NAVSEC : A Recommender System for 3D Network Security Visualizations

Troy Nunnally
CAP Group
Georgia Institute of Technology
Atlanta, GA 30332,USA
troy.nunnally@gatech.edu

Kulsoom Abdullah
CAP Group
Georgia Institute of Technology
Atlanta, GA 30332,USA
kulsoom@gatech.edu

A. Selcuk Uluagac
CAP Group
Georgia Institute of Technology
Atlanta, GA 30332,USA
selcuk@ece.gatech.edu

John A. Copeland
CSC Lab
Georgia Institute of Technology
Atlanta, GA 30332,USA
jcopeland@ece.gatech.edu

Raheem Beyah
CAP Group
Georgia Institute of Technology
Atlanta, GA 30332,USA
rbeyah@ece.gatech.edu

ABSTRACT

As network attacks increase in complexity, the ability to quickly analyze security data and mitigate the effect of these attacks becomes a difficult problem. To alleviate these challenges, researchers are looking into various two-dimensional (2D) and three-dimensional (3D) visualization tools to detect, identify, and analyze malicious attacks. These visualization tools often require advanced knowledge in networking, visualization, and information security to operate, navigate, and successfully examine malicious attacks. Novice users, deficient in the required advanced knowledge, may find navigation within these visualization tools difficult. Furthermore, expert users may be limited and costly. We discuss the use of a *modern recommender* system to aid in navigating within a complex 3D visualization for network security applications. We developed a visualization module called NAVSEC, a recommender system prototype for navigating in 3D network security visualization tools. NAVSEC recommends visualizations and interactions to novice users. Given visualization interaction input from a novice user and expert communities, NAVSEC is instrumental in reducing confusion for a novice user while navigating in a 3D visualization. We illustrate NAVSEC with a use-case from an emulated stealthy scanning attack disguised as a file transfer with multiple concurrent connections. We show that using NAVSEC, a novice user's visualization converges towards a visualization used to identify or detect a suspected attack by an expert user. As a result, NAVSEC can successfully guide the novice user in differentiating between complex network attacks and benign legitimate traffic with step-by-step created visualizations and suggested user interactions.

VizSec '13 October 14 2013, Atlanta, GA, USA

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General - Security and protection; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Design, Security, Algorithms

Keywords

Recommender System, Network Security, Monitoring, 3D Visualization, NAVSEC, Security Visualization

1. INTRODUCTION

Network administrators often evaluate security risks and malicious activity in the Internet Protocol (IP) traffic, intrusion detection systems (IDSs), firewalls, and host systems on a network. Previously, administrators examined network activity and behavior using textual representations [10, 1]. Textual representations become overwhelming as data volume increases and networks become more complex. As a result, vital data could be overlooked and data analysis could be lengthy.

Recently, researchers have been looking into techniques to convert abstract network data into visual 2D and 3D representations to quickly discover and identify malicious activity and network behavior [8, 18, 19]. They use the visual system's ability to organize large amounts of information to efficiently represent network characteristics. In this way, humans can recall and evaluate visual representations faster and more accurately than textual data.

There is a large body of work using 2D visualizations to visualize IDS logs, network management systems, and firewalls [11]. However, as the amount of information continues to increase, 2D visualizations can also be perceived as cluttered and limited [2]. Researchers address this issue by introducing interaction techniques such as linking and brushing [12] to manipulate a 2D visualization environment. Another approach is adding the z-direction (i.e., 3D) to allow more information to be visualized vs. its 2D counterparts [23].

However, both the addition of the z-direction and usage of interaction techniques increase the complexity of visualization environments. For instance, many of today's network security applications require a user to perform many interactions within a user interface (UI). These users must navigate through a vast visualization environment to successfully formulate accurate decisions about a network. These large number of possible interactions could overwhelm or confuse a user [4]. As a result, a user may be hindered from effectively navigating through visualizations and identifying vital network incidents (e.g., stealthy port scans and DDOS attacks). Furthermore, finding the visualization steps to complete critical tasks becomes difficult to accomplish and could take years to master [6].

Expert users could teach and guide novice users to use 3D visualization network security tools to find peculiar network activity and new attacks on the network. These expert users are technically skilled and understand the network environment and 3D visualization tools, allowing them to identify malicious attacks and make accurate decisions. However, the number of expert users could be limited and utilizing these expert users could be costly and resource intensive. Thus, tools are needed to assist novice users in navigating within complex visualization environments for network security applications and to help the novice users quickly accomplish specific tasks such as identifying attacks.

In this paper, we introduce a novel recommender system for 3D network security visualization tools called NAVSEC. Specifically, the *nearest neighbor* recommendation algorithm [16] is used in NAVSEC to aid novice users in navigating 3D visualizations. The nearest neighbor approach is commonly used in predictive analysis and machine learning algorithms to recognize patterns in datasets. It is also used in collaborative filtering algorithms for recommending movies. Collaborative filtering is the process of filtering for information such as user interactions for a 3D visualization using predictive methods (e.g., nearest neighbor, support vector machines, neural networks) involving collaboration among multiple agents (e.g., users of a visualization tool). These methods can be used as part of our system to assist users in learning how to discover interesting network attacks. Also, we implemented NAVSEC as a module in our earlier 3D visualization tool, P3D [18], for network security applications. To our knowledge, no work has been done in developing a recommender system to help a novice user make intelligent decisions about network attacks in 3D visualization applications. In this work, we focus on the case of navigating through 3D visualizations for novice users, but expect that our recommender system to be also applicable in complex visualizations or interaction environments.

The rest of this paper is organized as follows. A background and motivation on navigation in 3D visualizations for network security applications is presented in Section 2. Next, related work is discussed in Section 3. We discuss the details of NAVSEC design for assisting users navigating in 3D environments in Section 4. Next, we illustrate NAVSEC with an emulated use-case of a stealthy network-based scan disguised as an ftp transfer on a local area network in Section 5. Finally, we conclude the paper and discuss future work in Section 6.

2. BACKGROUND AND MOTIVATION

3D network security applications [24, 20] use visualization techniques to extend the visualization space and show more network activity than a single vantage point. Since displays are physically constrained to 2D devices (e.g., computer monitor), these 3D environments use visual cues such as shadowing and size to adequately represent depth. Most 3D visualization techniques for the network security field encompass a long range of networking data attributes such as IP address, port number, and TCP/IP protocol than can be viewed from a single vantage point. Thus, some visualization designers employ a *top-down* approach [13]. A top-down approach occurs when network administrators start at an initial visualization that reveals a single overview of the network and manipulate the visualization environment to discover more specific details to potentially detect network attacks. Other researchers use a *bottom-up* approach [5], which starts from an initial node on a network and expands to show the relationship between other nodes. This approach is useful where the dataset is too large for a top-down approach. In both top-down and bottom-up approaches, obtaining different views of the network requires a user to perform interactions that manipulate and navigate within these visualization environments. However, in many network security visualization tools, there could be hundreds of possible interaction combinations for a user to navigate [9]. For a novice user possessing little or no expertise, executing a complex task using a visualization tool, such as identifying a network attack, becomes difficult. Thus, a novice user may need assistance in performing the correct sequence of interactions. We propose to solve this problem by developing a system that could be used as a module for 3D visualization tools to recommend both visualizations and interactions for a network-based attack using a recommendation algorithm. This recommendation system recognizes similar patterns found in current and historical interaction data from a group of expert users and recommends a set of interactions to a novice user. Our recommendation system uses the nearest-neighbor machine learning algorithm to identify unknown attacks. We use a current user's and a group of expert users' past behavior to assist users in determining a visualization to use for identifying an attack and to aid a novice user in navigating and manipulating 3D visualizations for network security applications. This assistance in navigation leads to the reduction of interactions used to detect various network activity and guide a novice user to certain objectives such as detecting a stealthy network-based attack.

Navigating within a 3D visualization for network security applications can be difficult because a large amount of network data is portrayed in a screen-size visualization. Moreover, visualizations offer a limited amount of navigational information [16]. There are hundreds of details to track and one mistake could result in a misinterpretation of an attack that is being visualized. In addition, different interactions could be utilized to navigate through a 3D environment as well as to extend the visualization space. For instance, possible interactions include zooming in and out of the visualization, rotating about a pivot point, panning, or shifting the camera view of the visualization environment. Another interaction used to extend the 3D environment is the addition of a 2-axes plane that plots other network attributes such as average packet size vs. total number of packets. Our system, NAVSEC, collects sequences of interactions from a group of expert users and stores them into a database. The

active user's current and past interactions are also collected. The nearest-neighbor approach using cosine similarity generates both suggested visualizations and next interaction for a novice user based on the collected interactions. Using NAVSEC, users (specifically novice users who are exposed to unfamiliar visualizations) are recommended possible interactions to extend 3D visualizations and become aware of interaction options to detect attacks more effectively in their future use.

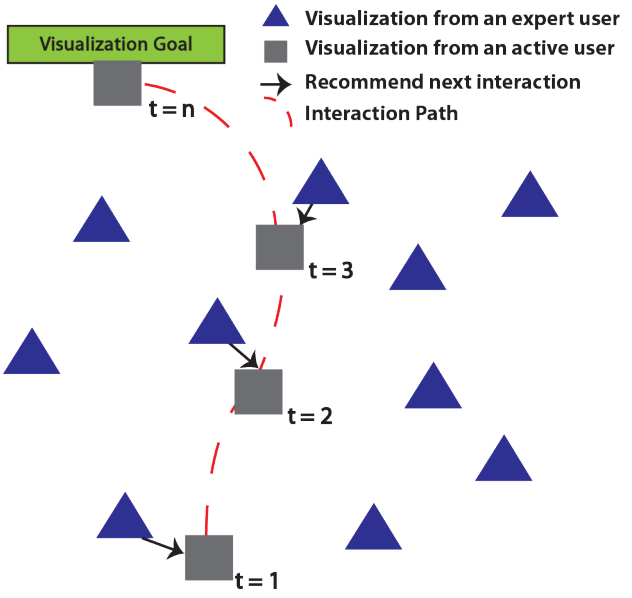

Figure 1: Nearest-neighbor approach for a network attack.

As mentioned previously and shown in Figure 1, our recommender system applies the nearest-neighbor approach [16] to an active (current) user's recent set of interactions at instance t (state of the active user) for an attack (e.g., DoS or an advanced stealthy port scan meant to bypass a firewall or subvert an IDS) as denoted by the square. This state is calculated using the active user's current set of interactions. At instance t=0, an initial visualization is presented to an active user. This visualization could be a representation of real-time network data or offline network dataset from a packet capture file. From this initial visualization, the user can speculate an attack, but needs further investigation by choosing an attack type portrayed as a button located in the tools palette on the visualization interface. As the active user interacts with a visualization tool, the active user's set of interactions at instance t is compared to sets of interactions from expert users. In our work, we assume expert users are knowledgable and successful in producing visualizations from a set of interactions to identify an attack. Each set of interactions is a sequence of interactions for the same type of attack speculated by the active user as denoted as triangles in Figure 1. The system picks the nearest-neighbor and recommends this set of interactions as denoted by the arrows until a visualization goal is reached. The recommendation system uses the number of occurrences of each interaction type between a novice user and group of expert users and computes the similarity to a set of novice user's interactions using a cosine similarity function. The visualization goal in Figure 1 is a visualization used by expert users to identify or detect a speculated attack. Furthermore, if an active user does not perform any interactions (e.g., instance t=0), but wishes to be recommended a set of interactions, the most commonly-used sequence of interactions produced by the group of expert users for an attack are used.

Depending on the active user's behavior, our recommender system dynamically generates the most similar set of interactions or visualization from a group of expert users. Our intention for this system is to recommend a set of interactions used to extend or manipulate the visualization space so an active user is guided along a path to reach a visualization goal at instance n. Our system only recommends how a novice user can reach a speculated attack based on expert users' interactions and does not promise to find any attack.

3. RELATED WORK

3.1 Recommender Systems

Recently, recommender systems have been used in recommending products and services such as movies, books, music that best match a user's interest correlated with users of similar interests to increase sales. However, these tools do not require nor use information on the correlations between the behavior of the active user and expert users. The novelty of our tool is in combining the interaction behavior of both the expert users and active user to assist in discovering network-based attacks. Other systems (e.g., [16]) recommend a single interaction for software applications such as AutoCAD. On the contrary, our work recommends a sequence of interactions, which can be executed by a single advanced action. As a result, our technique is instrumental in reducing the number of interactions a novice user might use to render a visualization technique. Our technique can lead to attack discovery with less number of interactions and to the efficient utilization of resources (e.g., memory and CPU utilization).

3.2 3D Network Security

Existing 3D visualizations have been created to visualize data using techniques such as iconic tree structures, bar charts [24, 20], and 3D scatter plots [14]. For example, PortVis [17] is a visualization tool that aids in detecting large-scale network security events and port activity. Also, NetBytes Viewer [22] visualizes the historical network flow data per port of an individual host machine or subnet on a network using a 3D impulse graph plot. These tools only consider the 4-tuple: source IP, destination IP, source port, and destination port. Thus, these tools show a small amount of detail and only display the counts of activities rather than the activities themselves. This lack of detail could lead to a misinterpretation of network attacks. Also, NetBytes Viewer is static; hence, new visualizations cannot be derived from NetBytes Viewer to detect interesting attacks. However, with integration of NAVSEC and NetBytes Viewer, we can provide a foundation to recommend new visualizations and help novice users of NetBytes to detect more types of attacks.

Furthermore, research has been performed in detecting unknown large-scale Internet attacks including Internet worms, DDoS attacks and network scanning activities using a parallel coordinate system [3]. Parallel coordinate attack visualization (PCAV) [3] uses hash algorithms to detect nine graphical signatures, and a parallel 3D coordinate system for

network security (P3D) [18] extends the visualization space by introducing a stereoscopic awareness region mechanism using 3D glasses to highlight important data and expand the visualization to help prevent occlusions. NAVSEC can be uniquely integrated as a module to reduce the amount of interactions used in these visualizations without any guidance. Moreover, NAVSEC focuses on aiding novice users by examining the interaction space rather than the visualization space.

Rasmussen [21] introduces NIMBLE, an incident management system through visualization. NIMBLE calculates the similarity for given IDS alerts and historical alerts classified by network administrators. His results show improved network analyst's accuracy in defending the network with the tool's visual display and the given recommendations. His primary visualization focuses on correlated IDS output rather than network traffic. Our recommender system is more comprehensive as it also accounts for understanding of attacks meant to mislead the user or subvert an IDS.

The Spinning Cube of Potential Doom [14] uses 3D scatter plots to represent network activity on three axes: the destination IP of the local network on the x-axis, the destination port on the y-axis, and the source IP on the z-axis. The color of the glyphs distinguishes the type of the connection (e.g., UDP or TCP). Their 3D scatter plots are useful in determining interesting patterns such as clusters or correlations for data using five parameters: source and destination IPs, source and destination ports, and connection type. Since the visualization is limited to five parameters, decoys cannot be detected without visualizing more parameters such as TCP flags and flow data. As a result, a deeper analysis of scanning behavior is not possible. NAVSEC addresses these limitations by visualizing and incorporating more data, allowing it to help uniquely characterize port scans and further understand scanning activity. In addition, NAVSEC uses a knowledge base of expert users so that as different attacks occur, the most similar visualization could dynamically be generated using our recommender system. As more expert users utilize the system, thereby increasing the expert user interaction database, the most similar visualization to the active user's behavior may evolve over time.

4. SYSTEM OVERVIEW

As mentioned in Section 2, our goal is to present a novice user with recommendations of a set of user interface (UI) interactions that produce visualizations that aid in effectively navigating within 3D visualizations. This set of interactions is shown as a textual list or as a visualization produced by automatically performing the listed interactions. NAVSEC provides the textual interactions as an option so that the user can iterate though the list for more detail on the experts reasoning in creating a visualization. This feature could also be conducive in education/training situations for allowing instructors to show practical examples of various expert users' identification of attacks. Showing a visualization rather than textual interactions allows the active user to view the visualization from an expert user by simply clicking a button. This feature reduces the amount of the interactions needed from an active user.

4.1 System Design

We design our recommender system as a modular web-based system, which can incorporate various machine learning techniques (e.g., nearest neighbors, support vector machines, neural networks) as modules. A developer can extend our recommender system by developing a custom module to meet their specific requirements. As a result, this modular approach increases the flexibility of the system. Furthermore, our system allows for multiple expert and active users to connect to a centralized web-server. Our system could potentially be expanded to accommodate load balancing and redundancy, common techniques used for reliability and scalability. Moreover, our system can support different expert user communities such as colleagues in a small security organization where resources are limited or training inexperienced military personnel. Additionally, we design the user interface to be unobtrusive; it does not force the user to respond to recommendations before continuing their work. To implement this system, we deploy a distributed system for both expert users and novice users. A light-weight visualization tool is deployed to each user with the NAVSEC module installed and the module connects to a centralized server for computation. As illustrated in Figure 2, the NAVSEC system consists of 4 components: *Active User*, *Expert User Community*, *Interaction Database*, and *Recommender*. We expand on each of these components in the following subsections.

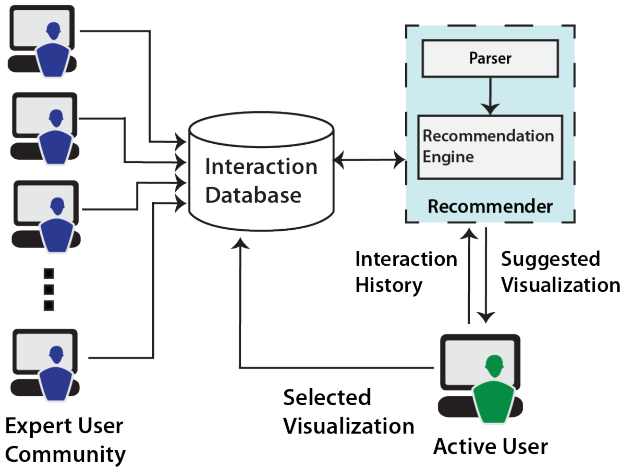

Figure 2: System Design of NAVSEC.

4.1.1 *Active User*

The active (i.e., current user) is defined as the one navigating the visualization tool with the NAVSEC module installed. In our work, the active user is a novice that may be confused on the next set of interactions or visualization to be performed to successfully identify an attack.

4.1.2 *Expert User Community*

The Expert User Community is a set of users with significant experience in the network security and visualization fields. An expert user could be colleagues in a security organization or IT department. These users interact with a visualization tool (e.g., P3D [18]), create visualizations used to detect or identify an attack on a network. These users categorize the visualizations by clicking on an attack category button located in the tools palette. In our work, we assume an expert has sufficient knowledge to perform the correct sequence of interactions to detect an attack. The sequence of

interactions (i.e., zoom, pan, translate, add plane) for each expert user and categorized attack is recorded and sent to the centralized Interaction Database prior to any active user activity.

4.1.3 Interaction Database

The Interaction Database is a collection of interaction sequences. Also, the Interaction Database stores the amount of interactions occurred by both expert and active users for each attack. This amount is used by the Recommender component to compute the popular set of visualizations and interactions for specific network attacks. The Recommender also receives interactions as input from an active user via HTTP requests and stores them into the Interaction Database. The data from the Interaction Database contains the interaction identification number which is a number that maps to the type of interaction performed by the active user. This identification number is used to convert the sequence of interactions into readable text. The Recommender performs a *cosine similarity analysis* on the active user data and suggests a set of interactions to find a particular type of attack. The next sub-section further discusses how cosine similarity analysis is used to compare an active user's set of interactions to a group of expert users' interactions and then finds the most similar set of interactions from the group of expert users. This set of interactions can be used to generate visualizations for the active user. After the active user performs the suggested sequence of interactions, the sequence is sent to the Interaction Database for future use.

4.1.4 Recommender

The Recommender parses the data contained in the Interaction Database and computes a set of interactions for recommendation to an active user in real-time as listed in Algorithm 1. This computation is performed by the Recommendation Engine (Figure 3). The Recommendation Engine uses two inputs: the interaction history of expert users and the current history of interactions of an active user. Then, Algorithm 1 *ComputeSetOfRecommendations( )* is used to calculate a recommended set of interactions from the two inputs. Our algorithm for recommending a set of interactions is divided into three parts: formulate an interaction vector, compute a similarity matrix, and recommend a set of interactions.

Our recommender system formulates an interaction vector (lines 2-8) by using the total number of occurrences n_j of the interaction type j for the kth attack session s_k. The interaction type is the categorized interaction performed by the user to manipulate a visualization such as *add left plane, zoom, rotate, add right plane*. An attack session is a sequence of interactions to visualize an attack. Furthermore, a user can perform multiple sessions for a type of specific attack a. For example, an expert user could visualize 10 port scans (K=10) where k is the session identifier and K is the total number of sessions for an attack. As given in Equation 1, each user's number of occurrences n_j is tabulated into an interaction vector $\vec{v_k}$ for the kth session where k is the unique identifier for each session. In Equation 1, n_j is the summation of each unique occurrence i of interaction type j for attack session s_k. Each interaction type j corresponds to the position of interaction vector $\vec{v_k}$ where J is the total number of interaction types. Thus, the total number of dimensions within an interaction vector $\vec{v_k}$ is equivalent to the total number of interaction types J. Each interaction vector $\vec{v_k}$ for all expert users is stored in the Interaction Database.

Algorithm 1 ComputeSetOfRecommendations( )

```
1: begin
2: % Formulate an interaction vector v_k.
3:   v_k = [n_0 n_1 ··· n_{J-1}]
4:   for (i = 0 to total_Number_Interactions_{s_k})
5:     if (typeOfInteration(i) == j)
6:       return n_j ++
7:     end if
8:   end for
9: % Compute a similarity matrix M.
10:  M = [M_0 M_1 ··· M_{K-1}]
11:  for (k = 0 to K − 1)
12:    M_k ← cos(v_k, v_h)
13:    return M_k
14: % Recommend a set of interactions.
15:  for (k = 0 to K − 1)
16:    if M_k > max(M)
17:      return session_k
18:    end if
19:  end for
20:  return Interaction_sequence_{s_k}
21: end
```

$$\vec{v_k} = [n_0\ n_1\ \cdots\ n_{J-1}]$$
$$where\ n_j\ =\ \sum_{i=0}^{I-1} i_j\ for\ attack\ session\ s_k. \tag{1}$$

Next, we use the cosine similarity function to produce a similarity matrix (Equation 2). This similarity matrix is derived from comparing the active user's interaction vector $\vec{v_h}$ to each expert user's interaction vector $\vec{v_k}$ for all sessions.

$$M_k = cos(\vec{v_k}, \vec{v_h}) = \frac{\vec{v_i} \cdot \vec{v_h}}{\|\vec{v_i}\| * \|\vec{v_h}\|} \tag{2}$$

Our recommendation algorithm loops through (lines 9-13) a set of interaction vectors for each session within the attack. Each interaction vector $\vec{v_k}$ of expert user performs a similar distance function such as cosine similarity to an active user current session $\vec{v_h}$. The resulting value M_k is stored into a user similarity matrix $\vec{M}$. The resulting similarity values stored in $\vec{M}$ ranges from 0-1. The value 0 means that vectors are orthogonal to each other and the set of interactions is not related. The value 1 means the vectors are collinear or the set of interactions is similar. Thus, values close to 1 indicate $\vec{v_h}$ is similar to $\vec{v_k}$.

Finally, we recommend a set of interactions (lines 14-20) by computing the closest score to 1 after taking the maximum value of similarity matrix $\vec{M}$. Each position in the similarity matrix $\vec{M}$ is mapped to a session identification number k and sequence of actions can be determined by selecting the sequence of interactions from the session identification number. Then, the sequence of interactions from the expert user session can be sent back to active user as the recommended set of interactions.

4.2 Implementation

We implemented NAVSEC as a module for a Parallel 3D coordinate system (P3D) [18] and illustrate the functionality of NAVSEC with a use-case scenario for advanced stealthy

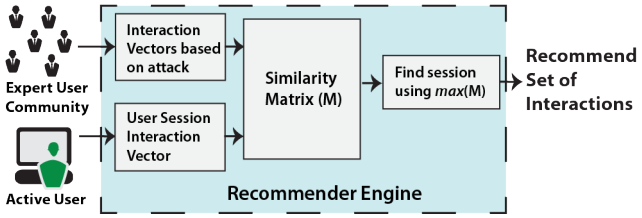

Figure 3: Design of Recommendation Engine.

port scans. P3D is used because unlike most 3D counterparts this tool has no theoretical limit in the number of network parameters that can be visualized, hence better able to detect visualization attacks [4] vs. a 2D/3D scattered plot matrix [3], uses stereoscopic 3D support and interactive techniques such as zooming and panning. P3D uses 2D YZ planes of network attributes positioned along the x-axis and shows the relationship of the attributes using colored lines. Each colored line represents the type of connection (e.g., TCP, UDP). To incorporate the NAVSEC into P3D, we extended P3D by modifying the interaction layer of FRE3DS [19] for both expert and active users. As discussed, NAVSEC produces a recommended set of interactions or visualizations. This set of recommended interactions are displayed in a web-based interface that the user can refer to when convenient. NAVSEC can be also displayed as a peripheral tool palette located within the P3D user interface. This tools palette contains a list of network attacks in the form of buttons. If a user clicks on an attack, the suggested visualization is presented to the user.

NAVSEC contains a client-side C++ component which is integrated as a module to send GET HTTP request of interactions to the server-side application. The server-side uses an Application Programming Interface (API) to receive HTTP requests. The NAVSEC API is developed using Codeigniter, a PHP framework used for rapid web development [15]. Codeigniter uses a Model-View-Controller architecture design to assist in code reusability so that the NAVSEC module code can be easily integrated with other visualization tools.

5. EVALUATION

In this section, we evaluate NAVSEC with a use-case scenario for stealthy port scanning attacks. We use stealthy port scanning attacks because these attacks are commonly used to bypass firewalls, subvert IDSs, and could often be misinterpreted by network administrators and are commonly used by attackers. Next, we discuss a novice user's confusion between visualizing a stealthy port scan disguised as a FTP scan and a file transfer using multiple concurrent TCP connections. A concurrent FTP transfer occurs when a client and server creates multiple TCP connections to increase total throughput of the file transfers. This network activity is commonly implemented by clicking the "Enable Concurrent/Multiple Connections for transfers" in FTP clients (e.g, Filezilla [7]). We show that NAVSEC can help avoid this confusion. We evaluate the number of interactions performed when NAVSEC is enabled vs. disabled. Finally, we show the convergence of interactions to a high similarity score, which denotes that an active user set of interactions approaches an expert user's set of interactions.

5.1 Stealthy Port Scanning Use-Case

Stealthy port scanning is a network attack used to discover exploitable communication channels by probing for vulnerable services in a form that goes undetected by traditional intrusion detection systems. Most network scanner tools contain the ability to forge various packets (e.g., RST packets) from a spoofed source and destination IP addresses as though they were coming from protected hosts behind the firewall. Also, a highly skilled attacker can perform scans that emulate legitimate network traffic. Although current visualization IDSs detect most scanning activity, more advanced attackers can perform stealthy port scanning attacks to subvert IDSs. Therefore, network administrators must distinguish stealthy attacks from legitimate network traffic.

As shown in Figure 4, we emulate a concurrent FTP file transfer of ten 20 Megabyte files using Filezilla [7] on the Windows Operation System (OS). The source IP 57.25.6.30 is attempting a TCP connection from 10 ephemeral ports (50332-50341) to 10 ephemeral ports (53829-53837, 53850) on a destination IP 57.25.6.100. The source IP is also attempting to connect from 18 ephemeral source ports to port 21 (default port for FTP service) on 57.25.6.100. These connections occur because two connections are used to initiate a FTP connection and send FTP commands for each data transfer. By examining this initial visualization, both a novice and expert user could suspect that an attacker is performing a stealthy scan from 10 multiple consecutive source ports to 10 destination ports rapidly, but below most IDSs scanning rate thresholds.

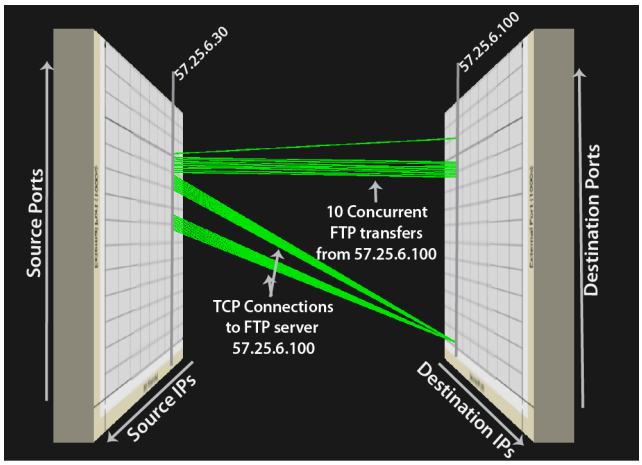

Figure 4: A potential stealthy port scan.

However, since an expert is familiar with P3D and network security, the expert can examine the network data further. The expert extends the visualization by performing interactions with the P3D interface such as color coding the lines to specific network protocol to keep the consistency of the visualization and adding a left plane with the z-axis as total size of packets in a TCP flow and the y-axis as number of packets in a TCP flow. By examining this TCP flow, the expert can determine the number of packets and size of packets in a TCP connection to evaluate if the connection is actually a port scan. Figure 5 shows an extension of P3D where the average size of the packets and the number of packets sent by the source ephemeral ports are high (1438 bytes and

147304 packets, respectively). Since a large amount of data is sent, the source node can be interpreted as a potential FTP connection in passive mode with multiple concurrent connections to increase FTP server's throughput rather than a port scan.

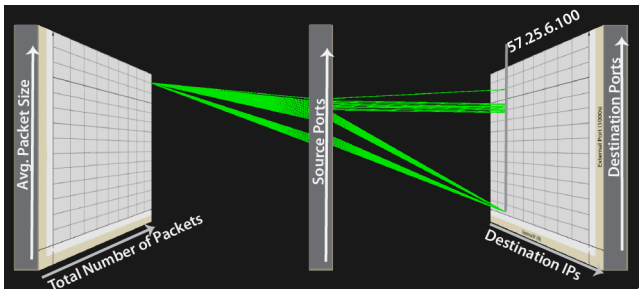

Figure 5: P3D [18] Multiple concurrent FTP file transfer.

On the contrary, if the total number of packets in a TCP connection and average packet size from the source are low (one packet) as illustrated in Figure 6, the visualization is potentially a stealthy port scan. An attacker can send packets with data in the payloads to further confuse the administrator. However, an expert user could further evaluate the scenario by adding a plane to show the number of ACK packets versus RST packets. If no ACK packets are being sent from the server, then a legitimate connection has not been established.

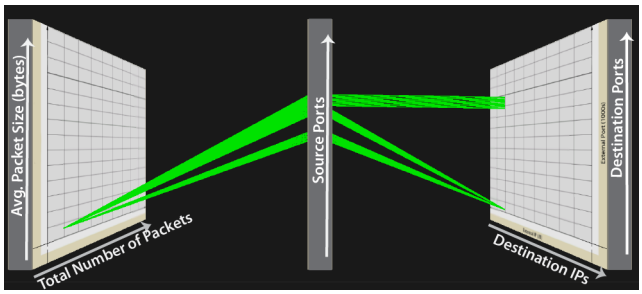

Figure 6: P3D [18] Steathy Scan.

An expert user can successfully distinguish attacks from legitimate traffic by expanding and manipulating the P3D environment. However, the ability to successfully and efficiently expand this visualization requires the user to possess an advanced background in network security and experience with the P3D visualization system. From the initial analysis in Figure 4, the novice may view multiple connections to multiple destination ports and mistakenly assume the network activity is a port scan. Moreover, due to the novice user's basic visualization knowledge, the novice may not have the knowledge to extend the visualization. Using NAVSEC system as explained in Section 4, the novice can use the expert user's suggested visualization by clicking the "port scan" button in the attack tools palette. By clicking on this button, P3D uses the recommender algorithm and input from group of expert users via the NAVSEC module to produce the most recommended visualization based on the cosine similarity function.

5.2 User Interaction Convergence Evaluation

In this section, we show the convergence of five sets of arbitrary instances of an active user session. As discussed in Section 2, an instance is denoted by the current set of interactions that was performed by the active user. For each instance, the recommended next interaction is performed using NAVSEC. Next, we compute the maximum similarity score after each recommended instance and plot it onto the graph as shown in Figure 7. If the maximum similarity score continues to converge towards the value 1 after each consecutive step in the set of performed interactions, then the visualization produced by the active user is becoming similar to a visualization used to identify or detect a suspected attack by an expert user. Thus, as illustrated in Figure 1, NAVSEC is guiding an active user to a visualization completed by an expert user.

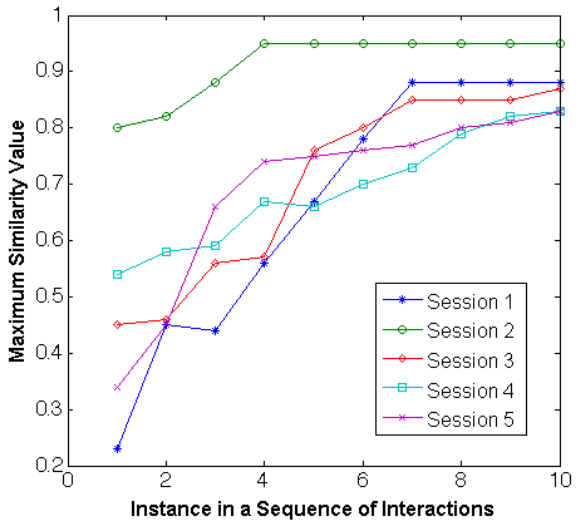

Figure 7: Convergence of Interactions

As shown in Figure 7, five sessions from active users is arbitrarily chosen from a group of active sessions. These sessions are compared to 40 emulated interaction vectors performed by expert users. Each interaction vector contains a set of 30 types of interactions (e.g., zoom out, zoom in rotate, add left plane, add line glyphs). The line graph shows that as the set of interaction increases, each active session is converging towards a value of 1 for the similarity score. The highest value is .95. This result suggests that with the use of NAVSEC, visualizations for the P3D tool converges towards an expert user's interaction set.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we developed and introduced NAVSEC to assist users in navigating 3D visualizations and to reduce the possible number of interactions within a visualization. Although there have been several studies on 2D/3D visualization techniques for network analysis, there has been little work on addressing the issue of navigation complexity within these visualization techniques and understanding and analyzing scans or attacks that could possibly mislead novice users [4]. NAVSEC uses advanced visualization recommendation techniques based on a database of interaction

sequences of an expert community to (1) enhance the novice user's experience, (2) provide easier understanding of 3D network security, and (3) perform faster analysis of the network data thereby increasing efficiency. We tested NAVSEC using P3D and FRE3DS framework [19] to reveal vital characteristics of a node on a network. Specifically, we showed that using NAVSEC, novice users are less likely to be confused when discovering advance attacks. In the future, we plan to apply our visualization design to the IPv6 address space, implement more user interactions, and evaluate their effects on real users.

7. REFERENCES

[1] S. Al-Mamory, A. Hamid, A. Abdul-Razak, and Z. Falah. String Matching Enhancement for Snort IDS. In *Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pages 1020–1023, Dec. 2010.

[2] W. Ark, C. D. Dryer, T. Selker, and S. Zhai. Representation Matters: The Effect of 3D Objects and a Spatial Metaphor in a Graphical User Interface. In *Proceedings of HCI on People and Computers*, pages 209–219, 1998.

[3] H. Choi, H. Lee, and H. Kim. Fast Detection and Visualization of Network Attacks on Parallel Coordinates. *Computers and Security*, 28(5):276 – 288, 2009.

[4] G. Conti, M. Ahamad, and J. Stasko. Attacking Information Visualization System Usability Overloading and Deceiving the Human. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, pages 89–100, 2005.

[5] C. Correa, T. Crnovrsanin, and K.-L. Ma. Visual Reasoning about Social Networks Using Centrality Sensitivity. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):106–120, 2012.

[6] O. Fikir, I. Yaz, and T. Ĕandzyer. A Movie Rating Prediction Algorithm with Collaborative Filtering. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 321–325, Aug. 2010.

[7] FileZilla. FileZilla. http://filezilla-project.org, 2009.

[8] J. Goodall. Visualization is Better! A Comparative Evaluation. In *Proceedings of the International Workshop on Visualization for Cyber Security (VizSEC)*, pages 57–68, Oct. 2009.

[9] I. Hsi and C. Potts. Studying the evolution and enhancement of software features. In *Proceedings of the 1st International Visual Informatics Conference on Visual Informatics (IVIC)*, 2009.

[10] S. Kakuru. Behavior Based Network Traffic Analysis Tool. In *Proceedings of the 3rd IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 649–652, May 2011.

[11] C. Kan, C.and Hu, Z. Wang, G. Wang, and X. Huang. Netvis: A Network Security Management Visualization Tool based on Treemap. In *Proceedings of the 2nd International Conference on Advanced Computer Control (ICACC)*, volume 4, pages 18–21, Mar. 2010.

[12] D. A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002.

[13] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen. Real-time and Forensic Network Data Analysis using Animated and Coordinated Visualization. In *Proceedings of the Sixth IEEE SMC Information Assurance Workshop (IAW)*, pages 42–49, June 2005.

[14] S. Lau. The Spinning Cube of Potential Doom. *Communications ACM*, 47(6):25–26, Jun. 2004.

[15] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. Hellerstein. GraphLab: A New Parallel Framework for Machine Learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2010.

[16] J. Matejka, W. Li, T. Grossman, and G. Fitzmaurice. Communitycommands: Command Recommendations for Software Applications. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 193–202, 2009.

[17] J. McPherson, K. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: A Tool for Port-based Detection of Security Events. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, pages 73–81, 2004.

[18] T. Nunnally, P. Chi, K. Abdullah, S. Uluagac, and R. Beyah. P3D: A Parallel Coordinate System for Network Security. In *Proceedings of the IEEE International Conference on Communications (ICC)*, June 2013.

[19] T. Nunnally, A. S. Uluagac, J. Copeland, and R. Beyah. 3DSVAT: 3D Stereoscopic Vulnerability Assessment Tool for Network Security. In *Proceedings of the 37th IEEE Conference on Local Computer Networks (LCN)*, 2012.

[20] A. Oline and D. Reiners. Exploring Three-Dimensional Visualization for Intrusion Detection. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC)*, pages 113–120, Oct. 2005.

[21] J. Rasmussen, K. Ehrlich, S. Ross, S. Kirk, D. Gruen, and J. Patterson. Nimble Cybersecurity Incident Management through Visualization and Defensible Recommendations. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security (VizSec)*, pages 102–113, 2010.

[22] T. Taylor, S. Brooks, J. Mchugh, and S. Brooks. Netbytes viewer: An Entity-based Netflow Visualization Utility for Identifying Intrusive Behavior. In *Proceedings of the 2007 Workshop on Visualization for Computer Security (VizSec)*, pages 101–114, 2008.

[23] C. Ware. *Information Visualization Perception for Design*, volume 1. Morgan Kaufmann, San Francisco, USA, 2004.

[24] I. Xydas, G. Miaoulis, P. Bonnefoi, D. Plemenos, and D. Ghazanfarpour. 3D Graph Visualization Prototype System for Intrusion Detection: A Surveillance Aid to Security Analysts. In *Proceedings of the 9th International Conference on Computer Graphics and Artificial Intelligence*, May 2006.