Songs of cyberspace: an update on sonifications of network traffic to support situational awareness

Mark Ballora*, Nicklaus A Giacobe, David L. Hall
College of Information Sciences and Technology,
The Pennsylvania State University, University Park, PA 16802

ABSTRACT

Building on our previous work, we extend sonification techniques to common network security data. In this current work, we examine packet flow and the creation of socket connections between a requestor's IP address and port number with the server's IP address and port number. Our goals for the aural rendering are twofold: to make certain conditions immediately apparent to untrained listeners, and to create a sound model capable of enough nuance that there is the possibility of unexpected patterns becoming apparent to a seasoned listener. This system could be used to potentially provide better cognitive refinement capabilities for data fusion systems, especially when multiple sources of data at various levels of refinement are presented to the human analyst.

Keywords: Sonification, auditory display, SuperCollider, network security, data fusion, JDL process model

1. INTRODUCTION

In our 2010 presentation[1] we described design decisions involved in sonifying activity represented in web logs. The goal was to improve the ability to detect intrusion attempts. This presentation will demonstrate further steps taken in this work.

At the time of the 2010 presentation, our focus was rendering requestor IP addresses and server return codes as recorded in web logs. We have since decided that it is more useful to examine packet flow and the creation of socket connections between a requestor's IP address and port number with the server's IP address and port number. Our dataset is a common baseline used in intrusion detection system evaluations, created in 1998.[2] We have found that patterns associated with intrusion attempts such as port scans and denials of service are readily audible. However, we also acknowledge that the nature of cyber threats has evolved since 1998, and these particular intrusion attempts would likely be considered highly unsophisticated by hackers in 2011.

In our present implementation, we signal certain types of activity levels with obvious sound signals (for example, internal communications within a network vs. communications with external sites, or which ports are being requested). A GUI allows an operator to adjust relative volume levels among the various sound streams, as well as to enable or disable certain cues. We are also trying to create an implicit presence of ongoing exchanges with a more abstract sound model of the overall IP space, in the hopes that such a model will be capable of producing a nuanced soundscape in which unexpected patterns can emerge for experienced listeners. Our overall conceptual approach is based on the Joint Directors of Laboratories (JDL) Data Fusion Process Model,[3] a general-purpose reference model that describes the overall process of combining data from varied sources for purposes of gaining a better understanding of the situation being observed.[4]

*ballora@psu.edu; phone 1 814 863-3386; fax 1 814 865-6785

Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2011,
Proc. of SPIE Vol. 8064, 80640P · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.883443

2. EARLIER WORK

Earlier work in the area of pattern recognition and cyber security has dealt with entropy as applied to aggregates of activity. Kolmogorov Complexity, which is a measure of a text string's level of predictability, or its "noisiness," has been shown to be a factor in detecting FTP attacks.[5] Complexity was estimated by using the utility tcpdump (http://tcpdump.org/) to convert packets to ASCII strings, which were then compressed. Packets that included various intrusions were compressed to smaller sizes, indicating that the presence of intrusions can be found in patterns that are not present in the "noise" of normal network activity.

Later work[6] described using maximum entropy in the detection of everyday network traffic anomalies such as worms, port scans, and denial of service attacks. A multi-dimensional classification system for packets was created according to destination port numbers and protocol (TCP, UDP, TCP SYN, or TCP RST). A baseline distribution was determined showing normal distribution of packet traffic. If traffic differed from the baseline by a certain amount, an alarm was raised. Another study[7] also showed differences in Kolmogorov Complexity when packets were concatenated into equal time windows and compressed. Packets containing intrusions attempts were more compressible, again implying that reduced complexity implies likely presence of an intrusion attempt.

The concept of reduced complexity as an indicator of intrusion attempts plays into the strengths of sonification and auditory display. Sonifications exploit the auditory system's high sensitivity to temporal changes and pattern recognition. Therefore, these earlier studies suggest that effective sonification of network activity could be expected to sound different during intrusion attempts. Normal traffic would presumably sound random and "noisy," while traffic containing intrusion attempts would sound more "regular," producing pattern-based renderings with noticeable rhythms or melodies.

The simplest functional type of auditory display is *simple triggering*, wherein a recording or simple synthesized sound is played when certain conditions occur. The limitation of this approach is that simple triggering of audio files lacks what computer music researchers term "control intimacy,"[8] which refers to a musician's many microscopic gestures that affect the sound and character of a performance. Similarly, the goal of an effective sonification is that small variations in the data should create corresponding subtle variations in the character of the sound that is created.

A closer level of control intimacy is achieved with *parameter-based* sonifications,[9] whereby the data values are mapped to synthesizer sound characteristics such as oscillator frequency, filter cutoff frequency, volume, or stereo panning. This approach is sometimes called an "auditory scatter plot." Synthesized sounds need to be handled with care, as they can easily become shrill, buzzy, or annoying in other ways when heard over extended periods. Designing them effectively is a challenge of orchestration—that is, of creating pleasant and nuanced timbres that complement each other well when they are combined to represent data dimensions. A multi-dimensional data set is sonified as a multi-instrumental synthesizer ensemble, and an effective design must create auditory gestalts.[10] While this approach is the most common one employed by sonification researchers, its potential weakness is that the mappings are imposed and may not be inherently related to the data. This can produce an unnatural sound (one not similar to anything heard in nature) that has characteristics that need to be learned, which may make learning to hear patterns more difficult.

We instinctively recognize subtle changes in our sound environments. Hermann[11] makes the distinction between two listening types. *Musical Listening* concerns itself with tracking acoustic properties such as pitch and timbre, and is the type of listening employed in parameter-based sonification work. In contrast, *Analytical Everyday Listening* concerns itself with identifying sources of sounds. (To take an everyday example, we are far more likely think something like, "That sounds like a car motor," than to think, "That sounds like a low pitched, throbbing, lowpass filtered noise with amplitude modulation.") In contrast to musical listening, this complex process of making judgments as to object's characteristics is largely instinctive (unlearned). Our auditory system routinely makes accurate and quick ascertainments of the source and characteristics of objects. When hearing the sound of a rolling ball, for example, we can make generally correct estimations of the size, speed, and material of the ball, based on a complex interplay of auditory parameters in the form of frequency changes over time.

Extrapolating on Herrmann's argument, we recognize that this type of analytical listening, while unlearned, certainly benefits from further learning efforts. This is seen in professionals such as seasoned auto mechanics and sonar operators, who gain insights from practiced, focused listening for subtle changes in the sound of a system.

Herrmann proposes *model-based* sonification, which involves mapping data values to resonances and/or mechanics of a *physical model.*[12] A physical model is a computer synthesis technique based on wave equations describing vibrating

objects. This approach creates an inextricable relationship between the data dimensions and the resulting sound event. An example model-based sonification might be a multi-dimensional data set mapped to a theoretical grid of masses and springs, simulating a virtual instrument. As the data evolve, the character of the instrument undergoes vibrational changes. This allows the possibility of subtle patterns to emerge within the quality of the sound field that would be lost with realization based on simple triggering, and may not be as readily observable with parameter-based sonification methodologies.

3. THE DATA SET

Our dataset consists of simulated traffic dumps that have become standard reference documents in the area of intrusion detection.[2] Each data point is an array consisting of information about a packet as it is observed traversing the network at a certain point in time. Connections between two nodes on a network are represented at the socket layer, in which the requesting and receiving IP addresses and port numbers can be identified.

IP addresses identify machines on a network. IPv4 addresses are typically written in "dot decimal notation," which consists of four octet values, each ranging from 0 to 255, derived from a 32-bit network identification number. IP addresses are usually seen in a format something like 186.236.75.12.

While the requestor's port number is somewhat arbitrary, the receiver port numbers specify a particular type of server functionality, and many are standardized (for example, port 80 is always an http request to load a web page). A socket refers to the requestor's IP and port number in combination with the receiver's IP and port number.

We work with information about the socket connections each packet exchange reflects; each data point is an array, which includes the date and time of the exchange, the sender's IP address and port number, and the receiver's IP address and port number. From the documentation made available with the dataset, we have determined that the network being studied consists of two internal subnets within the ranges 192.168.0.0/16 and 172.16.0.0/16.

4. SONIFICATION STRATEGIES

There are, as yet, no standardized software tools for sonification work. Researchers typically make custom applications of some type of software sound synthesis (SWSS) program. Such programs, the basis of computer music,[13] have existed since the 1950s. We sonify the data with the SWSS program SuperCollider (http://supercollider.sourceforge.net/), a specialized programming language designed for real-time audio applications. SuperCollider displays exceptional versatility, with capabilities including real time signal processing, algorithmic composition and inter-machine remote control. SuperCollider is well suited to our sonification model because of its computational efficiency, its array and list processing capability, its methodology for generating musical events (Streams) according to a programmer's instructions, and its interactive potential through the use of custom-designed graphical user interfaces (GUIs).

With a GUI modeled after the familiar interface of an audio mixer, we are attempting to apply the design mantra of "overview, Filter/Zoom and Details-on-Demand," which has been suggested as a guideline for organizing and displaying data in an information visualization system.[14] As we have pointed out,[15] this design principle is applicable to network security. The security analyst needs a useful overview of the defended system's current state, and the interface should provide the ability to zoom into specific regions of the defended network, select specific details from the raw data, and be able to filter them. A mixing board-like interface allows us to select elements to render or mute, and to adjust relative volume levels of the various sound streams.

4.1 Time Scaling

The rate at which events are rendered is based on the relative times between timestamps, multiplied by a scalar. This fundamental design element is unchanged from our 2010 description. The scalar is controllable by a slider on the GUI, allowing users to speed up or slow down the rendering rate at will. With time between packet time stamps being reflected as well, periods of higher or lower relative activity can be easily recognized, depending on whether one hears

sparse, occasional events or a flurry of sound. A pre-existing data set consisting of many hours of activity can be heard over a timescale on the order of minutes or seconds, depending on the iteration rate the listener chooses.

4.2 Level 1 Listening (the immediately obvious)

There are a number of activity types that an analyst might be interested in monitoring. One is the amount of internal vs. external activity: how much traffic is internal within each of the two subnets? or between the subnets? How much external traffic is going to each? Each of these five conditions is represented by a "whooshing" sound, with the pitch and reverberation on the whoosh makes each readily apparent. The relative volumes of these five whooshes are controllable via the GUI, allowing an analyst to listen to any or none of them at chosen volume levels.

Another feature of interest might be port requests. Analysts can select a certain port to monitor, which causes requests to that port to create a particular humming sound. As a default, we have determined that the commonly used ports shown in Table 1 are likely *not* of interest to an analyst, as requests to them represent normal, expected activity.

Table 1. Commonly used ports

Port	Request type
7	ECHO
20	FTP -- Data
21	FTP -- Control
22	SSH Remote Login Protocol
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
37	Time
53	Domain Name System (DNS)
69	Trivial File Transfer Protocol (TFTP)
79	Finger
80	HTTP
110	POP3
115	Simple File Transfer Protocol (SFTP)
137	NetBIOS Name Service
139	NetBIOS Datagram Service
143	Interim Mail Access Protocol (IMAP)
156	SQL Server
161	SNMP
194	Internet Relay Chat (IRC)
389	Lightweight Directory Access Protocol (LDAP)
443	HTTPS
445	Microsoft-DS
458	Apple QuickTime
546	DHCP Client
547	DHCP Server

In the default turnkey mode, ports other than those listed in Table 1 are represented by a humming sound, making it immediately apparent whether unusual activity is not occurring, is occurring sporadically, or is occurring a great deal. Systematic requests to unusual ports, which may be indicative of intrusion attempts, may be perceptible as melodies or rhythms. Should an analyst wish to monitor the activity on one of the ports in Table 1, s/he may choose to do so via the GUI, and a timbrally different humming sound indicates traffic on a chosen port.

4.3 Level 2 Listening (the subtle and nuanced)

Another layer of our sonification approach represents an attempt to capitalize on both musical and everyday listening by creating a familiar sound that is capable of subtle yet easily perceptible changes. Our goal is to build the potential for additional subtlety, with which experienced listeners can gain increased levels of understanding.

A gong is an instrument possessing a high degree of subtlety. The sound of a gong is commonly described in two stages, informally called the "rumble" and the "sizzle." By studying spectrograms of gongs, we have created a synthesized gong-like sound, which is based on eight frequency values (some of which are duplicated and multiplied by non-integer values to create the inharmonic sound inherent in gongs). A force factor between 0 and 1 also affects the amplitude of the frequencies and the timing and relative amplitude of the sizzle, so that higher force values produce the sound of a gong being hit with greater strength.

We map each socket exchange to a strike of the gong, with characteristics of the strike mapped to the packets' source and destination IP octets. The four values of the sending IP octet are parameters that form the rumble's timbre. The sizzle's timbre is created from the four values of the receiving octet.

Each strike's force value is derived from how often identical sockets are connected. We keep a running list of packets previously received, with the length being adjustable. With the iteration of each packet, the list is scanned for the same socket values. A higher percentage of these values in the list indicates an ongoing "conversation," and produces a stronger force value for the gong.

The panning of each strike is more abstract. Ideally, we would like to create a stereo pan position based on the geographic locations of the sender, so that packets received from Asia would appear at a very different pan position than packets received from South America, for example. However, obtaining geographic information on an IP address is not an automatic process, and requires querying external registries. As an alternative, we simply create a unique pan position based on the sender-receiver IP values, so that each combination of IP octets is mapped to a unique stereo pan position between -1 for extreme left and 1 for extreme right.

In using a gong timbre, we are hoping to create an implicit step from a Level 1 to Level 2 of the JDL fusion process.[4] In Level 1, entities are detected and characterized. The players on the field are identified—from the host itself to various data flows via normal socket connections. In Level 2, the current system state is summarized based on the active entities.

Many of the data points in our array represent continuing connections, packet exchange by packet exchange. The individual packets represent raw, de-contextualized data. They are analogous to individual pixels from an image or video. This affects the choice of sound types that can effectively represent it. Percussive sounds are disadvantageous because they become distracting, and tend to create an unpleasant listening experience (we refer to this informally as the "woodpecker effect"). Rather, packet exchanges need to be represented by sounds with long transients, so that successive events blend together as an extended presence. With the gong timbre, randomly distributed packets received sound like a low rumble appearing throughout the stereo field. The presence of distinct entities, in the form of ongoing connections, produce more forceful gong strikes, giving an immediate indication of their presence. This is akin to being able to recognize figures from a video from the aggregate of pixels. The gong model is meant to bring out continuing presences in the data, implicitly identifying discrete entities.

This design is meant to incorporate a great deal of frequency, timbre, and panning subtlety of the instrument. Given that IPv4 has 2^{32} possible addresses, we have a large amount of resolution in these sound parameters, producing changes that are very much microscopic in nature. This produces a subtle control source that is often difficult to obtain with other, more standard synthesis techniques. The changes in each parameter are unlikely to be terribly informative in and of themselves. IPv6's possible range of 2^{128} addresses magnifies this issue. Hearing small changes in pan position, for example, would likely be impossible, but in combination with the other pitch and timbral changes that might accompany it, the panning might add a useful layer of discrimination. By assigning characteristics of the data to as many parameters of the sound as possible, we leave the door open for unexpected auditory gestalts to emerge, which may enable analytical everyday listening in unforeseen ways.

5. PARTICULAR CHALLENGES OF CYBER SECURITY

As we identified in previous work, the JDL process model is helpful as a general-purpose reference model for representing sensor data in the domain of cyber security.[15] The challenges inherent in this domain make it appropriate for an auditory rendering as a potential mechanism for cognitive refinement. One particular challenge has to do with high data rates causing increased cognitive load. The combination of the text-based format commonly used in cyber security systems coupled with the high false alert rates can lead to analysts being overwhelmed and unable to ferret out real intrusions and attacks from the deluge of information. The Level 5 fusion process indicates that the HCI interface should provide access to and human control at each level of the fusion process,[16] but the question is how to do so without overwhelming the analyst with the details.

Another challenge has to do with a lack of a common model of understanding the organization of the network. In the field of cyber security, there is no generally accepted mental model of the problem space. Different analysts and fusion system designers may have different assumptions about what a defended network should "look like,"[15] and therefore by extension, should "sound like" in auditory renderings suggested by this work. There is unlikely to be a single optimal visual or audible representation.

REFERENCES

[1] Ballora, M. and Hall, D.L., "Do you see what I hear? Experiments in multi-channel sound and 3D visualization for network monitoring." Proc. SPIE 7709, 77090J (2010).

[2] Lippmann, R. P., et al. "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation." Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX) 2, 12-26 (2000).

[3] Kessler, O., Askin, K., Beck, N., Lynch, J., White, F., Buede, D., Hall, D., and Llinas, J. [Functional description of the data fusion process], Office of Naval Technology Naval Air Development Center, Warminster PA, (1991).

[4] Steinberg, A.N., Bowman, C.L., and White, F.E. "Revisions to the JDL data fusion model," Proc. SPIE 3719, 430-441 (1999).

[5] Evans, S.C. and Barnett, B. "Network Security Through Conservation of Complexity," Proceedings of the IEEE Military Communications Conference (MILCOM) 2002, 2, 1133-1138 (2002).

[6] Gu, Y, McCallum, A. and Towsley, D. "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation," ICM '05 Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement, 345-350 (2005).

[7] Eiland, E.E. and Liebrock. L.M. "An Application of Information Theory to Intrusion Detection," Proceedings of the Fourth IEEE International Workshop on Information Assurance (IWIA '06), 119-134 (2006).

[8] Moore, F.R. [Elements of Computer Music], PTR Prentice Hall, Englewood Cliffs NJ, (1990).

[9] Kramer, G. (Ed.) [Auditory Display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVIII], Addison Wesley, Reading MA, (1994).

[10] Bregman, A. [Auditory Scene Analysis: The Perceptual Organization of Sound], MIT Press, Cambridge MA, (1990).

[11] Hermann, T. [Sonification for Exploratory Data Analysis], Ph.D dissertation Bielefeld University, (2002).

[12] Smith, J.O. "Physical Modeling Using Digital Waveguides," Computer Music Journal 16(2), 74-91 (1992).

[13] Roads, C. [The Computer Music Tutorial], MIT Press, Cambridge MA, (1996).

[14] Shneiderman, B. "The eyes have it: A task by data type taxonomy for information visualizations," 1996 IEEE Symposium on Visual Languages, 336-343 (1996).

[15] Giacobe, N. "Application of the JDL data fusion process model for cyber security," Proc. SPIE 7710, 77100R (2010).

[16] Blasch, E. P., and Plano, S. "JDL level 5 fusion model: user refinement issues and applications in group tracking," Proc. SPIE 4729, 270-279 (2002).