

Attacking Information Visualization System Usability Overloading and Deceiving the Human

Gregory Conti
College of Computing
Georgia Institute of Technology

Mustaque Ahamad
College of Computing
Georgia Institute of Technology

John Stasko
College of Computing
Georgia Institute of Technology

ABSTRACT

Information visualization is an effective way to easily comprehend large amounts of data. For such systems to be truly effective, the information visualization designer must be aware of the ways in which their system may be manipulated and protect their users from attack. In addition, users should be aware of potential attacks in order to minimize or negate their effect. These attacks target the information visualization system as well as the perceptual, cognitive and motor capabilities of human end users. To identify and help counter these attacks we present a framework for information visualization system security analysis, a taxonomy of visualization attacks and technology independent principles for countering malicious visualizations. These themes are illustrated with case studies and working examples from the network security visualization domain, but are widely applicable to virtually any information visualization system.

CR Categories: H.5.2 [Information Systems]: Information Interfaces and Presentation - User Interfaces; C.2.3 [Computer-Communication Networks]: Network Operations: Network monitoring; C.2.0 [Computer-Communication Networks]: General - Security and Protection

Keywords: malicious visualizations, usability attacks, denial of information, secure visualization, information visualization

1 INTRODUCTION

Information visualization systems used for decision making must be designed with security in mind. Such systems are vulnerable to attack, either from malicious entities attempting to overwhelm, mislead or distract the human viewer or from non-malicious entities that accomplish the same result by accident. Some might believe that today's systems are not potential targets for attack. Clearly there are many domains where security is of minimal importance, but increasingly information visualization systems are being used to support critical decision making. For example, intelligence analysis, law enforcement, network security and business decision-support systems exist in an adversarial environment where it is likely that malicious entities are actively attempting to manipulate human end users. We believe that there is a clear threat today and there will be a growing problem into the foreseeable future. For information visualization systems to maintain relevance security *must* be considered. Information visualization systems inherently have the human tightly coupled in the system loop. In most cases, the human is the decision maker who will act upon (or not act upon) the information

presented and, as a result, the human is a high-payoff and likely target. *Any point* in the information visualization system may be attacked, from data collection to processing to final visualization, in order to impact human interpretation. A "minor" compromise of a single bit may have significant impact on the human (consider a change in the foreground color of a scatter plot to the background color). Major compromises may have far greater impact. Our primary goal is to identify these threats and vulnerabilities, as well as develop principles to counter or mitigate these attacks. By identifying the threats and weaknesses of their system, designers can make appropriate decisions to mitigate these vulnerabilities.

To see a sample attack in action, consider a visual intrusion detection system designed to supplement classical anomaly-based and signature-based intrusion detection systems. Such systems are typically co-located with a firewall at the border between the internal institutional network and the public Internet. This vantage point allows the system to observe and collect selected data from network traffic at entry and egress from the internal network. Our example system collects header data from network traffic and visualizes it in real-time. In particular, it captures the source and destination addresses of communicating network nodes, network protocols in use, source and destination ports (used for process to process communication across an Internet Protocol (IP) network, e.g. port 80 for a web server) as well as calculates a timestamp for each record. An adversary may easily inject arbitrary data into the visualization system, intermingled with legitimate users' traffic, due to weaknesses in current networking protocols. In our example, the adversary knows the system operator on the night shift is red-green colorblind. They also know that the default settings on the visualization system map the very common (99+% of traffic) Transmission Control Protocol (TCP) to green, the User Datagram Protocol (UDP) to blue and the Internet Control Management (ICMP) protocol to red. In addition, the attacker knows that the target node has serious ICMP and UDP vulnerabilities. The attacker waits until late in the operator's shift and launches an ICMP based attack. The already tired operator does not notice the red packet amidst the much greater noise of green packets. In this case, the attacker took advantage of the visualization system's color mapping to target a specific user, but many other techniques could have been used. We will describe and illustrate these attacks in later sections.

To help combat usability attacks against visualization systems this work includes several novel contributions: a framework for information visualization system security analysis, a taxonomy of malicious attacks as well as technology independent principles for designing information visualization systems that will resist attack. We illustrate and validate these contributions with results from the design, implementation and real-world use of a visual network intrusion detection system [1].

Information visualization systems are potentially vulnerable to a wide spectrum of attacks ranging from overt to subtle. An obvious attack is to simply corrupt the data. Akin to a denial of service (DoS) attack, an attack of this nature is likely to be immediately noticed by human users. While significant, in this work we are concerned with the more subtle denial of information attack [2]. Denial of information (DoI) attacks target the human by exceeding their perceptual, cognitive and motor capabilities. They reduce the ability of a human to acquire desired information. Even if a traditional DoS attack against a machine is not possible, the human utilizing the machine to process information may still succumb to a DoI attack [3]. Typically much more subtle (and potentially much more dangerous), DoI attacks can actively alter the decision making of human visualization system users without their knowledge. More specifically, for any visualization system, if an attacker can inject data into the dataset being visualized, or otherwise alter the dataflow, there exists the potential to exploit vulnerabilities in the human or the machine system. This exploitation can be used to accomplish some or all of the following high-level goals (inspired by well-established military information operations doctrine [4]):

- Mask a change in objects or actions that the system user has observed.
- Block the system user's perception and/or identification of objects or actions being introduced into the visualization system.
- Reinforce the system user's preconceived beliefs.
- Distract the system user's attention from other activities.
- Overload the visualization system or user's data collection and analytical capabilities.
- Create the illusion of strength where weakness exists.
- Create the illusion of weakness where strength exists.
- Accustom the system user to particular patterns of behavior that are exploitable at the time of the malicious entities choosing.
- Confuse the system user's expectations about an object's attributes and actions. In particular, to effect surprise in these areas.
- Reduce the system user's ability gain situational awareness and effectively make decisions.

To accomplish these goals, we make a key assumption: malicious entities may insert data into the dataset being visualized as well as deny access to, corrupt or alter the timeliness of data generated and communicated by networked data sources. We believe these assumptions to be reasonable. Many visualization systems gather information from potentially untrustworthy sources (such as unauthenticated Internet users or physically insecure sensors). In addition, data integrity and data availability are likewise susceptible to manipulation both in storage and in transit. Current cryptographic techniques can, if properly implemented, protect the integrity of data, but cannot guarantee availability. Consider that a small network device in the path of data flow can slow down or speed up transmission of sensor data despite cryptographic protection. Even more simply, a sensor could be unplugged at tactically important times. Given these assumptions, it is important to note that we will not concentrate on the more traditional, non-malicious problems associated with designing

information visualization systems as we believe they are currently being addressed. In most cases the problem of DoI attacks will remain even if these issues are addressed. Nor will we address general system attacks designed to broadly compromise as this is well addressed by the systems security community.

We argue that the ultimate goal of attacks against information visualization systems is to overload and deceive the human end users and force them to make incorrect conclusions and to take incorrect actions -- the exact antithesis of the goal of most information visualization system designers. This manipulation can be accomplished in a variety of ways, but ultimately these attacks corrupt data or alter dataflow in some way. They may occur quickly or over a long period at a barely perceptible, low level. The manipulation may take place at data generation, in transit over a communication network, at rest on a data storage device or during processing by a visualization engine. Attacks may be aggressive and essentially deny productive use of the system or may be subtle and covertly mislead. Either way, the result of an attack is an inaccurate picture as interpreted by the human end user. We have extensively reviewed these attacks and, for purposes of this paper, we will place emphasis on the more subtle attacks, but will also provide coverage of interesting more aggressive attacks. Aggressive attacks are almost certain to be noticed, but subtle attacks are more insidious and may be overlooked for an extended period of time. As a result, the negative impact of these attacks may be far greater.

The threats to information visualization systems are legion. Attackers may range from trusted internal users to external competitors and be motivated by competitive advantage, curiosity, intelligence gathering, notoriety, intellectual challenge or financial gain. To counter these attackers we argue that the only path to secure systems is via a thorough understanding of the possible threats and countermeasures. An effective technique to help secure systems is to conduct a threat analysis. Typically, this analysis includes the following elements: identifying assets you wish to protect, brainstorming known threats to the system, ranking the threats by severity, choosing how to respond to threats and choosing techniques and technologies (if any) to mitigate the threats [5]. We will include these elements during the course of the paper.

Section two of this paper discusses related work and places it in the field of current research. Section three presents a general framework for information visualization systems security analysis and identifies critical assets. Section four presents a detailed taxonomy of attacks. Section five provides countermeasures in the form of technology independent principles for information visualization designers to protect their systems and users from attack. Section six presents our conclusions and directions for future work.

2 RELATED WORK

The uniqueness of this work stems from the comprehensive analysis of the weaknesses of visualization systems, and their supporting data flow, including: data sources, data communications, data storage, processing, presentation and human interpretation. A novel taxonomy of attacks is presented as well as a technology independent set of design principles to assist in countering such attacks. While each information visualization system and technique has inherent strengths and weaknesses (see [6] for an excellent survey) most authors do not examine the potential of a malicious entity acting upon the system.

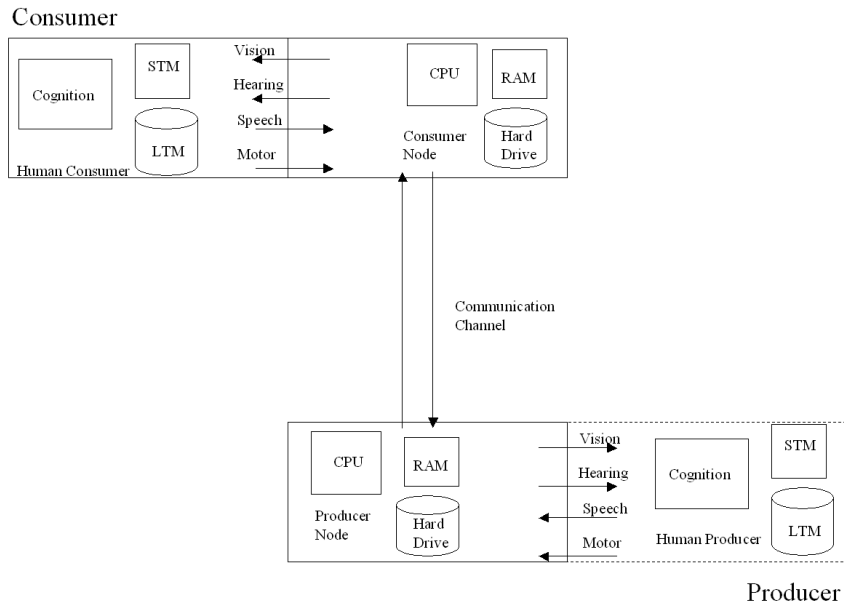


Figure 1. Generic producer-consumer information visualization system. Attacks influence any component, but the human end-user is the ultimate target.

The field of information warfare and the related fields of psychological warfare, propaganda and battlefield deception do include the notion of external malicious entities. In general, these fields seek to use deliberately false or misleading information to change people’s understanding through deception and confusion rather than persuasion and understanding [7]. In particular, the techniques of distraction, misinformation and disinformation are quite relevant, but do not specifically address information visualization. We will consider these applications in our work.

Information visualization, as an area, involves analysis of data sets in which the data is more abstract in nature, having no natural physical or geometrical representation [8]. Examples of data domains common to information visualization include statistics, financial data, text, and software. Research into the manipulation of information visualization systems is relatively uncommon, however. The VizLies special session of several IEEE Visualization conferences did address malicious visualization, but only in an informal manner, as entertainment at evening social functions. Several researchers have more formally considered the notion of malicious visualizations. Tufte addressed such concepts as the “lie factor,” disappearing baselines, the difference between height and volume in pictograms, misleading or missing scales, missing data, design dominating the data and the effect of 3D graphics on correct interpretation [9,10,11]. All are valid, but anecdotal, instances of malicious visualizations. Tufte further explores the boredom, wasted-time and degraded quality and credibility of communication by incorrectly utilizing PowerPoint presentation software [12,13]. While there are some interesting characteristics relevant to malicious visualizations (e.g. degraded quality of information and wasted time), these essays deal with the limitations of PowerPoint presentations in a non-interactive speaker to audience scenario. Books such as *How to Lie with Charts* [14] and *How to Lie with Statistics* [15] also explore techniques to design presentations and reports that mislead audiences or readers. In a similar vein, researchers such as Globus [16] and Bailey [17] focus on how system creators can massage their results to mislead audiences. Rogowitz considered

the application of perceptual rules to preventing “lying with visualization.” He did not consider external malicious entities [18].

From our perspective, the primary limitation of these works is that they focus on techniques the *creator* of the visualization system, business presentation, advertisement or statistical report can use to manipulate their audience. Our work assumes that this is not the case and that the creator of the information visualization system is non-malicious. *Our malicious entities attempt to attack the system itself, it’s data and the human attempting to utilize it. They are not the owners or creators of the system in question.*

3 SYSTEM MODEL

To best understand how attackers can accomplish the high-level goals presented in section one and to analyze how malicious visualizations manifest, we developed a generic producer-consumer information visualization system using a holistic systems approach (Figure 1). This architectural overview is useful for identifying assets by decomposing visualization systems and applications. The results can then be used to identify and prioritize the implementation of countermeasures.

The *consumer* is a combination of a human and machine. The machine presents the information to the human using a visualization method that relies on one of the human’s senses (typically vision). The human interacts with the interface using motor and speech commands and will draw conclusions based upon the information presented. The *producer* is the source of the data that will be visualized. In some cases, the producer will include a human who interacts with an information system to produce all or a portion of the data that will ultimately be visualized. In other cases, the producer will consist of only an information system that generates the data. No human is directly involved in data production (e.g. a sensor network). The producer may be co-located with the consumer, but it is more likely that the producer will need to communicate the data to the consumer via a communication channel.

Target					Attacks	Possible Countermeasures
Human	Processing	Cognition	Memory	Perceptual Buffers	Force desired colors to be used Force smaller font	Review annotation algorithms Limit range of colors, sizes allowed. Review preattentive literature for best interface objects
				Short Term	Display updates too rapidly	Compensate with buffers in visualization system.
				Long Term	Aggregation hides important detail Scaling lacks detailed enough resolution Attack paging of visualizations	Lack of long term overviews Background images of historical data Use of paged and side-by-side images and overlays. Create smart book of visual signatures
		Cognitive Processing	Degrade trust in system Attack when human is not watching Cry wolf Visualization software causes poor conceptual model	Display visualization's source data Create visual log files Ambient Visualization		
	Input	Vision		Causing occlusion of visual elements to conceal or manipulate visual presentation Inserting random noise into visualization Force less detailed scaling Occlusion of visualization elements Color choices impact color blind user	Develop alternative visualizations and views of data Include customizable filters Provide multiple coordinated views of data Choose smart default settings	
	Output	Motor		Cause alert which forces user motor response (e.g. clicking an OK button) Force the user to scroll UI requires unnecessary actions	Review improved triggering mechanisms Explore alternative interface designs	

Table 1: Denial of information attack taxonomy illustrating representative attacks by model human processor target

Each human and machine component processes data using subcomponents with finite resources. *Attacks can target any of these resources.* For the human, we chose to model these resources based on the Model Human Processor (MHP) architecture: short term memory, long-term memory, cognition as well as perception and motor processing [19]. For each machine, we used the common information systems model of machine resources: processing, short-term storage (RAM) and long-term storage (typically optical or magnetic storage media). The human and its associated information system interact through the classic human-computer interaction boundary. The human utilizes vision, hearing, speech and motor actions to interact with the information system. Other senses (e.g. touch and smell) are not shown in the model, due to the limited availability of effective interface technologies. The information system provides related input/output devices that support each of these human capabilities

(e.g. CRT, speakers/sound sub-system, microphone, keyboard and mouse).

4 INFORMATION VISUALIZATION ATTACK TAXONOMY

While attacks may range from overt to subtle, they share several common properties: they attempt to influence *how* you visualize, *what* you visualize or *when* you visualize. To this end, we present a taxonomy of attacks that follows the flow of information from human consumption back to initial data generation. We have developed a comprehensive taxonomy of attacks, but for purposes of this paper, we provide a representative overview of the taxonomy and illustrative examples to highlight the vulnerabilities and surprisingly effective exploits of traditional information visualization systems. We have chosen to follow the information flow from the human back towards data generation, believing that

this is an intuitive and natural way to illustrate an interesting spectrum of attacks. We will use the components along the path (see Table 1) to illustrate how and when attacks may manifest. Attacks may influence any component, but the human end-user is the ultimate target.

4.1 ATTACKING THE HUMAN

Humans are vulnerable targets with finite resources to perceive, interpret and act upon information. Attackers consume these resources *through* information visualization systems by altering the accuracy, completeness, consistency and timeliness of information flows. By focusing on human limitations these alterations create incomplete, ambiguous or incorrect visualizations that result in frustrated analysis, reasoning and decision-making. These malicious visualizations increase complexity, alter or destroy patterns, distort sequences and disrupt exploratory tasks which in turn may cause confusion, disorientation, disbelief, distraction or lack of trust. While not necessary, the effectiveness of attacks can be enhanced by specifically targeting individuals and their unique set of weaknesses and predispositions (consider our colorblind user from Section 1). The following sections examine attacks against the human using a slightly streamlined version of the Model Human Processor (MHP) model of cognitive processing, memory, vision and motor resources [18].

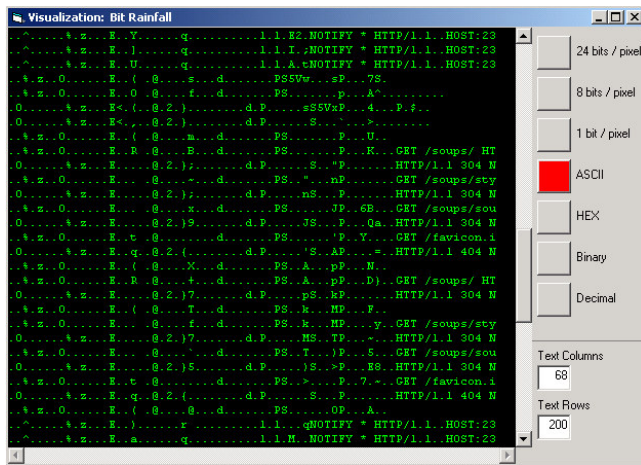


Figure 2. Semantic zoom visualization of network traffic.

4.1.1 Attacking Human Memory

Humans possess a limited ability to remember information over short and long periods of times. Arguably, humans can remember 7 +/- 2 “chunks” over a short period [20]. Regardless of the exact number, the human has a finite capability to retain and recall information. By exploiting this limitation an attacker can greatly increase their likelihood of success. These attacks may manifest themselves gradually such that the user fails to see the pattern. Alternatively attacks may target the users ability to recall legitimate activity to the detail required to detect malicious activity. Figure 2 illustrates this limitation. This system, designed by the authors, attempts to provide a semantic zooming capability [21] for network traffic by allowing the user to view network information at variety of different scales from course

grain overviews to high-resolution detail. The user selects the level of resolution using the scale on the right of the interface. Despite this attempt at allowing users to compare network traffic, it suffers from limitations of human memory. In our tests using the current configuration, users simply could not retain the context from one level to the next. Attackers could clearly exploit this weakness. To the best of our knowledge, no security visualization systems directly support the ability to closely compare images for subtle differences required to detect this class of attack. While Unix systems can use the *diff* command to compare text files, there is no equivalent visual diff. Likewise, there are no security visualization systems that allow users to seamlessly compare images in a side-by-side manner frustrating effective comparison.

4.1.2 Attacking Cognitive Processing

Cognitive processing deals with how humans interpret information. By exploiting weaknesses in this processing, an attacker can mislead the human and obscure or camouflage attacks as well as lead users to incorrect conclusions, perhaps even frustrating the users to the point they abandon use of the system altogether. Attacks can target attention, perception of time, decision-making, pattern recognition and perception of color and shape. Attackers may increase cognitive complexity, add spurious packets to eliminate suspicious outliers or demand the attention of the user. The following sections illustrate representative cognitive processing attacks against human attention and perception.

4.1.2.1 Attention

By their nature, information visualization systems require human attention. Depending on the design of the visualization and user interface the system may likely be tightly coupled with the user. It is impossible for a user to maintain 100% focus on their visualization system for long periods of time. Even a distraction lasting a few seconds can cause a user to miss key information. Alternatively, the attacker may overwhelm the user by demanding too much attention.

“Cry Wolf” Attack: From the classic children’s story, an attacker can trigger activity, which in a normal scenario would require user attention. As a result, if the system “cries wolf” enough times the operator will begin to lose trust and may disable the system altogether. As an example, an attacker may subvert the snort intrusion detection system by creating packets that trigger alerts [22]. Snort alerts the user when it detects a signature in network activity that matches a known attack in its database. The snort tool is specifically designed to attack users through snort [23]. Utilizing snort’s database of signatures, snort can generate network traffic that matches alert rules. Using snort, an attacker can trigger an alert at will.

Displacement Attack: Displacement attacks occur in visualizations where incoming data visually displaces older information. These visualizations are particularly susceptible to the limitations of human attention. Figure 3 is a network monitoring and intrusion detection visualization from the rumint system that displays network traffic in a scrolling display [24]. The bits of packets are plotted on the horizontal axis. As each packet arrives it is plotted one pixel lower on the vertical axis. When the display reaches the bottom of the display window, it begins plotting at the top of the display, overwriting previous contents. During the past year we have used this system in two operational settings. The first was with the Georgia Tech

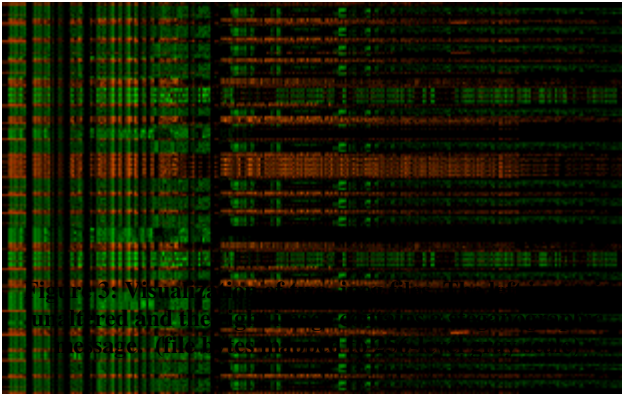


Figure 3: Binary rainfall visualization of network packets. (One packet per row)

Honeynet and the second was with a dedicated commercial Internet Service Provider (ISP) residential connection. In both instances, the network connection is not used for any legitimate traffic thus only malicious activity is seen. Network packets typically arrive in small groups averaging one to five minutes per packet. Scrolling in these instances is typically not a problem, as approximately 24 hours of traffic can be seen before older information is overwritten (although we have seen spikes in traffic where network activity has been significantly greater). To test the time required for an attacker to scroll information off the page we conducted several experiments and found that it required only 2-3 seconds to overwrite information on one of our research machines (AMD 2500+, Windows XP, 1GB RAM, 100MB Ethernet). It is important to note that the theoretical limit based on network bandwidth alone is on the order of ten-thousandths of a second. We believe that a small lapse in attention on the order of seconds, even by a dedicated observer, is a reasonable possibility that an attacker may exploit to destroy traces of malicious activity.

4.1.3 Attacking Visual Perception

Information visualization systems, and the great majority of interactive computing applications, rely heavily upon the human's perceptual capabilities. Visual perception is the processing of input stimuli based upon reflected wavelengths of light from which our brain constructs internal representations of the outside world [25]. By taking advantage of the strengths and weaknesses of visual perception, an attacker can alter this internal representation. Consider the optical illusions from classic psychology. Given the same input image, different subjects might interpret the picture differently. In other examples, subjects are unable to correctly judge spatial relationships. See the work by Bach for 52 online examples [26]. Examples of other known weakness include a blind spot in the visual field, motion induced blindness [27] and a limited ability to discriminate between colors. Even adaptations, which can be considered strengths in some instances, become weaknesses when manipulated by an adversary, such as preattentive processing [28] and afterimages caused by light/dark adaptation [29]. Beyond simple manipulation, even more aggressive attacks are possible. Small delays in virtual reality visualization systems can cause queasiness [30] and fifteen to twenty frames per second of certain images can trigger photosensitive epilepsy. (see Section 4.1.5)

Color Mapping Attack: The color mapping attack targets the use of color in visualizations. Humans have a limited ability to discriminate between colors, on the order of 300,000 colors [29].

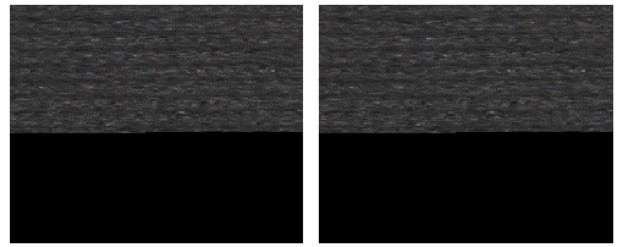


Figure 4: Binary visualization of two JPEG files. The left image is unaltered and the right image contains a steganographic message. Bytes from files are mapped to 256-level grayscale pixels.

Not all of these colors are interpreted as equivalent values, some are given heavier weight or draw more attention than others, and because color ranges are not uniform, normalization is used to help counteract the effect. See the work of Rogowitz and Treinish for an excellent discussion [18]. Most computing systems can present far more colors than a human can discern, 2^{24} possible colors is typical on today's hardware. Depending on the visualization system in use, features of the data are mapped, in a variety of ways, to colors used in the display. Limited display colors allow an attacker to hide activity due to aggregation. Large numbers of colors exceed or degrade the ability of humans to glean appropriate insights. It is due to these system presentation and human interpretation gaps that users are vulnerable, particularly when the system provides only a limited ability to customize colors for given tasks. Figure 4 comes from a visualization system created by the authors. It maps byte values from binary files to 256-level gray-scale pixels. In this example, the figure shows the file structure of two jpeg files. The left image is unaltered and the right image contains a steganographic message. Despite our ability to distinguish hundreds of thousands of colors, in our experiments, users were unable to find the modified bits. For future work we plan to pursue a visual *diff* tool, but the fact remains that for even a small number of colors, humans have difficulty in detecting differences. This weakness allows malicious entities to operate below the detectable threshold. Even the addition of a color legend is of little value. In a separate experiment we plotted network packets on a scatter plot using a commercial system. Even with only 100 different colors mapped to packet features (colors were chosen by the system) and

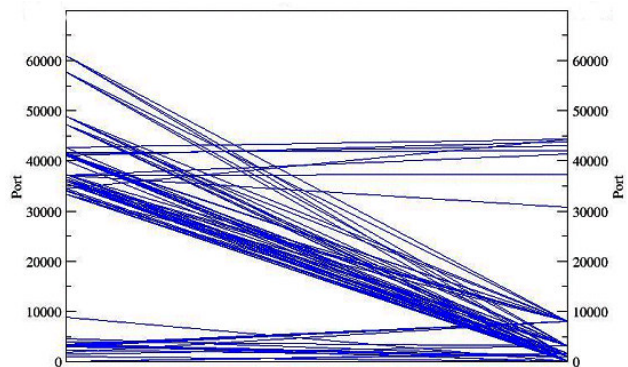


Figure 5: Autoscale and motor resources attack example (overview)

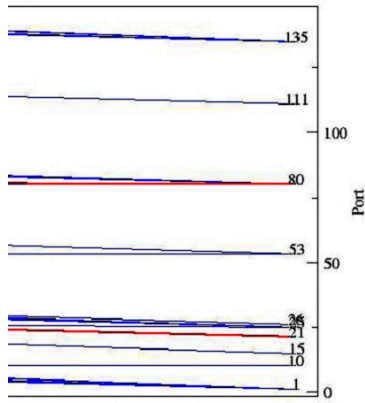


Figure 6: Autoscale and motor resources attack example. Note the targeted network services, originally hidden from view.

a color legend, users took considerable time to match the respective color to the appropriate value. In another experiment, using the same commercial system and a scatter plot, we plotted 1358 different network packets. We exceeded the number of categorical colors the system could provide and were forced to use a continuous scale. In this mode, no legend was provided. It proved impossible to identify the feature value from the color.

4.1.4 Attacking Motor Resources

This class of attack attempts to consume time and increase frustration by forcing user motor actions. Attacks may be as simple as forcing paging across multiple screens, consider the rumint system described in the *displacement attack*, but add a buffer that stores previous pages of images. As each screen is filled, the user must interact with the interface to observe previous activity. Another example is to force user thrashing by requiring constant swapping from detail to context and back. Figures 5 and 6 illustrate this attack. The dataset behind these figures comes from an unclassified attack/defend exercise, in which a National Security Agency red team attacked student-defended networks [31]. The user is presented with an overview of network activity in Figure 5, but to see the specific port-level the network activity in Figure 6 the user must zoom in and then back out to continue to monitor the overview. In this example the user would have to perform this operation ten times just to monitor the 1024 privileged ports on a Unix system.

4.1.5 Targeting Specific Humans

While the attacks described previously in section 4.2 were significant, even more effective attacks are possible if the specific human user is known. With this knowledge, an adversary may craft an attack that specifically exploits their target's weaknesses. Vision, memory, reflexes, experience and intelligence vary greatly between individuals. Even partial knowledge of the

specific end user gives the adversary an advantage; their attack may be markedly different for a 19-year-old male intern, a 37-year-old male disgruntled employee or a 58-year-old female veteran who has heavily corrected vision. We believe that some degree of knowledge of the human user to be a reasonable assumption. A few casual questions asked at an after-hours happy hour frequented by company employees would likely gain useful information. A comprehensive discussion of all such attacks is beyond the scope of this work, but we will illustrate the vulnerability by examining photosensitive epilepsy. While this condition is relatively rare, it does illustrate the increased risk when the attacker can target specific people and their weaknesses. We argue that related attacks can be launched when age, gender and/or medical details are known about users.

Extreme Information Overload Attack (Photosensitive Epilepsy): Epilepsy has a lifetime prevalence of about 3% and approximately 2.3 million people in the United States have the condition. Of this population, a percentage has photosensitive epilepsy. People with photosensitive epilepsy are susceptible to seizures brought on by rapidly flickering or flashing displays. In the late 1990's, thousands of people were sickened with nausea and dizziness by a Japanese Pokemon cartoon. In addition, there were 685 cases of apparent epileptic seizures [32]. The risk extends beyond the viewing of shows on televisions and computer monitors. Video games have also induced seizures and many now carry warning labels. It is important to note that the video game and video industries have since taken other proactive measures to limit future incidents; reducing the overall incidence of the problem. For example, the Pokemon cartoons were reviewed, frame-by-frame, before rebroadcast in the United States [30]. An attacker would do the opposite. Research by Harding indicates that the larger the area of the retina covered with the flashing display, the greater the likelihood of a seizure. In particular, flashing at the rate of 15-20 times per second was shown to be most likely to induce a seizure; 96% of people with photosensitive epilepsy were sensitive at this frequency. In addition to flashing, static patterns have induced seizures and the likelihood is dramatically increased when patterns are updated at the rate of 15-20 changes per second [32]. With the trend toward larger displays and higher resolution the situation is worsened. In our experiments we were able to generate network traffic that caused both static and updating patterns in our network visualization system that would possibly induce seizures in some photosensitive epileptics, but we did not proceed further due to safety concerns.

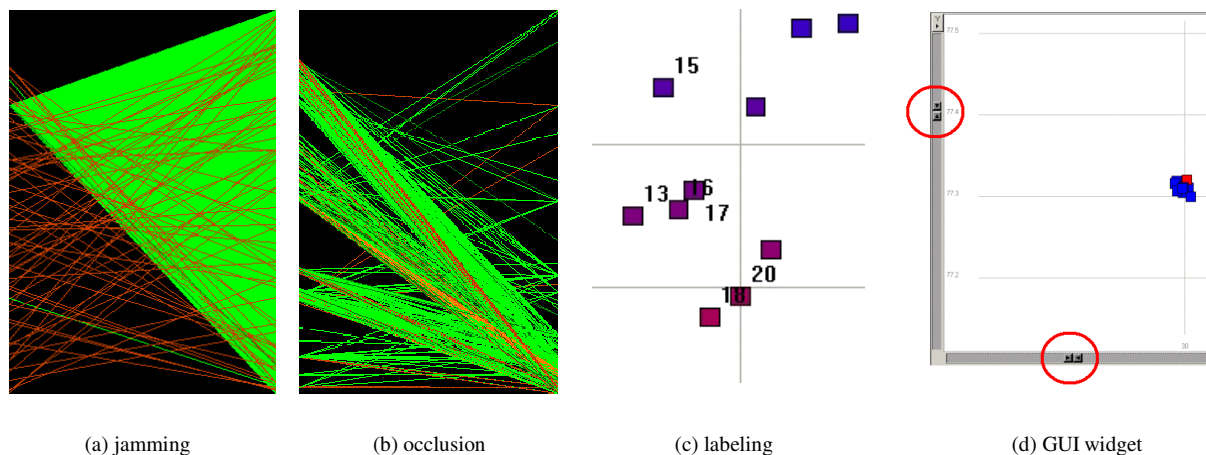


Figure 8: Representative attacks against the visualization

4.2 ATTACKING VISUALIZATION HARDWARE AND SOFTWARE

The attacker affects attacks against the human by influencing how information is visualized. As was the case for humans, the notion of specificity is important to consider. Many of the techniques described below are most effective when used against specific information visualization systems, but others are broadly applicable.

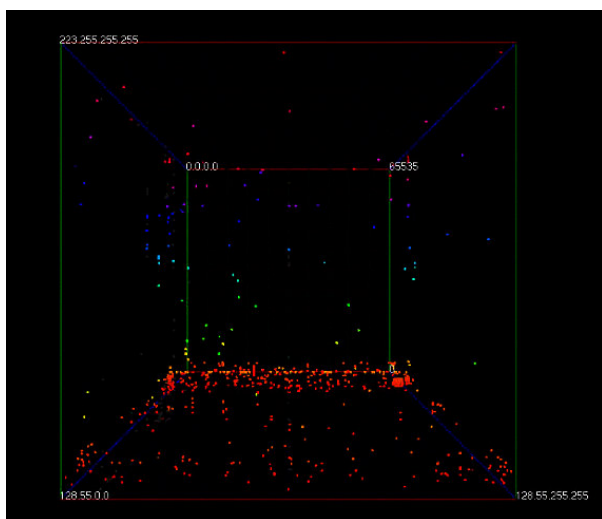


Figure 7: View of the “Spinning Cube of Potential Doom” a 3-D visualization of network traffic designed for security awareness. (round-off attack)

4.2.1 Processing Attacks

Processing attacks target the algorithms used to process and present the visualization. These algorithms range from simple graphic routines to advanced artificial intelligence or machine learning techniques. Attacks may be designed to increase computational complexity, e.g. creating a large number of objects such that the interface becomes sluggish or the visualization delays presentation of important information. Others may exploit intelligence embedded in the visualization system. Consider a generic spring layout algorithm. To be most effective, this

algorithm relies upon the graph to reach a stable state. Carefully constructed packets could be used to force constant destabilization. Other attacks may take advantage of bugs in the code or the calculations in use, such as interpolation or round-off. To provide a concrete example of the efficacy of these classes of attack, the following section illustrates the round-off attack in detail.

round-off attack: Consider the “spinning cube of potential doom” visualization system in Figure 7 [33]. Designed to provide insight into network attacks, it displays network traffic along three axes. The X-axis represents the destination IP addresses for a Class C network (65536 possible addresses), the Y-Axis displays destination ports from 0-65535 and the Z-axis displays source Internet addresses from 0.0.0.0 - 223.255.255.255 (no multicast). Assuming an approximate 1024 pixels for each axis. The X and Y axes round off 6 bits of information, leaving an opening for an attacker to operate within a space of 64 indistinguishable positions. More importantly, the Z axis rounds off approximately 22 bits of information, grouping source IP’s into buckets of over 4 million each. Thus an adversary could attack 64 machines on 64 ports from over 4 million source IP addresses and, due to round off, would only illuminate a *single* pixel. Note also that the visualization is also a target for a *color mapping* attack. It uses a “rainbow” color map representing TCP connection instances. Although a large number of colors are used, the actual color does not have “any meaning.”*

4.2.1.1 Attacking the Visualization

The heart of a visualization system are the visualizations it presents to the user. Closely intertwined with processing attacks, attacks against the visualization design will have an immediate effect on the user. Some visualizations were simply not designed to convey a certain type of activity, so an attacker may easily operate with impunity. In other cases, the design is such that a small amount of malicious data can destroy or reduce the effectiveness of the system. Designers are faced with large, potentially massive, datasets and limited screen real estate to present information and are forced to make design tradeoffs that can be exploited. The following are examples of such attacks.

* Except gray points which are completed TCP connections.

autoscale attack: Many visualization systems use autoscaling algorithms to provide an optimal display of information. Typically the algorithms zoom the perspective outward to display the entire dataset. While this is convenient in many cases, an attacker can easily exploit this vulnerability. The image shown in Figure 5 was created by the *xmgrace* open source information visualization tool [34]. A small number of packets sent by the attackers to those ports above 40,000, forced the autoscaling algorithm to zoom outward, thereby obscuring significant detail (Figure 6).

jamming attack: The jamming attack is a simple attack, akin to a visual denial of service. By observing what aspects of the dataset are used to control the physical location of objects on the screen visual noise can be inserted to partially or completely corrupt the information display. As noise is inserted, insightful patterns, relationships, anomalies, outliers and trends will disappear. We produced multiple versions of this class of attack in our network visualization system by generating network packets with appropriate headers. Figure 8(a) is a parallel coordinate plot of TCP/UDP ports. The left axis shows the attacker's source ports and the right axis shows the target machine's ports (on a 0-65535 scale). The image shows two jamming attacks, both done using the *packit* packet creation tool [35]. The first attack generated 200 UDP packets (in orange) with random source and destination ports. The second attack (in green) generated 2000 TCP packets from a single source port, but random destination ports. On a 100MB network, *packit* generated these malicious packets at over 6600 per second.

occlusion attack: Occlusion is a problem in many visualizations, particularly those in 3D, but any that plot new information over old are susceptible. An attacker can use this frequent shortcoming to hide malicious activity. In the Figure 8(b), an attacker's malicious activity is hidden behind newer activity.

labeling attack: Typically visualizations provide the ability to label the image. Depending on the labeling algorithm in use, this fact can be exploited. One popular commercial visualization system defaults to only 20 labels. If the user does not change this setting a large number of objects will not be labeled, greatly complicating user interpretation. See Figure 8(c) for an example. At the other end of the spectrum, some labeling algorithms do not limit the number of labels used and, by injecting extra data, an attack could cause the display to be obscured.

GUI widget attack: User interfaces only provide a limited ability to interact with the dataset. An attacker can exploit this limitation and prevent users from detecting malicious activity despite their best attempts. Figure 8(d) shows a cluster of network activity; because of the large range of values in the overall dataset (not shown), the user is unable to zoom in any further. Any movement of the sliders will cause the entire cluster to move off the screen. Note the two red circles. Each circle shows a double-ended slider at the closest possible position.

4.2.2 Storage Attacks

From our research, storage attacks against information visualization systems can occur primarily in the form of classic denial of service. Denial of information and not denial of service is the focus of the paper so we will touch only briefly on it here. Every information system has a finite amount of storage. By consuming all or most of this storage an attacker may subvert the intent of the visualization system. In the network security domain, a classic example is flooding the network with traffic,

sometimes legitimate (also known as the slashdot effect) and sometimes malicious (trigger logging events to the point that the hard disk fills or malicious activity is overwritten). Variants include filling up the buffers of network interface cards such that packets are dropped or consuming RAM to the point that the operating system needs to page memory to disk (or even thrash). All of these attacks negatively impact performance and could crash or slow the system. While not strictly a storage attack, it is well documented that, in shared user systems, one user's applications can consume resources to the performance detriment of other users. Correctly designed interfaces operate within very strict timing parameters and a sluggish interface (or visualization) that quickly becomes difficult or unusable could quickly occur.

4.2.3 Attacking Data Generation and Communication

By definition, information visualization systems present data to the user in order to provide insight. If the accuracy, reliability, timeliness, completeness or currency is threatened then the entire system is at risk. Attacking data quality early in the system flow is a means to an end and not an end unto itself. The tainted data will ultimately flow upstream to the visualization system which, in turn, will alter the user's perception and hence negatively impact task accomplishment. Recall that we do not consider data corruption attacks as we believe that they will be easily detected. We operate with the stricter assumption that an attacker can only insert data, and not modify existing data.

4.2.3.1 Attacking Data Generation

In our model, data can come from human and machine producers, both of which may prove unreliable despite the best intentions of the system designer. This notion is directly opposite to the common assumption that the "source must be good." While not the focus of this paper, physical attacks are the most straightforward attack. The most basic is physical destruction or theft which causes a failure to record data. More subtly, an attacker may spuriously add, remove or compromise information producing nodes via physical access or network attack. Consider physically turning a sensor on and off (or cutting power) which results in selected subsets of data being recorded. Note that this could occur with more than one sensor and provides the attacker the ability to paint a customized and comprehensive picture of the information space. Beyond physical access, we consider attacks that allow an attacker to operate remotely.

sensor blindness attack: Network-based blindness attacks allow an attacker to remotely crash selected packet capture sensors on the network. As an example, virtually all Windows-based network sniffing programs use the WinPcap [36] packet capture library. Versions of the library have known vulnerabilities that will crash the sensor.

selective sensor blindness attack: Similar to the *sensor blindness attack* this variant exploits differing operating system implementations of the network processing stack to avoid detection. For example, one operating system may accept a packet with an incorrect TCP checksum while another will silently ignore it. This inconsistency allows network intruders to operate without detection if the network sensor ignores the packet and a target machine accepts it. For more information see the work of Ptacek and Newsham [37].

spoofing source identity attack: Spoofing source identity is another common vulnerability, usually due to weak access controls or authentication, that allows users or network systems to

appear as legitimate producers. In the network domain, it is trivially easy to spoof IP packets. The protocol offers no protection and an attacker may transmit packets with spoofed source addresses that appear to come from legitimate sources.

interface spoofing attack: Interface spoofing attacks have existed since the beginning of shared computing systems. Typically they are used to trick legitimate users into revealing sensitive information, such as passwords. In the context of this paper, they can be used to trick legitimate users into submitting incorrect data to the visualization system. This technique can be seen when employing a variant of current phishing attacks. An attacker could send an email to a legitimate producer asking them to use a new website to submit information. Normal cues from the browser, such as the status bar, can be spoofed to prevent detection. See the work of Levy for more detail on this class of attack [38].

sampling rate attack: Sampling rate attacks exploit the periodicity of data collection. Due to the high rate of data flow observed by some sensors, by necessity, sample data at a constant or varying rate. This is typical in today's network visualization systems. Even in near real time systems, a five minute sampling rate is common. By gaining an understanding of when data is sampled, an attacker can avoid detection.

poisoned data attack: Poisoned data attacks are carefully crafted to inject a small amount of malformed or incorrect data to disrupt collection or analysis. These vulnerabilities may exist due to a lack of input validation at the producer as well as the consumer's system. As we mentioned earlier, a single legal packet can have significant impact on the end user, as was seen in the *autoscale attack*. The same can be accomplished with a small amount of seemingly legal, but maliciously formed data. An excellent example, is the recent spate of image files that exploit vulnerabilities in image processing libraries. A single such image can crash a visualization application or provide privileged access to the attacker.

4.2.3.2 Attacking the Communication Channel

Communication channels connect the information producing nodes to the information visualization system. Long a subject of network security discussion, there are a large number of vulnerabilities in current networking protocols. If communication links are not secured with message confidentiality and integrity protection, an adversary may easily perform a "man in the middle" attack and arbitrarily alter packets between the producer and the information visualization system. Also, as we have discussed, the network layer (IP) provides virtually no protection from spoofing source identity and other tampering. Common transport layer protocols (TCP and UDP), similarly provide limited protection. UDP makes no attempt. TCP relies upon the three-way handshake and session establishment to prevent spurious packets. Handshaking and session establishment provides only limited protection as an attacker can employ well-known TCP session hijacking techniques. Due to these weaknesses, an attacker can alter messages between producer and consumer at will, as well as observe all message traffic, unless some form of cryptographic protection is used. Even if a secured protocol is used, most will still be vulnerable to the following timing attack.

channel timing attack: By placing a properly configured network device in-line along the communication channel between the producer and the consumer, an attacker may affect a number of

timing based attacks. The channel timing attack allows the capture and replay, both faster and slower than actual, of network traffic. By altering the timeliness of how and when data is presented to users, an attacker may reduce or increase data density or alter the distribution of data values causing a direct impact on the visualization and the human. Time-series data is particularly vulnerable to this class of attack.

5 PRINCIPLES FOR COUNTERING MALICIOUS VISUALIZATIONS

There is no panacea that will absolutely protect information visualization systems from attack, but there are important design principles and assumptions that will mitigate the risk. Recall that any information visualization system in which a trusted or untrusted adversary can inject or modify information places the end user at risk. As we conducted the research associated with this paper we designed a variety of security information visualization systems and fielded them in operational settings. As a result of this experience we have learned a number of lessons. As you design or redesign systems of your own, we hope that you will consider these principles and assumptions. We believe they will greatly reduce the likelihood of many classes of successful attack. In other instances, there is no clear-cut solution and the only countermeasure is awareness of the vulnerability.

From our experience, often the initial design of the system itself was at fault, leading to easily exploitable vulnerabilities such as the displacement attack. Others are more difficult to implement and potentially require detailed information about the system in use or the specific user. By using these principles and considering these assumptions during design, threats may be pruned or reduced and prudent design tradeoffs may be made. Ultimately, as information visualization systems are used for critical applications we must continue to explore how we can effectively deal with threats in order to make such systems more secure and relevant.

5.1 EDUCATE THE USER

The user is the ultimate target of attackers and the success or failure of an attack depends, in large part, upon their individual susceptibility. To counter many forms of attack, train users to be alert for manipulation, aware of their personal weaknesses and to take maximum advantage of system customization capabilities to counter these weaknesses. As a result, users will be better protected and resistant to attack. The intelligence community uses similar techniques to help prevent successful social engineering attacks through security awareness training.

5.2 ASSUME AN INTELLIGENT, WELL INFORMED ADVERSARY

Information visualization systems of any import will be targets of attack. Underestimate the attacker at your own risk [39]. To best protect your system you must assume an intelligent and well-informed adversary. The attacker may gain information through open-source (publicly available information) or through social engineering. Seemingly unimportant data may prove to be extremely valuable. As an example, such information as the time lunch was served and the location of the dining hall, both considered to be trivial pieces of information, possibly enhanced the attack on the USS Cole. It is not unrealistic to assume that an attacker knows the visualization tool in use. This assumption is strengthened in areas where a single tool dominates or there is a

lack of diversity. In some cases, the attacker may possess the tool itself and the source code. This access allows an adversary full knowledge of its operational characteristics and implementation vulnerabilities (buffer sizes, defaults, scaling algorithms, color mapping etc.) This assumption also applies to your users, the same social techniques that are used to gather technical information can also be used to gain insight into specific operators and environmental conditions. An intelligent and well-informed adversary will target your specific system through its weakest link, at the worst time with the weakest user at the controls. The best defense is to look at your system through the eyes of an attacker, predict their likely attack courses of action and consider what you can do to counter or frustrate their actions.

5.3 DESIGN THE SYSTEM TO PROTECT THE USER.

Assume the system, including the implementation and supporting information flow (from source to human consumption), will be attacked. Given this assumption, every creator of a visualization system or technique should consider malicious applications and seek to create well thought out visualizations that are resistant to attack. At the time of creation, system designers do not necessarily know the full range of future use. Assume your system will be used for critical applications and attempt to predict second and third order effects.

Visualization systems typically have the human tightly coupled in the decision making loop. These systems require the limited resources of human attention and time, use them wisely. Even a small consumption of these resources by an adversary can cause unintended consequences on human decision-making. Customizable systems with intelligently chosen, attack resistant, defaults will help prevent overloading or deceiving the user, especially when combined with validated classical information visualization principles. If after your analysis, you cannot protect against a given class of attack before it reaches the user, at least assist the user in detecting one has taken place (detecting “wrongness”).

5.4 PROTECT THE DATA GENERATION AND DATA FLOW.

An information visualization system is only as good as the data upon which it depends. Your ultimate goal is to improve data quality by increasing the good and reducing the bad, with emphasis on the most dangerous. It does not take much bad data to cause significant damage. In the network security domain, a single bad packet can provide root level access, waste hours of an operator’s time due to a false snort alert or hide an attack due to an auto-scaling algorithm. In most instances, information visualization systems operate in environments in which an adversary can insert malicious data. Any source of data can be manipulated by a potentially malicious entity, including legitimate users, machine producers and other trusted sources. Your data should be protected by well-validated techniques such as input and source validation and cryptography.

While it is beyond the scope of this paper, designers should be aware of secure systems design best practices [5] and threat modeling [40]. In particular, consider secure protocol development (confidentiality, authentication and integrity, in particular), appropriate use (and the limits) of cryptography, suitable security and usage policies, physical security, intrusion detection and input validation. In high-risk applications,

physically closing the system to outsiders (air gapping) and the use of virtual machines to separate data and processing into logical groupings may be in order.

6 CONCLUSION AND DIRECTIONS FOR FUTURE WORK

Information Visualization is one way of effectively communicating information. Deception is one way to negatively affect this capability. Today’s systems are being used in critical applications to glean insights that are difficult to see using traditional non-visual techniques. As malicious entities become aware of the power of these tools, the tools themselves and the decision makers that use them will increasingly become the subject of attack. These vulnerabilities may manifest as significant attacks and we have provided real world examples to show that these attacks are real. Any system that uses data from malicious trusted or untrusted sources is at risk. Today’s visualization technology has not been designed with consideration of these risks and the notion of active malicious entities. Even carefully user-customized applications are vulnerable due to incorrect defaults, limitations in the visualizations themselves and weaknesses in the overall system. To help counter these attacks we have proposed a framework and taxonomy for analysis, presented viable attacks from the network security domain as well as design principles and assumptions to help create systems that protect both the system and the user. For the efficacy of information visualization to continue we must further explore denial of information attacks.

7 ACKNOWLEDGEMENTS

We would like to thank Dr. John Stasko’s research group for their thoughtful review and comment as well as Dr. Henry Owen, Julian Grizzard and Jeff Gribshaw for their insightful comments and free use of the Georgia Tech Honeynet. Finally, we would like to thank Lieutenant Colonel Ron Dodge and the United States Military Academy’s Information Technology and Operations Center (www.itoc.usma.edu) for their continued support.

8 REFERENCES

- [1] Conti, G. and Abdullah, K. Passive Visual Fingerprinting of Network Attack Tools. Workshop on Visualization and Data Mining for Computer Security (VizSEC), October 2004.
- [2] Ahamad, M., Mark, L., Lee, W., Omicienski, E., Dos Santos, A., Liu, L. and Pu, C. Guarding the Next Internet Frontier: Countering Denial of Information Attacks. New Security Paradigms Workshop, 2002.
- [3] Conti, G. and Ahamad, M. Countering Denial of Information Attacks. *IEEE Security and Privacy*. (accepted, to be published)
- [4] Army Battlefield Deception Operations. Air War College, United States Air Force. <http://www.au.af.mil/au/awc/awcgate/army/batdecep.htm>
- [5] Howard, M. and LeBlanc, D. *Writing Secure Code*. Microsoft Press, 2002.
- [6] Wilkinson, L. *The Grammar of Graphics*. Springer-Verlag, 1999.

- [7] Propaganda. *Disinfopedia*.
<http://www.disinfopedia.org/wiki.phtml?title=Propaganda>
- [8] Spence, R. *Information Visualization*. ACM Press, 2001.
- [9] Tufte, E. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, 1997.
- [10] Tufte, E. *Envisioning Information*. Graphics Press, 1990.
- [11] Tufte, E. *The Visual Display of Quantitative Information*. Second Edition. Graphics Press, 2001.
- [12] Tufte, E. Power Point is Evil. *Wired Magazine Online*,
http://www.wired.com/wired/archive/11.09/ppt2_pr.html
- [13] Tufte, E. *The Cognitive Style of PowerPoint*. Graphics Press, 2003.
- [14] Jones, G. *How to Lie With Charts*. Authors Choice Press, 2000.
- [15] Huff, D. *How to Lie With Statistics*. W. W. Norton and Company, 1993.
- [16] Globus, A. and Raible, E. Fourteen Ways to Say Nothing with Scientific Visualization. *Computer*, Vol. 27, No. 7, pp. 86-88, 1994.
- [17] Bailey, D. Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers. *Supercomputing Review*, August 1991, pp. 54-55.
- [18] Rogowitz, B., Treinish, L. and Bryson, S. How Not to Lie With Visualization. *Computers in Physics*, Vol. 10, No. 3, May/June 1996, pp. 268-273.
- [19] Card, S., Morgan, T. and Newell, A. *The Psychology of Human Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.
- [20] Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, vol. 63, 1956, pp. 81-97.
- [21] Bederson, B., et al. Pad++ A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, vol. 7, no. 1, 1996, pp 3-31.
- [22] Roesch, M. Snort: The Open Source Intrusion Detection System. <http://www.snort.org/>
- [23] Sniph's Cavern O' C0de, <http://www.stolenshoes.net/sniph/>.
- [24] Conti, G., Grizzard, J., Ahamad, M. and Owen, H. "Visual Exploration of Malicious Network Objects Using Semantic Zoom, Interactive Encoding and Dynamic Queries;" IEEE Symposium on Information Visualization - Workshop on Visualization for Computer Security (VizSEC), October 2005. (submitted, under review)
- [25] Cook, R. Visual Perception. From *Comparative Psychology: A Handbook*. Garland Publishing. Article available online at <http://www.pigeon.psy.tufts.edu/ecp.htm>.
- [26] Bach, M. Fifty-two Optical Illusions and Visual Phenomena. <http://www.michaelbach.de/ot/index.html>
- [27] Bonneh, Y., Cooperman, A. & Sagi, D. Motion Induced Blindness. *Nature*, vol. 411, 2001, pp. 798-801.
- [28] Healey, C. Perception in Visualization. Department of Computer Science, North Carolina State University. <http://www.csc.ncsu.edu/faculty/healey/PP/>
- [29] Morris, C. and Maisto, A. *Psychology: An Introduction*. 10th Edition, Prentice Hall. Summary available online at <http://cwx.prenhall.com/bookbind/pubbooks/morris2/chapter3/medialib/summary/1.html>
- [30] Hafner, K. Real Queasiness in Virtual Reality. *The New York Times Online*, November, 19, 1998.
- [31] Cyber-Defense Exercise. United States Military Academy. <http://www.itoc.usma.edu/cdx/>
- [32] Hardin, G. Photosensitive Epilepsy. *Epilepsy Matters*, Vol 9, No. 3, Summer 1998.
- [33] Lau, S. The Spinning Cube of Potential Doom. *Communications of the ACM*, Vol. 7, No. 46, June 2004.
- [34] Grace Project Homepage. <http://plasma-gate.weizmann.ac.il/Grace/>
- [35] Bounds, Darren. Packit - Network Injection and Capture. <http://packit.sourceforge.net/>
- [36] Windows Packet Capture Library. <http://winpcap.polito.it/>
- [37] Ptacek, T. and Newsham, T. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Secure Networks Inc., 1998.
- [38] Elias, L. Interface Illusions. *IEEE Security and Privacy*, Vol. 2, No. 6, November/December 2004, pp. 66-69.
- [39] Conti, G. Why Computer Scientists Should Attend Hacker Conferences. *Communications of the ACM*, March 2005.
- [40] Swiderski, F. and Snyder, W. Threat Modeling. Microsoft Press, 2004.

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, the Department of Defense or the United States Government.

This work was supported in part by the National Science Foundation Information Technology Research award 0121643.