

RESEARCH ARTICLE

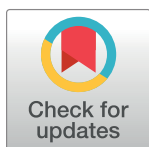
Sonification of network traffic flow for monitoring and situational awareness

Mohamed Debashi^{✉*}, Paul Vickers^{✉*}

Department of Computer and Information Sciences, Northumbria University, Newcastle-upon-Tyne, United Kingdom

[✉] These authors contributed equally to this work.

* mohamed.debashi@northumbria.ac.uk (MD); paul.vickers@northumbria.ac.uk (PV)



OPEN ACCESS

Citation: Debashi M, Vickers P (2018) Sonification of network traffic flow for monitoring and situational awareness. PLoS ONE 13(4): e0195948. <https://doi.org/10.1371/journal.pone.0195948>

Editor: Richard Mankin, US Department of Agriculture, UNITED STATES

Received: August 10, 2017

Accepted: March 22, 2018

Published: April 19, 2018

Copyright: © 2018 Debashi, Vickers. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by Libyan Embassy Cultural Attache, London (<http://culturalaffairs.libyanembassy.org/En/>), Student Number: 10233; Mohamed Debashi is the author who received the funding. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Maintaining situational awareness of what is happening within a computer network is challenging, not only because the behaviour happens within machines, but also because data traffic speeds and volumes are beyond human ability to process. Visualisation techniques are widely used to present information about network traffic dynamics. Although they provide operators with an overall view and specific information about particular traffic or attacks on the network, they often still fail to represent the events in an understandable way. Also, because they require visual attention they are not well suited to continuous monitoring scenarios in which network administrators must carry out other tasks. Here we present SoNSTAR (Sonification of Networks for SiTuational AwaReness), a real-time sonification system for monitoring computer networks to support network administrators' situational awareness. SoNSTAR provides an auditory representation of all the TCP/IP traffic within a network based on the different traffic flows between network hosts. A user study showed that SoNSTAR raises situational awareness levels by enabling operators to understand network behaviour and with the benefit of lower workload demands (as measured by the NASA TLX method) than visual techniques. SoNSTAR identifies network traffic features by inspecting the status flags of TCP/IP packet headers. Combinations of these features define particular traffic events which are mapped to recorded sounds to generate a soundscape that represents the real-time status of the network traffic environment. The sequence, timing, and loudness of the different sounds allow the network to be monitored and anomalous behaviour to be detected without the need to continuously watch a monitor screen.

Introduction

Visualisation has been used as a tool for monitoring networks in order to raise situational awareness levels. The static and dynamic visualisation of total and subtotal traffic information (such as bandwidth, speed and current performance) do not allow administrators to acquire a deep and clear understanding of their current network state. This is because attacks can appear like normal traffic and there are no specific rules that could enable administrators to set their network up to prevent or monitor all attacks. Furthermore, each network is unique and what

is normal behaviour in one network may be anomalous in another. Therefore, network administrators need tools to provide information in a way which helps them to build a solid understanding of their network environment's behaviour. Unfortunately, existing popular tools such as intrusion detection systems (IDS) and firewalls do not specify why and how certain events happened.

Visualisation and intrusion detection systems do not provide the protocol flow granularity required to understand how flows are behaving inside a network or why a security system generates false positive alerts or why specific alarms were raised. IDSs detect intrusions and record them to log files which network administrators must then inspect to try to understand the situation. Many IDSs send an email to the administrator for each intrusion record or incident and the volume of emails increases with the scale of the network. It is quite difficult to understand the relevance of aggregate records when receiving only the alarms for individual intrusion records. Modern attacks are sophisticated and can involve a range of techniques and methods. Thus, real time situational awareness is required for an overall understanding of the situation especially when real time intelligence and intuitive solutions are required. The graphical user interfaces (GUIs) of today's network monitoring and intrusion detection systems present information very superficially. For example, the time sequence of numbers of intrusions or incidents of the whole traffic domain may be visualised as polygonal charts. The operator may be required to perform many operations to explore detailed information, but in many cases network administrators are too busy to monitor the GUI. Moreover, when using visualisation tools administrators must look at a screen. Loss of concentration, visual fatigue, temporal demand and frustration increase when monitoring a screen for long periods. Extra screens will be required for additional staff. In addition, the huge volumes of data which need to be processed and presented cannot be visualised in real-time unless data reduction techniques are used.

Network measurement tools include hardware and software approaches to collect data and analyse traffic at different protocol levels. Network traffic analysers collect real-time data and perform online analysis and the majority of these systems use graphical displays to represent live traffic data.

Hildebrandt [1] proposed enhancing visualisation monitoring with sonification techniques because humans are sensitive to even small changes in the rhythms and sequences of sounds. Sonification may be defined as:

...the use of non-speech audio to convey information. More specifically, sonification is the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation [2, p. 5].

This makes sonification highly suitable for conveying information that changes over time. In the last few years there have been several attempts to develop network sonification systems in order to support network monitoring.

In order for sonification to serve network monitoring purposes administrators need to have a clear understanding of what is going on in their network environment so they can take appropriate action and prevent malicious activities and misuse of resources. The traffic volumes passing through today's networks are huge which makes it more difficult for them to be represented visually. However, if we enable people to sense and interact with the cyber environment and let the human brain do part of the processing work and to adjust the sound generated to ease analysis this may allow more about the cyber environment to be learned.

Situational awareness (SA)

Endsley defined situational awareness (SA) as: ‘the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future’ [3, p. 36]. Because it exists within computers and communication networks the cyber environment severely constrains human perception and so we are reliant on tools to provide perceptual access to what is happening within the network. Vickers et al. described the situation thus:

Many tools on which we rely for situational awareness are focused on specific detail. The peripheral vision (based on a range of senses) on which our instinctive threat models are based is very narrow when canalised by the tools we use to monitor the network environment. The majority of these tools use primarily visual cues (with the exception of alarms) to communicate situational awareness to operators. Put simply, situational awareness is the means by which protagonists in a particular environment perceive what is going on around them (including hostile, friendly, and environmental events), and understand the implications of these events in sufficient time to take appropriate action [4, p. 13].

Boyd’s OODA (observe, orient, decide, act) loop theory [5] has added more depth to the understanding of situational awareness. Boyd’s theory is based on his study of the decision making of combat pilots and the first stage (observe) involves taking in information about features of the environment. The orientation stage directs attention towards an adversary. The next stage involves deciding what action to take which is followed by acting upon that decision.

No system can implement the best security measures without interaction with people, but it is difficult to maintain high SA levels [6]. The real-time monitoring of the end-to-end flows and connections in a network is vital to allow better observation and orientation for faster decisions and actions so as to maintain healthy network resources in the face of constant changes in attack methods, motives and behaviours. In general, this work requires high experience and intelligence. Humans are by nature good listeners and are capable of processing auditory events through their experience and intelligence which makes them capable of using sonification for maintaining SA of sensitive cyber environments.

Sonification and computer network monitoring

Sonification is introduced to enable a listener to recognise changes in activities and patterns to enhance comprehension and projection as part of the SA process. Sound allows a network administrator to continue monitoring the network while performing other tasks [7] which may, in turn, decrease frustration and visual fatigue rates. The concept of changed network behaviour as an indicator of unhealthy activity or intrusion attempts is a reasonable motive for using sonification [8]. Sonification can have advantages over visualisation in different sectors. For example, real-time sonification using parameter mapping methods is used in the health sector. A recent study showed positive results and a high potential for using real-time auditory feedback-oriented training devices for fitness training or physical rehabilitation to increase the awareness of physiological responses [9].

There is a continuing threat of intrusion, denial of service attacks, or numerous other abuses of network resources which requires the monitoring of traffic flows passing through a

Table 1. Some network traffic protocols.

| Protocol | Definition | Layer | Usage |
|----------|------------------------------------|-------------|---|
| TCP | Transmission Control Protocol | Transport | Provides communication services between the Internet layer and the Applications layer and the Applications layer. |
| IP | Internet Protocol | Internet | Addressing hosts and putting data into packets (datagrams). |
| UDP | User Datagram Protocol | Transport | A connectionless communication service without handshaking. |
| ICMP | Internet Control Message Protocol | Internet | For network devices to send operational and error messages. |
| SNMP | Simple Network Management Protocol | Application | For collecting information about and managing devices on a network. |

The four network traffic protocols used in this paper.

<https://doi.org/10.1371/journal.pone.0195948.t001>

network [10]. The size of modern network traffic volumes makes it much harder to present real-time information visually [11]. However, there is no clear consensus yet about the pattern of cyber-attacks [12]. The assumption is that these behaviours and the rhythm associated with each type of attack should sound different or at least provide an indication of some features of any attack. Network traffic comprises packets of data which conform to specific protocols from the Internet protocol suite. The traffic protocols discussed in this paper are summarised in Table 1. For a fuller overview see [13].

Worrall [14] has described the NetSon project from its exploratory stage to a real-time sonification of network metadata. The project used the information extracted from data volumes by employing sampling techniques to extract a small group of data packets using the sFlow tool [15]. This method provides information about the network's traffic flow rate by making a sonification of sFlow packet data of device traffic (such as printers and servers) and load balancing traffic. NetSon also provides information to identify internal and external IP addresses. This tool could be used to support network traffic measurement tools or to identify and classify IP addresses for security purposes.

Mancuso et al. [16] used sonification to help 'cyber defenders' to detect evidence of cyber attacks by using data collected by Wireshark [17]. The data were used offline and the source and destination IP addresses were sonified using pairs of sequential musical notes separated by 100 ms, while packet size was used to control the loudness of the sound. An experiment revealed no improvement in operator performance when using the sonifications. However, it could be argued that sonification should be tailored so that traffic with specific signatures should sound different from other normal packets, or even that sound should be generated only for malicious signatures. This might increase performance and decrease the stress of the operator.

Vickers et al. [4, 11, 18] applied sonification to the inherent self-organised criticality observed in network traffic. Standard packet capture tools were used to gather network traffic which was then passed to the socs (self-organised criticality sonification) system which sonified the log returns of packet sizes at regular user-specified intervals. The extracted log returns provide information about the behaviour changes in the network. Knowledge of this behaviour could be used to detect unwanted behaviour. This system has potential to support both network traffic measurement and intrusion detection tools.

Wolf and Fiebrink [19] developed SonNet, a programming interface for sonifying computer network data. The prime motivation behind SonNet was to lower the practical barriers for artists and sound designers interested in accessing network data to create music. SonNet involves packet sniffing and offers network state analysis and easy access to

computer network data for composers. The tool supports the sonification of data using the UDP and TCP protocols. SonNet extracts network data at various levels from packet level information to network state information. Level 1 contains information about a single packet, level 2 contains information generated by computing and analysing the single packet information, and level 3 contains information about multiple packets. In a similar vein, Rutz et al. [20] introduced the SysSon platform for developing sonification applications for different types of users from domain scientists to sonification researchers, composers and sound artists.

InteNtion (Interactive Network Sonification) [21] is a project targeted at mapping network traffic activities to a musical aesthetic. Network traffic data is converted into MIDI messages and then sent to dedicated synthesisers to generate a dynamic mix of sounds as an interactive soundscape. The system uses Internet protocols including TCP and UDP segments, using very low-level packet information such as packet size, source and destination IP addresses and type of service. The work is still considered to be experimental and the system needs more development and better mapping to support network traffic monitoring. However, it provides an innovative way to monitor a network by using the entire data flow to create music.

Earlier work done by Ballora and Hall [22] explored the detection of intrusion signatures and patterns using human aural and visual recognition abilities to detect intrusions in real-time. IP addresses and return codes were used to generate sound as an informative and unobtrusive listening environment to develop web traffic SA. Ballora et al. [8] conducted another sonification experiment with a computer network based on socket connections using information such as the date and time of exchange and the sender's and receiver's IP addresses and port numbers. Ballora et al. [23] have also described the use of sonification in the detection of anomalous events. Sonification should enable the listener to differentiate between normal and anomalous network behaviour and to develop an understanding of what is actually happening in the network.

Kimoto and Ohno [24] introduced the Stetho network sonification system which was aimed at system administrators. NetSound was built on top of Stetho as a tool for end users. Stetho used network traffic information to generate sounds which provide the network administrator with information about the traffic. Stetho reads the tcpdump [25] commands, then uses them in regular expressions to generate corresponding MIDI events. Stetho processes each packet in the traffic. However, Stetho failed to detect all events and intrusions. Delays in sound generation and poor MIDI messages generated further problems.

Chafe and Leistikow [26] developed a tool for the measurement of round trip time when using a sequence of standard ping utility events to gather information about the quality of service of a network path, such as packet loss. They discussed the need to evaluate paths which carry interactive media streams in collaborative environments. They designed a stream-based method for the direct display of the critical qualities to the ear by continuously driving a bidirectional connection to create sound waves. They changed the network path to acoustic medium which their probe sets into vibration. Temporal levels of musical foreground, middle-ground and background are heard in the melodies generated from correspondence data.

SoNSTAR

The sonification of high-speed computer networks demands both high throughput and flexibility to handle and recognise new threats. It is possible that sonification is a viable solution

to this problem and could allow an administrator to listen in real-time to the state of each traffic flow. As a solution to these problems and issues, we propose SoNSTAR—Sonification of Networks for SiTuational AwaReness—to be used by network administrators as a monitoring tool to facilitate the acquisition and maintenance of network situational awareness. SoNSTAR would assist with the maintenance of security, awareness of anomalous events such as attacks, maintenance of network health through monitoring and tuning, and increasing the understanding of the cyber environment which is vital for network management the use of diagnosis to support the recognition phase in the situational awareness process.

A traffic flow is a flow of packets in a single connection between a source computer and a destination [27]. A single flow can be identified within a certain time period by its source and destination IP addresses and ports and its protocol type and layer (such as TCP, UDP and ICMP). As part of our technical solution, we have created a new flow type called IP flow which is identified within a certain time period by its source and destination IP addresses and protocol only (no port information). Thus, SoNSTAR uses these two flow types (traffic flow and IP flow).

SoNSTAR uses events to generate sounds. A flow event is a change in the behaviour or operation of a flow (traffic or IP). A single event represents a combination of features of a flow while a set of events represents flow behaviour which, in turn, represents the state of the network traffic.

In the TCP protocol, the packet header contains nine control flags, six of which (FIN, SYN, RST, PSH, ACK and URG) are used by SoNSTAR. Values of 1 and 0 denote whether a flag is set or unset, and the packet's type is determined by those flags that are set. A packet's type determines its role and function within the network traffic. Therefore, SoNSTAR collects counts of each packet type for both traffic- and IP-flows. A flow's status is determined by the respective packet type counts. SoNSTAR allows its user to listen to the status of the flows in the traffic by playing sounds that represent the flow behaviours.

Thus, SoNSTAR makes information about traffic perceptible allowing the network administrator to make decisions about network operation on the basis of recognising the sounds that describe the network environment. SoNSTAR allows users to set specific sounds for different flow status types and to tune the thresholds for triggering the sounds. What makes SoNSTAR distinctive compared to other available tools is that it allows the user to monitor general and specific behaviours in a human understandable form.

When designing a sonification system with the purpose of monitoring a system or network activity to gather administratively useful information, the design will involve a number of conditions and requirements. The sound has to support extended periods of listening, changes in status have to be easily grasped and accidental events have to be immediately noticed [24]. SoNSTAR sonifies each flow in a connection and collects information about the connection state by periodically gathering online flag information from each flow. Traffic features are extracted from the flag information aggregations and SoNSTAR then represents these features in a soundscape.

The term “soundscape” was introduced by Murray Schafer [28] and describes the sonic properties of landscapes. Sounds are a continuous and active property of all landscapes and soundscape ecology is generated from the sounds and spatial temporal patterns as they occur in a landscape environment, where each sound has special ecological characteristics [29]. In SoNSTAR the network environment is transformed into an acoustic environment as a soundscape and the combinations of sounds represent the current state of the network, just as the

combinations of sounds in a landscape provide information about what is happening in the environment.

SoNSTAR transforms the network environment to the soundscape of a forest (though it is fully configurable and allows any other soundscape to be used as desired). Just as a person would be able to infer information about what is happening in a forest by the sounds they hear, sounds in the soundscape represent events and unexpected or particularly loud sounds can draw the listener's attention to traffic behaviour that is out of the ordinary.

Using recorded sounds in sonification can be difficult, as there are limitations on how recordings can be used to represent traffic while still sounding realistic [30]. However, the use of recorded sounds is better than synthesised sounds, because it enables users to link events to familiar and understandable sounds. Sounds from a natural environment such as birds tweeting or animal sounds are easier to describe than artificially synthesized tones which may rely on specific terminology such as frequency and timbre [31]. The sounds provide us with immediate awareness of the types of events that are happening:

Modern cognitive science believes that to be able to read sound in this way, listener must have some inner understanding of how the properties of physical events are reflected in the sounds they make [32, p. xviii].

Therefore, a monitoring operator requires a good understanding of communication protocols and theoretical and practical knowledge about the expected behaviour in computer networks. SoNSTAR allows the user to make a relation between the meaning of the recorded sound and the event mapped to within the network environment. For example, a forest on a normal day will produce sounds of typical birds and animals, perhaps with a very light breeze. These events can be used to describe the normal state of a computer network environment, while human-made sounds, the sounds of predators and changes in weather (such as rain and thunder) can be used to represent abnormal or malicious network activity. SoNSTAR sonifications are generated using an event mapping method based on flag state information collected from each TCP packet for each flow in the network. This specialised abstraction of network features is extracted from the raw flow packets and transformed into classified sound groups of natural and human-made sounds.

SoNSTAR and network traffic monitoring

Commonly, administrators try to look directly at network traffic to understand it using tools such as Wireshark [17]. Network traffic volumes can be huge and the majority of the traffic involves normal data packets travelling between legitimate users on the network or across the Internet. TCP/IP packets carry control flags to allow the data to be received in sequence and to protect it from loss. In TCP/IP, if receipt of any packet is not confirmed by the destination, it will be sent again. In contrast, in the UDP protocol any packet sent will be considered as received and packets will be processed in the order they arrive regardless of whether the routing has caused them to be received out of sequence. In TCP/IP approximately 30%–40% of traffic concerns packets which are very important to administrators for enabling them to understand immediately what is happening in their network environment [33]. This means that the TCP/IP control packets SYN, SYN ACK, ACK, FIN and RST provide most of the information about network traffic state. UDP packets have to be monitored in such a way that allows administrators to recognise the current state. TCP/IP traffic accounts for more than

85% of packets entering and leaving a system or computer network [33, 34]; therefore, TCP traffic is considered a priority.

Network administrators typically identify anomalies in traffic from two sources. The first is simple network management protocol (SNMP) data from queries to network nodes. However, the data collected from the SNMP management information base (MIB) is wide ranging, and contains activity statistics such as total packets transmitted at a node. This source can only provide statistics about volumes of packets and bytes which provide useful information but cannot be used to understand the behaviour in the traffic flows and connections in the network. The second source is the monitoring of end-to-end packets, flows or connections. This data contains protocol-level information. This second source is typically used by intrusion detection systems. These two sources offer a practical base for the identification and recognition of anomalies as part of situational awareness [35].

SoNSTAR uses the second source and collects data by sniffing the traffic passing through a switch or a router from the mirroring outlet in real time or by reading stored PCAP files captured by any other available packet sniffing programs. The sniffer act as a sensor that periodically collects traffic information.

Several types of monitoring systems use network usage patterns for detection, measuring usage and summarising usage statistics based on user-defined parameters, and contrasting measurement aggregates with predefined thresholds and then responding when thresholds are met or exceeded or following queries from a security analyst [36].

Most detection methods (especially IDS) depend on packet headers or the payload or a combination of both to detect attacks and malicious activity. In anomaly-based systems analysis of the packet's payload is used to differentiate between normal traffic and anomalous activity. Signature-based systems rely on matching patterns with a database of the signatures of known attacks. The advantage of anomaly-based systems is that, unlike signature-based systems, they can detect attacks without any delay since new attacks can be detected as soon as they happen, while signature-based systems cannot detect novel attacks and can only match against known attack signatures [10, 37]. While anomaly-based systems can detect novel attacks they generate more false positive results and so risk blocking legitimate activity. Identifying the state of traffic from encrypted applications is a critical issue for numerous network tasks. In-depth packet inspection requires decryption in most cases, and this would affect any detection mechanism especially when trying to operate in real time [38–40].

SoNSTAR uses packet header information to generate sounds which periodically represent the status of aggregated packet information for multiple flows in the network. It is an anomaly-based system which generates different sounds according to the network state. This method can be used to provide a general or specific sonic representation of the traffic behaviour. Any changes in sound combinations then represent a new state or behaviour. An advantage of this approach is that an administrator using SoNSTAR can interact with the system and change and create the features to be sonified and assign sounds to those features. SoNSTAR is an additional tool that enables administrators to discover changes in and learn more about their environment in a way that enables the human mind to comprehend the mechanism of these changes and their causes.

A security system using real-time monitoring for situational awareness has to show changes in flow and connection states as they happen and provide an indication to the administrator about immediate events. SoNSTAR targets this type of monitoring to support existing security tools, acting as an additional tool aimed at raising situational awareness levels.

SoNSTAR design

Computer network defence requires analysts to detect both known and novel forms of attack in massive volumes of network data. Visualisation tools would potentially assist in the discovery of suspicious patterns of network activity, but few analysts so far are leveraging sonification technologies in their current security practice. We have designed SoNSTAR to suit the work practices and operational environments of those analysts.

The work's novelty consists in sonifying in real-time the extracted features of network traffic based on the control flag status of the packet header and the techniques developed to handle the interaction of the user with the system with the aim of increasing situational awareness levels. SoNSTAR inspects the flag statuses of each packet in the flows and extracts features by periodically counting each packet type and counting the number of flows and then uses this information to control the resulting soundscape. This results in a system that is complementary to and more informative than visualisation methods, but which can provide only limited goal-oriented information. This type of sonification, which allows the representation of large traffic volumes by representing traffic flow and IP flow states to reduce the amount of traffic information presented to the user, has not been done before.

In many systems, changes in performance could be used to indicate the vulnerability or robustness of a computer network [41]. Equally, changes of sounds could be used to indicate changes in network behaviour. The first goal of the design of the sonification system as part of the situational awareness process is either to monitor network assets or the network gateway and to find a way to sonify network component activity and traffic behaviour to enable the listener to detect any misuse or anomalous behaviour. This anomaly detection approach must first determine the normal behaviour of the object being monitored, and then use deviations from this baseline to build experience and knowledge to detect and identify possible malicious activities.

Monitoring tools try to present administrators with a complete representation of their complex network. Better network monitoring tools should allow administrators to perceive changes in their network in order to allow them to react immediately, and learn and understand more about the cyber environment. A real-time sonification monitoring tool should be able to do or assist with the following:

- **Identify and recognise malicious traffic:** Malicious traffic such as probes and denial of service attacks should be indicated.
- **Provide information about incidents or changes in behaviour:** An incident or change in traffic behaviour should be reported to allow the user to recognise which flows are malicious.
- **Represent network behaviour sonically and in a non-fatiguing and non-annoying way:** Sounds representing states have to be easily recognised and linked together by the user to allow comprehension as part of the situational awareness process.
- **Offer practicality:** Use of the system should be convenient for both incident response and real-time monitoring.
- **Indicate compromised machines:** A machine compromised by a hacker or malicious software such as worms or viruses should be indicated when ever possible.
- **Offer high throughput and flexibility:** The system should be able to handle large amounts of data in a timely manner and its operation should not be CPU-intensive.

Monitoring requirement of the tool

The output of such a system is meant to help the user to identify changes in traffic behaviour or recognise attacks immediately as part of the situational awareness process. This awareness is important and its lack could be costly and decisive for an organisation. It is important that the monitoring tool assists the user to analyse and interpret the traffic in the correct manner. Various common requirements for forensic analysis, visualisation and sonification monitoring tools are given in the literature [42–46] and the following requirements are based on them:

- **Usability:** Data sonified at the lowest packet information level would result in huge volumes of information which would be too difficult for the user to interpret. Therefore, the representation of this information by sound has to be designed so that the user can recognise normal and malicious activities. The information has to be represented by distinct sounds so that it is not misinterpreted.
- **Cognitive processes:** The time it takes to learn how to use and understand the system should be minimised.
- **Comprehensive:** The sounds generated have to represent, as far as possible, all output data at a given level of abstraction.
- **Accuracy:** The tool should guarantee that the output sounds are clearly distinguishable and that the margin of similarity should be presented to the user, for example as a log file, so that it can be confirmed and interpreted correctly.
- **Deterministic:** The tool should always generate the same output sounds when presented with the same input dataset or traffic when using same sound design.
- **Verifiable:** To ensure the accuracy of the tool, it should be possible to verify the results. This could be done manually or by using another tool.

Design solution

This section considers a design that can fulfil the requirements of real-time monitoring for situational awareness. The practical issues associated with the selected design are also discussed.

The SoNSTAR architecture diagram is illustrated in Fig 1. The system is implemented in Python using the pcap and dpkt libraries and Max/MSP [47]. The Python engine captures and processes the packet information and passes data to the Max/MSP patch which generates the audio.

SoNSTAR uses time a window period to arrange and control the timing of the operation of each process within the system (see Fig 2). SoNSTAR reads packets and unpacks them and filters the TCP packets and extracts counts during time window X. At the end of each time window, features are combined to generate higher-level aggregate features. The selected features are then represented as recorded sounds. These sounds are played during the next time window Y.

The time window is user defined and can be any value above 5 s which is the minimum required to allow SoNSTAR to collect sufficient packets to count target events. During development SoNSTAR was applied to a variety of network traffic (both datasets and live traffic) and a 20 s window was found to be a good default value to enable enough features to be collected for setting distinctive events. Because the size of a network determines only the number of flows passing through it, it does not affect the time window duration or SoNSTAR's ability to collect features and handle event conditions. Thus, it is left to the user to decide which time

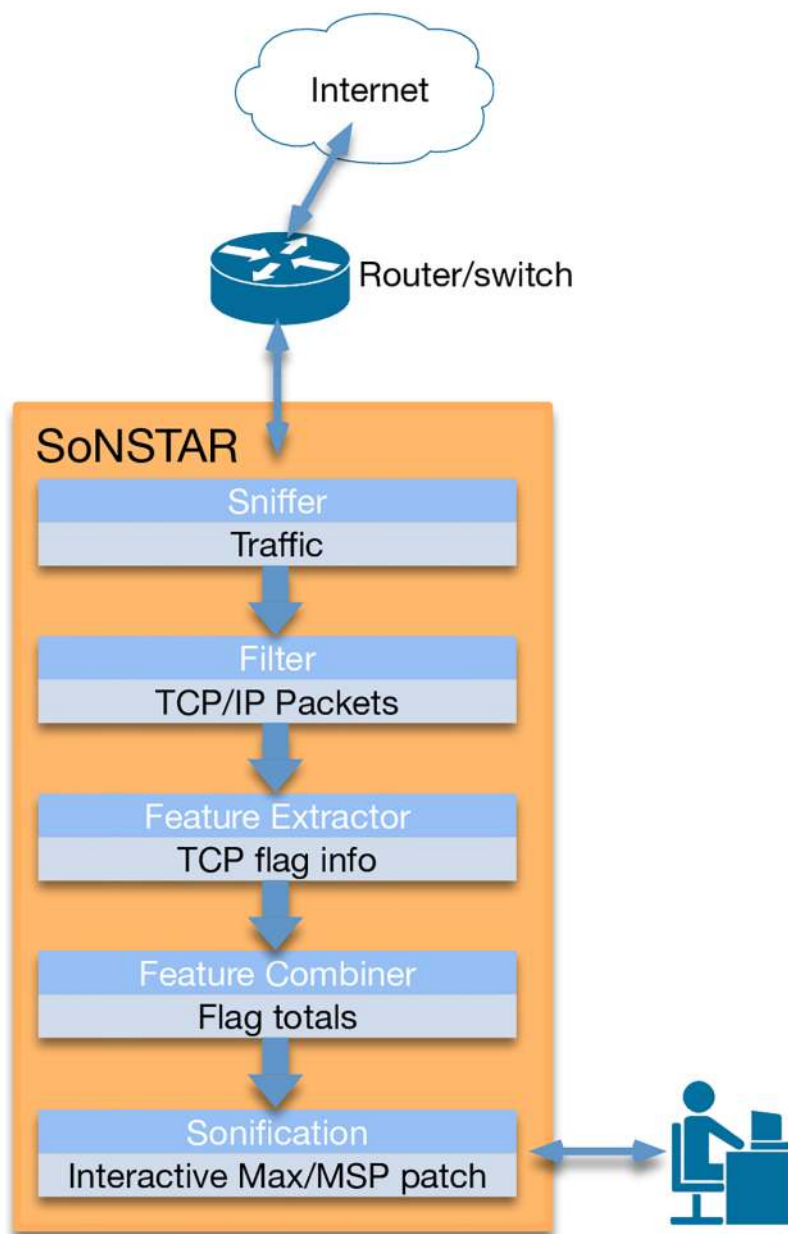


Fig 1. SoNSTAR architecture. The major components of the system.

<https://doi.org/10.1371/journal.pone.0195948.g001>

window most suits the events he or she wishes to target. Moreover, the time window value would affect the thresholds of event conditions which allow the user to develop an intuitive sense of the link between features and events. The only constraint is that a very large network might require more than one instance of SoNSTAR to be installed on lower hierarchy routers, each representing a smaller sub-network, to avoid overloading the processing capabilities of the machine on which SoNSTAR is installed.

The main SoNSTAR algorithm is shown in Algorithm 1. The SoNSTAR system comprises five blocks described below.

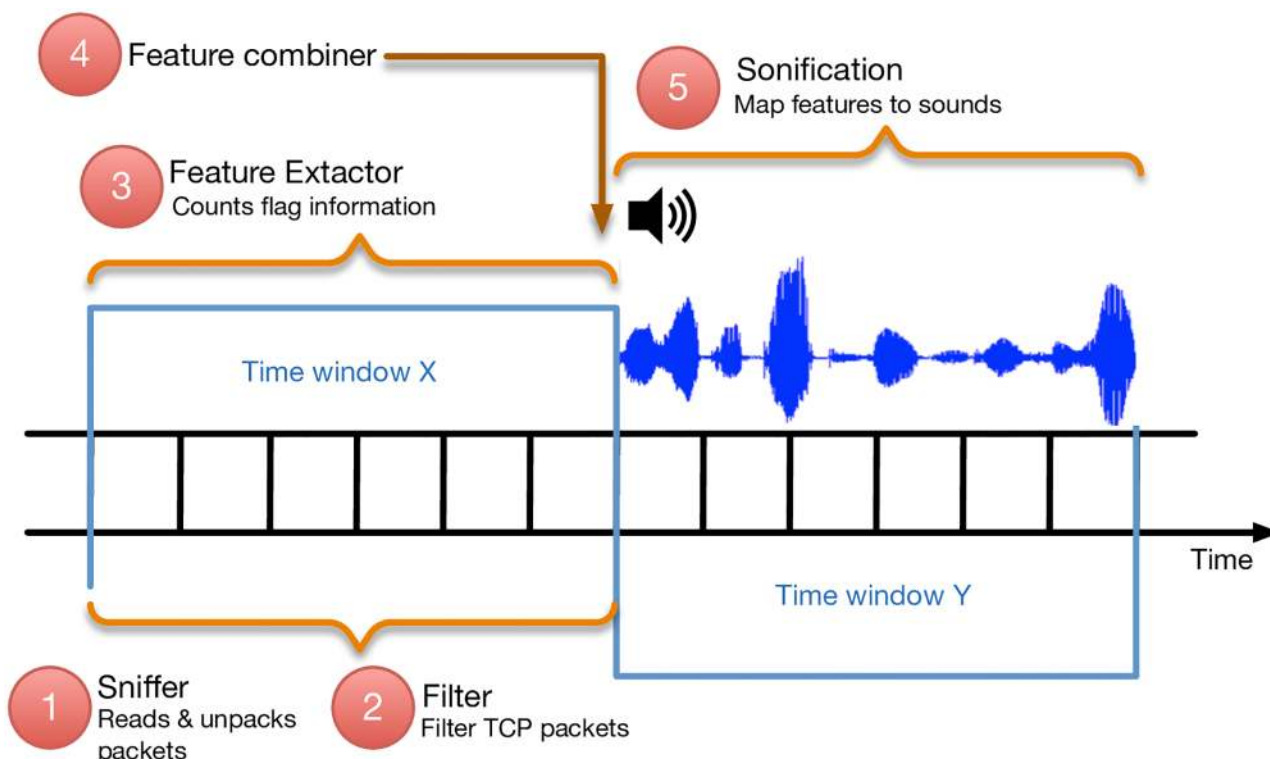


Fig 2. Time window processes. SoNSTAR aggregates flow data across time windows. This figure shows the process timing and sequencing across two time windows, X and Y.

<https://doi.org/10.1371/journal.pone.0195948.g002>

Algorithm 1 SoNSTAR's main algorithm

```

Set Timewindow period
Sniff packet and Get start time
if Packet == arrived then
  Unpack ethernet header
  Extract EtherType
  if EtherType == 0x0800 or 0x86DD then                                ▷ IP packet
    Unpack IP header
    Extract source and destination addresses
    Extract transmission protocol
  else
    Get next packet from the sniffer
  end if
if Protocolnumber == 6 then                                          ▷ TCP packet
  Unpack TCP header
  Extract flags information according to incoming or outgoing
  Count flags status according to incoming or outgoing
  if Timewindowperiod == finished then
    Extract current flag's features
    Extract new features from Features Combiner
    Apply thresholds to selected features
    Send messages to Max/MSP for sonification
  end if
  Get next packet from the sniffer a new Timewindow started
else
  Get next packet from the sniffer

```

```

    end if
else
    Get next packet from the sniffer
end if
Max/MSP Patch
if messages == arrived then
    Play sound of similar messages once
end if

```

Sniffer. The main input to the system is the raw traffic packets passing (incoming and outgoing) through the network and the Sniffer reads these packets in real time.

Filter. The Filter unpacks each ethernet frame, extracting the packet header information, and sending only TCP packets to the Feature Extractor. A TCP/IP packet has an EtherType value of 0x0800 or 0x86DD (denotes IP protocol) and a transmission protocol number of 6.

Feature extractor. Next, the Feature Extractor picks up each TCP packet, checks the flag values, and determines the packet type. If this flow has not been seen before it creates a new Traffic flow and IP flow and sets the counter for the current packet type to 1 for each flow. If the flow already exists the feature extractor increments its packet type counts by 1 according to the packet's direction (incoming or outgoing). This update happens for both flow types (traffic flow and IP flow). At the end of the time window, the set of traffic flows with their packet type counts and the set of IP flows with their packet type counts, in addition to number of traffic flows and number of IP flows are passed to next stage. The full feature sets for both flow and IP-flow packets can be found in the supplemental material (see [S1 Appendix](#)).

At the end of each time window SoNSTAR creates two log file reports consisting of the entire set of Traffic- and IP-flows and their respective packet type counts for any post-hoc inspection and review that may be required.

Feature combiner. The Feature Combiner enables the user to create new features by adding or subtracting particular flags (see [Table 2](#) for some examples). This enables the user to target specific flow events. Some of these combinations could be set according to user needs and understanding of the TCP protocol behaviours and rules. Some could be built over time while listening to and learning about the network environment's behaviours and sounds.

For example, TCP requires the use of specific mechanisms to establish connections between source and destination hosts. An established process is called the three-way handshake. The first step in the process is that the source (S) sends to the destination (D) a TCP packet with the SYN flag set. Next, D replies to S with a packet with the SYN and ACK flags set to acknowledge and accept the connection. Finally, S sends a packet to D with the ACK flag set

Table 2. Feature combinations.

| Feature Combination | Definition | Normal range |
|---------------------|-------------------------------------|--------------|
| FC 1 | SYN-out-IP – SYN-ACK-in-IP | ≤ 4 |
| FC 2 | SYN-in-IP – SYN-ACK-out-IP | ≤ 4 |
| FC 3 | FIN-out-IP – FIN-in-IP | ≤ 9 |
| FC 4 | FIN-in-IP – FIN-out-IP | ≤ 9 |
| FC 5 | SYN-in-IP + SYN-out-IP – FIN-out-IP | ≥ RST-out-IP |
| FC 6 | SYN-in-IP + SYN-out-IP – FIN-in-IP | ≥ RST-in-IP |
| FC 7 | FIN-in-IP – FIN-out-IP – RST-out-IP | ≤ 9 |
| FC 8 | FIN-out-IP – FIN-in-IP – RST-in-IP | ≤ 9 |

Illustration of the way packet counts (by flag type) are combined to denote specific feature combinations.

<https://doi.org/10.1371/journal.pone.0195948.t002>

indicating acknowledgment of the agreement. In this way the handshake process is successfully completed and the connection is established. After the exchange of data and at the end of the connection, either side will terminate the connection by sending a TCP packet with the FIN flag set [48]. Therefore, each flag's status gives us information about the flow and changes in flag status represent what is happening in the network.

At this stage of SoNSTAR design we have created some new features from previous IP flow features (see [S1 Appendix](#)) provided by the Feature Extractor (see [Table 2](#)). All of these features are now available for sonification.

Sonification. The final block in the system is Sonification. To make sense of the sonification we must assign sounds according to event conditions and thresholds and according to the understanding of flag status mechanisms for both flow types. Knowledge of these events could be learned over time while listening to the network environment, tuning the thresholds and experimenting with conditions to target particular behaviours and exploring log files.

Through development of this design recorded natural sounds have been assigned to various features to create a network soundscape environment. By operating SoNSTAR and listening to sounds and manipulating event conditions and tuning thresholds, new events and feature combinations can be defined (such as those new features listed in [Table 2](#)). Threshold values could vary according to the characteristics of the network being monitored.

Of the many features that could be monitored for intrusion detection purposes, some are truly useful and some are less significant, and may indeed be useless. A standalone IDS might generate many false positives or could ignore an anomaly (false negative) depending on its settings. There is no clear analytical model that provides the basis for a mathematical formula to precisely describe the input-output relationship [49]. Therefore, using SoNSTAR would provide that missing understanding of the decisions made by an IDS and allow its user to gain knowledge through monitoring the real behaviour and events of the flows within the traffic.

Every network is a unique environment. Relationships between features are important when applying sounds to the events chosen. This is what gives SoNSTAR a real advantage in exploring a network environment because the understanding of the traffic environment can be improved by taking into account feature relations. The idea behind using different recorded sounds from nature and human-made sounds to represent the network environment is to transform the experience into an interactive soundscape environment. The sounds generated express the behaviour of flows and their deviations from the normal state in order to increase situational awareness.

Representational techniques

Sonic representation is a challenge because of the huge volumes of traffic passing through each connection in the network. Each connection has a high potential number of flows depending on the nature of that connection and its purpose. SoNSTAR reduces the complexity of representing huge volumes of traffic by two methods. The first considers IP flows rather than traffic flows. A number of traffic flows could exist between any two hosts as each traffic flow is specific to a single port number. IP flows are not concerned with port numbers so the number of flows between any two hosts is reduced to one for sonification purposes (see [Fig 3](#)). In the second method SoNSTAR maintains counts of the packet types for each traffic flow to update the soundscape at a user-specified interval. Since network traffic consists of a number of flows which can be similar in their condition, so similar flows can be expressed once so that there is no repetition of the same sound. By doing this we have reduced the number of flow events that need to be sonified.

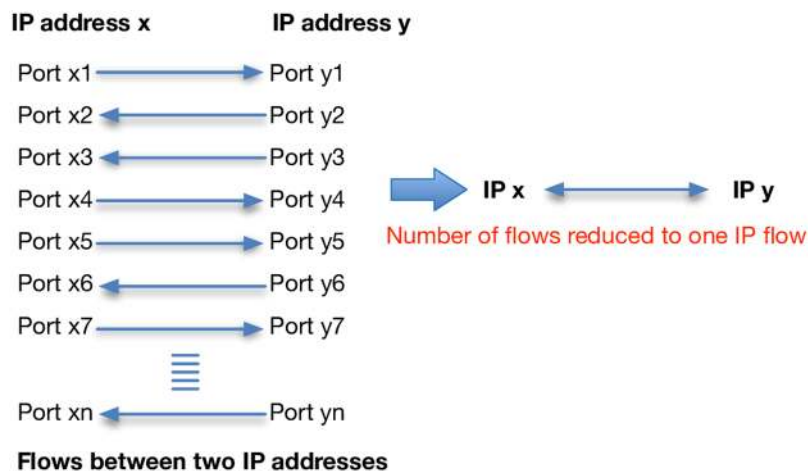


Fig 3. Conflation of multiple traffic flows to one IP flow. Seven traffic flows between different ports on the same sending and receiving hosts are reduced to a single IP flow.

<https://doi.org/10.1371/journal.pone.0195948.g003>

Recorded sounds (such as birds or rain) represent discrete events by playing a single natural sound every time the event occurs. The sounds chosen are diverse in nature and easily distinguishable by the listener.

Tuning the system. One begins to tune SoNSTAR for a particular network by starting with the three-way handshake mechanism and assigning it to a chosen sound. Then, each flow event of interest is mapped to a sound and then its frequency of occurrence is listened to over time in order to get a sense of its impact on network behaviour. The event's feature threshold value can then be adjusted to suit. It was noted during development that certain events tend to occur normally in every network or dataset. Network mechanisms and activities which are confirmed as normal events were mapped to sounds from a forest birds collection. Forest birds were used because they represent the normal state of a forest. Sounds that do not belong to the normal state of a forest were then used to represent rarer, unusual, or anomalous events. Fig 4 shows an example sonification of IP flows to represent network traffic state. (Listen to the file [S1 Audio](#) of normal traffic sonification in the supplemental material).

Events which are outside the normal range are represented according to the main flag type that caused that event. Sound representation is divided into five categories. The first category of network states represents ongoing events related to SYN or SYN-ACK packets (or combinations thereof) and is represented by weather-related sounds of rain or water. For example, the soundscape changes from rain to heavy rain to rain and thunder according to the number of packets that caused the event.

The second category represents ongoing FIN, ACK, URG, PSH or NULL packets (or combinations thereof) and is represented by animals or unusual birds. The third category represents ongoing RST events and is mapped to wind sounds. For example, when any host sends a high number of RST packets the sonification reflects the change in network state by playing a wind on grass sound; if the RST packet changes usual behaviour in relation to SYN and FIN packets, a heavy wind sound is played. The fourth category represents ongoing events related to traffic- or IP-flow counters and is represented by sounds of fire in the woods. The fifth category represents ongoing events confirmed as normal conditions and is represented by usual forest birds forming ongoing background sounds. Fig 5 shows an example of event representation in SoNSTAR.

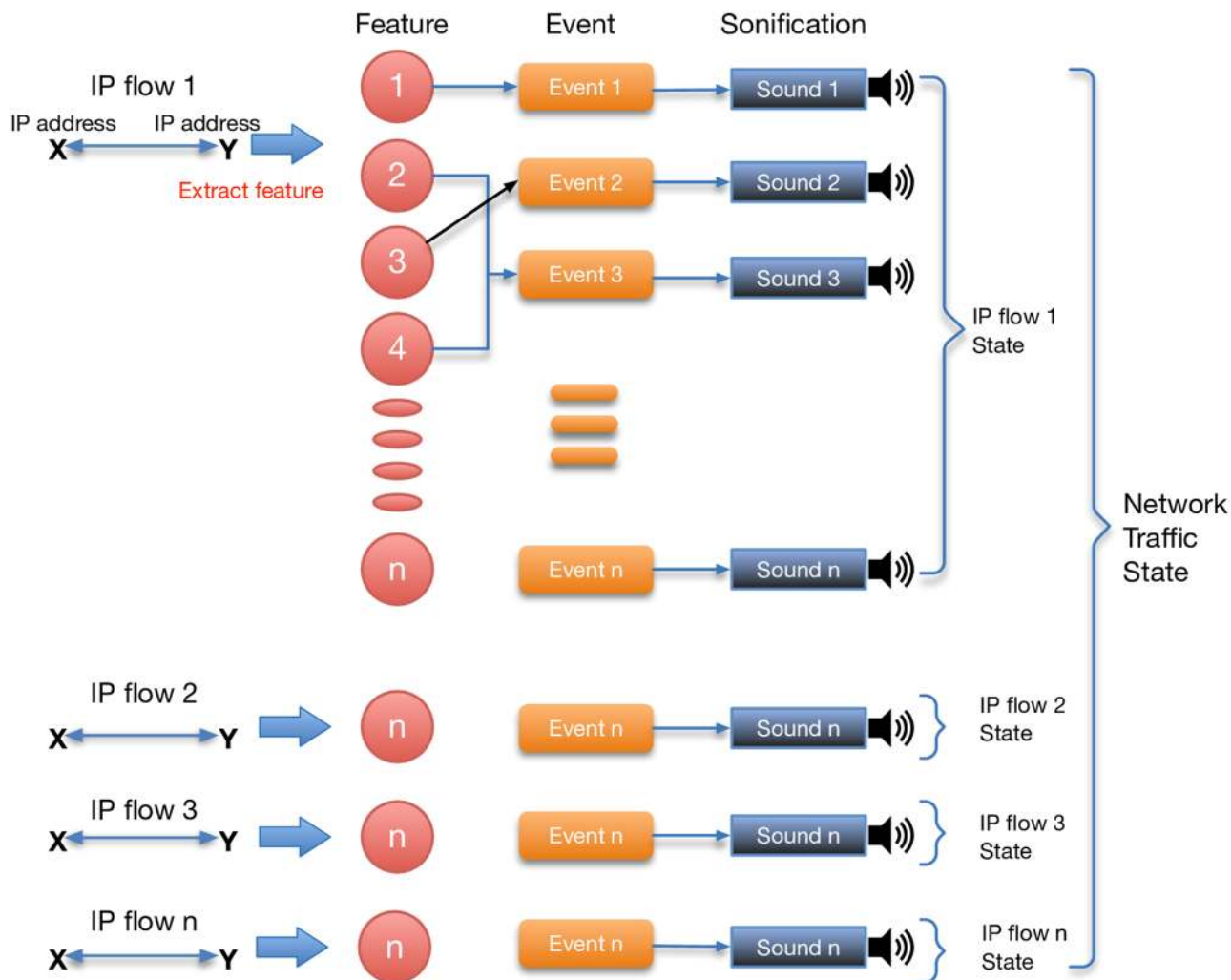


Fig 4. IP flow representation. Illustration of multiple IP flows containing a range of different events and even combinations are mapped to different sounds resulting in a sonic representation of the overall traffic state.

<https://doi.org/10.1371/journal.pone.0195948.g004>

For a better representation, incoming and outgoing events of the same type are represented such that incoming events are given worrying and louder sounds (more dangerous- or urgent-sounding versions of the sounds) than outgoing events which are quieter which are mapped to non-alarming animal sounds. Furthermore, it was observed that several events tend to occur together or in specific sequences for particular types of attack. Therefore, their sequenced sounds were examples of behaviours that were learned as SoNSTAR was used to begin exploring network traffic. It is posited that the information about network traffic provided by SoNSTAR can assist with the recognition of anomalies, both of known and unknown (not previously encountered) types.

Sound design and representation depend very much on personal taste and targeted behaviour. SoNSTAR provides the user with a choice of sound sets (e.g., forest, weather, and animals sounds, or even human-made ones) and assigns sounds according to the event features the user wishes to monitor.

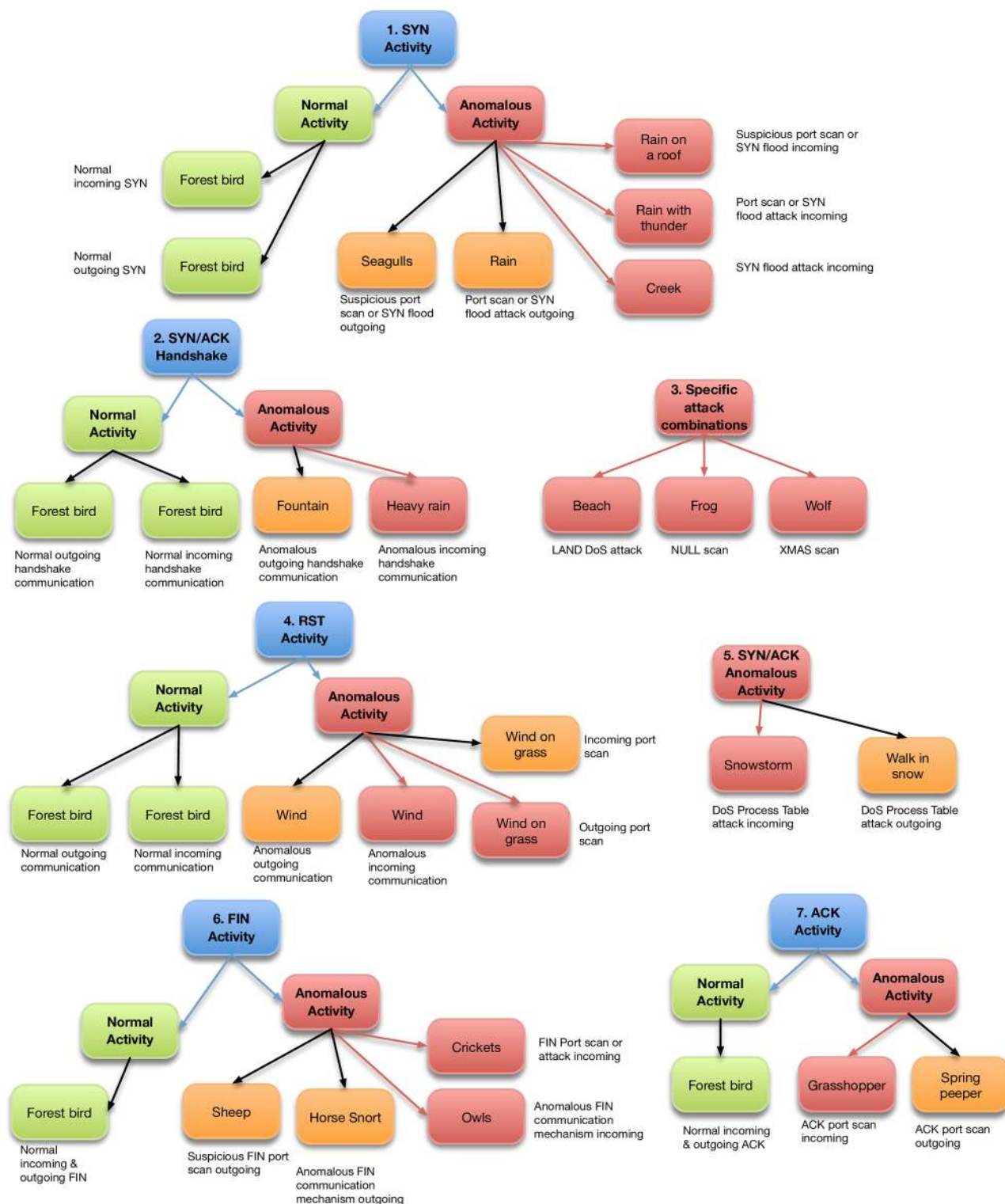


Fig 5. Event representation. Illustration of different events (identified the main flag type) being mapped to discrete sounds the SoNSTAR soundscape.

<https://doi.org/10.1371/journal.pone.0195948.g005>

SoNSTAR features sound mappings

The features used for sonification are aggregation counts of the flag status of each flow type in the traffic. For each feature thresholds are set such that sounds are generated only when the counts exceed the threshold. Users can select the thresholds appropriate to their network environment. A set of default mappings was created based on an understanding of TCP protocol theory and running SoNSTAR several times whilst carrying out simulated attacks in order to learn about traffic features. The thresholds used do not represent *a priori* fixed rules. However, experimenting with these thresholds requires an understanding of the flag relations in the TCP protocol. Network traffic is not static and what can be normal traffic behaviour in one context could be malicious elsewhere, and thus the expected numbers of flows could vary depending on the purpose of the network. The default event-to-sound mappings are listed in [Table 3](#).

Table 3. Feature-to-sound mappings.

| No | Feature Conditions | Sound |
|----|---|---------------|
| 1 | SYN-in-IP <30 and SYN-ACK-out-IP >0 and ACK-in-IP >0 and RST-out-IP <10 | Forest bird |
| 2 | SYN-in-IP >10 and SYN-in-IP <30 and PSH-ACK-out-IP <6 | Rain on roof |
| 3 | SYN-in-IP >20 and SYN-ACK-out-IP <10 | Rain on roof |
| 4 | SYN-in-IP >300 and SYN-ACK-out-IP <50 and SYN-in-IP <1000 | Thunder |
| 5 | SYN-in-IP >1000 | Creek |
| 6 | SYN-out-IP <10 and SYN-ACK-in-IP <2 and ACK-out-IP <3 | Rain |
| 7 | SYN-out-IP <30 and SYN-ACK-in-IP >0 and ACK-out-IP >0 and RST-in-IP <10 | Forest bird |
| 8 | ACK-in-IP >1 and the rest of IP flow feature equal 0 | Seagulls |
| 9 | ACK-out-IP >1 and the rest of IP flow feature equal 0 | Loon |
| 10 | FIN-in-IP >9 and FIN-in-IP >SYN-out-IP and FIN-in-IP >SYN-in-IP and FC-4 >10 | Cricket |
| 11 | FIN-in-IP <50 and (FIN-in-IP <= SYN-out-IP or FIN-in-IP <= SYN-in-IP) | Forest bird |
| 12 | FIN-out-IP >9 and FIN-out-IP >SYN-out-IP and FIN-out-IP >SYN-in-IP and FC-3 >10 | Sheep |
| 13 | FC-7 >9 | Owl |
| 14 | FC-7 <10 | Forest bird |
| 15 | FC-8 >9 | Horse snort |
| 16 | FC-8 <10 | Forest bird |
| 17 | NULL-in-IP >0 | Frog |
| 18 | NULL-out-IP >0 | Frog |
| 19 | URG-PSH-FIN-in-IP >0 | Wolf |
| 20 | URG-PSH-FIN-out-IP >0 | Wolf |
| 21 | LAND-in-IP >0 | Beach |
| 22 | LAND-out-IP >0 | Beach |
| 23 | RST-in-IP >25 and ACK-in-IP <250 | Wind on grass |
| 24 | RST-out-IP >25 and ACK-out-IP <250 | Wind on grass |
| 25 | FC-1 >4 | Fountain |
| 26 | FC-1 <5 | Forest bird |
| 27 | FC-2 >4 | Heavy rain |
| 28 | FC-2 <5 | Forest bird |
| 29 | RST-out-IP >5 and FC-5 <RST-out-IP and ACK-out-IP <7 | Wind |
| 30 | RST-in-IP >5 and FC-6 <RST-in-IP and ACK-in-IP <7 | Wind |
| 31 | SYN-ACK-out >20 | Snow storm |
| 32 | SYN-ACK-in >20 | Walk in snow |
| 33 | (Traffic Flow Counter) >1000 | Fire |
| 34 | (IP Flow Counter) >600 | Fire |

A selection of event conditions and their corresponding sounds.

<https://doi.org/10.1371/journal.pone.0195948.t003>

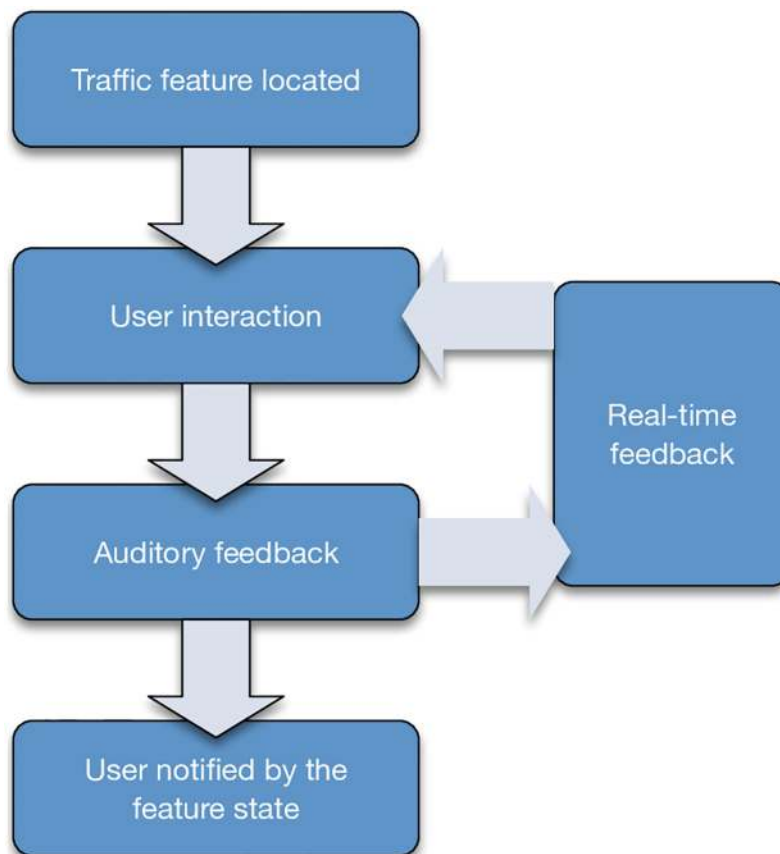


Fig 6. Interactive sonification. Model showing the interactive nature of the SoNSTAR sonification.

<https://doi.org/10.1371/journal.pone.0195948.g006>

SoNSTAR interactive sonification

SoNSTAR is an interactive sonification system. Users may change the time window period, manipulate features and thresholds and re-assign sounds, and then restart with the new settings online. The level of each event sound can be adjusted independently with a slider control and can even be muted if desired. Any sound can be assigned to any chosen flow event in real time enabling the user to re-design the sound environment completely. Fig 6 shows the SoNSTAR interactive sonification model.

Users interact with SoNSTAR according to their understanding of the sound generated by the network traffic environment so as to increase their situational awareness. SoNSTAR enables the user to interact immediately with the system and its traffic to identify anomalous behaviours. Hunt and Hermann advise that sonification designers should respect ‘the bindings between physical actions and acoustic reactions that we have been familiar with since birth’ [50, p. 295]. In a network environment this could mean that we would expect the sounds to change when the system is under attack and we expect networks to behave differently when they are under more stress. SoNSTAR uses multiple natural and man-made sounds to create the soundscape environment. When choosing the sounds, the natural reactions of users to the sounds is taken into consideration in order to allow users to sense and feel the network environment in relation to their own experience in the real world. SoNSTAR allows users to change sounds and create their own preferred acoustic environment in order to enable them

to choose the most suitable sounds which convey to them the state of the network in a maximally meaningful way. SoNSTAR transforms all of the network traffic into a rich auditory field that envelops the listener in a goal-driven exploratory methodology where the network traffic is first filtered and the user is left only with the specific features that they chose.

Experimental work and results

A user study was conducted to investigate the monitoring of network behaviour by participants using SoNSTAR and, in particular, to evaluate SoNSTAR as a complement to existing system security tools. Three experimental conditions were investigated: 1) audio feedback only using SoNSTAR, 2) visual feedback only using the Snort IDS, and 3) audio and visual feedback together (SoNSTAR and Snort).

SoNSTAR's current design is able to extract TCP, UDP and ICMP protocol packet information. In this experiment only TCP and ICMP packet header information was extracted with ICMP packet data being used to detect ping activities. The sound of a woodpecker sound was assigned to ICMP ping activity.

Network design

The experiment was conducted using two virtual networks running on the Virtualbox software. The first network was installed on a macOS 10.10.5 workstation with a 3.7 GHz quad-core processor, 16 GB 1866 MHz DDR3 ECC RAM and a 27-inch (2560 x 1440) display. This virtual network comprised four machines (Ubuntu 64-bit, Windows Server 64-bit, Kali Linux Debian 64-bit and macOS 10.11) in addition to the host machine.

The second virtual network was installed on a MacBook Pro running macOS 10.10.5 with a 2.5 GHz Intel core i7 processor, 16 GB 1600MHz DDR3 RAM and a 15.4-inch (2880 x 1800) Retina display. This network contained three machines (two Kali Linux Debian 64-bit installations and a Fedora 24 64-bit machine) in addition to the host machine. Fig 7 shows the network environment design.

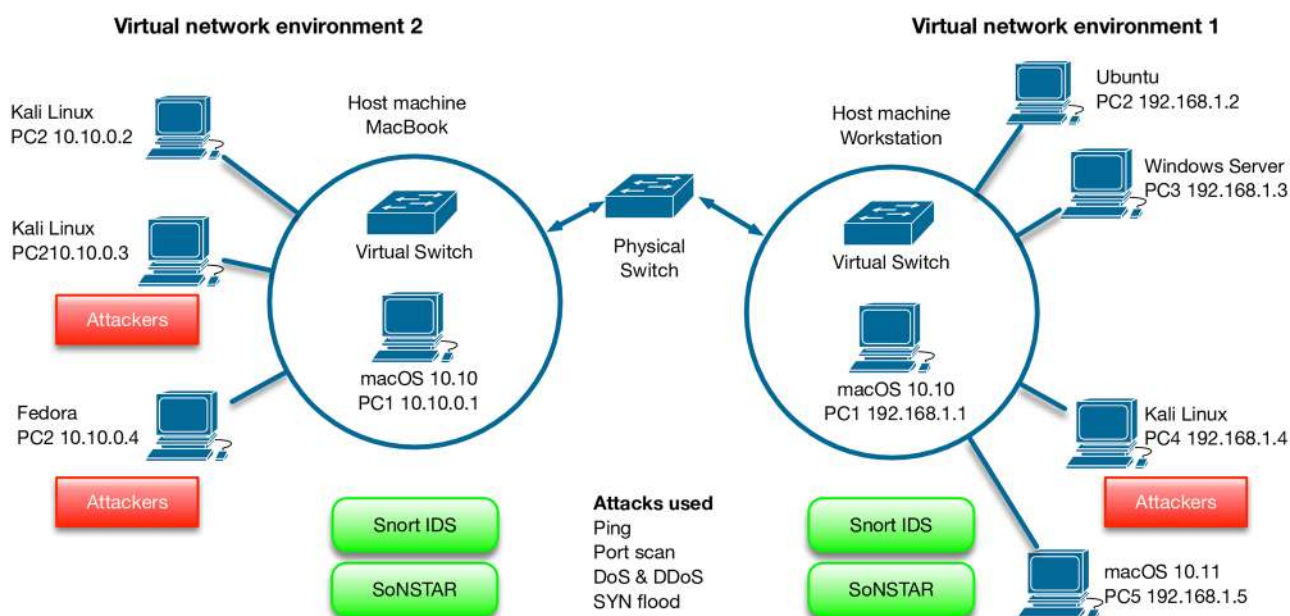


Fig 7. Virtual network environment. The virtual network environment design used in the experiment.

<https://doi.org/10.1371/journal.pone.0195948.g007>

These two virtual networks were connected through a router provided by Northumbria University. SoNSTAR and the Snort IDS were installed on both networks allowing each network to attack the other, and each machine to attack the other machines within its own local virtual network.

Participants

A call for participants was sent through the university email system to all MSc and PhD computer science and engineering students. Sixteen students responded to the email and ten participants (seven male, three female) were able to devote the time needed to participate in the study which took place in September 2016. All ten participants completed the study. All of the participants were aged from 25 to 45 years and were PhD and MSc students at the university (eight from the Department of Computer and Information Sciences and two from the Department of Mechanical Engineering). All participants had good knowledge of the use of computers and information technology and general knowledge about computer network security.

Experimental design

Each participant performed a network monitoring task under each of the three experimental conditions (audio only, visual only, audio-visual). Each task required participants to detect either 3 or 4 out of 7 attacks.

The participants were assigned randomly to use Snort or SoNSTAR (five participants each) and then to use them together. At the end of each task performance was calculated based on the number of true positive (TP), true negatives (TN), false positives (FP) and false negatives (FN), where:

- **TP:** the number of events which are correctly identified. The case was positive and was detected by the user as positive.
- **FP:** the number of events which are incorrectly identified. The case was negative but was detected by the user as positive.
- **TN:** the number of events which are correctly rejected. The case was negative and was detected by the user as negative.
- **FN:** the number of events which are incorrectly rejected. The case was positive but was detected by the user as negative.

These variables are then used to calculate several metrics to assess the effectiveness of SoNSTAR as follows.

The *recall* metric (also known as the true positive rate) indicates the proportion of positives which are correctly detected by participants and is given by:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The *precision* is the number of true positives amongst all the reported positives:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The F-measure is a weighted harmonic mean of the precision and recall [51, p.1147]:

$$F = 2 \cdot \left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \right)$$

The *accuracy* metric indicates the proportion of correct identifications of all instances:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The true negative rate (TNR) indicates the proportion of negatives that are correctly identified, such as the percentage of network events which are correctly identified as not occurred.

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

The false positive rate (FPR) indicates the proportion of positives that are incorrectly identified, such as the percentage of network events which are incorrectly identified as occurred.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

The false negative rate (FNR) indicates the proportion of negatives that are incorrectly identified.

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}}$$

Snort's detection rules were set to the defaults provided by the `snort.conf` file. SoNSTAR was set to the sound mappings presented in Table 3.

Four categories of behaviour were used in this experiment as follows:

- **Traffic:** using the Internet, such as playing a YouTube video.
- **Ping:** using an ICMP ping.
- **Port scan:** four types—SYN, Null, Xmas and FIN port scans.
- **DoS, DDoS** including first, SYN flood as type; and second, DDoS using spoofed IP addresses performed from the three machines in the virtual network.

These behaviours were performed using a normal terminal, Nmap scanner and Hping3 commands. The supplemental material contains the files S2, S3, S4, S5, S6, S7 and S8 Audio which are the SoNSTAR sonifications of the attacks used in this experiment.

Materials

Ethical approval for the research project was granted by Northumbria University under record number "RE-EE-13-140616-539ec5986be50".

Before beginning the experiment, each participant was given an informed consent declaration to sign (see S5 Appendix in the supplemental material). Following the giving of consent each participant completed the three tasks using a macOS 10.10.5 workstation equipped with a 27-inch monitor and Sony MDR-7506 Professional headphones.

A questionnaire was given to each participant. The questionnaire can be found in the supplemental material (see S2 Appendix). The first section elicited general participant information such as gender, level of education, speciality, department and year of study. The second section was a table for reporting detected malicious activities for the monitoring detection tasks for

the three task conditions. The questionnaire provided two tick boxes in front of each type of attack for the three task conditions.

The third section included evaluation of monitoring workload; upon completion of each experimental task participants completed the NASA-Task Load Index (TLX) assessment [52] to measure their performance workload. This includes mental demand, temporal demand, physical demand, performance, effort and frustration rates. Also there were extra ratings for detection confidence, ease of use, visual fatigue and sound fatigue included in the evaluation of both tools. For each of these rates, the participant had to provide an assessment rating on a scale of 0 to 10.

The participants were then asked to choose their preferred condition (SoNSTAR, Snort, or both together). They were also requested to provide their evaluations of Snort and SoNSTAR on a scale of 0 to 5 where 5 denotes the most positive assessment. Participants could also provide feedback about this experiment in the final section.

The training and guidelines sheet included a table containing the seven chosen attack types for the experiment as well as the detection of text in snort and detection sounds in SoNSTAR written in front of each attack. The first column contained the attack category, the second column the attack type name, the third column text expected by Snort and the fourth column a description of the sound events for each attack, explaining the extra understanding those sounds provide.

Procedure

Participants were informed that they would take the role of a network administrator to protect against malicious activities. The explanation of the experiment included three sections (one for each task condition) and where should they fill in the appropriate section for each task condition. The participants' virtual network computers were switched on and some music and YouTube videos were started to generate normal traffic across the network.

Participants were trained for about five minutes in the basics of the Snort IDS and another five minutes on SoNSTAR before starting each task condition. The rules for administration to protect their network and servers against attacks and malicious activities were explained including the seven specific attacks used in experiment. It was also explained how each task condition involves concentration and high attention for long periods to detect attacks in their early stages.

Training involved only the seven attack types used in this experiment. Participants were provided with a training and guidelines sheet and then trained on how Snort would show the detected attacks, and how Snort provides text warnings for each type. The seven attacks were demonstrated in real time. SoNSTAR training involved the same attacks but this time participants were provided with headphones and using the training and guidelines sheet they were asked to listen to the attacks one by one in real time. Any questions raised by participants were answered. They were not informed that SoNSTAR was a project under development so as to eliminate the effect of such knowledge on the results.

Each participant was provided with the questionnaire to fill in the outcomes for the three tasks. Five participants were assigned to the SoNSTAR condition for seven minutes first and then to the Snort condition for another seven minutes. They were then assigned to use both SoNSTAR and Snort for another seven minutes.

The other five participants were assigned to the Snort condition for seven minutes and then the SoNSTAR condition for another seven minutes. Then they were assigned to use both SoNSTAR and Snort for another seven minutes. This was done to eliminate the effect of using any one condition first.

Table 4. TP, TN, FP and FN.

| Metrics | Snort | SoNSTAR | Snort and SoNSTAR |
|---------|-------|---------|-------------------|
| TP | 31 | 33 | 30 |
| TN | 31 | 33 | 38 |
| FP | 7 | 4 | 2 |
| FN | 0 | 0 | 0 |

<https://doi.org/10.1371/journal.pone.0195948.t004>

During each period, the participants' networks received three or four real-time attacks. However, they were not informed about the number of malicious activities that could be expected. During each task, each participant was asked to continue speaking and were asked for more information about their understanding of security in order to affect their concentration to some extent.

Directly after completing each task, participants had to answer the rest of the questions regarding the Monitoring Evaluation Tasks for each tool. At the end of the experiment, the participants were asked to tick which was considered the best for them to use, Snort or SoNSTAR or both together. Then they were requested to complete the rest of the questionnaire.

Results

Several results are extracted from the questionnaire data as follows.

TP, TN, FP and FN results. The results for the three conditions are shown in [Table 4](#) as extracted from the questionnaire data. The results were calculated for the three conditions to assess SoNSTAR's capabilities as part of the situational awareness process. Based on these results, various metrics are calculated to evaluate the SoNSTAR sound design and the usability of the system.

The metrics calculated from the base variables are shown in [Table 5](#). The results show a maximum recall of 100% for the three state conditions. Meanwhile, the TNR was higher when using SoNSTAR (89.19%) compared to Snort (81.58%). However, when participants used both together this rose to 95%. The FPR was higher when using Snort (18.42%) than SoNSTAR (10.81%). However, when participants used both together this decreased to 5%.

Accuracy was calculated for the three state conditions used in the experiment. Accuracy of recognition was highest when using both Snort and SoNSTAR together at 97.14%. SoNSTAR alone maintained higher accuracy than Snort alone, at 94.29% and 89.86% respectively.

Precision was also calculated for the three state conditions used in this experiment. Precision of recognition was highest when using both Snort and SoNSTAR together at 93.75%.

Table 5. Evaluation results.

| Metrics | Snort | SoNSTAR | Snort & SoNSTAR |
|-----------|--------|---------|-----------------|
| Recall | 100% | 100% | 100% |
| Precision | 81.58% | 89.19% | 93.75% |
| F-measure | 89.86% | 94.29% | 96.77% |
| Accuracy | 89.86% | 94.29% | 97.14% |
| TNR | 81.58% | 89.19% | 95% |
| FPR | 18.42% | 10.81% | 5% |
| FNR | 0% | 0% | 0% |

<https://doi.org/10.1371/journal.pone.0195948.t005>

Table 6. NASA-Task Load Index results.

| No | Task Load Index | Snort | SoNSTAR |
|----|----------------------|-------|---------|
| 1 | Mental Demand Rate | 58% | 45% |
| 2 | Temporal Demand Rate | 65% | 31% |
| 3 | Physical Demand Rate | 28% | 24% |
| 4 | Performance Rate | 82% | 92% |
| 5 | Effort Rate | 41% | 19% |
| 6 | Frustration Rate | 71% | 36% |

<https://doi.org/10.1371/journal.pone.0195948.t006>

Table 7. Additional SoNSTAR evaluation (index results).

| No | Task Load Index | Snort | SoNSTAR |
|----|------------------------------|-------|---------|
| 1 | Detection Confidence Rate | 88% | 90% |
| 2 | Ease of Use Rate | 86% | 96% |
| 3 | Visual or Sound Fatigue Rate | 59% | 40% |

<https://doi.org/10.1371/journal.pone.0195948.t007>

Table 8. Additional SoNSTAR evaluation (preference results).

| Index | Snort | SoNSTAR | Both together |
|-------------|-------|---------|---------------|
| Best to use | 10% | 30% | 60% |

<https://doi.org/10.1371/journal.pone.0195948.t008>

SoNSTAR alone maintained higher precision again compared to Snort at 89.19% and 81.58% respectively.

The F-measure was highest when using both Snort and SoNSTAR together at 96.77%. SoNSTAR achieved a higher F-measure than Snort at 94.29% and 89.86% respectively.

NASA-Task Load Index results. The NASA-Task Load Index results are shown in [Table 6](#).

Additional evaluation results. Additional SoNSTAR evaluation results are shown in [Table 7](#).

[Table 8](#) shows participants' opinions about whether using Snort and SoNSTAR alone or together would be best for monitoring.

[Table 9](#) shows participants' opinions about Snort and SoNSTAR from horrible (H) to fantastic (F).

The most remarkable feedback was that a participant advised that if it is possible to add a visual panel showing the name, colour and image of what is generating the sound, this would help to distinguish the recorded sound in order to facilitate learning and confirmation. For example, some initially had trouble distinguishing between the sound of rain and the sound of rain on a roof, and a visual key might have helped to learn the sounds quicker.

Table 9. Horrible to fantastic evaluation.

| Tool | H (100%) | H (50%) | Average | F (50%) | F (100%) |
|---------|----------|---------|---------|---------|----------|
| Snort | 0 | 0 | 40% | 10% | 50% |
| SoNSTAR | 0 | 10% | 30% | 0% | 60% |

<https://doi.org/10.1371/journal.pone.0195948.t009>

Discussion

These experimental results clearly show improvements in monitoring when using sonification compared to the visual method only. Although the detection rate was 100% for the three state conditions, we can still see improvements in the accuracy, precision and F-measure scores for the sonification conditions. Although the training of participants was very brief and the computer security background of most of the participants was basic, they were able to use both systems well in a short time.

SoNSTAR can fill the gap between the network operator and the traffic environment by providing an auditory link between traffic behaviour and the operator's mind. The experiment showed that users with a basic knowledge of computer network principles can use SoNSTAR to learn how flows are behaving inside network traffic and to recognise the typical combinations of the packet types within the flows. To ease the learning process, SoNSTAR generates log files holding counts of the packets within flows in addition to messages that indicate which flow event generated which specific sound. The IP-flow and traffic-flow log files together can explain in detail how the packet counts of the two flow types are related (see [S3 Appendix](#) IP flow log file and [S4 Appendix](#) Traffic flow log file in the supplemental material). These reports support the user in understanding the links between flag states, packet counts, flow counts, and the resultant sequences of sounds. The project repository [\[53\]](#) contains the source code and Max/MSP patches necessary to install and run SoNSTAR together with sample data files, example output stored in audio files, and instructions on how to use the system.

Some behaviours will create multiple sounds. For example, when performing a SYN scan, the attacker will send a number of packets with the SYN flag set to 1 to a number of targeted ports. If the port is open the receiver would send back a packet with the SYN flag set to 1 and the ACK flag set to 1 as a reply to accept the connection. The attacker either sends back a packet with the FIN flag set to 1 to cut the connection (the TCP half-open scan type) or sends two packets, the first with the ACK flag set to 1 to confirm the connection and then the second packet with the FIN flag set to 1 to cut the connection (the TCP connect scan type). If the port is closed the receiver would send back a packet with the RST flag set to 1, and if there is no response it means that the port is filtered.

As we set SoNSTAR to default settings, as soon as it receives many SYN in an IP flow, SoNSTAR will play the rain-on-a-roof sound and this would tell the user that an unusual number of SYN packets is arriving. If the TCP handshake was not correct, that event would generate a heavy rain sound which would tell the user that there is a problem with connecting to a specific IP address. If the number of SYN packets was high, SoNSTAR will play a thunder sound and this will tell the user that someone is scanning a large number of the system ports of a specific IP address. If the number was huge it would be considered a DoS attack and the sound of fire would be played. If the scanned system started to send out RST packets, SoNSTAR would play the sound of wind, confirming that it is a scan attack. This is a complex process, but SoNSTAR would deal with any changes in behaviour and play sets of sounds according to what events are happening in the network. The user could identify any new behaviour according to the set of sounds played.

Using such a tool to explore and tune a network is important due to the different natures of networks and the different expected behaviours with different thresholds. For example, this tool could be used to tune IDS settings to look for new features and events which could be used to identify threats on a particular network. Using SoNSTAR to draw a normal base line for a specific network behaviour would help to make rules and thresholds for specific environment which will raise the situation awareness in general.

Another advantage that SoNSTAR provides is that it generates log files which could help any user to learn and confirm how are packet types counts of IP flow effects recorded sound played and to evaluate theoretically any event and ideas of any new feature (Review the [S3 Appendix](#) IP flow log file in the supplemental material which contains a SYN scan). SoNSTAR could help users and network students to explore network protocols and to learn more about network traffic. The use of SoNSTAR would enable them to think directly about the logic of any behaviour in network traffic and would give them the opportunity to express their own ideas and to test and learn from them.

Using SoNSTAR reduced mental demand, temporal demand, effort and frustration rates compared to using Snort (visual tool) and this would be more obvious if the monitoring took place over long periods. This confirms that the use of SoNSTAR increases the situational awareness of the operator as it not only gives a warning of the attacks, but gives more than that so that the operator listens to what happens to the network directly allowing them to make links between behaviours and sounds. This relationship enables him or her to understand what is happening accurately by distinguishing the sounds and their meanings leading to comprehension of network state allowing decisions to be made directly without reference to any alarms reporting or going through long forensic processes to understand the behaviour.

Conclusion and further work

This study indicated that using sonification improved the monitoring process, even for people who have only basic knowledge of network monitoring. Using sound reduced the overall work load experienced by users. Participants were able to recognise and comprehend behaviours and decide which attack was performed which demonstrated it is possible for a human operator to quickly learn about the network environment in a way that results in increased situational awareness. Although the system could be evaluated manually by going through its log files, this experiment evaluated the practicality of using sonification in live monitoring tasks. The results suggest that using SoNSTAR to explore new event sand features would bring benefits for network monitoring in general and intrusion detection systems in particular.

There is more work that can be done to further improve attack detection and situational awareness. For example, using SoNSTAR's feature extractor and combiner it was possible to define a number of traffic features. Discovering these from scratch is a time consuming task so public databases of features and feature combinations could be created. Developing a language for describing these discovered features and events based together with the log files to explain how these feature can be used could assist future designers of intrusion detection systems.

Vertical flow behaviour occurs when a single host receives many flows across a range of ports from a single source host. Horizontal flow behaviour occurs when a defined range of ports receives flows across a defined range of destination hosts. A major contribution of this research was to reduce the complexity of huge volumes of traffic in order to be comprehensively sonified by using IP flows in detecting network behaviour, especially vertical flow behaviours. The sonic mapping of network events based on packet type counts has not been seen before. We suggest further work to aimed at the identification of horizontal flow behaviours.

Other important future work consists of developing a method to create features for UDP, ICMP, IRC and other protocols for sonification by SoNSTAR so that larger quantities of common traffic can be covered. Also SoNSTAR has very high potential to represent SCADA (Supervisory Control and Data Aquisition) systems because of their unique nature. If their normal behaviour patterns are tested, zero-day vulnerability mitigation can be increased.

Further study could be done to evaluate how SoNSTAR can be used to help students and computer professionals learn about network traffic.

Supporting information

S1 Appendix. SoNSTAR: Flow and IP flow feature information array contents. This file shows the contents of the feature information arrays for Traffic flows and IP flows.
(PDF)

S2 Appendix. Experiment: The questionnaire. This file shows the contents of the questionnaire used in this experiment for evaluation.
(PDF)

S3 Appendix. SoNSTAR: IP flow log file. Text file consists IP flows information. Each row consists first, time window number, then the IP flow number within current time window, then host A IP address, host B IP address, and then the feature counts sorted in the same sequence in [S1 Appendix](#).
(TEXT)

S4 Appendix. SoNSTAR: Traffic flow log file. Text file consists of traffic flow information. Each row comprises the time window number, the Traffic flow number within the current time window, the host A IP address, host B IP address, then host A port number, host B port number, and then the feature counts sorted in the same sequence as [S1 Appendix](#).
(TEXT)

S5 Appendix. Experiment: The informed consent form. This file shows the contents of the consent form used in this experiment.
(PDF)

S1 Audio. Normal traffic behaviour. SoNSTAR normal events sounds audio file.
(AIF)

S2 Audio. FIN behaviour. SoNSTAR FIN scan audio file. The scan was performed using hping3.
(AIF)

S3 Audio. Xmas behaviour. SoNSTAR heavy Xmas scan audio file. The scan was performed using Nmap.
(AIF)

S4 Audio. NULL behaviour. SoNSTAR low NULL scan audio file. The scan was performed using hping3.
(AIF)

S5 Audio. NULL behaviour. SoNSTAR heavy NULL scan audio file. The scan was performed using hping3.
(AIF)

S6 Audio. SYN behaviour. SoNSTAR heavy full connection SYN scan audio file. The scan was performed using Nmap.
(AIF)

S7 Audio. Ping behaviour. SoNSTAR SYN-Flood-DoS audio file. Sounds of SYN flood attack behaviour for denial of service purposes; performed using hping3.
(AIF)

S8 Audio. Ping behaviour. SoNSTAR Null-DDoS audio file. DDoS (distributed denial of service) using null packet type; performed using hping3.
(AIF)

Author Contributions

Formal analysis: Mohamed Debashi.

Methodology: Mohamed Debashi, Paul Vickers.

Software: Mohamed Debashi, Paul Vickers.

Supervision: Paul Vickers.

Writing – original draft: Mohamed Debashi, Paul Vickers.

Writing – review & editing: Mohamed Debashi, Paul Vickers.

References

1. Hildebrandt T. Towards Enhancing Business Process Monitoring with Sonification. In: Business Process Management Workshops. Springer; 2014. p. 529–536.
2. Kramer G, Walker B, Bonebright T, Cook P, Flowers JH, Miner N, et al. Sonification Report: Status of the Field and Research Agenda. ICAD/NSF; 1999. Available from: <http://sonify.psych.gatech.edu/publications/pdfs/1999-NSF-Report.pdf>.
3. Endsley MR. Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors: The Journal of the Human Factors and Ergonomics Society. 1995; 37(1):32–64.
4. Vickers P, Laing C, Fairfax T. Sonification of a Network's Self-Organized Criticality for Real-Time Situational Awareness. Displays. 2017; 47:12–24.
5. Angerman WS. Coming Full Circle with Boyd's OODA Loop Ideas: An Analysis of Innovation Diffusion and Evolution [Masters Thesis]. Airforce Institute of Technology. Wright-Patterson AFB, Ohio, USA; 2004.
6. Lakkaraju K, Yurcik W, Lee AJ. NVisionIP: Netflow Visualizations of System State for Security Situational Awareness. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security. ACM; 2004. p. 65–72.
7. Vickers P. Sonification for Process Monitoring. In: Hermann T, Hunt AD, Neuhoof J, editors. The Sonification Handbook. Berlin: Logos Verlag; 2011. p. 455–492. Available from: <http://sonification.de/handbook/index.php/chapters/chapter18/>.
8. Ballora M, Giacobe NA, Hall DL. Songs of Cyberspace: An Update on Sonifications of Network Traffic to Support Situational Awareness. In: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2011. vol. 8064. International Society for Optics and Photonics; 2011. p. 80640P.
9. Yang J, Hunt A. Real-Time Sonification of Biceps Curl Exercise Using Muscular Activity and Kinematics. In: Vogt K, Andreopoulou A, Goudarzi V, editors. Proceedings of the 21st International Conference on Auditory Display (ICAD 2015). Graz, Austria: Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts Graz (KUG); 2015. p. 289–293.
10. Di Pietro R, Mancini LV. Intrusion Detection Systems. Springer; 2008.
11. Fairfax T, Laing C, Vickers P. Network Situational Awareness: Sonification & Visualization in the Cyber Battlespace. In: Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance. Advances in Digital Crime, Forensics, and Cyber Terrorism (ADCFCT). IGI Global; 2014. p. 334–349. Available from: <http://nrl.northumbria.ac.uk/15649/>.
12. Jajodia S, Liu P, Swarup V, Wang C, editors. Cyber Situational Awareness. Springer; 2010.
13. Wikipedia contributors. Internet protocol suite—Wikipedia, The Free Encyclopedia; 2018. Available from: https://en.wikipedia.org/w/index.php?title=Internet_protocol_suite&oldid=829116103 [cited 9-March-2018].
14. Worrall D. Realtime Sonification and Visualisation of Network Metadata. In: Vogt K, Andreopoulou A, Goudarzi V, editors. Proceedings of the 21st International Conference on Auditory Display (ICAD 2015). Graz, Austria: Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts Graz (KUG); 2015. p. 337–339.

15. Panchen S, Phaal P, McKee N. InMon Corporation's SFlow: A Method for Monitoring Traffic in Switched and Routed Networks; 2001. <https://tools.ietf.org/html/rfc3176>.
16. Mancuso VF, Greenlee ET, Funke G, Dukes A, Menke L, Brown R, et al. Augmenting Cyber Defender Performance and Workload Through Sonified Displays. *Procedia Manufacturing*. 2015; 3:5214–5221. <https://doi.org/10.1016/j.promfg.2015.07.589>
17. Wireshark Foundation. Wireshark Tool; 2017. <https://www.wireshark.org/>.
18. Vickers P. Nuson-SOCS: Self-Organized Criticality Sonification; 2016. <https://github.com/paulvickers/nuson-SOCS>.
19. Wolf KE, Fiebrink R. SonNet: A Code Interface for Sonifying Computer Network Data. In: NIME, 13—13th International Conference on New Interfaces for Musical Expression; 2013. p. 503–506.
20. Rutz HH, Vogt K, Höldrich R. The SysSon Platform: A Computer Music Perspective of Sonification. In: Vogt K, Andreopoulou A, Goudarzi V, editors. ICAD 15: Proceedings of the 21st International Conference on Auditory Display. Graz, Austria: Institute of Electronic Music and Acoustics (IEM), University of Music and Performing Arts Graz (KUG); 2015. p. 188–196.
21. Giot R, Courbe Y. InteNtion—Interactive Network Sonification. In: Nees MA, Walker BN, Freeman J, editors. Proceedings of the 18th International Conference on Auditory Display (ICAD 2012). Georgia Institute of Technology; 2012. p. 235–236.
22. Ballora M, Hall DL. Do You See What I Hear: Experiments in Multi-Channel Sound and 3D Visualization for Network Monitoring. In: Cyber Security, Situation Management, and Impact Assessment II; and Visual Analytics for Homeland Defense and Security II. vol. 7709. International Society for Optics and Photonics; 2010. p. 77090J.
23. Ballora M, Cole RJ, Kruesi H, Greene H, Monahan G, Hall DL. Use of Sonification in the Detection of Anomalous Events. In: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2012. vol. 8407. International Society for Optics and Photonics; 2012. p. 84070S.
24. Kimoto M, Ohno H. Design and Implementation of Stetho Network Sonification System. In: Proceedings of the 2002 International Computer Music Conference; 2002. p. 273–279.
25. The Tcpdump Group. Tcpdump; 2017. <https://github.com/the-tcpdump-group/tcpdump>.
26. Chafe C, Leistikow R. Levels of Temporal Resolution in Sonification of Network Performance. In: Hii-pakka J, Zacharov N, Takala T, editors. ICAD 2001 7th International Conference on Auditory Display. Espoo, Finland: ICAD; 2001. p. 50–55.
27. Brownlee N, Mills C, Ruth G. Traffic Flow Measurement: Architecture; 1999. <https://tools.ietf.org/html/rfc2722>.
28. Schafer RM. The Tuning of the World. Random House; 1977.
29. Pijanowski BC, Villanueva-Rivera LJ, Dumyahn SL, Farina A, Krause BL, Napoletano BM, et al. Soundscape Ecology: The Science of Sound in the Landscape. *BioScience*. 2011; 61(3):203–216. <https://doi.org/10.1525/bio.2011.61.3.6>
30. Kramer G. Some Organizing Principles for Representing Data with Sound. In: Kramer G, editor. Auditory Display. vol. XVIII of Santa Fe Institute, Studies in the Sciences of Complexity Proceedings. Reading, MA: Addison-Wesley; 1994. p. 185–222.
31. Wolf KE, Gliner G, Fiebrink R. A Model for Data-Driven Sonification Using Soundscapes. In: Proceedings of the 20th International Conference on Intelligent User Interfaces Companion. IUI Companion'15. Atlanta, GA: ACM; 2015. p. 97–100.
32. Kramer G. Preface. In: Kramer G, editor. Auditory Display. vol. XVIII of Santa Fe Institute, Studies in the Sciences of Complexity Proceedings. Reading, MA: Addison-Wesley; 1994. p. xxiii–xxxviii.
33. Shah K, Bohacek S, Broido A. Feasibility of Detecting TCP-SYN Scanning at a Backbone Router. In: American Control Conference, 2004. Proceedings of the 2004. vol. 2. IEEE; 2004. p. 988–995.
34. Balram S, Wiscy M. Detection of TCP SYN Scanning Using Packet Counts and Neural Network. In: 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems; 2008. p. 646–649.
35. Barford P, Kline J, Plonka D, Ron A. A Signal Analysis of Network Traffic Anomalies. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement. ACM; 2002. p. 71–82.
36. Bowman WB. System and Method for Detecting Fraudulent Network Usage Patterns Using Real-Time Network Monitoring; 1997. US Patent 5,627,886/ Google Patents.
37. Srinivasan T, Vijaykumar V, Chandrasekar R. A Self-Organized Agent-Based Architecture for Power-Aware Intrusion Detection in Wireless Ad-Hoc Networks. In: Computing & Informatics, 2006. ICOCI'06. International Conference On. IEEE; 2006. p. 1–6.

38. Alshammari R, Zincir-Heywood AN. Can Encrypted Traffic Be Identified Without Port Numbers, IP Addresses and Payload Inspection? *Computer Networks*. 2011; 55(6):1326–1350. <https://doi.org/10.1016/j.comnet.2010.12.002>
39. Stallings W, Brown L. *Computer Security: Principles and Practice*. 3rd ed. Pearson Education Limited; 2014.
40. Bernaille L, Teixeira R. Early Recognition of Encrypted Applications. In: Uhlig S, Papagiannaki K, Bonaventure O, editors. *Passive and Active Network Measurement*. Berlin, Heidelberg: Springer; 2007. p. 165–175.
41. Criado R, Flores J, Hernández-Bermejo B, Pello J, Romance M. Effective Measurement of Network Vulnerability Under Random and Intentional Attacks. *Journal of Mathematical Modelling and Algorithms*. 2005; 4(3):307–316. <https://doi.org/10.1007/s10852-005-9006-1>
42. Carrier B. Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers. *International Journal of Digital Evidence*. 2003; 1(4):1–12.
43. Malandrino D, Mea D, Negro A, Palmieri G, Scarano V. NeMoS: Network Monitoring with Sound. In: Brazil E, Shinn-Cunningham B, editors. *Proceedings of the 2003 International Conference on Auditory Display*, Boston, MA, USA. Georgia Institute of Technology. Boston, MA: ICAD; 2003. p. 251–254.
44. Neuhoff JG, Wayand J, Kramer G. Pitch and Loudness Interact in Auditory Displays: Can the Data Get Lost in the Map? *Journal of Experimental Psychology: Applied*. 2002; 8(1):17. PMID: [12009173](https://pubmed.ncbi.nlm.nih.gov/12009173/)
45. Bass T. Intrusion Detection Systems and Multisensor Data Fusion. *Communications of the ACM*. 2000; 43(4):99–105. <https://doi.org/10.1145/332051.332079>
46. Baier G, Hermann T, Sahle S, Stephani U. Sonified Epileptic Rhythms. In: Stockman T, Nickerson LV, Frauenberger C, Edwards ADN, Brock D, editors. *ICAD 2006—the 12th Meeting of the International Conference on Auditory Display*. London, UK; 2006. p. 148–151.
47. Cycling '74 Max/MSP; 2018. <https://cycling74.com/products/max>.
48. Schuba CL, Krsul IV, Kuhn MG, Spafford EH, Sundaram A, Zamboni D. Analysis of a Denial of Service Attack on TCP. In: *Proceedings of the 1997 IEEE Symposium on Security and Privacy*. IEEE; 1997. p. 208–223.
49. Mulkamala S, Sung A, Abraham A. *Cyber Security Challenges: Designing Efficient Intrusion Detection Systems and Antivirus Tools*. Vemuri V Rao, *Enhancing Computer Security with Smart Technology* (Auerbach, 2006). 2005; p. 125–163.
50. Hunt A, Hermann T. Interactive Sonification. In: Hermann T, Hunt AD, Neuhoff J, editors. *The Sonification Handbook*. Berlin: Logos Verlag; 2011. p. 273–298.
51. Liu L, Özsu MT, editors. *Encyclopedia of Database Systems*. Springer Berlin, Heidelberg, Germany; 2009.
52. Hart SG, Staveland LE. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*. 1988; 52:139–183.
53. Debashi M, Vickers P. Nuson-SoNSTAR: Sonification of Networks for SiTuational AwaReness; 2017. <https://github.com/nuson/SoNSTAR>.