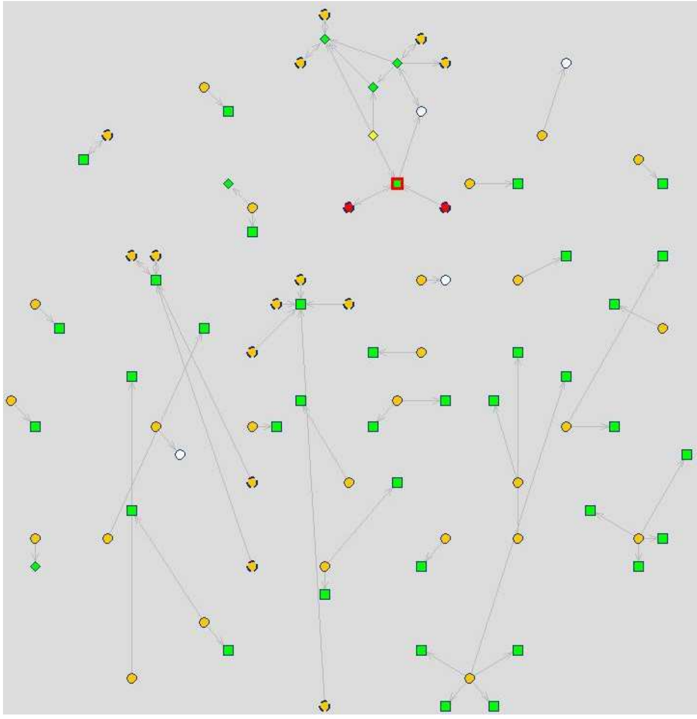


Figure 4.14: The visualization representing the treetable visualization after it has been cleaned of cluttered data.

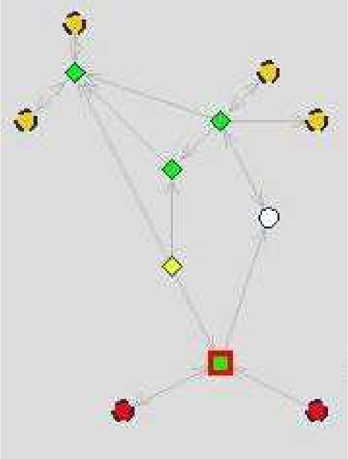


Figure 4.15: The analyst is focusing attention in this area of the visualization. This area shows the relationships that each actor has with the other.

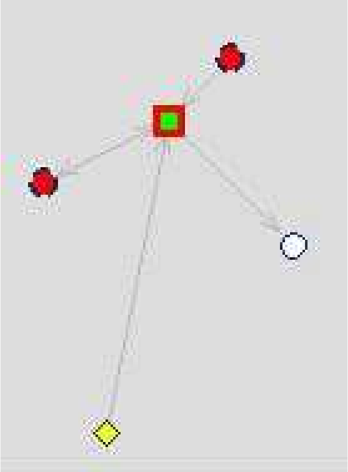
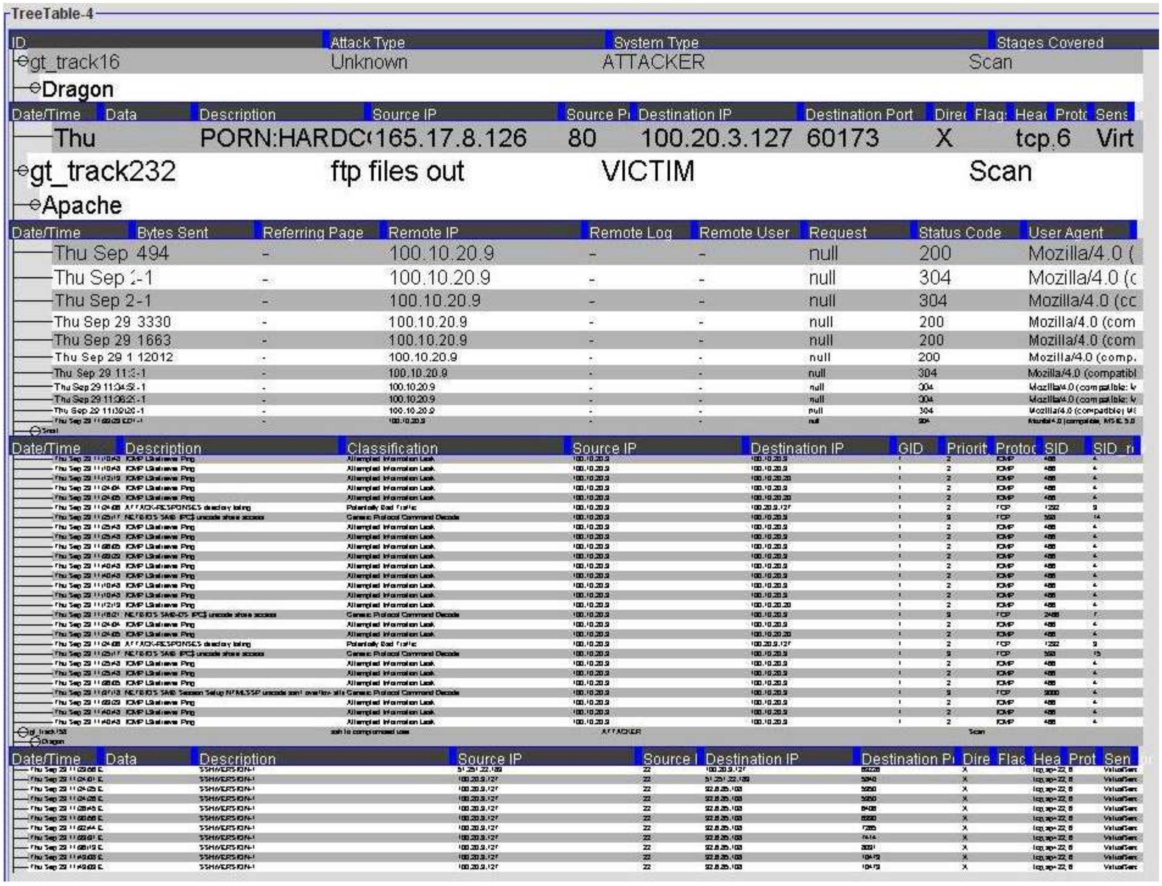


Figure 4.16: Using the toolkit, the analyst cleans the remaining data to show only the attackers, victims and one unknown.

The analyst uses the toolkit to focus closer into this area using the filter. Figure 4.16 shows the force-directed visualization after the analyst focuses on the victim and attackers using the filter to explicitly filter for machine-type equal to victim or attacker but not background in order to exclude background attackers.

Figure 4.17 shows the analyst three expanded tracks that are related to the nodes and links in the force-directed graph visualization. Take note that the analyst



TreeTable-4

ID	Attack Type	System Type	Stages Covered
gt_track16	Unknown	ATTACKER	Scan

Dragon

Date/Time	Data	Description	Source IP	Source P	Destination IP	Destination Port	Direc	Flag	Head	Protc	Sens
Thu		PORN:HARDC	165.17.8.126	80	100.20.3.127	60173	X		tcp,6		Virt

ID	Attack Type	System Type	Stages Covered
gt_track232	ftp files out	VICTIM	Scan

Apache

Date/Time	Bytes Sent	Referring Page	Remote IP	Remote Log	Remote User	Request	Status Code	User Agent
Thu Sep	494	-	100.10.20.9	-	-	null	200	Mozilla/4.0 (
Thu Sep	2-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (c
Thu Sep	2-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (cc
Thu Sep 29	3330	-	100.10.20.9	-	-	null	200	Mozilla/4.0 (com
Thu Sep 29	1663	-	100.10.20.9	-	-	null	200	Mozilla/4.0 (com
Thu Sep 29 1	12012	-	100.10.20.9	-	-	null	200	Mozilla/4.0 (comp.
Thu Sep 29 11:	-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (compatibl
Thu Sep 29 11:34:5	-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (compatible; M
Thu Sep 29 11:36:2	-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (compatible; M
Thu Sep 29 11:39:20	-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (compatible; MS
Thu Sep 29 11:69:29 ED	-1	-	100.10.20.9	-	-	null	304	Mozilla/4.0 (compatible; MSIE 5.0

Snort

Date/Time	Description	Classification	Source IP	Destination IP	GID	Priorit	Protoc	SID	SID r
Thu Sep 29 11:10:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:10:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:12:19	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.20	1	2	ICMP	466	4
Thu Sep 29 11:24:04	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:24:05	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.20	1	2	ICMP	466	4
Thu Sep 29 11:24:08	ATTACK-RESPONSES directory listing	Potentially Bad Traffic	100.10.20.9	100.20.3.127	1	2	TCP	1292	9
Thu Sep 29 11:25:17	NETBIOS SMB IPC$ unicode share access	Generic Protocol Command Decode	100.10.20.9	100.10.20.9	1	3	TCP	538	14
Thu Sep 29 11:25:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:25:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:68:05	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:69:29	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:40:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:40:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:10:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:10:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:12:19	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.20	1	2	ICMP	466	4
Thu Sep 29 11:18:21	NETBIOS SMB-DS IPC$ unicode share access	Generic Protocol Command Decode	100.10.20.9	100.10.20.9	1	3	TCP	2466	7
Thu Sep 29 11:24:04	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:24:05	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.20	1	2	ICMP	466	4
Thu Sep 29 11:24:08	ATTACK-RESPONSES directory listing	Potentially Bad Traffic	100.10.20.9	100.20.3.127	1	2	TCP	1292	9
Thu Sep 29 11:25:17	NETBIOS SMB IPC$ unicode share access	Generic Protocol Command Decode	100.10.20.9	100.10.20.9	1	3	TCP	538	15
Thu Sep 29 11:25:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:25:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:68:05	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:87:13	NETBIOS SMB Session Setup NTMLSSP unicode asn1 overflow atte	Generic Protocol Command Decode	100.10.20.9	100.10.20.9	1	3	TCP	3000	4
Thu Sep 29 11:69:29	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:40:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4
Thu Sep 29 11:40:43	ICMP L3retriever Ping	Attempted Information Leak	100.10.20.9	100.10.20.9	1	2	ICMP	466	4

ID	Attack Type	System Type	Stages Covered
gt_track158	ssh to compromised user	ATTACKER	Scan

Dragon

Date/Time	Data	Description	Source IP	Source I	Destination IP	Destination P	Dire	Flac	Hea	Prot	Sen
Thu Sep 29 11:23:56 E		SSH:VERSION-1	51.251.22.189	22	100.20.3.127	60228	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:24:01 E		SSH:VERSION-1	100.20.3.127	22	51.251.22.189	5940	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:24:25 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	5950	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:24:26 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	5950	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:28:45 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	8408	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:60:66 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	6000	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:62:44 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	7285	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:63:01 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	7414	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:68:12 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	8081	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:49:03 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	10479	X		tcp,sp=22,8		VirtualSens
Thu Sep 29 11:49:03 E		SSH:VERSION-1	100.20.3.127	22	52.8.26.103	10479	X		tcp,sp=22,8		VirtualSens

Figure 4.17: The analyst expands the three tracks to their fullest to reveal the important alerts that are associated to the nodes and links in the force directed graph visualization.

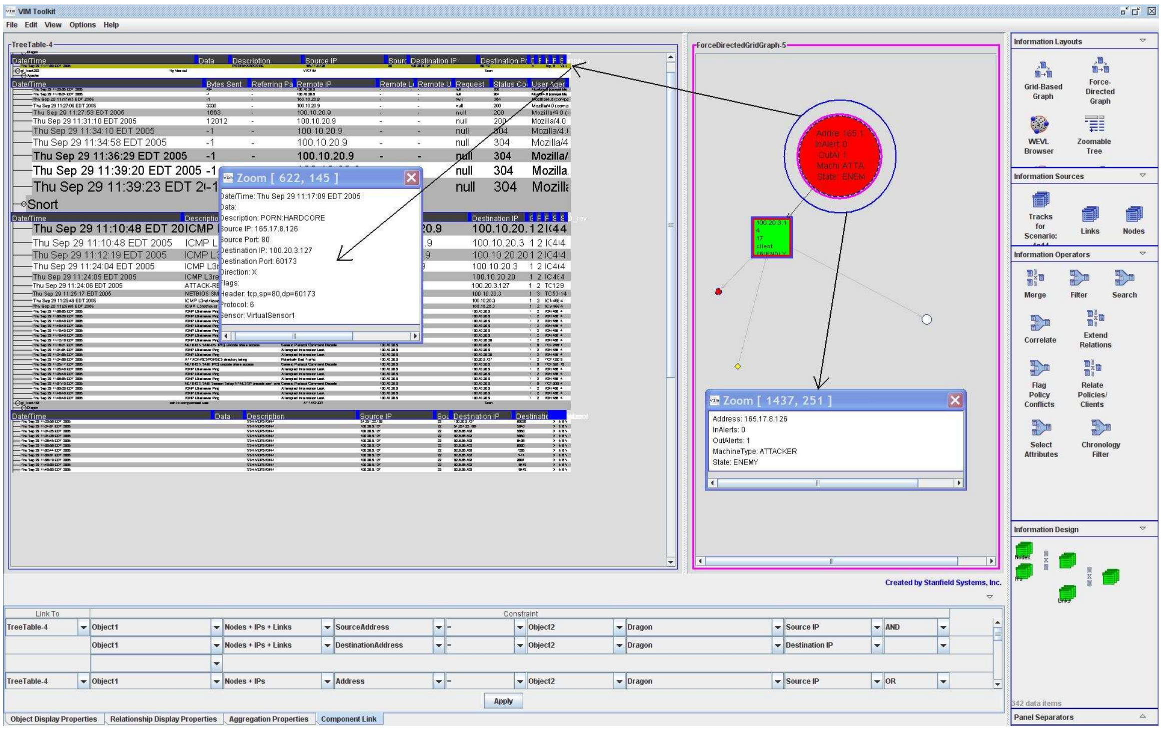
started with 257 tracks that held tens of thousands of alerts and the analyst has ended this filter process with three tracks and fewer than 50 alerts.

4.1.4 Analysis of the Visualizations. After configuring the visualizations and filtering them accordingly, the analyst can analysis the two visualizations to attempt to create a storyline of the network attack. The goal is to have the analyst create SA from these visualizations. If the analyst can put together a storyline of events that are based on truth then he/she comprehended what occurred and in the process gained valuable SA.

The analyst starts with the force-directed graph visualization because he/she can focus on a node or link and have the corresponding alerts in the treetable highlight.

This is the easier approach then attempting to sift through the numerous alerts in the treetable visualization.

To begin, the analyst highlights either attacker. The attacker usually initiates the cyber attack. The analyst is looking at the timestamps of each highlighted alert to determine which attacker initiated the attack. Figure 4.18 shows the highlighted attacker who initiated the attack on the first victim.



Zoom [622, 145]

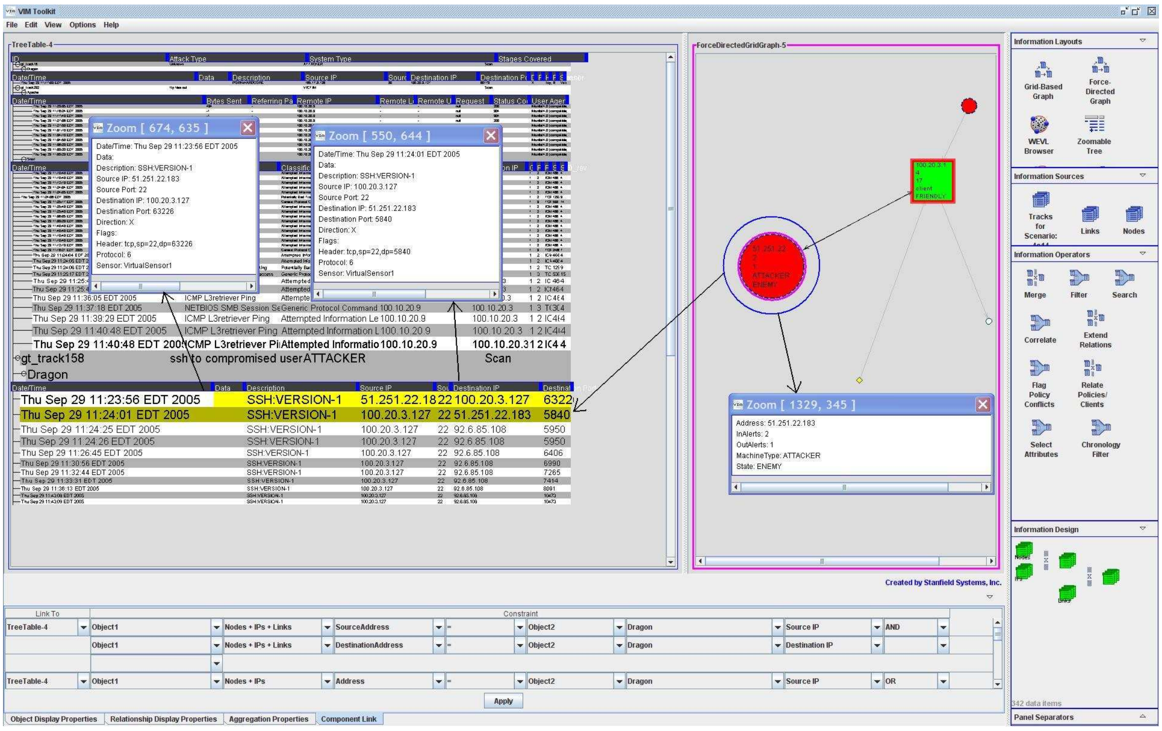
Date/Time: Thu Sep 29 11:17:09 EDT 2005
Data:
Description: PORN:HARDCORE
Source IP: 165.17.8.126
Source Port: 80
Destination IP: 100.20.3.127
Destination Port: 60173
Direction: X
Flags:
Header: tcp,sp=80,dp=60173
Protocol: 6
Sensor: VirtualSensor1

Zoom [1437, 251]

Address: 165.17.8.126
InAlerts: 0
OutAlerts: 1
MachineType: ATTACKER
State: ENEMY

Figure 4.18: The analyst focuses on the attacker node representing IP address 165.17.8.126 which highlights the dragon alert from 165.17.8.126 to the first victim, 100.20.3.127, in track 16. The alert is a porn alert so its unclear at the moment if 100.20.3.127 is compromised and if so, how it is compromised because there are no other alerts at this time.

The analyst focuses on the second attacker node in the force-directed graph visualization. The alerts corresponding to the attacker highlight in the treetable visualization. Figure 4.19 shows the analyst processing on the force-directed graph visualization and how his/her focus switches to the treetable visualization with the aid of the toolkit.



Zoom [674, 635]

Date/Time: Thu Sep 29 11:23:56 EDT 2005
Data:
Description: SSH:VERSION-1
Source IP: 51.251.22.183
Source Port: 22
Destination IP: 100.20.3.127
Destination Port: 63226
Direction: X
Flags:
Header: tcp,sp=22,dp=63226
Protocol: 6
Sensor: VirtualSensor1

Zoom [550, 644]

Date/Time: Thu Sep 29 11:24:01 EDT 2005
Data:
Description: SSH:VERSION-1
Source IP: 100.20.3.127
Source Port: 22
Destination IP: 51.251.22.183
Destination Port: 5840
Direction: X
Flags:
Header: tcp,sp=22,dp=5840
Protocol: 6
Sensor: VirtualSensor1

Zoom [1329, 345]

Address: 51.251.22.183
InAlerts: 2
OutAlerts: 1
MachineType: ATTACKER
State: ENEMY

ID	Attack Type	System Type	Stages Covered
gt_track158	ssh to compromised user	ATTACKER	Scan

Dragon

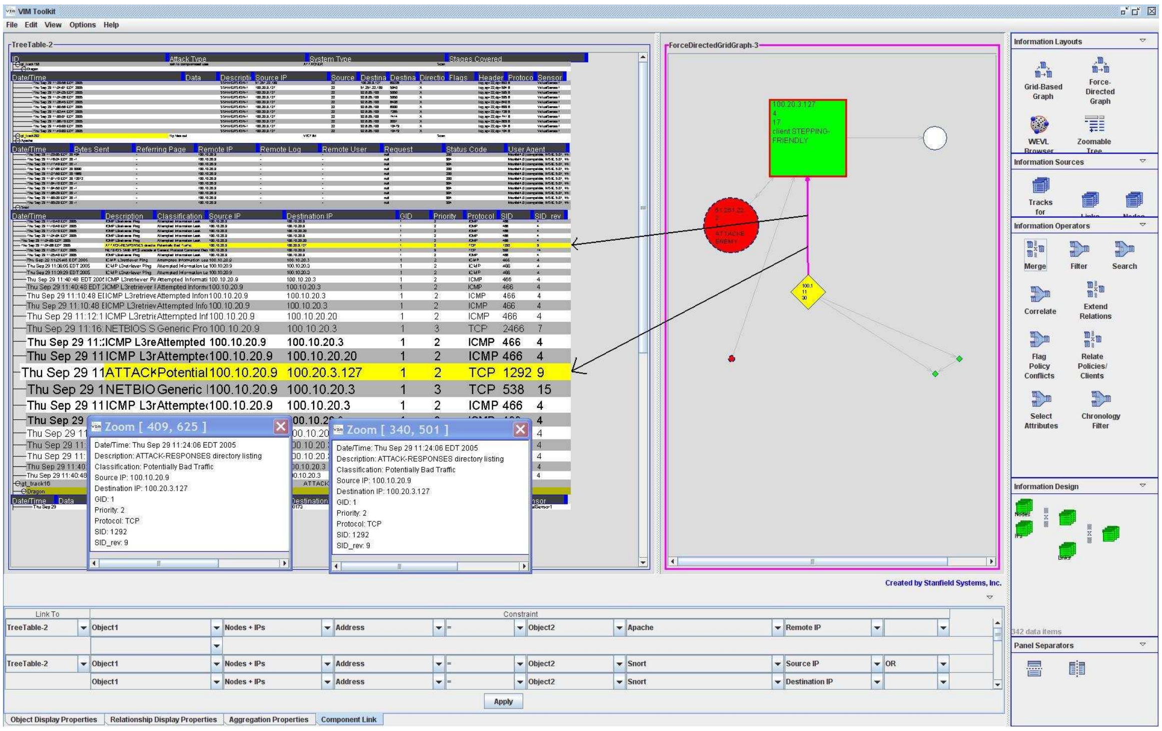
Date/Time	Data	Description	Source IP	Sou	Destination IP	Destinat
Thu Sep 29 11:23:56 EDT 2005		SSH:VERSION-1	51.251.22.18	22	100.20.3.127	6322
Thu Sep 29 11:24:01 EDT 2005		SSH:VERSION-1	100.20.3.127	22	51.251.22.183	5840
Thu Sep 29 11:24:25 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	5950
Thu Sep 29 11:24:26 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	5950
Thu Sep 29 11:26:45 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	6406
Thu Sep 29 11:30:56 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	6990
Thu Sep 29 11:32:44 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	7265
Thu Sep 29 11:33:31 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	7414
Thu Sep 29 11:36:13 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	8091
Thu Sep 29 11:43:08 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	10473
Thu Sep 29 11:43:09 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	10473

Link To					Constraint				
TreeTable-4	Object1	Nodes + IPs + Links	SourceAddress	=	Object2	Dragon	Source IP	AND	
	Object1	Nodes + IPs + Links	DestinationAddress	=	Object2	Dragon	Destination IP		
TreeTable-4	Object1	Nodes + IPs	Address	=	Object2	Dragon	Source IP	OR	

Figure 4.19: The analyst focuses on the attacker node representing IP address 51.251.22.183 which highlights 2 dragon alerts indicating bi-directional communication between 51.251.22.183 to the first victim, 100.20.3.127. The alerts are of Secure Shell (SSH) origin so the unauthorized attacker, 51.251.22.183, has opened a secure connection to the first victim. It is unclear at this point how the unauthorized IP address obtained the username and password to open a successful SSH connection, but immediately the analyst knows the first victim has been compromised. The unauthorized user has the same user privileges as the authorized user on 100.20.3.127.

At this stage, the way in which the unauthorized user obtained the username and password is unknown so the analyst would notify other network security professionals. The analyst searches the force directed graph visualization for any evidence that the attacker has compromised the host machine, 100.20.3.127. Figure 4.20 shows the two alerts associated with the compromise.

It is obvious at this point that the attacker has compromised the machine and potentially has access to any other machines connected to it. Level of access would depend on the level of access that the compromised user on 100.20.3.127 has. The files



Zoom [409, 625]

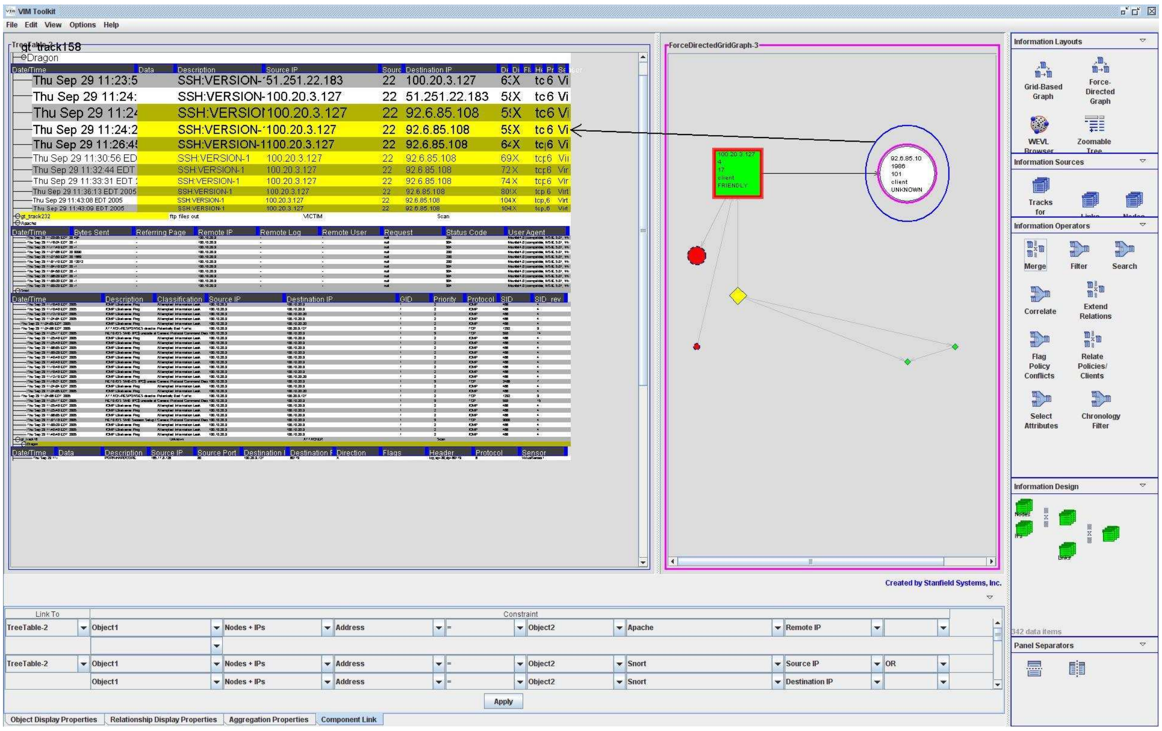
Date/Time: Thu Sep 29 11:24:06 EDT 2005
Description: ATTACK-RESPONSES directory listing
Classification: Potentially Bad Traffic
Source IP: 100.10.20.9
Destination IP: 100.20.3.127
GID: 1
Priority: 2
Protocol: TCP
SID: 1292
SID_rev: 9

Zoom [340, 501]

Date/Time: Thu Sep 29 11:24:06 EDT 2005
Description: ATTACK-RESPONSES directory listing
Classification: Potentially Bad Traffic
Source IP: 100.10.20.9
Destination IP: 100.20.3.127
GID: 1
Priority: 2
Protocol: TCP
SID: 1292
SID_rev: 9

Figure 4.20: The analyst notices the SID number 1292 referenced in both Snort alerts and researches it using a search engine. Perhaps the analyst already knows the signature referenced by this SID number. The alert is associated with post-compromise behavior indicating the use of Windows Directory listing tools. The attacker used an exploit to gain a command shell to execute a dir command. He/She is attempting to gain additional information on any vulnerable electronic documents on the host machine.

on the server at 100.10.20.9 is compromised after the attacker downloads the files from it to 100.20.3.127. From 100.20.3.127, the attacker ftps the files from 100.20.3.127 and 100.10.10.9 to 92.6.85.108 as evidenced by Figure 4.21.



gt_track158

Dragon

Date/Time	Data	Description	Source IP	Sourc	Destination IP	Di	Di	Fl	Hi	Pr	Se
Thu Sep 29 11:23:5		SSH:VERSION-'	51.251.22.183	22	100.20.3.127	6:	X		tc	6	Vi
Thu Sep 29 11:24:		SSH:VERSION-	100.20.3.127	22	51.251.22.183	5(	X		tc	6	Vi
Thu Sep 29 11:2		SSH:VERSIOI	100.20.3.127	22	92.6.85.108	5!	X		tc	6	Vi
Thu Sep 29 11:24:2		SSH:VERSION-'	100.20.3.127	22	92.6.85.108	5!	X		tc	6	Vi
Thu Sep 29 11:26:4		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	6	X		tc	6	Vi
Thu Sep 29 11:30:56 ED		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	69	X		tcp	6	Vi
Thu Sep 29 11:32:44 EDT		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	72	X		tcp	6	Vir
Thu Sep 29 11:33:31 EDT		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	74	X		tcp	6	Vir
Thu Sep 29 11:36:13 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	80	X		tcp	6	Virt
Thu Sep 29 11:43:08 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	104	X		tcp	6	Virt
Thu Sep 29 11:43:09 EDT 2005		SSH:VERSION-1	100.20.3.127	22	92.6.85.108	104	X		tcp	6	Virt

gt_track232 ftp files out VICTIM Scan

Figure 4.21: The analyst focuses on IP address 92.6.85.108. Nine dragon alerts were generated due to unauthorized SSH connections from 100.20.3.127 to 92.6.85.108. The analyst concludes that the attacker ftp'd the files to this IP address.

After the attack, they would do a follow-up investigation that involves interrogation on the authorized user of 100.10.3.127. The user would explain his/her unauthorized usage of government property to access an unauthorized porn website. The investigation would conclude that the attacker at 51.251.22.183 used the information gained from the website to open an SSH connection to the user's machine.

4.1.4.1 Summary. The PNP exploit cyber attack scenario was described in detail. First, the loading of the information sources to create the necessary visualizations was discussed. Second, the force directed graph visualization was configured to map colors and shapes to important data attributes within the visualization. The colors and shapes allowed the analyst to process key patterns and relationships quickly. Next, the visualizations were linked together bi-directionally to allow the analyst to view associated data across them. Lastly, the filter was utilized to reduce the information sources within the visualization to a humanly-manageable level.

All of these steps enabled the analyst to step through the scenario smoothly. The analyst was able to recognize the beginning of the scenario and follow the attacker as he/she moved from his/her origin into the first victim machine then cause a second machine to be a victim and ultimately succeed by ftp-ing the files off-site.

4.2 Research Observations

In this section, key observations from different scenarios will be highlighted to demonstrate the toolkit's effectiveness in visualizing the SA for the analyst.

4.2.0.2 Roles of Actors. The force-directed graph visualization when configured using the visualization setup outlined in Section 4.1.2 effectively visualizes the scenario parameters that this research effort was given in the form of artifacts. Specifically, the role and success of each actor was accurately portrayed by the visualization. For example, this research knows that background scanners are solely reconnaissance efforts therefore the analyst should not see two-way IDS alert communication between a scanner and a network asset. Figure 4.23 illustrates an example of the scanning activity. The analyst can examine the alerts by focusing on the link between the outside scanner and an inside asset.

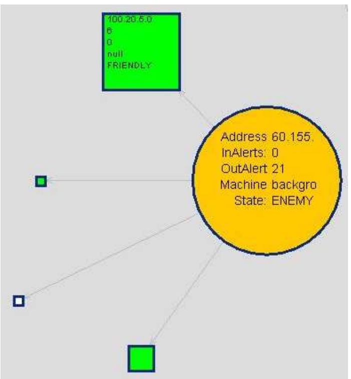
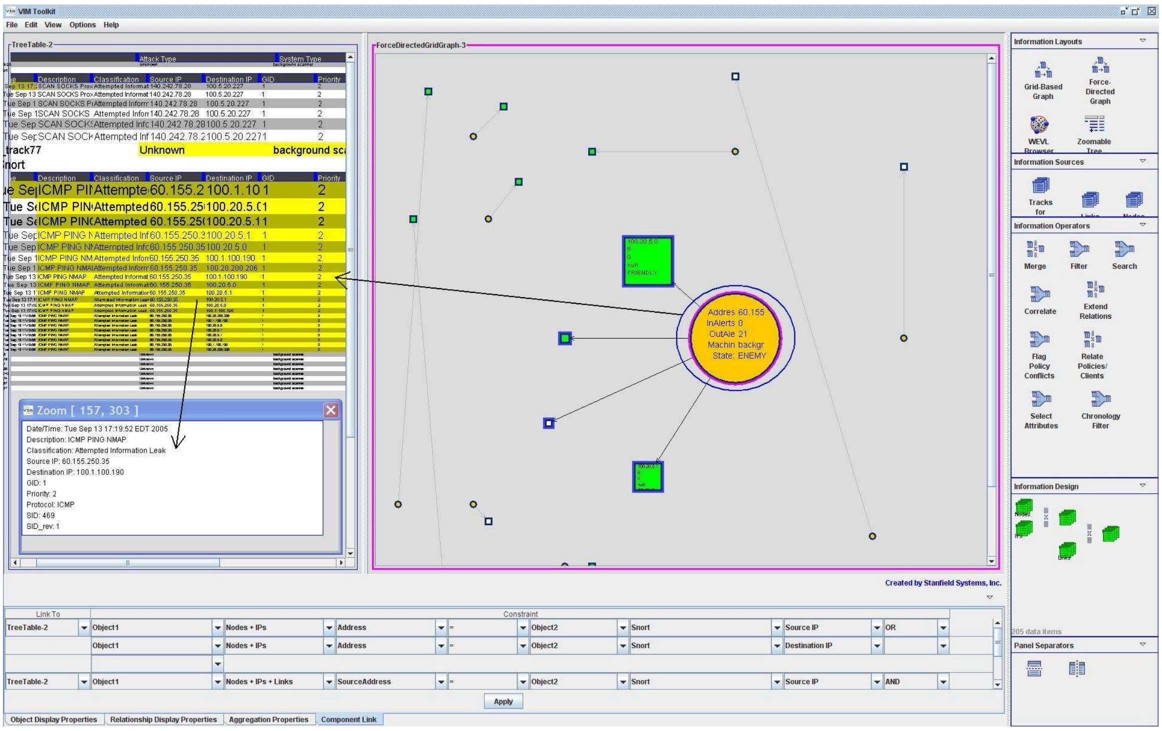


Figure 4.22: The analyst views directed alerts from an outside threat to several inside assets.



Zoom [157, 303]

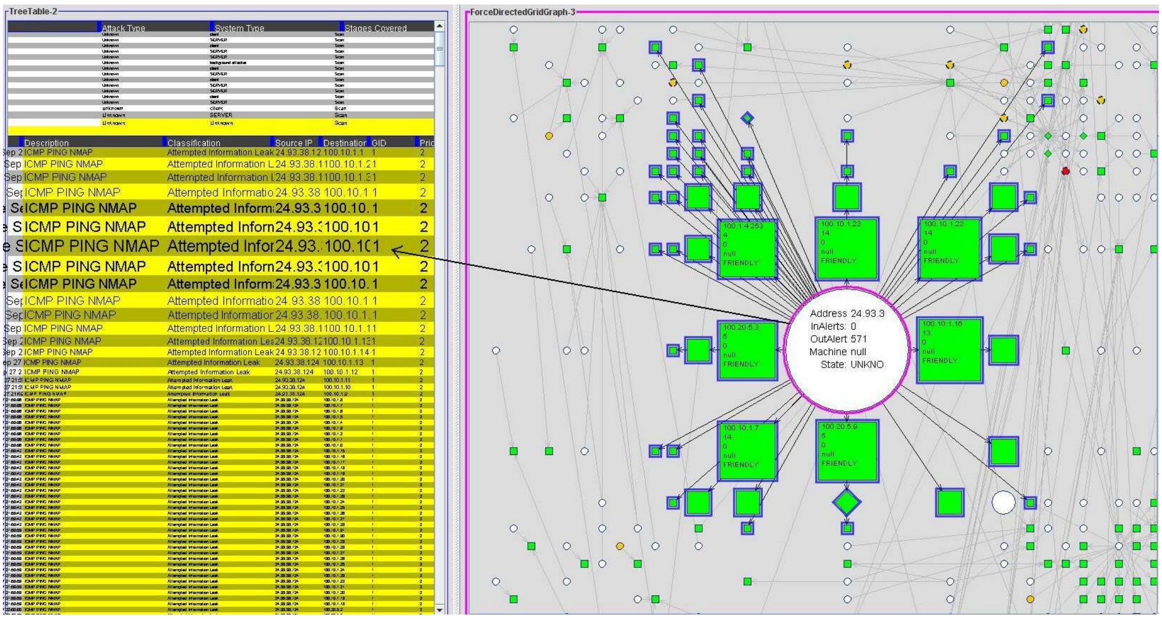
Date/Time: Tue Sep 13 17:19:52 EDT 2005
Description: ICMP PING NMAP
Classification: Attempted Information Leak
Source IP: 60.155.250.35
Destination IP: 100.1.100.190
GID: 1
Priority: 2
Protocol: ICMP
SID: 469
SID_rev: 1

Link To	Constraint							
TreeTable-2	Object1	Nodes + IPs	Address	=	Object2	Snort	Source IP	OR
	Object1	Nodes + IPs	Address	=	Object2	Snort	Destination IP	
TreeTable-2	Object1	Nodes + IPs + Links	SourceAddress	=	Object2	Snort	Source IP	AND

Figure 4.23: The analyst focuses on the background scanner node within the force directed graph visualization to highlight the linked alerts in the treetable visualization. Snort has generated the alerts based on the scanner using NMAP's ICMP Pinging attempts to leak information unsuccessfully from the target IP address.

4.2.0.3 Visualizing Network Activity. The analyst can utilize the toolkit to determine if the current network activities being visualized are normal or abnormal. Specifically, the analyst loads the firewall misconfiguration scenario in Section 3.2.4.5 demonstrating that the visualizations can aid the analyst in determining if the proper protections and/or procedures are executing properly on the network. Using the visualization setup outlined in Chapter IV, Figure 4.24 shows a dramatic increase in alert activity of the background scanners and attackers. The background scanners and attackers have access to the Inner enclave, represented as green diamonds in the visualization.

The analyst is alerted to a possible misconfiguration because normally the background scanners and attackers appear as shown in Figure 4.25.



	Attack Type	System Type	Stages Covered
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	SERVER	Scan
	Unknown	background attacke	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	SERVER	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	Unknown	Scan

	Description	Classification	Source IP	Destinatior	GID	Pric
Sep 2	ICMP PING NMAP	Attempted Information Leak	24.93.38.12	100.10.1.1	1	2
Sep	ICMP PING NMAP	Attempted Information L	24.93.38.1	100.10.1.2	1	2
Sep	ICMP PING NMAP	Attempted Information L	24.93.38.1	100.10.1.3	1	2
Sep	ICMP PING NMAP	Attempted Informatio	24.93.38	100.10.1	1	2
e Se	ICMP PING NMAP	Attempted Inform	24.93.3	100.10.	1	2
e S	ICMP PING NMAP	Attempted Inforn	24.93.3	100.10	1	2
e S	ICMP PING NMAP	Attempted Infor	24.93.	100.10	1	2
e S	ICMP PING NMAP	Attempted Inforn	24.93.3	100.10	1	2
e Se	ICMP PING NMAP	Attempted Inform	24.93.3	100.10.	1	2
Sep	ICMP PING NMAP	Attempted Informatio	24.93.38	100.10.1	1	2
Sep	ICMP PING NMAP	Attempted Informatior	24.93.38.	100.10.1.	1	2
Sep	ICMP PING NMAP	Attempted Information L	24.93.38.1	100.10.1.1	1	2
Sep 2	ICMP PING NMAP	Attempted Information Lea	24.93.38.12	100.10.1.13	1	2
Sep 2	ICMP PING NMAP	Attempted Information Leak	24.93.38.12	100.10.1.14	1	2
ep 27	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.13	1	2
p 27 2	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.12	1	2
27 21:5	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.11	1	2
27 21:5	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.10	1	2
27 21:5	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.9	1	2

Figure 4.24: The analyst focuses on the on an unknown node that generated 571 alerts to friend IP addresses as well as several IP addresses within the inner enclave. The Snort alerts are NMAP ICMP PING. The unknown node is behaving similarly to that of background scanners. The unknown node has visible alert links to the inner enclave represented as green diamond objects. The firewall should have prevented the connection between unknown, background attackers and scanners to the inner enclave. This is a definite flag that the firewall protecting the inner enclave is setup improperly.

The analyst having researched the alert activity of the unknown node can modify the IP address's machine type attribute to accurately reflect its true motivations. The unknown node is a background scanner and should be labeled as such. Figure 4.26 illustrates the end result of the analyst relabelling the unknown node as a background scanner.

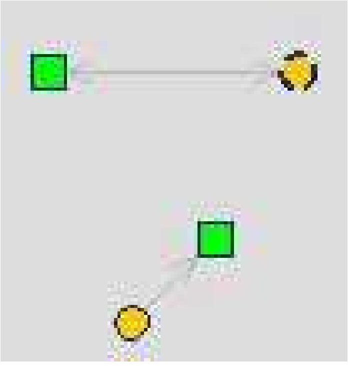
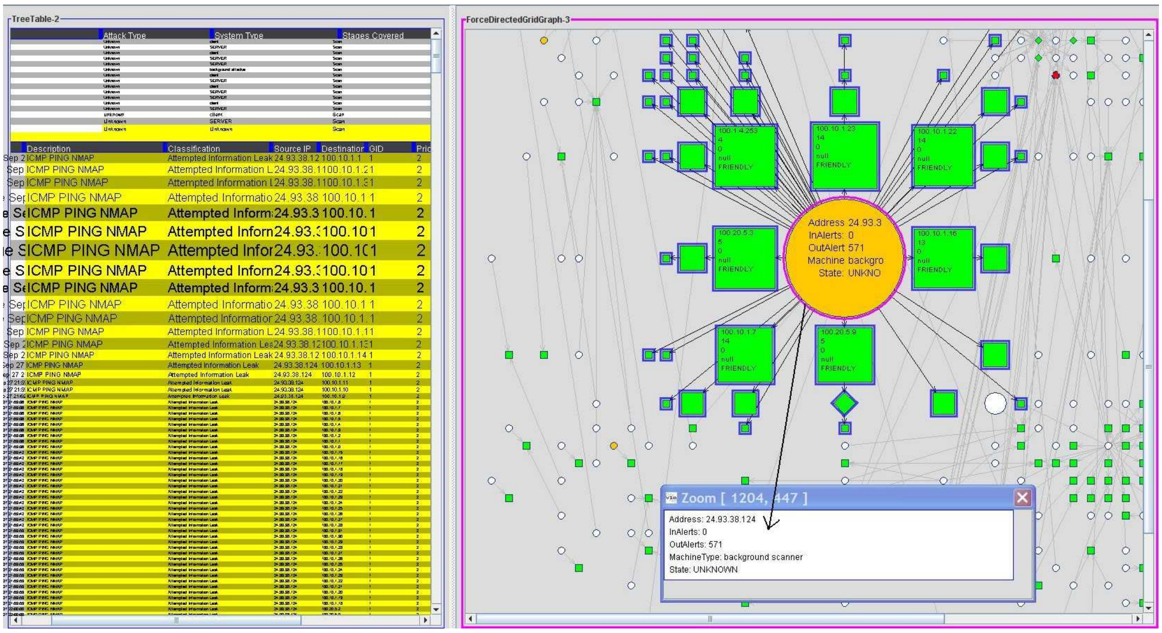


Figure 4.25: The normal activity of background scanners and attackers seen in the other cyber attack scenarios that have a properly configured firewall. There is a dramatic difference in activity between the background scanners and attackers in this figure compared to the activity of Figure 4.26.



	Attack Type	System Type	Stages Covered
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	SERVER	Scan
	Unknown	background attacker	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	SERVER	Scan
	Unknown	client	Scan
	Unknown	SERVER	Scan
	unknown	client	Scan
	Unknown	SERVER	Scan
	Unknown	Unknown	Scan

	Description	Classification	Source IP	Destination	GID	Pric
Sep 2	ICMP PING NMAP	Attempted Information Leak	24.93.38.12	100.10.1.1	1	2
Sep	ICMP PING NMAP	Attempted Information L	24.93.38.1	100.10.1.2	1	2
Sep	ICMP PING NMAP	Attempted Information L	24.93.38.1	100.10.1.3	1	2
Sep	ICMP PING NMAP	Attempted Informatio	24.93.38	100.10.1	1	2
e Se	ICMP PING NMAP	Attempted Inform	24.93.3	100.10.	1	2
e S	ICMP PING NMAP	Attempted Inforn	24.93.	100.10	1	2
e S	ICMP PING NMAP	Attempted Infor	24.93.	100.1(	1	2
e S	ICMP PING NMAP	Attempted Inforn	24.93.	100.10	1	2
e Se	ICMP PING NMAP	Attempted Inform	24.93.3	100.10.	1	2
Sep	ICMP PING NMAP	Attempted Informatio	24.93.38	100.10.1	1	2
Sep	ICMP PING NMAP	Attempted Informatior	24.93.38.	100.10.1.	1	2
Sep	ICMP PING NMAP	Attempted Information L	24.93.38.1	100.10.1.1	1	2
Sep 2	ICMP PING NMAP	Attempted Information Lea	24.93.38.12	100.10.1.13	1	2
Sep 2	ICMP PING NMAP	Attempted Information Leak	24.93.38.12	100.10.1.14	1	2
Sep 27	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.13	1	2
ep 27 2	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.12	1	2
p 27 21:5	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.11	1	2
p 27 21:5	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.10	1	2
27 21:5	ICMP PING NMAP	Attempted Information Leak	24.93.38.124	100.10.1.9	1	2

Address: 24.93.38.124
InAlerts: 0
OutAlerts: 571
MachineType: background scanner
State: UNKNOWN

Figure 4.26: The analyst relabels the unknown node as a background scanner because he/she has sufficient evidence to conclude that it is a background scanner. The analyst focused on the unknown node and the linked alerts in the treetable displayed that in fact the alerts generated by the unknown node where of the same type typically generated by background scanners. This scanner used NMAP to IP sweep all available friendly IP addresses that were available to the scanner in order gain important network information such as available services or Operating System versions. Notice that toolkit automatically refreshes the object display properties of the visualization to register the analyst's action. The node changed from white to orange to indicate an increase in threat.

V. Conclusions

This chapter presents a summary of the research conducted and conclusions from the analysis provided in Chapter 4. Significance of the research, contributions and areas for future research is discussed.

5.1 Research Summary and Conclusions

Network analysts use intrusion detection systems as the traditional method to defend computer networks and resources against network attacks. The difficulty to defend against such attacks increases as networks and bandwidth have become larger allowing greater client usage. Network analysts employ several IDSs in an attempt to gain a better understanding of the current situation yet in reality, the high priority attacks are obscured by the large volume of low priority scan alerts, false-positive and redundant event alerts. Additionally, network analysts are unable to cognitively process the IDS alerts at the rate required in order to gain truthful Situational Awareness. Lastly, IDSs do not facilitate the effort required of analysts to associate disparate alerts into a multi-stage attack. This research proposed using the SA Reference Model to increase the Situation Awareness for network analysts. Specifically, the raw event alerts generated by the IDSs are cleaned and catalogued into collections by a parsing algorithm. Those collections feed into VIM, a software system, in order to visualize the network activity.

The research investigated the use of visualizations to enhance a network analyst's ability to gain Situational Awareness relating to network activity. Allowing the analyst flexibility in visualization and display property selection aided the analyst in developing SA. The research concluded the development of SA is an active process between the analyst and the toolkit. The analyst configures the visualization(s) to his or her preference then uses the powerful filters to reduce the visualization into a meaningful picture. The analyst was able to filter tens of thousands of IDS alerts into a small subset of key alerts related to the attack. The research found that the fused alert data alone was unable to provide the analyst the network context required in order to develop SA. This research found that the integration of other network data in conjunction with the fused alert data was more effective in building the SA. Lastly, the research was able to produce many of the parameters of the experiment visually. For example, the research was able to visualize the incorrect configuration of the network firewall. Using the visualization, the analyst was able to see that there was something wrong. The analyst saw background scanners having access to the inner enclave of the network which was supposedly protected.

5.2 Research Limitations

The research was limited in several ways which were not realized until later in the process. The analyst gains SA by being presented factual information that is pertinent to the existing situation. The analyst uses the information in conjunction with his or her own knowledge and experience having been monitoring the network previously to build the SA. The research utilized fused alert data from a mock network based on a existing network, referenced in Appendix A. The researcher had no experience with the real network therefore no reference ability to make decisions for questions such as what are the critical assets of the network or is this alert load between two IP addresses significant? Many more questions arise during this SA building exercise but as an outside analyst, the researcher is unable to provide the answers. The research was given five scenarios in order to develop this reference ability which was deficient. Given an analyst with more experience on the network in which the fused alert data is derived, the SA-forming process is more efficient and effective. Next, the fused alert data at the track level was not very helpful in and of itself. Of the four track attributes it contained, two of those attributes rarely changed in terms of value leaving the research having to work with the IDS alerts within those tracks. Next, for each scenario, the tracks were not given a timestamp. The MMA generates tracks simultaneously but it feeds a track to the toolkit when it has determined that the track is complete. It was extremely difficult to generate a timeline of track creation with no timestamp. Having that timeline could of been another indication to the analyst that something significant was occurring.

5.3 Future Work

There are several things that can be achieved in future efforts. Further testing is required using more scenarios with an analyst that is familiar with the testing environment specifically the network, referenced in Appendix A, that produced the alerts used in the fused alert data. The toolkit supports the notion of time therefore more effort should be focused on updating the visualizations through time. Timestamping the tracks would be very beneficial so that the treetable visualization updates accordingly with the other graph-like visualizations. Next, is to make the toolkit a real-time system. This requires the MMA pushing the tracks into the toolkit's database and having the database push those changes to the toolkit or have the toolkit query for new updates based on the analyst's current configuration. Next, the object shapes need to be redone. Using Java2D is limiting especially when designing new shapes. The use of the image icons would be more beneficial and the focusing to see the object's data would be handled the same. The analyst does not need to view the data contents of the objects after he or she has assigned object display properties. Next, a redesign of the toolkit's bottom panel that includes the component-linking and object display properties is required. The process of assigning the constraints on the visualizations is time-consuming and is distracting. Next, research into hardware acceleration is available to increase the performance of the toolkit. Finally, follow-on research is necessary to metrically determine the benefit of SA development with the VIM toolkit versus SA development without the VIM toolkit. This would involve real-time usability testing with experienced analysts.

Appendix A. Computer Network Testing Background

A.1 Network Infrastructure

The computer network used to run the cyber attack scenarios against is a fictional network based on a real network named Open Source Information System (OSIS). It is an unclassified network used by the Intelligence Community to share sensitive-but-unclassified information. OSIS is primarily a network for the display of finished intelligence product, rather than a source of collaboration or collection. The model of network is simulated by a testbed of approximately 20 physical machines. The testbed includes real hosts (clients and servers) and network infrastructure components (routers, hubs and firewalls); and several hundred virtual hosts simulated by traffic generation machines. The simulated OSIS is connected to a model Internet, which is simulated by two other traffic generation machines. The actual structure of the complete OSIS-Internet model is given in Figure A.1 and Figure A.2 below.

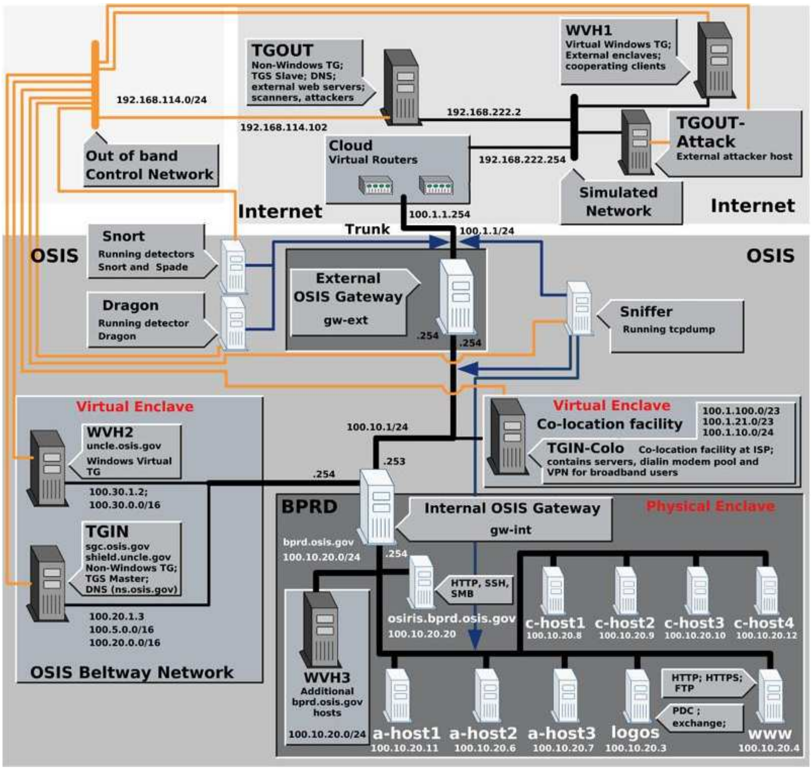


Figure A.1: The infrastructure of the OSIS network.

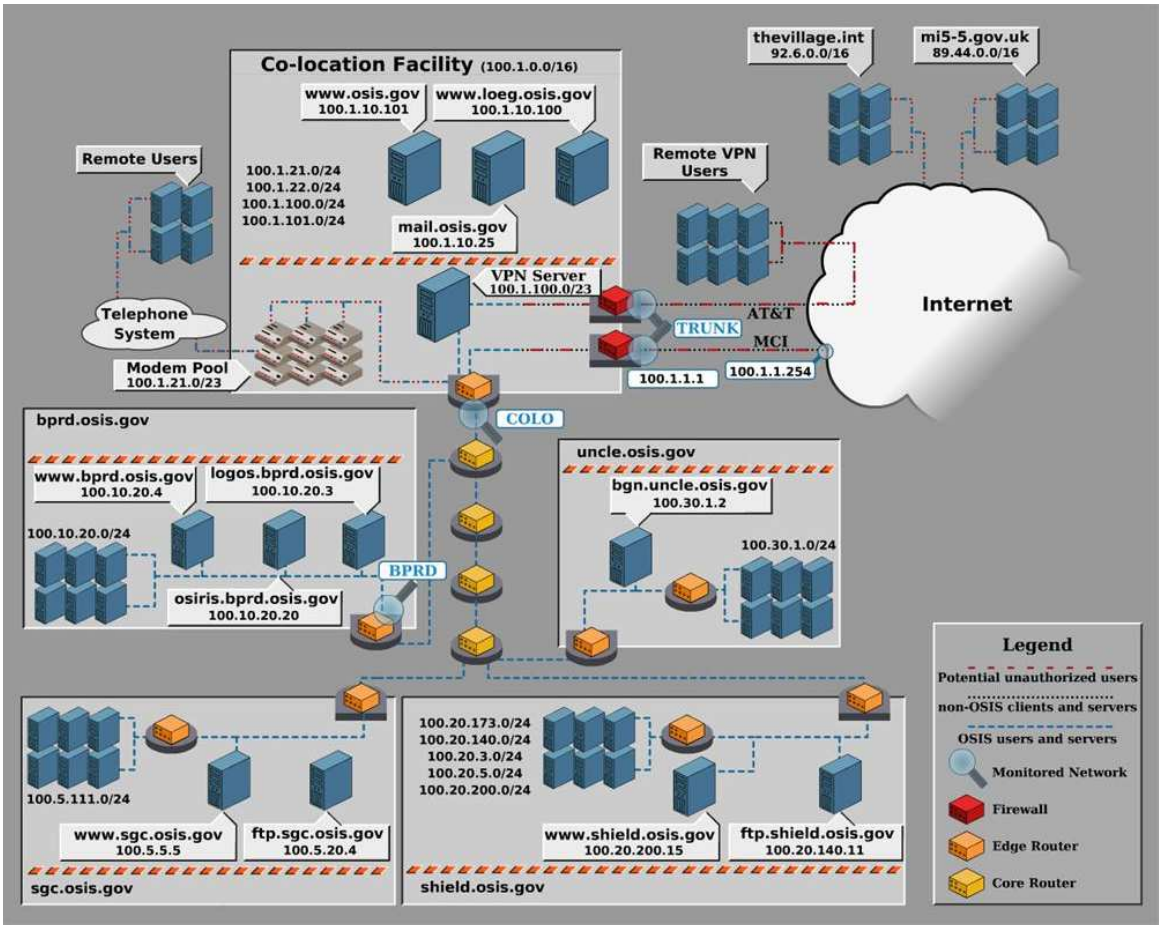

Co-location Facility (100.1.0.0/16)
www.osis.gov 100.1.10.101
www.loeg.osis.gov 100.1.10.100
thevillage.int 92.6.0.0/16
mi5-5.gov.uk 89.44.0.0/16
Remote Users
100.1.21.0/24
100.1.22.0/24
100.1.100.0/24
100.1.101.0/24
Remote VPN Users
mail.osis.gov 100.1.10.25
VPN Server 100.1.100.0/23
Telephone System
Modem Pool 100.1.21.0/23
AT&T
TRUNK
MCI
Internet
100.1.1.1
100.1.1.254
COLO
bprd.osis.gov
www.bprd.osis.gov 100.10.20.4
logos.bprd.osis.gov 100.10.20.3
100.10.20.0/24
osiris.bprd.osis.gov 100.10.20.20
BPRD
uncle.osis.gov
bgn.uncle.osis.gov 100.30.1.2
100.30.1.0/24
Legend
Potential unauthorized users
non-OSIS clients and servers
OSIS users and servers
Monitored Network
Firewall
Edge Router
Core Router
100.5.111.0/24
www.sgc.osis.gov 100.5.5.5
ftp.sgc.osis.gov 100.5.20.4
sgc.osis.gov
100.20.173.0/24
100.20.140.0/24
100.20.3.0/24
100.20.5.0/24
100.20.200.0/24
www.shield.osis.gov 100.20.200.15
ftp.shield.osis.gov 100.20.140.11
shield.osis.gov

Figure A.2: The infrastructure of the notional OSIS network that consists of an internet-connected backbone combined with several fictional Intelligence communities.

This appendix includes the steps leading up to choosing an information layout. As soon as the information sources are loaded, the analyst can choose an information layout to visualize the information source.

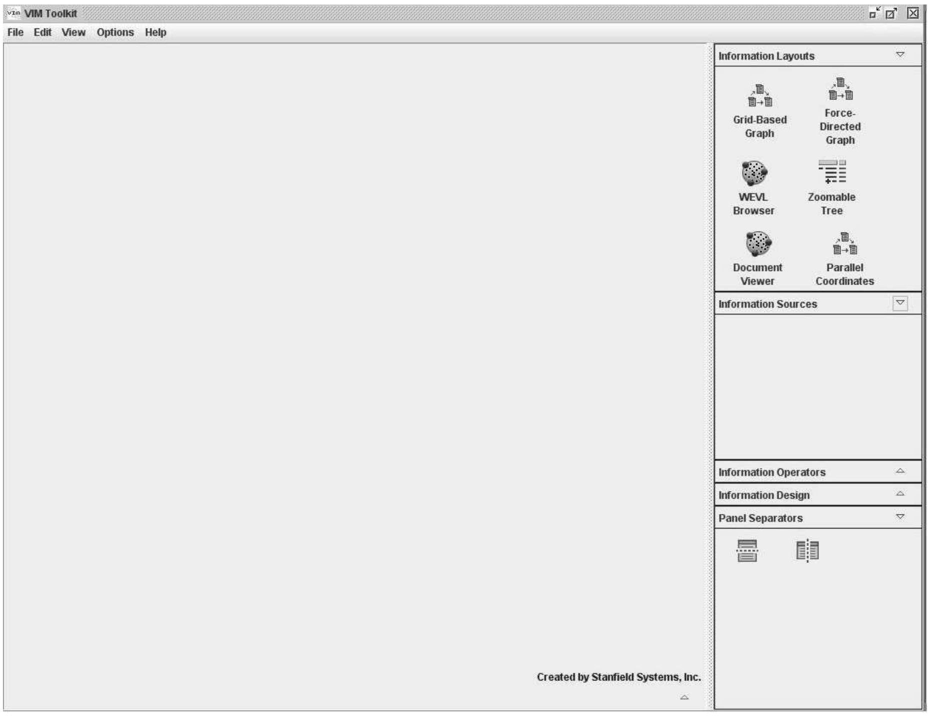


Figure B.1: The status of the toolkit when it starts up.

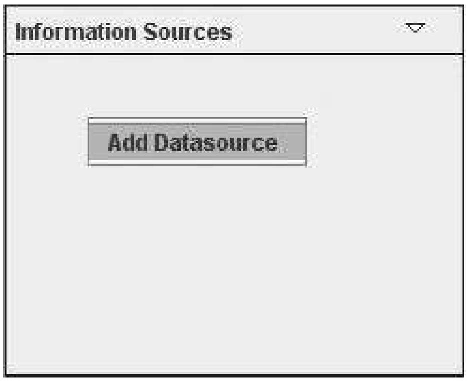
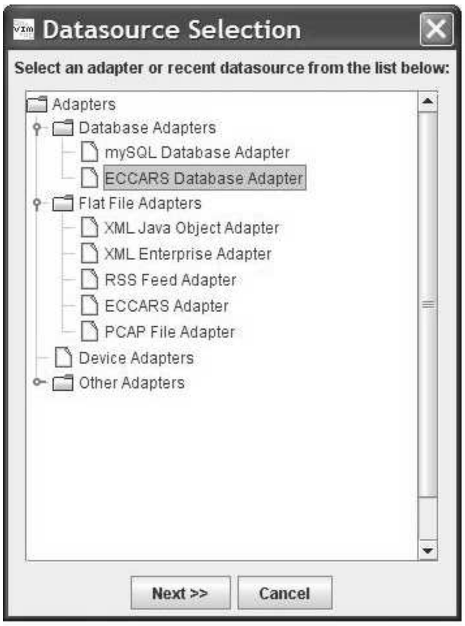


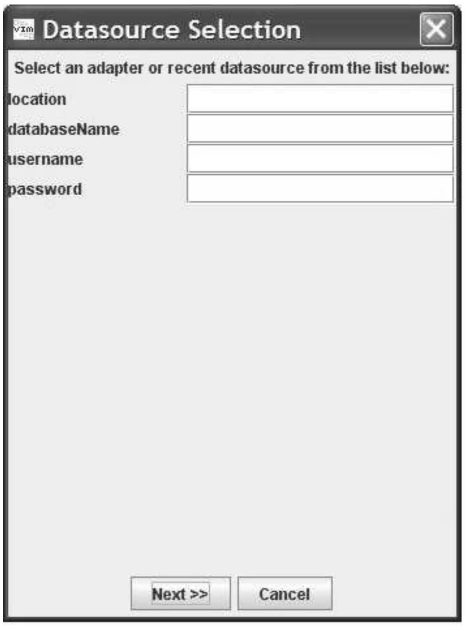
Figure B.2: Analyst right-clicks the information source area and chooses to add data source.



Datasource Selection

Select an adapter or recent datasource from the list below:

Figure B.3: Choosing to load an information source will cause this pop-up selection menu in upper left corner of the workspace. The darkened choice is the adaptor to load the fused alert database.



Datasource Selection

Select an adapter or recent datasource from the list below:

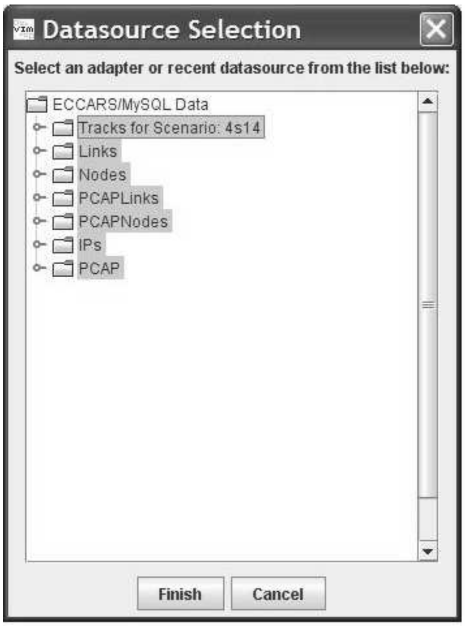
location

databaseName

username

password

Figure B.4: Prompt for analyst to enter in the connection information to connect to the database.



Datasource Selection

Select an adapter or recent datasource from the list below:

Figure B.5: All the tables within the database are available. The analyst selects whichever database tables he or she prefers.

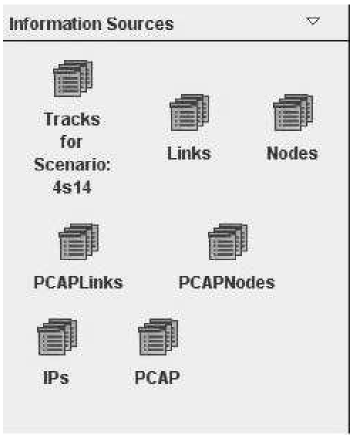


Figure B.6: The database tables are present in the information source box ready to be loaded into an information layout

At this point, the analyst is ready to choose the information layout. Chapter IV includes a detailed example of loading information layouts and designing the visualization.

Bibliography

1. Amoroso, Edward. *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response.* Intrusion.Net Books, 1999. ISBN 0-9666700-7-8.

2. Answers. "Protocol-based intrusion detection system — Answers.com", 2008. URL `http://www.answers.com/topic/protocol-based-intrusion-detection-system`.

3. Avitia, Serafin, Stuart Kurkowski, and Luke van der Hoeven. "Interactive Visualization of Fused Intrusion Detection Data". *Proceedings of the 3rd International Conference on Information Warfare and Security (ICIW)*. 2008.

4. Blustein, James, Ching-Lung Fu, and Daniel L. Silver. "Information visualization for an intrusion detection system". *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia 2005*, 278–279. ACM, New York, NY, USA, 2005. ISBN 1-59593-168-6.

5. Board, National Foreign Intelligence. "2002 National Intelligence Estimate", 2002. URL `http://www.fas.org/irp/cia/product/iraq-wmd-nie.pdf`.

6. de Boer, Pieter and Martin Pels. "Host-Based Intrusion Detection Systems", February 2005. URL `http://staff.science.uva.nl/~delaat/snb-2004-2005/p19/report.pdf`.

7. Boyd, John. "CSCE 525 Chapter 1 slides", 2007.

8. Cid, Daniel B. "OSSEC Official Website", 2008. URL `http://www.ossec.net/main/ossecteam`.

9. Conti, Greg. "Security Data Visualization: Graphical Techniques for Network Analysis", 2007.

10. Cumiford, Leslie D. "Situation Awareness for Cyber Defense", 2006.

11. D'Amico, Anita and Michael Kocka. "Information Assurance Visualizations for Specific Stages of Situational Awareness and Intended Uses: Lessons Learned". *Proceedings of the IEEE Workshop on Visualization for Computer Security 2005*, 107–112. Secure Decisions, Northport, NY, USA, 2005. ISBN 0780394771.

12. Endsley, Mica R. "Theoretical Underpinnings of Situation Awareness: A Critical Review", 2000.

13. Foresti, Stefano, Yarden Livnat, Shaun Moon, and Robert Erbacher. "Visual Correlation of Network Alerts". *Proceedings of the IEEE Computer Graphics and Applications Volume 26*, 48–59. March 2006.

14. Fullmer, Mark and Steve Romig. "The OSU Flow-tools Package and Cisco NetFlow Logs". *Proceedings of the 14th Systems Administration Conference 2000*, 85–89. USENIX Association, New Orleans, LA, USA, 2000. URL http://www.usenix.org/events/lisa00/full_papers/fullmer/fullmer.pdf.

15. Inc., Enterasys Networks. "Dragon IDS", 2008. URL http://www.enterasys.com.

16. Inc., Sourcefire. "Snort.org Official Website", 2008. URL http://www.snort.org.

17. Lakkaraju, Kiran, William Yurcik, and Adam J. Lee. "NVisionIP: netflow visualizations of system state for security situational awareness". *Proceedings of the ACM workshop on Visualization and data mining for computer security 2004*, 65–72. ACM, New York, NY, USA, October 2004. ISBN 1-58113-974-8.

18. Livnat, Yarden, Jim Agutter, Shaun Moon, Robert F. Erbacher, and Stefano Foresti. "A Visualization Paradigm for Network Intrusion Detection". *Proceedings from the Sixth Annual IEEE on the Systems, Man and Cybernetics (SMC) Information Assurance Workshop 2005*, 85–89. Scientific Computing and Imaging Institute, June 2005. ISBN 0-7803-9290-6.

19. McClure, Stuart, Joel Scambray, and George Kurtz. *Hacking Exposed Third Edition*. Osborne/McGraw-Hill, 2001. ISBN 0-07-219381-6.

20. McPherson, Jonathon, Kwan-Liu Ma, Paul Krystosk, Tony Bartoletti, and Marvin Christensen. "PortVis: A Tool for Port-Based Detection of Security Events". *Proceedings of ACM Workshop on Visualization and Data Mining for Computer Security 2004*, 73–81. Association for Computer Machinery, October 2004.

21. van Riel, Jean-Pierre and Barry Irwin. "InetVis, a visual tool for network telescope traffic analysis". *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa 2006*, 85–89. ACM, New York, NY, USA, 2006. ISBN 1-59593-288-7.

22. Salerno, John, Michael Hinman, and Douglas Boulware. "Building A Framework For Situation Awareness". *Proceedings of the International Conference on Information Fusion 2004*, 219–226. International Society of Information Fusion, Stockholm, Sweden, June 2005. ISBN 917056115.

23. Salerno, John, Michael Hinman, and Douglas Boulware. "A Situation Awareness Model Applied To Multiple Domains". *Proceedings of SPIE Vol. 5813 on the Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2005*, 65–74. International Society for Optical Engineering, Bellingham, WA, USA, March 2005.

24. SecureDecisions. "VIAssist: Visual Assistant for Information Assurance Analysis", November 2007. URL http://www.securedecisions.com/viassist.

25. Systems, Stanfield. “Visual Information Management Toolkit Installation Instructions & Version Description Document”, 2007.

26. Systems, Stanfield. “Visual Information Management Toolkit Software Design Specification”, 2007.

27. Tadda, George, John Salerno, Douglas Boulware, Michael Hinman, and Samuel Gorton. “Realizing situation awareness in a cyber environment”. *Proceedings of SPIE Vol. 6242 on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, 65–72. International Society for Optical Engineering, Bellingham, WA, USA, April 2006. ISBN 1-58113-974-8.

28. Tufte, Edward R. *The Visual Display of Quantitative Information.* Graphics Press, 2001.

29. University, Carnegie Mellon. “CERT Statistics”, 2008. URL `http://www.cert.org/stats/`.

30. Wikipedia. “Application protocol-based Intrusion Detection System — Wikipedia, The Free Encyclopedia”, 2008. URL `http://en.wikipedia.org/wiki/Application_protocol-based_intrusion_detection_system`.

31. Yin, Xiaoxin, William Yurcik, Michael Treaster, Yifan Li, and Kiran Lakkaraju. “VisFlowConnect: netflow visualizations of link relationships for security situational awareness”. *Proceedings of the ACM workshop on Visualization and data mining for computer security 2004*, 26–34. ACM, New York, NY, USA, 2004. ISBN 1-58113-974-8.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 19–06–2008	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sept 2006 — Jun 2008

4. TITLE AND SUBTITLE Developing Network Situational Awareness through Visualizations of Fused Intrusion Detection System Alerts	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Serafin Avitia V, Mr.	**5d. PROJECT NUMBER** 08-290
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GCS/ENG/08-23
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Mr. George Tadda AFRL/RIEA 525 Brooks Road Rome, NY 13441-4503 315-330-3957 George.Tadda@rl.af.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

With networks increasing in physical size, bandwidth, traffic volume, and malicious activity, analysts are experiencing greater difficulty in developing network situation awareness. Traditionally, network analysts have used Intrusion Detection Systems to gain awareness but this method is outdated when analysts are unable to process the alerts at the rate they are being generated. Analysts are unwittingly placing the computer assets they are charged to protect at risk when they are unable to detect these network attacks. This research effort examines the theory, application, and results of using visualizations of fused alert data to develop network situational awareness. The fused alerts offer analysts fewer false-positives, less redundancy and alert quantity due to the pre-processing. Visualization offers the analyst quicker visual processing and potential pattern recognition. This research utilized the Visual Information Management toolkit created by Stanfield Systems Inc. to generate meaningful visualizations of the fused alert data. The fused alert data was combined with other network data such as IP address information, network topology and tcpdump data.

15. SUBJECT TERMS

IDS Visualization, Network Situational Awareness, IDS Analysis, Network Defense, Network Information Visualization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Lt Col Stuart H. Kurkowski (ENG)
U	U	U	UU	98	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255–3636 x7228, Stuart.Kurkowski@afit.edu